Computer Algebra Independent Integration Tests

Summer 2023 edition

4-Trig-functions/4.3-Tangent/105-4.3.4.2-a+b-tan-^m-c+d-tan-^n-A+B-tan+C-tan^2-

Nasser M. Abbasi

September 5, 2023 Compiled on September 5, 2023 at 7:57pm

Contents

1 Introduction **3**

2 detailed summary tables of results **21**

3 Listing of integrals **67**

4 Appendix **2207**

CHAPTER 1

INTRODUCTION

1.1 Listing of CAS systems tested 4
1.2 Results 5
1.3 Time and leaf size Performance 8
1.4 Performance based on number of rules Rubi used 10
1.5 Performance based on number of steps Rubi used 11
1.6 Solved integrals histogram based on leaf size of result 12
1.7 Solved integrals histogram based on CPU time used 13
1.8 Leaf size vs. CPU time used 14
1.9 list of integrals with no known antiderivative 15
1.10 List of integrals solved by CAS but has no known antiderivative 15
1.11 list of integrals solved by CAS but failed verification 15
1.12 Timing 16
1.13 Verification 16
1.14 Important notes about some of the results 16
1.15 Design of the test system 19

This report gives the result of running the computer algebra independent integration test.The download section in on the main webpage contains links to download the problems in plain text format used for all CAS systems. The number of integrals in this report is [171]. This is test number [105].

1.1 Listing of CAS systems tested

The following are the CAS systems tested:

1. Mathematica 13.3.1 (August 16, 2023) on windows 10.
2. Rubi 4.16.1 (Dec 19, 2018) on Mathematica 13.3 on windows 10
3. Maple 2023.1 (July, 12, 2023) on windows 10.
4. Maxima 5.47 (June 1, 2023) using Lisp SBCL 2.3.0 on Linux via sagemath 10.1 (Aug 20, 2023).
5. FriCAS 1.3.9 (July 8, 2023) based on sbcl 2.3.0 on Linux via sagemath 10.1 (Aug 20, 2023).
6. Giac/Xcas 1.9.0-57 (June 26, 2023) on Linux via sagemath 10.1 (Aug 20, 2023).
7. Sympy 1.12 (May 10, 2023) Using Python 3.11.3 on Linux.
8. Mupad using Matlab 2021a with Symbolic Math Toolbox Version 8.7 on windows 10.

Maxima and Fricas and Giac are called using Sagemath. This was done using Sagemath `integrate` command by changing the name of the algorithm to use the different CAS systems.

Sympy was run directly in Python not via sagemath.

1.2 Results

Important note: A number of problems in this test suite have no antiderivative in closed form. This means the antiderivative of these integrals can not be expressed in terms of elementary, special functions or `Hypergeometric2F1` functions. `RootSum` and `RootOf` are not allowed. If a CAS returns the above integral unevaluated within the time limit, then the result is counted as passed and assigned an A grade.

However, if CAS times out, then it is assigned an F grade even if the integral is not integrable, as this implies CAS could not determine that the integral is not integrable in the time limit.

If a CAS returns an antiderivative to such an integral, it is assigned an A grade automatically and this special result is listed in the introduction section of each individual test report to make it easy to identify as this can be important result to investigate.

The results given in in the table below reflects the above.

System	% solved	% Failed
Rubi	100.00 (171)	0.00 (0)
Mathematica	98.83 (169)	1.17 (2)
Maple	71.35 (122)	28.65 (49)
Fricas	66.08 (113)	33.92 (58)
Mupad	60.23 (103)	39.77 (68)
Giac	49.12 (84)	50.88 (87)
Maxima	49.12 (84)	50.88 (87)
Sympy	36.84 (63)	63.16 (108)

Table 1.1: Percentage solved for each CAS

The table below gives additional break down of the grading of quality of the antiderivatives generated by each CAS. The grading is given using the letters A,B,C and F with A being the best quality. The grading is accomplished by comparing the antiderivative generated with the optimal antiderivatives included in the test suite. The following table describes the meaning of these grades.

grade	description
A	Integral was solved and antiderivative is optimal in quality and leaf size.
B	Integral was solved and antiderivative is optimal in quality but leaf size is larger than twice the optimal antiderivatives leaf size.
C	Integral was solved and antiderivative is non-optimal in quality. This can be due to one or more of the following reasons 1. antiderivative contains a hypergeometric function and the optimal antiderivative does not. 2. antiderivative contains a special function and the optimal antiderivative does not. 3. antiderivative contains the imaginary unit and the optimal antiderivative does not.
F	Integral was not solved. Either the integral was returned unevaluated within the time limit, or it timed out, or CAS hanged or crashed or an exception was raised.

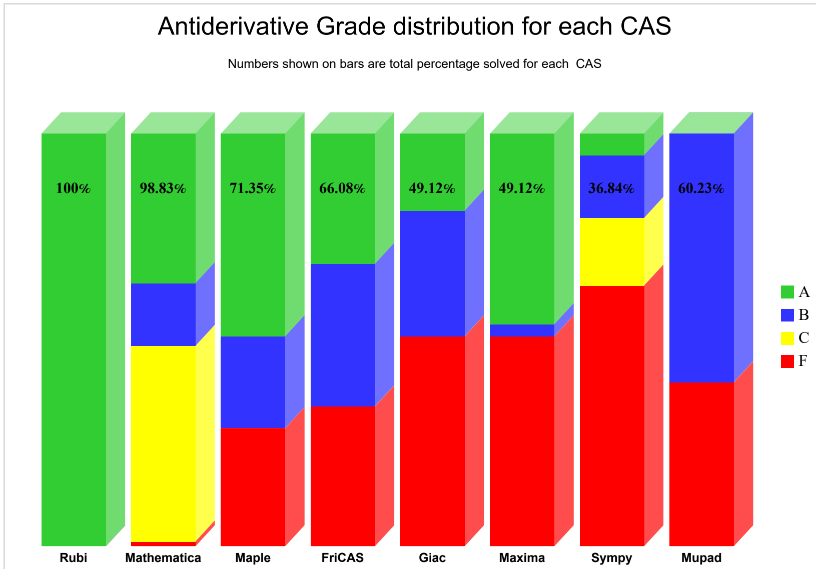
Table 1.2: Description of grading applied to integration result

Grading is implemented for all CAS systems. Based on the above, the following table summarizes the grading for this test suite.

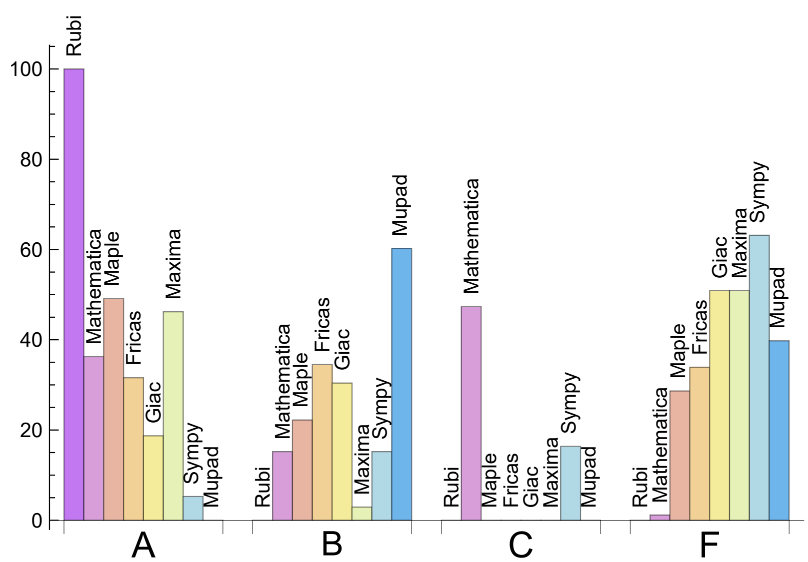
System	% A grade	% B grade	% C grade	% F grade
Rubi	100.000	0.000	0.000	0.000
Maple	49.123	22.222	0.000	28.655
Maxima	46.199	2.924	0.000	50.877
Mathematica	36.257	15.205	47.368	1.170
Fricas	31.579	34.503	0.000	33.918
Giac	18.713	30.409	0.000	50.877
Sympy	5.263	15.205	16.374	63.158
Mupad	0.000	60.234	0.000	39.766

Table 1.3: Antiderivative Grade distribution of each CAS

The following is a Bar chart illustration of the data in the above table.



The figure below compares the grades of the CAS systems.



The following table shows the distribution of the different types of failures for each CAS. There are 3 types failures. The first is when CAS returns the input within the time limit, which means it could not solve it. This is the typical failure and given as **F**.

The second failure is due to time out. CAS could not solve the integral within the 3 minutes time limit which is assigned. This is assigned **F(-1)**.

The third is due to an exception generated, indicated as **F(-2)**. This most likely indicates an interface problem between sagemath and the CAS (applicable only to FriCAS, Maxima and Giac) or it could be an indication of an internal error in the CAS itself. This type of error requires more investigation to determine the cause.

System	Number failed	Percentage normal failure	Percentage timeout failure	Percentage exception failure
Rubi	0	0.00	0.00	0.00
Mathematica	2	100.00	0.00	0.00
Maple	49	26.53	73.47	0.00
Fricas	58	22.41	77.59	0.00
Mupad	68	0.00	100.00	0.00
Giac	87	14.94	85.06	0.00
Maxima	87	33.33	44.83	21.84
Sympy	108	75.00	5.56	19.44

Table 1.4: Failure statistics for each CAS

1.3 Time and leaf size Performance

The table below summarizes the performance of each CAS system in terms of time used and leaf size of results.

Mean size is the average leaf size produced by the CAS (before any normalization). The Normalized mean is relative to the mean size of the optimal anti-derivative given in the input files.

For example, if CAS has **Normalized mean** of 3, then the mean size of its leaf size is 3 times as large as the mean size of the optimal leaf size.

Median size is value of leaf size where half the values are larger than this and half are smaller (before any normalization). i.e. The Middle value.

Similarly the **Normalized median** is relative to the median leaf size of the optimal.

For example, if a CAS has Normalized median of 1.2, then its median is 1.2 as large as the median leaf size of the optimal.

System	Mean time (sec)
Maxima	0.36
Maple	0.40
Giac	1.97
Rubi	2.28
Sympy	3.63
Mathematica	4.88
Mupad	21.11
Fricas	26.31

Table 1.5: Time performance for each CAS

System	Mean size	Normalized mean	Median size	Normalized median
Rubi	323.47	1.00	287.00	1.00
Maxima	375.69	1.35	217.50	1.20
Mathematica	808.14	1.92	290.00	1.26
Giac	1752.95	5.51	492.00	2.30
Sympy	3297.92	13.67	711.00	2.70
Maple	3979.42	11.03	347.00	1.22
Mupad	13887.19	40.23	307.00	1.38
Fricas	15738.06	48.61	505.00	2.10

Table 1.6: Leaf size performance for each CAS

1.4 Performance based on number of rules Rubi used

This section shows how each CAS performed based on the number of rules Rubi needed to solve the same integral. One diagram is given for each CAS.

On the y axis is the percentage solved which Rubi itself needed the number of rules given the x axis. These plots show that as more rules are needed then most CAS system percentage of solving decreases which indicates the integral is becoming more complicated to solve.

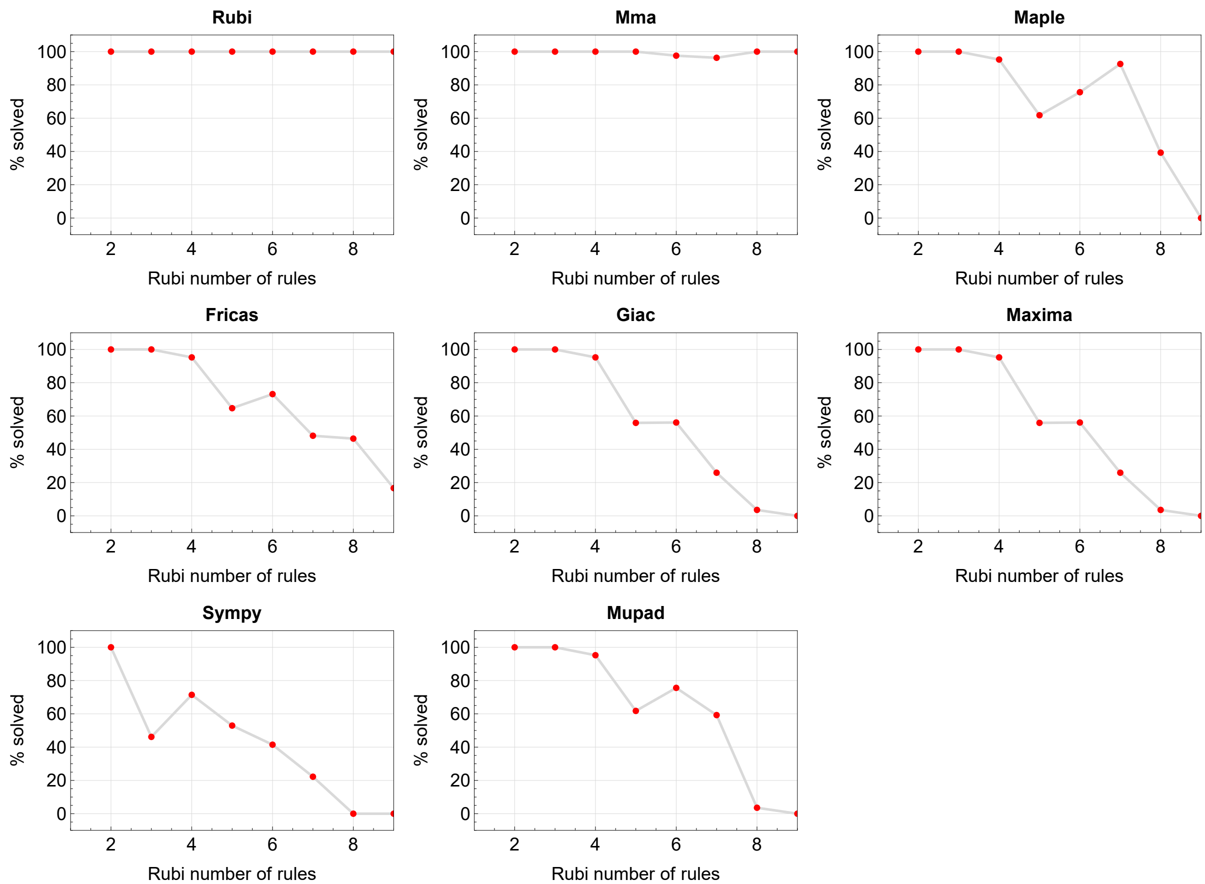


Figure 1.1: Solving statistics per number of Rubi rules used

1.5 Performance based on number of steps Rubi used

This section shows how each CAS performed based on the number of steps Rubi needed to solve the same integral. Note that the number of steps Rubi needed can be much higher than the number of rules, as the same rule could be used more than once.

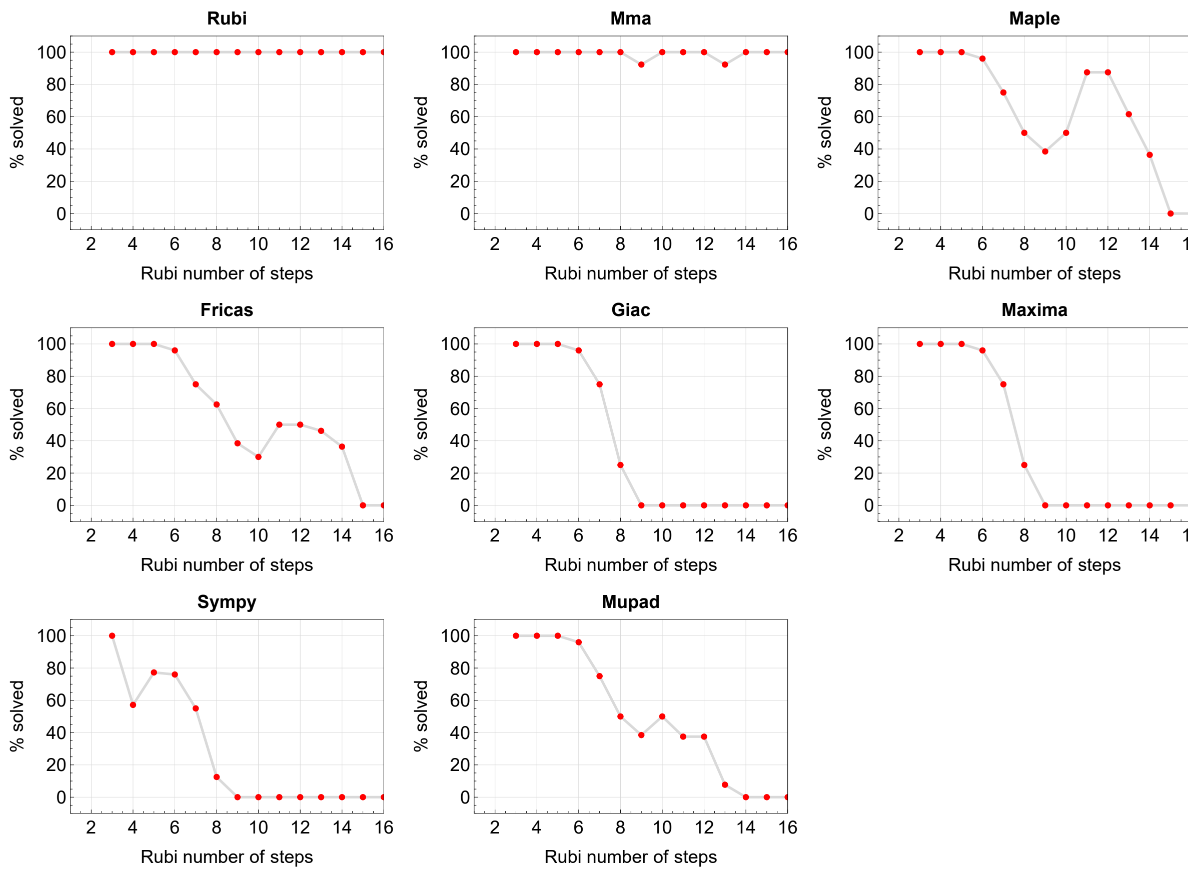


Figure 1.2: Solving statistics per number of Rubi steps used

The above diagram show that the precentage of solved intergals decreases for most CAS systems as the number of steps increases. As expected, for integrals that required less steps by Rubi, CAS systems had more success which indicates the integral was not as hard to solve. As Rubi needed more steps to solve the integral, the solved percentage decreased for most CAS systems which indicates the integral is becoming harder to solve.

1.6 Solved integrals histogram based on leaf size of result

The following shows the distribution of solved integrals for each CAS system based on leaf size of the antiderivatives produced by each CAS. It shows that most integrals solved produced leaf size less than about 100 to 150. The bin size used is 40.

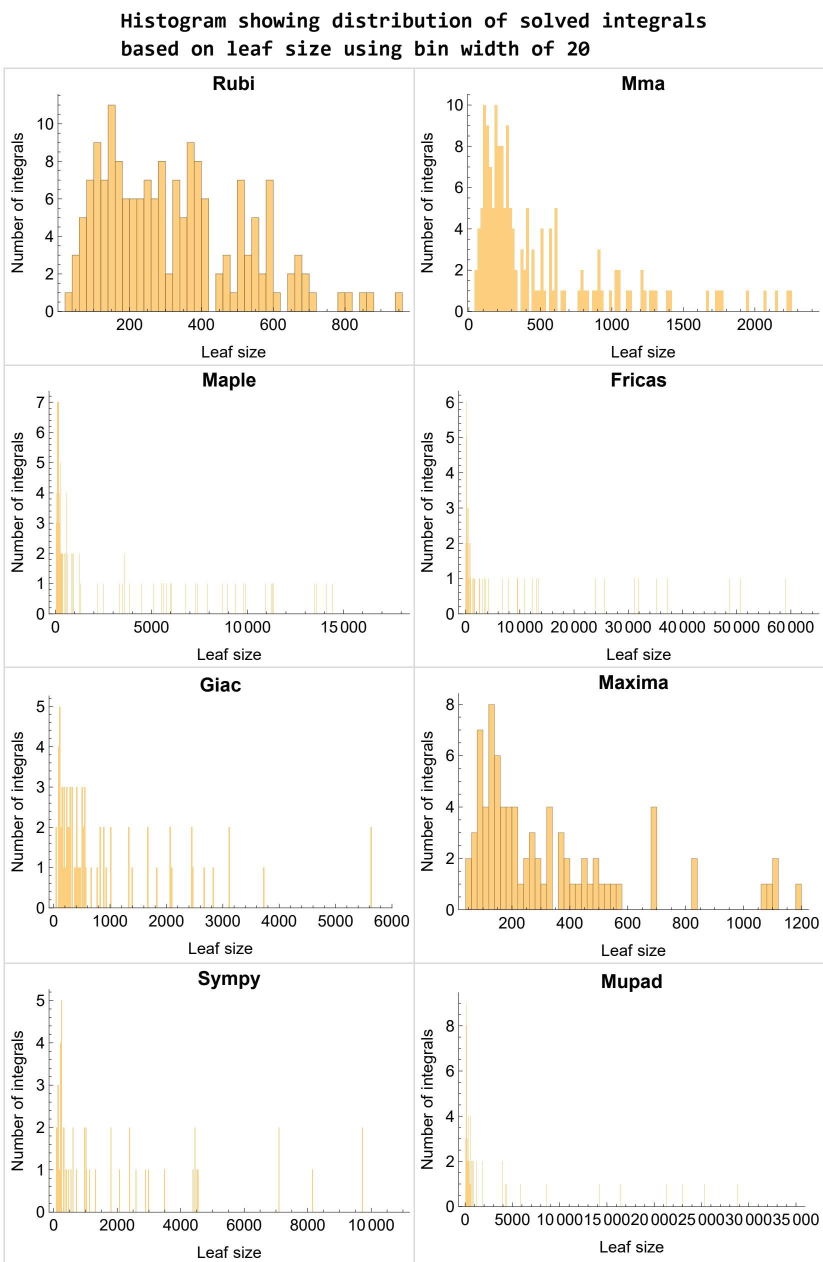


Figure 1.3: Solved integrals based on leaf size distribution

1.7 Solved integrals histogram based on CPU time used

The following shows the distribution of solved integrals for each CAS system based on CPU time used in seconds. The bin size used is 0.1 second.

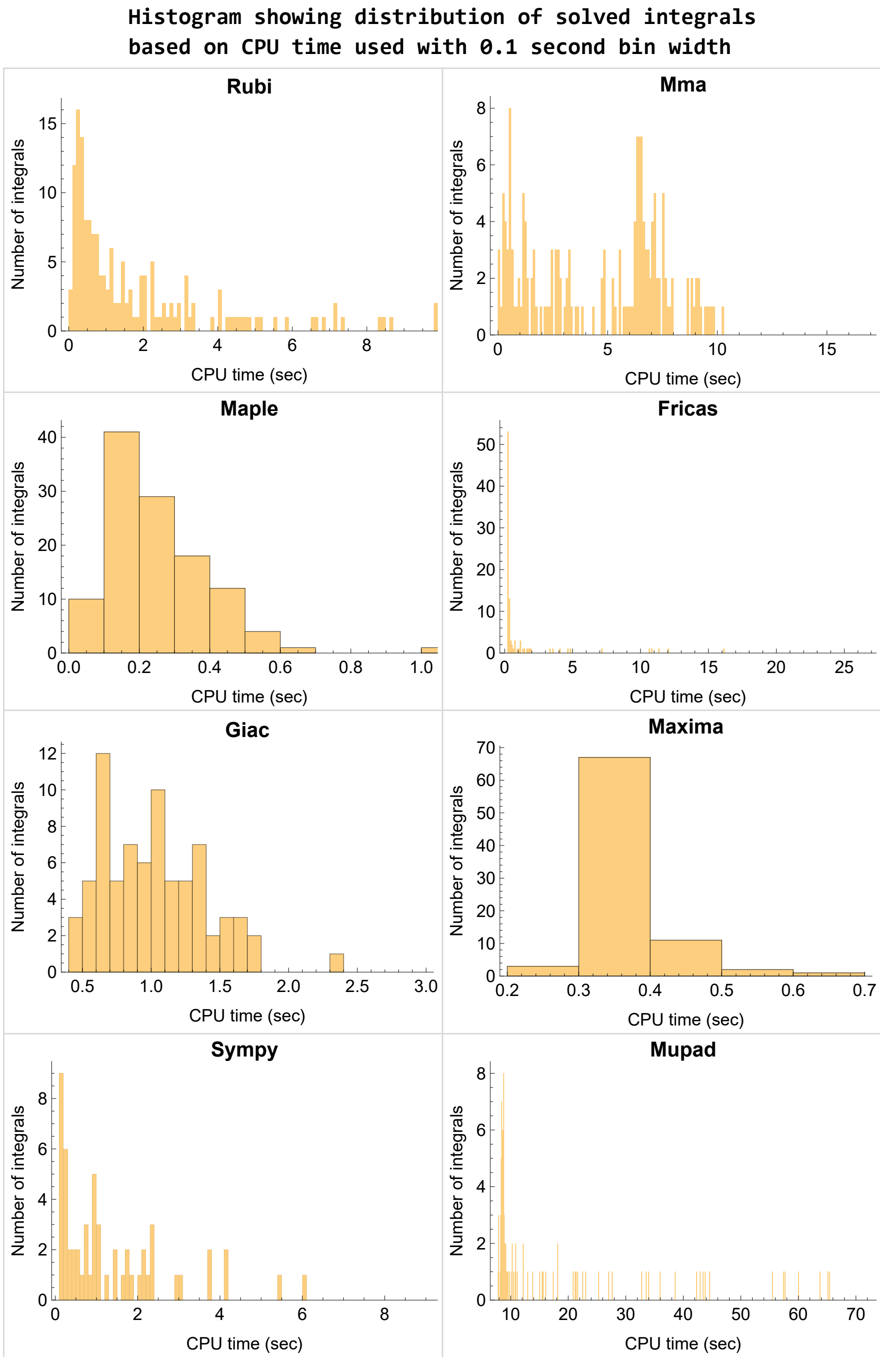


Figure 1.4: Solved integrals histogram based on CPU time used

1.8 Leaf size vs. CPU time used

The following shows the relation between the CPU time used to solve an integral and the leaf size of the antiderivative.

The result for Fricas, Maxima and Giac is shifted more to the right than the other CAS system due to the use of sagemath to call them, which causes an initial slight delay in the timing to start the integration due to overhead of starting a new process each time. This should also be taken into account when looking at the timing of these three CAS systems. Direct calls not using sagemath would result in faster timings, but current implementation uses sagemath as this makes testing much easier to do.

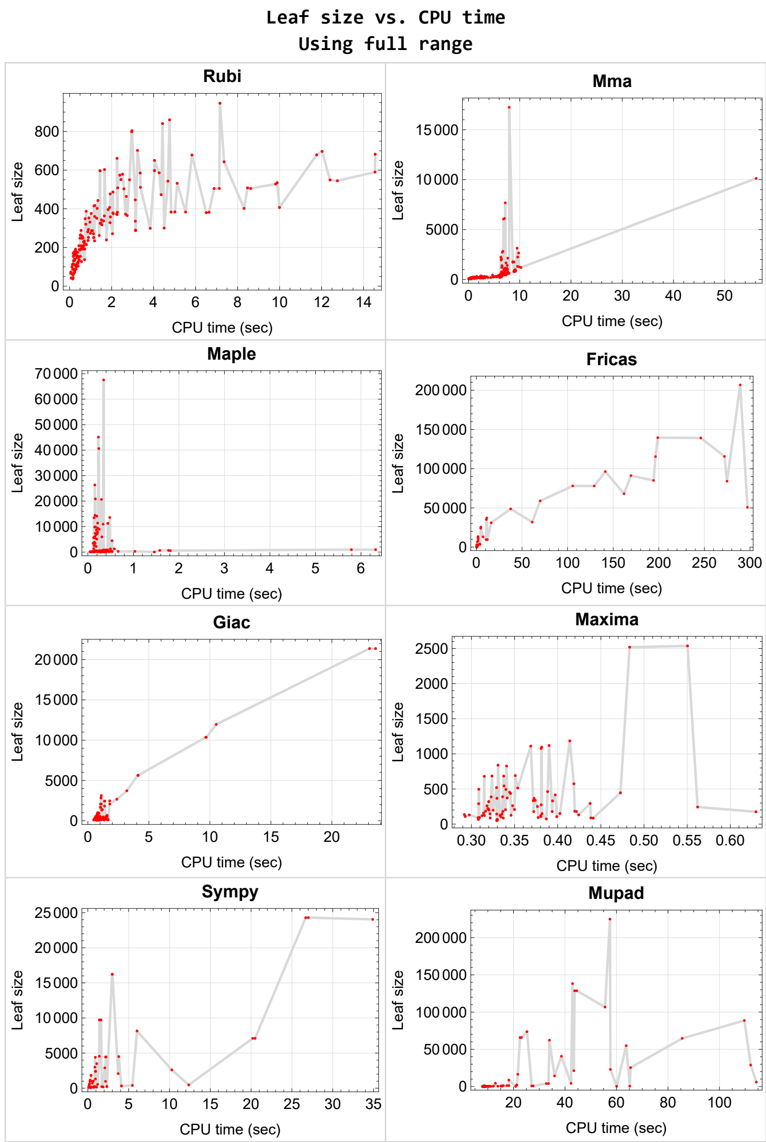


Figure 1.5: Leaf size vs. CPU time. Full range

1.9 list of integrals with no known antiderivative

{}

1.10 List of integrals solved by CAS but has no known antiderivative

Rubi {}

Mathematica {}

Maple {}

Maxima {}

Fricas {}

Sympy {}

Giac {}

Mupad {}

1.11 list of integrals solved by CAS but failed verification

The following are integrals solved by CAS but the verification phase failed to verify the anti-derivative produced is correct. This does not necessarily mean that the anti-derivative is wrong as additional methods of verification might be needed, or more time is needed (3 minutes time limit was used). These integrals are listed here to make it possible to do further investigation to determine why the result could not be verified.

Rubi {}

Mathematica {146}

Maple {}

Maxima Verification phase not currently implemented.

Fricas Verification phase not currently implemented.

Sympy Verification phase not currently implemented.

Giac Verification phase not currently implemented.

Mupad Verification phase not currently implemented.

1.12 Timing

The command `AbsoluteTiming[]` was used in Mathematica to obtain the elapsed time for each integrate call. In Maple, the command `Usage` was used as in the following example

```
cpu_time := Usage(assign ('result_of_int',int(expr,x)),output='realtime'
```

For all other CAS systems, the elapsed time to complete each integral was found by taking the difference between the time after the call completed from the time before the call was made. This was done using Python's `time.time()` call.

All elapsed times shown are in seconds. A time limit of 3 CPU minutes was used for each integral. If the integrate command did not complete within this time limit, the integral was aborted and considered to have failed and assigned an F grade. The time used by failed integrals due to time out was not counted in the final statistics.

1.13 Verification

A verification phase was applied on the result of integration for `Rubi` and `Mathematica`.

Future version of this report will implement verification for the other CAS systems. For the integrals whose result was not run through a verification phase, it is assumed that the antiderivative was correct.

Verification phase also had 3 minutes time out. An integral whose result was not verified could still be correct, but further investigation is needed on those integrals. These integrals were marked in the summary table below and also in each integral separate section so they are easy to identify and locate.

1.14 Important notes about some of the results

Important note about Maxima results

Since tests were run in a batch mode, and using an automated script, then any integral where Maxima needed an interactive response from the user to answer a question during the evaluation of the integral will fail.

The exception raised is `ValueError`. Therefore Maxima results is lower than what would result if Maxima was run directly and each question was answered correctly.

The percentage of such failures were not counted for each test file, but for an example, for the `Timofeev` test file, there were about 14 such integrals out of total 705, or about 2 percent. This percentage can be higher or lower depending on the specific input test file.

Such integrals can be identified by looking at the output of the integration in each section for Maxima. The exception message will indicate the cause of error.

Maxima integrate was run using SageMath with the following settings set by default

```
'besselexpand : true'
'display2d : false'
'domain : complex'
'keepfloat : true'
'load(to_poly_solve)'
'load(simplify_sum)'
'load(abs_integrate)' 'load(diag)'
```

SageMath automatic loading of Maxima `abs_integrate` was found to cause some problems. So the following code was added to disable this effect.

```
from sage.interfaces.maxima_lib import maxima_lib
maxima_lib.set('extra_definite_integration_methods', '[]')
maxima_lib.set('extra_integration_methods', '[]')
```

See `https://ask.sagemath.org/question/43088/integrate-results-that-are-different-from-using-maxima/` for reference.

Important note about FriCAS result

There were few integrals which failed due to SageMath interface and not because FriCAS system could not do the integration.

These will fail With error `Exception raised: NotImplementedError`.

The number of such cases seems to be very small. About 1 or 2 percent of all integrals. These can be identified by looking at the exception message given in the result.

Important note about finding leaf size of antiderivative

For Mathematica, Rubi, and Maple, the builtin system function `LeafSize` was used to find the leaf size of each antiderivative.

The other CAS systems (SageMath and Sympy) do not have special builtin function for this purpose at this time. Therefore the leaf size for Fricas and Sympy antiderivative was determined using the following function, thanks to user `slelievre` at `https://ask.sagemath.org/question/57123/could-we-have-a-leaf_count-function-in-base-sagemath/`

```
def tree_size(expr):
    r"""
    Return the tree size of this expression.
    """
    if expr not in SR:
        # deal with lists, tuples, vectors
        return 1 + sum(tree_size(a) for a in expr)
    expr = SR(expr)
```

```
  x, aa = expr.operator(), expr.operands()
  if x is None:
     return 1
  else:
     return 1 + sum(tree_size(a) for a in aa)
```

For Sympy, which was called directly from Python, the following code was used to obtain the leafsize of its result

```
try:
  # 1.7 is a fudge factor since it is low side from actual leaf count
  leafCount = round(1.7*count_ops(anti))

  except Exception as ee:
       leafCount =1
```

Important note about Mupad results

Matlab's symbolic toolbox does not have a leaf count function to measure the size of the antiderivative. Maple was used to determine the leaf size of Mupad output by post processing Mupad result.

Currently no grading of the antiderivative for Mupad is implemented. If it can integrate the problem, it was assigned a B grade automatically as a placeholder. In the future, when grading function is implemented for Mupad, the tests will be rerun again.

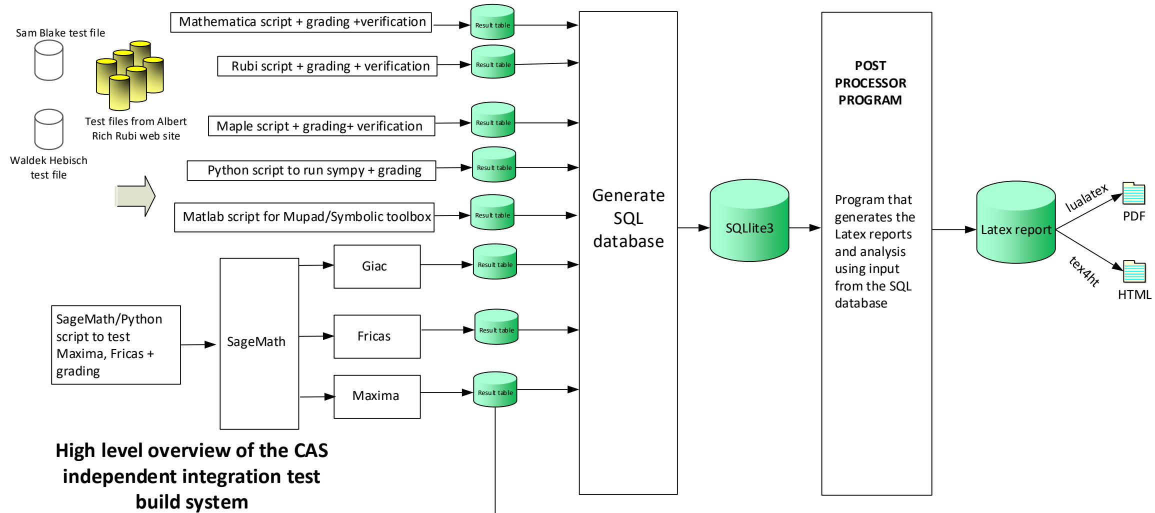
The following is an example of using Matlab's symbolic toolbox (Mupad) to solve an integral

```
integrand = evalin(symengine,'cos(x)*sin(x)')
the_variable = evalin(symengine,'x')
anti = int(integrand,the_variable)
```

Which gives `sin(x)^2/2`

1.15 Design of the test system

The following diagram gives a high level view of the current test build system.



One record (line) per one integral result. The line is CSV comma separated. This is description of each record

1. integer, the problem number.
2. integer. 0 for failed, 1 for passed, -1 for timeout, -2 for CAS specific exception. (this is not the grade field)
3. integer. Leaf size of result.
4. integer. Leaf size of the optimal antiderivative.
5. number. CPU time used to solve this integral. 0 if failed.
6. string. The integral in Latex format
7. string. The input used in CAS own syntax.
8. string. The result (antiderivative) produced by CAS in Latex format
9. string. The optimal antiderivative in Latex format.
10. integer. 0 or 1. Indicates if problem has known antiderivative or not
11. String. The result (antiderivative) in CAS own syntax.
12. String. The grade of the antiderivative. Can be “A”, “B”, “C”, or “F”
13. String. Small string description of why the grade was given.
14. integer. 1 if result was verified or 0 if not verified. (For mma, rubi and maple only)

The following fields are present only in ***Rubi*** *Table file*

15. integer. Number of steps used.
16. integer. Number of rules used.
17. integer. Integrand leaf size.
18. real number. Ratio. Field 16 over field 17
19. String of form “{n,n,..}” which is list of the rules used by Rubi
20. String. The optimal antiderivative in Mathematica syntx

Nasser M. Abbasi
June 27, 2023
Design.vsdx

CHAPTER 2

DETAILED SUMMARY TABLES OF RESULTS

2.1 List of integrals sorted by grade for each CAS 22
2.2 Detailed conclusion table per each integral for all CAS systems 26
2.3 Detailed conclusion table specific for Rubi results 61

2.1 List of integrals sorted by grade for each CAS

Rubi 22
Mma 22
Maple 23
Fricas 23
Maxima 24
Giac 24
Mupad 24
Sympy 25

Rubi

A grade { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171 }

B grade { }

C grade { }

F normal fail { }

F(-1) timedout fail { }

F(-2) exception fail { }

Mma

A grade { 1, 2, 3, 4, 28, 45, 46, 47, 48, 53, 74, 76, 82, 88, 91, 92, 93, 94, 98, 99, 100, 101, 104, 105, 106, 107, 111, 112, 113, 114, 115, 120, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 141, 142, 147, 148, 149, 150, 151, 152, 155, 156, 157, 158, 161, 162, 163, 166, 167, 168, 169, 170 }

B grade { 75, 81, 83, 89, 90, 95, 96, 97, 102, 103, 108, 109, 110, 121, 126, 127, 138, 140, 143, 146, 153, 154, 159, 160, 165, 171 }

C grade { 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 50, 51, 52, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 77, 78, 79, 80, 84, 85, 86, 87, 116, 117, 118, 119, 122, 123, 124, 125, 139, 144, 145 }

F normal fail { 49, 164 }

F(-1) timedout fail { }

F(-2) exception fail { }

Maple

A grade { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89 }

B grade { 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127 }

C grade { }

F normal fail { 45, 46, 47, 48, 49, 164, 165, 166, 167, 168, 169, 170, 171 }

F(-1) timedout fail { 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163 }

F(-2) exception fail { }

Fricas

A grade { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 34, 35, 50, 51, 52, 53, 54, 57, 58, 59, 60, 61, 64, 65, 66, 67, 70, 71, 72, 73, 74, 79, 80 }

B grade { 32, 33, 36, 37, 38, 39, 40, 41, 42, 43, 44, 55, 56, 62, 63, 68, 69, 75, 76, 77, 78, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 97, 98, 99, 100, 104, 105, 106, 110, 111, 112, 113, 118, 119, 124, 125, 130, 131, 132, 137, 138, 148, 149, 150, 151, 155 }

C grade { }

F normal fail { 45, 46, 47, 48, 49, 164, 165, 166, 167, 168, 169, 170, 171 }

F(-1) timedout fail { 94, 95, 96, 101, 102, 103, 107, 108, 109, 114, 115, 116, 117, 120, 121, 122, 123, 126, 127, 128, 129, 133, 134, 135, 136, 139, 140, 141, 142, 143, 144, 145, 146, 147, 152, 153, 154, 156, 157, 158, 159, 160, 161, 162, 163 }

F(-2) exception fail { }

Maxima

A grade { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 77, 78, 79, 80, 81, 84, 85, 86, 87 }

B grade { 76, 82, 83, 88, 89 }

C grade { }

F normal fail { 45, 46, 49, 91, 92, 93, 100, 106, 112, 113, 128, 129, 130, 131, 135, 136, 137, 141, 148, 149, 150, 151, 156, 164, 166, 167, 168, 169, 170 }

F(-1) timedout fail { 47, 48, 90, 97, 98, 99, 104, 105, 110, 111, 116, 117, 118, 119, 122, 123, 124, 125, 132, 134, 138, 139, 142, 143, 144, 145, 147, 152, 153, 154, 155, 157, 158, 159, 160, 162, 163, 165, 171 }

F(-2) exception fail { 94, 95, 96, 101, 102, 103, 107, 108, 109, 114, 115, 120, 121, 126, 127, 133, 140, 146, 161 }

Giac

A grade { 3, 4, 11, 12, 13, 18, 19, 20, 21, 25, 26, 27, 28, 29, 30, 31, 32, 33, 37, 38, 39, 44, 54, 55, 61, 67, 70, 71, 72, 73, 74, 79 }

B grade { 1, 2, 5, 6, 7, 8, 9, 10, 14, 15, 16, 17, 22, 23, 24, 34, 35, 36, 40, 41, 42, 43, 50, 51, 52, 53, 56, 57, 58, 59, 60, 62, 63, 64, 65, 66, 68, 69, 75, 76, 77, 78, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89 }

C grade { }

F normal fail { 45, 46, 47, 126, 130, 164, 165, 166, 167, 168, 169, 170, 171 }

F(-1) timedout fail { 48, 49, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 127, 128, 129, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163 }

F(-2) exception fail { }

Mupad

A grade { }

B grade { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 92, 93, 94, 95, 100, 101, 106, 110, 111, 112, 113, 114, 115, 117, 118, 119, 123, 124, 125 }

C grade { }

F normal fail { }

F(-1) timedout fail { 45, 46, 47, 48, 49, 90, 91, 96, 97, 98, 99, 102, 103, 104, 105, 107, 108, 109, 116, 120, 121, 122, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171 }

F(-2) exception fail { }

Sympy

A grade { 1, 2, 9, 10, 11, 18, 19, 20, 24 }

B grade { 3, 4, 5, 6, 7, 8, 12, 13, 14, 15, 16, 17, 21, 22, 23, 50, 51, 52, 53, 57, 58, 59, 60, 64, 65, 66 }

C grade { 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 54, 55, 61, 62, 67, 68, 70, 71, 72, 73, 74, 77, 78, 79, 80 }

F normal fail { 45, 46, 47, 48, 49, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 165, 166, 167, 168, 169, 171 }

F(-1) timedout fail { 83, 89, 107, 108, 109, 146 }

F(-2) exception fail { 38, 39, 40, 41, 42, 43, 44, 56, 63, 69, 75, 76, 81, 82, 84, 85, 86, 87, 88, 164, 170 }

2.2 Detailed conclusion table per each integral for all CAS systems

Detailed conclusion table per each integral is given by the table below. The elapsed time is in seconds. For failed result it is given as **F(-1)** if the failure was due to timeout. It is given as **F(-2)** if the failure was due to an exception being raised, which could indicate a bug in the system. If the failure was due to integral not being evaluated within the time limit, then it is given as **F**.

In this table, the column `N.S.` means **normalized size** and is defined as $\frac{\text{antiderivative leaf size}}{\text{optimal antiderivative leaf size}}$. To make the table fit the page, the name `Mathematica` was abbreviated to `MMA`.

Problem 1	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	87	87	86	88	86	85	139	937	84
N.S.	1	1.00	0.99	1.01	0.99	0.98	1.60	10.77	0.97
time (sec)	N/A	0.158	0.647	0.279	0.337	0.245	0.114	0.908	8.205

Problem 2	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	A	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	66	66	67	69	66	66	105	556	63
N.S.	1	1.00	1.02	1.05	1.00	1.00	1.59	8.42	0.95
time (sec)	N/A	0.056	0.329	0.046	0.308	0.244	0.103	0.652	7.970

Problem 3	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	B	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	42	42	59	46	50	50	82	50	58
N.S.	1	1.00	1.40	1.10	1.19	1.19	1.95	1.19	1.38
time (sec)	N/A	0.075	0.075	1.460	0.329	0.247	0.294	0.726	7.988

Problem 4	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	B	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	37	37	49	43	52	59	85	53	69
N.S.	1	1.00	1.32	1.16	1.41	1.59	2.30	1.43	1.86
time (sec)	N/A	0.148	0.064	0.286	0.330	0.257	0.412	0.902	8.309

Problem 5	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	B	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	43	43	88	53	68	73	116	119	87
N.S.	1	1.00	2.05	1.23	1.58	1.70	2.70	2.77	2.02
time (sec)	N/A	0.168	0.033	0.286	0.330	0.260	0.719	1.067	8.489

Problem 6	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	B	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	66	66	77	77	86	95	143	179	108
N.S.	1	1.00	1.17	1.17	1.30	1.44	2.17	2.71	1.64
time (sec)	N/A	0.207	0.507	0.339	0.322	0.242	1.072	1.308	7.977

Problem 7	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	B	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	87	87	101	95	104	121	173	237	127
N.S.	1	1.00	1.16	1.09	1.20	1.39	1.99	2.72	1.46
time (sec)	N/A	0.253	1.081	0.355	0.334	0.247	1.875	1.609	7.863

Problem 8	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	B	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	108	108	100	108	122	138	204	299	145
N.S.	1	1.00	0.93	1.00	1.13	1.28	1.89	2.77	1.34
time (sec)	N/A	0.295	1.255	0.348	0.315	0.254	2.321	1.337	8.637

Problem 9	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	A	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	148	148	221	148	147	146	250	2078	151
N.S.	1	1.00	1.49	1.00	0.99	0.99	1.69	14.04	1.02
time (sec)	N/A	0.361	6.256	0.096	0.321	0.273	0.155	1.794	8.465

Problem 10	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	A	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	112	112	172	120	120	119	194	1389	121
N.S.	1	1.00	1.54	1.07	1.07	1.06	1.73	12.40	1.08
time (sec)	N/A	0.136	1.961	0.063	0.312	0.255	0.130	1.192	8.432

Problem 11	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	A	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	87	87	96	87	91	91	151	95	91
N.S.	1	1.00	1.10	1.00	1.05	1.05	1.74	1.09	1.05
time (sec)	N/A	0.156	0.522	0.253	0.308	0.252	0.459	1.006	8.301

Problem 12	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	B	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	70	70	91	80	85	92	136	86	90
N.S.	1	1.00	1.30	1.14	1.21	1.31	1.94	1.23	1.29
time (sec)	N/A	0.215	0.311	0.224	0.441	0.282	0.750	1.336	8.589

Problem 13	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	B	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	72	72	100	84	93	112	158	118	100
N.S.	1	1.00	1.39	1.17	1.29	1.56	2.19	1.64	1.39
time (sec)	N/A	0.242	0.299	0.350	0.377	0.254	1.010	1.674	8.563

Problem 14	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	B	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	88	88	123	107	120	122	206	237	127
N.S.	1	1.00	1.40	1.22	1.36	1.39	2.34	2.69	1.44
time (sec)	N/A	0.311	0.384	0.385	0.380	0.255	1.760	0.868	8.697

Problem 15	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	B	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	118	118	152	136	149	157	252	334	156
N.S.	1	1.00	1.29	1.15	1.26	1.33	2.14	2.83	1.32
time (sec)	N/A	0.363	1.266	0.404	0.382	0.260	2.315	0.908	8.740

Problem 16	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	B	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	151	151	180	162	175	191	304	435	182
N.S.	1	1.00	1.19	1.07	1.16	1.26	2.01	2.88	1.21
time (sec)	N/A	0.431	3.147	0.464	0.630	0.252	4.123	0.978	8.709

Problem 17	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	B	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	165	165	209	180	179	178	313	2670	181
N.S.	1	1.00	1.27	1.09	1.08	1.08	1.90	16.18	1.10
time (sec)	N/A	0.216	1.765	0.109	0.422	0.265	0.166	2.372	8.544

Problem 18	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	A	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	140	140	130	139	143	142	248	158	142
N.S.	1	1.00	0.93	0.99	1.02	1.01	1.77	1.13	1.01
time (sec)	N/A	0.245	1.128	0.283	0.329	0.253	0.767	1.527	8.716

Problem 19	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	A	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	117	117	113	121	124	133	211	129	118
N.S.	1	1.00	0.97	1.03	1.06	1.14	1.80	1.10	1.01
time (sec)	N/A	0.387	0.504	0.265	0.333	0.274	0.976	1.621	9.325

Problem 20	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	A	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	119	119	113	118	125	145	214	152	114
N.S.	1	1.00	0.95	0.99	1.05	1.22	1.80	1.28	0.96
time (sec)	N/A	0.393	0.518	0.256	0.346	0.266	1.701	1.220	8.777

Problem 21	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	B	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	127	127	126	136	142	162	253	193	135
N.S.	1	1.00	0.99	1.07	1.12	1.28	1.99	1.52	1.06
time (sec)	N/A	0.421	0.487	0.243	0.335	0.276	2.326	1.285	8.766

Problem 22	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	B	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	154	154	164	172	180	181	323	390	169
N.S.	1	1.00	1.06	1.12	1.17	1.18	2.10	2.53	1.10
time (sec)	N/A	0.514	1.337	0.311	0.336	0.254	4.109	1.431	8.509

Problem 23	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	B	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	191	191	199	209	215	225	391	528	204
N.S.	1	1.00	1.04	1.09	1.13	1.18	2.05	2.76	1.07
time (sec)	N/A	0.626	0.807	0.314	0.320	0.259	5.461	1.477	8.472

Problem 24	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	A	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	233	233	237	243	250	266	462	670	238
N.S.	1	1.00	1.02	1.04	1.07	1.14	1.98	2.88	1.02
time (sec)	N/A	0.701	1.245	0.357	0.376	0.258	12.364	1.573	8.427

Problem 25	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	C	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	127	127	138	127	130	190	1306	135	144
N.S.	1	1.00	1.09	1.00	1.02	1.50	10.28	1.06	1.13
time (sec)	N/A	0.565	1.577	0.122	0.297	0.271	0.928	0.655	8.201

Problem 26	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	C	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	101	101	118	101	109	149	1020	110	117
N.S.	1	1.00	1.17	1.00	1.08	1.48	10.10	1.09	1.16
time (sec)	N/A	0.311	0.664	0.132	0.293	0.281	0.666	0.487	8.249

Problem 27	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	C	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	85	85	98	87	94	110	711	95	100
N.S.	1	1.00	1.15	1.02	1.11	1.29	8.36	1.12	1.18
time (sec)	N/A	0.212	0.205	0.078	0.309	0.264	0.563	0.499	8.910

Problem 28	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	C	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	58	58	67	66	88	76	541	94	93
N.S.	1	1.00	1.16	1.14	1.52	1.31	9.33	1.62	1.60
time (sec)	N/A	0.161	0.157	0.255	0.438	0.258	1.203	0.668	9.091

Problem 29	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	C	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	80	80	113	95	107	118	966	113	115
N.S.	1	1.00	1.41	1.19	1.34	1.48	12.08	1.41	1.44
time (sec)	N/A	0.245	0.369	0.282	0.381	0.271	2.169	0.802	9.197

Problem 30	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	C	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	103	103	138	122	131	177	2067	157	140
N.S.	1	1.00	1.34	1.18	1.27	1.72	20.07	1.52	1.36
time (sec)	N/A	0.409	0.981	0.316	0.424	0.267	3.717	1.075	9.882

Problem 31	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	C	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	137	137	163	152	158	234	2596	214	175
N.S.	1	1.00	1.19	1.11	1.15	1.71	18.95	1.56	1.28
time (sec)	N/A	0.716	1.517	0.350	0.314	0.280	10.291	1.355	10.245

Problem 32	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	B	C	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	208	208	193	172	220	434	4541	290	210
N.S.	1	1.00	0.93	0.83	1.06	2.09	21.83	1.39	1.01
time (sec)	N/A	0.636	6.086	0.191	0.319	0.310	1.408	0.688	9.236

Problem 33	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	B	C	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	157	157	146	155	197	311	3497	244	165
N.S.	1	1.00	0.93	0.99	1.25	1.98	22.27	1.55	1.05
time (sec)	N/A	0.358	2.900	0.128	0.325	0.289	1.084	0.561	8.649

Problem 34	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	C	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	115	115	140	145	185	221	2995	241	163
N.S.	1	1.00	1.22	1.26	1.61	1.92	26.04	2.10	1.42
time (sec)	N/A	0.167	2.304	0.091	0.316	0.256	0.869	0.535	8.715

Problem 35	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	C	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	111	111	190	141	177	222	2895	234	153
N.S.	1	1.00	1.71	1.27	1.59	2.00	26.08	2.11	1.38
time (sec)	N/A	0.235	2.448	0.292	0.372	0.281	2.073	0.869	8.615

Problem 36	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	B	C	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	137	137	159	163	208	323	4502	279	180
N.S.	1	1.00	1.16	1.19	1.52	2.36	32.86	2.04	1.31
time (sec)	N/A	0.458	2.607	0.405	0.350	0.314	3.788	1.175	10.179

Problem 37	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	B	C	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	192	192	193	196	262	465	8143	362	230
N.S.	1	1.00	1.01	1.02	1.36	2.42	42.41	1.89	1.20
time (sec)	N/A	0.722	3.807	0.536	0.348	0.317	6.028	1.233	11.128

Problem 38	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	B	F(-2)	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	331	331	275	263	389	890	0	505	335
N.S.	1	1.00	0.83	0.79	1.18	2.69	0.00	1.53	1.01
time (sec)	N/A	0.969	5.214	0.216	0.336	0.344	0.000	1.018	9.470

Problem 39	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	B	F(-2)	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	250	250	235	242	366	666	0	458	307
N.S.	1	1.00	0.94	0.97	1.46	2.66	0.00	1.83	1.23
time (sec)	N/A	0.636	6.388	0.152	0.373	0.308	0.000	0.834	8.518

Problem 40	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	B	F(-2)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	189	189	288	223	333	478	0	410	280
N.S.	1	1.00	1.52	1.18	1.76	2.53	0.00	2.17	1.48
time (sec)	N/A	0.468	6.238	0.112	0.372	0.268	0.000	0.682	8.768

Problem 41	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	B	F(-2)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	179	179	188	213	330	488	0	410	282
N.S.	1	1.00	1.05	1.19	1.84	2.73	0.00	2.29	1.58
time (sec)	N/A	0.302	4.350	0.140	0.394	0.285	0.000	0.689	8.648

Problem 42	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	B	F(-2)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	175	175	243	208	321	482	0	409	279
N.S.	1	1.00	1.39	1.19	1.83	2.75	0.00	2.34	1.59
time (sec)	N/A	0.383	4.743	0.496	0.320	0.282	0.000	1.250	8.562

Problem 43	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	B	F(-2)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	215	215	223	243	372	683	0	479	315
N.S.	1	1.00	1.04	1.13	1.73	3.18	0.00	2.23	1.47
time (sec)	N/A	0.801	3.311	0.665	0.341	0.329	0.000	1.241	10.872

Problem 44	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	B	F(-2)	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	287	287	288	289	454	917	0	560	380
N.S.	1	1.00	1.00	1.01	1.58	3.20	0.00	1.95	1.32
time (sec)	N/A	1.089	6.448	1.033	0.344	0.368	0.000	1.578	13.886

Problem 45	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F	F	F	F	F	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	132	132	110	0	0	0	0	0	0
N.S.	1	1.00	0.83	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	0.210	0.628	0.000	0.000	0.000	0.000	0.000	0.000

Problem 46	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F	F	F	F	F	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	154	154	115	0	0	0	0	0	0
N.S.	1	1.00	0.75	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	0.175	0.423	0.000	0.000	0.000	0.000	0.000	0.000

Problem 47	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F	F(-1)	F	F	F	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	170	170	133	0	0	0	0	0	0
N.S.	1	1.00	0.78	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	0.185	0.547	0.000	0.000	0.000	0.000	0.000	0.000

Problem 48	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F	F(-1)	F	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	170	170	133	0	0	0	0	0	0
N.S.	1	1.00	0.78	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	0.174	0.532	0.000	0.000	0.000	0.000	0.000	0.000

Problem 49	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	F	F	F	F	F	F(-1)	F(-1)
verified	N/A	Yes	N/A	N/A	TBD	TBD	TBD	TBD	TBD
size	328	328	0	0	0	0	0	0	0
N.S.	1	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	1.928	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Problem 50	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	B	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	353	353	300	347	416	415	1001	10353	477
N.S.	1	1.00	0.85	0.98	1.18	1.18	2.84	29.33	1.35
time (sec)	N/A	0.921	6.420	0.302	0.397	0.270	0.294	9.657	8.875

Problem 51	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	B	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	248	248	243	246	274	273	617	5631	300
N.S.	1	1.00	0.98	0.99	1.10	1.10	2.49	22.71	1.21
time (sec)	N/A	0.497	3.673	0.151	0.381	0.253	0.209	4.129	8.987

Problem 52	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	B	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	161	161	161	147	151	150	326	2475	153
N.S.	1	1.00	1.00	0.91	0.94	0.93	2.02	15.37	0.95
time (sec)	N/A	0.276	1.691	0.085	0.403	0.260	0.145	1.789	8.420

Problem 53	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	B	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	73	73	76	75	74	74	131	761	75
N.S.	1	1.00	1.04	1.03	1.01	1.01	1.79	10.42	1.03
time (sec)	N/A	0.063	0.521	0.053	0.387	0.239	0.116	0.778	8.703

Problem 54	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	C	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	156	155	148	173	183	226	2387	182	186
N.S.	1	0.99	0.95	1.11	1.17	1.45	15.30	1.17	1.19
time (sec)	N/A	0.388	1.195	0.148	0.420	0.366	0.964	0.550	9.559

Problem 55	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	B	C	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	265	265	216	321	338	556	9721	518	1875
N.S.	1	1.00	0.82	1.21	1.28	2.10	36.68	1.95	7.08
time (sec)	N/A	0.519	2.788	0.140	0.374	0.394	1.410	0.655	21.300

Problem 56	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	B	F(-2)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	320	320	379	494	574	987	0	1006	502
N.S.	1	1.00	1.18	1.54	1.79	3.08	0.00	3.14	1.57
time (sec)	N/A	0.773	6.337	0.195	0.419	0.299	0.000	0.807	15.528

Problem 57	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	B	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	661	661	573	546	691	690	1819	21368	891
N.S.	1	1.00	0.87	0.83	1.05	1.04	2.75	32.33	1.35
time (sec)	N/A	2.261	6.735	0.447	0.351	0.276	0.393	23.581	8.861

Problem 58	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	B	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	443	443	383	392	463	462	1134	11957	561
N.S.	1	1.00	0.86	0.88	1.05	1.04	2.56	26.99	1.27
time (sec)	N/A	1.352	6.552	0.256	0.388	0.263	0.296	10.518	8.359

Problem 59	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	B	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	266	264	241	246	260	259	617	5631	300
N.S.	1	0.99	0.91	0.92	0.98	0.97	2.32	21.17	1.13
time (sec)	N/A	0.512	3.015	0.146	0.318	0.261	0.201	4.094	8.421

Problem 60	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	B	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	131	131	176	141	135	134	241	1825	141
N.S.	1	1.00	1.34	1.08	1.03	1.02	1.84	13.93	1.08
time (sec)	N/A	0.186	1.133	0.077	0.292	0.268	0.132	1.363	8.379

Problem 61	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	C	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	254	252	190	317	290	397	4444	331	325
N.S.	1	0.99	0.75	1.25	1.14	1.56	17.50	1.30	1.28
time (sec)	N/A	0.900	3.211	0.184	0.308	0.493	2.172	0.697	10.714

Problem 62	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	B	C	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	415	415	277	552	496	964	16225	893	3958
N.S.	1	1.00	0.67	1.33	1.20	2.32	39.10	2.15	9.54
time (sec)	N/A	1.166	6.407	0.309	0.309	0.589	2.972	0.837	32.728

Problem 63	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	B	F(-2)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	597	597	1041	865	839	1699	0	1668	807
N.S.	1	1.00	1.74	1.45	1.41	2.85	0.00	2.79	1.35
time (sec)	N/A	1.447	7.045	0.439	0.331	0.742	0.000	1.015	27.621

Problem 64	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	B	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	603	603	419	546	680	679	1819	21368	891
N.S.	1	1.00	0.69	0.91	1.13	1.13	3.02	35.44	1.48
time (sec)	N/A	1.663	6.655	0.435	0.315	0.285	0.391	23.088	8.632

Problem 65	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	B	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	389	387	297	347	387	386	1001	10353	478
N.S.	1	0.99	0.76	0.89	0.99	0.99	2.57	26.61	1.23
time (sec)	N/A	0.792	6.368	0.237	0.323	0.279	0.275	9.706	8.378

Problem 66	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	B	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	191	191	212	210	202	201	410	3720	221
N.S.	1	1.00	1.11	1.10	1.06	1.05	2.15	19.48	1.16
time (sec)	N/A	0.283	2.612	0.106	0.320	0.259	0.170	3.192	8.170

Problem 67	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	C	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	363	363	255	501	436	623	7096	559	508
N.S.	1	1.00	0.70	1.38	1.20	1.72	19.55	1.54	1.40
time (sec)	N/A	1.658	4.894	0.253	0.345	0.794	20.196	0.968	12.193

Problem 68	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	B	C	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	574	574	1024	829	685	1512	24300	1329	701
N.S.	1	1.00	1.78	1.44	1.19	2.63	42.33	2.32	1.22
time (sec)	N/A	2.390	7.593	0.364	0.324	1.156	26.693	1.133	15.062

Problem 69	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	B	F(-2)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	798	798	1409	1271	1119	2549	0	2441	1172
N.S.	1	1.00	1.77	1.59	1.40	3.19	0.00	3.06	1.47
time (sec)	N/A	2.959	7.565	0.496	0.390	1.444	0.000	1.358	17.390

Problem 70	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	C	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	337	337	258	500	445	627	7096	559	508
N.S.	1	1.00	0.77	1.48	1.32	1.86	21.06	1.66	1.51
time (sec)	N/A	1.618	4.863	0.245	0.473	0.773	20.489	0.964	12.131

Problem 71	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	C	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	236	236	190	317	294	390	4444	331	325
N.S.	1	1.00	0.81	1.34	1.25	1.65	18.83	1.40	1.38
time (sec)	N/A	0.880	3.228	0.208	0.438	0.468	2.234	0.672	10.243

Problem 72	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	C	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	156	156	148	173	178	212	2387	182	186
N.S.	1	1.00	0.95	1.11	1.14	1.36	15.30	1.17	1.19
time (sec)	N/A	0.378	1.176	0.147	0.394	0.325	0.924	0.528	9.169

Problem 73	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	C	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	99	99	117	100	106	118	966	106	109
N.S.	1	1.00	1.18	1.01	1.07	1.19	9.76	1.07	1.10
time (sec)	N/A	0.114	0.238	0.099	0.399	0.272	0.582	0.475	8.809

Problem 74	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	A	A	C	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	165	164	313	197	243	301	24052	268	196
N.S.	1	0.99	1.90	1.19	1.47	1.82	145.77	1.62	1.19
time (sec)	N/A	0.305	1.617	0.275	0.562	0.455	34.895	0.602	20.811

Problem 75	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	B	A	A	B	F(-2)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	281	281	572	364	520	1345	0	832	393
N.S.	1	1.00	2.04	1.30	1.85	4.79	0.00	2.96	1.40
time (sec)	N/A	0.866	7.287	0.521	0.329	1.075	0.000	0.769	60.064

Problem 76	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	B	B	F(-2)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	477	477	898	647	1096	3643	0	2080	65819
N.S.	1	1.00	1.88	1.36	2.30	7.64	0.00	4.36	137.99
time (sec)	N/A	1.937	8.915	1.580	0.381	3.390	0.000	1.095	22.517

Problem 77	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	B	C	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	579	579	1022	829	684	1477	24300	1327	701
N.S.	1	1.00	1.77	1.43	1.18	2.55	41.97	2.29	1.21
time (sec)	N/A	2.523	7.599	0.336	0.337	1.103	27.015	1.138	15.618

Problem 78	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	B	C	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	417	417	277	552	493	939	16225	893	3958
N.S.	1	1.00	0.66	1.32	1.18	2.25	38.91	2.14	9.49
time (sec)	N/A	1.203	5.568	0.270	0.340	0.561	3.011	0.781	33.596

Problem 79	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	C	A	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	292	288	221	321	319	505	9721	515	1875
N.S.	1	0.99	0.76	1.10	1.09	1.73	33.29	1.76	6.42
time (sec)	N/A	0.548	2.479	0.245	0.320	0.338	1.619	0.622	21.226

Problem 80	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	A	C	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	140	140	207	173	205	256	4396	291	184
N.S.	1	1.00	1.48	1.24	1.46	1.83	31.40	2.08	1.31
time (sec)	N/A	0.254	2.714	0.085	0.339	0.265	0.918	0.545	10.549

Problem 81	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	B	A	A	B	F(-2)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	293	293	592	365	513	1275	0	832	430
N.S.	1	1.00	2.02	1.25	1.75	4.35	0.00	2.84	1.47
time (sec)	N/A	0.898	7.593	0.398	0.354	1.110	0.000	0.767	65.171

Problem 82	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	B	B	F(-2)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	509	508	984	577	1185	4174	0	2823	73684
N.S.	1	1.00	1.93	1.13	2.33	8.20	0.00	5.55	144.76
time (sec)	N/A	2.292	9.124	1.811	0.414	3.540	0.000	1.074	25.217

Problem 83	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	B	A	B	B	F(-1)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	841	841	1758	951	2519	9594	0	3115	128667
N.S.	1	1.00	2.09	1.13	3.00	11.41	0.00	3.70	152.99
time (sec)	N/A	4.426	8.883	6.321	0.483	10.632	0.000	1.102	44.526

Problem 84	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	B	F(-2)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	804	804	454	1271	1110	2490	0	2441	1172
N.S.	1	1.00	0.56	1.58	1.38	3.10	0.00	3.04	1.46
time (sec)	N/A	2.971	7.366	0.581	0.369	1.326	0.000	1.388	18.168

Problem 85	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	B	F(-2)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	597	597	1044	865	827	1618	0	1663	807
N.S.	1	1.00	1.75	1.45	1.39	2.71	0.00	2.79	1.35
time (sec)	N/A	1.451	6.980	0.400	0.341	0.649	0.000	1.047	27.059

Problem 86	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	B	F(-2)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	352	349	378	493	543	897	0	1006	502
N.S.	1	0.99	1.07	1.40	1.54	2.55	0.00	2.86	1.43
time (sec)	N/A	0.743	6.404	0.171	0.338	0.291	0.000	0.809	15.424

Problem 87	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	A	A	B	F(-2)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	209	209	261	262	367	566	0	531	327
N.S.	1	1.00	1.25	1.25	1.76	2.71	0.00	2.54	1.56
time (sec)	N/A	0.445	5.515	0.142	0.330	0.297	0.000	0.689	10.858

Problem 88	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	A	B	B	F(-2)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	487	487	912	649	1078	3496	0	2078	65817
N.S.	1	1.00	1.87	1.33	2.21	7.18	0.00	4.27	135.15
time (sec)	N/A	2.063	9.018	1.771	0.381	4.009	0.000	1.036	23.069

Problem 89	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	B	A	B	B	F(-1)	B	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	861	860	1732	949	2537	9567	0	3115	128666
N.S.	1	1.00	2.01	1.10	2.95	11.11	0.00	3.62	149.44
time (sec)	N/A	4.762	8.626	5.789	0.550	12.033	0.000	1.100	43.729

Problem 90	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	B	B	F(-1)	B	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	464	464	1232	4473	0	35153	0	0	0
N.S.	1	1.00	2.66	9.64	0.00	75.76	0.00	0.00	0.00
time (sec)	N/A	2.701	6.545	0.531	0.000	10.873	0.000	0.000	0.000

Problem 91	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	B	F	B	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	325	325	314	3353	0	23984	0	0	0
N.S.	1	1.00	0.97	10.32	0.00	73.80	0.00	0.00	0.00
time (sec)	N/A	1.458	5.243	0.158	0.000	4.685	0.000	0.000	0.000

Problem 92	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	B	F	B	F	F(-1)	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	224	224	220	2218	0	12410	0	0	22955
N.S.	1	1.00	0.98	9.90	0.00	55.40	0.00	0.00	102.48
time (sec)	N/A	0.703	2.159	0.182	0.000	1.664	0.000	0.000	57.645

Problem 93	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	B	F	B	F	F(-1)	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	155	155	150	1312	0	2588	0	0	1199
N.S.	1	1.00	0.97	8.46	0.00	16.70	0.00	0.00	7.74
time (sec)	N/A	0.331	0.592	0.129	0.000	0.353	0.000	0.000	16.041

Problem 94	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	B	F(-2)	F(-1)	F	F(-1)	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	234	234	233	3576	0	0	0	0	62245
N.S.	1	1.00	1.00	15.28	0.00	0.00	0.00	0.00	266.00
time (sec)	N/A	1.192	0.761	0.153	0.000	0.000	0.000	0.000	33.934

Problem 95	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	B	B	F(-2)	F(-1)	F	F(-1)	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	317	317	764	5778	0	0	0	0	138318
N.S.	1	1.00	2.41	18.23	0.00	0.00	0.00	0.00	436.33
time (sec)	N/A	1.535	6.461	0.133	0.000	0.000	0.000	0.000	42.933

Problem 96	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	B	B	F(-2)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	543	543	2819	9797	0	0	0	0	0
N.S.	1	1.00	5.19	18.04	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	4.683	6.721	0.150	0.000	0.000	0.000	0.000	0.000

Problem 97	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	B	B	F(-1)	B	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	550	550	1290	10952	0	84950	0	0	0
N.S.	1	1.00	2.35	19.91	0.00	154.45	0.00	0.00	0.00
time (sec)	N/A	2.854	6.574	0.335	0.000	194.267	0.000	0.000	0.000

Problem 98	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	B	F(-1)	B	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	396	396	510	7939	0	58971	0	0	0
N.S.	1	1.00	1.29	20.05	0.00	148.92	0.00	0.00	0.00
time (sec)	N/A	1.835	6.434	0.212	0.000	69.930	0.000	0.000	0.000

Problem 99	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	B	F(-1)	B	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	273	273	260	5107	0	31081	0	0	0
N.S.	1	1.00	0.95	18.71	0.00	113.85	0.00	0.00	0.00
time (sec)	N/A	0.998	4.837	0.194	0.000	16.158	0.000	0.000	0.000

Problem 100	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	B	F	B	F	F(-1)	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	187	187	202	2500	0	6846	0	0	4260
N.S.	1	1.00	1.08	13.37	0.00	36.61	0.00	0.00	22.78
time (sec)	N/A	0.492	1.319	0.138	0.000	0.950	0.000	0.000	42.332

Problem 101	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	B	F(-2)	F(-1)	F	F(-1)	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	271	271	266	6055	0	0	0	0	106783
N.S.	1	1.00	0.98	22.34	0.00	0.00	0.00	0.00	394.03
time (sec)	N/A	2.044	2.661	0.151	0.000	0.000	0.000	0.000	55.548

Problem 102	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	B	B	F(-2)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	372	372	2738	9865	0	0	0	0	0
N.S.	1	1.00	7.36	26.52	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	2.666	6.611	0.167	0.000	0.000	0.000	0.000	0.000

Problem 103	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	B	B	F(-2)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	532	532	7678	14441	0	0	0	0	0
N.S.	1	1.00	14.43	27.14	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	5.126	7.176	0.158	0.000	0.000	0.000	0.000	0.000

Problem 104	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	B	F(-1)	B	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	503	503	564	11280	0	91140	0	0	0
N.S.	1	1.00	1.12	22.43	0.00	181.19	0.00	0.00	0.00
time (sec)	N/A	2.588	6.575	0.436	0.000	169.464	0.000	0.000	0.000

Problem 105	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	B	F(-1)	B	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	353	351	324	7294	0	48734	0	0	0
N.S.	1	0.99	0.92	20.66	0.00	138.06	0.00	0.00	0.00
time (sec)	N/A	1.294	5.511	0.209	0.000	37.570	0.000	0.000	0.000

Problem 106	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	B	F	B	F	F(-1)	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	229	229	262	3562	0	10840	0	0	5863
N.S.	1	1.00	1.14	15.55	0.00	47.34	0.00	0.00	25.60
time (sec)	N/A	0.664	2.204	0.150	0.000	1.984	0.000	0.000	114.330

Problem 107	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	B	F(-2)	F(-1)	F(-1)	F(-1)	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	336	336	322	8698	0	0	0	0	0
N.S.	1	1.00	0.96	25.89	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	3.132	5.756	0.185	0.000	0.000	0.000	0.000	0.000

Problem 108	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	B	B	F(-2)	F(-1)	F(-1)	F(-1)	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	473	473	6112	14119	0	0	0	0	0
N.S.	1	1.00	12.92	29.85	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	4.362	7.015	0.196	0.000	0.000	0.000	0.000	0.000

Problem 109	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	B	B	F(-2)	F(-1)	F(-1)	F(-1)	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	643	643	17248	20663	0	0	0	0	0
N.S.	1	1.00	26.82	32.14	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	7.357	7.951	0.298	0.000	0.000	0.000	0.000	0.000

Problem 110	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	B	B	F(-1)	B	F	F(-1)	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	407	407	1200	5978	0	37247	0	0	28858
N.S.	1	1.00	2.95	14.69	0.00	91.52	0.00	0.00	70.90
time (sec)	N/A	1.908	6.508	0.308	0.000	11.353	0.000	0.000	112.142

Problem 111	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	B	F(-1)	B	F	F(-1)	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	287	287	414	5513	0	25627	0	0	21254
N.S.	1	1.00	1.44	19.21	0.00	89.29	0.00	0.00	74.06
time (sec)	N/A	1.050	6.361	0.155	0.000	4.807	0.000	0.000	43.418

Problem 112	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	B	F	B	F	F(-1)	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	194	194	192	3853	0	13473	0	0	16400
N.S.	1	1.00	0.99	19.86	0.00	69.45	0.00	0.00	84.54
time (sec)	N/A	0.544	1.630	0.142	0.000	1.749	0.000	0.000	21.650

Problem 113	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	B	F	B	F	F(-1)	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	133	133	129	3463	0	3194	0	0	4326
N.S.	1	1.00	0.97	26.04	0.00	24.02	0.00	0.00	32.53
time (sec)	N/A	0.238	0.238	0.115	0.000	0.385	0.000	0.000	12.950

Problem 114	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	B	F(-2)	F(-1)	F	F(-1)	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	210	210	194	13474	0	0	0	0	25341
N.S.	1	1.00	0.92	64.16	0.00	0.00	0.00	0.00	120.67
time (sec)	N/A	0.674	0.454	0.136	0.000	0.000	0.000	0.000	65.482

Problem 115	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	B	F(-2)	F(-1)	F	F(-1)	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	327	327	521	20870	0	0	0	0	225004
N.S.	1	1.00	1.59	63.82	0.00	0.00	0.00	0.00	688.09
time (sec)	N/A	1.492	6.259	0.165	0.000	0.000	0.000	0.000	57.472

Problem 116	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	B	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	511	511	920	11255	0	0	0	0	0
N.S.	1	1.00	1.80	22.03	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	3.375	6.880	0.441	0.000	0.000	0.000	0.000	0.000

Problem 117	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	B	F(-1)	F(-1)	F	F(-1)	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	343	343	476	9399	0	0	0	0	54886
N.S.	1	1.00	1.39	27.40	0.00	0.00	0.00	0.00	160.02
time (sec)	N/A	1.516	6.568	0.216	0.000	0.000	0.000	0.000	63.779

Problem 118	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	B	F(-1)	B	F	F(-1)	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	201	201	290	7396	0	31879	0	0	40542
N.S.	1	1.00	1.44	36.80	0.00	158.60	0.00	0.00	201.70
time (sec)	N/A	0.642	2.782	0.160	0.000	61.209	0.000	0.000	38.597

Problem 119	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	B	F(-1)	B	F	F(-1)	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	157	157	218	5613	0	7982	0	0	8588
N.S.	1	1.00	1.39	35.75	0.00	50.84	0.00	0.00	54.70
time (sec)	N/A	0.332	1.107	0.127	0.000	1.834	0.000	0.000	18.185

Problem 120	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	B	F(-2)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	262	262	296	26343	0	0	0	0	0
N.S.	1	1.00	1.13	100.55	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	1.413	5.350	0.153	0.000	0.000	0.000	0.000	0.000

Problem 121	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	B	B	F(-2)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	447	446	2078	40619	0	0	0	0	0
N.S.	1	1.00	4.65	90.87	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	3.128	6.428	0.240	0.000	0.000	0.000	0.000	0.000

Problem 122	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	B	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	585	585	670	13586	0	0	0	0	0
N.S.	1	1.00	1.15	23.22	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	3.363	6.930	0.481	0.000	0.000	0.000	0.000	0.000

Problem 123	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	B	F(-1)	F(-1)	F	F(-1)	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	358	358	502	11360	0	0	0	0	88684
N.S.	1	1.00	1.40	31.73	0.00	0.00	0.00	0.00	247.72
time (sec)	N/A	1.945	6.634	0.215	0.000	0.000	0.000	0.000	109.692

Problem 124	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	B	F(-1)	B	F	F(-1)	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	273	271	300	8963	0	50755	0	0	64641
N.S.	1	0.99	1.10	32.83	0.00	185.92	0.00	0.00	236.78
time (sec)	N/A	1.140	3.162	0.248	0.000	297.231	0.000	0.000	85.534

Problem 125	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	B	F(-1)	B	F	F(-1)	B
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	209	209	223	6788	0	13143	0	0	14163
N.S.	1	1.00	1.07	32.48	0.00	62.89	0.00	0.00	67.77
time (sec)	N/A	0.540	0.984	0.180	0.000	7.154	0.000	0.000	35.921

Problem 126	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	B	B	F(-2)	F(-1)	F	F	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	365	365	1948	45119	0	0	0	0	0
N.S.	1	1.00	5.34	123.61	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	2.745	6.395	0.231	0.000	0.000	0.000	0.000	0.000

Problem 127	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	B	B	F(-2)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	Yes	TBD	TBD	TBD	TBD	TBD
size	679	678	6052	67570	0	0	0	0	0
N.S.	1	1.00	8.91	99.51	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	5.828	6.841	0.346	0.000	0.000	0.000	0.000	0.000

Problem 128	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F(-1)	F	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	679	679	1202	0	0	0	0	0	0
N.S.	1	1.00	1.77	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	11.766	10.255	0.000	0.000	0.000	0.000	0.000	0.000

Problem 129	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F(-1)	F	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	505	505	835	0	0	0	0	0	0
N.S.	1	1.00	1.65	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	8.625	9.103	0.000	0.000	0.000	0.000	0.000	0.000

Problem 130	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F(-1)	F	B	F	F	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	381	383	619	0	0	68078	0	0	0
N.S.	1	1.01	1.62	0.00	0.00	178.68	0.00	0.00	0.00
time (sec)	N/A	5.525	7.944	0.000	0.000	161.870	0.000	0.000	0.000

Problem 131	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F(-1)	F	B	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	287	287	441	0	0	78051	0	0	0
N.S.	1	1.00	1.54	0.00	0.00	271.95	0.00	0.00	0.00
time (sec)	N/A	3.146	4.705	0.000	0.000	105.636	0.000	0.000	0.000

Problem 132	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F(-1)	F(-1)	B	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	300	300	412	0	0	139535	0	0	0
N.S.	1	1.00	1.37	0.00	0.00	465.12	0.00	0.00	0.00
time (sec)	N/A	4.505	5.862	0.000	0.000	198.892	0.000	0.000	0.000

Problem 133	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F(-1)	F(-2)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	370	370	600	0	0	0	0	0	0
N.S.	1	1.00	1.62	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	2.259	7.097	0.000	0.000	0.000	0.000	0.000	0.000

Problem 134	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F(-1)	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	597	597	1109	0	0	0	0	0	0
N.S.	1	1.00	1.86	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	4.041	7.514	0.000	0.000	0.000	0.000	0.000	0.000

Problem 135	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F(-1)	F	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	682	682	1304	0	0	0	0	0	0
N.S.	1	1.00	1.91	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	14.554	9.563	0.000	0.000	0.000	0.000	0.000	0.000

Problem 136	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F(-1)	F	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	508	508	867	0	0	0	0	0	0
N.S.	1	1.00	1.71	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	8.477	9.231	0.000	0.000	0.000	0.000	0.000	0.000

Problem 137	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F(-1)	F	B	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	384	384	613	0	0	115434	0	0	0
N.S.	1	1.00	1.60	0.00	0.00	300.61	0.00	0.00	0.00
time (sec)	N/A	5.014	7.860	0.000	0.000	196.391	0.000	0.000	0.000

Problem 138	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	B	F(-1)	F(-1)	B	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	382	382	1664	0	0	206814	0	0	0
N.S.	1	1.00	4.36	0.00	0.00	541.40	0.00	0.00	0.00
time (sec)	N/A	6.640	7.328	0.000	0.000	289.269	0.000	0.000	0.000

Problem 139	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	F(-1)	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	402	402	519	0	0	0	0	0	0
N.S.	1	1.00	1.29	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	8.309	6.761	0.000	0.000	0.000	0.000	0.000	0.000

Problem 140	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	B	F(-1)	F(-2)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	586	586	3134	0	0	0	0	0	0
N.S.	1	1.00	5.35	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	4.268	9.499	0.000	0.000	0.000	0.000	0.000	0.000

Problem 141	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F(-1)	F	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	697	697	1261	0	0	0	0	0	0
N.S.	1	1.00	1.81	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	12.026	9.878	0.000	0.000	0.000	0.000	0.000	0.000

Problem 142	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F(-1)	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	505	505	780	0	0	0	0	0	0
N.S.	1	1.00	1.54	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	7.130	9.038	180.000	0.000	0.000	0.000	0.000	0.000

Problem 143	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	B	F(-1)	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	535	535	1774	0	0	0	0	0	0
N.S.	1	1.00	3.32	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	9.886	8.664	0.000	0.000	0.000	0.000	0.000	0.000

Problem 144	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	F(-1)	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	545	545	802	0	0	0	0	0	0
N.S.	1	1.00	1.47	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	12.755	7.632	0.000	0.000	0.000	0.000	0.000	0.000

Problem 145	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	C	F(-1)	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	590	590	641	0	0	0	0	0	0
N.S.	1	1.00	1.09	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	14.547	7.142	0.000	0.000	0.000	0.000	0.000	0.000

Problem 146	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	B	F(-1)	F(-2)	F(-1)	F(-1)	F(-1)	F(-1)
verified	N/A	Yes	No	N/A	TBD	TBD	TBD	TBD	TBD
size	946	946	10121	0	0	0	0	0	0
N.S.	1	1.00	10.70	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	7.159	56.131	0.000	0.000	0.000	0.000	0.000	0.000

Problem 147	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F(-1)	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	505	505	785	0	0	0	0	0	0
N.S.	1	1.00	1.55	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	6.881	8.844	0.000	0.000	0.000	0.000	0.000	0.000

Problem 148	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F(-1)	F	B	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	383	383	607	0	0	115594	0	0	0
N.S.	1	1.00	1.58	0.00	0.00	301.81	0.00	0.00	0.00
time (sec)	N/A	4.833	7.764	0.000	0.000	271.990	0.000	0.000	0.000

Problem 149	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F(-1)	F	B	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	290	290	456	0	0	77916	0	0	0
N.S.	1	1.00	1.57	0.00	0.00	268.68	0.00	0.00	0.00
time (sec)	N/A	3.137	7.042	0.000	0.000	129.202	0.000	0.000	0.000

Problem 150	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F(-1)	F	B	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	239	239	362	0	0	96324	0	0	0
N.S.	1	1.00	1.51	0.00	0.00	403.03	0.00	0.00	0.00
time (sec)	N/A	1.756	2.418	0.000	0.000	141.315	0.000	0.000	0.000

Problem 151	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F(-1)	F	B	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	251	251	264	0	0	83974	0	0	0
N.S.	1	1.00	1.05	0.00	0.00	334.56	0.00	0.00	0.00
time (sec)	N/A	1.145	2.868	0.000	0.000	274.810	0.000	0.000	0.000

Problem 152	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F(-1)	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	375	375	388	0	0	0	0	0	0
N.S.	1	1.00	1.03	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	2.077	6.664	0.000	0.000	0.000	0.000	0.000	0.000

Problem 153	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	B	F(-1)	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	528	528	2245	0	0	0	0	0	0
N.S.	1	1.00	4.25	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	9.810	9.615	0.000	0.000	0.000	0.000	0.000	0.000

Problem 154	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	B	F(-1)	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	380	380	2141	0	0	0	0	0	0
N.S.	1	1.00	5.63	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	6.507	7.679	0.000	0.000	0.000	0.000	0.000	0.000

Problem 155	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F(-1)	F(-1)	B	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	299	299	403	0	0	139175	0	0	0
N.S.	1	1.00	1.35	0.00	0.00	465.47	0.00	0.00	0.00
time (sec)	N/A	3.841	6.112	0.000	0.000	246.045	0.000	0.000	0.000

Problem 156	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F(-1)	F	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	251	251	275	0	0	0	0	0	0
N.S.	1	1.00	1.10	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	1.151	3.529	0.000	0.000	0.000	0.000	0.000	0.000

Problem 157	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F(-1)	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	383	382	484	0	0	0	0	0	0
N.S.	1	1.00	1.26	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	2.278	6.874	0.000	0.000	0.000	0.000	0.000	0.000

Problem 158	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F(-1)	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	598	598	902	0	0	0	0	0	0
N.S.	1	1.00	1.51	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	4.042	7.167	0.000	0.000	0.000	0.000	0.000	0.000

Problem 159	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	B	F(-1)	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	549	549	2650	0	0	0	0	0	0
N.S.	1	1.00	4.83	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	12.403	9.792	0.000	0.000	0.000	0.000	0.000	0.000

Problem 160	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	B	F(-1)	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	407	407	1135	0	0	0	0	0	0
N.S.	1	1.00	2.79	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	9.996	7.165	180.000	0.000	0.000	0.000	0.000	0.000

Problem 161	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F(-1)	F(-2)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	373	373	609	0	0	0	0	0	0
N.S.	1	1.00	1.63	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	2.245	7.111	0.000	0.000	0.000	0.000	0.000	0.000

Problem 162	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F(-1)	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	379	379	403	0	0	0	0	0	0
N.S.	1	1.00	1.06	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	2.061	5.999	0.000	0.000	0.000	0.000	0.000	0.000

Problem 163	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F(-1)	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	651	650	903	0	0	0	0	0	0
N.S.	1	1.00	1.39	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	4.045	7.207	180.000	0.000	0.000	0.000	0.000	0.000

Problem 164	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	F	F	F	F	F(-2)	F	F(-1)
verified	N/A	Yes	N/A	N/A	TBD	TBD	TBD	TBD	TBD
size	376	376	0	0	0	0	0	0	0
N.S.	1	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	1.030	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Problem 165	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	B	F	F(-1)	F	F	F	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	560	551	1390	0	0	0	0	0	0
N.S.	1	0.98	2.48	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	2.449	6.544	0.000	0.000	0.000	0.000	0.000	0.000

Problem 166	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F	F	F	F	F	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	363	360	505	0	0	0	0	0	0
N.S.	1	0.99	1.39	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	1.194	6.374	0.000	0.000	0.000	0.000	0.000	0.000

Problem 167	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F	F	F	F	F	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	247	247	202	0	0	0	0	0	0
N.S.	1	1.00	0.82	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	0.594	3.274	0.000	0.000	0.000	0.000	0.000	0.000

Problem 168	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F	F	F	F	F	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	178	178	135	0	0	0	0	0	0
N.S.	1	1.00	0.76	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	0.220	0.285	0.000	0.000	0.000	0.000	0.000	0.000

Problem 169	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F	F	F	F	F	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	258	258	204	0	0	0	0	0	0
N.S.	1	1.00	0.79	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	0.604	1.272	0.000	0.000	0.000	0.000	0.000	0.000

Problem 170	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	A	F	F	F	F(-2)	F	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	403	402	563	0	0	0	0	0	0
N.S.	1	1.00	1.40	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	1.316	6.245	0.000	0.000	0.000	0.000	0.000	0.000

Problem 171	Optimal	Rubi	MMA	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	N/A	A	B	F	F(-1)	F	F	F	F(-1)
verified	N/A	Yes	Yes	N/A	TBD	TBD	TBD	TBD	TBD
size	702	702	2238	0	0	0	0	0	0
N.S.	1	1.00	3.19	0.00	0.00	0.00	0.00	0.00	0.00
time (sec)	N/A	3.236	6.325	0.000	0.000	0.000	0.000	0.000	0.000

2.3 Detailed conclusion table specific for Rubi results

The following table is specific to Rubi only. It gives additional statistics for each integral. the column **steps** is the number of steps used by Rubi to obtain the antiderivative. The **rules** column is the number of unique rules used. The **integrand size** column is the leaf size of the integrand. Finally the ratio $\frac{\text{number of rules}}{\text{integrand size}}$ is also given. The larger this ratio is, the harder the integral is to solve. In this test file, problem number [38] had the largest ratio of [.200000000000000011]

Table 2.1: Rubi specific breakdown of results for each integral

#	grade	number of steps used	number of unique rules	normalized antiderivative leaf size	integrand leaf size	$\frac{\text{number of rules}}{\text{integrand leaf size}}$
1	A	5	5	1.00	36	0.139
2	A	3	3	1.00	30	0.100
3	A	3	3	1.00	36	0.083
4	A	5	4	1.00	38	0.105
5	A	4	4	1.00	38	0.105
6	A	5	5	1.00	38	0.132
7	A	6	5	1.00	38	0.132
8	A	7	5	1.00	38	0.132
9	A	6	6	1.00	38	0.158
10	A	4	4	1.00	32	0.125
11	A	4	4	1.00	38	0.105
12	A	5	4	1.00	40	0.100
13	A	5	4	1.00	40	0.100
14	A	5	5	1.00	40	0.125
15	A	6	6	1.00	40	0.150
16	A	7	6	1.00	40	0.150
17	A	5	4	1.00	32	0.125
18	A	5	4	1.00	38	0.105
19	A	6	5	1.00	40	0.125
20	A	6	5	1.00	40	0.125
21	A	6	5	1.00	40	0.125
22	A	6	6	1.00	40	0.150
23	A	7	7	1.00	40	0.175
24	A	8	7	1.00	40	0.175

Continued on next page

Table 2.1 – continued from previous page

#	grade	number of steps used	number of unique rules	normalized antiderivative leaf size	integrand leaf size	$\frac{\text{number of rules}}{\text{integrand leaf size}}$
25	A	7	7	1.00	40	0.175
26	A	6	6	1.00	38	0.158
27	A	6	4	1.00	32	0.125
28	A	3	3	1.00	38	0.079
29	A	4	4	1.00	40	0.100
30	A	5	5	1.00	40	0.125
31	A	6	6	1.00	40	0.150
32	A	7	7	1.00	40	0.175
33	A	6	6	1.00	38	0.158
34	A	3	3	1.00	32	0.094
35	A	4	4	1.00	38	0.105
36	A	5	5	1.00	40	0.125
37	A	6	6	1.00	40	0.150
38	A	8	8	1.00	40	0.200
39	A	7	7	1.00	40	0.175
40	A	5	5	1.00	38	0.132
41	A	4	4	1.00	32	0.125
42	A	5	4	1.00	38	0.105
43	A	6	6	1.00	40	0.150
44	A	7	6	1.00	40	0.150
45	A	7	5	1.00	39	0.128
46	A	7	5	1.00	39	0.128
47	A	7	5	1.00	41	0.122
48	A	7	5	1.00	41	0.122
49	A	13	7	1.00	43	0.163
50	A	6	5	1.00	43	0.116
51	A	5	5	1.00	43	0.116
52	A	4	4	1.00	41	0.098
53	A	3	3	1.00	31	0.097
54	A	5	5	0.99	43	0.116
55	A	5	5	1.00	43	0.116
56	A	4	4	1.00	43	0.093
57	A	7	6	1.00	45	0.133
58	A	6	6	1.00	45	0.133
59	A	5	5	0.99	43	0.116

Continued on next page

Table 2.1 – continued from previous page

#	grade	number of steps used	number of unique rules	normalized antiderivative leaf size	integrand leaf size	number of rules / integrand leaf size
60	A	4	4	1.00	33	0.121
61	A	6	6	0.99	45	0.133
62	A	6	6	1.00	45	0.133
63	A	6	6	1.00	45	0.133
64	A	7	6	1.00	45	0.133
65	A	6	5	0.99	43	0.116
66	A	5	4	1.00	33	0.121
67	A	7	6	1.00	45	0.133
68	A	7	7	1.00	45	0.156
69	A	7	6	1.00	45	0.133
70	A	7	6	1.00	45	0.133
71	A	6	6	1.00	45	0.133
72	A	5	5	1.00	43	0.116
73	A	4	4	1.00	33	0.121
74	A	3	2	0.99	45	0.044
75	A	4	3	1.00	45	0.067
76	A	5	3	1.00	45	0.067
77	A	7	7	1.00	45	0.156
78	A	6	6	1.00	45	0.133
79	A	5	5	0.99	43	0.116
80	A	3	3	1.00	33	0.091
81	A	4	3	1.00	45	0.067
82	A	5	3	1.00	45	0.067
83	A	6	3	1.00	45	0.067
84	A	7	6	1.00	45	0.133
85	A	6	6	1.00	45	0.133
86	A	4	4	0.99	43	0.093
87	A	4	4	1.00	33	0.121
88	A	5	3	1.00	45	0.067
89	A	6	3	1.00	45	0.067
90	A	12	8	1.00	47	0.170
91	A	11	8	1.00	47	0.170
92	A	10	7	1.00	45	0.156
93	A	9	6	1.00	35	0.171
94	A	12	7	1.00	47	0.149

Continued on next page

Table 2.1 – continued from previous page

#	grade	number of steps used	number of unique rules	normalized antiderivative leaf size	integrand leaf size	$\frac{\text{number of rules}}{\text{integrand leaf size}}$
95	A	12	7	1.00	47	0.149
96	A	13	8	1.00	47	0.170
97	A	13	8	1.00	47	0.170
98	A	12	8	1.00	47	0.170
99	A	11	7	1.00	45	0.156
100	A	10	6	1.00	35	0.171
101	A	13	7	1.00	47	0.149
102	A	13	8	1.00	47	0.170
103	A	13	7	1.00	47	0.149
104	A	13	8	1.00	47	0.170
105	A	12	7	0.99	45	0.156
106	A	11	6	1.00	35	0.171
107	A	14	7	1.00	47	0.149
108	A	14	8	1.00	47	0.170
109	A	14	8	1.00	47	0.170
110	A	11	7	1.00	47	0.149
111	A	10	7	1.00	47	0.149
112	A	9	6	1.00	45	0.133
113	A	8	5	1.00	35	0.143
114	A	11	6	1.00	47	0.128
115	A	12	7	1.00	47	0.149
116	A	11	8	1.00	47	0.170
117	A	10	7	1.00	47	0.149
118	A	9	6	1.00	45	0.133
119	A	8	5	1.00	35	0.143
120	A	12	7	1.00	47	0.149
121	A	13	7	1.00	47	0.149
122	A	11	7	1.00	47	0.149
123	A	10	7	1.00	47	0.149
124	A	9	6	0.99	45	0.133
125	A	9	6	1.00	35	0.171
126	A	13	7	1.00	47	0.149
127	A	14	7	1.00	47	0.149
128	A	16	8	1.00	49	0.163
129	A	15	8	1.00	49	0.163

Continued on next page

Table 2.1 – continued from previous page

#	grade	number of steps used	number of unique rules	normalized antiderivative leaf size	integrand leaf size	$\frac{\text{number of rules}}{\text{integrand leaf size}}$
130	A	14	8	1.01	49	0.163
131	A	13	8	1.00	49	0.163
132	A	13	8	1.00	49	0.163
133	A	9	6	1.00	49	0.122
134	A	10	6	1.00	49	0.122
135	A	16	8	1.00	49	0.163
136	A	15	8	1.00	49	0.163
137	A	14	8	1.00	49	0.163
138	A	14	9	1.00	49	0.184
139	A	14	8	1.00	49	0.163
140	A	10	6	1.00	49	0.122
141	A	16	8	1.00	49	0.163
142	A	15	8	1.00	49	0.163
143	A	15	9	1.00	49	0.184
144	A	15	9	1.00	49	0.184
145	A	15	8	1.00	49	0.163
146	A	11	6	1.00	49	0.122
147	A	15	8	1.00	49	0.163
148	A	14	8	1.00	49	0.163
149	A	13	8	1.00	49	0.163
150	A	12	7	1.00	49	0.143
151	A	8	5	1.00	49	0.102
152	A	9	5	1.00	49	0.102
153	A	15	9	1.00	49	0.184
154	A	14	9	1.00	49	0.184
155	A	13	8	1.00	49	0.163
156	A	8	5	1.00	49	0.102
157	A	9	5	1.00	49	0.102
158	A	10	5	1.00	49	0.102
159	A	15	9	1.00	49	0.184
160	A	14	8	1.00	49	0.163
161	A	9	6	1.00	49	0.122
162	A	9	5	1.00	49	0.102
163	A	10	5	1.00	49	0.102
164	A	9	6	1.00	45	0.133

Continued on next page

Table 2.1 – continued from previous page

#	grade	number of steps used	number of unique rules	normalized antiderivative leaf size	integrand leaf size	$\frac{\text{number of rules}}{\text{integrand leaf size}}$
165	A	9	6	0.98	45	0.133
166	A	8	6	0.99	45	0.133
167	A	7	5	1.00	43	0.116
168	A	6	4	1.00	33	0.121
169	A	8	5	1.00	45	0.111
170	A	9	6	1.00	45	0.133
171	A	10	6	1.00	45	0.133

CHAPTER 3

LISTING OF INTEGRALS

3.1 $\int \tan(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$ 73
3.2 $\int (a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$ 79
3.3 $\int \cot(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$ 84
3.4 $\int \cot^2(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$ 89
3.5 $\int \cot^3(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$ 94
3.6 $\int \cot^4(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$ 100
3.7 $\int \cot^5(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$ 106
3.8 $\int \cot^6(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$ 112
3.9 $\int \tan(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$ 118
3.10 $\int (a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$ 126
3.11 $\int \cot(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$ 132
3.12 $\int \cot^2(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$ 137
3.13 $\int \cot^3(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$ 143
3.14 $\int \cot^4(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$ 149
3.15 $\int \cot^5(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$ 155
3.16 $\int \cot^6(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$ 162
3.17 $\int (a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$ 169
3.18 $\int \cot(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$ 177
3.19 $\int \cot^2(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$ 183
3.20 $\int \cot^3(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$ 189
3.21 $\int \cot^4(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$ 196
3.22 $\int \cot^5(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$ 202
3.23 $\int \cot^6(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$ 209
3.24 $\int \cot^7(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$ 217
3.25 $\int \frac{\tan^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx$ 226
3.26 $\int \frac{\tan(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx$ 233
3.27 $\int \frac{B\tan(c+dx)+C\tan^2(c+dx)}{a+b\tan(c+dx)}\,dx$ 240

3.28 $\int \frac{\cot(c+dx)(B\tan(c+dx)+C\tan^2(c+dx))}{a+b\tan(c+dx)}\,dx$ . . . 246
3.29 $\int \frac{\cot^2(c+dx)(B\tan(c+dx)+C\tan^2(c+dx))}{a+b\tan(c+dx)}\,dx$ . . . 251
3.30 $\int \frac{\cot^3(c+dx)(B\tan(c+dx)+C\tan^2(c+dx))}{a+b\tan(c+dx)}\,dx$ . . . 257
3.31 $\int \frac{\cot^4(c+dx)(B\tan(c+dx)+C\tan^2(c+dx))}{a+b\tan(c+dx)}\,dx$ . . . 264
3.32 $\int \frac{\tan^2(c+dx)(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^2}\,dx$ . . . 272
3.33 $\int \frac{\tan(c+dx)(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^2}\,dx$ . . . 282
3.34 $\int \frac{B\tan(c+dx)+C\tan^2(c+dx)}{(a+b\tan(c+dx))^2}\,dx$ . . . 290
3.35 $\int \frac{\cot(c+dx)(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^2}\,dx$ . . . 297
3.36 $\int \frac{\cot^2(c+dx)(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^2}\,dx$ . . . 304
3.37 $\int \frac{\cot^3(c+dx)(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^2}\,dx$ . . . 313
3.38 $\int \frac{\tan^3(c+dx)(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^3}\,dx$ . . . 325
3.39 $\int \frac{\tan^2(c+dx)(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^3}\,dx$ . . . 335
3.40 $\int \frac{\tan(c+dx)(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^3}\,dx$ . . . 343
3.41 $\int \frac{B\tan(c+dx)+C\tan^2(c+dx)}{(a+b\tan(c+dx))^3}\,dx$ . . . 350
3.42 $\int \frac{\cot(c+dx)(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^3}\,dx$ . . . 357
3.43 $\int \frac{\cot^2(c+dx)(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^3}\,dx$ . . . 364
3.44 $\int \frac{\cot^3(c+dx)(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^3}\,dx$ . . . 372
3.45 $\int \tan^2(c+dx)(b\tan(c+dx))^n\,(A+B\tan(c+dx)+C\tan^2(c+dx))\,dx$ . . . 381
3.46 $\int \tan^m(c+dx)(b\tan(c+dx))^n\,(A+B\tan(c+dx)+C\tan^2(c+dx))\,dx$ . . . 386
3.47 $\int \tan^m(c+dx)\sqrt{b\tan(c+dx)}(A+B\tan(c+dx)+C\tan^2(c+dx))\,dx$ . . . 392
3.48 $\int \frac{\tan^m(c+dx)(A+B\tan(c+dx)+C\tan^2(c+dx))}{\sqrt{b\tan(c+dx)}}\,dx$ . . . 398
3.49 $\int \frac{\tan^m(c+dx)(A+B\tan(c+dx)+C\tan^2(c+dx))}{\sqrt{a+b\tan(c+dx)}}\,dx$ . . . 404
3.50 $\int (a+b\tan(e+fx))^3(c+d\tan(e+fx))\,(A+B\tan(e+fx)+C\tan^2(e+fx))\,dx$ 411
3.51 $\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))\,(A+B\tan(e+fx)+C\tan^2(e+fx))\,dx$ 426
3.52 $\int (a+b\tan(e+fx))(c+d\tan(e+fx))\,(A+B\tan(e+fx)+C\tan^2(e+fx))\,dx$ 437
3.53 $\int (c+d\tan(e+fx))\,(A+B\tan(e+fx)+C\tan^2(e+fx))\,dx$ . . . 445
3.54 $\int \frac{(c+d\tan(e+fx))(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx$ . . . 450
3.55 $\int \frac{(c+d\tan(e+fx))(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx$ . . . 457
3.56 $\int \frac{(c+d\tan(e+fx))(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx$ . . . 470
3.57 $\int (a+b\tan(e+fx))^3(c+d\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))\,dx$ 478
3.58 $\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))\,dx$ 502
3.59 $\int (a+b\tan(e+fx))(c+d\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))\,dx$ 519
3.60 $\int (c+d\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))\,dx$ . . . 530
3.61 $\int \frac{(c+d\tan(e+fx))^2(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx$ . . . 537
3.62 $\int \frac{(c+d\tan(e+fx))^2(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx$ . . . 547

3.63 $\int \frac{(c+d\tan(e+fx))^2(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx$ . . . 565
3.64 $\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$ 576
3.65 $\int (a+b\tan(e+fx))(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$ 600
3.66 $\int (c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$ . . . 615
3.67 $\int \frac{(c+d\tan(e+fx))^3(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx$ . . . 624
3.68 $\int \frac{(c+d\tan(e+fx))^3(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx$ . . . 636
3.69 $\int \frac{(c+d\tan(e+fx))^3(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx$ . . . 659
3.70 $\int \frac{(a+b\tan(e+fx))^3(A+B\tan(e+fx)+C\tan^2(e+fx))}{c+d\tan(e+fx)}\,dx$ . . . 671
3.71 $\int \frac{(a+b\tan(e+fx))^2(A+B\tan(e+fx)+C\tan^2(e+fx))}{c+d\tan(e+fx)}\,dx$ . . . 683
3.72 $\int \frac{(a+b\tan(e+fx))(A+B\tan(e+fx)+C\tan^2(e+fx))}{c+d\tan(e+fx)}\,dx$ . . . 693
3.73 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{c+d\tan(e+fx)}\,dx$ . . . 700
3.74 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))}\,dx$ . . . 706
3.75 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))}\,dx$ . . . 723
3.76 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^3(c+d\tan(e+fx))}\,dx$ . . . 731
3.77 $\int \frac{(a+b\tan(e+fx))^3(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^2}\,dx$ . . . 771
3.78 $\int \frac{(a+b\tan(e+fx))^2(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^2}\,dx$ . . . 794
3.79 $\int \frac{(a+b\tan(e+fx))(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^2}\,dx$ . . . 812
3.80 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^2}\,dx$ . . . 825
3.81 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^2}\,dx$ . . . 833
3.82 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^2}\,dx$ . . . 841
3.83 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^3(c+d\tan(e+fx))^2}\,dx$ . . . 885
3.84 $\int \frac{(a+b\tan(e+fx))^3(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^3}\,dx$ . . . 957
3.85 $\int \frac{(a+b\tan(e+fx))^2(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^3}\,dx$ . . . 969
3.86 $\int \frac{(a+b\tan(e+fx))(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^3}\,dx$ . . . 980
3.87 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^3}\,dx$ . . . 988
3.88 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^3}\,dx$ . . . 995
3.89 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^3}\,dx$ . . . 1035
3.90 $\int (a+b\tan(e+fx))^3\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$ 1107
3.91 $\int (a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$ 1119
3.92 $\int (a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$ 1129
3.93 $\int \sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$ . . . 1148
3.94 $\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx$ . . . 1158
3.95 $\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx$ . . . 1195
3.96 $\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx$ . . . 1266

3.97 $\int (a+b\tan(e+fx))^3(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$ 1275
3.98 $\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$ 1287
3.99 $\int (a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$ 1297
3.100 $\int (c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$ 1305
3.101 $\int \frac{(c+d\tan(e+fx))^{3/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx$ 1315
3.102 $\int \frac{(c+d\tan(e+fx))^{3/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx$ 1370
3.103 $\int \frac{(c+d\tan(e+fx))^{3/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx$ 1379
3.104 $\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$ 1387
3.105 $\int (a+b\tan(e+fx))(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$ 1397
3.106 $\int (c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$ 1406
3.107 $\int \frac{(c+d\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx$ 1418
3.108 $\int \frac{(c+d\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx$ 1426
3.109 $\int \frac{(c+d\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx$ 1435
3.110 $\int \frac{(a+b\tan(e+fx))^3(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{c+d\tan(e+fx)}}\,dx$ 1444
3.111 $\int \frac{(a+b\tan(e+fx))^2(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{c+d\tan(e+fx)}}\,dx$ 1467
3.112 $\int \frac{(a+b\tan(e+fx))(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{c+d\tan(e+fx)}}\,dx$ 1485
3.113 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx$ 1501
3.114 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}}\,dx$ 1511
3.115 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}}\,dx$ 1528
3.116 $\int \frac{(a+b\tan(e+fx))^3(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx$ 1637
3.117 $\int \frac{(a+b\tan(e+fx))^2(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx$ 1647
3.118 $\int \frac{(a+b\tan(e+fx))(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx$ 1680
3.119 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^{3/2}}\,dx$ 1705
3.120 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}}\,dx$ 1714
3.121 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}}\,dx$ 1721
3.122 $\int \frac{(a+b\tan(e+fx))^3(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx$ 1730
3.123 $\int \frac{(a+b\tan(e+fx))^2(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx$ 1739
3.124 $\int \frac{(a+b\tan(e+fx))(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx$ 1788
3.125 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^{5/2}}\,dx$ 1826
3.126 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^{5/2}}\,dx$ 1839
3.127 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}}\,dx$ 1848
3.128 $\int (a+b\tan(e+fx))^{5/2}\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$ 1856
3.129 $\int (a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$ 1867
3.130 $\int \sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$ 1876

3.131 $\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{a+b\tan(e+fx)}}\,dx$ 1884
3.132 $\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{3/2}}\,dx$ 1891
3.133 $\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{5/2}}\,dx$ 1898
3.134 $\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{7/2}}\,dx$ 1905
3.135 $\int (a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$ 1913
3.136 $\int \sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$ 1924
3.137 $\int \frac{(c+d\tan(e+fx))^{3/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{a+b\tan(e+fx)}}\,dx$ 1933
3.138 $\int \frac{(c+d\tan(e+fx))^{3/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{3/2}}\,dx$ 1941
3.139 $\int \frac{(c+d\tan(e+fx))^{3/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{5/2}}\,dx$ 1950
3.140 $\int \frac{(c+d\tan(e+fx))^{3/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{7/2}}\,dx$ 1959
3.141 $\int \sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$ 1968
3.142 $\int \frac{(c+d\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{a+b\tan(e+fx)}}\,dx$ 1979
3.143 $\int \frac{(c+d\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{3/2}}\,dx$ 1988
3.144 $\int \frac{(c+d\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{5/2}}\,dx$ 1998
3.145 $\int \frac{(c+d\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{7/2}}\,dx$ 2007
3.146 $\int \frac{(c+d\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{9/2}}\,dx$ 2017
3.147 $\int \frac{(a+b\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{c+d\tan(e+fx)}}\,dx$ 2026
3.148 $\int \frac{(a+b\tan(e+fx))^{3/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{c+d\tan(e+fx)}}\,dx$ 2035
3.149 $\int \frac{\sqrt{a+b\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{c+d\tan(e+fx)}}\,dx$ 2043
3.150 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx$ 2050
3.151 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}\,dx$ 2056
3.152 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{5/2}\sqrt{c+d\tan(e+fx)}}\,dx$ 2062
3.153 $\int \frac{(a+b\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx$ 2068
3.154 $\int \frac{(a+b\tan(e+fx))^{3/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx$ 2078
3.155 $\int \frac{\sqrt{a+b\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx$ 2087
3.156 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}\,dx$ 2094
3.157 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}\,dx$ 2100
3.158 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{5/2}(c+d\tan(e+fx))^{3/2}}\,dx$ 2107
3.159 $\int \frac{(a+b\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx$ 2114
3.160 $\int \frac{(a+b\tan(e+fx))^{3/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx$ 2124
3.161 $\int \frac{\sqrt{a+b\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx$ 2132

3.162 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}\,dx$ 2139

3.163 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{5/2}}\,dx$ 2145

3.164 $\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^n\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$ 2152

3.165 $\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$ 2158

3.166 $\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$ 2167

3.167 $\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$ 2175

3.168 $\int (a+b\tan(e+fx))^m\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$ 2181

3.169 $\int \frac{(a+b\tan(e+fx))^m\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{c+d\tan(e+fx)}\,dx$ 2186

3.170 $\int \frac{(a+b\tan(e+fx))^m\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^2}\,dx$ 2192

3.171 $\int \frac{(a+b\tan(e+fx))^m\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^3}\,dx$ 2199

3.1 $\int \tan(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2($

Optimal result 73
Rubi [A] (verified) 73
Mathematica [A] (verified) 75
Maple [A] (verified) 75
Fricas [A] (verification not implemented) 76
Sympy [A] (verification not implemented) 76
Maxima [A] (verification not implemented) 77
Giac [B] (verification not implemented) 77
Mupad [B] (verification not implemented) 78

Optimal result

Integrand size = 36, antiderivative size = 87

$$\begin{aligned}\int \tan(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx\\ =-((aB-bC)x)+\frac{(bB+aC)\log(\cos(c+dx))}{d}\\ +\frac{(aB-bC)\tan(c+dx)}{d}+\frac{(bB+aC)\tan^2(c+dx)}{2d}+\frac{bC\tan^3(c+dx)}{3d}\end{aligned}$$

```
[Out] -(B*a-C*b)*x+(B*b+C*a)*ln(cos(d*x+c))/d+(B*a-C*b)*tan(d*x+c)/d+1/2*(B*b+C*a
)*tan(d*x+c)^2/d+1/3*b*C*tan(d*x+c)^3/d
```

Rubi [A] (verified)

Time = 0.16 (sec) , antiderivative size = 87, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.139, Rules used = {3713, 3673, 3609, 3606, 3556}

$$\begin{aligned}\int \tan(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx\\ =\frac{(aC+bB)\tan^2(c+dx)}{2d}+\frac{(aB-bC)\tan(c+dx)}{d}\\ +\frac{(aC+bB)\log(\cos(c+dx))}{d}-x(aB-bC)+\frac{bC\tan^3(c+dx)}{3d}\end{aligned}$$

```
[In] Int[Tan[c + d*x]*(a + b*Tan[c + d*x])*(B*Tan[c + d*x] + C*Tan[c + d*x]^2),x
]
```

```
[Out] -((a*B - b*C)*x) + ((b*B + a*C)*Log[Cos[c + d*x]])/d + ((a*B - b*C)*Tan[c +
 d*x])/d + ((b*B + a*C)*Tan[c + d*x]^2)/(2*d) + (b*C*Tan[c + d*x]^3)/(3*d)
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3606

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
 f*x], x], x] + Simp[b*d*(Tan[e + f*x]/f), x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3609

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[d*((a + b*Tan[e + f*x])^m/(f*m)), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3673

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[B
*d*((a + b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*
x])^m*Simp[A*c - B*d + (B*c + A*d)*Tan[e + f*x], x], x] /; FreeQ[{a, b, c,
d, e, f, A, B, m}, x] && NeQ[b*c - a*d, 0] &&  !LeQ[m, -1]
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \tan^2(c+dx)(a+b\tan(c+dx))(B+C\tan(c+dx))\,dx \\
&= \frac{bC\tan^3(c+dx)}{3d} + \int \tan^2(c+dx)(aB-bC+(bB+aC)\tan(c+dx))\,dx \\
&= \frac{(bB+aC)\tan^2(c+dx)}{2d} + \frac{bC\tan^3(c+dx)}{3d} \\
&\quad + \int \tan(c+dx)(-bB-aC+(aB-bC)\tan(c+dx))\,dx
\end{aligned}$$

$$= -((aB - bC)x) + \frac{(aB - bC)\tan(c + dx)}{d} + \frac{(bB + aC)\tan^2(c + dx)}{2d} + \frac{bC\tan^3(c + dx)}{3d} + (-bB - aC)\int \tan(c + dx)\,dx$$

$$= -((aB - bC)x) + \frac{(bB + aC)\log(\cos(c + dx))}{d} + \frac{(aB - bC)\tan(c + dx)}{d} + \frac{(bB + aC)\tan^2(c + dx)}{2d} + \frac{bC\tan^3(c + dx)}{3d}$$

Mathematica [A] (verified)

Time = 0.65 (sec) , antiderivative size = 86, normalized size of antiderivative = 0.99

$$\int \tan(c + dx)(a + b\tan(c + dx))\left(B\tan(c + dx) + C\tan^2(c + dx)\right)\,dx$$

$$= \frac{(-6aB + 6bC)\arctan(\tan(c + dx)) + 6(bB + aC)\log(\cos(c + dx)) + 6(aB - bC)\tan(c + dx) + 3(bB +}{6d}$$

```
[In] Integrate[Tan[c + d*x]*(a + b*Tan[c + d*x])*(B*Tan[c + d*x] + C*Tan[c + d*x
]^2),x]
```

```
[Out] ((-6*a*B + 6*b*C)*ArcTan[Tan[c + d*x]] + 6*(b*B + a*C)*Log[Cos[c + d*x]] +
6*(a*B - b*C)*Tan[c + d*x] + 3*(b*B + a*C)*Tan[c + d*x]^2 + 2*b*C*Tan[c + d
*x]^3)/(6*d)
```

Maple [A] (verified)

Time = 0.28 (sec) , antiderivative size = 88, normalized size of antiderivative = 1.01

method	result
norman	$(-Ba + Cb)\,x + \frac{(Ba - Cb)\tan(dx+c)}{d} + \frac{(Bb + Ca)\tan(dx+c)^2}{2d} + \frac{bC\tan(dx+c)^3}{3d} - \frac{(Bb+Ca)\ln\left(1+\tan(dx\right.}{2d}$
parts	$\frac{(Bb+Ca)\left(\frac{\tan(dx+c)^2}{2} - \frac{\ln\left(1+\tan(dx+c)^2\right)}{2}\right)}{d} + \frac{Ba(\tan(dx+c) - \arctan(\tan(dx+c)))}{d} + \frac{Cb\left(\frac{\tan(dx+c)^3}{3} - \tan(dx+c)+\right.}{d}$
derivativedivides	$\frac{\frac{Cb\tan(dx+c)^3}{3} + \frac{Bb\tan(dx+c)^2}{2} + \frac{Ca\tan(dx+c)^2}{2} + Ba\tan(dx+c) - Cb\tan(dx+c) + \frac{(-Bb-Ca)\ln\left(1+\tan(dx+c)^2\right)}{2} + (-Ba+C}{d}$
default	$\frac{\frac{Cb\tan(dx+c)^3}{3} + \frac{Bb\tan(dx+c)^2}{2} + \frac{Ca\tan(dx+c)^2}{2} + Ba\tan(dx+c) - Cb\tan(dx+c) + \frac{(-Bb-Ca)\ln\left(1+\tan(dx+c)^2\right)}{2} + (-Ba+C}{d}$
parallelrisch	$-\frac{-2Cb\tan(dx+c)^3 + 6Badx - 3Bb\tan(dx+c)^2 - 6Cbdx - 3Ca\tan(dx+c)^2 + 3B\ln\left(1+\tan(dx+c)^2\right)b - 6Ba\tan(dx+c) +}{6d}$
risch	$-iBbx - iCax - Bax + Cbx - \frac{2iBbc}{d} - \frac{2iCac}{d} + \frac{2i\left(-3iBb\,e^{4i(dx+c)} - 3iCa\,e^{4i(dx+c)} + 3Ba\,e^{4i(dx+c)}\right.}{}$

```
[In] int(tan(d*x+c)*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,method=_RET
URNVERBOSE)
```

```
[Out] (-B*a+C*b)*x+(B*a-C*b)*tan(d*x+c)/d+1/2*(B*b+C*a)*tan(d*x+c)^2/d+1/3*b*C*ta
n(d*x+c)^3/d-1/2*(B*b+C*a)/d*ln(1+tan(d*x+c)^2)
```

Fricas [A] (verification not implemented)

none

Time = 0.24 (sec) , antiderivative size = 85, normalized size of antiderivative = 0.98

$$\int \tan(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{2\,Cb\tan\left(dx+c\right)^3-6\left(Ba-Cb\right)dx+3\left(Ca+Bb\right)\tan\left(dx+c\right)^2+3\left(Ca+Bb\right)\log\left(\frac{1}{\tan(dx+c)^2+1}\right)+6\left(Ba\right.}{6\,d}$$

```
[In] integrate(tan(d*x+c)*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, algo
rithm="fricas")
```

```
[Out] 1/6*(2*C*b*tan(d*x + c)^3 - 6*(B*a - C*b)*d*x + 3*(C*a + B*b)*tan(d*x + c)^
2 + 3*(C*a + B*b)*log(1/(tan(d*x + c)^2 + 1)) + 6*(B*a - C*b)*tan(d*x + c))
/d
```

Sympy [A] (verification not implemented)

Time = 0.11 (sec) , antiderivative size = 139, normalized size of antiderivative = 1.60

$$\int \tan(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\begin{cases}-Bax+\frac{Ba\tan\left(c+dx\right)}{d}-\frac{Bb\log\left(\tan^2\left(c+dx\right)+1\right)}{2d}+\frac{Bb\tan^2\left(c+dx\right)}{2d}-\frac{Ca\log\left(\tan^2\left(c+dx\right)+1\right)}{2d}+\frac{Ca\tan^2\left(c+dx\right)}{2d}+Cbx+\frac{Cb\,\mathrm{t}}{}\\ x(a+b\tan\left(c\right))\left(B\tan\left(c\right)+C\tan^2\left(c\right)\right)\tan\left(c\right)\end{cases}$$

```
[In] integrate(tan(d*x+c)*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)**2),x)
```

```
[Out] Piecewise((-B*a*x + B*a*tan(c + d*x)/d - B*b*log(tan(c + d*x)**2 + 1)/(2*d)
 + B*b*tan(c + d*x)**2/(2*d) - C*a*log(tan(c + d*x)**2 + 1)/(2*d) + C*a*tan
(c + d*x)**2/(2*d) + C*b*x + C*b*tan(c + d*x)**3/(3*d) - C*b*tan(c + d*x)/d
, Ne(d, 0)), (x*(a + b*tan(c))*(B*tan(c) + C*tan(c)**2)*tan(c), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.34 (sec) , antiderivative size = 86, normalized size of antiderivative = 0.99

$$\int \tan(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx = \frac{2\,Cb\tan\left(dx+c\right)^3+3\,(Ca+Bb)\tan\left(dx+c\right)^2-6\,(Ba-Cb)(dx+c)-3\,(Ca+Bb)\log\left(\tan\left(dx+c\right)^2\right.}{6\,d}$$

```
[In] integrate(tan(d*x+c)*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, algo
rithm="maxima")
```

```
[Out] 1/6*(2*C*b*tan(d*x + c)^3 + 3*(C*a + B*b)*tan(d*x + c)^2 - 6*(B*a - C*b)*(d
*x + c) - 3*(C*a + B*b)*log(tan(d*x + c)^2 + 1) + 6*(B*a - C*b)*tan(d*x + c
))/d
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 937 vs. 2(83) = 166.

Time = 0.91 (sec) , antiderivative size = 937, normalized size of antiderivative = 10.77

$$\int \tan(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx = \text{Too large to display}$$

```
[In] integrate(tan(d*x+c)*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, algo
rithm="giac")
```

```
[Out] -1/6*(6*B*a*d*x*tan(d*x)^3*tan(c)^3 - 6*C*b*d*x*tan(d*x)^3*tan(c)^3 - 3*C*a
*log(4*(tan(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(d*x)^2*tan(c)^2 +
 tan(d*x)^2 + tan(c)^2 + 1))*tan(d*x)^3*tan(c)^3 - 3*B*b*log(4*(tan(d*x)^2*
tan(c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(d*x)^2*tan(c)^2 + tan(d*x)^2 + tan(c
)^2 + 1))*tan(d*x)^3*tan(c)^3 - 18*B*a*d*x*tan(d*x)^2*tan(c)^2 + 18*C*b*d*x
*tan(d*x)^2*tan(c)^2 - 3*C*a*tan(d*x)^3*tan(c)^3 - 3*B*b*tan(d*x)^3*tan(c)^
3 + 9*C*a*log(4*(tan(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(d*x)^2*t
an(c)^2 + tan(d*x)^2 + tan(c)^2 + 1))*tan(d*x)^2*tan(c)^2 + 9*B*b*log(4*(ta
n(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(d*x)^2*tan(c)^2 + tan(d*x)^
2 + tan(c)^2 + 1))*tan(d*x)^2*tan(c)^2 + 6*B*a*tan(d*x)^3*tan(c)^2 - 6*C*b*
tan(d*x)^3*tan(c)^2 + 6*B*a*tan(d*x)^2*tan(c)^3 - 6*C*b*tan(d*x)^2*tan(c)^3
 + 18*B*a*d*x*tan(d*x)*tan(c) - 18*C*b*d*x*tan(d*x)*tan(c) - 3*C*a*tan(d*x)
^3*tan(c) - 3*B*b*tan(d*x)^3*tan(c) + 3*C*a*tan(d*x)^2*tan(c)^2 + 3*B*b*tan
(d*x)^2*tan(c)^2 - 3*C*a*tan(d*x)*tan(c)^3 - 3*B*b*tan(d*x)*tan(c)^3 + 2*C*
b*tan(d*x)^3 - 9*C*a*log(4*(tan(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(c) + 1)/(t
an(d*x)^2*tan(c)^2 + tan(d*x)^2 + tan(c)^2 + 1))*tan(d*x)*tan(c) - 9*B*b*lo
```

```
g(4*(tan(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(d*x)^2*tan(c)^2 + ta
n(d*x)^2 + tan(c)^2 + 1))*tan(d*x)*tan(c) - 12*B*a*tan(d*x)^2*tan(c) + 18*C
*b*tan(d*x)^2*tan(c) - 12*B*a*tan(d*x)*tan(c)^2 + 18*C*b*tan(d*x)*tan(c)^2
+ 2*C*b*tan(c)^3 - 6*B*a*d*x + 6*C*b*d*x + 3*C*a*tan(d*x)^2 + 3*B*b*tan(d*x
)^2 - 3*C*a*tan(d*x)*tan(c) - 3*B*b*tan(d*x)*tan(c) + 3*C*a*tan(c)^2 + 3*B*
b*tan(c)^2 + 3*C*a*log(4*(tan(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan
(d*x)^2*tan(c)^2 + tan(d*x)^2 + tan(c)^2 + 1)) + 3*B*b*log(4*(tan(d*x)^2*ta
n(c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(d*x)^2*tan(c)^2 + tan(d*x)^2 + tan(c)^
2 + 1)) + 6*B*a*tan(d*x) - 6*C*b*tan(d*x) + 6*B*a*tan(c) - 6*C*b*tan(c) + 3
*C*a + 3*B*b)/(d*tan(d*x)^3*tan(c)^3 - 3*d*tan(d*x)^2*tan(c)^2 + 3*d*tan(d*
x)*tan(c) - d)
```

Mupad [B] (verification not implemented)

Time = 8.20 (sec) , antiderivative size = 84, normalized size of antiderivative = 0.97

$$\int \tan(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{\tan(c+d\,x)\,\left(B\,a-C\,b\right)-\ln\left(\tan(c+d\,x)^2+1\right)\,\left(\frac{B\,b}{2}+\frac{C\,a}{2}\right)+\tan(c+d\,x)^2\,\left(\frac{B\,b}{2}+\frac{C\,a}{2}\right)-d\,x\,(B\,a-C\,b)}{d}$$

```
[In] int(tan(c + d*x)*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x)),x
)
```

```
[Out] (tan(c + d*x)*(B*a - C*b) - log(tan(c + d*x)^2 + 1)*((B*b)/2 + (C*a)/2) + t
an(c + d*x)^2*((B*b)/2 + (C*a)/2) - d*x*(B*a - C*b) + (C*b*tan(c + d*x)^3)/
3)/d
```

3.2 $\int (a+b\tan(c+dx)) \left(B\tan(c+dx)+C\tan^2(c+dx)\right) dx$

Optimal result 79
Rubi [A] (verified) 79
Mathematica [A] (verified) 80
Maple [A] (verified) 81
Fricas [A] (verification not implemented) 81
Sympy [A] (verification not implemented) 82
Maxima [A] (verification not implemented) 82
Giac [B] (verification not implemented) 82
Mupad [B] (verification not implemented) 83

Optimal result

Integrand size = 30, antiderivative size = 66

$$\int (a+b\tan(c+dx)) \left(B\tan(c+dx)+C\tan^2(c+dx)\right) dx = -((bB+aC)x)-\frac{(aB-bC)\log(\cos(c+dx))}{d}+\frac{bB\tan(c+dx)}{d}+\frac{C(a+b\tan(c+dx))^2}{2bd}$$

```
[Out] -(B*b+C*a)*x-(B*a-C*b)*ln(cos(d*x+c))/d+b*B*tan(d*x+c)/d+1/2*C*(a+b*tan(d*x
+c))^2/b/d
```

Rubi [A] (verified)

Time = 0.06 (sec) , antiderivative size = 66, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 3, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.100, Rules used = {3711, 3606, 3556}

$$\int (a+b\tan(c+dx)) \left(B\tan(c+dx)+C\tan^2(c+dx)\right) dx = -\frac{(aB-bC)\log(\cos(c+dx))}{d}-x(aC+bB)+\frac{C(a+b\tan(c+dx))^2}{2bd}+\frac{bB\tan(c+dx)}{d}$$

```
[In] Int[(a + b*Tan[c + d*x])*(B*Tan[c + d*x] + C*Tan[c + d*x]^2),x]
```

```
[Out] -((b*B + a*C)*x) - ((a*B - b*C)*Log[Cos[c + d*x]])/d + (b*B*Tan[c + d*x])/d
 + (C*(a + b*Tan[c + d*x])^2)/(2*b*d)
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3606

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
 f*x], x], x] + Simp[b*d*(Tan[e + f*x]/f), x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rubi steps

$$\text{integral} = \frac{C(a+b\tan(c+dx))^2}{2bd} + \int (a+b\tan(c+dx))(-C+B\tan(c+dx))\,dx$$

$$= -((bB+aC)x) + \frac{bB\tan(c+dx)}{d} + \frac{C(a+b\tan(c+dx))^2}{2bd} + (aB-bC)\int \tan(c+dx)\,dx$$

$$= -((bB+aC)x) - \frac{(aB-bC)\log(\cos(c+dx))}{d} + \frac{bB\tan(c+dx)}{d} + \frac{C(a+b\tan(c+dx))^2}{2bd}$$

Mathematica [A] (verified)

Time = 0.33 (sec) , antiderivative size = 67, normalized size of antiderivative = 1.02

$$\int (a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx = \frac{-2(bB+aC)\arctan(\tan(c+dx))+2(-aB+bC)\log(\cos(c+dx))+2(bB+aC)\tan(c+dx)+bC\tan^2(c+dx)}{2d}$$

```
[In] Integrate[(a + b*Tan[c + d*x])*(B*Tan[c + d*x] + C*Tan[c + d*x]^2),x]
[Out] (-2*(b*B + a*C)*ArcTan[Tan[c + d*x]] + 2*(-(a*B) + b*C)*Log[Cos[c + d*x]] +
 2*(b*B + a*C)*Tan[c + d*x] + b*C*Tan[c + d*x]^2)/(2*d)
```

Maple [A] (verified)

Time = 0.05 (sec) , antiderivative size = 69, normalized size of antiderivative = 1.05

method	result
norman	$(-Bb-Ca)\,x+\frac{(Bb+Ca)\tan(dx+c)}{d}+\frac{Cb\tan(dx+c)^{2}}{2d}+\frac{(Ba-Cb)\ln\left(1+\tan(dx+c)^{2}\right)}{2d}$
derivativedivides	$\frac{\frac{C\tan(dx+c)^{2}b}{2}+B\tan(dx+c)b+C\tan(dx+c)a+\frac{(Ba-Cb)\ln\left(1+\tan(dx+c)^{2}\right)}{2}+(-Bb-Ca)\arctan(\tan(dx+c))}{d}$
default	$\frac{\frac{C\tan(dx+c)^{2}b}{2}+B\tan(dx+c)b+C\tan(dx+c)a+\frac{(Ba-Cb)\ln\left(1+\tan(dx+c)^{2}\right)}{2}+(-Bb-Ca)\arctan(\tan(dx+c))}{d}$
parts	$\frac{(Bb+Ca)(\tan(dx+c)-\arctan(\tan(dx+c)))}{d}+\frac{Ba\ln\left(1+\tan(dx+c)^{2}\right)}{2d}+\frac{Cb\left(\frac{\tan(dx+c)^{2}}{2}-\frac{\ln\left(1+\tan(dx+c)^{2}\right)}{2}\right)}{d}$
parallelrisch	$\frac{-2Bbdx-2Cadx+C\tan(dx+c)^{2}b+B\ln\left(1+\tan(dx+c)^{2}\right)a+2B\tan(dx+c)b-C\ln\left(1+\tan(dx+c)^{2}\right)b+2C\tan(dx+c)a}{2d}$
risch	$-Bbx-Cax+iBax-iCbx+\frac{2iBac}{d}-\frac{2iCbc}{d}+\frac{2i\left(-iCb\,\mathrm{e}^{2i(dx+c)}+Bb\,\mathrm{e}^{2i(dx+c)}+Ca\,\mathrm{e}^{2i(dx+c)}+Bb-\right.}{d\left(\mathrm{e}^{2i(dx+c)}+1\right)^{2}}$

```
[In] int((a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,method=_RETURNVERBOSE)
```

```
[Out] (-B*b-C*a)*x+(B*b+C*a)/d*tan(d*x+c)+1/2*C*b/d*tan(d*x+c)^2+1/2*(B*a-C*b)/d*
ln(1+tan(d*x+c)^2)
```

Fricas [A] (verification not implemented)

none

Time = 0.24 (sec) , antiderivative size = 66, normalized size of antiderivative = 1.00

$$\int (a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^{2}(c+dx)\right)\,dx$$
$$=\frac{Cb\tan\left(dx+c\right)^{2}-2\left(Ca+Bb\right)dx-\left(Ba-Cb\right)\log\left(\frac{1}{\tan(dx+c)^{2}+1}\right)+2\left(Ca+Bb\right)\tan\left(dx+c\right)}{2\,d}$$

```
[In] integrate((a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, algorithm="fric
as")
```

```
[Out] 1/2*(C*b*tan(d*x + c)^2 - 2*(C*a + B*b)*d*x - (B*a - C*b)*log(1/(tan(d*x +
c)^2 + 1)) + 2*(C*a + B*b)*tan(d*x + c))/d
```

Sympy [A] (verification not implemented)

Time = 0.10 (sec) , antiderivative size = 105, normalized size of antiderivative = 1.59

$$\int (a+b\tan(c+dx)) \left(B\tan(c+dx)+C\tan^2(c+dx)\right) \, dx$$
$$= \begin{cases} \frac{Ba\log\left(\tan^2(c+dx)+1\right)}{2d} - Bbx + \frac{Bb\tan(c+dx)}{d} - Cax + \frac{Ca\tan(c+dx)}{d} - \frac{Cb\log\left(\tan^2(c+dx)+1\right)}{2d} + \frac{Cb\tan^2(c+dx)}{2d} & \text{for } d \neq \\ x(a+b\tan(c))\left(B\tan(c)+C\tan^2(c)\right) & \text{otherw} \end{cases}$$

```
[In] integrate((a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)**2),x)
```

```
[Out] Piecewise((B*a*log(tan(c + d*x)**2 + 1)/(2*d) - B*b*x + B*b*tan(c + d*x)/d
- C*a*x + C*a*tan(c + d*x)/d - C*b*log(tan(c + d*x)**2 + 1)/(2*d) + C*b*tan
(c + d*x)**2/(2*d), Ne(d, 0)), (x*(a + b*tan(c))*(B*tan(c) + C*tan(c)**2),
True))
```

Maxima [A] (verification not implemented)

none

Time = 0.31 (sec) , antiderivative size = 66, normalized size of antiderivative = 1.00

$$\int (a+b\tan(c+dx)) \left(B\tan(c+dx)+C\tan^2(c+dx)\right) \, dx$$
$$= \frac{Cb\tan(dx+c)^2 - 2(Ca+Bb)(dx+c) + (Ba-Cb)\log\left(\tan(dx+c)^2+1\right) + 2(Ca+Bb)\tan(dx+c)}{2d}$$

```
[In] integrate((a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, algorithm="maxi
ma")
```

```
[Out] 1/2*(C*b*tan(d*x + c)^2 - 2*(C*a + B*b)*(d*x + c) + (B*a - C*b)*log(tan(d*x
 + c)^2 + 1) + 2*(C*a + B*b)*tan(d*x + c))/d
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 556 vs. 2(64) = 128.

Time = 0.65 (sec) , antiderivative size = 556, normalized size of antiderivative = 8.42

$$\int (a+b\tan(c+dx)) \left(B\tan(c+dx)+C\tan^2(c+dx)\right) \, dx =$$
$$-\frac{2\,Cadx\tan(dx)^2\tan(c)^2 + 2\,Bbdx\tan(dx)^2\tan(c)^2 + Ba\log\left(\frac{4\left(\tan(dx)^2\tan(c)^2 - 2\tan(dx)\tan(c) + 1\right)}{\tan(dx)^2\tan(c)^2 + \tan(dx)^2 + \tan(c)^2 + 1}\right)\tan(d}{}$$

```
[In] integrate((a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, algorithm="giac
")
```

```
[Out] -1/2*(2*C*a*d*x*tan(d*x)^2*tan(c)^2 + 2*B*b*d*x*tan(d*x)^2*tan(c)^2 + B*a*l
og(4*(tan(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(d*x)^2*tan(c)^2 + t
an(d*x)^2 + tan(c)^2 + 1))*tan(d*x)^2*tan(c)^2 - C*b*log(4*(tan(d*x)^2*tan(
c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(d*x)^2*tan(c)^2 + tan(d*x)^2 + tan(c)^2
+ 1))*tan(d*x)^2*tan(c)^2 - 4*C*a*d*x*tan(d*x)*tan(c) - 4*B*b*d*x*tan(d*x)*
tan(c) - C*b*tan(d*x)^2*tan(c)^2 - 2*B*a*log(4*(tan(d*x)^2*tan(c)^2 - 2*tan
(d*x)*tan(c) + 1)/(tan(d*x)^2*tan(c)^2 + tan(d*x)^2 + tan(c)^2 + 1))*tan(d*
x)*tan(c) + 2*C*b*log(4*(tan(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(
d*x)^2*tan(c)^2 + tan(d*x)^2 + tan(c)^2 + 1))*tan(d*x)*tan(c) + 2*C*a*tan(d
*x)^2*tan(c) + 2*B*b*tan(d*x)^2*tan(c) + 2*C*a*tan(d*x)*tan(c)^2 + 2*B*b*ta
n(d*x)*tan(c)^2 + 2*C*a*d*x + 2*B*b*d*x - C*b*tan(d*x)^2 - C*b*tan(c)^2 + B
*a*log(4*(tan(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(d*x)^2*tan(c)^2
 + tan(d*x)^2 + tan(c)^2 + 1)) - C*b*log(4*(tan(d*x)^2*tan(c)^2 - 2*tan(d*x
)*tan(c) + 1)/(tan(d*x)^2*tan(c)^2 + tan(d*x)^2 + tan(c)^2 + 1)) - 2*C*a*ta
n(d*x) - 2*B*b*tan(d*x) - 2*C*a*tan(c) - 2*B*b*tan(c) - C*b)/(d*tan(d*x)^2*
tan(c)^2 - 2*d*tan(d*x)*tan(c) + d)
```

Mupad [B] (verification not implemented)

Time = 7.97 (sec) , antiderivative size = 63, normalized size of antiderivative = 0.95

$$\int (a+b\tan(c+dx)) \left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx = \frac{\tan(c+d\,x)\,\left(B\,b+C\,a\right)+\ln\left(\tan(c+d\,x)^2+1\right)\,\left(\frac{B\,a}{2}-\frac{C\,b}{2}\right)-d\,x\,\left(B\,b+C\,a\right)+\frac{C\,b\tan(c+d\,x)^2}{2}}{d}$$

```
[In] int((B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x)),x)
```

```
[Out] (tan(c + d*x)*(B*b + C*a) + log(tan(c + d*x)^2 + 1)*((B*a)/2 - (C*b)/2) - d
*x*(B*b + C*a) + (C*b*tan(c + d*x)^2)/2)/d
```

3.3 $\int \cot(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c\right.$

Optimal result 84
Rubi [A] (verified) 84
Mathematica [A] (verified) 85
Maple [A] (verified) 86
Fricas [A] (verification not implemented) 86
Sympy [B] (verification not implemented) 86
Maxima [A] (verification not implemented) 87
Giac [A] (verification not implemented) 87
Mupad [B] (verification not implemented) 88

Optimal result

Integrand size = 36, antiderivative size = 42

$$\int \cot(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx = (aB-bC)x-\frac{(bB+aC)\log(\cos(c+dx))}{d}+\frac{bC\tan(c+dx)}{d}$$

```
[Out] (B*a-C*b)*x-(B*b+C*a)*ln(cos(d*x+c))/d+b*C*tan(d*x+c)/d
```

Rubi [A] (verified)

Time = 0.08 (sec) , antiderivative size = 42, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 3, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.083, Rules used = {3713, 3606, 3556}

$$\int \cot(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx = -\frac{(aC+bB)\log(\cos(c+dx))}{d}+x(aB-bC)+\frac{bC\tan(c+dx)}{d}$$

```
[In] Int[Cot[c + d*x]*(a + b*Tan[c + d*x])*(B*Tan[c + d*x] + C*Tan[c + d*x]^2),x
]
```

```
[Out] (a*B - b*C)*x - ((b*B + a*C)*Log[Cos[c + d*x]])/d + (b*C*Tan[c + d*x])/d
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3606

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
 f*x], x], x] + Simp[b*d*(Tan[e + f*x]/f), x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\begin{aligned}\text{integral} &= \int (a+b\tan(c+dx))(B+C\tan(c+dx))\,dx\\ &= (aB-bC)x+\frac{bC\tan(c+dx)}{d}+(bB+aC)\int\tan(c+dx)\,dx\\ &= (aB-bC)x-\frac{(bB+aC)\log(\cos(c+dx))}{d}+\frac{bC\tan(c+dx)}{d}\end{aligned}$$

Mathematica [A] (verified)

Time = 0.07 (sec) , antiderivative size = 59, normalized size of antiderivative = 1.40

$$\begin{aligned}&\int\cot(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx\\ &= aBx-\frac{bC\arctan(\tan(c+dx))}{d}-\frac{bB\log(\cos(c+dx))}{d}\\ &\quad-\frac{aC\log(\cos(c+dx))}{d}+\frac{bC\tan(c+dx)}{d}\end{aligned}$$

```
[In] Integrate[Cot[c + d*x]*(a + b*Tan[c + d*x])*(B*Tan[c + d*x] + C*Tan[c + d*x
]^2),x]
```

```
[Out] a*B*x - (b*C*ArcTan[Tan[c + d*x]])/d - (b*B*Log[Cos[c + d*x]])/d - (a*C*Log
[Cos[c + d*x]])/d + (b*C*Tan[c + d*x])/d
```

Maple [A] (verified)

Time = 1.46 (sec) , antiderivative size = 46, normalized size of antiderivative = 1.10

method	result
parallelrisch	$\frac{(Bb+Ca)\ln\left(\sec(dx+c)^2\right)+2Cb\tan(dx+c)+2dx(Ba-Cb)}{2d}$
norman	$(Ba-Cb)\,x+\frac{bC\tan(dx+c)}{d}+\frac{(Bb+Ca)\ln\left(1+\tan(dx+c)^2\right)}{2d}$
derivativedivides	$\frac{-Bb\ln(\cos(dx+c))+Cb(\tan(dx+c)-dx-c)+Ba(dx+c)-Ca\ln(\cos(dx+c))}{d}$
default	$\frac{-Bb\ln(\cos(dx+c))+Cb(\tan(dx+c)-dx-c)+Ba(dx+c)-Ca\ln(\cos(dx+c))}{d}$
risch	$iBbx+iCax+Bax-Cbx+\frac{2iBbc}{d}+\frac{2iCac}{d}+\frac{2iCb}{d(e^{2i(dx+c)}+1)}-\frac{\ln(e^{2i(dx+c)}+1)Bb}{d}-\frac{\ln(e^{2i(dx+c)}}{d}$

```
[In] int(cot(d*x+c)*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,method=_RET
URNVERBOSE)
```

```
[Out] 1/2*((B*b+C*a)*ln(sec(d*x+c)^2)+2*C*b*tan(d*x+c)+2*d*x*(B*a-C*b))/d
```

Fricas [A] (verification not implemented)

none

Time = 0.25 (sec) , antiderivative size = 50, normalized size of antiderivative = 1.19

$$\int\cot(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx=\frac{2\,(Ba-Cb)dx+2\,Cb\tan\left(dx+c\right)-(Ca+Bb)\log\left(\frac{1}{\tan(dx+c)^2+1}\right)}{2\,d}$$

```
[In] integrate(cot(d*x+c)*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, algo
rithm="fricas")
```

```
[Out] 1/2*(2*(B*a - C*b)*d*x + 2*C*b*tan(d*x + c) - (C*a + B*b)*log(1/(tan(d*x +
c)^2 + 1)))/d
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 82 vs. $2(36)=72$.

Time = 0.29 (sec) , antiderivative size = 82, normalized size of antiderivative = 1.95

$$\int\cot(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx=\begin{cases}Bax+\frac{Bb\log\left(\tan^2\left(c+dx\right)+1\right)}{2d}+\frac{Ca\log\left(\tan^2\left(c+dx\right)+1\right)}{2d}-Cbx+\frac{Cb\tan\left(c+dx\right)}{d} & \text{for } d\neq 0\\ x(a+b\tan\left(c\right))\left(B\tan\left(c\right)+C\tan^2\left(c\right)\right)\cot\left(c\right) & \text{otherwise}\end{cases}$$

```
[In] integrate(cot(d*x+c)*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)**2),x)
```

```
[Out] Piecewise((B*a*x + B*b*log(tan(c + d*x)**2 + 1)/(2*d) + C*a*log(tan(c + d*x
)**2 + 1)/(2*d) - C*b*x + C*b*tan(c + d*x)/d, Ne(d, 0)), (x*(a + b*tan(c))*
(B*tan(c) + C*tan(c)**2)*cot(c), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.33 (sec) , antiderivative size = 50, normalized size of antiderivative = 1.19

$$\int \cot(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{2\,Cb\tan\left(dx+c\right)+2\,(Ba-Cb)(dx+c)+(Ca+Bb)\log\left(\tan\left(dx+c\right)^2+1\right)}{2\,d}$$

```
[In] integrate(cot(d*x+c)*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, algo
rithm="maxima")
```

```
[Out] 1/2*(2*C*b*tan(d*x + c) + 2*(B*a - C*b)*(d*x + c) + (C*a + B*b)*log(tan(d*x
 + c)^2 + 1))/d
```

Giac [A] (verification not implemented)

none

Time = 0.73 (sec) , antiderivative size = 50, normalized size of antiderivative = 1.19

$$\int \cot(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{2\,Cb\tan\left(dx+c\right)+2\,(Ba-Cb)(dx+c)+(Ca+Bb)\log\left(\tan\left(dx+c\right)^2+1\right)}{2\,d}$$

```
[In] integrate(cot(d*x+c)*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, algo
rithm="giac")
```

```
[Out] 1/2*(2*C*b*tan(d*x + c) + 2*(B*a - C*b)*(d*x + c) + (C*a + B*b)*log(tan(d*x
 + c)^2 + 1))/d
```

Mupad [B] (verification not implemented)

Time = 7.99 (sec) , antiderivative size = 58, normalized size of antiderivative = 1.38

$$\int \cot(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=B\,a\,x-C\,b\,x+\frac{C\,b\tan(c+d\,x)}{d}+\frac{B\,b\,\ln\left(\tan(c+d\,x)^2+1\right)}{2\,d}+\frac{C\,a\,\ln\left(\tan(c+d\,x)^2+1\right)}{2\,d}$$

```
[In] int(cot(c + d*x)*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x)),x
)
```

```
[Out] B*a*x - C*b*x + (C*b*tan(c + d*x))/d + (B*b*log(tan(c + d*x)^2 + 1))/(2*d)
+ (C*a*log(tan(c + d*x)^2 + 1))/(2*d)
```

3.4 $\int \cot^2(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)$

Optimal result 89
Rubi [A] (verified) 89
Mathematica [A] (verified) 90
Maple [A] (verified) 91
Fricas [A] (verification not implemented) 91
Sympy [B] (verification not implemented) 92
Maxima [A] (verification not implemented) 92
Giac [A] (verification not implemented) 92
Mupad [B] (verification not implemented) 93

Optimal result

Integrand size = 38, antiderivative size = 37

$$\int \cot^2(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx = (bB+aC)x-\frac{bC\log(\cos(c+dx))}{d}+\frac{aB\log(\sin(c+dx))}{d}$$

```
[Out] (B*b+C*a)*x-b*C*ln(cos(d*x+c))/d+a*B*ln(sin(d*x+c))/d
```

Rubi [A] (verified)

Time = 0.15 (sec) , antiderivative size = 37, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 4, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.105, Rules used = {3713, 3670, 3556, 3612}

$$\int \cot^2(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx = x(aC+bB)+\frac{aB\log(\sin(c+dx))}{d}-\frac{bC\log(\cos(c+dx))}{d}$$

```
[In] Int[Cot[c + d*x]^2*(a + b*Tan[c + d*x])*(B*Tan[c + d*x] + C*Tan[c + d*x]^2)
,x]
```

```
[Out] (b*B + a*C)*x - (b*C*Log[Cos[c + d*x]])/d + (a*B*Log[Sin[c + d*x]])/d
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3612

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[(a*c + b*d)*(x/(a^2 + b^2)), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3670

```
Int[(((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_
.)*(x_)]))/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Dist[B*(d/
b), Int[Tan[e + f*x], x], x] + Dist[1/b, Int[Simp[A*b*c + (A*b*d + B*(b*c -
 a*d))*Tan[e + f*x], x]/(a + b*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d,
e, f, A, B}, x] && NeQ[b*c - a*d, 0]
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \cot(c+dx)(a+b\tan(c+dx))(B+C\tan(c+dx))\,dx \\
&= (bC)\int \tan(c+dx)\,dx + \int \cot(c+dx)(aB+(bB+aC)\tan(c+dx))\,dx \\
&= (bB+aC)x - \frac{bC\log(\cos(c+dx))}{d} + (aB)\int \cot(c+dx)\,dx \\
&= (bB+aC)x - \frac{bC\log(\cos(c+dx))}{d} + \frac{aB\log(\sin(c+dx))}{d}
\end{aligned}$$

Mathematica [A] (verified)

Time = 0.06 (sec) , antiderivative size = 49, normalized size of antiderivative = 1.32

$$\int \cot^2(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$= bBx + aCx + \frac{aB\log(\cos(c+dx))}{d} - \frac{bC\log(\cos(c+dx))}{d} + \frac{aB\log(\tan(c+dx))}{d}$$

```
[In] Integrate[Cot[c + d*x]^2*(a + b*Tan[c + d*x])*(B*Tan[c + d*x] + C*Tan[c + d
*x]^2),x]
```

```
[Out] b*B*x + a*C*x + (a*B*Log[Cos[c + d*x]])/d - (b*C*Log[Cos[c + d*x]])/d + (a*
B*Log[Tan[c + d*x]])/d
```

Maple [A] (verified)

Time = 0.29 (sec) , antiderivative size = 43, normalized size of antiderivative = 1.16

method	result	size
derivativedivides	$\frac{Bb(dx+c)-Cb\ln(\cos(dx+c))+Ba\ln(\sin(dx+c))+Ca(dx+c)}{d}$	43
default	$\frac{Bb(dx+c)-Cb\ln(\cos(dx+c))+Ba\ln(\sin(dx+c))+Ca(dx+c)}{d}$	43
parallelrisch	$\frac{(-Ba+Cb)\ln\left(\sec(dx+c)^2\right)+2Ba\ln(\tan(dx+c))+2x(Bb+Ca)d}{2d}$	47
norman	$(Bb+Ca)\,x+\frac{Ba\ln(\tan(dx+c))}{d}-\frac{(Ba-Cb)\ln\left(1+\tan(dx+c)^2\right)}{2d}$	48
risch	$Bbx+Cax-iBax+iCbx+\frac{2iCbc}{d}-\frac{2iBac}{d}-\frac{\ln\left(e^{2i(dx+c)}+1\right)Cb}{d}+\frac{Ba\ln\left(e^{2i(dx+c)}-1\right)}{d}$	77

```
[In] int(cot(d*x+c)^2*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,method=_R
ETURNVERBOSE)
```

```
[Out] 1/d*(B*b*(d*x+c)-C*b*ln(cos(d*x+c))+B*a*ln(sin(d*x+c))+C*a*(d*x+c))
```

Fricas [A] (verification not implemented)

none

Time = 0.26 (sec) , antiderivative size = 59, normalized size of antiderivative = 1.59

$$\int \cot^2(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx = \frac{2\,(Ca+Bb)dx+Ba\log\left(\frac{\tan(dx+c)^2}{\tan(dx+c)^2+1}\right)-Cb\log\left(\frac{1}{\tan(dx+c)^2+1}\right)}{2\,d}$$

```
[In] integrate(cot(d*x+c)^2*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="fricas")
```

```
[Out] 1/2*(2*(C*a + B*b)*d*x + B*a*log(tan(d*x + c)^2/(tan(d*x + c)^2 + 1)) - C*b
*log(1/(tan(d*x + c)^2 + 1)))/d
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 85 vs. $2(34) = 68$.

Time = 0.41 (sec) , antiderivative size = 85, normalized size of antiderivative = 2.30

$$\int \cot^2(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\begin{cases} -\frac{Ba\log(\tan^2(c+dx)+1)}{2d}+\frac{Ba\log(\tan(c+dx))}{d}+Bbx+Cax+\frac{Cb\log(\tan^2(c+dx)+1)}{2d} & \text{for } d\neq 0 \\ x(a+b\tan(c))\left(B\tan(c)+C\tan^2(c)\right)\cot^2(c) & \text{otherwise} \end{cases}$$

```
[In] integrate(cot(d*x+c)**2*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)**2),x)
[Out] Piecewise((-B*a*log(tan(c + d*x)**2 + 1)/(2*d) + B*a*log(tan(c + d*x))/d +
B*b*x + C*a*x + C*b*log(tan(c + d*x)**2 + 1)/(2*d), Ne(d, 0)), (x*(a + b*ta
n(c))*(B*tan(c) + C*tan(c)**2)*cot(c)**2, True))
```

Maxima [A] (verification not implemented)

none

Time = 0.33 (sec) , antiderivative size = 52, normalized size of antiderivative = 1.41

$$\int \cot^2(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{2\,Ba\log(\tan(dx+c))+2\,(Ca+Bb)(dx+c)-(Ba-Cb)\log\left(\tan(dx+c)^2+1\right)}{2\,d}$$

```
[In] integrate(cot(d*x+c)^2*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="maxima")
[Out] 1/2*(2*B*a*log(tan(d*x + c)) + 2*(C*a + B*b)*(d*x + c) - (B*a - C*b)*log(ta
n(d*x + c)^2 + 1))/d
```

Giac [A] (verification not implemented)

none

Time = 0.90 (sec) , antiderivative size = 53, normalized size of antiderivative = 1.43

$$\int \cot^2(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{2\,Ba\log(|\tan(dx+c)|)+2\,(Ca+Bb)(dx+c)-(Ba-Cb)\log\left(\tan(dx+c)^2+1\right)}{2\,d}$$

```
[In] integrate(cot(d*x+c)^2*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="giac")
[Out] 1/2*(2*B*a*log(abs(tan(d*x + c))) + 2*(C*a + B*b)*(d*x + c) - (B*a - C*b)*l
og(tan(d*x + c)^2 + 1))/d
```

Mupad [B] (verification not implemented)

Time = 8.31 (sec) , antiderivative size = 69, normalized size of antiderivative = 1.86

$$\int \cot^2(c+dx)(a+b\tan(c+dx)) \left(B\tan(c+dx)+C\tan^2(c+dx)\right) \, dx = \frac{B\,a\,\ln\left(\tan(c+d\,x)\right)}{d} - \frac{\ln\left(\tan(c+d\,x)-\mathrm{i}\right)\,\left(B+C\,1\mathrm{i}\right)\,\left(a+b\,1\mathrm{i}\right)}{2\,d} + \frac{\ln\left(\tan(c+d\,x)+1\mathrm{i}\right)\,\left(B-C\,1\mathrm{i}\right)\,\left(b+a\,1\mathrm{i}\right)\,1\mathrm{i}}{2\,d}$$

```
[In] int(cot(c + d*x)^2*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))
,x)
```

```
[Out] (log(tan(c + d*x) + 1i)*(B - C*1i)*(a*1i + b)*1i)/(2*d) - (log(tan(c + d*x)
 - 1i)*(B + C*1i)*(a + b*1i))/(2*d) + (B*a*log(tan(c + d*x)))/d
```

3.5 $\int \cot^3(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)$

Optimal result 94
Rubi [A] (verified) 94
Mathematica [C] (verified) 96
Maple [A] (verified) 96
Fricas [A] (verification not implemented) 97
Sympy [B] (verification not implemented) 97
Maxima [A] (verification not implemented) 98
Giac [B] (verification not implemented) 98
Mupad [B] (verification not implemented) 98

Optimal result

Integrand size = 38, antiderivative size = 43

$$\int \cot^3(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx = -((aB-bC)x)-\frac{aB\cot(c+dx)}{d}+\frac{(bB+aC)\log(\sin(c+dx))}{d}$$

```
[Out] -(B*a-C*b)*x-a*B*cot(d*x+c)/d+(B*b+C*a)*ln(sin(d*x+c))/d
```

Rubi [A] (verified)

Time = 0.17 (sec) , antiderivative size = 43, normalized size of antiderivative = 1.00, number of steps used = 4, number of rules used = 4, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.105, Rules used = {3713, 3672, 3612, 3556}

$$\int \cot^3(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx = \frac{(aC+bB)\log(\sin(c+dx))}{d}-(x(aB-bC))-\frac{aB\cot(c+dx)}{d}$$

```
[In] Int[Cot[c + d*x]^3*(a + b*Tan[c + d*x])*(B*Tan[c + d*x] + C*Tan[c + d*x]^2)
,x]
```

```
[Out] -((a*B - b*C)*x) - (a*B*Cot[c + d*x])/d + ((b*B + a*C)*Log[Sin[c + d*x]])/d
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3612

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[(a*c + b*d)*(x/(a^2 + b^2)), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3672

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[(b
*c - a*d)*(A*b - a*B)*((a + b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1)*(a^2 + b^2
))), x] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[a*A*c +
 b*B*c + A*b*d - a*B*d - (A*b*c - a*B*c - a*A*d - b*B*d)*Tan[e + f*x], x],
x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] && NeQ[b*c - a*d, 0] && LtQ[m,
 -1] && NeQ[a^2 + b^2, 0]
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$
\begin{aligned}
\text{integral} &= \int \cot^2(c+dx)(a+b\tan(c+dx))(B+C\tan(c+dx))\,dx \\
&= -\frac{aB\cot(c+dx)}{d} + \int \cot(c+dx)(bB+aC-(aB-bC)\tan(c+dx))\,dx \\
&= -((aB-bC)x) - \frac{aB\cot(c+dx)}{d} + (bB+aC)\int \cot(c+dx)\,dx \\
&= -((aB-bC)x) - \frac{aB\cot(c+dx)}{d} + \frac{(bB+aC)\log(\sin(c+dx))}{d}
\end{aligned}
$$

Mathematica [C] (verified)

Result contains higher order function than in optimal. Order 5 vs. order 3 in optimal.

Time = 0.03 (sec) , antiderivative size = 88, normalized size of antiderivative = 2.05

$$
\begin{aligned}
&\int \cot^3(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx \\
&= bCx - \frac{aB\cot(c+dx)\,\text{Hypergeometric2F1}\left(-\frac{1}{2},1,\frac{1}{2},-\tan^2(c+dx)\right)}{d} \\
&\quad + \frac{bB\log(\cos(c+dx))}{d} + \frac{aC\log(\cos(c+dx))}{d} \\
&\quad + \frac{bB\log(\tan(c+dx))}{d} + \frac{aC\log(\tan(c+dx))}{d}
\end{aligned}
$$

```
[In] Integrate[Cot[c + d*x]^3*(a + b*Tan[c + d*x])*(B*Tan[c + d*x] + C*Tan[c + d
*x]^2),x]
```

```
[Out] b*C*x - (a*B*Cot[c + d*x]*Hypergeometric2F1[-1/2, 1, 1/2, -Tan[c + d*x]^2])
/d + (b*B*Log[Cos[c + d*x]])/d + (a*C*Log[Cos[c + d*x]])/d + (b*B*Log[Tan[c
 + d*x]])/d + (a*C*Log[Tan[c + d*x]])/d
```

Maple [A] (verified)

Time = 0.29 (sec) , antiderivative size = 53, normalized size of antiderivative = 1.23

method	result
derivativedivides	$\frac{Bb\ln(\sin(dx+c))+Cb(dx+c)+Ba(-\cot(dx+c)-dx-c)+Ca\ln(\sin(dx+c))}{d}$
default	$\frac{Bb\ln(\sin(dx+c))+Cb(dx+c)+Ba(-\cot(dx+c)-dx-c)+Ca\ln(\sin(dx+c))}{d}$
parallelrisch	$\frac{(-Bb-Ca)\ln\left(\sec(dx+c)^2\right)+(2Bb+2Ca)\ln(\tan(dx+c))-2B\cot(dx+c)a-2dx(Ba-Cb)}{2d}$
norman	$\frac{(-Ba+Cb)x\tan(dx+c)^2-\frac{Ba\tan(dx+c)}{d}}{\tan(dx+c)^2}+\frac{(Bb+Ca)\ln(\tan(dx+c))}{d}-\frac{(Bb+Ca)\ln\left(1+\tan(dx+c)^2\right)}{2d}$
risch	$-iBbx-iCax-Bax+Cbx-\frac{2iBbc}{d}-\frac{2iCac}{d}-\frac{2iBa}{d\left(e^{2i(dx+c)}-1\right)}+\frac{\ln\left(e^{2i(dx+c)}-1\right)Bb}{d}+\frac{\ln(e^{2i(dx+}}{d}$

```
[In] int(cot(d*x+c)^3*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,method=_R
ETURNVERBOSE)
```

```
[Out] 1/d*(B*b*ln(sin(d*x+c))+C*b*(d*x+c)+B*a*(-cot(d*x+c)-d*x-c)+C*a*ln(sin(d*x+
c)))
```

Fricas [A] (verification not implemented)

none

Time = 0.26 (sec) , antiderivative size = 73, normalized size of antiderivative = 1.70

$$\int \cot^3(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=-\frac{2\,(Ba-Cb)dx\tan\left(dx+c\right)-(Ca+Bb)\log\left(\frac{\tan(dx+c)^2}{\tan(dx+c)^2+1}\right)\tan\left(dx+c\right)+2\,Ba}{2\,d\tan\left(dx+c\right)}$$

```
[In] integrate(cot(d*x+c)^3*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="fricas")
```

```
[Out] -1/2*(2*(B*a - C*b)*d*x*tan(d*x + c) - (C*a + B*b)*log(tan(d*x + c)^2/(tan(
d*x + c)^2 + 1))*tan(d*x + c) + 2*B*a)/(d*tan(d*x + c))
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 116 vs. $2(36) = 72$.

Time = 0.72 (sec) , antiderivative size = 116, normalized size of antiderivative = 2.70

$$\int \cot^3(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\begin{cases}\text{NaN}\\ x(a+b\tan\left(c\right))\left(B\tan\left(c\right)+C\tan^2\left(c\right)\right)\cot^3\left(c\right)\\ \text{NaN}\\ -Bax-\frac{Ba}{d\tan\left(c+dx\right)}-\frac{Bb\log\left(\tan^2\left(c+dx\right)+1\right)}{2d}+\frac{Bb\log\left(\tan\left(c+dx\right)\right)}{d}-\frac{Ca\log\left(\tan^2\left(c+dx\right)+1\right)}{2d}+\frac{Ca\log\left(\tan\left(c+dx\right)\right)}{d}+Cbx\end{cases}$$

```
[In] integrate(cot(d*x+c)**3*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)**2),x)
```

```
[Out] Piecewise((nan, Eq(c, 0) & Eq(d, 0)), (x*(a + b*tan(c))*(B*tan(c) + C*tan(c
)**2)*cot(c)**3, Eq(d, 0)), (nan, Eq(c, -d*x)), (-B*a*x - B*a/(d*tan(c + d*
x)) - B*b*log(tan(c + d*x)**2 + 1)/(2*d) + B*b*log(tan(c + d*x))/d - C*a*lo
g(tan(c + d*x)**2 + 1)/(2*d) + C*a*log(tan(c + d*x))/d + C*b*x, True))
```

Maxima [A] (verification not implemented)

none

Time = 0.33 (sec) , antiderivative size = 68, normalized size of antiderivative = 1.58

$$\int \cot^3(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx = -\frac{2\,(Ba-Cb)(dx+c)+(Ca+Bb)\log\left(\tan\left(dx+c\right)^2+1\right)-2\,(Ca+Bb)\log\left(\tan\left(dx+c\right)\right)+\frac{2\,Ba}{\tan(dx+c)}}{2\,d}$$

```
[In] integrate(cot(d*x+c)^3*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="maxima")
```

```
[Out] -1/2*(2*(B*a - C*b)*(d*x + c) + (C*a + B*b)*log(tan(d*x + c)^2 + 1) - 2*(C*
a + B*b)*log(tan(d*x + c)) + 2*B*a/tan(d*x + c))/d
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 119 vs. 2(43) = 86.

Time = 1.07 (sec) , antiderivative size = 119, normalized size of antiderivative = 2.77

$$\int \cot^3(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx = \frac{Ba\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)-2\,(Ba-Cb)(dx+c)-2\,(Ca+Bb)\log\left(\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^2+1\right)+2\,(Ca+Bb)\log\left(\right.}{2\,d}$$

```
[In] integrate(cot(d*x+c)^3*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="giac")
```

```
[Out] 1/2*(B*a*tan(1/2*d*x + 1/2*c) - 2*(B*a - C*b)*(d*x + c) - 2*(C*a + B*b)*log
(tan(1/2*d*x + 1/2*c)^2 + 1) + 2*(C*a + B*b)*log(abs(tan(1/2*d*x + 1/2*c)))
 - (2*C*a*tan(1/2*d*x + 1/2*c) + 2*B*b*tan(1/2*d*x + 1/2*c) + B*a)/tan(1/2*
d*x + 1/2*c))/d
```

Mupad [B] (verification not implemented)

Time = 8.49 (sec) , antiderivative size = 87, normalized size of antiderivative = 2.02

$$\int \cot^3(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx = \frac{\ln\left(\tan(c+d\,x)\right)\,\left(B\,b+C\,a\right)}{d}-\frac{\ln\left(\tan(c+d\,x)+1\mathrm{i}\right)\,\left(B-C\,1\mathrm{i}\right)\,\left(b+a\,1\mathrm{i}\right)}{2\,d}-\frac{B\,a\cot(c+d\,x)}{d}+\frac{\ln\left(\tan(c+d\,x)-\mathrm{i}\right)\,\left(B+C\,1\mathrm{i}\right)\,\left(a+b\,1\mathrm{i}\right)\,1\mathrm{i}}{2\,d}$$

```
[In] int(cot(c + d*x)^3*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))
,x)
```

```
[Out] (log(tan(c + d*x))*(B*b + C*a))/d + (log(tan(c + d*x) - 1i)*(B + C*1i)*(a +
 b*1i)*1i)/(2*d) - (log(tan(c + d*x) + 1i)*(B - C*1i)*(a*1i + b))/(2*d) - (
B*a*cot(c + d*x))/d
```

3.6 $\int \cot^4(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)dx$

Optimal result 100
Rubi [A] (verified) 100
Mathematica [C] (verified) 102
Maple [A] (verified) 102
Fricas [A] (verification not implemented) 103
Sympy [B] (verification not implemented) 103
Maxima [A] (verification not implemented) 104
Giac [B] (verification not implemented) 104
Mupad [B] (verification not implemented) 105

Optimal result

Integrand size = 38, antiderivative size = 66

$$\int \cot^4(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=-((bB+aC)x)-\frac{(bB+aC)\cot(c+dx)}{d}-\frac{aB\cot^2(c+dx)}{2d}-\frac{(aB-bC)\log(\sin(c+dx))}{d}$$

```
[Out] -(B*b+C*a)*x-(B*b+C*a)*cot(d*x+c)/d-1/2*a*B*cot(d*x+c)^2/d-(B*a-C*b)*ln(sin
(d*x+c))/d
```

Rubi [A] (verified)

Time = 0.21 (sec) , antiderivative size = 66, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.132, Rules used = {3713, 3672, 3610, 3612, 3556}

$$\int \cot^4(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=-\frac{(aC+bB)\cot(c+dx)}{d}-\frac{(aB-bC)\log(\sin(c+dx))}{d}-x(aC+bB)-\frac{aB\cot^2(c+dx)}{2d}$$

```
[In] Int[Cot[c + d*x]^4*(a + b*Tan[c + d*x])*(B*Tan[c + d*x] + C*Tan[c + d*x]^2)
,x]
```

```
[Out] -((b*B + a*C)*x) - ((b*B + a*C)*Cot[c + d*x])/d - (a*B*Cot[c + d*x]^2)/(2*d
) - ((a*B - b*C)*Log[Sin[c + d*x]])/d
```

<u>Rule 3556</u>

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3610

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(b*c - a*d)*((a + b*Tan[e + f*x])^(m + 1)/
(f*(m + 1)*(a^2 + b^2))), x] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])
^(m + 1)*Simp[a*c + b*d - (b*c - a*d)*Tan[e + f*x], x], x], x] /; FreeQ[{a,
 b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && LtQ[m, -1]
```

Rule 3612

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[(a*c + b*d)*(x/(a^2 + b^2)), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3672

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[(b
*c - a*d)*(A*b - a*B)*((a + b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1)*(a^2 + b^2
))), x] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[a*A*c +
 b*B*c + A*b*d - a*B*d - (A*b*c - a*B*c - a*A*d - b*B*d)*Tan[e + f*x], x],
x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] && NeQ[b*c - a*d, 0] && LtQ[m,
 -1] && NeQ[a^2 + b^2, 0]
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \cot^3(c+dx)(a+b\tan(c+dx))(B+C\tan(c+dx))\,dx \\
&= -\frac{aB\cot^2(c+dx)}{2d} + \int \cot^2(c+dx)(bB+aC-(aB-bC)\tan(c+dx))\,dx \\
&= -\frac{(bB+aC)\cot(c+dx)}{d} - \frac{aB\cot^2(c+dx)}{2d} \\
&\quad + \int \cot(c+dx)(-aB+bC-(bB+aC)\tan(c+dx))\,dx
\end{aligned}$$

$$= -((bB+aC)x) - \frac{(bB+aC)\cot(c+dx)}{d} - \frac{aB\cot^2(c+dx)}{2d} + (-aB+bC)\int \cot(c+dx)\,dx$$

$$= -((bB+aC)x) - \frac{(bB+aC)\cot(c+dx)}{d} - \frac{aB\cot^2(c+dx)}{2d} - \frac{(aB-bC)\log(\sin(c+dx))}{d}$$

Mathematica [C] (verified)

Result contains higher order function than in optimal. Order 5 vs. order 3 in optimal.

Time = 0.51 (sec) , antiderivative size = 77, normalized size of antiderivative = 1.17

$$\int \cot^4(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx =$$

$$-\frac{aB\cot^2(c+dx)+2(bB+aC)\cot(c+dx)\,\text{Hypergeometric2F1}\left(-\frac{1}{2},1,\frac{1}{2},-\tan^2(c+dx)\right)+2(aB-bC}{2d}$$

```
[In] Integrate[Cot[c + d*x]^4*(a + b*Tan[c + d*x])*(B*Tan[c + d*x] + C*Tan[c + d
*x]^2),x]
```

```
[Out] -1/2*(a*B*Cot[c + d*x]^2 + 2*(b*B + a*C)*Cot[c + d*x]*Hypergeometric2F1[-1/
2, 1, 1/2, -Tan[c + d*x]^2] + 2*(a*B - b*C)*(Log[Cos[c + d*x]] + Log[Tan[c
+ d*x]]))/d
```

Maple [A] (verified)

Time = 0.34 (sec) , antiderivative size = 77, normalized size of antiderivative = 1.17

method	result
derivativedivides	$\frac{Bb(-\cot(dx+c)-dx-c)+Cb\ln(\sin(dx+c))+Ba\left(-\frac{\cot(dx+c)^2}{2}-\ln(\sin(dx+c))\right)+Ca(-\cot(dx+c)-dx-c)}{d}$
default	$\frac{Bb(-\cot(dx+c)-dx-c)+Cb\ln(\sin(dx+c))+Ba\left(-\frac{\cot(dx+c)^2}{2}-\ln(\sin(dx+c))\right)+Ca(-\cot(dx+c)-dx-c)}{d}$
parallelrisch	$\frac{-Ba\cot(dx+c)^2-2Bbdx-2Cadx-2Bb\cot(dx+c)-2Ba\ln(\tan(dx+c))+B\ln\left(\sec(dx+c)^2\right)a-2Ca\cot(dx+c)+2C\ln(\tan}{2d}$
norman	$\frac{(-Bb-Ca)x\tan(dx+c)^3-\frac{(Bb+Ca)\tan(dx+c)^2}{d}-\frac{Ba\tan(dx+c)}{2d}}{\tan(dx+c)^3}-\frac{(Ba-Cb)\ln(\tan(dx+c))}{d}+\frac{(Ba-Cb)\ln\left(1+\tan(dx+c\right.}{2d}$
risch	$-Bbx-Cax+iBax-iCbx+\frac{2iBac}{d}-\frac{2iCbc}{d}-\frac{2i\left(iBa\,e^{2i(dx+c)}+Bb\,e^{2i(dx+c)}+Ca\,e^{2i(dx+c)}-Bb-Ca\right.}{d\left(e^{2i(dx+c)}-1\right)^2}$

```
[In] int(cot(d*x+c)^4*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,method=_R
ETURNVERBOSE)
```

```
[Out] 1/d*(B*b*(-cot(d*x+c)-d*x-c)+C*b*ln(sin(d*x+c))+B*a*(-1/2*cot(d*x+c)^2-ln(s
in(d*x+c)))+C*a*(-cot(d*x+c)-d*x-c))
```

Fricas [A] (verification not implemented)

none

Time = 0.24 (sec) , antiderivative size = 95, normalized size of antiderivative = 1.44

$$\int \cot^4(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx = -\frac{(Ba-Cb)\log\left(\frac{\tan(dx+c)^2}{\tan(dx+c)^2+1}\right)\tan(dx+c)^2+(2(Ca+Bb)dx+Ba)\tan(dx+c)^2+Ba+2(Ca+Bb)}{2d\tan(dx+c)^2}$$

```
[In] integrate(cot(d*x+c)^4*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="fricas")
```

```
[Out] -1/2*((B*a - C*b)*log(tan(d*x + c)^2/(tan(d*x + c)^2 + 1))*tan(d*x + c)^2 +
 (2*(C*a + B*b)*d*x + B*a)*tan(d*x + c)^2 + B*a + 2*(C*a + B*b)*tan(d*x + c
))/(d*tan(d*x + c)^2)
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 143 vs. 2(56) = 112.

Time = 1.07 (sec) , antiderivative size = 143, normalized size of antiderivative = 2.17

$$\int \cot^4(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx = \begin{cases} \text{NaN} \\ x(a+b\tan(c))\left(B\tan(c)+C\tan^2(c)\right)\cot^4(c) \\ \text{NaN} \\ \frac{Ba\log(\tan^2(c+dx)+1)}{2d} - \frac{Ba\log(\tan(c+dx))}{d} - \frac{Ba}{2d\tan^2(c+dx)} - Bbx - \frac{Bb}{d\tan(c+dx)} - Cax - \frac{Ca}{d\tan(c+dx)} - \frac{Cb\log(\tan^2(c+dx)+1)}{2d} \end{cases}$$

```
[In] integrate(cot(d*x+c)**4*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)**2),x)
```

```
[Out] Piecewise((nan, Eq(c, 0) & Eq(d, 0)), (x*(a + b*tan(c))*(B*tan(c) + C*tan(c
)**2)*cot(c)**4, Eq(d, 0)), (nan, Eq(c, -d*x)), (B*a*log(tan(c + d*x)**2 +
1)/(2*d) - B*a*log(tan(c + d*x))/d - B*a/(2*d*tan(c + d*x)**2) - B*b*x - B*
b/(d*tan(c + d*x)) - C*a*x - C*a/(d*tan(c + d*x)) - C*b*log(tan(c + d*x)**2
 + 1)/(2*d) + C*b*log(tan(c + d*x))/d, True))
```

Maxima [A] (verification not implemented)

none

Time = 0.32 (sec) , antiderivative size = 86, normalized size of antiderivative = 1.30

$$\int \cot^4(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx =$$
$$-\frac{2\,(Ca+Bb)(dx+c)-(Ba-Cb)\log\left(\tan\left(dx+c\right)^2+1\right)+2\,(Ba-Cb)\log\left(\tan\left(dx+c\right)\right)+\frac{Ba+2\,(Ca+B}{\tan(dx}}{2\,d}$$

```
[In] integrate(cot(d*x+c)^4*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="maxima")
```

```
[Out] -1/2*(2*(C*a + B*b)*(d*x + c) - (B*a - C*b)*log(tan(d*x + c)^2 + 1) + 2*(B*
a - C*b)*log(tan(d*x + c)) + (B*a + 2*(C*a + B*b)*tan(d*x + c))/tan(d*x + c
)^2)/d
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 179 vs. 2(64) = 128.

Time = 1.31 (sec) , antiderivative size = 179, normalized size of antiderivative = 2.71

$$\int \cot^4(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx =$$
$$-\frac{Ba\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^2-4\,Ca\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)-4\,Bb\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)+8\,(Ca+Bb)(dx+c)-8\,(Ba-C}{}$$

```
[In] integrate(cot(d*x+c)^4*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="giac")
```

```
[Out] -1/8*(B*a*tan(1/2*d*x + 1/2*c)^2 - 4*C*a*tan(1/2*d*x + 1/2*c) - 4*B*b*tan(1
/2*d*x + 1/2*c) + 8*(C*a + B*b)*(d*x + c) - 8*(B*a - C*b)*log(tan(1/2*d*x +
 1/2*c)^2 + 1) + 8*(B*a - C*b)*log(abs(tan(1/2*d*x + 1/2*c))) - (12*B*a*tan
(1/2*d*x + 1/2*c)^2 - 12*C*b*tan(1/2*d*x + 1/2*c)^2 - 4*C*a*tan(1/2*d*x + 1
/2*c) - 4*B*b*tan(1/2*d*x + 1/2*c) - B*a)/tan(1/2*d*x + 1/2*c)^2)/d
```

Mupad [B] (verification not implemented)

Time = 7.98 (sec) , antiderivative size = 108, normalized size of antiderivative = 1.64

$$\int \cot^4(c+dx)(a+b\tan(c+dx)) \left(B\tan(c+dx)+C\tan^2(c+dx)\right) \,dx$$
$$= -\frac{\ln(\tan(c+d\,x))\,(B\,a-C\,b)}{d} - \frac{\cot(c+d\,x)^2\,\left(\frac{B\,a}{2}+\tan(c+d\,x)\,(B\,b+C\,a)\right)}{d} + \frac{\ln(\tan(c+d\,x)-\mathrm{i})\,(B+C\,1\mathrm{i})\,(a+b\,1\mathrm{i})}{2\,d} - \frac{\ln(\tan(c+d\,x)+1\mathrm{i})\,(B-C\,1\mathrm{i})\,(b+a\,1\mathrm{i})\,1\mathrm{i}}{2\,d}$$

```
[In] int(cot(c + d*x)^4*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))
,x)
```

```
[Out] (log(tan(c + d*x) - 1i)*(B + C*1i)*(a + b*1i))/(2*d) - (cot(c + d*x)^2*((B*
a)/2 + tan(c + d*x)*(B*b + C*a)))/d - (log(tan(c + d*x))*(B*a - C*b))/d - (
log(tan(c + d*x) + 1i)*(B - C*1i)*(a*1i + b)*1i)/(2*d)
```

3.7 $\int \cot^5(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)$

Optimal result 106
Rubi [A] (verified) 106
Mathematica [C] (verified) 108
Maple [A] (verified) 109
Fricas [A] (verification not implemented) 109
Sympy [B] (verification not implemented) 110
Maxima [A] (verification not implemented) 110
Giac [B] (verification not implemented) 111
Mupad [B] (verification not implemented) 111

Optimal result

Integrand size = 38, antiderivative size = 87

$$\int \cot^5(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx = (aB-bC)x+\frac{(aB-bC)\cot(c+dx)}{d}-\frac{(bB+aC)\cot^2(c+dx)}{2d}-\frac{aB\cot^3(c+dx)}{3d}-\frac{(bB+aC)\log(\sin(c+dx))}{d}$$

```
[Out] (B*a-C*b)*x+(B*a-C*b)*cot(d*x+c)/d-1/2*(B*b+C*a)*cot(d*x+c)^2/d-1/3*a*B*cot
(d*x+c)^3/d-(B*b+C*a)*ln(sin(d*x+c))/d
```

Rubi [A] (verified)

Time = 0.25 (sec) , antiderivative size = 87, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.132, Rules used = {3713, 3672, 3610, 3612, 3556}

$$\int \cot^5(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx = -\frac{(aC+bB)\cot^2(c+dx)}{2d}+\frac{(aB-bC)\cot(c+dx)}{d}-\frac{(aC+bB)\log(\sin(c+dx))}{d}+x(aB-bC)-\frac{aB\cot^3(c+dx)}{3d}$$

```
[In] Int[Cot[c + d*x]^5*(a + b*Tan[c + d*x])*(B*Tan[c + d*x] + C*Tan[c + d*x]^2)
,x]
```

```
[Out] (a*B - b*C)*x + ((a*B - b*C)*Cot[c + d*x])/d - ((b*B + a*C)*Cot[c + d*x]^2)
/(2*d) - (a*B*Cot[c + d*x]^3)/(3*d) - ((b*B + a*C)*Log[Sin[c + d*x]])/d
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3610

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(b*c - a*d)*((a + b*Tan[e + f*x])^(m + 1)/
(f*(m + 1)*(a^2 + b^2))), x] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])
^(m + 1)*Simp[a*c + b*d - (b*c - a*d)*Tan[e + f*x], x], x], x] /; FreeQ[{a,
 b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && LtQ[m, -1]
```

Rule 3612

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[(a*c + b*d)*(x/(a^2 + b^2)), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3672

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[(b
*c - a*d)*(A*b - a*B)*((a + b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1)*(a^2 + b^2
))), x] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[a*A*c +
 b*B*c + A*b*d - a*B*d - (A*b*c - a*B*c - a*A*d - b*B*d)*Tan[e + f*x], x],
x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] && NeQ[b*c - a*d, 0] && LtQ[m,
 -1] && NeQ[a^2 + b^2, 0]
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\text{integral} = \int \cot^4(c+dx)(a+b\tan(c+dx))(B+C\tan(c+dx))\,dx$$

$$= -\frac{aB\cot^3(c+dx)}{3d} + \int \cot^3(c+dx)(bB+aC-(aB-bC)\tan(c+dx))\,dx$$

$$= -\frac{(bB + aC)\cot^2(c + dx)}{2d} - \frac{aB\cot^3(c + dx)}{3d} + \int \cot^2(c + dx)(-aB + bC - (bB + aC)\tan(c + dx))\,dx$$

$$= \frac{(aB - bC)\cot(c + dx)}{d} - \frac{(bB + aC)\cot^2(c + dx)}{2d} - \frac{aB\cot^3(c + dx)}{3d} + \int \cot(c + dx)(-bB - aC + (aB - bC)\tan(c + dx))\,dx$$

$$= (aB - bC)x + \frac{(aB - bC)\cot(c + dx)}{d} - \frac{(bB + aC)\cot^2(c + dx)}{2d} - \frac{aB\cot^3(c + dx)}{3d} + (-bB - aC)\int \cot(c + dx)\,dx$$

$$= (aB - bC)x + \frac{(aB - bC)\cot(c + dx)}{d} - \frac{(bB + aC)\cot^2(c + dx)}{2d} - \frac{aB\cot^3(c + dx)}{3d} - \frac{(bB + aC)\log(\sin(c + dx))}{d}$$

Mathematica [C] (verified)

Result contains higher order function than in optimal. Order 5 vs. order 3 in optimal.

Time = 1.08 (sec) , antiderivative size = 101, normalized size of antiderivative = 1.16

$$\int \cot^5(c + dx)(a + b\tan(c + dx))\left(B\tan(c + dx) + C\tan^2(c + dx)\right)\,dx = -\frac{2aB\cot^3(c + dx)\,\text{Hypergeometric2F1}\left(-\frac{3}{2}, 1, -\frac{1}{2}, -\tan^2(c + dx)\right) + 6bC\cot(c + dx)\,\text{Hypergeometric2F}}{6d}$$

```
[In] Integrate[Cot[c + d*x]^5*(a + b*Tan[c + d*x])*(B*Tan[c + d*x] + C*Tan[c + d
*x]^2),x]
```

```
[Out] -1/6*(2*a*B*Cot[c + d*x]^3*Hypergeometric2F1[-3/2, 1, -1/2, -Tan[c + d*x]^2
] + 6*b*C*Cot[c + d*x]*Hypergeometric2F1[-1/2, 1, 1/2, -Tan[c + d*x]^2] + 3
*(b*B + a*C)*(Cot[c + d*x]^2 + 2*(Log[Cos[c + d*x]] + Log[Tan[c + d*x]])))/
d
```

Maple [A] (verified)

Time = 0.36 (sec) , antiderivative size = 95, normalized size of antiderivative = 1.09

method	result
derivativedivides	$\frac{Bb\left(-\frac{\cot(dx+c)^2}{2}-\ln(\sin(dx+c))\right)+Cb(-\cot(dx+c)-dx-c)+Ba\left(-\frac{\cot(dx+c)^3}{3}+\cot(dx+c)+dx+c\right)+Ca\left(-\frac{\cot(dx+c}{2}\right.}{d}$
default	$\frac{Bb\left(-\frac{\cot(dx+c)^2}{2}-\ln(\sin(dx+c))\right)+Cb(-\cot(dx+c)-dx-c)+Ba\left(-\frac{\cot(dx+c)^3}{3}+\cot(dx+c)+dx+c\right)+Ca\left(-\frac{\cot(dx+c}{2}\right.}{d}$
parallelrisch	$\frac{-2Ba\cot(dx+c)^3-3Bb\cot(dx+c)^2+6Badx-3Ca\cot(dx+c)^2-6Cbdx+6B\cot(dx+c)a-6B\ln(\tan(dx+c))b+3B\ln\left(\mathrm{se}\right.}{6d}$
norman	$\frac{\frac{(Ba-Cb)\tan(dx+c)^3}{d}+(Ba-Cb)x\tan(dx+c)^4-\frac{(Bb+Ca)\tan(dx+c)^2}{2d}-\frac{Ba\tan(dx+c)}{3d}}{\tan(dx+c)^4}-\frac{(Bb+Ca)\ln(\tan(dx+c))}{d}+\frac{(Bb}{}$
risch	$iBbx+iCax+Bax-Cbx+\frac{2iBbc}{d}+\frac{2iCac}{d}-\frac{2i\left(3iBb\,\mathrm{e}^{4i(dx+c)}+3iCa\,\mathrm{e}^{4i(dx+c)}-6Ba\,\mathrm{e}^{4i(dx+c)}+3Cl\right.}{}$

```
[In] int(cot(d*x+c)^5*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,method=_R
ETURNVERBOSE)
```

```
[Out] 1/d*(B*b*(-1/2*cot(d*x+c)^2-ln(sin(d*x+c)))+C*b*(-cot(d*x+c)-d*x-c)+B*a*(-1
/3*cot(d*x+c)^3+cot(d*x+c)+d*x+c)+C*a*(-1/2*cot(d*x+c)^2-ln(sin(d*x+c))))
```

Fricas [A] (verification not implemented)

none

Time = 0.25 (sec) , antiderivative size = 121, normalized size of antiderivative = 1.39

$$\int \cot^5(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx = -\frac{3\,(Ca+Bb)\log\left(\frac{\tan(dx+c)^2}{\tan(dx+c)^2+1}\right)\tan\left(dx+c\right)^3-3\,(2\,(Ba-Cb)dx-Ca-Bb)\tan\left(dx+c\right)^3-6\,(Ba-C}{6\,d\tan\left(dx+c\right)^3}$$

```
[In] integrate(cot(d*x+c)^5*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="fricas")
```

```
[Out] -1/6*(3*(C*a + B*b)*log(tan(d*x + c)^2/(tan(d*x + c)^2 + 1))*tan(d*x + c)^3
 - 3*(2*(B*a - C*b)*d*x - C*a - B*b)*tan(d*x + c)^3 - 6*(B*a - C*b)*tan(d*x
 + c)^2 + 2*B*a + 3*(C*a + B*b)*tan(d*x + c))/(d*tan(d*x + c)^3)
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 173 vs. 2(75) = 150.

Time = 1.88 (sec) , antiderivative size = 173, normalized size of antiderivative = 1.99

$$\int \cot^5(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\begin{cases}\text{NaN}\\ x(a+b\tan{(c)})\left(B\tan{(c)}+C\tan^2{(c)}\right)\cot^5{(c)}\\ \text{NaN}\\ Bax+\frac{Ba}{d\tan{(c+dx)}}-\frac{Ba}{3d\tan^3{(c+dx)}}+\frac{Bb\log{(\tan^2{(c+dx)}+1)}}{2d}-\frac{Bb\log{(\tan{(c+dx)})}}{d}-\frac{Bb}{2d\tan^2{(c+dx)}}+\frac{Ca\log{(\tan^2{(c+dx)}+1)}}{2d}\end{cases}$$

```
[In] integrate(cot(d*x+c)**5*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)**2),x)
```

```
[Out] Piecewise((nan, Eq(c, 0) & Eq(d, 0)), (x*(a + b*tan(c))*(B*tan(c) + C*tan(c
)**2)*cot(c)**5, Eq(d, 0)), (nan, Eq(c, -d*x)), (B*a*x + B*a/(d*tan(c + d*x
)) - B*a/(3*d*tan(c + d*x)**3) + B*b*log(tan(c + d*x)**2 + 1)/(2*d) - B*b*l
og(tan(c + d*x))/d - B*b/(2*d*tan(c + d*x)**2) + C*a*log(tan(c + d*x)**2 +
1)/(2*d) - C*a*log(tan(c + d*x))/d - C*a/(2*d*tan(c + d*x)**2) - C*b*x - C*
b/(d*tan(c + d*x)), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.33 (sec) , antiderivative size = 104, normalized size of antiderivative = 1.20

$$\int \cot^5(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{6\,(Ba-Cb)(dx+c)+3\,(Ca+Bb)\log\left(\tan{(dx+c)}^2+1\right)-6\,(Ca+Bb)\log\left(\tan{(dx+c)}\right)+\frac{6\,(Ba-Cb)\tan}{}}{6\,d}$$

```
[In] integrate(cot(d*x+c)^5*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="maxima")
```

```
[Out] 1/6*(6*(B*a - C*b)*(d*x + c) + 3*(C*a + B*b)*log(tan(d*x + c)^2 + 1) - 6*(C
*a + B*b)*log(tan(d*x + c)) + (6*(B*a - C*b)*tan(d*x + c)^2 - 2*B*a - 3*(C*
a + B*b)*tan(d*x + c))/tan(d*x + c)^3)/d
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 237 vs. 2(83) = 166.

Time = 1.61 (sec) , antiderivative size = 237, normalized size of antiderivative = 2.72

$$\int \cot^5(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{Ba\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^3-3\,Ca\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^2-3\,Bb\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^2-15\,Ba\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)+12\,Cb\,\mathrm{t}}{}$$

```
[In] integrate(cot(d*x+c)^5*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="giac")
```

```
[Out] 1/24*(B*a*tan(1/2*d*x + 1/2*c)^3 - 3*C*a*tan(1/2*d*x + 1/2*c)^2 - 3*B*b*tan
(1/2*d*x + 1/2*c)^2 - 15*B*a*tan(1/2*d*x + 1/2*c) + 12*C*b*tan(1/2*d*x + 1/
2*c) + 24*(B*a - C*b)*(d*x + c) + 24*(C*a + B*b)*log(tan(1/2*d*x + 1/2*c)^2
 + 1) - 24*(C*a + B*b)*log(abs(tan(1/2*d*x + 1/2*c))) + (44*C*a*tan(1/2*d*x
 + 1/2*c)^3 + 44*B*b*tan(1/2*d*x + 1/2*c)^3 + 15*B*a*tan(1/2*d*x + 1/2*c)^2
 - 12*C*b*tan(1/2*d*x + 1/2*c)^2 - 3*C*a*tan(1/2*d*x + 1/2*c) - 3*B*b*tan(1
/2*d*x + 1/2*c) - B*a)/tan(1/2*d*x + 1/2*c)^3)/d
```

Mupad [B] (verification not implemented)

Time = 7.86 (sec) , antiderivative size = 127, normalized size of antiderivative = 1.46

$$\int \cot^5(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=-\frac{\cot(c+d\,x)^3\,\left((C\,b-B\,a)\,\tan(c+d\,x)^2+\left(\frac{B\,b}{2}+\frac{C\,a}{2}\right)\,\tan(c+d\,x)+\frac{B\,a}{3}\right)}{d}$$
$$-\frac{\ln\left(\tan(c+d\,x)\right)\,(B\,b+C\,a)}{d}-\frac{\ln\left(\tan(c+d\,x)-\mathrm{i}\right)\,(B+C\,1\mathrm{i})\,(a+b\,1\mathrm{i})\,1\mathrm{i}}{2\,d}$$
$$+\frac{\ln\left(\tan(c+d\,x)+1\mathrm{i}\right)\,(B-C\,1\mathrm{i})\,(b+a\,1\mathrm{i})}{2\,d}$$

```
[In] int(cot(c + d*x)^5*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))
,x)
```

```
[Out] (log(tan(c + d*x) + 1i)*(B - C*1i)*(a*1i + b))/(2*d) - (log(tan(c + d*x))*(
B*b + C*a))/d - (log(tan(c + d*x) - 1i)*(B + C*1i)*(a + b*1i)*1i)/(2*d) - (
cot(c + d*x)^3*((B*a)/3 + tan(c + d*x)*((B*b)/2 + (C*a)/2) - tan(c + d*x)^2
*(B*a - C*b)))/d
```

3.8 $\int \cot^6(c+dx)(a+b\tan(c+dx)) (B\tan(c+dx)+C\tan^2($

Optimal result 112
Rubi [A] (verified) 112
Mathematica [C] (verified) 114
Maple [A] (verified) 115
Fricas [A] (verification not implemented) 115
Sympy [B] (verification not implemented) 116
Maxima [A] (verification not implemented) 116
Giac [B] (verification not implemented) 117
Mupad [B] (verification not implemented) 117

Optimal result

Integrand size = 38, antiderivative size = 108

$$\begin{aligned}&\int \cot^6(c+dx)(a+b\tan(c+dx)) \left(B\tan(c+dx)+C\tan^2(c+dx)\right) \,dx\\&=(bB+aC)x+\frac{(bB+aC)\cot(c+dx)}{d}+\frac{(aB-bC)\cot^2(c+dx)}{2d}\\&\quad-\frac{(bB+aC)\cot^3(c+dx)}{3d}-\frac{aB\cot^4(c+dx)}{4d}+\frac{(aB-bC)\log(\sin(c+dx))}{d}\end{aligned}$$

```
[Out] (B*b+C*a)*x+(B*b+C*a)*cot(d*x+c)/d+1/2*(B*a-C*b)*cot(d*x+c)^2/d-1/3*(B*b+C*
a)*cot(d*x+c)^3/d-1/4*a*B*cot(d*x+c)^4/d+(B*a-C*b)*ln(sin(d*x+c))/d
```

Rubi [A] (verified)

Time = 0.29 (sec) , antiderivative size = 108, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.132, Rules used = {3713, 3672, 3610, 3612, 3556}

$$\begin{aligned}&\int \cot^6(c+dx)(a+b\tan(c+dx)) \left(B\tan(c+dx)+C\tan^2(c+dx)\right) \,dx\\&=-\frac{(aC+bB)\cot^3(c+dx)}{3d}+\frac{(aB-bC)\cot^2(c+dx)}{2d}+\frac{(aC+bB)\cot(c+dx)}{d}\\&\quad+\frac{(aB-bC)\log(\sin(c+dx))}{d}+x(aC+bB)-\frac{aB\cot^4(c+dx)}{4d}\end{aligned}$$

```
[In] Int[Cot[c + d*x]^6*(a + b*Tan[c + d*x])*(B*Tan[c + d*x] + C*Tan[c + d*x]^2)
,x]
```

```
[Out] (b*B + a*C)*x + ((b*B + a*C)*Cot[c + d*x])/d + ((a*B - b*C)*Cot[c + d*x]^2)
/(2*d) - ((b*B + a*C)*Cot[c + d*x]^3)/(3*d) - (a*B*Cot[c + d*x]^4)/(4*d) +
((a*B - b*C)*Log[Sin[c + d*x]])/d
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3610

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(b*c - a*d)*((a + b*Tan[e + f*x])^(m + 1)/
(f*(m + 1)*(a^2 + b^2))), x] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])
^(m + 1)*Simp[a*c + b*d - (b*c - a*d)*Tan[e + f*x], x], x], x] /; FreeQ[{a,
 b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && LtQ[m, -1]
```

Rule 3612

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[(a*c + b*d)*(x/(a^2 + b^2)), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3672

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[(b
*c - a*d)*(A*b - a*B)*((a + b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1)*(a^2 + b^2
))), x] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[a*A*c +
 b*B*c + A*b*d - a*B*d - (A*b*c - a*B*c - a*A*d - b*B*d)*Tan[e + f*x], x],
x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] && NeQ[b*c - a*d, 0] && LtQ[m,
 -1] && NeQ[a^2 + b^2, 0]
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\begin{aligned}\text{integral} &= \int \cot^5(c+dx)(a+b\tan(c+dx))(B+C\tan(c+dx))\,dx\\ &= -\frac{aB\cot^4(c+dx)}{4d}+\int \cot^4(c+dx)(bB+aC-(aB-bC)\tan(c+dx))\,dx\end{aligned}$$

$$
\begin{aligned}
&= -\frac{(bB+aC)\cot^3(c+dx)}{3d} - \frac{aB\cot^4(c+dx)}{4d} \\
&\quad + \int \cot^3(c+dx)(-aB+bC-(bB+aC)\tan(c+dx))\,dx \\
&= \frac{(aB-bC)\cot^2(c+dx)}{2d} - \frac{(bB+aC)\cot^3(c+dx)}{3d} - \frac{aB\cot^4(c+dx)}{4d} \\
&\quad + \int \cot^2(c+dx)(-bB-aC+(aB-bC)\tan(c+dx))\,dx \\
&= \frac{(bB+aC)\cot(c+dx)}{d} + \frac{(aB-bC)\cot^2(c+dx)}{2d} - \frac{(bB+aC)\cot^3(c+dx)}{3d} \\
&\quad - \frac{aB\cot^4(c+dx)}{4d} + \int \cot(c+dx)(aB-bC+(bB+aC)\tan(c+dx))\,dx \\
&= (bB+aC)x + \frac{(bB+aC)\cot(c+dx)}{d} + \frac{(aB-bC)\cot^2(c+dx)}{2d} \\
&\quad - \frac{(bB+aC)\cot^3(c+dx)}{3d} - \frac{aB\cot^4(c+dx)}{4d} + (aB-bC)\int \cot(c+dx)\,dx \\
&= (bB+aC)x + \frac{(bB+aC)\cot(c+dx)}{d} + \frac{(aB-bC)\cot^2(c+dx)}{2d} \\
&\quad - \frac{(bB+aC)\cot^3(c+dx)}{3d} - \frac{aB\cot^4(c+dx)}{4d} + \frac{(aB-bC)\log(\sin(c+dx))}{d}
\end{aligned}
$$

Mathematica [C] (verified)

Result contains higher order function than in optimal. Order 5 vs. order 3 in optimal.

Time = 1.25 (sec) , antiderivative size = 100, normalized size of antiderivative = 0.93

$$
\int \cot^6(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx = -\frac{4(bB+aC)\cot^3(c+dx)\,\text{Hypergeometric2F1}\left(-\frac{3}{2},1,-\frac{1}{2},-\tan^2(c+dx)\right)+3((-2aB+2bC)\cot^2(c+a}{12d}
$$

```
[In] Integrate[Cot[c + d*x]^6*(a + b*Tan[c + d*x])*(B*Tan[c + d*x] + C*Tan[c + d
*x]^2),x]
```

```
[Out] -1/12*(4*(b*B + a*C)*Cot[c + d*x]^3*Hypergeometric2F1[-3/2, 1, -1/2, -Tan[c
 + d*x]^2] + 3*((-2*a*B + 2*b*C)*Cot[c + d*x]^2 + a*B*Cot[c + d*x]^4 - 4*(a
*B - b*C)*(Log[Cos[c + d*x]] + Log[Tan[c + d*x]])))/d
```

Maple [A] (verified)

Time = 0.35 (sec) , antiderivative size = 108, normalized size of antiderivative = 1.00

method	result
derivativedivides	$\frac{Bb\left(-\frac{\cot(dx+c)^3}{3}+\cot(dx+c)+dx+c\right)+Cb\left(-\frac{\cot(dx+c)^2}{2}-\ln(\sin(dx+c))\right)+Ba\left(-\frac{\cot(dx+c)^4}{4}+\frac{\cot(dx+c)^2}{2}+\ln(\sin(dx-}{d}$
default	$\frac{Bb\left(-\frac{\cot(dx+c)^3}{3}+\cot(dx+c)+dx+c\right)+Cb\left(-\frac{\cot(dx+c)^2}{2}-\ln(\sin(dx+c))\right)+Ba\left(-\frac{\cot(dx+c)^4}{4}+\frac{\cot(dx+c)^2}{2}+\ln(\sin(dx-}{d}$
norman	$\frac{\frac{(Bb+Ca)\tan(dx+c)^4}{d}+(Bb+Ca)x\tan(dx+c)^5-\frac{(Bb+Ca)\tan(dx+c)^2}{3d}+\frac{(Ba-Cb)\tan(dx+c)^3}{2d}-\frac{Ba\tan(dx+c)}{4d}}{\tan(dx+c)^5}+\frac{(Ba-Cb)\ln}{}$
parallelrisch	$\frac{-3Ba\cot(dx+c)^4-4Bb\cot(dx+c)^3-4Ca\cot(dx+c)^3+6Ba\cot(dx+c)^2+12Bbdx-6Cb\cot(dx+c)^2+12Cadx+12Bb\cot}{12d}$
risch	$Bbx+Cax-iBax+iCbx-\frac{2iBac}{d}+\frac{2iCbc}{d}-\frac{2\left(-6iBb\,e^{6i(dx+c)}-6iCa\,e^{6i(dx+c)}+6Ba\,e^{6i(dx+c)}-3C\right.}{}$

[In] int(cot(d*x+c)^6*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,method=_RETURNVERBOSE)

[Out] 1/d*(B*b*(-1/3*cot(d*x+c)^3+cot(d*x+c)+d*x+c)+C*b*(-1/2*cot(d*x+c)^2-ln(sin(d*x+c)))+B*a*(-1/4*cot(d*x+c)^4+1/2*cot(d*x+c)^2+ln(sin(d*x+c)))+C*a*(-1/3*cot(d*x+c)^3+cot(d*x+c)+d*x+c))

Fricas [A] (verification not implemented)

none

Time = 0.25 (sec) , antiderivative size = 138, normalized size of antiderivative = 1.28

$$\int \cot^6(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{6\,(Ba-Cb)\log\left(\frac{\tan(dx+c)^2}{\tan(dx+c)^2+1}\right)\tan\left(dx+c\right)^4+3\,(4\,(Ca+Bb)dx+3\,Ba-2\,Cb)\tan\left(dx+c\right)^4+12\,(Ca+}{12\,d\tan\left(dx+c\right)^4}$$

[In] integrate(cot(d*x+c)^6*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, algorithm="fricas")

[Out] 1/12*(6*(B*a - C*b)*log(tan(d*x + c)^2/(tan(d*x + c)^2 + 1))*tan(d*x + c)^4 + 3*(4*(C*a + B*b)*d*x + 3*B*a - 2*C*b)*tan(d*x + c)^4 + 12*(C*a + B*b)*tan(d*x + c)^3 + 6*(B*a - C*b)*tan(d*x + c)^2 - 3*B*a - 4*(C*a + B*b)*tan(d*x + c))/(d*tan(d*x + c)^4)

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 204 vs. 2(95) = 190.

Time = 2.32 (sec) , antiderivative size = 204, normalized size of antiderivative = 1.89

$$\int \cot^6(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$= \begin{cases} \mathrm{NaN} \\ x(a+b\tan{(c)})\left(B\tan{(c)}+C\tan^2{(c)}\right)\cot^6{(c)} \\ \mathrm{NaN} \\ -\frac{Ba\log{(\tan^2{(c+dx)}+1)}}{2d}+\frac{Ba\log{(\tan{(c+dx)})}}{d}+\frac{Ba}{2d\tan^2{(c+dx)}}-\frac{Ba}{4d\tan^4{(c+dx)}}+Bbx+\frac{Bb}{d\tan{(c+dx)}}-\frac{Bb}{3d\tan^3{(c+dx)}}+C \end{cases}$$

```
[In] integrate(cot(d*x+c)**6*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)**2),x)
```

```
[Out] Piecewise((nan, Eq(c, 0) & Eq(d, 0)), (x*(a + b*tan(c))*(B*tan(c) + C*tan(c
)**2)*cot(c)**6, Eq(d, 0)), (nan, Eq(c, -d*x)), (-B*a*log(tan(c + d*x)**2 +
 1)/(2*d) + B*a*log(tan(c + d*x))/d + B*a/(2*d*tan(c + d*x)**2) - B*a/(4*d*
tan(c + d*x)**4) + B*b*x + B*b/(d*tan(c + d*x)) - B*b/(3*d*tan(c + d*x)**3)
 + C*a*x + C*a/(d*tan(c + d*x)) - C*a/(3*d*tan(c + d*x)**3) + C*b*log(tan(c
 + d*x)**2 + 1)/(2*d) - C*b*log(tan(c + d*x))/d - C*b/(2*d*tan(c + d*x)**2)
, True))
```

Maxima [A] (verification not implemented)

none

Time = 0.31 (sec) , antiderivative size = 122, normalized size of antiderivative = 1.13

$$\int \cot^6(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$= \frac{12\,(Ca+Bb)(dx+c)-6\,(Ba-Cb)\log\left(\tan{(dx+c)}^2+1\right)+12\,(Ba-Cb)\log\left(\tan{(dx+c)}\right)+\frac{12\,(Ca+Bb)}{}}{12\,d}$$

```
[In] integrate(cot(d*x+c)^6*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="maxima")
```

```
[Out] 1/12*(12*(C*a + B*b)*(d*x + c) - 6*(B*a - C*b)*log(tan(d*x + c)^2 + 1) + 12
*(B*a - C*b)*log(tan(d*x + c)) + (12*(C*a + B*b)*tan(d*x + c)^3 + 6*(B*a -
C*b)*tan(d*x + c)^2 - 3*B*a - 4*(C*a + B*b)*tan(d*x + c))/tan(d*x + c)^4)/d
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 299 vs. 2(102) = 204.

Time = 1.34 (sec) , antiderivative size = 299, normalized size of antiderivative = 2.77

$$\int \cot^6(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx =$$

$$-\frac{3\,Ba\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^4-8\,Ca\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^3-8\,Bb\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^3-36\,Ba\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^2+24}{}$$

```
[In] integrate(cot(d*x+c)^6*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="giac")
```

```
[Out] -1/192*(3*B*a*tan(1/2*d*x + 1/2*c)^4 - 8*C*a*tan(1/2*d*x + 1/2*c)^3 - 8*B*b
*tan(1/2*d*x + 1/2*c)^3 - 36*B*a*tan(1/2*d*x + 1/2*c)^2 + 24*C*b*tan(1/2*d*
x + 1/2*c)^2 + 120*C*a*tan(1/2*d*x + 1/2*c) + 120*B*b*tan(1/2*d*x + 1/2*c)
- 192*(C*a + B*b)*(d*x + c) + 192*(B*a - C*b)*log(tan(1/2*d*x + 1/2*c)^2 +
1) - 192*(B*a - C*b)*log(abs(tan(1/2*d*x + 1/2*c))) + (400*B*a*tan(1/2*d*x
+ 1/2*c)^4 - 400*C*b*tan(1/2*d*x + 1/2*c)^4 - 120*C*a*tan(1/2*d*x + 1/2*c)^
3 - 120*B*b*tan(1/2*d*x + 1/2*c)^3 - 36*B*a*tan(1/2*d*x + 1/2*c)^2 + 24*C*b
*tan(1/2*d*x + 1/2*c)^2 + 8*C*a*tan(1/2*d*x + 1/2*c) + 8*B*b*tan(1/2*d*x +
1/2*c) + 3*B*a)/tan(1/2*d*x + 1/2*c)^4)/d
```

Mupad [B] (verification not implemented)

Time = 8.64 (sec) , antiderivative size = 145, normalized size of antiderivative = 1.34

$$\int \cot^6(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$

$$= \frac{\ln\left(\tan(c+d\,x)\right)\,\left(B\,a-C\,b\right)}{d} - \frac{\cot(c+d\,x)^4\,\left(\left(-B\,b-C\,a\right)\,\tan(c+d\,x)^3+\left(\frac{C\,b}{2}-\frac{B\,a}{2}\right)\,\tan(c+d\,x)^2+\left(\frac{B\,b}{3}+\frac{C\,a}{3}\right)\,\tan(c+d\,x)+\frac{B\,a}{4}\right)}{d} - \frac{\ln\left(\tan(c+d\,x)-\mathrm{i}\right)\,\left(B+C\,1\mathrm{i}\right)\,\left(a+b\,1\mathrm{i}\right)}{2\,d} + \frac{\ln\left(\tan(c+d\,x)+1\mathrm{i}\right)\,\left(B-C\,1\mathrm{i}\right)\,\left(b+a\,1\mathrm{i}\right)\,1\mathrm{i}}{2\,d}$$

```
[In] int(cot(c + d*x)^6*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))
,x)
```

```
[Out] (log(tan(c + d*x))*(B*a - C*b))/d - (cot(c + d*x)^4*((B*a)/4 + tan(c + d*x)
*((B*b)/3 + (C*a)/3) - tan(c + d*x)^3*(B*b + C*a) - tan(c + d*x)^2*((B*a)/2
 - (C*b)/2)))/d - (log(tan(c + d*x) - 1i)*(B + C*1i)*(a + b*1i))/(2*d) + (l
og(tan(c + d*x) + 1i)*(B - C*1i)*(a*1i + b)*1i)/(2*d)
```

3.9 $\int \tan(c+dx)(a+b\tan(c+dx))^2 \left(B\tan(c+dx)+C\tan^2(\right.$

Optimal result 118
Rubi [A] (verified) 118
Mathematica [C] (verified) 121
Maple [A] (verified) 121
Fricas [A] (verification not implemented) 122
Sympy [A] (verification not implemented) 122
Maxima [A] (verification not implemented) 123
Giac [B] (verification not implemented) 123
Mupad [B] (verification not implemented) 125

Optimal result

Integrand size = 38, antiderivative size = 148

$$\begin{aligned}&\int \tan(c+dx)(a+b\tan(c+dx))^2 \left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx\\&=-\left(\left(a^2B-b^2B-2abC\right)x\right)+\frac{(2abB+a^2C-b^2C)\log(\cos(c+dx))}{d}\\&\quad-\frac{b(bB+aC)\tan(c+dx)}{d}-\frac{C(a+b\tan(c+dx))^2}{2d}\\&\quad+\frac{(4bB-aC)(a+b\tan(c+dx))^3}{12b^2d}+\frac{C\tan(c+dx)(a+b\tan(c+dx))^3}{4bd}\end{aligned}$$

```
[Out] -(B*a^2-B*b^2-2*C*a*b)*x+(2*B*a*b+C*a^2-C*b^2)*ln(cos(d*x+c))/d-b*(B*b+C*a)
*tan(d*x+c)/d-1/2*C*(a+b*tan(d*x+c))^2/d+1/12*(4*B*b-C*a)*(a+b*tan(d*x+c))^
3/b^2/d+1/4*C*tan(d*x+c)*(a+b*tan(d*x+c))^3/b/d
```

Rubi [A] (verified)

Time = 0.36 (sec) , antiderivative size = 148, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.158, Rules used = {3713, 3688, 3711, 3609, 3606, 3556}

$$\begin{aligned}&\int \tan(c+dx)(a+b\tan(c+dx))^2 \left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx\\&=\frac{(a^2C+2abB-b^2C)\log(\cos(c+dx))}{d}-x\left(a^2B-2abC-b^2B\right)\\&\quad+\frac{(4bB-aC)(a+b\tan(c+dx))^3}{12b^2d}-\frac{b(aC+bB)\tan(c+dx)}{d}\\&\quad+\frac{C\tan(c+dx)(a+b\tan(c+dx))^3}{4bd}-\frac{C(a+b\tan(c+dx))^2}{2d}\end{aligned}$$

```
[In] Int[Tan[c + d*x]*(a + b*Tan[c + d*x])^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^2)
,x]
```

```
[Out] -((a^2*B - b^2*B - 2*a*b*C)*x) + ((2*a*b*B + a^2*C - b^2*C)*Log[Cos[c + d*x
]])/d - (b*(b*B + a*C)*Tan[c + d*x])/d - (C*(a + b*Tan[c + d*x])^2)/(2*d) +
 ((4*b*B - a*C)*(a + b*Tan[c + d*x])^3)/(12*b^2*d) + (C*Tan[c + d*x]*(a + b
*Tan[c + d*x])^3)/(4*b*d)
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3606

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
 f*x], x], x] + Simp[b*d*(Tan[e + f*x]/f), x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3609

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[d*((a + b*Tan[e + f*x])^m/(f*m)), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3688

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Si
mp[b*B*(a + b*Tan[e + f*x])^(m - 1)*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(m +
 n))), x] + Dist[1/(d*(m + n)), Int[(a + b*Tan[e + f*x])^(m - 2)*(c + d*Tan
[e + f*x])^n*Simp[a^2*A*d*(m + n) - b*B*(b*c*(m - 1) + a*d*(n + 1)) + d*(m
+ n)*(2*a*A*b + B*(a^2 - b^2))*Tan[e + f*x] - (b*B*(b*c - a*d)*(m - 1) - b*
(A*b + a*B)*d*(m + n))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e,
f, A, B, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2,
 0] && GtQ[m, 1] && (IntegerQ[m] || IntegersQ[2*m, 2*n]) &&  !(IGtQ[n, 1] &
& ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
```

```
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \tan^2(c+dx)(a+b\tan(c+dx))^2(B+C\tan(c+dx))\,dx \\
&= \frac{C\tan(c+dx)(a+b\tan(c+dx))^3}{4bd} \\
&\quad + \frac{\int (a+b\tan(c+dx))^2\left(-aC-4bC\tan(c+dx)+(4bB-aC)\tan^2(c+dx)\right)\,dx}{4b} \\
&= \frac{(4bB-aC)(a+b\tan(c+dx))^3}{12b^2d}+\frac{C\tan(c+dx)(a+b\tan(c+dx))^3}{4bd} \\
&\quad + \frac{\int (a+b\tan(c+dx))^2(-4bB-4bC\tan(c+dx))\,dx}{4b} \\
&= -\frac{C(a+b\tan(c+dx))^2}{2d}+\frac{(4bB-aC)(a+b\tan(c+dx))^3}{12b^2d} \\
&\quad + \frac{C\tan(c+dx)(a+b\tan(c+dx))^3}{4bd} \\
&\quad + \frac{\int (a+b\tan(c+dx))(-4b(aB-bC)-4b(bB+aC)\tan(c+dx))\,dx}{4b} \\
&= -\left(\left(a^2B-b^2B-2abC\right)x\right)-\frac{b(bB+aC)\tan(c+dx)}{d} \\
&\quad - \frac{C(a+b\tan(c+dx))^2}{2d}+\frac{(4bB-aC)(a+b\tan(c+dx))^3}{12b^2d} \\
&\quad + \frac{C\tan(c+dx)(a+b\tan(c+dx))^3}{4bd}+\left(-2abB-a^2C+b^2C\right)\int \tan(c+dx)\,dx \\
&= -\left(\left(a^2B-b^2B-2abC\right)x\right)+\frac{\left(2abB+a^2C-b^2C\right)\log(\cos(c+dx))}{d} \\
&\quad - \frac{b(bB+aC)\tan(c+dx)}{d}-\frac{C(a+b\tan(c+dx))^2}{2d} \\
&\quad + \frac{(4bB-aC)(a+b\tan(c+dx))^3}{12b^2d}+\frac{C\tan(c+dx)(a+b\tan(c+dx))^3}{4bd}
\end{aligned}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 6.26 (sec) , antiderivative size = 221, normalized size of antiderivative = 1.49

$$\int \tan(c+dx)(a+b\tan(c+dx))^2 \left(B\tan(c+dx)+C\tan^2(c+dx)\right) dx$$
$$= \frac{C\tan(c+dx)(a+b\tan(c+dx))^3}{4bd}$$
$$+ \frac{\frac{(4bB-aC)(a+b\tan(c+dx))^3}{3bd} + \frac{2((bB-aC)(i(a+ib)^2\log(i-\tan(c+dx))-i(a-ib)^2\log(i+\tan(c+dx))-2b^2\tan(c+dx))-C((ia-b)^3\log(i-\text{ta}}{d}}{4b}$$

```
[In] Integrate[Tan[c + d*x]*(a + b*Tan[c + d*x])^2*(B*Tan[c + d*x] + C*Tan[c + d
*x]^2),x]
```

```
[Out] (C*Tan[c + d*x]*(a + b*Tan[c + d*x])^3)/(4*b*d) + (((4*b*B - a*C)*(a + b*Ta
n[c + d*x])^3)/(3*b*d) + (2*((b*B - a*C)*(I*(a + I*b)^2*Log[I - Tan[c + d*x
]] - I*(a - I*b)^2*Log[I + Tan[c + d*x]] - 2*b^2*Tan[c + d*x]) - C*((I*a -
b)^3*Log[I - Tan[c + d*x]] - (I*a + b)^3*Log[I + Tan[c + d*x]] + 6*a*b^2*Ta
n[c + d*x] + b^3*Tan[c + d*x]^2)))/d)/(4*b)
```

Maple [A] (verified)

Time = 0.10 (sec) , antiderivative size = 148, normalized size of antiderivative = 1.00

method	result
parts	$\frac{(B\,b^2+2Cab)\left(\frac{\tan(dx+c)^3}{3}-\tan(dx+c)+\arctan(\tan(dx+c))\right)}{d}+\frac{(2Bab+C\,a^2)\left(\frac{\tan(dx+c)^2}{2}-\frac{\ln\left(1+\tan(dx+c)^2\right)}{2}\right)}{d}+$
norman	$(-B\,a^2+B\,b^2+2Cab)\,x+\frac{(B\,a^2-B\,b^2-2Cab)\tan(dx+c)}{d}+\frac{(2Bab+C\,a^2-C\,b^2)\tan(dx+c)^2}{2d}+\frac{C\,b^2\,\text{ta}}{}$
derivativedivides	$\frac{C\,b^2\tan(dx+c)^4}{4}+\frac{B\,b^2\tan(dx+c)^3}{3}+\frac{2Cab\tan(dx+c)^3}{3}+Bab\tan(dx+c)^2+\frac{C\,a^2\tan(dx+c)^2}{2}-\frac{C\,b^2\tan(dx+c)^2}{2}+B\,a^2\tan(dx+$
default	$\frac{C\,b^2\tan(dx+c)^4}{4}+\frac{B\,b^2\tan(dx+c)^3}{3}+\frac{2Cab\tan(dx+c)^3}{3}+Bab\tan(dx+c)^2+\frac{C\,a^2\tan(dx+c)^2}{2}-\frac{C\,b^2\tan(dx+c)^2}{2}+B\,a^2\tan(dx+$
parallelrisch	$-\frac{-3C\,b^2\tan(dx+c)^4-4B\,b^2\tan(dx+c)^3-8Cab\tan(dx+c)^3+12B\,a^2dx-12B\,b^2dx-12Bab\tan(dx+c)^2-24Cabdx-6C}{}$
risch	$-B\,a^2x+B\,b^2x+2Cabx-iC\,a^2x+iC\,b^2x+\frac{2i\left(6iC\,b^2e^{4i(dx+c)}-6iBab\,e^{6i(dx+c)}+6iC\,b^2e^{6i(dx+c)}\right.}{}$

```
[In] int(tan(d*x+c)*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,method=_R
ETURNVERBOSE)
```

```
[Out] (B*b^2+2*C*a*b)/d*(1/3*tan(d*x+c)^3-tan(d*x+c)+arctan(tan(d*x+c)))+(2*B*a*b
+C*a^2)/d*(1/2*tan(d*x+c)^2-1/2*ln(1+tan(d*x+c)^2))+B*a^2/d*(tan(d*x+c)-arc
```

```
tan(tan(d*x+c))+C*b^2/d*(1/4*tan(d*x+c)^4-1/2*tan(d*x+c)^2+1/2*ln(1+tan(d*
x+c)^2))
```

Fricas [A] (verification not implemented)

none

Time = 0.27 (sec) , antiderivative size = 146, normalized size of antiderivative = 0.99

$$\int \tan(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{3\,Cb^2\tan\left(dx+c\right)^4+4\,(2\,Cab+Bb^2)\tan\left(dx+c\right)^3-12\,(Ba^2-2\,Cab-Bb^2)dx+6\,(Ca^2+2\,Bab-Cb^2)}{12\,d}$$

```
[In] integrate(tan(d*x+c)*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="fricas")
```

```
[Out] 1/12*(3*C*b^2*tan(d*x + c)^4 + 4*(2*C*a*b + B*b^2)*tan(d*x + c)^3 - 12*(B*a
^2 - 2*C*a*b - B*b^2)*d*x + 6*(C*a^2 + 2*B*a*b - C*b^2)*tan(d*x + c)^2 + 6*
(C*a^2 + 2*B*a*b - C*b^2)*log(1/(tan(d*x + c)^2 + 1)) + 12*(B*a^2 - 2*C*a*b
 - B*b^2)*tan(d*x + c))/d
```

Sympy [A] (verification not implemented)

Time = 0.15 (sec) , antiderivative size = 250, normalized size of antiderivative = 1.69

$$\int \tan(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\begin{cases}-Ba^2x+\frac{Ba^2\tan\left(c+dx\right)}{d}-\frac{Bab\log\left(\tan^2\left(c+dx\right)+1\right)}{d}+\frac{Bab\tan^2\left(c+dx\right)}{d}+Bb^2x+\frac{Bb^2\tan^3\left(c+dx\right)}{3d}-\frac{Bb^2\tan\left(c+dx\right)}{d}-\frac{Ca^2 l}{} \\ x(a+b\tan\left(c\right))^2\left(B\tan\left(c\right)+C\tan^2\left(c\right)\right)\tan\left(c\right)\end{cases}$$

```
[In] integrate(tan(d*x+c)*(a+b*tan(d*x+c))**2*(B*tan(d*x+c)+C*tan(d*x+c)**2),x)
```

```
[Out] Piecewise((-B*a**2*x + B*a**2*tan(c + d*x)/d - B*a*b*log(tan(c + d*x)**2 +
1)/d + B*a*b*tan(c + d*x)**2/d + B*b**2*x + B*b**2*tan(c + d*x)**3/(3*d) -
B*b**2*tan(c + d*x)/d - C*a**2*log(tan(c + d*x)**2 + 1)/(2*d) + C*a**2*tan(
c + d*x)**2/(2*d) + 2*C*a*b*x + 2*C*a*b*tan(c + d*x)**3/(3*d) - 2*C*a*b*tan
(c + d*x)/d + C*b**2*log(tan(c + d*x)**2 + 1)/(2*d) + C*b**2*tan(c + d*x)**
4/(4*d) - C*b**2*tan(c + d*x)**2/(2*d), Ne(d, 0)), (x*(a + b*tan(c))**2*(B*
tan(c) + C*tan(c)**2)*tan(c), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.32 (sec) , antiderivative size = 147, normalized size of antiderivative = 0.99

$$\int \tan(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{3\,Cb^2\tan\left(dx+c\right)^4+4\left(2\,Cab+Bb^2\right)\tan\left(dx+c\right)^3+6\left(Ca^2+2\,Bab-Cb^2\right)\tan\left(dx+c\right)^2-12\left(Ba^2-2\right.}{}$$

```
[In] integrate(tan(d*x+c)*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="maxima")
```

```
[Out] 1/12*(3*C*b^2*tan(d*x + c)^4 + 4*(2*C*a*b + B*b^2)*tan(d*x + c)^3 + 6*(C*a^
2 + 2*B*a*b - C*b^2)*tan(d*x + c)^2 - 12*(B*a^2 - 2*C*a*b - B*b^2)*(d*x + c
) - 6*(C*a^2 + 2*B*a*b - C*b^2)*log(tan(d*x + c)^2 + 1) + 12*(B*a^2 - 2*C*a
*b - B*b^2)*tan(d*x + c))/d
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 2078 vs. 2(141) = 282.

Time = 1.79 (sec) , antiderivative size = 2078, normalized size of antiderivative = 14.04

$$\int \tan(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx=\text{Too large to display}$$

```
[In] integrate(tan(d*x+c)*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="giac")
```

```
[Out] -1/12*(12*B*a^2*d*x*tan(d*x)^4*tan(c)^4 - 24*C*a*b*d*x*tan(d*x)^4*tan(c)^4
- 12*B*b^2*d*x*tan(d*x)^4*tan(c)^4 - 6*C*a^2*log(4*(tan(d*x)^2*tan(c)^2 - 2
*tan(d*x)*tan(c) + 1)/(tan(d*x)^2*tan(c)^2 + tan(d*x)^2 + tan(c)^2 + 1))*ta
n(d*x)^4*tan(c)^4 - 12*B*a*b*log(4*(tan(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(c)
 + 1)/(tan(d*x)^2*tan(c)^2 + tan(d*x)^2 + tan(c)^2 + 1))*tan(d*x)^4*tan(c)^
4 + 6*C*b^2*log(4*(tan(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(d*x)^2
*tan(c)^2 + tan(d*x)^2 + tan(c)^2 + 1))*tan(d*x)^4*tan(c)^4 - 48*B*a^2*d*x*
tan(d*x)^3*tan(c)^3 + 96*C*a*b*d*x*tan(d*x)^3*tan(c)^3 + 48*B*b^2*d*x*tan(d
*x)^3*tan(c)^3 - 6*C*a^2*tan(d*x)^4*tan(c)^4 - 12*B*a*b*tan(d*x)^4*tan(c)^4
 + 9*C*b^2*tan(d*x)^4*tan(c)^4 + 24*C*a^2*log(4*(tan(d*x)^2*tan(c)^2 - 2*ta
n(d*x)*tan(c) + 1)/(tan(d*x)^2*tan(c)^2 + tan(d*x)^2 + tan(c)^2 + 1))*tan(d
*x)^3*tan(c)^3 + 48*B*a*b*log(4*(tan(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(c) +
1)/(tan(d*x)^2*tan(c)^2 + tan(d*x)^2 + tan(c)^2 + 1))*tan(d*x)^3*tan(c)^3 -
 24*C*b^2*log(4*(tan(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(d*x)^2*t
an(c)^2 + tan(d*x)^2 + tan(c)^2 + 1))*tan(d*x)^3*tan(c)^3 + 12*B*a^2*tan(d*
```

```
x)^4*tan(c)^3 - 24*C*a*b*tan(d*x)^4*tan(c)^3 - 12*B*b^2*tan(d*x)^4*tan(c)^3
 + 12*B*a^2*tan(d*x)^3*tan(c)^4 - 24*C*a*b*tan(d*x)^3*tan(c)^4 - 12*B*b^2*t
an(d*x)^3*tan(c)^4 + 72*B*a^2*d*x*tan(d*x)^2*tan(c)^2 - 144*C*a*b*d*x*tan(d
*x)^2*tan(c)^2 - 72*B*b^2*d*x*tan(d*x)^2*tan(c)^2 - 6*C*a^2*tan(d*x)^4*tan(
c)^2 - 12*B*a*b*tan(d*x)^4*tan(c)^2 + 6*C*b^2*tan(d*x)^4*tan(c)^2 + 12*C*a^
2*tan(d*x)^3*tan(c)^3 + 24*B*a*b*tan(d*x)^3*tan(c)^3 - 24*C*b^2*tan(d*x)^3*
tan(c)^3 - 6*C*a^2*tan(d*x)^2*tan(c)^4 - 12*B*a*b*tan(d*x)^2*tan(c)^4 + 6*C
*b^2*tan(d*x)^2*tan(c)^4 + 8*C*a*b*tan(d*x)^4*tan(c) + 4*B*b^2*tan(d*x)^4*t
an(c) - 36*C*a^2*log(4*(tan(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(d
*x)^2*tan(c)^2 + tan(d*x)^2 + tan(c)^2 + 1))*tan(d*x)^2*tan(c)^2 - 72*B*a*b
*log(4*(tan(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(d*x)^2*tan(c)^2 +
 tan(d*x)^2 + tan(c)^2 + 1))*tan(d*x)^2*tan(c)^2 + 36*C*b^2*log(4*(tan(d*x)
^2*tan(c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(d*x)^2*tan(c)^2 + tan(d*x)^2 + ta
n(c)^2 + 1))*tan(d*x)^2*tan(c)^2 - 36*B*a^2*tan(d*x)^3*tan(c)^2 + 96*C*a*b*
tan(d*x)^3*tan(c)^2 + 48*B*b^2*tan(d*x)^3*tan(c)^2 - 36*B*a^2*tan(d*x)^2*ta
n(c)^3 + 96*C*a*b*tan(d*x)^2*tan(c)^3 + 48*B*b^2*tan(d*x)^2*tan(c)^3 + 8*C*
a*b*tan(d*x)*tan(c)^4 + 4*B*b^2*tan(d*x)*tan(c)^4 - 3*C*b^2*tan(d*x)^4 - 48
*B*a^2*d*x*tan(d*x)*tan(c) + 96*C*a*b*d*x*tan(d*x)*tan(c) + 48*B*b^2*d*x*ta
n(d*x)*tan(c) + 12*C*a^2*tan(d*x)^3*tan(c) + 24*B*a*b*tan(d*x)^3*tan(c) - 2
4*C*b^2*tan(d*x)^3*tan(c) - 12*C*a^2*tan(d*x)^2*tan(c)^2 - 24*B*a*b*tan(d*x
)^2*tan(c)^2 + 12*C*b^2*tan(d*x)^2*tan(c)^2 + 12*C*a^2*tan(d*x)*tan(c)^3 +
24*B*a*b*tan(d*x)*tan(c)^3 - 24*C*b^2*tan(d*x)*tan(c)^3 - 3*C*b^2*tan(c)^4
- 8*C*a*b*tan(d*x)^3 - 4*B*b^2*tan(d*x)^3 + 24*C*a^2*log(4*(tan(d*x)^2*tan(
c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(d*x)^2*tan(c)^2 + tan(d*x)^2 + tan(c)^2
+ 1))*tan(d*x)*tan(c) + 48*B*a*b*log(4*(tan(d*x)^2*tan(c)^2 - 2*tan(d*x)*ta
n(c) + 1)/(tan(d*x)^2*tan(c)^2 + tan(d*x)^2 + tan(c)^2 + 1))*tan(d*x)*tan(c
) - 24*C*b^2*log(4*(tan(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(d*x)^
2*tan(c)^2 + tan(d*x)^2 + tan(c)^2 + 1))*tan(d*x)*tan(c) + 36*B*a^2*tan(d*x
)^2*tan(c) - 96*C*a*b*tan(d*x)^2*tan(c) - 48*B*b^2*tan(d*x)^2*tan(c) + 36*B
*a^2*tan(d*x)*tan(c)^2 - 96*C*a*b*tan(d*x)*tan(c)^2 - 48*B*b^2*tan(d*x)*tan
(c)^2 - 8*C*a*b*tan(c)^3 - 4*B*b^2*tan(c)^3 + 12*B*a^2*d*x - 24*C*a*b*d*x -
 12*B*b^2*d*x - 6*C*a^2*tan(d*x)^2 - 12*B*a*b*tan(d*x)^2 + 6*C*b^2*tan(d*x)
^2 + 12*C*a^2*tan(d*x)*tan(c) + 24*B*a*b*tan(d*x)*tan(c) - 24*C*b^2*tan(d*x
)*tan(c) - 6*C*a^2*tan(c)^2 - 12*B*a*b*tan(c)^2 + 6*C*b^2*tan(c)^2 - 6*C*a^
2*log(4*(tan(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(d*x)^2*tan(c)^2
+ tan(d*x)^2 + tan(c)^2 + 1)) - 12*B*a*b*log(4*(tan(d*x)^2*tan(c)^2 - 2*tan
(d*x)*tan(c) + 1)/(tan(d*x)^2*tan(c)^2 + tan(d*x)^2 + tan(c)^2 + 1)) + 6*C*
b^2*log(4*(tan(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(d*x)^2*tan(c)^
2 + tan(d*x)^2 + tan(c)^2 + 1)) - 12*B*a^2*tan(d*x) + 24*C*a*b*tan(d*x) + 1
2*B*b^2*tan(d*x) - 12*B*a^2*tan(c) + 24*C*a*b*tan(c) + 12*B*b^2*tan(c) - 6*
C*a^2 - 12*B*a*b + 9*C*b^2)/(d*tan(d*x)^4*tan(c)^4 - 4*d*tan(d*x)^3*tan(c)^
3 + 6*d*tan(d*x)^2*tan(c)^2 - 4*d*tan(d*x)*tan(c) + d)
```

Mupad [B] (verification not implemented)

Time = 8.47 (sec) , antiderivative size = 151, normalized size of antiderivative = 1.02

$$\int \tan(c+dx)(a+b\tan(c+dx))^2 \left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$= x\left(-B\,a^2+2\,C\,a\,b+B\,b^2\right)+\frac{\tan(c+d\,x)^3\left(\frac{B\,b^2}{3}+\frac{2\,C\,a\,b}{3}\right)}{d}$$
$$-\frac{\tan(c+d\,x)\,\left(-B\,a^2+2\,C\,a\,b+B\,b^2\right)}{d}-\frac{\ln\left(\tan(c+d\,x)^2+1\right)\,\left(\frac{C\,a^2}{2}+B\,a\,b-\frac{C\,b^2}{2}\right)}{d}$$
$$+\frac{\tan(c+d\,x)^2\left(\frac{C\,a^2}{2}+B\,a\,b-\frac{C\,b^2}{2}\right)}{d}+\frac{C\,b^2\tan(c+d\,x)^4}{4\,d}$$

```
[In] int(tan(c + d*x)*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))^2
,x)
```

```
[Out] x*(B*b^2 - B*a^2 + 2*C*a*b) + (tan(c + d*x)^3*((B*b^2)/3 + (2*C*a*b)/3))/d
- (tan(c + d*x)*(B*b^2 - B*a^2 + 2*C*a*b))/d - (log(tan(c + d*x)^2 + 1)*((C
*a^2)/2 - (C*b^2)/2 + B*a*b))/d + (tan(c + d*x)^2*((C*a^2)/2 - (C*b^2)/2 +
B*a*b))/d + (C*b^2*tan(c + d*x)^4)/(4*d)
```

3.10 $\int (a+b\tan(c+dx))^2 \left(B\tan(c+dx)+C\tan^2(c+dx)\right) d$

Optimal result 126
Rubi [A] (verified) 126
Mathematica [C] (verified) 128
Maple [A] (verified) 128
Fricas [A] (verification not implemented) 129
Sympy [A] (verification not implemented) 129
Maxima [A] (verification not implemented) 130
Giac [B] (verification not implemented) 130
Mupad [B] (verification not implemented) 131

Optimal result

Integrand size = 32, antiderivative size = 112

$$\int (a+b\tan(c+dx))^2 \left(B\tan(c+dx)+C\tan^2(c+dx)\right) dx$$
$$= -\left(\left(2abB+a^2C-b^2C\right)x\right) - \frac{(a^2B-b^2B-2abC)\log(\cos(c+dx))}{d} + \frac{b(aB-bC)\tan(c+dx)}{d} + \frac{B(a+b\tan(c+dx))^2}{2d} + \frac{C(a+b\tan(c+dx))^3}{3bd}$$

```
[Out] -(2*B*a*b+C*a^2-C*b^2)*x-(B*a^2-B*b^2-2*C*a*b)*ln(cos(d*x+c))/d+b*(B*a-C*b)
*tan(d*x+c)/d+1/2*B*(a+b*tan(d*x+c))^2/d+1/3*C*(a+b*tan(d*x+c))^3/b/d
```

Rubi [A] (verified)

Time = 0.14 (sec) , antiderivative size = 112, normalized size of antiderivative = 1.00, number of steps used = 4, number of rules used = 4, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.125, Rules used = {3711, 3609, 3606, 3556}

$$\int (a+b\tan(c+dx))^2 \left(B\tan(c+dx)+C\tan^2(c+dx)\right) dx$$
$$= -\frac{(a^2B-2abC-b^2B)\log(\cos(c+dx))}{d} - x\left(a^2C+2abB-b^2C\right) + \frac{b(aB-bC)\tan(c+dx)}{d} + \frac{B(a+b\tan(c+dx))^2}{2d} + \frac{C(a+b\tan(c+dx))^3}{3bd}$$

```
[In] Int[(a + b*Tan[c + d*x])^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^2),x]
```

```
[Out] -((2*a*b*B + a^2*C - b^2*C)*x) - ((a^2*B - b^2*B - 2*a*b*C)*Log[Cos[c + d*x
]])/d + (b*(a*B - b*C)*Tan[c + d*x])/d + (B*(a + b*Tan[c + d*x])^2)/(2*d) +
 (C*(a + b*Tan[c + d*x])^3)/(3*b*d)
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3606

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
 f*x], x], x] + Simp[b*d*(Tan[e + f*x]/f), x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3609

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[d*((a + b*Tan[e + f*x])^m/(f*m)), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \frac{C(a+b\tan(c+dx))^3}{3bd}+\int (a+b\tan(c+dx))^2(-C+B\tan(c+dx))\,dx\\
&= \frac{B(a+b\tan(c+dx))^2}{2d}+\frac{C(a+b\tan(c+dx))^3}{3bd}\\
&\quad+\int (a+b\tan(c+dx))(-bB-aC+(aB-bC)\tan(c+dx))\,dx\\
&= -\left(\left(2abB+a^2C-b^2C\right)x\right)+\frac{b(aB-bC)\tan(c+dx)}{d}+\frac{B(a+b\tan(c+dx))^2}{2d}\\
&\quad+\frac{C(a+b\tan(c+dx))^3}{3bd}+\left(a^2B-b^2B-2abC\right)\int \tan(c+dx)\,dx\\
&= -\left(\left(2abB+a^2C-b^2C\right)x\right)-\frac{(a^2B-b^2B-2abC)\log(\cos(c+dx))}{d}\\
&\quad+\frac{b(aB-bC)\tan(c+dx)}{d}+\frac{B(a+b\tan(c+dx))^2}{2d}+\frac{C(a+b\tan(c+dx))^3}{3bd}
\end{aligned}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 1.96 (sec) , antiderivative size = 172, normalized size of antiderivative = 1.54

$$\int (a+b\tan(c+dx))^2 \left(B\tan(c+dx)+C\tan^2(c+dx)\right) dx$$
$$= \frac{2C(a+b\tan(c+dx))^3+3(aB+bC)\left(i((a+ib)^2\log(i-\tan(c+dx))-(a-ib)^2\log(i+\tan(c+dx)))-\right.}{}$$

```
[In] Integrate[(a + b*Tan[c + d*x])^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^2),x]
```

```
[Out] (2*C*(a + b*Tan[c + d*x])^3 + 3*(a*B + b*C)*(I*((a + I*b)^2*Log[I - Tan[c +
 d*x]] - (a - I*b)^2*Log[I + Tan[c + d*x]]) - 2*b^2*Tan[c + d*x]) + 3*B*((I
*a - b)^3*Log[I - Tan[c + d*x]] - (I*a + b)^3*Log[I + Tan[c + d*x]] + 6*a*b
^2*Tan[c + d*x] + b^3*Tan[c + d*x]^2))/(6*b*d)
```

Maple [A] (verified)

Time = 0.06 (sec) , antiderivative size = 120, normalized size of antiderivative = 1.07

method	result
norman	$\left(-2Bab-C\,a^2+C\,b^2\right)x+\frac{\left(2Bab+C\,a^2-C\,b^2\right)\tan(dx+c)}{d}+\frac{C\,b^2\tan(dx+c)^3}{3d}+\frac{b(Bb+2Ca)\tan(dx+c)^2}{2d}$
parts	$\frac{\left(B\,b^2+2Cab\right)\left(\frac{\tan(dx+c)^2}{2}-\frac{\ln\left(1+\tan(dx+c)^2\right)}{2}\right)}{d}+\frac{\left(2Bab+C\,a^2\right)\left(\tan(dx+c)-\arctan(\tan(dx+c))\right)}{d}+\frac{B\,a^2\ln\left(1+\tan(\right.}{2d}$
derivativedivides	$\frac{\frac{C\,b^2\tan(dx+c)^3}{3}+\frac{B\,b^2\tan(dx+c)^2}{2}+Cab\tan(dx+c)^2+2Bab\tan(dx+c)+C\,a^2\tan(dx+c)-C\,b^2\tan(dx+c)+\frac{\left(B\,a^2-B\,b^2-2C\right.}{}}{d}$
default	$\frac{\frac{C\,b^2\tan(dx+c)^3}{3}+\frac{B\,b^2\tan(dx+c)^2}{2}+Cab\tan(dx+c)^2+2Bab\tan(dx+c)+C\,a^2\tan(dx+c)-C\,b^2\tan(dx+c)+\frac{\left(B\,a^2-B\,b^2-2C\right.}{}}{d}$
parallelrisch	$\frac{2C\,b^2\tan(dx+c)^3-12Babdx+3B\,b^2\tan(dx+c)^2-6C\,a^2dx+6C\,b^2dx+6Cab\tan(dx+c)^2+3B\ln\left(1+\tan(dx+c)^2\right)a^2-3B\,\mathrm{l}}{6d}$
risch	$-iB\,b^2x+\frac{2iB\,a^2c}{d}+\frac{2i\left(-3iB\,b^2\mathrm{e}^{4i(dx+c)}-6iCab\,\mathrm{e}^{4i(dx+c)}+6Bab\,\mathrm{e}^{4i(dx+c)}+3C\,a^2\mathrm{e}^{4i(dx+c)}-6C\,b^2\mathrm{e}^{4i(dx+c)}-3i\right.}{3d(}$

```
[In] int((a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,method=_RETURNVERBOS
E)
```

```
[Out] (-2*B*a*b-C*a^2+C*b^2)*x+(2*B*a*b+C*a^2-C*b^2)/d*tan(d*x+c)+1/3*C*b^2/d*tan
(d*x+c)^3+1/2*b*(B*b+2*C*a)/d*tan(d*x+c)^2+1/2*(B*a^2-B*b^2-2*C*a*b)/d*ln(1
+tan(d*x+c)^2)
```

Fricas [A] (verification not implemented)

none

Time = 0.25 (sec) , antiderivative size = 119, normalized size of antiderivative = 1.06

$$\int (a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{2\,Cb^2\tan\left(dx+c\right)^3-6\left(Ca^2+2\,Bab-Cb^2\right)dx+3\left(2\,Cab+Bb^2\right)\tan\left(dx+c\right)^2-3\left(Ba^2-2\,Cab-Bb^2\right)}{6\,d}$$

```
[In] integrate((a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, algorithm="fr
icas")
```

```
[Out] 1/6*(2*C*b^2*tan(d*x + c)^3 - 6*(C*a^2 + 2*B*a*b - C*b^2)*d*x + 3*(2*C*a*b
+ B*b^2)*tan(d*x + c)^2 - 3*(B*a^2 - 2*C*a*b - B*b^2)*log(1/(tan(d*x + c)^2
 + 1)) + 6*(C*a^2 + 2*B*a*b - C*b^2)*tan(d*x + c))/d
```

Sympy [A] (verification not implemented)

Time = 0.13 (sec) , antiderivative size = 194, normalized size of antiderivative = 1.73

$$\int (a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\begin{cases}\frac{Ba^2\log\left(\tan^2\left(c+dx\right)+1\right)}{2d}-2Babx+\frac{2Bab\tan\left(c+dx\right)}{d}-\frac{Bb^2\log\left(\tan^2\left(c+dx\right)+1\right)}{2d}+\frac{Bb^2\tan^2\left(c+dx\right)}{2d}-Ca^2x+\frac{Ca^2\tan\left(c+dx\right)}{d} \\ x(a+b\tan\left(c\right))^2\left(B\tan\left(c\right)+C\tan^2\left(c\right)\right)\end{cases}$$

```
[In] integrate((a+b*tan(d*x+c))**2*(B*tan(d*x+c)+C*tan(d*x+c)**2),x)
```

```
[Out] Piecewise((B*a**2*log(tan(c + d*x)**2 + 1)/(2*d) - 2*B*a*b*x + 2*B*a*b*tan(
c + d*x)/d - B*b**2*log(tan(c + d*x)**2 + 1)/(2*d) + B*b**2*tan(c + d*x)**2
/(2*d) - C*a**2*x + C*a**2*tan(c + d*x)/d - C*a*b*log(tan(c + d*x)**2 + 1)/
d + C*a*b*tan(c + d*x)**2/d + C*b**2*x + C*b**2*tan(c + d*x)**3/(3*d) - C*b
**2*tan(c + d*x)/d, Ne(d, 0)), (x*(a + b*tan(c))**2*(B*tan(c) + C*tan(c)**2
), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.31 (sec) , antiderivative size = 120, normalized size of antiderivative = 1.07

$$\int (a+b\tan(c+dx))^2 \left(B\tan(c+dx)+C\tan^2(c+dx)\right) \, dx$$
$$= \frac{2\,Cb^2\tan\left(dx+c\right)^3 + 3\left(2\,Cab + Bb^2\right)\tan\left(dx+c\right)^2 - 6\left(Ca^2 + 2\,Bab - Cb^2\right)(dx+c) + 3\left(Ba^2 - 2\,Cab - \right.}{6\,d}$$

```
[In] integrate((a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, algorithm="ma
xima")
```

```
[Out] 1/6*(2*C*b^2*tan(d*x + c)^3 + 3*(2*C*a*b + B*b^2)*tan(d*x + c)^2 - 6*(C*a^2
 + 2*B*a*b - C*b^2)*(d*x + c) + 3*(B*a^2 - 2*C*a*b - B*b^2)*log(tan(d*x + c
)^2 + 1) + 6*(C*a^2 + 2*B*a*b - C*b^2)*tan(d*x + c))/d
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1389 vs. 2(108) = 216.

Time = 1.19 (sec) , antiderivative size = 1389, normalized size of antiderivative = 12.40

$$\int (a+b\tan(c+dx))^2 \left(B\tan(c+dx)+C\tan^2(c+dx)\right) \, dx = \text{Too large to display}$$

```
[In] integrate((a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, algorithm="gi
ac")
```

```
[Out] -1/6*(6*C*a^2*d*x*tan(d*x)^3*tan(c)^3 + 12*B*a*b*d*x*tan(d*x)^3*tan(c)^3 -
6*C*b^2*d*x*tan(d*x)^3*tan(c)^3 + 3*B*a^2*log(4*(tan(d*x)^2*tan(c)^2 - 2*ta
n(d*x)*tan(c) + 1)/(tan(d*x)^2*tan(c)^2 + tan(d*x)^2 + tan(c)^2 + 1))*tan(d
*x)^3*tan(c)^3 - 6*C*a*b*log(4*(tan(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(c) + 1
)/(tan(d*x)^2*tan(c)^2 + tan(d*x)^2 + tan(c)^2 + 1))*tan(d*x)^3*tan(c)^3 -
3*B*b^2*log(4*(tan(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(d*x)^2*tan
(c)^2 + tan(d*x)^2 + tan(c)^2 + 1))*tan(d*x)^3*tan(c)^3 - 18*C*a^2*d*x*tan(
d*x)^2*tan(c)^2 - 36*B*a*b*d*x*tan(d*x)^2*tan(c)^2 + 18*C*b^2*d*x*tan(d*x)^
2*tan(c)^2 - 6*C*a*b*tan(d*x)^3*tan(c)^3 - 3*B*b^2*tan(d*x)^3*tan(c)^3 - 9*
B*a^2*log(4*(tan(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(d*x)^2*tan(c
)^2 + tan(d*x)^2 + tan(c)^2 + 1))*tan(d*x)^2*tan(c)^2 + 18*C*a*b*log(4*(tan
(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(d*x)^2*tan(c)^2 + tan(d*x)^2
 + tan(c)^2 + 1))*tan(d*x)^2*tan(c)^2 + 9*B*b^2*log(4*(tan(d*x)^2*tan(c)^2
- 2*tan(d*x)*tan(c) + 1)/(tan(d*x)^2*tan(c)^2 + tan(d*x)^2 + tan(c)^2 + 1))
*tan(d*x)^2*tan(c)^2 + 6*C*a^2*tan(d*x)^3*tan(c)^2 + 12*B*a*b*tan(d*x)^3*ta
n(c)^2 - 6*C*b^2*tan(d*x)^3*tan(c)^2 + 6*C*a^2*tan(d*x)^2*tan(c)^3 + 12*B*a
```

```
*b*tan(d*x)^2*tan(c)^3 - 6*C*b^2*tan(d*x)^2*tan(c)^3 + 18*C*a^2*d*x*tan(d*x
)*tan(c) + 36*B*a*b*d*x*tan(d*x)*tan(c) - 18*C*b^2*d*x*tan(d*x)*tan(c) - 6*
C*a*b*tan(d*x)^3*tan(c) - 3*B*b^2*tan(d*x)^3*tan(c) + 6*C*a*b*tan(d*x)^2*ta
n(c)^2 + 3*B*b^2*tan(d*x)^2*tan(c)^2 - 6*C*a*b*tan(d*x)*tan(c)^3 - 3*B*b^2*
tan(d*x)*tan(c)^3 + 2*C*b^2*tan(d*x)^3 + 9*B*a^2*log(4*(tan(d*x)^2*tan(c)^2
 - 2*tan(d*x)*tan(c) + 1)/(tan(d*x)^2*tan(c)^2 + tan(d*x)^2 + tan(c)^2 + 1)
)*tan(d*x)*tan(c) - 18*C*a*b*log(4*(tan(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(c)
 + 1)/(tan(d*x)^2*tan(c)^2 + tan(d*x)^2 + tan(c)^2 + 1))*tan(d*x)*tan(c) -
9*B*b^2*log(4*(tan(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(d*x)^2*tan
(c)^2 + tan(d*x)^2 + tan(c)^2 + 1))*tan(d*x)*tan(c) - 12*C*a^2*tan(d*x)^2*t
an(c) - 24*B*a*b*tan(d*x)^2*tan(c) + 18*C*b^2*tan(d*x)^2*tan(c) - 12*C*a^2*
tan(d*x)*tan(c)^2 - 24*B*a*b*tan(d*x)*tan(c)^2 + 18*C*b^2*tan(d*x)*tan(c)^2
 + 2*C*b^2*tan(c)^3 - 6*C*a^2*d*x - 12*B*a*b*d*x + 6*C*b^2*d*x + 6*C*a*b*ta
n(d*x)^2 + 3*B*b^2*tan(d*x)^2 - 6*C*a*b*tan(d*x)*tan(c) - 3*B*b^2*tan(d*x)*
tan(c) + 6*C*a*b*tan(c)^2 + 3*B*b^2*tan(c)^2 - 3*B*a^2*log(4*(tan(d*x)^2*ta
n(c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(d*x)^2*tan(c)^2 + tan(d*x)^2 + tan(c)^
2 + 1)) + 6*C*a*b*log(4*(tan(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(
d*x)^2*tan(c)^2 + tan(d*x)^2 + tan(c)^2 + 1)) + 3*B*b^2*log(4*(tan(d*x)^2*t
an(c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(d*x)^2*tan(c)^2 + tan(d*x)^2 + tan(c)
^2 + 1)) + 6*C*a^2*tan(d*x) + 12*B*a*b*tan(d*x) - 6*C*b^2*tan(d*x) + 6*C*a^
2*tan(c) + 12*B*a*b*tan(c) - 6*C*b^2*tan(c) + 6*C*a*b + 3*B*b^2)/(d*tan(d*x
)^3*tan(c)^3 - 3*d*tan(d*x)^2*tan(c)^2 + 3*d*tan(d*x)*tan(c) - d)
```

Mupad [B] (verification not implemented)

Time = 8.43 (sec) , antiderivative size = 121, normalized size of antiderivative = 1.08

$$\int (a+b\tan(c+dx))^2 \left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{\tan(c+d\,x)^2\,\left(\frac{B\,b^2}{2}+C\,a\,b\right)}{d}-x\,\left(C\,a^2+2\,B\,a\,b-C\,b^2\right)+\frac{\tan(c+d\,x)\,\left(C\,a^2+2\,B\,a\,b-C\,b^2\right)}{d}-\frac{\ln\left(\tan(c+d\,x)^2+1\right)\,\left(-\frac{B\,a^2}{2}+C\,a\,b+\frac{B\,b^2}{2}\right)}{d}+\frac{C\,b^2\,\tan(c+d\,x)^3}{3\,d}$$

```
[In] int((B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))^2,x)
```

```
[Out] (tan(c + d*x)^2*((B*b^2)/2 + C*a*b))/d - x*(C*a^2 - C*b^2 + 2*B*a*b) + (tan
(c + d*x)*(C*a^2 - C*b^2 + 2*B*a*b))/d - (log(tan(c + d*x)^2 + 1)*((B*b^2)/
2 - (B*a^2)/2 + C*a*b))/d + (C*b^2*tan(c + d*x)^3)/(3*d)
```

3.11 $\int \cot(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2\right.$

Optimal result 132
Rubi [A] (verified) 132
Mathematica [C] (verified) 134
Maple [A] (verified) 134
Fricas [A] (verification not implemented) 135
Sympy [A] (verification not implemented) 135
Maxima [A] (verification not implemented) 135
Giac [A] (verification not implemented) 136
Mupad [B] (verification not implemented) 136

Optimal result

Integrand size = 38, antiderivative size = 87

$$\begin{aligned}&\int \cot(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx\\&=\left(a^2B-b^2B-2abC\right)x-\frac{(2abB+a^2C-b^2C)\log(\cos(c+dx))}{d}\\&\quad+\frac{b(bB+aC)\tan(c+dx)}{d}+\frac{C(a+b\tan(c+dx))^2}{2d}\end{aligned}$$

```
[Out] (B*a^2-B*b^2-2*C*a*b)*x-(2*B*a*b+C*a^2-C*b^2)*ln(cos(d*x+c))/d+b*(B*b+C*a)*
tan(d*x+c)/d+1/2*C*(a+b*tan(d*x+c))^2/d
```

Rubi [A] (verified)

Time = 0.16 (sec) , antiderivative size = 87, normalized size of antiderivative = 1.00, number of steps used = 4, number of rules used = 4, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.105, Rules used = {3713, 3609, 3606, 3556}

$$\begin{aligned}&\int \cot(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx\\&=-\frac{(a^2C+2abB-b^2C)\log(\cos(c+dx))}{d}+x\left(a^2B-2abC-b^2B\right)\\&\quad+\frac{b(aC+bB)\tan(c+dx)}{d}+\frac{C(a+b\tan(c+dx))^2}{2d}\end{aligned}$$

```
[In] Int[Cot[c + d*x]*(a + b*Tan[c + d*x])^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^2)
,x]
```

```
[Out] (a^2*B - b^2*B - 2*a*b*C)*x - ((2*a*b*B + a^2*C - b^2*C)*Log[Cos[c + d*x]])
/d + (b*(b*B + a*C)*Tan[c + d*x])/d + (C*(a + b*Tan[c + d*x])^2)/(2*d)
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3606

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
 f*x], x], x] + Simp[b*d*(Tan[e + f*x]/f), x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3609

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[d*((a + b*Tan[e + f*x])^m/(f*m)), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int (a+b\tan(c+dx))^2(B+C\tan(c+dx))\,dx \\
&= \frac{C(a+b\tan(c+dx))^2}{2d} + \int (a+b\tan(c+dx))(aB-bC+(bB+aC)\tan(c+dx))\,dx \\
&= \left(a^2B-b^2B-2abC\right)x + \frac{b(bB+aC)\tan(c+dx)}{d} \\
&\quad + \frac{C(a+b\tan(c+dx))^2}{2d} + \left(2abB+a^2C-b^2C\right)\int \tan(c+dx)\,dx \\
&= \left(a^2B-b^2B-2abC\right)x - \frac{\left(2abB+a^2C-b^2C\right)\log(\cos(c+dx))}{d} \\
&\quad + \frac{b(bB+aC)\tan(c+dx)}{d} + \frac{C(a+b\tan(c+dx))^2}{2d}
\end{aligned}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 0.52 (sec) , antiderivative size = 96, normalized size of antiderivative = 1.10

$$\int \cot(c+dx)(a+b\tan(c+dx))^2 \left(B\tan(c+dx)+C\tan^2(c+dx)\right) dx$$
$$= \frac{(a+ib)^2(-iB+C)\log(i-\tan(c+dx))+(a-ib)^2(iB+C)\log(i+\tan(c+dx))+2b(bB+2aC)\tan(c-}{2d}$$

```
[In] Integrate[Cot[c + d*x]*(a + b*Tan[c + d*x])^2*(B*Tan[c + d*x] + C*Tan[c + d
*x]^2),x]
```

```
[Out] ((a + I*b)^2*((-I)*B + C)*Log[I - Tan[c + d*x]] + (a - I*b)^2*(I*B + C)*Log
[I + Tan[c + d*x]] + 2*b*(b*B + 2*a*C)*Tan[c + d*x] + b^2*C*Tan[c + d*x]^2)
/(2*d)
```

Maple [A] (verified)

Time = 0.25 (sec) , antiderivative size = 87, normalized size of antiderivative = 1.00

method	result
parallelrisch	$\frac{(2Bab+Ca^2-Cb^2)\ln\left(\sec(dx+c)^2\right)+Cb^2\tan(dx+c)^2+(2Bb^2+4Cab)\tan(dx+c)+2dx(Ba^2-Bb^2-2Cab)}{2d}$
norman	$(Ba^2-Bb^2-2Cab)x+\frac{b(Bb+2Ca)\tan(dx+c)}{d}+\frac{Cb^2\tan(dx+c)^2}{2d}+\frac{(2Bab+Ca^2-Cb^2)\ln\left(1+\tan(dx+\right.}{2d}$
derivativedivides	$-\frac{\frac{\left(-2Bab-Ca^2+Cb^2\right)\ln\left(\cot(dx+c)^2+1\right)}{2}+(Ba^2-Bb^2-2Cab)\left(\frac{\pi}{2}-\operatorname{arccot}(\cot(dx+c))\right)+(2Bab+Ca^2-Cb^2)\ln(\cot(dx+c}{d}$
default	$-\frac{\frac{\left(-2Bab-Ca^2+Cb^2\right)\ln\left(\cot(dx+c)^2+1\right)}{2}+(Ba^2-Bb^2-2Cab)\left(\frac{\pi}{2}-\operatorname{arccot}(\cot(dx+c))\right)+(2Bab+Ca^2-Cb^2)\ln(\cot(dx+c}{d}$
risch	$Ba^2x-Bb^2x-2Cabx-\frac{2iCb^2c}{d}-iCb^2x+\frac{2iCa^2c}{d}+iCa^2x+\frac{4iBabc}{d}+2iBabx+\frac{2ib(-iC}{}$

```
[In] int(cot(d*x+c)*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,method=_R
ETURNVERBOSE)
```

```
[Out] 1/2*((2*B*a*b+C*a^2-C*b^2)*ln(sec(d*x+c)^2)+C*b^2*tan(d*x+c)^2+(2*B*b^2+4*C
*a*b)*tan(d*x+c)+2*d*x*(B*a^2-B*b^2-2*C*a*b))/d
```

Fricas [A] (verification not implemented)

none

Time = 0.25 (sec) , antiderivative size = 91, normalized size of antiderivative = 1.05

$$\int \cot(c+dx)(a+b\tan(c+dx))^2 \left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{Cb^2\tan(dx+c)^2+2(Ba^2-2Cab-Bb^2)dx-(Ca^2+2Bab-Cb^2)\log\left(\frac{1}{\tan(dx+c)^2+1}\right)+2(2Cab+Bb$$

```
[In] integrate(cot(d*x+c)*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="fricas")
```

```
[Out] 1/2*(C*b^2*tan(d*x + c)^2 + 2*(B*a^2 - 2*C*a*b - B*b^2)*d*x - (C*a^2 + 2*B*
a*b - C*b^2)*log(1/(tan(d*x + c)^2 + 1)) + 2*(2*C*a*b + B*b^2)*tan(d*x + c)
)/d
```

Sympy [A] (verification not implemented)

Time = 0.46 (sec) , antiderivative size = 151, normalized size of antiderivative = 1.74

$$\int \cot(c+dx)(a+b\tan(c+dx))^2 \left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\begin{cases} Ba^2x+\frac{Bab\log(\tan^2(c+dx)+1)}{d}-Bb^2x+\frac{Bb^2\tan(c+dx)}{d}+\frac{Ca^2\log(\tan^2(c+dx)+1)}{2d}-2Cabx+\frac{2Cab\tan(c+dx)}{d}-\frac{Cb^2}{} \\ x(a+b\tan(c))^2\left(B\tan(c)+C\tan^2(c)\right)\cot(c) \end{cases}$$

```
[In] integrate(cot(d*x+c)*(a+b*tan(d*x+c))**2*(B*tan(d*x+c)+C*tan(d*x+c)**2),x)
```

```
[Out] Piecewise((B*a**2*x + B*a*b*log(tan(c + d*x)**2 + 1)/d - B*b**2*x + B*b**2*
tan(c + d*x)/d + C*a**2*log(tan(c + d*x)**2 + 1)/(2*d) - 2*C*a*b*x + 2*C*a*
b*tan(c + d*x)/d - C*b**2*log(tan(c + d*x)**2 + 1)/(2*d) + C*b**2*tan(c + d
*x)**2/(2*d), Ne(d, 0)), (x*(a + b*tan(c))**2*(B*tan(c) + C*tan(c)**2)*cot(
c), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.31 (sec) , antiderivative size = 91, normalized size of antiderivative = 1.05

$$\int \cot(c+dx)(a+b\tan(c+dx))^2 \left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{Cb^2\tan(dx+c)^2+2(Ba^2-2Cab-Bb^2)(dx+c)+(Ca^2+2Bab-Cb^2)\log\left(\tan(dx+c)^2+1\right)+2(2}{2d}$$

```
[In] integrate(cot(d*x+c)*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="maxima")
```

```
[Out] 1/2*(C*b^2*tan(d*x + c)^2 + 2*(B*a^2 - 2*C*a*b - B*b^2)*(d*x + c) + (C*a^2
+ 2*B*a*b - C*b^2)*log(tan(d*x + c)^2 + 1) + 2*(2*C*a*b + B*b^2)*tan(d*x +
c))/d
```

Giac [A] (verification not implemented)

none

Time = 1.01 (sec) , antiderivative size = 95, normalized size of antiderivative = 1.09

$$\int \cot(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$= \frac{Cb^2\tan(dx+c)^2+4\,Cab\tan(dx+c)+2\,Bb^2\tan(dx+c)+2\,(Ba^2-2\,Cab-Bb^2)(dx+c)+(Ca^2+2\,}{2\,d}$$

```
[In] integrate(cot(d*x+c)*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="giac")
```

```
[Out] 1/2*(C*b^2*tan(d*x + c)^2 + 4*C*a*b*tan(d*x + c) + 2*B*b^2*tan(d*x + c) + 2
*(B*a^2 - 2*C*a*b - B*b^2)*(d*x + c) + (C*a^2 + 2*B*a*b - C*b^2)*log(tan(d*
x + c)^2 + 1))/d
```

Mupad [B] (verification not implemented)

Time = 8.30 (sec) , antiderivative size = 91, normalized size of antiderivative = 1.05

$$\int \cot(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$= \frac{\ln\left(\tan(c+d\,x)^2+1\right)\,\left(\frac{C\,a^2}{2}+B\,a\,b-\frac{C\,b^2}{2}\right)}{d}-x\,\left(-B\,a^2+2\,C\,a\,b+B\,b^2\right)$$
$$+\frac{\tan(c+d\,x)\,\left(B\,b^2+2\,C\,a\,b\right)}{d}+\frac{C\,b^2\tan(c+d\,x)^2}{2\,d}$$

```
[In] int(cot(c + d*x)*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))^2
,x)
```

```
[Out] (log(tan(c + d*x)^2 + 1)*((C*a^2)/2 - (C*b^2)/2 + B*a*b))/d - x*(B*b^2 - B*
a^2 + 2*C*a*b) + (tan(c + d*x)*(B*b^2 + 2*C*a*b))/d + (C*b^2*tan(c + d*x)^2
)/(2*d)
```

3.12 $\int \cot^2(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan\right.$

Optimal result 137
Rubi [A] (verified) 137
Mathematica [C] (verified) 139
Maple [A] (verified) 139
Fricas [A] (verification not implemented) 140
Sympy [B] (verification not implemented) 140
Maxima [A] (verification not implemented) 141
Giac [A] (verification not implemented) 141
Mupad [B] (verification not implemented) 141

Optimal result

Integrand size = 40, antiderivative size = 70

$$\begin{aligned}&\int \cot^2(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx\\&=\left(2abB+a^2C-b^2C\right)x-\frac{b(bB+2aC)\log(\cos(c+dx))}{d}\\&\quad+\frac{a^2B\log(\sin(c+dx))}{d}+\frac{b^2C\tan(c+dx)}{d}\end{aligned}$$

```
[Out] (2*B*a*b+C*a^2-C*b^2)*x-b*(B*b+2*C*a)*ln(cos(d*x+c))/d+a^2*B*ln(sin(d*x+c))
/d+b^2*C*tan(d*x+c)/d
```

Rubi [A] (verified)

Time = 0.22 (sec) , antiderivative size = 70, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 4, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.100, Rules used = {3713, 3687, 3705, 3556}

$$\begin{aligned}&\int \cot^2(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx\\&=x\left(a^2C+2abB-b^2C\right)+\frac{a^2B\log(\sin(c+dx))}{d}\\&\quad-\frac{b(2aC+bB)\log(\cos(c+dx))}{d}+\frac{b^2C\tan(c+dx)}{d}\end{aligned}$$

```
[In] Int[Cot[c + d*x]^2*(a + b*Tan[c + d*x])^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^
2),x]
```

```
[Out] (2*a*b*B + a^2*C - b^2*C)*x - (b*(b*B + 2*a*C)*Log[Cos[c + d*x]])/d + (a^2*
B*Log[Sin[c + d*x]])/d + (b^2*C*Tan[c + d*x])/d
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3687

```
Int[(((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^2*((A_.) + (B_.)*tan[(e_.) + (
f_.)*(x_)]))/((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[b^2
*B*(Tan[e + f*x]/(d*f)), x] + Dist[1/d, Int[(a^2*A*d - b^2*B*c + (2*a*A*b +
 B*(a^2 - b^2))*d*Tan[e + f*x] + (A*b^2*d - b*B*(b*c - 2*a*d))*Tan[e + f*x]
^2)/(c + d*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] && N
eQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0]
```

Rule 3705

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/tan[(e_.) + (f_.)*(x_)], x_Symbol] :> Simp[B*x, x] + (Dist[A, Int[1/Tan[e
 + f*x], x], x] + Dist[C, Int[Tan[e + f*x], x], x]) /; FreeQ[{e, f, A, B, C
}, x] && NeQ[A, C]
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \cot(c+dx)(a+b\tan(c+dx))^2(B+C\tan(c+dx))\,dx \\
&= \frac{b^2C\tan(c+dx)}{d} + \int \cot(c+dx)\left(a^2B + \left(2abB + \left(a^2-b^2\right)C\right)\tan(c+dx)\right. \\
&\qquad \left. + \left(b^2B + 2abC\right)\tan^2(c+dx)\right)\,dx \\
&= \left(2abB + a^2C - b^2C\right)x + \frac{b^2C\tan(c+dx)}{d} \\
&\quad + \left(a^2B\right)\int \cot(c+dx)\,dx + (b(bB+2aC))\int \tan(c+dx)\,dx \\
&= \left(2abB + a^2C - b^2C\right)x - \frac{b(bB+2aC)\log(\cos(c+dx))}{d} \\
&\quad + \frac{a^2B\log(\sin(c+dx))}{d} + \frac{b^2C\tan(c+dx)}{d}
\end{aligned}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 0.31 (sec) , antiderivative size = 91, normalized size of antiderivative = 1.30

$$\int \cot^2(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx = -\frac{(a+ib)^2(B+iC)\log(i-\tan(c+dx))-2a^2B\log(\tan(c+dx))+(a-ib)^2(B-iC)\log(i+\tan(c+d}{2d}$$

```
[In] Integrate[Cot[c + d*x]^2*(a + b*Tan[c + d*x])^2*(B*Tan[c + d*x] + C*Tan[c +
 d*x]^2),x]
```

```
[Out] -1/2*((a + I*b)^2*(B + I*C)*Log[I - Tan[c + d*x]] - 2*a^2*B*Log[Tan[c + d*x
]] + (a - I*b)^2*(B - I*C)*Log[I + Tan[c + d*x]] - 2*b^2*C*Tan[c + d*x])/d
```

Maple [A] (verified)

Time = 0.22 (sec) , antiderivative size = 80, normalized size of antiderivative = 1.14

method	result
parallelrisch	$\frac{\left(-B\,a^2+B\,b^2+2Cab\right)\ln\left(\sec(dx+c)^2\right)+2B\,a^2\ln(\tan(dx+c))+2C\,b^2\tan(dx+c)+4dx\left(Bab+\frac{1}{2}C\,a^2-\frac{1}{2}C\,b^2\right)}{2d}$
derivativedivides	$-\frac{\frac{\left(B\,a^2-B\,b^2-2Cab\right)\ln\left(\cot(dx+c)^2+1\right)}{2}+\left(2Bab+C\,a^2-C\,b^2\right)\left(\frac{\pi}{2}-\operatorname{arccot}(\cot(dx+c))\right)-\frac{C\,b^2}{\cot(dx+c)}+b(Bb+2Ca)\ln(\cot(d}{d}$
default	$-\frac{\frac{\left(B\,a^2-B\,b^2-2Cab\right)\ln\left(\cot(dx+c)^2+1\right)}{2}+\left(2Bab+C\,a^2-C\,b^2\right)\left(\frac{\pi}{2}-\operatorname{arccot}(\cot(dx+c))\right)-\frac{C\,b^2}{\cot(dx+c)}+b(Bb+2Ca)\ln(\cot(d}{d}$
norman	$\frac{\left(2Bab+C\,a^2-C\,b^2\right)x\tan(dx+c)+\frac{C\,b^2\tan(dx+c)^2}{d}}{\tan(dx+c)}+\frac{B\,a^2\ln(\tan(dx+c))}{d}-\frac{\left(B\,a^2-B\,b^2-2Cab\right)\ln\left(1+\tan(dx+c)^2\right)}{2d}$
risch	$iB\,b^2x+\frac{4iCabc}{d}+\frac{2iC\,b^2}{d\left(e^{2i(dx+c)}+1\right)}+2Babx+C\,a^2x-C\,b^2x+2iCabx-\frac{2iB\,a^2c}{d}+\frac{2iB\,b^2c}{d}-$

```
[In] int(cot(d*x+c)^2*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,method=
_RETURNVERBOSE)
```

```
[Out] 1/2*((-B*a^2+B*b^2+2*C*a*b)*ln(sec(d*x+c)^2)+2*B*a^2*ln(tan(d*x+c))+2*C*b^2
*tan(d*x+c)+4*d*x*(B*a*b+1/2*C*a^2-1/2*C*b^2))/d
```

Fricas [A] (verification not implemented)

none

Time = 0.28 (sec) , antiderivative size = 92, normalized size of antiderivative = 1.31

$$\int \cot^2(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{Ba^2\log\left(\frac{\tan(dx+c)^2}{\tan(dx+c)^2+1}\right)+2\,Cb^2\tan(dx+c)+2\,(Ca^2+2\,Bab-Cb^2)dx-(2\,Cab+Bb^2)\log\left(\frac{1}{\tan(dx+c)^2+1}\right)}{2\,d}$$

```
[In] integrate(cot(d*x+c)^2*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="fricas")
```

```
[Out] 1/2*(B*a^2*log(tan(d*x + c)^2/(tan(d*x + c)^2 + 1)) + 2*C*b^2*tan(d*x + c)
+ 2*(C*a^2 + 2*B*a*b - C*b^2)*d*x - (2*C*a*b + B*b^2)*log(1/(tan(d*x + c)^2
 + 1)))/d
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 136 vs. 2(66) = 132.

Time = 0.75 (sec) , antiderivative size = 136, normalized size of antiderivative = 1.94

$$\int \cot^2(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\begin{cases}-\frac{Ba^2\log\left(\tan^2(c+dx)+1\right)}{2d}+\frac{Ba^2\log\left(\tan(c+dx)\right)}{d}+2Babx+\frac{Bb^2\log\left(\tan^2(c+dx)+1\right)}{2d}+Ca^2x+\frac{Cab\log\left(\tan^2(c+dx)+1\right)}{d}-C \\ x(a+b\tan(c))^2\left(B\tan(c)+C\tan^2(c)\right)\cot^2(c)\end{cases}$$

```
[In] integrate(cot(d*x+c)**2*(a+b*tan(d*x+c))**2*(B*tan(d*x+c)+C*tan(d*x+c)**2),
x)
```

```
[Out] Piecewise((-B*a**2*log(tan(c + d*x)**2 + 1)/(2*d) + B*a**2*log(tan(c + d*x)
)/d + 2*B*a*b*x + B*b**2*log(tan(c + d*x)**2 + 1)/(2*d) + C*a**2*x + C*a*b*
log(tan(c + d*x)**2 + 1)/d - C*b**2*x + C*b**2*tan(c + d*x)/d, Ne(d, 0)), (
x*(a + b*tan(c))**2*(B*tan(c) + C*tan(c)**2)*cot(c)**2, True))
```

Maxima [A] (verification not implemented)

none

Time = 0.44 (sec) , antiderivative size = 85, normalized size of antiderivative = 1.21

$$\int \cot^2(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{2\,Ba^2\log\left(\tan\left(dx+c\right)\right)+2\,Cb^2\tan\left(dx+c\right)+2\,(Ca^2+2\,Bab-Cb^2)(dx+c)-(Ba^2-2\,Cab-Bb^2)\log}{2\,d}$$

```
[In] integrate(cot(d*x+c)^2*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="maxima")
```

```
[Out] 1/2*(2*B*a^2*log(tan(d*x + c)) + 2*C*b^2*tan(d*x + c) + 2*(C*a^2 + 2*B*a*b
- C*b^2)*(d*x + c) - (B*a^2 - 2*C*a*b - B*b^2)*log(tan(d*x + c)^2 + 1))/d
```

Giac [A] (verification not implemented)

none

Time = 1.34 (sec) , antiderivative size = 86, normalized size of antiderivative = 1.23

$$\int \cot^2(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{2\,Ba^2\log\left(\left|\tan\left(dx+c\right)\right|\right)+2\,Cb^2\tan\left(dx+c\right)+2\,(Ca^2+2\,Bab-Cb^2)(dx+c)-(Ba^2-2\,Cab-Bb^2)\,l}{2\,d}$$

```
[In] integrate(cot(d*x+c)^2*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="giac")
```

```
[Out] 1/2*(2*B*a^2*log(abs(tan(d*x + c))) + 2*C*b^2*tan(d*x + c) + 2*(C*a^2 + 2*B
*a*b - C*b^2)*(d*x + c) - (B*a^2 - 2*C*a*b - B*b^2)*log(tan(d*x + c)^2 + 1)
)/d
```

Mupad [B] (verification not implemented)

Time = 8.59 (sec) , antiderivative size = 90, normalized size of antiderivative = 1.29

$$\int \cot^2(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{B\,a^2\,\ln\left(\tan(c+d\,x)\right)}{d}+\frac{\ln\left(\tan(c+d\,x)+1\mathrm{i}\right)\,\left(B-C\,1\mathrm{i}\right)\,\left(b+a\,1\mathrm{i}\right)^2}{2\,d}$$
$$+\frac{C\,b^2\tan(c+d\,x)}{d}+\frac{\ln\left(\tan(c+d\,x)-\mathrm{i}\right)\,\left(B+C\,1\mathrm{i}\right)\,\left(-b+a\,1\mathrm{i}\right)^2}{2\,d}$$

```
[In] int(cot(c + d*x)^2*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))
^2,x)
```

```
[Out] (B*a^2*log(tan(c + d*x)))/d + (log(tan(c + d*x) + 1i)*(B - C*1i)*(a*1i + b)
^2)/(2*d) + (C*b^2*tan(c + d*x))/d + (log(tan(c + d*x) - 1i)*(B + C*1i)*(a*
1i - b)^2)/(2*d)
```

3.13 $\int \cot^3(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan$

Optimal result 143
Rubi [A] (verified) 143
Mathematica [C] (verified) 145
Maple [A] (verified) 145
Fricas [A] (verification not implemented) 146
Sympy [B] (verification not implemented) 146
Maxima [A] (verification not implemented) 147
Giac [A] (verification not implemented) 147
Mupad [B] (verification not implemented) 147

Optimal result

Integrand size = 40, antiderivative size = 72

$$\begin{aligned}&\int \cot^3(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx\\&=-\left(\left(a^2B-b^2B-2abC\right)x\right)-\frac{a^2B\cot(c+dx)}{d}\\&\quad-\frac{b^2C\log(\cos(c+dx))}{d}+\frac{a(2bB+aC)\log(\sin(c+dx))}{d}\end{aligned}$$

```
[Out] -(B*a^2-B*b^2-2*C*a*b)*x-a^2*B*cot(d*x+c)/d-b^2*C*ln(cos(d*x+c))/d+a*(2*B*b
+C*a)*ln(sin(d*x+c))/d
```

Rubi [A] (verified)

Time = 0.24 (sec) , antiderivative size = 72, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 4, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.100, Rules used = {3713, 3685, 3705, 3556}

$$\begin{aligned}&\int \cot^3(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx\\&=-x\left(a^2B-2abC-b^2B\right)-\frac{a^2B\cot(c+dx)}{d}\\&\quad+\frac{a(aC+2bB)\log(\sin(c+dx))}{d}-\frac{b^2C\log(\cos(c+dx))}{d}\end{aligned}$$

```
[In] Int[Cot[c + d*x]^3*(a + b*Tan[c + d*x])^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^
2),x]
```

```
[Out] -((a^2*B - b^2*B - 2*a*b*C)*x) - (a^2*B*Cot[c + d*x])/d - (b^2*C*Log[Cos[c
+ d*x]])/d + (a*(2*b*B + a*C)*Log[Sin[c + d*x]])/d
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3685

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^2*((A_.) + (B_.)*tan[(e_.) + (f
_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Simp[
(-(B*c - A*d))*(b*c - a*d)^2*((c + d*Tan[e + f*x])^(n + 1)/(f*d^2*(n + 1)*(
c^2 + d^2))), x] + Dist[1/(d*(c^2 + d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)
*Simp[B*(b*c - a*d)^2 + A*d*(a^2*c - b^2*c + 2*a*b*d) + d*(B*(a^2*c - b^2*c
 + 2*a*b*d) + A*(2*a*b*c - a^2*d + b^2*d))*Tan[e + f*x] + b^2*B*(c^2 + d^2)
*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] && NeQ[b*
c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[n, -1]
```

Rule 3705

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/tan[(e_.) + (f_.)*(x_)], x_Symbol] :> Simp[B*x, x] + (Dist[A, Int[1/Tan[e
 + f*x], x], x] + Dist[C, Int[Tan[e + f*x], x], x]) /; FreeQ[{e, f, A, B, C
}, x] && NeQ[A, C]
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \cot^2(c+dx)(a+b\tan(c+dx))^2(B+C\tan(c+dx))\,dx \\
&= -\frac{a^2B\cot(c+dx)}{d}+\int \cot(c+dx)\left(a(2bB+aC)\right. \\
&\qquad \left.-\left(a^2B-b^2B-2abC\right)\tan(c+dx)+b^2C\tan^2(c+dx)\right)\,dx \\
&= -\left(\left(a^2B-b^2B-2abC\right)x\right)-\frac{a^2B\cot(c+dx)}{d} \\
&\quad +\left(b^2C\right)\int \tan(c+dx)\,dx+(a(2bB+aC))\int \cot(c+dx)\,dx
\end{aligned}$$

$$= -\left(\left(a^2B - b^2B - 2abC\right)x\right) - \frac{a^2B\cot(c+dx)}{d} - \frac{b^2C\log(\cos(c+dx))}{d} + \frac{a(2bB+aC)\log(\sin(c+dx))}{d}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 0.30 (sec) , antiderivative size = 100, normalized size of antiderivative = 1.39

$$\int \cot^3(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$= \frac{-2a^2B\cot(c+dx)+i(a+ib)^2(B+iC)\log(i-\tan(c+dx))+2a(2bB+aC)\log(\tan(c+dx))-(a-i}{2d}$$

```
[In] Integrate[Cot[c + d*x]^3*(a + b*Tan[c + d*x])^2*(B*Tan[c + d*x] + C*Tan[c +
 d*x]^2),x]
```

```
[Out] (-2*a^2*B*Cot[c + d*x] + I*(a + I*b)^2*(B + I*C)*Log[I - Tan[c + d*x]] + 2*
a*(2*b*B + a*C)*Log[Tan[c + d*x]] - (a - I*b)^2*(I*B + C)*Log[I + Tan[c + d
*x]])/(2*d)
```

Maple [A] (verified)

Time = 0.35 (sec) , antiderivative size = 84, normalized size of antiderivative = 1.17

method	result
derivativedivides	$\frac{B\,b^2(dx+c)-C\,b^2\ln(\cos(dx+c))+2Bab\ln(\sin(dx+c))+2Cab(dx+c)+B\,a^2(-\cot(dx+c)-dx-c)+C\,a^2\ln(\sin(dx+c))}{d}$
default	$\frac{B\,b^2(dx+c)-C\,b^2\ln(\cos(dx+c))+2Bab\ln(\sin(dx+c))+2Cab(dx+c)+B\,a^2(-\cot(dx+c)-dx-c)+C\,a^2\ln(\sin(dx+c))}{d}$
parallelrisch	$\frac{(-2Bab-C\,a^2+C\,b^2)\ln\left(\sec(dx+c)^2\right)+(4Bab+2C\,a^2)\ln(\tan(dx+c))-2B\,a^2\cot(dx+c)-2dx\left(B\,a^2-B\,b^2-2Cab\right)}{2d}$
norman	$\frac{(-B\,a^2+B\,b^2+2Cab)x\tan(dx+c)^2-\frac{B\,a^2\tan(dx+c)}{d}}{\tan(dx+c)^2}+\frac{a(2Bb+Ca)\ln(\tan(dx+c))}{d}-\frac{(2Bab+C\,a^2-C\,b^2)\ln\left(1+\tan\right.}{2d}$
risch	$-B\,a^2x+B\,b^2x+2Cabx-\frac{2iC\,a^2c}{d}+iC\,b^2x-iC\,a^2x-\frac{2iB\,a^2}{d\left(e^{2i(dx+c)}-1\right)}-2iBabx-\frac{4iBab}{d}$

```
[In] int(cot(d*x+c)^3*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,method=
_RETURNVERBOSE)
```

```
[Out] 1/d*(B*b^2*(d*x+c)-C*b^2*ln(cos(d*x+c))+2*B*a*b*ln(sin(d*x+c))+2*C*a*b*(d*x
+c)+B*a^2*(-cot(d*x+c)-d*x-c)+C*a^2*ln(sin(d*x+c)))
```

Fricas [A] (verification not implemented)

none

Time = 0.25 (sec) , antiderivative size = 112, normalized size of antiderivative = 1.56

$$\int \cot^3(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx = -\frac{Cb^2\log\left(\frac{1}{\tan(dx+c)^2+1}\right)\tan\left(dx+c\right)+2\left(Ba^2-2\,Cab-Bb^2\right)dx\tan\left(dx+c\right)+2\,Ba^2-\left(Ca^2+2\,Bab\right)\log\left(\frac{\tan(dx+c)^2}{\tan(dx+c)^2+1}\right)\tan\left(dx+c\right)}{2\,d\tan\left(dx+c\right)}$$

```
[In] integrate(cot(d*x+c)^3*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="fricas")
```

```
[Out] -1/2*(C*b^2*log(1/(tan(d*x + c)^2 + 1))*tan(d*x + c) + 2*(B*a^2 - 2*C*a*b -
 B*b^2)*d*x*tan(d*x + c) + 2*B*a^2 - (C*a^2 + 2*B*a*b)*log(tan(d*x + c)^2/(
tan(d*x + c)^2 + 1))*tan(d*x + c))/(d*tan(d*x + c))
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 158 vs. 2(66) = 132.

Time = 1.01 (sec) , antiderivative size = 158, normalized size of antiderivative = 2.19

$$\int \cot^3(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx = \begin{cases} \text{NaN} \\ x(a+b\tan(c))^2\left(B\tan(c)+C\tan^2(c)\right)\cot^3(c) \\ \text{NaN} \\ -Ba^2x-\frac{Ba^2}{d\tan(c+dx)}-\frac{Bab\log(\tan^2(c+dx)+1)}{d}+\frac{2Bab\log(\tan(c+dx))}{d}+Bb^2x-\frac{Ca^2\log(\tan^2(c+dx)+1)}{2d}+\frac{Ca^2\log(\tan(c+dx))}{d}+2Cabx+\frac{Cb^2\log(\tan^2(c+dx)+1)}{2d} \end{cases}$$

```
[In] integrate(cot(d*x+c)**3*(a+b*tan(d*x+c))**2*(B*tan(d*x+c)+C*tan(d*x+c)**2),
x)
```

```
[Out] Piecewise((nan, Eq(c, 0) & Eq(d, 0)), (x*(a + b*tan(c))**2*(B*tan(c) + C*ta
n(c)**2)*cot(c)**3, Eq(d, 0)), (nan, Eq(c, -d*x)), (-B*a**2*x - B*a**2/(d*t
an(c + d*x)) - B*a*b*log(tan(c + d*x)**2 + 1)/d + 2*B*a*b*log(tan(c + d*x))
/d + B*b**2*x - C*a**2*log(tan(c + d*x)**2 + 1)/(2*d) + C*a**2*log(tan(c +
d*x))/d + 2*C*a*b*x + C*b**2*log(tan(c + d*x)**2 + 1)/(2*d), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.38 (sec) , antiderivative size = 93, normalized size of antiderivative = 1.29

$$\int \cot^3(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx =$$
$$-\frac{2\,(Ba^2-2\,Cab-Bb^2)(dx+c)+(Ca^2+2\,Bab-Cb^2)\log\left(\tan\left(dx+c\right)^2+1\right)-2\,(Ca^2+2\,Bab)\log\left(\right.}{2\,d}$$

```
[In] integrate(cot(d*x+c)^3*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="maxima")
```

```
[Out] -1/2*(2*(B*a^2 - 2*C*a*b - B*b^2)*(d*x + c) + (C*a^2 + 2*B*a*b - C*b^2)*log
(tan(d*x + c)^2 + 1) - 2*(C*a^2 + 2*B*a*b)*log(tan(d*x + c)) + 2*B*a^2/tan(
d*x + c))/d
```

Giac [A] (verification not implemented)

none

Time = 1.67 (sec) , antiderivative size = 118, normalized size of antiderivative = 1.64

$$\int \cot^3(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx =$$
$$-\frac{2\,(Ba^2-2\,Cab-Bb^2)(dx+c)+(Ca^2+2\,Bab-Cb^2)\log\left(\tan\left(dx+c\right)^2+1\right)-2\,(Ca^2+2\,Bab)\log\left(\right.}{2\,d}$$

```
[In] integrate(cot(d*x+c)^3*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="giac")
```

```
[Out] -1/2*(2*(B*a^2 - 2*C*a*b - B*b^2)*(d*x + c) + (C*a^2 + 2*B*a*b - C*b^2)*log
(tan(d*x + c)^2 + 1) - 2*(C*a^2 + 2*B*a*b)*log(abs(tan(d*x + c))) + 2*(C*a^
2*tan(d*x + c) + 2*B*a*b*tan(d*x + c) + B*a^2)/tan(d*x + c))/d
```

Mupad [B] (verification not implemented)

Time = 8.56 (sec) , antiderivative size = 100, normalized size of antiderivative = 1.39

$$\int \cot^3(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{\ln\left(\tan(c+d\,x)\right)\,\left(C\,a^2+2\,B\,b\,a\right)}{d}-\frac{\ln\left(\tan(c+d\,x)-\mathrm{i}\right)\,\left(-C+B\,1\mathrm{i}\right)\,\left(-b+a\,1\mathrm{i}\right)^2}{2\,d}$$
$$+\frac{\ln\left(\tan(c+d\,x)+1\mathrm{i}\right)\,\left(C+B\,1\mathrm{i}\right)\,\left(b+a\,1\mathrm{i}\right)^2}{2\,d}-\frac{B\,a^2\cot(c+d\,x)}{d}$$

```
[In] int(cot(c + d*x)^3*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))
^2,x)
```

```
[Out] (log(tan(c + d*x))*(C*a^2 + 2*B*a*b))/d - (log(tan(c + d*x) - 1i)*(B*1i - C
)*(a*1i - b)^2)/(2*d) + (log(tan(c + d*x) + 1i)*(B*1i + C)*(a*1i + b)^2)/(2
*d) - (B*a^2*cot(c + d*x))/d
```

3.14 $\int \cot^4(c+dx)(a+b\tan(c+dx))^2 \left(B\tan(c+dx)+C\tan$

Optimal result 149
Rubi [A] (verified) 149
Mathematica [C] (verified) 151
Maple [A] (verified) 151
Fricas [A] (verification not implemented) 152
Sympy [B] (verification not implemented) 153
Maxima [A] (verification not implemented) 153
Giac [B] (verification not implemented) 154
Mupad [B] (verification not implemented) 154

Optimal result

Integrand size = 40, antiderivative size = 88

$$\begin{aligned}
&\int \cot^4(c+dx)(a+b\tan(c+dx))^2 \left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx\\
&=\left(b^2C-a(2bB+aC)\right)x-\frac{a(2bB+aC)\cot(c+dx)}{d}\\
&\quad-\frac{a^2B\cot^2(c+dx)}{2d}-\frac{(a^2B-b^2B-2abC)\log(\sin(c+dx))}{d}
\end{aligned}$$

```
[Out] (C*b^2-a*(2*B*b+C*a))*x-a*(2*B*b+C*a)*cot(d*x+c)/d-1/2*a^2*B*cot(d*x+c)^2/d
-(B*a^2-B*b^2-2*C*a*b)*ln(sin(d*x+c))/d
```

Rubi [A] (verified)

Time = 0.31 (sec) , antiderivative size = 88, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.125, Rules used = {3713, 3685, 3709, 3612, 3556}

$$\begin{aligned}
&\int \cot^4(c+dx)(a+b\tan(c+dx))^2 \left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx\\
&=-\frac{(a^2B-2abC-b^2B)\log(\sin(c+dx))}{d}-\frac{a^2B\cot^2(c+dx)}{2d}\\
&\quad+x\left(b^2C-a(aC+2bB)\right)-\frac{a(aC+2bB)\cot(c+dx)}{d}
\end{aligned}$$

```
[In] Int[Cot[c + d*x]^4*(a + b*Tan[c + d*x])^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^
2),x]
```

```
[Out] (b^2*C - a*(2*b*B + a*C))*x - (a*(2*b*B + a*C)*Cot[c + d*x])/d - (a^2*B*Cot
[c + d*x]^2)/(2*d) - ((a^2*B - b^2*B - 2*a*b*C)*Log[Sin[c + d*x]])/d
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3612

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[(a*c + b*d)*(x/(a^2 + b^2)), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3685

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^2*((A_.) + (B_.)*tan[(e_.) + (f
_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Simp[
(-(B*c - A*d))*(b*c - a*d)^2*((c + d*Tan[e + f*x])^(n + 1)/(f*d^2*(n + 1)*(
c^2 + d^2))), x] + Dist[1/(d*(c^2 + d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)
*Simp[B*(b*c - a*d)^2 + A*d*(a^2*c - b^2*c + 2*a*b*d) + d*(B*(a^2*c - b^2*c
 + 2*a*b*d) + A*(2*a*b*c - a^2*d + b^2*d))*Tan[e + f*x] + b^2*B*(c^2 + d^2)
*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] && NeQ[b*
c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[n, -1]
```

Rule 3709

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 -
 a*b*B + a^2*C)*((a + b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1)*(a^2 + b^2))), x
] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[b*B + a*(A -
C) - (A*b - a*B - b*C)*Tan[e + f*x], x], x], x] /; FreeQ[{a, b, e, f, A, B,
 C}, x] && NeQ[A*b^2 - a*b*B + a^2*C, 0] && LtQ[m, -1] && NeQ[a^2 + b^2, 0]
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\text{integral} = \int \cot^3(c+dx)(a+b\tan(c+dx))^2(B+C\tan(c+dx))\,dx$$

$$
\begin{aligned}
&= -\frac{a^2B\cot^2(c+dx)}{2d} + \int \cot^2(c+dx)\left(a(2bB+aC)\right. \\
&\qquad \left. - \left(a^2B - b^2B - 2abC\right)\tan(c+dx) + b^2C\tan^2(c+dx)\right)\,dx \\
&= -\frac{a(2bB+aC)\cot(c+dx)}{d} - \frac{a^2B\cot^2(c+dx)}{2d} \\
&\quad + \int \cot(c+dx)\left(-a^2B + b^2B + 2abC + \left(b^2C - a(2bB+aC)\right)\tan(c+dx)\right)\,dx \\
&= \left(b^2C - a(2bB+aC)\right)x - \frac{a(2bB+aC)\cot(c+dx)}{d} \\
&\quad - \frac{a^2B\cot^2(c+dx)}{2d} + \left(-a^2B + b^2B + 2abC\right)\int \cot(c+dx)\,dx \\
&= \left(b^2C - a(2bB+aC)\right)x - \frac{a(2bB+aC)\cot(c+dx)}{d} \\
&\quad - \frac{a^2B\cot^2(c+dx)}{2d} - \frac{(a^2B - b^2B - 2abC)\log(\sin(c+dx))}{d}
\end{aligned}
$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 0.38 (sec) , antiderivative size = 123, normalized size of antiderivative = 1.40

$$
\begin{aligned}
&\int \cot^4(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx) + C\tan^2(c+dx)\right)\,dx \\
&= \frac{-2a(2bB+aC)\cot(c+dx) - a^2B\cot^2(c+dx) + (a+ib)^2(B+iC)\log(i-\tan(c+dx)) - 2(a^2B - b^2B}{2d}
\end{aligned}
$$

```
[In] Integrate[Cot[c + d*x]^4*(a + b*Tan[c + d*x])^2*(B*Tan[c + d*x] + C*Tan[c +
 d*x]^2),x]
```

```
[Out] (-2*a*(2*b*B + a*C)*Cot[c + d*x] - a^2*B*Cot[c + d*x]^2 + (a + I*b)^2*(B +
I*C)*Log[I - Tan[c + d*x]] - 2*(a^2*B - b^2*B - 2*a*b*C)*Log[Tan[c + d*x]]
+ (a - I*b)^2*(B - I*C)*Log[I + Tan[c + d*x]])/(2*d)
```

Maple [A] (verified)

Time = 0.38 (sec) , antiderivative size = 107, normalized size of antiderivative = 1.22

method	result
derivativedivides	$\frac{B\,b^2\ln(\sin(dx+c))+C\,b^2(dx+c)+2Bab(-\cot(dx+c)-dx-c)+2Cab\ln(\sin(dx+c))+B\,a^2\left(-\frac{\cot(dx+c)^2}{2}-\ln(\sin(dx+c))\right)}{d}$
default	$\frac{B\,b^2\ln(\sin(dx+c))+C\,b^2(dx+c)+2Bab(-\cot(dx+c)-dx-c)+2Cab\ln(\sin(dx+c))+B\,a^2\left(-\frac{\cot(dx+c)^2}{2}-\ln(\sin(dx+c))\right)}{d}$
parallelrisch	$\frac{(B\,a^2-B\,b^2-2Cab)\ln\left(\sec(dx+c)^2\right)+(-2B\,a^2+2B\,b^2+4Cab)\ln(\tan(dx+c))-B\,a^2\cot(dx+c)^2+(-4Bab-2C\,a^2)\cot(d}{2d}$
norman	$\frac{(-2Bab-C\,a^2+C\,b^2)x\tan(dx+c)^3-\frac{B\,a^2\tan(dx+c)}{2d}-\frac{a(2Bb+Ca)\tan(dx+c)^2}{d}}{\tan(dx+c)^3}-\frac{(B\,a^2-B\,b^2-2Cab)\ln(\tan(dx+c))}{d}+$
risch	$-iB\,b^2x-\frac{2iB\,b^2c}{d}+\frac{2iB\,a^2c}{d}-2Babx-C\,a^2x+C\,b^2x-\frac{4iCabc}{d}-\frac{2ia(2Bb\,e^{2i(dx+c)}+Ca\,e^{2i(dx+c)}}{d(e^{2i(dx+c}}$

```
[In] int(cot(d*x+c)^4*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,method=
_RETURNVERBOSE)
```

```
[Out] 1/d*(B*b^2*ln(sin(d*x+c))+C*b^2*(d*x+c)+2*B*a*b*(-cot(d*x+c)-d*x-c)+2*C*a*b
*ln(sin(d*x+c))+B*a^2*(-1/2*cot(d*x+c)^2-ln(sin(d*x+c)))+C*a^2*(-cot(d*x+c)
-d*x-c))
```

Fricas [A] (verification not implemented)

none

Time = 0.25 (sec) , antiderivative size = 122, normalized size of antiderivative = 1.39

$$\int \cot^4(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx =$$

$$-\frac{(Ba^2-2\,Cab-Bb^2)\log\left(\frac{\tan(dx+c)^2}{\tan(dx+c)^2+1}\right)\tan\left(dx+c\right)^2+Ba^2+\left(Ba^2+2\,(Ca^2+2\,Bab-Cb^2)dx\right)\tan\left(dx\right.}{2\,d\tan\left(dx+c\right)^2}$$

```
[In] integrate(cot(d*x+c)^4*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="fricas")
```

```
[Out] -1/2*((B*a^2 - 2*C*a*b - B*b^2)*log(tan(d*x + c)^2/(tan(d*x + c)^2 + 1))*ta
n(d*x + c)^2 + B*a^2 + (B*a^2 + 2*(C*a^2 + 2*B*a*b - C*b^2)*d*x)*tan(d*x +
c)^2 + 2*(C*a^2 + 2*B*a*b)*tan(d*x + c))/(d*tan(d*x + c)^2)
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 206 vs. 2(78) = 156.

Time = 1.76 (sec) , antiderivative size = 206, normalized size of antiderivative = 2.34

$$\int \cot^4(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\begin{cases}\mathrm{NaN}\\ x(a+b\tan{(c)})^2\left(B\tan{(c)}+C\tan^2{(c)}\right)\cot^4{(c)}\\ \mathrm{NaN}\\ \frac{Ba^2\log{(\tan^2{(c+dx)}+1)}}{2d}-\frac{Ba^2\log{(\tan{(c+dx)})}}{d}-\frac{Ba^2}{2d\tan^2{(c+dx)}}-2Babx-\frac{2Bab}{d\tan{(c+dx)}}-\frac{Bb^2\log{(\tan^2{(c+dx)}+1)}}{2d}+\frac{Bb^2\log\end{cases}$$

```
[In] integrate(cot(d*x+c)**4*(a+b*tan(d*x+c))**2*(B*tan(d*x+c)+C*tan(d*x+c)**2),
x)
```

```
[Out] Piecewise((nan, Eq(c, 0) & Eq(d, 0)), (x*(a + b*tan(c))**2*(B*tan(c) + C*ta
n(c)**2)*cot(c)**4, Eq(d, 0)), (nan, Eq(c, -d*x)), (B*a**2*log(tan(c + d*x)
**2 + 1)/(2*d) - B*a**2*log(tan(c + d*x))/d - B*a**2/(2*d*tan(c + d*x)**2)
- 2*B*a*b*x - 2*B*a*b/(d*tan(c + d*x)) - B*b**2*log(tan(c + d*x)**2 + 1)/(2
*d) + B*b**2*log(tan(c + d*x))/d - C*a**2*x - C*a**2/(d*tan(c + d*x)) - C*a
*b*log(tan(c + d*x)**2 + 1)/d + 2*C*a*b*log(tan(c + d*x))/d + C*b**2*x, Tru
e))
```

Maxima [A] (verification not implemented)

none

Time = 0.38 (sec) , antiderivative size = 120, normalized size of antiderivative = 1.36

$$\int \cot^4(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx =$$
$$-\frac{2\,(Ca^2+2\,Bab-Cb^2)(dx+c)-(Ba^2-2\,Cab-Bb^2)\log\left(\tan\left(dx+c\right)^2+1\right)+2\,(Ba^2-2\,Cab-Bb}{2\,d}$$

```
[In] integrate(cot(d*x+c)^4*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="maxima")
```

```
[Out] -1/2*(2*(C*a^2 + 2*B*a*b - C*b^2)*(d*x + c) - (B*a^2 - 2*C*a*b - B*b^2)*log
(tan(d*x + c)^2 + 1) + 2*(B*a^2 - 2*C*a*b - B*b^2)*log(tan(d*x + c)) + (B*a
^2 + 2*(C*a^2 + 2*B*a*b)*tan(d*x + c))/tan(d*x + c)^2)/d
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 237 vs. 2(86) = 172.

Time = 0.87 (sec) , antiderivative size = 237, normalized size of antiderivative = 2.69

$$\int \cot^4(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx = -\frac{Ba^2\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^2-4\,Ca^2\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)-8\,Bab\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)+8\,(Ca^2+2\,Bab-Cb^2)(dx+c}{}$$

```
[In] integrate(cot(d*x+c)^4*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="giac")
```

```
[Out] -1/8*(B*a^2*tan(1/2*d*x + 1/2*c)^2 - 4*C*a^2*tan(1/2*d*x + 1/2*c) - 8*B*a*b
*tan(1/2*d*x + 1/2*c) + 8*(C*a^2 + 2*B*a*b - C*b^2)*(d*x + c) - 8*(B*a^2 -
2*C*a*b - B*b^2)*log(tan(1/2*d*x + 1/2*c)^2 + 1) + 8*(B*a^2 - 2*C*a*b - B*b
^2)*log(abs(tan(1/2*d*x + 1/2*c))) - (12*B*a^2*tan(1/2*d*x + 1/2*c)^2 - 24*
C*a*b*tan(1/2*d*x + 1/2*c)^2 - 12*B*b^2*tan(1/2*d*x + 1/2*c)^2 - 4*C*a^2*ta
n(1/2*d*x + 1/2*c) - 8*B*a*b*tan(1/2*d*x + 1/2*c) - B*a^2)/tan(1/2*d*x + 1/
2*c)^2)/d
```

Mupad [B] (verification not implemented)

Time = 8.70 (sec) , antiderivative size = 127, normalized size of antiderivative = 1.44

$$\begin{aligned}&\int \cot^4(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx\\&=\frac{\ln\left(\tan(c+d\,x)\right)\,\left(-B\,a^2+2\,C\,a\,b+B\,b^2\right)}{d}\\&-\frac{\cot(c+d\,x)^2\,\left(\frac{B\,a^2}{2}+\tan(c+d\,x)\,\left(C\,a^2+2\,B\,b\,a\right)\right)}{d}\\&-\frac{\ln\left(\tan(c+d\,x)+1\mathrm{i}\right)\,\left(B-C\,1\mathrm{i}\right)\,{\left(b+a\,1\mathrm{i}\right)}^2}{2\,d}\\&-\frac{\ln\left(\tan(c+d\,x)-\mathrm{i}\right)\,\left(B+C\,1\mathrm{i}\right)\,{\left(-b+a\,1\mathrm{i}\right)}^2}{2\,d}\end{aligned}$$

```
[In] int(cot(c + d*x)^4*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))
^2,x)
```

```
[Out] (log(tan(c + d*x))*(B*b^2 - B*a^2 + 2*C*a*b))/d - (cot(c + d*x)^2*((B*a^2)/
2 + tan(c + d*x)*(C*a^2 + 2*B*a*b)))/d - (log(tan(c + d*x) + 1i)*(B - C*1i)
*(a*1i + b)^2)/(2*d) - (log(tan(c + d*x) - 1i)*(B + C*1i)*(a*1i - b)^2)/(2*
d)
```

3.15 $\int \cot^5(c+dx)(a+b\tan(c+dx))^2 \left(B\tan(c+dx)+C\tan\right.$

Optimal result 155
Rubi [A] (verified) 155
Mathematica [C] (verified) 158
Maple [A] (verified) 158
Fricas [A] (verification not implemented) 159
Sympy [B] (verification not implemented) 159
Maxima [A] (verification not implemented) 160
Giac [B] (verification not implemented) 160
Mupad [B] (verification not implemented) 161

Optimal result

Integrand size = 40, antiderivative size = 118

$$\begin{aligned}&\int \cot^5(c+dx)(a+b\tan(c+dx))^2 \left(B\tan(c+dx)+C\tan^2(c+dx)\right) \, dx\\&=\left(a^2B-b^2B-2abC\right)x+\frac{(a^2B-b^2B-2abC)\cot(c+dx)}{d}-\frac{a(2bB+aC)\cot^2(c+dx)}{2d}\\&\quad-\frac{a^2B\cot^3(c+dx)}{3d}+\frac{(b^2C-a(2bB+aC))\log(\sin(c+dx))}{d}\end{aligned}$$

```
[Out] (B*a^2-B*b^2-2*C*a*b)*x+(B*a^2-B*b^2-2*C*a*b)*cot(d*x+c)/d-1/2*a*(2*B*b+C*a
)*cot(d*x+c)^2/d-1/3*a^2*B*cot(d*x+c)^3/d+(C*b^2-a*(2*B*b+C*a))*ln(sin(d*x+
c))/d
```

Rubi [A] (verified)

Time = 0.36 (sec) , antiderivative size = 118, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.150, Rules used = {3713, 3685, 3709, 3610, 3612, 3556}

$$\begin{aligned}&\int \cot^5(c+dx)(a+b\tan(c+dx))^2 \left(B\tan(c+dx)+C\tan^2(c+dx)\right) \, dx\\&=\frac{(a^2B-2abC-b^2B)\cot(c+dx)}{d}+x\left(a^2B-2abC-b^2B\right)-\frac{a^2B\cot^3(c+dx)}{3d}\\&\quad+\frac{(b^2C-a(aC+2bB))\log(\sin(c+dx))}{d}-\frac{a(aC+2bB)\cot^2(c+dx)}{2d}\end{aligned}$$

```
[In] Int[Cot[c + d*x]^5*(a + b*Tan[c + d*x])^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^
2),x]
```

```
[Out] (a^2*B - b^2*B - 2*a*b*C)*x + ((a^2*B - b^2*B - 2*a*b*C)*Cot[c + d*x])/d -
(a*(2*b*B + a*C)*Cot[c + d*x]^2)/(2*d) - (a^2*B*Cot[c + d*x]^3)/(3*d) + ((b
^2*C - a*(2*b*B + a*C))*Log[Sin[c + d*x]])/d
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3610

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(b*c - a*d)*((a + b*Tan[e + f*x])^(m + 1)/
(f*(m + 1)*(a^2 + b^2))), x] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])
^(m + 1)*Simp[a*c + b*d - (b*c - a*d)*Tan[e + f*x], x], x], x] /; FreeQ[{a,
 b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && LtQ[m, -1]
```

Rule 3612

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[(a*c + b*d)*(x/(a^2 + b^2)), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3685

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^2*((A_.) + (B_.)*tan[(e_.) + (f
_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Simp[
(-(B*c - A*d))*(b*c - a*d)^2*((c + d*Tan[e + f*x])^(n + 1)/(f*d^2*(n + 1)*(
c^2 + d^2))), x] + Dist[1/(d*(c^2 + d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)
*Simp[B*(b*c - a*d)^2 + A*d*(a^2*c - b^2*c + 2*a*b*d) + d*(B*(a^2*c - b^2*c
 + 2*a*b*d) + A*(2*a*b*c - a^2*d + b^2*d))*Tan[e + f*x] + b^2*B*(c^2 + d^2)
*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] && NeQ[b*
c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[n, -1]
```

Rule 3709

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 -
 a*b*B + a^2*C)*((a + b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1)*(a^2 + b^2))), x
] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[b*B + a*(A -
C) - (A*b - a*B - b*C)*Tan[e + f*x], x], x], x] /; FreeQ[{a, b, e, f, A, B,
 C}, x] && NeQ[A*b^2 - a*b*B + a^2*C, 0] && LtQ[m, -1] && NeQ[a^2 + b^2, 0]
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \cot^4(c+dx)(a+b\tan(c+dx))^2(B+C\tan(c+dx))\,dx \\
&= -\frac{a^2B\cot^3(c+dx)}{3d}+\int \cot^3(c+dx)\left(a(2bB+aC)\right. \\
&\qquad \left.-\left(a^2B-b^2B-2abC\right)\tan(c+dx)+b^2C\tan^2(c+dx)\right)\,dx \\
&= -\frac{a(2bB+aC)\cot^2(c+dx)}{2d}-\frac{a^2B\cot^3(c+dx)}{3d} \\
&\quad +\int \cot^2(c+dx)\left(-a^2B+b^2B+2abC+\left(b^2C-a(2bB+aC)\right)\tan(c+dx)\right)\,dx \\
&= \frac{(a^2B-b^2B-2abC)\cot(c+dx)}{d}-\frac{a(2bB+aC)\cot^2(c+dx)}{2d}-\frac{a^2B\cot^3(c+dx)}{3d} \\
&\quad +\int \cot(c+dx)\left(b^2C-a(2bB+aC)+\left(a^2B-b^2B-2abC\right)\tan(c+dx)\right)\,dx \\
&= \left(a^2B-b^2B-2abC\right)x+\frac{(a^2B-b^2B-2abC)\cot(c+dx)}{d} \\
&\quad -\frac{a(2bB+aC)\cot^2(c+dx)}{2d}-\frac{a^2B\cot^3(c+dx)}{3d} \\
&\quad +\left(b^2C-a(2bB+aC)\right)\int \cot(c+dx)\,dx \\
&= \left(a^2B-b^2B-2abC\right)x+\frac{(a^2B-b^2B-2abC)\cot(c+dx)}{d} \\
&\quad -\frac{a(2bB+aC)\cot^2(c+dx)}{2d}-\frac{a^2B\cot^3(c+dx)}{3d} \\
&\quad +\frac{(b^2C-a(2bB+aC))\log(\sin(c+dx))}{d}
\end{aligned}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 1.27 (sec) , antiderivative size = 152, normalized size of antiderivative = 1.29

$$\int \cot^5(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$= \frac{6(a^2B-b^2B-2abC)\cot(c+dx)-3a(2bB+aC)\cot^2(c+dx)-2a^2B\cot^3(c+dx)+3(a+ib)^2(-iB+}{}$$

```
[In] Integrate[Cot[c + d*x]^5*(a + b*Tan[c + d*x])^2*(B*Tan[c + d*x] + C*Tan[c +
 d*x]^2),x]
```

```
[Out] (6*(a^2*B - b^2*B - 2*a*b*C)*Cot[c + d*x] - 3*a*(2*b*B + a*C)*Cot[c + d*x]^
2 - 2*a^2*B*Cot[c + d*x]^3 + 3*(a + I*b)^2*((-I)*B + C)*Log[I - Tan[c + d*x
]] - 6*(2*a*b*B + a^2*C - b^2*C)*Log[Tan[c + d*x]] + 3*(a - I*b)^2*(I*B + C
)*Log[I + Tan[c + d*x]])/(6*d)
```

Maple [A] (verified)

Time = 0.40 (sec) , antiderivative size = 136, normalized size of antiderivative = 1.15

method	result
derivativedivides	$\frac{B\,b^2(-\cot(dx+c)-dx-c)+C\,b^2\ln(\sin(dx+c))+2Bab\left(-\frac{\cot(dx+c)^2}{2}-\ln(\sin(dx+c))\right)+2Cab(-\cot(dx+c)-dx-c)+B\,a^2}{d}$
default	$\frac{B\,b^2(-\cot(dx+c)-dx-c)+C\,b^2\ln(\sin(dx+c))+2Bab\left(-\frac{\cot(dx+c)^2}{2}-\ln(\sin(dx+c))\right)+2Cab(-\cot(dx+c)-dx-c)+B\,a^2}{d}$
parallelrisch	$\frac{3(2Bab+C\,a^2-C\,b^2)\ln\left(\sec(dx+c)^2\right)+6(-2Bab-C\,a^2+C\,b^2)\ln(\tan(dx+c))-2B\,a^2\cot(dx+c)^3+3(-2Bab-C\,a^2)\cot(}{6d}$
norman	$\frac{\frac{(B\,a^2-B\,b^2-2Cab)\tan(dx+c)^3}{d}+(B\,a^2-B\,b^2-2Cab)x\tan(dx+c)^4-\frac{B\,a^2\tan(dx+c)}{3d}-\frac{a(2Bb+Ca)\tan(dx+c)^2}{2d}}{\tan(dx+c)^4}-\frac{(2Bab+C}{}$
risch	$B\,a^2x-B\,b^2x-2Cabx+\frac{4iBabc}{d}-iC\,b^2x+iC\,a^2x-\frac{2i(6iBab\,e^{4i(dx+c)}+3iC\,a^2e^{4i(dx+c)}-6B\,a^2e^{4}}{}$

```
[In] int(cot(d*x+c)^5*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,method=
_RETURNVERBOSE)
```

```
[Out] 1/d*(B*b^2*(-cot(d*x+c)-d*x-c)+C*b^2*ln(sin(d*x+c))+2*B*a*b*(-1/2*cot(d*x+c
)^2-ln(sin(d*x+c)))+2*C*a*b*(-cot(d*x+c)-d*x-c)+B*a^2*(-1/3*cot(d*x+c)^3+co
t(d*x+c)+d*x+c)+C*a^2*(-1/2*cot(d*x+c)^2-ln(sin(d*x+c))))
```

Fricas [A] (verification not implemented)

none

Time = 0.26 (sec) , antiderivative size = 157, normalized size of antiderivative = 1.33

$$\int \cot^5(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx =$$
$$-\frac{3\left(Ca^2+2\,Bab-Cb^2\right)\log\left(\frac{\tan(dx+c)^2}{\tan(dx+c)^2+1}\right)\tan\left(dx+c\right)^3+3\left(Ca^2+2\,Bab-2\left(Ba^2-2\,Cab-Bb^2\right)dx\right)\tan}{6\,d\tan\left(dx+\right.}$$

```
[In] integrate(cot(d*x+c)^5*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="fricas")
```

```
[Out] -1/6*(3*(C*a^2 + 2*B*a*b - C*b^2)*log(tan(d*x + c)^2/(tan(d*x + c)^2 + 1))*
tan(d*x + c)^3 + 3*(C*a^2 + 2*B*a*b - 2*(B*a^2 - 2*C*a*b - B*b^2)*d*x)*tan(
d*x + c)^3 + 2*B*a^2 - 6*(B*a^2 - 2*C*a*b - B*b^2)*tan(d*x + c)^2 + 3*(C*a^
2 + 2*B*a*b)*tan(d*x + c))/(d*tan(d*x + c)^3)
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 252 vs. 2(107) = 214.

Time = 2.31 (sec) , antiderivative size = 252, normalized size of antiderivative = 2.14

$$\int \cot^5(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\begin{cases}\text{NaN}\\ x(a+b\tan(c))^2\left(B\tan(c)+C\tan^2(c)\right)\cot^5(c)\\ \text{NaN}\\ Ba^2x+\frac{Ba^2}{d\tan(c+dx)}-\frac{Ba^2}{3d\tan^3(c+dx)}+\frac{Bab\log(\tan^2(c+dx)+1)}{d}-\frac{2Bab\log(\tan(c+dx))}{d}-\frac{Bab}{d\tan^2(c+dx)}-Bb^2x-\frac{Bb}{d\tan(c}\end{cases}$$

```
[In] integrate(cot(d*x+c)**5*(a+b*tan(d*x+c))**2*(B*tan(d*x+c)+C*tan(d*x+c)**2),
x)
```

```
[Out] Piecewise((nan, Eq(c, 0) & Eq(d, 0)), (x*(a + b*tan(c))**2*(B*tan(c) + C*ta
n(c)**2)*cot(c)**5, Eq(d, 0)), (nan, Eq(c, -d*x)), (B*a**2*x + B*a**2/(d*ta
n(c + d*x)) - B*a**2/(3*d*tan(c + d*x)**3) + B*a*b*log(tan(c + d*x)**2 + 1)
/d - 2*B*a*b*log(tan(c + d*x))/d - B*a*b/(d*tan(c + d*x)**2) - B*b**2*x - B
*b**2/(d*tan(c + d*x)) + C*a**2*log(tan(c + d*x)**2 + 1)/(2*d) - C*a**2*log
(tan(c + d*x))/d - C*a**2/(2*d*tan(c + d*x)**2) - 2*C*a*b*x - 2*C*a*b/(d*ta
n(c + d*x)) - C*b**2*log(tan(c + d*x)**2 + 1)/(2*d) + C*b**2*log(tan(c + d*
x))/d, True))
```

Maxima [A] (verification not implemented)

none

Time = 0.38 (sec) , antiderivative size = 149, normalized size of antiderivative = 1.26

$$\int \cot^5(c+dx)(a+b\tan(c+dx))^2 \left(B\tan(c+dx)+C\tan^2(c+dx)\right) dx$$
$$= \frac{6\,(Ba^2-2\,Cab-Bb^2)(dx+c)+3\,(Ca^2+2\,Bab-Cb^2)\log\left(\tan\left(dx+c\right)^2+1\right)-6\,(Ca^2+2\,Bab-Cb^2)}{6\,d}$$

```
[In] integrate(cot(d*x+c)^5*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="maxima")
```

```
[Out] 1/6*(6*(B*a^2 - 2*C*a*b - B*b^2)*(d*x + c) + 3*(C*a^2 + 2*B*a*b - C*b^2)*lo
g(tan(d*x + c)^2 + 1) - 6*(C*a^2 + 2*B*a*b - C*b^2)*log(tan(d*x + c)) - (2*
B*a^2 - 6*(B*a^2 - 2*C*a*b - B*b^2)*tan(d*x + c)^2 + 3*(C*a^2 + 2*B*a*b)*ta
n(d*x + c))/tan(d*x + c)^3)/d
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 334 vs. 2(114) = 228.

Time = 0.91 (sec) , antiderivative size = 334, normalized size of antiderivative = 2.83

$$\int \cot^5(c+dx)(a+b\tan(c+dx))^2 \left(B\tan(c+dx)+C\tan^2(c+dx)\right) dx$$
$$= \frac{Ba^2\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^3-3\,Ca^2\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^2-6\,Bab\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^2-15\,Ba^2\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)+24\,C}{}$$

```
[In] integrate(cot(d*x+c)^5*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="giac")
```

```
[Out] 1/24*(B*a^2*tan(1/2*d*x + 1/2*c)^3 - 3*C*a^2*tan(1/2*d*x + 1/2*c)^2 - 6*B*a
*b*tan(1/2*d*x + 1/2*c)^2 - 15*B*a^2*tan(1/2*d*x + 1/2*c) + 24*C*a*b*tan(1/
2*d*x + 1/2*c) + 12*B*b^2*tan(1/2*d*x + 1/2*c) + 24*(B*a^2 - 2*C*a*b - B*b^
2)*(d*x + c) + 24*(C*a^2 + 2*B*a*b - C*b^2)*log(tan(1/2*d*x + 1/2*c)^2 + 1)
 - 24*(C*a^2 + 2*B*a*b - C*b^2)*log(abs(tan(1/2*d*x + 1/2*c))) + (44*C*a^2*
tan(1/2*d*x + 1/2*c)^3 + 88*B*a*b*tan(1/2*d*x + 1/2*c)^3 - 44*C*b^2*tan(1/2
*d*x + 1/2*c)^3 + 15*B*a^2*tan(1/2*d*x + 1/2*c)^2 - 24*C*a*b*tan(1/2*d*x +
1/2*c)^2 - 12*B*b^2*tan(1/2*d*x + 1/2*c)^2 - 3*C*a^2*tan(1/2*d*x + 1/2*c) -
 6*B*a*b*tan(1/2*d*x + 1/2*c) - B*a^2)/tan(1/2*d*x + 1/2*c)^3)/d
```

Mupad [B] (verification not implemented)

Time = 8.74 (sec) , antiderivative size = 156, normalized size of antiderivative = 1.32

$$\int \cot^5(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx =$$

$$-\frac{\cot(c+d\,x)^3\,\left(\frac{B\,a^2}{3}+\tan(c+d\,x)^2\,\left(-B\,a^2+2\,C\,a\,b+B\,b^2\right)+\tan(c+d\,x)\,\left(\frac{C\,a^2}{2}+B\,b\,a\right)\right)}{d}$$
$$-\frac{\ln\left(\tan(c+d\,x)\right)\,\left(C\,a^2+2\,B\,a\,b-C\,b^2\right)}{d}$$
$$+\frac{\ln\left(\tan(c+d\,x)-\mathrm{i}\right)\,\left(-C+B\,1\mathrm{i}\right)\,\left(-b+a\,1\mathrm{i}\right)^2}{2\,d}$$
$$-\frac{\ln\left(\tan(c+d\,x)+1\mathrm{i}\right)\,\left(C+B\,1\mathrm{i}\right)\,\left(b+a\,1\mathrm{i}\right)^2}{2\,d}$$

```
[In] int(cot(c + d*x)^5*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))
^2,x)
```

```
[Out] (log(tan(c + d*x) - 1i)*(B*1i - C)*(a*1i - b)^2)/(2*d) - (log(tan(c + d*x))
*(C*a^2 - C*b^2 + 2*B*a*b))/d - (cot(c + d*x)^3*((B*a^2)/3 + tan(c + d*x)^2
*(B*b^2 - B*a^2 + 2*C*a*b) + tan(c + d*x)*((C*a^2)/2 + B*a*b)))/d - (log(ta
n(c + d*x) + 1i)*(B*1i + C)*(a*1i + b)^2)/(2*d)
```

3.16 $\int \cot^6(c+dx)(a+b\tan(c+dx))^2 \left(B\tan(c+dx)+C\tan^2(c+dx)\right)dx$

Optimal result 162
Rubi [A] (verified) 162
Mathematica [C] (verified) 165
Maple [A] (verified) 165
Fricas [A] (verification not implemented) 166
Sympy [B] (verification not implemented) 166
Maxima [A] (verification not implemented) 167
Giac [B] (verification not implemented) 167
Mupad [B] (verification not implemented) 168

Optimal result

Integrand size = 40, antiderivative size = 151

$$\begin{aligned}&\int \cot^6(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx\\&=\left(2abB+a^2C-b^2C\right)x-\frac{(b^2C-a(2bB+aC))\cot(c+dx)}{d}\\&\quad+\frac{(a^2B-b^2B-2abC)\cot^2(c+dx)}{2d}-\frac{a(2bB+aC)\cot^3(c+dx)}{3d}\\&\quad-\frac{a^2B\cot^4(c+dx)}{4d}+\frac{(a^2B-b^2B-2abC)\log(\sin(c+dx))}{d}\end{aligned}$$

```
[Out] (2*B*a*b+C*a^2-C*b^2)*x-(C*b^2-a*(2*B*b+C*a))*cot(d*x+c)/d+1/2*(B*a^2-B*b^2
-2*C*a*b)*cot(d*x+c)^2/d-1/3*a*(2*B*b+C*a)*cot(d*x+c)^3/d-1/4*a^2*B*cot(d*x
+c)^4/d+(B*a^2-B*b^2-2*C*a*b)*ln(sin(d*x+c))/d
```

Rubi [A] (verified)

Time = 0.43 (sec) , antiderivative size = 151, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.150, Rules used = {3713, 3685, 3709, 3610, 3612, 3556}

$$\begin{aligned}&\int \cot^6(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx\\&=\frac{(a^2B-2abC-b^2B)\cot^2(c+dx)}{2d}+\frac{(a^2B-2abC-b^2B)\log(\sin(c+dx))}{d}\\&\quad+x\left(a^2C+2abB-b^2C\right)-\frac{a^2B\cot^4(c+dx)}{4d}\\&\quad-\frac{(b^2C-a(aC+2bB))\cot(c+dx)}{d}-\frac{a(aC+2bB)\cot^3(c+dx)}{3d}\end{aligned}$$

```
[In] Int[Cot[c + d*x]^6*(a + b*Tan[c + d*x])^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^
2),x]
```

```
[Out] (2*a*b*B + a^2*C - b^2*C)*x - ((b^2*C - a*(2*b*B + a*C))*Cot[c + d*x])/d +
((a^2*B - b^2*B - 2*a*b*C)*Cot[c + d*x]^2)/(2*d) - (a*(2*b*B + a*C)*Cot[c +
 d*x]^3)/(3*d) - (a^2*B*Cot[c + d*x]^4)/(4*d) + ((a^2*B - b^2*B - 2*a*b*C)*
Log[Sin[c + d*x]])/d
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3610

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(b*c - a*d)*((a + b*Tan[e + f*x])^(m + 1)/
(f*(m + 1)*(a^2 + b^2))), x] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])
^(m + 1)*Simp[a*c + b*d - (b*c - a*d)*Tan[e + f*x], x], x], x] /; FreeQ[{a,
 b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && LtQ[m, -1]
```

Rule 3612

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[(a*c + b*d)*(x/(a^2 + b^2)), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3685

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^2*((A_.) + (B_.)*tan[(e_.) + (f
_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Simp[
(-(B*c - A*d))*(b*c - a*d)^2*((c + d*Tan[e + f*x])^(n + 1)/(f*d^2*(n + 1)*(
c^2 + d^2))), x] + Dist[1/(d*(c^2 + d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)
*Simp[B*(b*c - a*d)^2 + A*d*(a^2*c - b^2*c + 2*a*b*d) + d*(B*(a^2*c - b^2*c
 + 2*a*b*d) + A*(2*a*b*c - a^2*d + b^2*d))*Tan[e + f*x] + b^2*B*(c^2 + d^2)
*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] && NeQ[b*
c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[n, -1]
```

Rule 3709

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 -
 a*b*B + a^2*C)*((a + b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1)*(a^2 + b^2))), x
] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[b*B + a*(A -
C) - (A*b - a*B - b*C)*Tan[e + f*x], x], x], x] /; FreeQ[{a, b, e, f, A, B,
```

```
C}, x] && NeQ[A*b^2 - a*b*B + a^2*C, 0] && LtQ[m, -1] && NeQ[a^2 + b^2, 0]
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$
\begin{aligned}
\text{integral} &= \int \cot^5(c+dx)(a+b\tan(c+dx))^2(B+C\tan(c+dx))\,dx \\
&= -\frac{a^2B\cot^4(c+dx)}{4d} + \int \cot^4(c+dx)\left(a(2bB+aC)\right. \\
&\qquad \left. -\left(a^2B-b^2B-2abC\right)\tan(c+dx)+b^2C\tan^2(c+dx)\right)\,dx \\
&= -\frac{a(2bB+aC)\cot^3(c+dx)}{3d} - \frac{a^2B\cot^4(c+dx)}{4d} \\
&\quad + \int \cot^3(c+dx)\left(-a^2B+b^2B+2abC+\left(b^2C-a(2bB+aC)\right)\tan(c+dx)\right)\,dx \\
&= \frac{(a^2B-b^2B-2abC)\cot^2(c+dx)}{2d} - \frac{a(2bB+aC)\cot^3(c+dx)}{3d} - \frac{a^2B\cot^4(c+dx)}{4d} \\
&\quad + \int \cot^2(c+dx)\left(b^2C-a(2bB+aC)+\left(a^2B-b^2B-2abC\right)\tan(c+dx)\right)\,dx \\
&= -\frac{(b^2C-a(2bB+aC))\cot(c+dx)}{d} + \frac{(a^2B-b^2B-2abC)\cot^2(c+dx)}{2d} \\
&\quad - \frac{a(2bB+aC)\cot^3(c+dx)}{3d} - \frac{a^2B\cot^4(c+dx)}{4d} \\
&\quad + \int \cot(c+dx)\left(a^2B-b^2B-2abC+\left(2abB+a^2C-b^2C\right)\tan(c+dx)\right)\,dx \\
&= \left(2abB+a^2C-b^2C\right)x - \frac{(b^2C-a(2bB+aC))\cot(c+dx)}{d} \\
&\quad + \frac{(a^2B-b^2B-2abC)\cot^2(c+dx)}{2d} - \frac{a(2bB+aC)\cot^3(c+dx)}{3d} \\
&\quad - \frac{a^2B\cot^4(c+dx)}{4d} + \left(a^2B-b^2B-2abC\right)\int \cot(c+dx)\,dx
\end{aligned}
$$

$$= \left(2abB + a^2C - b^2C\right)x - \frac{(b^2C - a(2bB + aC))\cot(c + dx)}{d} + \frac{(a^2B - b^2B - 2abC)\cot^2(c + dx)}{2d} - \frac{a(2bB + aC)\cot^3(c + dx)}{3d} - \frac{a^2B\cot^4(c + dx)}{4d} + \frac{(a^2B - b^2B - 2abC)\log(\sin(c + dx))}{d}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 3.15 (sec) , antiderivative size = 180, normalized size of antiderivative = 1.19

$$\int \cot^6(c + dx)(a + b\tan(c + dx))^2 \left(B\tan(c + dx) + C\tan^2(c + dx)\right) dx = \frac{12(2abB + a^2C - b^2C)\cot(c + dx) + 6(a^2B - b^2B - 2abC)\cot^2(c + dx) - 4a(2bB + aC)\cot^3(c + dx) -}{}$$

```
[In] Integrate[Cot[c + d*x]^6*(a + b*Tan[c + d*x])^2*(B*Tan[c + d*x] + C*Tan[c +
 d*x]^2),x]
```

```
[Out] (12*(2*a*b*B + a^2*C - b^2*C)*Cot[c + d*x] + 6*(a^2*B - b^2*B - 2*a*b*C)*Co
t[c + d*x]^2 - 4*a*(2*b*B + a*C)*Cot[c + d*x]^3 - 3*a^2*B*Cot[c + d*x]^4 -
6*((a + I*b)^2*(B + I*C)*Log[I - Tan[c + d*x]] + (-2*a^2*B + 2*b^2*B + 4*a*
b*C)*Log[Tan[c + d*x]] + (a - I*b)^2*(B - I*C)*Log[I + Tan[c + d*x]]))/(12*
d)
```

Maple [A] (verified)

Time = 0.46 (sec) , antiderivative size = 162, normalized size of antiderivative = 1.07

method	result
derivativedivides	$\frac{B\,b^2\left(-\frac{\cot(dx+c)^2}{2}-\ln(\sin(dx+c))\right)+C\,b^2(-\cot(dx+c)-dx-c)+2Bab\left(-\frac{\cot(dx+c)^3}{3}+\cot(dx+c)+dx+c\right)+2Cab\left(-\frac{c}{}\right.}{}$
default	$\frac{B\,b^2\left(-\frac{\cot(dx+c)^2}{2}-\ln(\sin(dx+c))\right)+C\,b^2(-\cot(dx+c)-dx-c)+2Bab\left(-\frac{\cot(dx+c)^3}{3}+\cot(dx+c)+dx+c\right)+2Cab\left(-\frac{c}{}\right.}{}$
parallelrisch	$\frac{6\left(-B\,a^2+B\,b^2+2Cab\right)\ln\left(\sec(dx+c)^2\right)+12\left(B\,a^2-B\,b^2-2Cab\right)\ln(\tan(dx+c))-3B\,a^2\cot(dx+c)^4+4\left(-2Bab-C\,a^2\right)c}{12d}$
norman	$\frac{\frac{\left(2Bab+C\,a^2-C\,b^2\right)\tan(dx+c)^4}{d}+\left(2Bab+C\,a^2-C\,b^2\right)x\tan(dx+c)^5+\frac{\left(B\,a^2-B\,b^2-2Cab\right)\tan(dx+c)^3}{2d}-\frac{B\,a^2\tan(dx+c)}{4d}-\frac{a(2}{}}{\tan(dx+c)^5}$
risch	$iB\,b^2x+\frac{2iB\,b^2c}{d}-\frac{2iB\,a^2c}{d}+2Babx+C\,a^2x-C\,b^2x+2iCabx+\frac{4iCabc}{d}-iB\,a^2x+\frac{20iC}{}$

```
[In] int(cot(d*x+c)^6*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,method=
_RETURNVERBOSE)
```

```
[Out] 1/d*(B*b^2*(-1/2*cot(d*x+c)^2-ln(sin(d*x+c)))+C*b^2*(-cot(d*x+c)-d*x-c)+2*B
*a*b*(-1/3*cot(d*x+c)^3+cot(d*x+c)+d*x+c)+2*C*a*b*(-1/2*cot(d*x+c)^2-ln(sin
(d*x+c)))+B*a^2*(-1/4*cot(d*x+c)^4+1/2*cot(d*x+c)^2+ln(sin(d*x+c)))+C*a^2*(
-1/3*cot(d*x+c)^3+cot(d*x+c)+d*x+c))
```

Fricas [A] (verification not implemented)

none

Time = 0.25 (sec) , antiderivative size = 191, normalized size of antiderivative = 1.26

$$\int \cot^6(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{6\left(Ba^2-2\,Cab-Bb^2\right)\log\left(\frac{\tan(dx+c)^2}{\tan(dx+c)^2+1}\right)\tan\left(dx+c\right)^4+3\left(3\,Ba^2-4\,Cab-2\,Bb^2+4\left(Ca^2+2\,Bab-Cb\right.\right.}{}$$

```
[In] integrate(cot(d*x+c)^6*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="fricas")
```

```
[Out] 1/12*(6*(B*a^2 - 2*C*a*b - B*b^2)*log(tan(d*x + c)^2/(tan(d*x + c)^2 + 1))*
tan(d*x + c)^4 + 3*(3*B*a^2 - 4*C*a*b - 2*B*b^2 + 4*(C*a^2 + 2*B*a*b - C*b^
2)*d*x)*tan(d*x + c)^4 + 12*(C*a^2 + 2*B*a*b - C*b^2)*tan(d*x + c)^3 - 3*B*
a^2 + 6*(B*a^2 - 2*C*a*b - B*b^2)*tan(d*x + c)^2 - 4*(C*a^2 + 2*B*a*b)*tan(
d*x + c))/(d*tan(d*x + c)^4)
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 304 vs. 2(136) = 272.

Time = 4.12 (sec) , antiderivative size = 304, normalized size of antiderivative = 2.01

$$\int \cot^6(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\begin{cases}\text{NaN}\\ x(a+b\tan(c))^2\left(B\tan(c)+C\tan^2(c)\right)\cot^6(c)\\ \text{NaN}\\ -\frac{Ba^2\log(\tan^2(c+dx)+1)}{2d}+\frac{Ba^2\log(\tan(c+dx))}{d}+\frac{Ba^2}{2d\tan^2(c+dx)}-\frac{Ba^2}{4d\tan^4(c+dx)}+2Babx+\frac{2Bab}{d\tan(c+dx)}-\frac{2Bab}{3d\tan^3(c+dx)}\end{cases}$$

```
[In] integrate(cot(d*x+c)**6*(a+b*tan(d*x+c))**2*(B*tan(d*x+c)+C*tan(d*x+c)**2),
x)
```

```
[Out] Piecewise((nan, Eq(c, 0) & Eq(d, 0)), (x*(a + b*tan(c))**2*(B*tan(c) + C*ta
n(c)**2)*cot(c)**6, Eq(d, 0)), (nan, Eq(c, -d*x)), (-B*a**2*log(tan(c + d*x
)**2 + 1)/(2*d) + B*a**2*log(tan(c + d*x))/d + B*a**2/(2*d*tan(c + d*x)**2)
 - B*a**2/(4*d*tan(c + d*x)**4) + 2*B*a*b*x + 2*B*a*b/(d*tan(c + d*x)) - 2*
B*a*b/(3*d*tan(c + d*x)**3) + B*b**2*log(tan(c + d*x)**2 + 1)/(2*d) - B*b**
2*log(tan(c + d*x))/d - B*b**2/(2*d*tan(c + d*x)**2) + C*a**2*x + C*a**2/(d
*tan(c + d*x)) - C*a**2/(3*d*tan(c + d*x)**3) + C*a*b*log(tan(c + d*x)**2 +
 1)/d - 2*C*a*b*log(tan(c + d*x))/d - C*a*b/(d*tan(c + d*x)**2) - C*b**2*x
- C*b**2/(d*tan(c + d*x)), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.63 (sec) , antiderivative size = 175, normalized size of antiderivative = 1.16

$$\int \cot^6(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{12\,(Ca^2+2\,Bab-Cb^2)(dx+c)-6\,(Ba^2-2\,Cab-Bb^2)\log\left(\tan\left(dx+c\right)^2+1\right)+12\,(Ba^2-2\,Cab-E}{12\,d}$$

```
[In] integrate(cot(d*x+c)^6*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="maxima")
```

```
[Out] 1/12*(12*(C*a^2 + 2*B*a*b - C*b^2)*(d*x + c) - 6*(B*a^2 - 2*C*a*b - B*b^2)*
log(tan(d*x + c)^2 + 1) + 12*(B*a^2 - 2*C*a*b - B*b^2)*log(tan(d*x + c)) +
(12*(C*a^2 + 2*B*a*b - C*b^2)*tan(d*x + c)^3 - 3*B*a^2 + 6*(B*a^2 - 2*C*a*b
 - B*b^2)*tan(d*x + c)^2 - 4*(C*a^2 + 2*B*a*b)*tan(d*x + c))/tan(d*x + c)^4
)/d
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 435 vs. 2(145) = 290.

Time = 0.98 (sec) , antiderivative size = 435, normalized size of antiderivative = 2.88

$$\int \cot^6(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx=$$
$$-\frac{3\,Ba^2\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^4-8\,Ca^2\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^3-16\,Bab\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^3-36\,Ba^2\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)}{}$$

```
[In] integrate(cot(d*x+c)^6*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="giac")
```

```
[Out] -1/192*(3*B*a^2*tan(1/2*d*x + 1/2*c)^4 - 8*C*a^2*tan(1/2*d*x + 1/2*c)^3 - 1
6*B*a*b*tan(1/2*d*x + 1/2*c)^3 - 36*B*a^2*tan(1/2*d*x + 1/2*c)^2 + 48*C*a*b
```

```
*tan(1/2*d*x + 1/2*c)^2 + 24*B*b^2*tan(1/2*d*x + 1/2*c)^2 + 120*C*a^2*tan(1
/2*d*x + 1/2*c) + 240*B*a*b*tan(1/2*d*x + 1/2*c) - 96*C*b^2*tan(1/2*d*x + 1
/2*c) - 192*(C*a^2 + 2*B*a*b - C*b^2)*(d*x + c) + 192*(B*a^2 - 2*C*a*b - B*
b^2)*log(tan(1/2*d*x + 1/2*c)^2 + 1) - 192*(B*a^2 - 2*C*a*b - B*b^2)*log(ab
s(tan(1/2*d*x + 1/2*c))) + (400*B*a^2*tan(1/2*d*x + 1/2*c)^4 - 800*C*a*b*ta
n(1/2*d*x + 1/2*c)^4 - 400*B*b^2*tan(1/2*d*x + 1/2*c)^4 - 120*C*a^2*tan(1/2
*d*x + 1/2*c)^3 - 240*B*a*b*tan(1/2*d*x + 1/2*c)^3 + 96*C*b^2*tan(1/2*d*x +
 1/2*c)^3 - 36*B*a^2*tan(1/2*d*x + 1/2*c)^2 + 48*C*a*b*tan(1/2*d*x + 1/2*c)
^2 + 24*B*b^2*tan(1/2*d*x + 1/2*c)^2 + 8*C*a^2*tan(1/2*d*x + 1/2*c) + 16*B*
a*b*tan(1/2*d*x + 1/2*c) + 3*B*a^2)/tan(1/2*d*x + 1/2*c)^4)/d
```

Mupad [B] (verification not implemented)

Time = 8.71 (sec) , antiderivative size = 182, normalized size of antiderivative = 1.21

$$\int \cot^6(c+dx)(a+b\tan(c+dx))^2 \left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx = -\frac{\cot(c+d\,x)^4\left(\frac{B\,a^2}{4}+\tan(c+d\,x)^2\left(-\frac{B\,a^2}{2}+C\,a\,b+\frac{B\,b^2}{2}\right)-\tan(c+d\,x)^3\left(C\,a^2+2\,B\,a\,b-C\,b^2\right)+\tan\right.}{d} - \frac{\ln\left(\tan(c+d\,x)\right)\,\left(-B\,a^2+2\,C\,a\,b+B\,b^2\right)}{d} + \frac{\ln\left(\tan(c+d\,x)+1\mathrm{i}\right)\,\left(B-C\,1\mathrm{i}\right)\,\left(b+a\,1\mathrm{i}\right)^2}{2\,d} + \frac{\ln\left(\tan(c+d\,x)-\mathrm{i}\right)\,\left(B+C\,1\mathrm{i}\right)\,\left(-b+a\,1\mathrm{i}\right)^2}{2\,d}$$

```
[In] int(cot(c + d*x)^6*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))
^2,x)
```

```
[Out] (log(tan(c + d*x) + 1i)*(B - C*1i)*(a*1i + b)^2)/(2*d) - (log(tan(c + d*x))
*(B*b^2 - B*a^2 + 2*C*a*b))/d - (cot(c + d*x)^4*((B*a^2)/4 + tan(c + d*x)^2
*((B*b^2)/2 - (B*a^2)/2 + C*a*b) - tan(c + d*x)^3*(C*a^2 - C*b^2 + 2*B*a*b)
 + tan(c + d*x)*((C*a^2)/3 + (2*B*a*b)/3)))/d + (log(tan(c + d*x) - 1i)*(B
+ C*1i)*(a*1i - b)^2)/(2*d)
```

3.17 $\int (a+b\tan(c+dx))^3 \left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$

Optimal result 169
Rubi [A] (verified) 169
Mathematica [C] (verified) 171
Maple [A] (verified) 171
Fricas [A] (verification not implemented) 172
Sympy [B] (verification not implemented) 173
Maxima [A] (verification not implemented) 173
Giac [B] (verification not implemented) 174
Mupad [B] (verification not implemented) 176

Optimal result

Integrand size = 32, antiderivative size = 165

$$\begin{aligned}&\int (a+b\tan(c+dx))^3 \left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx\\&=-\left(\left(3a^2bB-b^3B+a^3C-3ab^2C\right)x\right)-\frac{(a^3B-3ab^2B-3a^2bC+b^3C)\log(\cos(c+dx))}{d}\\&\quad+\frac{b(a^2B-b^2B-2abC)\tan(c+dx)}{d}+\frac{(aB-bC)(a+b\tan(c+dx))^2}{2d}\\&\quad+\frac{B(a+b\tan(c+dx))^3}{3d}+\frac{C(a+b\tan(c+dx))^4}{4bd}\end{aligned}$$

```
[Out] -(3*B*a^2*b-B*b^3+C*a^3-3*C*a*b^2)*x-(B*a^3-3*B*a*b^2-3*C*a^2*b+C*b^3)*ln(c
os(d*x+c))/d+b*(B*a^2-B*b^2-2*C*a*b)*tan(d*x+c)/d+1/2*(B*a-C*b)*(a+b*tan(d*
x+c))^2/d+1/3*B*(a+b*tan(d*x+c))^3/d+1/4*C*(a+b*tan(d*x+c))^4/b/d
```

Rubi [A] (verified)

Time = 0.22 (sec) , antiderivative size = 165, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 4, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.125, Rules used = {3711, 3609, 3606, 3556}

$$\begin{aligned}&\int (a+b\tan(c+dx))^3 \left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx\\&=\frac{b(a^2B-2abC-b^2B)\tan(c+dx)}{d}-\frac{(a^3B-3a^2bC-3ab^2B+b^3C)\log(\cos(c+dx))}{d}\\&\quad-x\left(a^3C+3a^2bB-3ab^2C-b^3B\right)+\frac{(aB-bC)(a+b\tan(c+dx))^2}{2d}\\&\quad+\frac{B(a+b\tan(c+dx))^3}{3d}+\frac{C(a+b\tan(c+dx))^4}{4bd}\end{aligned}$$

```
[In] Int[(a + b*Tan[c + d*x])^3*(B*Tan[c + d*x] + C*Tan[c + d*x]^2),x]
```

```
[Out] -((3*a^2*b*B - b^3*B + a^3*C - 3*a*b^2*C)*x) - ((a^3*B - 3*a*b^2*B - 3*a^2*
b*C + b^3*C)*Log[Cos[c + d*x]])/d + (b*(a^2*B - b^2*B - 2*a*b*C)*Tan[c + d*
x])/d + ((a*B - b*C)*(a + b*Tan[c + d*x])^2)/(2*d) + (B*(a + b*Tan[c + d*x]
)^3)/(3*d) + (C*(a + b*Tan[c + d*x])^4)/(4*b*d)
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3606

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
 f*x], x], x] + Simp[b*d*(Tan[e + f*x]/f), x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3609

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[d*((a + b*Tan[e + f*x])^m/(f*m)), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \frac{C(a+b\tan(c+dx))^4}{4bd} + \int (a+b\tan(c+dx))^3(-C+B\tan(c+dx))\,dx \\
&= \frac{B(a+b\tan(c+dx))^3}{3d} + \frac{C(a+b\tan(c+dx))^4}{4bd} \\
&\quad + \int (a+b\tan(c+dx))^2(-bB-aC+(aB-bC)\tan(c+dx))\,dx \\
&= \frac{(aB-bC)(a+b\tan(c+dx))^2}{2d} + \frac{B(a+b\tan(c+dx))^3}{3d} + \frac{C(a+b\tan(c+dx))^4}{4bd} \\
&\quad + \int (a+b\tan(c+dx))\left(-2abB-a^2C+b^2C+\left(a^2B-b^2B-2abC\right)\tan(c+dx)\right)\,dx
\end{aligned}$$

$$= -\left(\left(3a^2bB - b^3B + a^3C - 3ab^2C\right)x\right) + \frac{b(a^2B - b^2B - 2abC)\tan(c+dx)}{d} + \frac{(aB - bC)(a + b\tan(c+dx))^2}{2d} + \frac{B(a + b\tan(c+dx))^3}{3d} + \frac{C(a + b\tan(c+dx))^4}{4bd} + \left(a^3B - 3ab^2B - 3a^2bC + b^3C\right)\int \tan(c+dx)\,dx$$

$$= -\left(\left(3a^2bB - b^3B + a^3C - 3ab^2C\right)x\right) - \frac{(a^3B - 3ab^2B - 3a^2bC + b^3C)\log(\cos(c+dx))}{d} + \frac{b(a^2B - b^2B - 2abC)\tan(c+dx)}{d} + \frac{(aB - bC)(a + b\tan(c+dx))^2}{2d} + \frac{B(a + b\tan(c+dx))^3}{3d} + \frac{C(a + b\tan(c+dx))^4}{4bd}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 1.76 (sec) , antiderivative size = 209, normalized size of antiderivative = 1.27

$$\int (a + b\tan(c+dx))^3 \left(B\tan(c+dx) + C\tan^2(c+dx)\right)\,dx$$

$$= \frac{-6i(a+ib)^4B\log(i - \tan(c+dx)) + 6i(a-ib)^4B\log(i + \tan(c+dx)) - 12b^2(-6a^2 + b^2)\,B\tan(c+dx}{}$$

```
[In] Integrate[(a + b*Tan[c + d*x])^3*(B*Tan[c + d*x] + C*Tan[c + d*x]^2),x]
```

```
[Out] ((-6*I)*(a + I*b)^4*B*Log[I - Tan[c + d*x]] + (6*I)*(a - I*b)^4*B*Log[I + T
an[c + d*x]] - 12*b^2*(-6*a^2 + b^2)*B*Tan[c + d*x] + 24*a*b^3*B*Tan[c + d*
x]^2 + 4*b^4*B*Tan[c + d*x]^3 + 3*C*(a + b*Tan[c + d*x])^4 - 6*(a*B + b*C)*
((I*a - b)^3*Log[I - Tan[c + d*x]] - (I*a + b)^3*Log[I + Tan[c + d*x]] + 6*
a*b^2*Tan[c + d*x] + b^3*Tan[c + d*x]^2))/(12*b*d)
```

Maple [A] (verified)

Time = 0.11 (sec) , antiderivative size = 180, normalized size of antiderivative = 1.09

method	result
norman	$\left(-3B\,a^{2}b+B\,b^{3}-C\,a^{3}+3Ca\,b^{2}\right)x+\frac{\left(3B\,a^{2}b-B\,b^{3}+C\,a^{3}-3Ca\,b^{2}\right)\tan(dx+c)}{d}+\frac{C\,b^{3}\tan(dx+c)^{4}}{4d}+$
parts	$\frac{\left(B\,b^{3}+3Ca\,b^{2}\right)\left(\frac{\tan(dx+c)^{3}}{3}-\tan(dx+c)+\arctan(\tan(dx+c))\right)}{d}+\frac{\left(3Ba\,b^{2}+3C\,a^{2}b\right)\left(\frac{\tan(dx+c)^{2}}{2}-\frac{\ln\left(1+\tan(dx+c)^{2}\right)}{2}\right)}{d}$
derivativedivides	$\frac{C\,b^{3}\tan(dx+c)^{4}}{4}+\frac{B\,b^{3}\tan(dx+c)^{3}}{3}+Ca\,b^{2}\tan(dx+c)^{3}+\frac{3Ba\,b^{2}\tan(dx+c)^{2}}{2}+\frac{3C\,a^{2}b\tan(dx+c)^{2}}{2}-\frac{C\,b^{3}\tan(dx+c)^{2}}{2}+3B\,a^{2}b\tan$
default	$\frac{C\,b^{3}\tan(dx+c)^{4}}{4}+\frac{B\,b^{3}\tan(dx+c)^{3}}{3}+Ca\,b^{2}\tan(dx+c)^{3}+\frac{3Ba\,b^{2}\tan(dx+c)^{2}}{2}+\frac{3C\,a^{2}b\tan(dx+c)^{2}}{2}-\frac{C\,b^{3}\tan(dx+c)^{2}}{2}+3B\,a^{2}b\tan$
parallelrisch	$\frac{3C\,b^{3}\tan(dx+c)^{4}+4B\,b^{3}\tan(dx+c)^{3}+12Ca\,b^{2}\tan(dx+c)^{3}-36B\,a^{2}bdx+12B\,b^{3}dx+18Ba\,b^{2}\tan(dx+c)^{2}-12C\,a^{3}dx+36}{}$
risch	$-3iBa\,b^{2}x+iB\,a^{3}x+iC\,b^{3}x-\frac{6iBa\,b^{2}c}{d}-3B\,a^{2}bx+B\,b^{3}x-C\,a^{3}x+3Ca\,b^{2}x-\frac{6iC\,a^{2}bc}{d}$

[In] int((a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,method=_RETURNVERBOS E)

[Out] (-3*B*a^2*b+B*b^3-C*a^3+3*C*a*b^2)*x+(3*B*a^2*b-B*b^3+C*a^3-3*C*a*b^2)/d*ta n(d*x+c)+1/4*C*b^3/d*tan(d*x+c)^4+1/2*b*(3*B*a*b+3*C*a^2-C*b^2)/d*tan(d*x+c)^2+1/3*b^2*(B*b+3*C*a)/d*tan(d*x+c)^3+1/2*(B*a^3-3*B*a*b^2-3*C*a^2*b+C*b^3)/d*ln(1+tan(d*x+c)^2)

Fricas [A] (verification not implemented)

none

Time = 0.26 (sec) , antiderivative size = 178, normalized size of antiderivative = 1.08

$$\int(a+b\tan(c+dx))^{3}\left(B\tan(c+dx)+C\tan^{2}(c+dx)\right)\,dx$$
$$=\frac{3\,Cb^{3}\tan\left(dx+c\right)^{4}+4\left(3\,Cab^{2}+Bb^{3}\right)\tan\left(dx+c\right)^{3}-12\left(Ca^{3}+3\,Ba^{2}b-3\,Cab^{2}-Bb^{3}\right)dx+6\left(3\,Ca^{2}b+\right.}{}$$

[In] integrate((a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, algorithm="fr icas")

[Out] 1/12*(3*C*b^3*tan(d*x + c)^4 + 4*(3*C*a*b^2 + B*b^3)*tan(d*x + c)^3 - 12*(C *a^3 + 3*B*a^2*b - 3*C*a*b^2 - B*b^3)*d*x + 6*(3*C*a^2*b + 3*B*a*b^2 - C*b^ 3)*tan(d*x + c)^2 - 6*(B*a^3 - 3*C*a^2*b - 3*B*a*b^2 + C*b^3)*log(1/(tan(d* x + c)^2 + 1)) + 12*(C*a^3 + 3*B*a^2*b - 3*C*a*b^2 - B*b^3)*tan(d*x + c))/d

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 313 vs. 2(151) = 302.

Time = 0.17 (sec) , antiderivative size = 313, normalized size of antiderivative = 1.90

$$\int (a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\begin{cases}\frac{Ba^3\log\left(\tan^2(c+dx)+1\right)}{2d}-3Ba^2bx+\frac{3Ba^2b\tan(c+dx)}{d}-\frac{3Bab^2\log\left(\tan^2(c+dx)+1\right)}{2d}+\frac{3Bab^2\tan^2(c+dx)}{2d}+Bb^3x+\frac{Bb^3\tan\end{cases}$$
$$x(a+b\tan(c))^3\left(B\tan(c)+C\tan^2(c)\right)$$

```
[In] integrate((a+b*tan(d*x+c))**3*(B*tan(d*x+c)+C*tan(d*x+c)**2),x)
```

```
[Out] Piecewise((B*a**3*log(tan(c + d*x)**2 + 1)/(2*d) - 3*B*a**2*b*x + 3*B*a**2*
b*tan(c + d*x)/d - 3*B*a*b**2*log(tan(c + d*x)**2 + 1)/(2*d) + 3*B*a*b**2*t
an(c + d*x)**2/(2*d) + B*b**3*x + B*b**3*tan(c + d*x)**3/(3*d) - B*b**3*tan
(c + d*x)/d - C*a**3*x + C*a**3*tan(c + d*x)/d - 3*C*a**2*b*log(tan(c + d*x
)**2 + 1)/(2*d) + 3*C*a**2*b*tan(c + d*x)**2/(2*d) + 3*C*a*b**2*x + C*a*b**
2*tan(c + d*x)**3/d - 3*C*a*b**2*tan(c + d*x)/d + C*b**3*log(tan(c + d*x)**
2 + 1)/(2*d) + C*b**3*tan(c + d*x)**4/(4*d) - C*b**3*tan(c + d*x)**2/(2*d),
 Ne(d, 0)), (x*(a + b*tan(c))**3*(B*tan(c) + C*tan(c)**2), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.42 (sec) , antiderivative size = 179, normalized size of antiderivative = 1.08

$$\int (a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{3\,Cb^3\tan(dx+c)^4+4\,(3\,Cab^2+Bb^3)\tan(dx+c)^3+6\,(3\,Ca^2b+3\,Bab^2-Cb^3)\tan(dx+c)^2-12\,(Ca^3}{}$$

```
[In] integrate((a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, algorithm="ma
xima")
```

```
[Out] 1/12*(3*C*b^3*tan(d*x + c)^4 + 4*(3*C*a*b^2 + B*b^3)*tan(d*x + c)^3 + 6*(3*
C*a^2*b + 3*B*a*b^2 - C*b^3)*tan(d*x + c)^2 - 12*(C*a^3 + 3*B*a^2*b - 3*C*a
*b^2 - B*b^3)*(d*x + c) + 6*(B*a^3 - 3*C*a^2*b - 3*B*a*b^2 + C*b^3)*log(tan
(d*x + c)^2 + 1) + 12*(C*a^3 + 3*B*a^2*b - 3*C*a*b^2 - B*b^3)*tan(d*x + c))
/d
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 2670 vs. 2(159) = 318.

Time = 2.37 (sec) , antiderivative size = 2670, normalized size of antiderivative = 16.18

$$\int (a+b\tan(c+dx))^3 \left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx=\text{Too large to display}$$

```
[In] integrate((a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, algorithm="gi
ac")
```

```
[Out] -1/12*(12*C*a^3*d*x*tan(d*x)^4*tan(c)^4 + 36*B*a^2*b*d*x*tan(d*x)^4*tan(c)^
4 - 36*C*a*b^2*d*x*tan(d*x)^4*tan(c)^4 - 12*B*b^3*d*x*tan(d*x)^4*tan(c)^4 +
 6*B*a^3*log(4*(tan(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(d*x)^2*ta
n(c)^2 + tan(d*x)^2 + tan(c)^2 + 1))*tan(d*x)^4*tan(c)^4 - 18*C*a^2*b*log(4
*(tan(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(d*x)^2*tan(c)^2 + tan(d
*x)^2 + tan(c)^2 + 1))*tan(d*x)^4*tan(c)^4 - 18*B*a*b^2*log(4*(tan(d*x)^2*t
an(c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(d*x)^2*tan(c)^2 + tan(d*x)^2 + tan(c)
^2 + 1))*tan(d*x)^4*tan(c)^4 + 6*C*b^3*log(4*(tan(d*x)^2*tan(c)^2 - 2*tan(d
*x)*tan(c) + 1)/(tan(d*x)^2*tan(c)^2 + tan(d*x)^2 + tan(c)^2 + 1))*tan(d*x)
^4*tan(c)^4 - 48*C*a^3*d*x*tan(d*x)^3*tan(c)^3 - 144*B*a^2*b*d*x*tan(d*x)^3
*tan(c)^3 + 144*C*a*b^2*d*x*tan(d*x)^3*tan(c)^3 + 48*B*b^3*d*x*tan(d*x)^3*t
an(c)^3 - 18*C*a^2*b*tan(d*x)^4*tan(c)^4 - 18*B*a*b^2*tan(d*x)^4*tan(c)^4 +
 9*C*b^3*tan(d*x)^4*tan(c)^4 - 24*B*a^3*log(4*(tan(d*x)^2*tan(c)^2 - 2*tan(
d*x)*tan(c) + 1)/(tan(d*x)^2*tan(c)^2 + tan(d*x)^2 + tan(c)^2 + 1))*tan(d*x
)^3*tan(c)^3 + 72*C*a^2*b*log(4*(tan(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(c) +
1)/(tan(d*x)^2*tan(c)^2 + tan(d*x)^2 + tan(c)^2 + 1))*tan(d*x)^3*tan(c)^3 +
 72*B*a*b^2*log(4*(tan(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(d*x)^2
*tan(c)^2 + tan(d*x)^2 + tan(c)^2 + 1))*tan(d*x)^3*tan(c)^3 - 24*C*b^3*log(
4*(tan(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(d*x)^2*tan(c)^2 + tan(
d*x)^2 + tan(c)^2 + 1))*tan(d*x)^3*tan(c)^3 + 12*C*a^3*tan(d*x)^4*tan(c)^3
+ 36*B*a^2*b*tan(d*x)^4*tan(c)^3 - 36*C*a*b^2*tan(d*x)^4*tan(c)^3 - 12*B*b^
3*tan(d*x)^4*tan(c)^3 + 12*C*a^3*tan(d*x)^3*tan(c)^4 + 36*B*a^2*b*tan(d*x)^
3*tan(c)^4 - 36*C*a*b^2*tan(d*x)^3*tan(c)^4 - 12*B*b^3*tan(d*x)^3*tan(c)^4
+ 72*C*a^3*d*x*tan(d*x)^2*tan(c)^2 + 216*B*a^2*b*d*x*tan(d*x)^2*tan(c)^2 -
216*C*a*b^2*d*x*tan(d*x)^2*tan(c)^2 - 72*B*b^3*d*x*tan(d*x)^2*tan(c)^2 - 18
*C*a^2*b*tan(d*x)^4*tan(c)^2 - 18*B*a*b^2*tan(d*x)^4*tan(c)^2 + 6*C*b^3*tan
(d*x)^4*tan(c)^2 + 36*C*a^2*b*tan(d*x)^3*tan(c)^3 + 36*B*a*b^2*tan(d*x)^3*t
an(c)^3 - 24*C*b^3*tan(d*x)^3*tan(c)^3 - 18*C*a^2*b*tan(d*x)^2*tan(c)^4 - 1
8*B*a*b^2*tan(d*x)^2*tan(c)^4 + 6*C*b^3*tan(d*x)^2*tan(c)^4 + 12*C*a*b^2*ta
n(d*x)^4*tan(c) + 4*B*b^3*tan(d*x)^4*tan(c) + 36*B*a^3*log(4*(tan(d*x)^2*ta
n(c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(d*x)^2*tan(c)^2 + tan(d*x)^2 + tan(c)^
2 + 1))*tan(d*x)^2*tan(c)^2 - 108*C*a^2*b*log(4*(tan(d*x)^2*tan(c)^2 - 2*ta
n(d*x)*tan(c) + 1)/(tan(d*x)^2*tan(c)^2 + tan(d*x)^2 + tan(c)^2 + 1))*tan(d
*x)^2*tan(c)^2 - 108*B*a*b^2*log(4*(tan(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(c)
```

```
 + 1)/(tan(d*x)^2*tan(c)^2 + tan(d*x)^2 + tan(c)^2 + 1))*tan(d*x)^2*tan(c)^
2 + 36*C*b^3*log(4*(tan(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(d*x)^
2*tan(c)^2 + tan(d*x)^2 + tan(c)^2 + 1))*tan(d*x)^2*tan(c)^2 - 36*C*a^3*tan
(d*x)^3*tan(c)^2 - 108*B*a^2*b*tan(d*x)^3*tan(c)^2 + 144*C*a*b^2*tan(d*x)^3
*tan(c)^2 + 48*B*b^3*tan(d*x)^3*tan(c)^2 - 36*C*a^3*tan(d*x)^2*tan(c)^3 - 1
08*B*a^2*b*tan(d*x)^2*tan(c)^3 + 144*C*a*b^2*tan(d*x)^2*tan(c)^3 + 48*B*b^3
*tan(d*x)^2*tan(c)^3 + 12*C*a*b^2*tan(d*x)*tan(c)^4 + 4*B*b^3*tan(d*x)*tan(
c)^4 - 3*C*b^3*tan(d*x)^4 - 48*C*a^3*d*x*tan(d*x)*tan(c) - 144*B*a^2*b*d*x*
tan(d*x)*tan(c) + 144*C*a*b^2*d*x*tan(d*x)*tan(c) + 48*B*b^3*d*x*tan(d*x)*t
an(c) + 36*C*a^2*b*tan(d*x)^3*tan(c) + 36*B*a*b^2*tan(d*x)^3*tan(c) - 24*C*
b^3*tan(d*x)^3*tan(c) - 36*C*a^2*b*tan(d*x)^2*tan(c)^2 - 36*B*a*b^2*tan(d*x
)^2*tan(c)^2 + 12*C*b^3*tan(d*x)^2*tan(c)^2 + 36*C*a^2*b*tan(d*x)*tan(c)^3
+ 36*B*a*b^2*tan(d*x)*tan(c)^3 - 24*C*b^3*tan(d*x)*tan(c)^3 - 3*C*b^3*tan(c
)^4 - 12*C*a*b^2*tan(d*x)^3 - 4*B*b^3*tan(d*x)^3 - 24*B*a^3*log(4*(tan(d*x)
^2*tan(c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(d*x)^2*tan(c)^2 + tan(d*x)^2 + ta
n(c)^2 + 1))*tan(d*x)*tan(c) + 72*C*a^2*b*log(4*(tan(d*x)^2*tan(c)^2 - 2*ta
n(d*x)*tan(c) + 1)/(tan(d*x)^2*tan(c)^2 + tan(d*x)^2 + tan(c)^2 + 1))*tan(d
*x)*tan(c) + 72*B*a*b^2*log(4*(tan(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(c) + 1)
/(tan(d*x)^2*tan(c)^2 + tan(d*x)^2 + tan(c)^2 + 1))*tan(d*x)*tan(c) - 24*C*
b^3*log(4*(tan(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(d*x)^2*tan(c)^
2 + tan(d*x)^2 + tan(c)^2 + 1))*tan(d*x)*tan(c) + 36*C*a^3*tan(d*x)^2*tan(c
) + 108*B*a^2*b*tan(d*x)^2*tan(c) - 144*C*a*b^2*tan(d*x)^2*tan(c) - 48*B*b^
3*tan(d*x)^2*tan(c) + 36*C*a^3*tan(d*x)*tan(c)^2 + 108*B*a^2*b*tan(d*x)*tan
(c)^2 - 144*C*a*b^2*tan(d*x)*tan(c)^2 - 48*B*b^3*tan(d*x)*tan(c)^2 - 12*C*a
*b^2*tan(c)^3 - 4*B*b^3*tan(c)^3 + 12*C*a^3*d*x + 36*B*a^2*b*d*x - 36*C*a*b
^2*d*x - 12*B*b^3*d*x - 18*C*a^2*b*tan(d*x)^2 - 18*B*a*b^2*tan(d*x)^2 + 6*C
*b^3*tan(d*x)^2 + 36*C*a^2*b*tan(d*x)*tan(c) + 36*B*a*b^2*tan(d*x)*tan(c) -
 24*C*b^3*tan(d*x)*tan(c) - 18*C*a^2*b*tan(c)^2 - 18*B*a*b^2*tan(c)^2 + 6*C
*b^3*tan(c)^2 + 6*B*a^3*log(4*(tan(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(c) + 1)
/(tan(d*x)^2*tan(c)^2 + tan(d*x)^2 + tan(c)^2 + 1)) - 18*C*a^2*b*log(4*(tan
(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(d*x)^2*tan(c)^2 + tan(d*x)^2
 + tan(c)^2 + 1)) - 18*B*a*b^2*log(4*(tan(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(
c) + 1)/(tan(d*x)^2*tan(c)^2 + tan(d*x)^2 + tan(c)^2 + 1)) + 6*C*b^3*log(4*
(tan(d*x)^2*tan(c)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(d*x)^2*tan(c)^2 + tan(d*
x)^2 + tan(c)^2 + 1)) - 12*C*a^3*tan(d*x) - 36*B*a^2*b*tan(d*x) + 36*C*a*b^
2*tan(d*x) + 12*B*b^3*tan(d*x) - 12*C*a^3*tan(c) - 36*B*a^2*b*tan(c) + 36*C
*a*b^2*tan(c) + 12*B*b^3*tan(c) - 18*C*a^2*b - 18*B*a*b^2 + 9*C*b^3)/(d*tan
(d*x)^4*tan(c)^4 - 4*d*tan(d*x)^3*tan(c)^3 + 6*d*tan(d*x)^2*tan(c)^2 - 4*d*
tan(d*x)*tan(c) + d)
```

Mupad [B] (verification not implemented)

Time = 8.54 (sec) , antiderivative size = 181, normalized size of antiderivative = 1.10

$$\int (a+b\tan(c+dx))^3 \left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$= x\left(-C\,a^3-3\,B\,a^2\,b+3\,C\,a\,b^2+B\,b^3\right)-\frac{\tan(c+d\,x)^2\,\left(\frac{C\,b^3}{2}-\frac{3\,a\,b\,(B\,b+C\,a)}{2}\right)}{d}$$
$$-\frac{\tan(c+d\,x)\,\left(-C\,a^3-3\,B\,a^2\,b+3\,C\,a\,b^2+B\,b^3\right)}{d}$$
$$+\frac{\ln\left(\tan(c+d\,x)^2+1\right)\,\left(\frac{B\,a^3}{2}-\frac{3\,C\,a^2\,b}{2}-\frac{3\,B\,a\,b^2}{2}+\frac{C\,b^3}{2}\right)}{d}$$
$$+\frac{\tan(c+d\,x)^3\,\left(\frac{B\,b^3}{3}+C\,a\,b^2\right)}{d}+\frac{C\,b^3\,\tan(c+d\,x)^4}{4\,d}$$

```
[In] int((B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))^3,x)
```

```
[Out] x*(B*b^3 - C*a^3 - 3*B*a^2*b + 3*C*a*b^2) - (tan(c + d*x)^2*((C*b^3)/2 - (3
*a*b*(B*b + C*a))/2))/d - (tan(c + d*x)*(B*b^3 - C*a^3 - 3*B*a^2*b + 3*C*a*
b^2))/d + (log(tan(c + d*x)^2 + 1)*((B*a^3)/2 + (C*b^3)/2 - (3*B*a*b^2)/2 -
 (3*C*a^2*b)/2))/d + (tan(c + d*x)^3*((B*b^3)/3 + C*a*b^2))/d + (C*b^3*tan(
c + d*x)^4)/(4*d)
```

3.18 $\int \cot(c+dx)(a+b\tan(c+dx))^3 \left(B\tan(c+dx)+C\tan^2(c+dx)\right) dx$

Optimal result . . . 177
Rubi [A] (verified) . . . 177
Mathematica [C] (verified) . . . 179
Maple [A] (verified) . . . 179
Fricas [A] (verification not implemented) . . . 180
Sympy [A] (verification not implemented) . . . 181
Maxima [A] (verification not implemented) . . . 181
Giac [A] (verification not implemented) . . . 182
Mupad [B] (verification not implemented) . . . 182

Optimal result

Integrand size = 38, antiderivative size = 140

$$\begin{aligned}&\int \cot(c+dx)(a+b\tan(c+dx))^3 \left(B\tan(c+dx)+C\tan^2(c+dx)\right) dx\\ &= \left(a^3B-3ab^2B-3a^2bC+b^3C\right)x-\frac{(3a^2bB-b^3B+a^3C-3ab^2C)\log(\cos(c+dx))}{d}\\ &\quad+\frac{b(2abB+a^2C-b^2C)\tan(c+dx)}{d}\\ &\quad+\frac{(bB+aC)(a+b\tan(c+dx))^2}{2d}+\frac{C(a+b\tan(c+dx))^3}{3d}\end{aligned}$$

```
[Out] (B*a^3-3*B*a*b^2-3*C*a^2*b+C*b^3)*x-(3*B*a^2*b-B*b^3+C*a^3-3*C*a*b^2)*ln(co
s(d*x+c))/d+b*(2*B*a*b+C*a^2-C*b^2)*tan(d*x+c)/d+1/2*(B*b+C*a)*(a+b*tan(d*x
+c))^2/d+1/3*C*(a+b*tan(d*x+c))^3/d
```

Rubi [A] (verified)

Time = 0.24 (sec) , antiderivative size = 140, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 4, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.105, Rules used = {3713, 3609, 3606, 3556}

$$\begin{aligned}&\int \cot(c+dx)(a+b\tan(c+dx))^3 \left(B\tan(c+dx)+C\tan^2(c+dx)\right) dx\\ &= \frac{b(a^2C+2abB-b^2C)\tan(c+dx)}{d}-\frac{(a^3C+3a^2bB-3ab^2C-b^3B)\log(\cos(c+dx))}{d}\\ &\quad+x\left(a^3B-3a^2bC-3ab^2B+b^3C\right)\\ &\quad+\frac{(aC+bB)(a+b\tan(c+dx))^2}{2d}+\frac{C(a+b\tan(c+dx))^3}{3d}\end{aligned}$$

```
[In] Int[Cot[c + d*x]*(a + b*Tan[c + d*x])^3*(B*Tan[c + d*x] + C*Tan[c + d*x]^2)
,x]
```

```
[Out] (a^3*B - 3*a*b^2*B - 3*a^2*b*C + b^3*C)*x - ((3*a^2*b*B - b^3*B + a^3*C - 3
*a*b^2*C)*Log[Cos[c + d*x]])/d + (b*(2*a*b*B + a^2*C - b^2*C)*Tan[c + d*x])
/d + ((b*B + a*C)*(a + b*Tan[c + d*x])^2)/(2*d) + (C*(a + b*Tan[c + d*x])^3
)/(3*d)
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3606

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
 f*x], x], x] + Simp[b*d*(Tan[e + f*x]/f), x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3609

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[d*((a + b*Tan[e + f*x])^m/(f*m)), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int (a+b\tan(c+dx))^3(B+C\tan(c+dx))\,dx \\
&= \frac{C(a+b\tan(c+dx))^3}{3d}+\int (a+b\tan(c+dx))^2(aB-bC+(bB+aC)\tan(c+dx))\,dx \\
&= \frac{(bB+aC)(a+b\tan(c+dx))^2}{2d}+\frac{C(a+b\tan(c+dx))^3}{3d} \\
&\quad +\int (a+b\tan(c+dx))\left(a^2B-b^2B-2abC+\left(2abB+a^2C-b^2C\right)\tan(c+dx)\right)\,dx
\end{aligned}$$

$$= \left(a^3B - 3ab^2B - 3a^2bC + b^3C\right)x + \frac{b(2abB + a^2C - b^2C)\tan(c+dx)}{d} + \frac{(bB + aC)(a + b\tan(c+dx))^2}{2d} + \frac{C(a + b\tan(c+dx))^3}{3d} + \left(3a^2bB - b^3B + a^3C - 3ab^2C\right)\int \tan(c+dx)\,dx$$

$$= \left(a^3B - 3ab^2B - 3a^2bC + b^3C\right)x - \frac{(3a^2bB - b^3B + a^3C - 3ab^2C)\log(\cos(c+dx))}{d} + \frac{b(2abB + a^2C - b^2C)\tan(c+dx)}{d} + \frac{(bB + aC)(a + b\tan(c+dx))^2}{2d} + \frac{C(a + b\tan(c+dx))^3}{3d}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 1.13 (sec) , antiderivative size = 130, normalized size of antiderivative = 0.93

$$\int \cot(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$= \frac{3(a+ib)^3(-iB+C)\log(i-\tan(c+dx)) + 3(a-ib)^3(iB+C)\log(i+\tan(c+dx)) + 6b(3abB+3a^2C}{6d}$$

```
[In] Integrate[Cot[c + d*x]*(a + b*Tan[c + d*x])^3*(B*Tan[c + d*x] + C*Tan[c + d
*x]^2),x]
```

```
[Out] (3*(a + I*b)^3*((-I)*B + C)*Log[I - Tan[c + d*x]] + 3*(a - I*b)^3*(I*B + C)
*Log[I + Tan[c + d*x]] + 6*b*(3*a*b*B + 3*a^2*C - b^2*C)*Tan[c + d*x] + 3*b
^2*(b*B + 3*a*C)*Tan[c + d*x]^2 + 2*b^3*C*Tan[c + d*x]^3)/(6*d)
```

Maple [A] (verified)

Time = 0.28 (sec) , antiderivative size = 139, normalized size of antiderivative = 0.99

method	result
parallelrisch	$\frac{3(3B\,a^2b-B\,b^3+C\,a^3-3Ca\,b^2)\ln\left(\sec(dx+c)^2\right)+2C\,b^3\tan(dx+c)^3+3(B\,b^3+3Ca\,b^2)\tan(dx+c)^2+6(3Ba\,b^2+3C\,a^2b-C}{6d}$
norman	$(B\,a^3-3Ba\,b^2-3C\,a^2b+C\,b^3)\,x+\frac{b(3Bab+3C\,a^2-C\,b^2)\tan(dx+c)}{d}+\frac{C\,b^3\tan(dx+c)^3}{3d}+\frac{b^2(Bb+3C}{}$
derivativedivides	$\frac{\frac{C\,b^3\tan(dx+c)^3}{3}+\frac{B\,b^3\tan(dx+c)^2}{2}+\frac{3Ca\,b^2\tan(dx+c)^2}{2}+3Ba\,b^2\tan(dx+c)+3C\,a^2b\tan(dx+c)-C\,b^3\tan(dx+c)+\frac{(3B\,a^2b-B}{}}{d}$
default	$\frac{\frac{C\,b^3\tan(dx+c)^3}{3}+\frac{B\,b^3\tan(dx+c)^2}{2}+\frac{3Ca\,b^2\tan(dx+c)^2}{2}+3Ba\,b^2\tan(dx+c)+3C\,a^2b\tan(dx+c)-C\,b^3\tan(dx+c)+\frac{(3B\,a^2b-B}{}}{d}$
risch	$B\,a^3x-3Ba\,b^2x-3C\,a^2bx+C\,b^3x-\frac{2iB\,b^3c}{d}+\frac{2iC\,a^3c}{d}-iB\,b^3x+iC\,a^3x+\frac{2ib(-3iB\,b^2e^{4i(c}}{}$

[In] int(cot(d*x+c)*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,method=_RETURNVERBOSE)

[Out] 1/6*(3*(3*B*a^2*b-B*b^3+C*a^3-3*C*a*b^2)*ln(sec(d*x+c)^2)+2*C*b^3*tan(d*x+c)^3+3*(B*b^3+3*C*a*b^2)*tan(d*x+c)^2+6*(3*B*a*b^2+3*C*a^2*b-C*b^3)*tan(d*x+c)+6*d*x*(B*a^3-3*B*a*b^2-3*C*a^2*b+C*b^3))/d

Fricas [A] (verification not implemented)

none

Time = 0.25 (sec) , antiderivative size = 142, normalized size of antiderivative = 1.01

$$\int\cot(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{2\,Cb^3\tan{(dx+c)}^3+6\,(Ba^3-3\,Ca^2b-3\,Bab^2+Cb^3)dx+3\,(3\,Cab^2+Bb^3)\tan{(dx+c)}^2-3\,(Ca^3+3\,B}{6\,d}$$

[In] integrate(cot(d*x+c)*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, algorithm="fricas")

[Out] 1/6*(2*C*b^3*tan(d*x + c)^3 + 6*(B*a^3 - 3*C*a^2*b - 3*B*a*b^2 + C*b^3)*d*x + 3*(3*C*a*b^2 + B*b^3)*tan(d*x + c)^2 - 3*(C*a^3 + 3*B*a^2*b - 3*C*a*b^2 - B*b^3)*log(1/(tan(d*x + c)^2 + 1)) + 6*(3*C*a^2*b + 3*B*a*b^2 - C*b^3)*tan(d*x + c))/d

Sympy [A] (verification not implemented)

Time = 0.77 (sec) , antiderivative size = 248, normalized size of antiderivative = 1.77

$$\int \cot(c+dx)(a+b\tan(c+dx))^3 \left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$= \begin{cases} Ba^3x + \frac{3Ba^2b\log\left(\tan^2(c+dx)+1\right)}{2d} - 3Bab^2x + \frac{3Bab^2\tan(c+dx)}{d} - \frac{Bb^3\log\left(\tan^2(c+dx)+1\right)}{2d} + \frac{Bb^3\tan^2(c+dx)}{2d} + \frac{Ca^3\log(\text{t}}{} \\ x(a+b\tan(c))^3\left(B\tan(c)+C\tan^2(c)\right)\cot(c) \end{cases}$$

```
[In] integrate(cot(d*x+c)*(a+b*tan(d*x+c))**3*(B*tan(d*x+c)+C*tan(d*x+c)**2),x)
```

```
[Out] Piecewise((B*a**3*x + 3*B*a**2*b*log(tan(c + d*x)**2 + 1)/(2*d) - 3*B*a*b**
2*x + 3*B*a*b**2*tan(c + d*x)/d - B*b**3*log(tan(c + d*x)**2 + 1)/(2*d) + B
*b**3*tan(c + d*x)**2/(2*d) + C*a**3*log(tan(c + d*x)**2 + 1)/(2*d) - 3*C*a
**2*b*x + 3*C*a**2*b*tan(c + d*x)/d - 3*C*a*b**2*log(tan(c + d*x)**2 + 1)/(
2*d) + 3*C*a*b**2*tan(c + d*x)**2/(2*d) + C*b**3*x + C*b**3*tan(c + d*x)**3
/(3*d) - C*b**3*tan(c + d*x)/d, Ne(d, 0)), (x*(a + b*tan(c))**3*(B*tan(c) +
 C*tan(c)**2)*cot(c), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.33 (sec) , antiderivative size = 143, normalized size of antiderivative = 1.02

$$\int \cot(c+dx)(a+b\tan(c+dx))^3 \left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$= \frac{2\,Cb^3\tan(dx+c)^3 + 3\,(3\,Cab^2+Bb^3)\tan(dx+c)^2 + 6\,(Ba^3-3\,Ca^2b-3\,Bab^2+Cb^3)(dx+c) + 3\,(Ca}{6\,d}$$

```
[In] integrate(cot(d*x+c)*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="maxima")
```

```
[Out] 1/6*(2*C*b^3*tan(d*x + c)^3 + 3*(3*C*a*b^2 + B*b^3)*tan(d*x + c)^2 + 6*(B*a
^3 - 3*C*a^2*b - 3*B*a*b^2 + C*b^3)*(d*x + c) + 3*(C*a^3 + 3*B*a^2*b - 3*C*
a*b^2 - B*b^3)*log(tan(d*x + c)^2 + 1) + 6*(3*C*a^2*b + 3*B*a*b^2 - C*b^3)*
tan(d*x + c))/d
```

Giac [A] (verification not implemented)

none

Time = 1.53 (sec) , antiderivative size = 158, normalized size of antiderivative = 1.13

$$\int \cot(c+dx)(a+b\tan(c+dx))^3 \left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$= \frac{2\,Cb^3\tan\left(dx+c\right)^3+9\,Cab^2\tan\left(dx+c\right)^2+3\,Bb^3\tan\left(dx+c\right)^2+18\,Ca^2b\tan\left(dx+c\right)+18\,Bab^2\tan\left(dx+\right.}{}$$

```
[In] integrate(cot(d*x+c)*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="giac")
```

```
[Out] 1/6*(2*C*b^3*tan(d*x + c)^3 + 9*C*a*b^2*tan(d*x + c)^2 + 3*B*b^3*tan(d*x +
c)^2 + 18*C*a^2*b*tan(d*x + c) + 18*B*a*b^2*tan(d*x + c) - 6*C*b^3*tan(d*x
+ c) + 6*(B*a^3 - 3*C*a^2*b - 3*B*a*b^2 + C*b^3)*(d*x + c) + 3*(C*a^3 + 3*B
*a^2*b - 3*C*a*b^2 - B*b^3)*log(tan(d*x + c)^2 + 1))/d
```

Mupad [B] (verification not implemented)

Time = 8.72 (sec) , antiderivative size = 142, normalized size of antiderivative = 1.01

$$\int \cot(c+dx)(a+b\tan(c+dx))^3 \left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$= x\left(B\,a^3-3\,C\,a^2\,b-3\,B\,a\,b^2+C\,b^3\right) - \frac{\ln\left(\tan(c+d\,x)^2+1\right)\left(-\frac{C\,a^3}{2}-\frac{3\,B\,a^2\,b}{2}+\frac{3\,C\,a\,b^2}{2}+\frac{B\,b^3}{2}\right)}{d} + \frac{\tan(c+d\,x)^2\left(\frac{B\,b^3}{2}+\frac{3\,C\,a\,b^2}{2}\right)}{d} - \frac{\tan(c+d\,x)\,\left(C\,b^3-3\,a\,b\,(B\,b+C\,a)\right)}{d} + \frac{C\,b^3\tan(c+d\,x)^3}{3\,d}$$

```
[In] int(cot(c + d*x)*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))^3
,x)
```

```
[Out] x*(B*a^3 + C*b^3 - 3*B*a*b^2 - 3*C*a^2*b) - (log(tan(c + d*x)^2 + 1)*((B*b^
3)/2 - (C*a^3)/2 - (3*B*a^2*b)/2 + (3*C*a*b^2)/2))/d + (tan(c + d*x)^2*((B*
b^3)/2 + (3*C*a*b^2)/2))/d - (tan(c + d*x)*(C*b^3 - 3*a*b*(B*b + C*a)))/d +
 (C*b^3*tan(c + d*x)^3)/(3*d)
```

3.19 $\int \cot^2(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)dx$

Optimal result 183
Rubi [A] (verified) 183
Mathematica [C] (verified) 185
Maple [A] (verified) 186
Fricas [A] (verification not implemented) 186
Sympy [A] (verification not implemented) 187
Maxima [A] (verification not implemented) 187
Giac [A] (verification not implemented) 188
Mupad [B] (verification not implemented) 188

Optimal result

Integrand size = 40, antiderivative size = 117

$$\begin{aligned}&\int \cot^2(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx\\&=\left(3a^2bB-b^3B+a^3C-3ab^2C\right)x-\frac{b(3abB+3a^2C-b^2C)\log(\cos(c+dx))}{d}\\&\quad+\frac{a^3B\log(\sin(c+dx))}{d}+\frac{b^2(bB+2aC)\tan(c+dx)}{d}+\frac{bC(a+b\tan(c+dx))^2}{2d}\end{aligned}$$

```
[Out] (3*B*a^2*b-B*b^3+C*a^3-3*C*a*b^2)*x-b*(3*B*a*b+3*C*a^2-C*b^2)*ln(cos(d*x+c)
)/d+a^3*B*ln(sin(d*x+c))/d+b^2*(B*b+2*C*a)*tan(d*x+c)/d+1/2*b*C*(a+b*tan(d*
x+c))^2/d
```

Rubi [A] (verified)

Time = 0.39 (sec) , antiderivative size = 117, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.125, Rules used = {3713, 3688, 3718, 3705, 3556}

$$\begin{aligned}&\int \cot^2(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx\\&=\frac{a^3B\log(\sin(c+dx))}{d}-\frac{b(3a^2C+3abB-b^2C)\log(\cos(c+dx))}{d}\\&\quad+x\left(a^3C+3a^2bB-3ab^2C-b^3B\right)+\frac{b^2(2aC+bB)\tan(c+dx)}{d}+\frac{bC(a+b\tan(c+dx))^2}{2d}\end{aligned}$$

```
[In] Int[Cot[c + d*x]^2*(a + b*Tan[c + d*x])^3*(B*Tan[c + d*x] + C*Tan[c + d*x]^
2),x]
```

```
[Out] (3*a^2*b*B - b^3*B + a^3*C - 3*a*b^2*C)*x - (b*(3*a*b*B + 3*a^2*C - b^2*C)*
Log[Cos[c + d*x]])/d + (a^3*B*Log[Sin[c + d*x]])/d + (b^2*(b*B + 2*a*C)*Tan
[c + d*x])/d + (b*C*(a + b*Tan[c + d*x])^2)/(2*d)
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3688

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Si
mp[b*B*(a + b*Tan[e + f*x])^(m - 1)*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(m +
 n))), x] + Dist[1/(d*(m + n)), Int[(a + b*Tan[e + f*x])^(m - 2)*(c + d*Tan
[e + f*x])^n*Simp[a^2*A*d*(m + n) - b*B*(b*c*(m - 1) + a*d*(n + 1)) + d*(m
+ n)*(2*a*A*b + B*(a^2 - b^2))*Tan[e + f*x] - (b*B*(b*c - a*d)*(m - 1) - b*
(A*b + a*B)*d*(m + n))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e,
f, A, B, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2,
 0] && GtQ[m, 1] && (IntegerQ[m] || IntegersQ[2*m, 2*n]) &&  !(IGtQ[n, 1] &
& ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3705

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/tan[(e_.) + (f_.)*(x_)], x_Symbol] :> Simp[B*x, x] + (Dist[A, Int[1/Tan[e
 + f*x], x], x] + Dist[C, Int[Tan[e + f*x], x], x]) /; FreeQ[{e, f, A, B, C
}, x] && NeQ[A, C]
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
```

```
!LtQ[n, -1]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \cot(c+dx)(a+b\tan(c+dx))^3(B+C\tan(c+dx))\,dx \\
&= \frac{bC(a+b\tan(c+dx))^2}{2d}+\frac{1}{2}\int \cot(c+dx)(a+b\tan(c+dx))\left(2a^2B\right.\\
&\quad \left.+2\left(2abB+a^2C-b^2C\right)\tan(c+dx)+2b(bB+2aC)\tan^2(c+dx)\right)\,dx \\
&= \frac{b^2(bB+2aC)\tan(c+dx)}{d}+\frac{bC(a+b\tan(c+dx))^2}{2d}-\frac{1}{2}\int \cot(c+dx)\left(-2a^3B\right.\\
&\quad -2\left(3a^2bB-b^3B+a^3C-3ab^2C\right)\tan(c+dx)\\
&\quad \left.-2b\left(3abB+3a^2C-b^2C\right)\tan^2(c+dx)\right)\,dx \\
&= \left(3a^2bB-b^3B+a^3C-3ab^2C\right)x+\frac{b^2(bB+2aC)\tan(c+dx)}{d}\\
&\quad +\frac{bC(a+b\tan(c+dx))^2}{2d}+\left(a^3B\right)\int \cot(c+dx)\,dx\\
&\quad +\left(b\left(3abB+3a^2C-b^2C\right)\right)\int \tan(c+dx)\,dx \\
&= \left(3a^2bB-b^3B+a^3C-3ab^2C\right)x-\frac{b(3abB+3a^2C-b^2C)\log(\cos(c+dx))}{d}\\
&\quad +\frac{a^3B\log(\sin(c+dx))}{d}+\frac{b^2(bB+2aC)\tan(c+dx)}{d}+\frac{bC(a+b\tan(c+dx))^2}{2d}
\end{aligned}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 0.50 (sec) , antiderivative size = 113, normalized size of antiderivative = 0.97

$$\int \cot^2(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{-(a+ib)^3(B+iC)\log(i-\tan(c+dx))+2a^3B\log(\tan(c+dx))-(a-ib)^3(B-iC)\log(i+\tan(c+d$$
$$2d$$

```
[In] Integrate[Cot[c + d*x]^2*(a + b*Tan[c + d*x])^3*(B*Tan[c + d*x] + C*Tan[c +
 d*x]^2),x]
```

```
[Out] (-((a + I*b)^3*(B + I*C)*Log[I - Tan[c + d*x]]) + 2*a^3*B*Log[Tan[c + d*x]]
 - (a - I*b)^3*(B - I*C)*Log[I + Tan[c + d*x]] + 2*b^2*(b*B + 3*a*C)*Tan[c
+ d*x] + b^3*C*Tan[c + d*x]^2)/(2*d)
```

Maple [A] (verified)

Time = 0.26 (sec) , antiderivative size = 121, normalized size of antiderivative = 1.03

method	result
parallelrisch	$\frac{\left(-B\,a^{3}+3 B a\,b^{2}+3 C\,a^{2} b-C\,b^{3}\right) \ln\left(\sec(dx+c)^{2}\right)+2 B\,a^{3} \ln(\tan(dx+c))+C\,b^{3} \tan(dx+c)^{2}+\left(2 B\,b^{3}+6 C a\,b^{2}\right) \tan(dx+c)+}{2 d}$
derivativedivides	$\frac{\frac{C\,b^{3} \tan(dx+c)^{2}}{2}+B\,b^{3} \tan(dx+c)+3 C a\,b^{2} \tan(dx+c)+B\,a^{3} \ln(\tan(dx+c))+\frac{\left(-B\,a^{3}+3 B a\,b^{2}+3 C\,a^{2} b-C\,b^{3}\right) \ln\left(1+\tan(dx+c)^{2}\right.}{2}}{d}$
default	$\frac{\frac{C\,b^{3} \tan(dx+c)^{2}}{2}+B\,b^{3} \tan(dx+c)+3 C a\,b^{2} \tan(dx+c)+B\,a^{3} \ln(\tan(dx+c))+\frac{\left(-B\,a^{3}+3 B a\,b^{2}+3 C\,a^{2} b-C\,b^{3}\right) \ln\left(1+\tan(dx+c)^{2}\right.}{2}}{d}$
norman	$\frac{\left(3 B\,a^{2} b-B\,b^{3}+C\,a^{3}-3 C a\,b^{2}\right) x \tan(dx+c)+\frac{b^{2} (B b+3 C a) \tan(dx+c)^{2}}{d}+\frac{C\,b^{3} \tan(dx+c)^{3}}{2 d}}{\tan(dx+c)}+\frac{B\,a^{3} \ln(\tan(dx+c))}{d}-\frac{(B\,a^{3}-}{}$
risch	$\frac{6 i B a\,b^{2} c}{d}-i B\,a^{3} x+3 i C\,a^{2} b x+\frac{2 i b^{2} \left(B b\,{\mathrm e}^{2 i (dx+c)}+3 C a\,{\mathrm e}^{2 i (dx+c)}-i C b\,{\mathrm e}^{2 i (dx+c)}+B b+3 C a\right)}{d \left({\mathrm e}^{2 i (dx+c)}+1\right)^{2}}+3 B\,a^{2} b x-$

```
[In] int(cot(d*x+c)^2*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,method=
_RETURNVERBOSE)
```

```
[Out] 1/2*((-B*a^3+3*B*a*b^2+3*C*a^2*b-C*b^3)*ln(sec(d*x+c)^2)+2*B*a^3*ln(tan(d*x
+c))+C*b^3*tan(d*x+c)^2+(2*B*b^3+6*C*a*b^2)*tan(d*x+c)+6*d*(B*a^2*b-1/3*B*b
^3+1/3*C*a^3-C*a*b^2)*x)/d
```

Fricas [A] (verification not implemented)

none

Time = 0.27 (sec) , antiderivative size = 133, normalized size of antiderivative = 1.14

$$\int \cot^2(c+dx)(a+b\tan(c+dx))^3 \left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{Cb^3\tan{(dx+c)}^2+Ba^3\log\left(\frac{\tan(dx+c)^2}{\tan(dx+c)^2+1}\right)+2\,(Ca^3+3\,Ba^2b-3\,Cab^2-Bb^3)dx-(3\,Ca^2b+3\,Bab^2-C}{2\,d}$$

```
[In] integrate(cot(d*x+c)^2*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="fricas")
```

```
[Out] 1/2*(C*b^3*tan(d*x + c)^2 + B*a^3*log(tan(d*x + c)^2/(tan(d*x + c)^2 + 1))
+ 2*(C*a^3 + 3*B*a^2*b - 3*C*a*b^2 - B*b^3)*d*x - (3*C*a^2*b + 3*B*a*b^2 -
C*b^3)*log(1/(tan(d*x + c)^2 + 1)) + 2*(3*C*a*b^2 + B*b^3)*tan(d*x + c))/d
```

Sympy [A] (verification not implemented)

Time = 0.98 (sec) , antiderivative size = 211, normalized size of antiderivative = 1.80

$$\int \cot^2(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\begin{cases} -\frac{Ba^3\log\left(\tan^2(c+dx)+1\right)}{2d}+\frac{Ba^3\log\left(\tan(c+dx)\right)}{d}+3Ba^2bx+\frac{3Bab^2\log\left(\tan^2(c+dx)+1\right)}{2d}-Bb^3x+\frac{Bb^3\tan(c+dx)}{d}+Ca^3 \\ x(a+b\tan(c))^3\left(B\tan(c)+C\tan^2(c)\right)\cot^2(c) \end{cases}$$

```
[In] integrate(cot(d*x+c)**2*(a+b*tan(d*x+c))**3*(B*tan(d*x+c)+C*tan(d*x+c)**2),
x)
```

```
[Out] Piecewise((-B*a**3*log(tan(c + d*x)**2 + 1)/(2*d) + B*a**3*log(tan(c + d*x)
)/d + 3*B*a**2*b*x + 3*B*a*b**2*log(tan(c + d*x)**2 + 1)/(2*d) - B*b**3*x +
 B*b**3*tan(c + d*x)/d + C*a**3*x + 3*C*a**2*b*log(tan(c + d*x)**2 + 1)/(2*
d) - 3*C*a*b**2*x + 3*C*a*b**2*tan(c + d*x)/d - C*b**3*log(tan(c + d*x)**2
+ 1)/(2*d) + C*b**3*tan(c + d*x)**2/(2*d), Ne(d, 0)), (x*(a + b*tan(c))**3*
(B*tan(c) + C*tan(c)**2)*cot(c)**2, True))
```

Maxima [A] (verification not implemented)

none

Time = 0.33 (sec) , antiderivative size = 124, normalized size of antiderivative = 1.06

$$\int \cot^2(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{Cb^3\tan(dx+c)^2+2\,Ba^3\log\left(\tan(dx+c)\right)+2\,(Ca^3+3\,Ba^2b-3\,Cab^2-Bb^3)(dx+c)-(Ba^3-3\,Ca^2}{2\,d}$$

```
[In] integrate(cot(d*x+c)^2*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="maxima")
```

```
[Out] 1/2*(C*b^3*tan(d*x + c)^2 + 2*B*a^3*log(tan(d*x + c)) + 2*(C*a^3 + 3*B*a^2*
b - 3*C*a*b^2 - B*b^3)*(d*x + c) - (B*a^3 - 3*C*a^2*b - 3*B*a*b^2 + C*b^3)*
log(tan(d*x + c)^2 + 1) + 2*(3*C*a*b^2 + B*b^3)*tan(d*x + c))/d
```

Giac [A] (verification not implemented)

none

Time = 1.62 (sec) , antiderivative size = 129, normalized size of antiderivative = 1.10

$$\int \cot^2(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{Cb^3\tan(dx+c)^2+2Ba^3\log(|\tan(dx+c)|)+6Cab^2\tan(dx+c)+2Bb^3\tan(dx+c)+2(Ca^3+3Ba^2}{2d}$$

```
[In] integrate(cot(d*x+c)^2*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="giac")
```

```
[Out] 1/2*(C*b^3*tan(d*x + c)^2 + 2*B*a^3*log(abs(tan(d*x + c))) + 6*C*a*b^2*tan(
d*x + c) + 2*B*b^3*tan(d*x + c) + 2*(C*a^3 + 3*B*a^2*b - 3*C*a*b^2 - B*b^3)
*(d*x + c) - (B*a^3 - 3*C*a^2*b - 3*B*a*b^2 + C*b^3)*log(tan(d*x + c)^2 + 1
))/d
```

Mupad [B] (verification not implemented)

Time = 9.32 (sec) , antiderivative size = 118, normalized size of antiderivative = 1.01

$$\int \cot^2(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{\tan(c+d\,x)\,\left(B\,b^3+3\,C\,a\,b^2\right)}{d}+\frac{B\,a^3\,\ln\left(\tan(c+d\,x)\right)}{d}+\frac{C\,b^3\tan(c+d\,x)^2}{2\,d}-\frac{\ln\left(\tan(c+d\,x)+1\mathrm{i}\right)\,\left(B-C\,1\mathrm{i}\right)\,\left(b+a\,1\mathrm{i}\right)^3\,1\mathrm{i}}{2\,d}-\frac{\ln\left(\tan(c+d\,x)-\mathrm{i}\right)\,\left(B+C\,1\mathrm{i}\right)\,\left(-b+a\,1\mathrm{i}\right)^3\,1\mathrm{i}}{2\,d}$$

```
[In] int(cot(c + d*x)^2*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))
^3,x)
```

```
[Out] (tan(c + d*x)*(B*b^3 + 3*C*a*b^2))/d + (B*a^3*log(tan(c + d*x)))/d - (log(t
an(c + d*x) + 1i)*(B - C*1i)*(a*1i + b)^3*1i)/(2*d) - (log(tan(c + d*x) - 1
i)*(B + C*1i)*(a*1i - b)^3*1i)/(2*d) + (C*b^3*tan(c + d*x)^2)/(2*d)
```

3.20 $\int \cot^3(c+dx)(a+b\tan(c+dx))^3 \left(B\tan(c+dx)+C\tan^2(c+dx)\right) dx$

Optimal result 189
Rubi [A] (verified) 189
Mathematica [C] (verified) 192
Maple [A] (verified) 192
Fricas [A] (verification not implemented) 193
Sympy [A] (verification not implemented) 193
Maxima [A] (verification not implemented) 194
Giac [A] (verification not implemented) 194
Mupad [B] (verification not implemented) 194

Optimal result

Integrand size = 40, antiderivative size = 119

$$\begin{aligned}&\int \cot^3(c+dx)(a+b\tan(c+dx))^3 \left(B\tan(c+dx)+C\tan^2(c+dx)\right) dx\\&=-\left(\left(a^3B-3ab^2B-3a^2bC+b^3C\right)x\right)\\&\quad-\frac{b^2(bB+3aC)\log(\cos(c+dx))}{d}+\frac{a^2(3bB+aC)\log(\sin(c+dx))}{d}\\&\quad+\frac{b^2(aB+bC)\tan(c+dx)}{d}-\frac{aB\cot(c+dx)(a+b\tan(c+dx))^2}{d}\end{aligned}$$

```
[Out] -(B*a^3-3*B*a*b^2-3*C*a^2*b+C*b^3)*x-b^2*(B*b+3*C*a)*ln(cos(d*x+c))/d+a^2*(
3*B*b+C*a)*ln(sin(d*x+c))/d+b^2*(B*a+C*b)*tan(d*x+c)/d-a*B*cot(d*x+c)*(a+b*
tan(d*x+c))^2/d
```

Rubi [A] (verified)

Time = 0.39 (sec) , antiderivative size = 119, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.125, Rules used = {3713, 3686, 3718, 3705, 3556}

$$\begin{aligned}&\int \cot^3(c+dx)(a+b\tan(c+dx))^3 \left(B\tan(c+dx)+C\tan^2(c+dx)\right) dx\\&=\frac{a^2(aC+3bB)\log(\sin(c+dx))}{d}-x\left(a^3B-3a^2bC-3ab^2B+b^3C\right)\\&\quad+\frac{b^2(aB+bC)\tan(c+dx)}{d}-\frac{b^2(3aC+bB)\log(\cos(c+dx))}{d}\\&\quad-\frac{aB\cot(c+dx)(a+b\tan(c+dx))^2}{d}\end{aligned}$$

```
[In] Int[Cot[c + d*x]^3*(a + b*Tan[c + d*x])^3*(B*Tan[c + d*x] + C*Tan[c + d*x]^
2),x]
```

```
[Out] -((a^3*B - 3*a*b^2*B - 3*a^2*b*C + b^3*C)*x) - (b^2*(b*B + 3*a*C)*Log[Cos[c
 + d*x]])/d + (a^2*(3*b*B + a*C)*Log[Sin[c + d*x]])/d + (b^2*(a*B + b*C)*Ta
n[c + d*x])/d - (a*B*Cot[c + d*x]*(a + b*Tan[c + d*x])^2)/d
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3686

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Si
mp[(b*c - a*d)*(B*c - A*d)*(a + b*Tan[e + f*x])^(m - 1)*((c + d*Tan[e + f*x
])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dist[1/(d*(n + 1)*(c^2 + d^2)),
 Int[(a + b*Tan[e + f*x])^(m - 2)*(c + d*Tan[e + f*x])^(n + 1)*Simp[a*A*d*(
b*d*(m - 1) - a*c*(n + 1)) + (b*B*c - (A*b + a*B)*d)*(b*c*(m - 1) + a*d*(n
+ 1)) - d*((a*A - b*B)*(b*c - a*d) + (A*b + a*B)*(a*c + b*d))*(n + 1)*Tan[e
 + f*x] - b*(d*(A*b*c + a*B*c - a*A*d)*(m + n) - b*B*(c^2*(m - 1) - d^2*(n
+ 1)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] &&
NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 1] &&
 LtQ[n, -1] && (IntegerQ[m] || IntegersQ[2*m, 2*n])
```

Rule 3705

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/tan[(e_.) + (f_.)*(x_)], x_Symbol] :> Simp[B*x, x] + (Dist[A, Int[1/Tan[e
 + f*x], x], x] + Dist[C, Int[Tan[e + f*x], x], x]) /; FreeQ[{e, f, A, B, C
}, x] && NeQ[A, C]
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
```

```
1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rubi steps

$$
\begin{aligned}
\text{integral} &= \int \cot^2(c+dx)(a+b\tan(c+dx))^3(B+C\tan(c+dx))\,dx \\
&= -\frac{aB\cot(c+dx)(a+b\tan(c+dx))^2}{d} \\
&\quad + \int \cot(c+dx)(a+b\tan(c+dx))\left(a(3bB+aC)-\left(a^2B-b^2B-2abC\right)\tan(c+dx)\right. \\
&\qquad \left. + b(aB+bC)\tan^2(c+dx)\right)\,dx \\
&= \frac{b^2(aB+bC)\tan(c+dx)}{d} - \frac{aB\cot(c+dx)(a+b\tan(c+dx))^2}{d} \\
&\quad - \int \cot(c+dx)\left(-a^2(3bB+aC)+\left(a^3B-3ab^2B-3a^2bC+b^3C\right)\tan(c+dx)\right. \\
&\qquad \left. - b^2(bB+3aC)\tan^2(c+dx)\right)\,dx \\
&= -\left(\left(a^3B-3ab^2B-3a^2bC+b^3C\right)x\right) + \frac{b^2(aB+bC)\tan(c+dx)}{d} \\
&\quad - \frac{aB\cot(c+dx)(a+b\tan(c+dx))^2}{d} + \left(a^2(3bB+aC)\right)\int \cot(c+dx)\,dx \\
&\quad + \left(b^2(bB+3aC)\right)\int \tan(c+dx)\,dx \\
&= -\left(\left(a^3B-3ab^2B-3a^2bC+b^3C\right)x\right) - \frac{b^2(bB+3aC)\log(\cos(c+dx))}{d} \\
&\quad + \frac{a^2(3bB+aC)\log(\sin(c+dx))}{d} + \frac{b^2(aB+bC)\tan(c+dx)}{d} \\
&\quad - \frac{aB\cot(c+dx)(a+b\tan(c+dx))^2}{d}
\end{aligned}
$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 0.52 (sec) , antiderivative size = 113, normalized size of antiderivative = 0.95

$$\int \cot^3(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{-2a^3B\cot(c+dx)+i(a+ib)^3(B+iC)\log(i-\tan(c+dx))+2a^2(3bB+aC)\log(\tan(c+dx))+(ia+b}{2d}$$

```
[In] Integrate[Cot[c + d*x]^3*(a + b*Tan[c + d*x])^3*(B*Tan[c + d*x] + C*Tan[c +
 d*x]^2),x]
```

```
[Out] (-2*a^3*B*Cot[c + d*x] + I*(a + I*b)^3*(B + I*C)*Log[I - Tan[c + d*x]] + 2*
a^2*(3*b*B + a*C)*Log[Tan[c + d*x]] + (I*a + b)^3*(B - I*C)*Log[I + Tan[c +
 d*x]] + 2*b^3*C*Tan[c + d*x])/(2*d)
```

Maple [A] (verified)

Time = 0.26 (sec) , antiderivative size = 118, normalized size of antiderivative = 0.99

method	result
parallelrisch	$\frac{\left(-3B\,a^2b+B\,b^3-C\,a^3+3Ca\,b^2\right)\ln\left(\sec(dx+c)^2\right)+\left(6B\,a^2b+2C\,a^3\right)\ln(\tan(dx+c))-2B\cot(dx+c)a^3+2C\,b^3\tan(dx+c)-2}{2d}$
derivativedivides	$\frac{C\,b^3\tan(dx+c)-\frac{B\,a^3}{\tan(dx+c)}+a^2(3Bb+Ca)\ln(\tan(dx+c))+\frac{\left(-3B\,a^2b+B\,b^3-C\,a^3+3Ca\,b^2\right)\ln\left(1+\tan(dx+c)^2\right)}{2}+\left(-B\,a^3+3Ba\right.}{d}$
default	$\frac{C\,b^3\tan(dx+c)-\frac{B\,a^3}{\tan(dx+c)}+a^2(3Bb+Ca)\ln(\tan(dx+c))+\frac{\left(-3B\,a^2b+B\,b^3-C\,a^3+3Ca\,b^2\right)\ln\left(1+\tan(dx+c)^2\right)}{2}+\left(-B\,a^3+3Ba\right.}{d}$
norman	$\frac{\left(-B\,a^3+3Ba\,b^2+3C\,a^2b-C\,b^3\right)x\tan(dx+c)^2+\frac{C\,b^3\tan(dx+c)^3}{d}-\frac{B\,a^3\tan(dx+c)}{d}}{\tan(dx+c)^2}+\frac{a^2(3Bb+Ca)\ln(\tan(dx+c))}{d}-\frac{(3B}{}$
risch	$-B\,a^3x+3Ba\,b^2x+3C\,a^2bx-C\,b^3x-\frac{2iC\,a^3c}{d}+iB\,b^3x-iC\,a^3x+\frac{2iB\,b^3c}{d}-\frac{2i\left(B\,a^3e^{2i(dx}\right.}{d\left(e^{2i(\right.}}$

```
[In] int(cot(d*x+c)^3*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,method=
_RETURNVERBOSE)
```

```
[Out] 1/2*((-3*B*a^2*b+B*b^3-C*a^3+3*C*a*b^2)*ln(sec(d*x+c)^2)+(6*B*a^2*b+2*C*a^3
)*ln(tan(d*x+c))-2*B*cot(d*x+c)*a^3+2*C*b^3*tan(d*x+c)-2*d*x*(B*a^3-3*B*a*b
^2-3*C*a^2*b+C*b^3))/d
```

Fricas [A] (verification not implemented)

none

Time = 0.27 (sec) , antiderivative size = 145, normalized size of antiderivative = 1.22

$$\int \cot^3(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{2\,Cb^3\tan\left(dx+c\right)^2-2\,Ba^3-2\,(Ba^3-3\,Ca^2b-3\,Bab^2+Cb^3)dx\tan\left(dx+c\right)+(Ca^3+3\,Ba^2b)\log\left(\frac{\text{ta}}{\text{tan}}\right.}{2\,d\tan\left(dx+c\right)}$$

```
[In] integrate(cot(d*x+c)^3*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="fricas")
```

```
[Out] 1/2*(2*C*b^3*tan(d*x + c)^2 - 2*B*a^3 - 2*(B*a^3 - 3*C*a^2*b - 3*B*a*b^2 +
C*b^3)*d*x*tan(d*x + c) + (C*a^3 + 3*B*a^2*b)*log(tan(d*x + c)^2/(tan(d*x +
 c)^2 + 1))*tan(d*x + c) - (3*C*a*b^2 + B*b^3)*log(1/(tan(d*x + c)^2 + 1))*
tan(d*x + c))/(d*tan(d*x + c))
```

Sympy [A] (verification not implemented)

Time = 1.70 (sec) , antiderivative size = 214, normalized size of antiderivative = 1.80

$$\int \cot^3(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\begin{cases}\text{NaN}\\ x(a+b\tan(c))^3\left(B\tan(c)+C\tan^2(c)\right)\cot^3(c)\\ \text{NaN}\\ -Ba^3x-\frac{Ba^3}{d\tan(c+dx)}-\frac{3Ba^2b\log(\tan^2(c+dx)+1)}{2d}+\frac{3Ba^2b\log(\tan(c+dx))}{d}+3Bab^2x+\frac{Bb^3\log(\tan^2(c+dx)+1)}{2d}-\frac{Ca^3\,\text{lo}}{}\end{cases}$$

```
[In] integrate(cot(d*x+c)**3*(a+b*tan(d*x+c))**3*(B*tan(d*x+c)+C*tan(d*x+c)**2),
x)
```

```
[Out] Piecewise((nan, Eq(c, 0) & Eq(d, 0)), (x*(a + b*tan(c))**3*(B*tan(c) + C*ta
n(c)**2)*cot(c)**3, Eq(d, 0)), (nan, Eq(c, -d*x)), (-B*a**3*x - B*a**3/(d*t
an(c + d*x)) - 3*B*a**2*b*log(tan(c + d*x)**2 + 1)/(2*d) + 3*B*a**2*b*log(t
an(c + d*x))/d + 3*B*a*b**2*x + B*b**3*log(tan(c + d*x)**2 + 1)/(2*d) - C*a
**3*log(tan(c + d*x)**2 + 1)/(2*d) + C*a**3*log(tan(c + d*x))/d + 3*C*a**2*
b*x + 3*C*a*b**2*log(tan(c + d*x)**2 + 1)/(2*d) - C*b**3*x + C*b**3*tan(c +
 d*x)/d, True))
```

Maxima [A] (verification not implemented)

none

Time = 0.35 (sec) , antiderivative size = 125, normalized size of antiderivative = 1.05

$$\int \cot^3(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{2\,Cb^3\tan\left(dx+c\right)-\frac{2\,Ba^3}{\tan(dx+c)}-2\left(Ba^3-3\,Ca^2b-3\,Bab^2+Cb^3\right)(dx+c)-\left(Ca^3+3\,Ba^2b-3\,Cab^2-Bb\right.}{2\,d}$$

```
[In] integrate(cot(d*x+c)^3*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="maxima")
```

```
[Out] 1/2*(2*C*b^3*tan(d*x + c) - 2*B*a^3/tan(d*x + c) - 2*(B*a^3 - 3*C*a^2*b - 3
*B*a*b^2 + C*b^3)*(d*x + c) - (C*a^3 + 3*B*a^2*b - 3*C*a*b^2 - B*b^3)*log(t
an(d*x + c)^2 + 1) + 2*(C*a^3 + 3*B*a^2*b)*log(tan(d*x + c)))/d
```

Giac [A] (verification not implemented)

none

Time = 1.22 (sec) , antiderivative size = 152, normalized size of antiderivative = 1.28

$$\int \cot^3(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{2\,Cb^3\tan\left(dx+c\right)-2\left(Ba^3-3\,Ca^2b-3\,Bab^2+Cb^3\right)(dx+c)-\left(Ca^3+3\,Ba^2b-3\,Cab^2-Bb^3\right)\log\left(\tan\right.}{2\,d}$$

```
[In] integrate(cot(d*x+c)^3*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="giac")
```

```
[Out] 1/2*(2*C*b^3*tan(d*x + c) - 2*(B*a^3 - 3*C*a^2*b - 3*B*a*b^2 + C*b^3)*(d*x
+ c) - (C*a^3 + 3*B*a^2*b - 3*C*a*b^2 - B*b^3)*log(tan(d*x + c)^2 + 1) + 2*
(C*a^3 + 3*B*a^2*b)*log(abs(tan(d*x + c))) - 2*(C*a^3*tan(d*x + c) + 3*B*a^
2*b*tan(d*x + c) + B*a^3)/tan(d*x + c))/d
```

Mupad [B] (verification not implemented)

Time = 8.78 (sec) , antiderivative size = 114, normalized size of antiderivative = 0.96

$$\int \cot^3(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{\ln\left(\tan(c+d\,x)\right)\,\left(C\,a^3+3\,B\,b\,a^2\right)}{d}-\frac{B\,a^3\cot(c+d\,x)}{d}$$
$$+\frac{C\,b^3\tan(c+d\,x)}{d}+\frac{\ln\left(\tan(c+d\,x)-\mathrm{i}\right)\,\left(B+C\,1\mathrm{i}\right)\,\left(a+b\,1\mathrm{i}\right)^3\,1\mathrm{i}}{2\,d}$$
$$-\frac{\ln\left(\tan(c+d\,x)+1\mathrm{i}\right)\,\left(B-C\,1\mathrm{i}\right)\,\left(a-b\,1\mathrm{i}\right)^3\,1\mathrm{i}}{2\,d}$$

[In] int(cot(c + d*x)^3*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))^3,x)

[Out] (log(tan(c + d*x))*(C*a^3 + 3*B*a^2*b))/d + (log(tan(c + d*x) - 1i)*(B + C*1i)*(a + b*1i)^3*1i)/(2*d) - (log(tan(c + d*x) + 1i)*(B - C*1i)*(a - b*1i)^3*1i)/(2*d) - (B*a^3*cot(c + d*x))/d + (C*b^3*tan(c + d*x))/d

3.21 $\int \cot^4(c+dx)(a+b\tan(c+dx))^3 \left(B\tan(c+dx)+C\tan\right.$

Optimal result 196
Rubi [A] (verified) 196
Mathematica [C] (verified) 198
Maple [A] (verified) 199
Fricas [A] (verification not implemented) 199
Sympy [B] (verification not implemented) 200
Maxima [A] (verification not implemented) 200
Giac [A] (verification not implemented) 201
Mupad [B] (verification not implemented) 201

Optimal result

Integrand size = 40, antiderivative size = 127

$$\int \cot^4(c+dx)(a+b\tan(c+dx))^3 \left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$= -\left(\left(3a^2bB-b^3B+a^3C-3ab^2C\right)x\right)-\frac{a^2(2bB+aC)\cot(c+dx)}{d}-\frac{b^3C\log(\cos(c+dx))}{d}$$
$$-\frac{a(a^2B-3b^2B-3abC)\log(\sin(c+dx))}{d}-\frac{aB\cot^2(c+dx)(a+b\tan(c+dx))^2}{2d}$$

```
[Out] -(3*B*a^2*b-B*b^3+C*a^3-3*C*a*b^2)*x-a^2*(2*B*b+C*a)*cot(d*x+c)/d-b^3*C*ln(
cos(d*x+c))/d-a*(B*a^2-3*B*b^2-3*C*a*b)*ln(sin(d*x+c))/d-1/2*a*B*cot(d*x+c)
^2*(a+b*tan(d*x+c))^2/d
```

Rubi [A] (verified)

Time = 0.42 (sec) , antiderivative size = 127, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.125, Rules used = {3713, 3686, 3716, 3705, 3556}

$$\int \cot^4(c+dx)(a+b\tan(c+dx))^3 \left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$= -\frac{a(a^2B-3abC-3b^2B)\log(\sin(c+dx))}{d}$$
$$-\frac{a^2(aC+2bB)\cot(c+dx)}{d}-x\left(a^3C+3a^2bB-3ab^2C-b^3B\right)$$
$$-\frac{aB\cot^2(c+dx)(a+b\tan(c+dx))^2}{2d}-\frac{b^3C\log(\cos(c+dx))}{d}$$

```
[In] Int[Cot[c + d*x]^4*(a + b*Tan[c + d*x])^3*(B*Tan[c + d*x] + C*Tan[c + d*x]^
2),x]
```

```
[Out] -((3*a^2*b*B - b^3*B + a^3*C - 3*a*b^2*C)*x) - (a^2*(2*b*B + a*C)*Cot[c + d
*x])/d - (b^3*C*Log[Cos[c + d*x]])/d - (a*(a^2*B - 3*b^2*B - 3*a*b*C)*Log[S
in[c + d*x]])/d - (a*B*Cot[c + d*x]^2*(a + b*Tan[c + d*x])^2)/(2*d)
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3686

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Si
mp[(b*c - a*d)*(B*c - A*d)*(a + b*Tan[e + f*x])^(m - 1)*((c + d*Tan[e + f*x
])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dist[1/(d*(n + 1)*(c^2 + d^2)),
 Int[(a + b*Tan[e + f*x])^(m - 2)*(c + d*Tan[e + f*x])^(n + 1)*Simp[a*A*d*(
b*d*(m - 1) - a*c*(n + 1)) + (b*B*c - (A*b + a*B)*d)*(b*c*(m - 1) + a*d*(n
+ 1)) - d*((a*A - b*B)*(b*c - a*d) + (A*b + a*B)*(a*c + b*d))*(n + 1)*Tan[e
 + f*x] - b*(d*(A*b*c + a*B*c - a*A*d)*(m + n) - b*B*(c^2*(m - 1) - d^2*(n
+ 1)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] &&
NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 1] &&
 LtQ[n, -1] && (IntegerQ[m] || IntegersQ[2*m, 2*n])
```

Rule 3705

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/tan[(e_.) + (f_.)*(x_)], x_Symbol] :> Simp[B*x, x] + (Dist[A, Int[1/Tan[e
 + f*x], x], x] + Dist[C, Int[Tan[e + f*x], x], x]) /; FreeQ[{e, f, A, B, C
}, x] && NeQ[A, C]
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rule 3716

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.
)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(-(b*c - a*d))*(c^2*C - B*c*d + A*d^2)*((c
+ d*Tan[e + f*x])^(n + 1)/(d^2*f*(n + 1)*(c^2 + d^2))), x] + Dist[1/(d*(c^2
 + d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)*Simp[a*d*(A*c - c*C + B*d) + b*(
c^2*C - B*c*d + A*d^2) + d*(A*b*c + a*B*c - b*c*C - a*A*d + b*B*d + a*C*d)*
```

```
Tan[e + f*x] + b*C*(c^2 + d^2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c
, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] && LtQ[n,
 -1]
```

Rubi steps

$$
\begin{aligned}
\text{integral} &= \int \cot^3(c+dx)(a+b\tan(c+dx))^3(B+C\tan(c+dx))\,dx \\
&= -\frac{aB\cot^2(c+dx)(a+b\tan(c+dx))^2}{2d} \\
&\quad+\frac{1}{2}\int \cot^2(c+dx)(a+b\tan(c+dx))\left(2a(2bB+aC)\right. \\
&\qquad\left.-2\left(a^2B-b^2B-2abC\right)\tan(c+dx)+2b^2C\tan^2(c+dx)\right)\,dx \\
&= -\frac{a^2(2bB+aC)\cot(c+dx)}{d}-\frac{aB\cot^2(c+dx)(a+b\tan(c+dx))^2}{2d} \\
&\quad+\frac{1}{2}\int \cot(c+dx)\left(-2a\left(a^2B-3b^2B-3abC\right)\right. \\
&\qquad\left.-2\left(3a^2bB-b^3B+a^3C-3ab^2C\right)\tan(c+dx)+2b^3C\tan^2(c+dx)\right)\,dx \\
&= -\left(\left(3a^2bB-b^3B+a^3C-3ab^2C\right)x\right)-\frac{a^2(2bB+aC)\cot(c+dx)}{d} \\
&\quad-\frac{aB\cot^2(c+dx)(a+b\tan(c+dx))^2}{2d}+\left(b^3C\right)\int\tan(c+dx)\,dx \\
&\quad-\left(a\left(a^2B-3b^2B-3abC\right)\right)\int\cot(c+dx)\,dx \\
&= -\left(\left(3a^2bB-b^3B+a^3C-3ab^2C\right)x\right)-\frac{a^2(2bB+aC)\cot(c+dx)}{d}-\frac{b^3C\log(\cos(c+dx))}{d} \\
&\quad-\frac{a\left(a^2B-3b^2B-3abC\right)\log(\sin(c+dx))}{d}-\frac{aB\cot^2(c+dx)(a+b\tan(c+dx))^2}{2d}
\end{aligned}
$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 0.49 (sec) , antiderivative size = 126, normalized size of antiderivative = 0.99

$$
\int \cot^4(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx
$$

$$
=\frac{-2a^2(3bB+aC)\cot(c+dx)-a^3B\cot^2(c+dx)+(a+ib)^3(B+iC)\log(i-\tan(c+dx))-2a(a^2B-3b^{}}{2d}
$$

```
[In] Integrate[Cot[c + d*x]^4*(a + b*Tan[c + d*x])^3*(B*Tan[c + d*x] + C*Tan[c +
 d*x]^2),x]
```

```
[Out] (-2*a^2*(3*b*B + a*C)*Cot[c + d*x] - a^3*B*Cot[c + d*x]^2 + (a + I*b)^3*(B
+ I*C)*Log[I - Tan[c + d*x]] - 2*a*(a^2*B - 3*b^2*B - 3*a*b*C)*Log[Tan[c +
d*x]] + (a - I*b)^3*(B - I*C)*Log[I + Tan[c + d*x]])/(2*d)
```

Maple [A] (verified)

Time = 0.24 (sec) , antiderivative size = 136, normalized size of antiderivative = 1.07

method	result
parallelrisch	$\frac{\left(B\,a^{3}-3B a\,b^{2}-3C\,a^{2}b+C\,b^{3}\right)\ln\left(\sec(dx+c)^{2}\right)+\left(-2B\,a^{3}+6B a\,b^{2}+6C\,a^{2}b\right)\ln(\tan(dx+c))-B\,a^{3}\cot(dx+c)^{2}+(-6B\,a}{2d}$
derivativedivides	$\frac{-\frac{B\,a^{3}}{2\tan(dx+c)^{2}}-\frac{a^{2}(3Bb+Ca)}{\tan(dx+c)}-a\left(B\,a^{2}-3B\,b^{2}-3Cab\right)\ln(\tan(dx+c))+\frac{\left(B\,a^{3}-3B a\,b^{2}-3C\,a^{2}b+C\,b^{3}\right)\ln\left(1+\tan(dx+c)^{2}\right)}{2}+(-}{d}$
default	$\frac{-\frac{B\,a^{3}}{2\tan(dx+c)^{2}}-\frac{a^{2}(3Bb+Ca)}{\tan(dx+c)}-a\left(B\,a^{2}-3B\,b^{2}-3Cab\right)\ln(\tan(dx+c))+\frac{\left(B\,a^{3}-3B a\,b^{2}-3C\,a^{2}b+C\,b^{3}\right)\ln\left(1+\tan(dx+c)^{2}\right)}{2}+(-}{d}$
norman	$\frac{\left(-3B\,a^{2}b+B\,b^{3}-C\,a^{3}+3Ca\,b^{2}\right)x\tan(dx+c)^{3}-\frac{B\,a^{3}\tan(dx+c)}{2d}-\frac{a^{2}(3Bb+Ca)\tan(dx+c)^{2}}{d}}{\tan(dx+c)^{3}}+\frac{\left(B\,a^{3}-3B a\,b^{2}-3C\,a^{2}b+C\right.}{2d}$
risch	$iB\,a^{3}x-3iC\,a^{2}bx-3iBa\,b^{2}x+\frac{2iC\,b^{3}c}{d}-3B\,a^{2}bx+B\,b^{3}x-C\,a^{3}x+3Ca\,b^{2}x-\frac{2ia^{2}(3}{}$

[In] int(cot(d*x+c)^4*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,method=_RETURNVERBOSE)

[Out] 1/2*((B*a^3-3*B*a*b^2-3*C*a^2*b+C*b^3)*ln(sec(d*x+c)^2)+(-2*B*a^3+6*B*a*b^2+6*C*a^2*b)*ln(tan(d*x+c))-B*a^3*cot(d*x+c)^2+(-6*B*a^2*b-2*C*a^3)*cot(d*x+c)-6*d*(B*a^2*b-1/3*B*b^3+1/3*C*a^3-C*a*b^2)*x)/d

Fricas [A] (verification not implemented)

none

Time = 0.28 (sec) , antiderivative size = 162, normalized size of antiderivative = 1.28

$$\int \cot^4(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx = -\frac{Cb^3\log\left(\frac{1}{\tan(dx+c)^2+1}\right)\tan(dx+c)^2+Ba^3+(Ba^3-3\,Ca^2b-3\,Bab^2)\log\left(\frac{\tan(dx+c)^2}{\tan(dx+c)^2+1}\right)\tan(dx+c)^2}{2\,d\tan(dx+}$$

[In] integrate(cot(d*x+c)^4*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, algorithm="fricas")

[Out] -1/2*(C*b^3*log(1/(tan(d*x + c)^2 + 1))*tan(d*x + c)^2 + B*a^3 + (B*a^3 - 3*C*a^2*b - 3*B*a*b^2)*log(tan(d*x + c)^2/(tan(d*x + c)^2 + 1))*tan(d*x + c)^2 + (B*a^3 + 2*(C*a^3 + 3*B*a^2*b - 3*C*a*b^2 - B*b^3)*d*x)*tan(d*x + c)^2 + 2*(C*a^3 + 3*B*a^2*b)*tan(d*x + c))/(d*tan(d*x + c)^2)

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 253 vs. 2(121) = 242.

Time = 2.33 (sec) , antiderivative size = 253, normalized size of antiderivative = 1.99

$$\int \cot^4(c+dx)(a+b\tan(c+dx))^3 \left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\begin{cases} \text{NaN} \\ x(a+b\tan(c))^3\left(B\tan(c)+C\tan^2(c)\right)\cot^4(c) \\ \text{NaN} \\ \frac{Ba^3\log(\tan^2(c+dx)+1)}{2d}-\frac{Ba^3\log(\tan(c+dx))}{d}-\frac{Ba^3}{2d\tan^2(c+dx)}-3Ba^2bx-\frac{3Ba^2b}{d\tan(c+dx)}-\frac{3Bab^2\log(\tan^2(c+dx)+1)}{2d}+\frac{3Ba}{} \end{cases}$$

```
[In] integrate(cot(d*x+c)**4*(a+b*tan(d*x+c))**3*(B*tan(d*x+c)+C*tan(d*x+c)**2),
x)
```

```
[Out] Piecewise((nan, Eq(c, 0) & Eq(d, 0)), (x*(a + b*tan(c))**3*(B*tan(c) + C*ta
n(c)**2)*cot(c)**4, Eq(d, 0)), (nan, Eq(c, -d*x)), (B*a**3*log(tan(c + d*x)
**2 + 1)/(2*d) - B*a**3*log(tan(c + d*x))/d - B*a**3/(2*d*tan(c + d*x)**2)
- 3*B*a**2*b*x - 3*B*a**2*b/(d*tan(c + d*x)) - 3*B*a*b**2*log(tan(c + d*x)*
*2 + 1)/(2*d) + 3*B*a*b**2*log(tan(c + d*x))/d + B*b**3*x - C*a**3*x - C*a*
*3/(d*tan(c + d*x)) - 3*C*a**2*b*log(tan(c + d*x)**2 + 1)/(2*d) + 3*C*a**2*
b*log(tan(c + d*x))/d + 3*C*a*b**2*x + C*b**3*log(tan(c + d*x)**2 + 1)/(2*d
), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.34 (sec) , antiderivative size = 142, normalized size of antiderivative = 1.12

$$\int \cot^4(c+dx)(a+b\tan(c+dx))^3 \left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx =$$
$$-\frac{2\,(Ca^3+3\,Ba^2b-3\,Cab^2-Bb^3)(dx+c)-(Ba^3-3\,Ca^2b-3\,Bab^2+Cb^3)\log\left(\tan(dx+c)^2+1\right)+2}{2\,d}$$

```
[In] integrate(cot(d*x+c)^4*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="maxima")
```

```
[Out] -1/2*(2*(C*a^3 + 3*B*a^2*b - 3*C*a*b^2 - B*b^3)*(d*x + c) - (B*a^3 - 3*C*a^
2*b - 3*B*a*b^2 + C*b^3)*log(tan(d*x + c)^2 + 1) + 2*(B*a^3 - 3*C*a^2*b - 3
*B*a*b^2)*log(tan(d*x + c)) + (B*a^3 + 2*(C*a^3 + 3*B*a^2*b)*tan(d*x + c))/
tan(d*x + c)^2)/d
```

Giac [A] (verification not implemented)

none

Time = 1.28 (sec) , antiderivative size = 193, normalized size of antiderivative = 1.52

$$\int \cot^4(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx =$$

$$-\frac{2\,(Ca^3+3\,Ba^2b-3\,Cab^2-Bb^3)(dx+c)-(Ba^3-3\,Ca^2b-3\,Bab^2+Cb^3)\log\left(\tan\left(dx+c\right)^2+1\right)+}{}$$

```
[In] integrate(cot(d*x+c)^4*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="giac")
```

```
[Out] -1/2*(2*(C*a^3 + 3*B*a^2*b - 3*C*a*b^2 - B*b^3)*(d*x + c) - (B*a^3 - 3*C*a^
2*b - 3*B*a*b^2 + C*b^3)*log(tan(d*x + c)^2 + 1) + 2*(B*a^3 - 3*C*a^2*b - 3
*B*a*b^2)*log(abs(tan(d*x + c))) - (3*B*a^3*tan(d*x + c)^2 - 9*C*a^2*b*tan(
d*x + c)^2 - 9*B*a*b^2*tan(d*x + c)^2 - 2*C*a^3*tan(d*x + c) - 6*B*a^2*b*ta
n(d*x + c) - B*a^3)/tan(d*x + c)^2)/d
```

Mupad [B] (verification not implemented)

Time = 8.77 (sec) , antiderivative size = 135, normalized size of antiderivative = 1.06

$$\begin{aligned}
&\int \cot^4(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx \\
&= \frac{\ln\left(\tan(c+d\,x)\right)\,\left(-B\,a^3+3\,C\,a^2\,b+3\,B\,a\,b^2\right)}{d} \\
&- \frac{\cot(c+d\,x)^2\,\left(\tan(c+d\,x)\,\left(C\,a^3+3\,B\,b\,a^2\right)+\frac{B\,a^3}{2}\right)}{d} \\
&+ \frac{\ln\left(\tan(c+d\,x)+1\mathrm{i}\right)\,\left(B-C\,1\mathrm{i}\right)\,{\left(b+a\,1\mathrm{i}\right)}^3\,1\mathrm{i}}{2\,d} \\
&+ \frac{\ln\left(\tan(c+d\,x)-\mathrm{i}\right)\,\left(B+C\,1\mathrm{i}\right)\,{\left(-b+a\,1\mathrm{i}\right)}^3\,1\mathrm{i}}{2\,d}
\end{aligned}$$

```
[In] int(cot(c + d*x)^4*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))
^3,x)
```

```
[Out] (log(tan(c + d*x))*(3*B*a*b^2 - B*a^3 + 3*C*a^2*b))/d - (cot(c + d*x)^2*(ta
n(c + d*x)*(C*a^3 + 3*B*a^2*b) + (B*a^3)/2))/d + (log(tan(c + d*x) + 1i)*(B
 - C*1i)*(a*1i + b)^3*1i)/(2*d) + (log(tan(c + d*x) - 1i)*(B + C*1i)*(a*1i
- b)^3*1i)/(2*d)
```

3.22 $\int \cot^5(c+dx)(a+b\tan(c+dx))^3 \left(B\tan(c+dx)+C\tan^2(c+dx)\right)dx$

Optimal result 202
Rubi [A] (verified) 202
Mathematica [C] (verified) 205
Maple [A] (verified) 205
Fricas [A] (verification not implemented) 206
Sympy [B] (verification not implemented) 206
Maxima [A] (verification not implemented) 207
Giac [B] (verification not implemented) 207
Mupad [B] (verification not implemented) 208

Optimal result

Integrand size = 40, antiderivative size = 154

$$\begin{aligned}&\int \cot^5(c+dx)(a+b\tan(c+dx))^3 \left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx\\
&=\left(a^3B-3ab^2B-3a^2bC+b^3C\right)x+\frac{a(3a^2B-8b^2B-9abC)\cot(c+dx)}{3d}\\
&\quad-\frac{a^2(5bB+3aC)\cot^2(c+dx)}{6d}-\frac{(3a^2bB-b^3B+a^3C-3ab^2C)\log(\sin(c+dx))}{d}\\
&\quad-\frac{aB\cot^3(c+dx)(a+b\tan(c+dx))^2}{3d}\end{aligned}$$

```
[Out] (B*a^3-3*B*a*b^2-3*C*a^2*b+C*b^3)*x+1/3*a*(3*B*a^2-8*B*b^2-9*C*a*b)*cot(d*x
+c)/d-1/6*a^2*(5*B*b+3*C*a)*cot(d*x+c)^2/d-(3*B*a^2*b-B*b^3+C*a^3-3*C*a*b^2
)*ln(sin(d*x+c))/d-1/3*a*B*cot(d*x+c)^3*(a+b*tan(d*x+c))^2/d
```

Rubi [A] (verified)

Time = 0.51 (sec) , antiderivative size = 154, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.150, Rules used = {3713, 3686, 3716, 3709, 3612, 3556}

$$\begin{aligned}&\int \cot^5(c+dx)(a+b\tan(c+dx))^3 \left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx\\
&=\frac{a(3a^2B-9abC-8b^2B)\cot(c+dx)}{3d}-\frac{a^2(3aC+5bB)\cot^2(c+dx)}{6d}\\
&\quad-\frac{(a^3C+3a^2bB-3ab^2C-b^3B)\log(\sin(c+dx))}{d}\\
&\quad+x\left(a^3B-3a^2bC-3ab^2B+b^3C\right)-\frac{aB\cot^3(c+dx)(a+b\tan(c+dx))^2}{3d}\end{aligned}$$

```
[In] Int[Cot[c + d*x]^5*(a + b*Tan[c + d*x])^3*(B*Tan[c + d*x] + C*Tan[c + d*x]^
2),x]
```

```
[Out] (a^3*B - 3*a*b^2*B - 3*a^2*b*C + b^3*C)*x + (a*(3*a^2*B - 8*b^2*B - 9*a*b*C
)*Cot[c + d*x])/(3*d) - (a^2*(5*b*B + 3*a*C)*Cot[c + d*x]^2)/(6*d) - ((3*a^
2*b*B - b^3*B + a^3*C - 3*a*b^2*C)*Log[Sin[c + d*x]])/d - (a*B*Cot[c + d*x]
^3*(a + b*Tan[c + d*x])^2)/(3*d)
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3612

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[(a*c + b*d)*(x/(a^2 + b^2)), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3686

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Si
mp[(b*c - a*d)*(B*c - A*d)*(a + b*Tan[e + f*x])^(m - 1)*((c + d*Tan[e + f*x
])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dist[1/(d*(n + 1)*(c^2 + d^2)),
 Int[(a + b*Tan[e + f*x])^(m - 2)*(c + d*Tan[e + f*x])^(n + 1)*Simp[a*A*d*(
b*d*(m - 1) - a*c*(n + 1)) + (b*B*c - (A*b + a*B)*d)*(b*c*(m - 1) + a*d*(n
+ 1)) - d*((a*A - b*B)*(b*c - a*d) + (A*b + a*B)*(a*c + b*d))*(n + 1)*Tan[e
 + f*x] - b*(d*(A*b*c + a*B*c - a*A*d)*(m + n) - b*B*(c^2*(m - 1) - d^2*(n
+ 1)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] &&
NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 1] &&
 LtQ[n, -1] && (IntegerQ[m] || IntegersQ[2*m, 2*n])
```

Rule 3709

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 -
 a*b*B + a^2*C)*((a + b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1)*(a^2 + b^2))), x
] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[b*B + a*(A -
C) - (A*b - a*B - b*C)*Tan[e + f*x], x], x], x] /; FreeQ[{a, b, e, f, A, B,
 C}, x] && NeQ[A*b^2 - a*b*B + a^2*C, 0] && LtQ[m, -1] && NeQ[a^2 + b^2, 0]
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
```

```
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rule 3716

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.
)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(-(b*c - a*d))*(c^2*C - B*c*d + A*d^2)*((c
+ d*Tan[e + f*x])^(n + 1)/(d^2*f*(n + 1)*(c^2 + d^2))), x] + Dist[1/(d*(c^2
 + d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)*Simp[a*d*(A*c - c*C + B*d) + b*(
c^2*C - B*c*d + A*d^2) + d*(A*b*c + a*B*c - b*c*C - a*A*d + b*B*d + a*C*d)*
Tan[e + f*x] + b*C*(c^2 + d^2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c
, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] && LtQ[n,
 -1]
```

Rubi steps

$$\text{integral} = \int \cot^4(c+dx)(a+b\tan(c+dx))^3(B+C\tan(c+dx))\,dx$$

$$= -\frac{aB\cot^3(c+dx)(a+b\tan(c+dx))^2}{3d} + \frac{1}{3}\int \cot^3(c+dx)(a+b\tan(c+dx))\left(a(5bB+3aC) - 3\left(a^2B-b^2B-2abC\right)\tan(c+dx) - b(aB-3bC)\tan^2(c+dx)\right)dx$$

$$= -\frac{a^2(5bB+3aC)\cot^2(c+dx)}{6d} - \frac{aB\cot^3(c+dx)(a+b\tan(c+dx))^2}{3d} + \frac{1}{3}\int \cot^2(c+dx)\left(-a\left(3a^2B-8b^2B-9abC\right) - 3\left(3a^2bB-b^3B+a^3C-3ab^2C\right)\tan(c+dx) - b^2(aB-3bC)\tan^2(c+dx)\right)dx$$

$$= \frac{a(3a^2B-8b^2B-9abC)\cot(c+dx)}{3d} - \frac{a^2(5bB+3aC)\cot^2(c+dx)}{6d} - \frac{aB\cot^3(c+dx)(a+b\tan(c+dx))^2}{3d} + \frac{1}{3}\int \cot(c+dx)\left(-3\left(3a^2bB-b^3B+a^3C-3ab^2C\right) + 3\left(a^3B-3ab^2B-3a^2bC+b^3C\right)\tan(c+dx)\right)dx$$

$$= \left(a^3B-3ab^2B-3a^2bC+b^3C\right)x + \frac{a(3a^2B-8b^2B-9abC)\cot(c+dx)}{3d} - \frac{a^2(5bB+3aC)\cot^2(c+dx)}{6d} - \frac{aB\cot^3(c+dx)(a+b\tan(c+dx))^2}{3d} + \left(-3a^2bB+b^3B-a^3C+3ab^2C\right)\int \cot(c+dx)\,dx$$

$$
\begin{aligned}
&= \left(a^3B - 3ab^2B - 3a^2bC + b^3C\right)x + \frac{a(3a^2B - 8b^2B - 9abC)\cot(c+dx)}{3d} \\
&\quad - \frac{a^2(5bB + 3aC)\cot^2(c+dx)}{6d} - \frac{(3a^2bB - b^3B + a^3C - 3ab^2C)\log(\sin(c+dx))}{d} \\
&\quad - \frac{aB\cot^3(c+dx)(a + b\tan(c+dx))^2}{3d}
\end{aligned}
$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 1.34 (sec) , antiderivative size = 164, normalized size of antiderivative = 1.06

$$
\int \cot^5(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx) + C\tan^2(c+dx)\right)\,dx
$$

$$
= \frac{6a(a^2B - 3b^2B - 3abC)\cot(c+dx) - 3a^2(3bB + aC)\cot^2(c+dx) - 2a^3B\cot^3(c+dx) + 3(a+ib)^3(-i}{}
$$

```
[In] Integrate[Cot[c + d*x]^5*(a + b*Tan[c + d*x])^3*(B*Tan[c + d*x] + C*Tan[c +
 d*x]^2),x]
```

```
[Out] (6*a*(a^2*B - 3*b^2*B - 3*a*b*C)*Cot[c + d*x] - 3*a^2*(3*b*B + a*C)*Cot[c +
 d*x]^2 - 2*a^3*B*Cot[c + d*x]^3 + 3*(a + I*b)^3*((-I)*B + C)*Log[I - Tan[c
 + d*x]] - 6*(3*a^2*b*B - b^3*B + a^3*C - 3*a*b^2*C)*Log[Tan[c + d*x]] + 3*
(a - I*b)^3*(I*B + C)*Log[I + Tan[c + d*x]])/(6*d)
```

Maple [A] (verified)

Time = 0.31 (sec) , antiderivative size = 172, normalized size of antiderivative = 1.12

method	result
parallelrisch	$\frac{3(3B\,a^2b-B\,b^3+C\,a^3-3Ca\,b^2)\ln\left(\sec(dx+c)^2\right)+6(-3B\,a^2b+B\,b^3-C\,a^3+3Ca\,b^2)\ln(\tan(dx+c))-2B\,a^3\cot(dx+c)^3+}{6d}$
derivativedivides	$\frac{(-3B\,a^2b+B\,b^3-C\,a^3+3Ca\,b^2)\ln(\tan(dx+c))-\frac{B\,a^3}{3\tan(dx+c)^3}-\frac{a^2(3Bb+Ca)}{2\tan(dx+c)^2}+\frac{a\left(B\,a^2-3B\,b^2-3Cab\right)}{\tan(dx+c)}+\frac{\left(3B\,a^2b-B\,b^3+C\,a^3\right.}{}}{d}$
default	$\frac{(-3B\,a^2b+B\,b^3-C\,a^3+3Ca\,b^2)\ln(\tan(dx+c))-\frac{B\,a^3}{3\tan(dx+c)^3}-\frac{a^2(3Bb+Ca)}{2\tan(dx+c)^2}+\frac{a\left(B\,a^2-3B\,b^2-3Cab\right)}{\tan(dx+c)}+\frac{\left(3B\,a^2b-B\,b^3+C\,a^3\right.}{}}{d}$
norman	$\frac{(B\,a^3-3Ba\,b^2-3C\,a^2b+C\,b^3)x\tan(dx+c)^4+\frac{a\left(B\,a^2-3B\,b^2-3Cab\right)\tan(dx+c)^3}{d}-\frac{B\,a^3\tan(dx+c)}{3d}-\frac{a^2(3Bb+Ca)\tan(dx+c)^2}{2d}}{\tan(dx+c)^4}$
risch	$B\,a^3x-3Ba\,b^2x-3C\,a^2bx+C\,b^3x-\frac{6iCa\,b^2c}{d}+\frac{6iB\,a^2bc}{d}-iB\,b^3x+iC\,a^3x-\frac{2ia\left(9iBab\,e\right.}{}$

```
[In] int(cot(d*x+c)^5*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,method=
_RETURNVERBOSE)
```

```
[Out] 1/6*(3*(3*B*a^2*b-B*b^3+C*a^3-3*C*a*b^2)*ln(sec(d*x+c)^2)+6*(-3*B*a^2*b+B*b
^3-C*a^3+3*C*a*b^2)*ln(tan(d*x+c))-2*B*a^3*cot(d*x+c)^3+3*(-3*B*a^2*b-C*a^3
)*cot(d*x+c)^2+6*a*cot(d*x+c)*(B*a^2-3*B*b^2-3*C*a*b)+6*d*x*(B*a^3-3*B*a*b^
2-3*C*a^2*b+C*b^3))/d
```

Fricas [A] (verification not implemented)

none

Time = 0.25 (sec) , antiderivative size = 181, normalized size of antiderivative = 1.18

$$\int \cot^5(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx =$$

$$-\frac{3\left(Ca^3+3\,Ba^2b-3\,Cab^2-Bb^3\right)\log\left(\frac{\tan(dx+c)^2}{\tan(dx+c)^2+1}\right)\tan\left(dx+c\right)^3+2\,Ba^3+3\left(Ca^3+3\,Ba^2b-2\left(Ba^3-\right.\right.}{6}$$

```
[In] integrate(cot(d*x+c)^5*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="fricas")
```

```
[Out] -1/6*(3*(C*a^3 + 3*B*a^2*b - 3*C*a*b^2 - B*b^3)*log(tan(d*x + c)^2/(tan(d*x
 + c)^2 + 1))*tan(d*x + c)^3 + 2*B*a^3 + 3*(C*a^3 + 3*B*a^2*b - 2*(B*a^3 -
3*C*a^2*b - 3*B*a*b^2 + C*b^3)*d*x)*tan(d*x + c)^3 - 6*(B*a^3 - 3*C*a^2*b -
 3*B*a*b^2)*tan(d*x + c)^2 + 3*(C*a^3 + 3*B*a^2*b)*tan(d*x + c))/(d*tan(d*x
 + c)^3)
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 323 vs. 2(150) = 300.

Time = 4.11 (sec) , antiderivative size = 323, normalized size of antiderivative = 2.10

$$\int \cot^5(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$

$$=\begin{cases}\text{NaN}\\ x(a+b\tan(c))^3\left(B\tan(c)+C\tan^2(c)\right)\cot^5(c)\\ \text{NaN}\\ Ba^3x+\frac{Ba^3}{d\tan(c+dx)}-\frac{Ba^3}{3d\tan^3(c+dx)}+\frac{3Ba^2b\log(\tan^2(c+dx)+1)}{2d}-\frac{3Ba^2b\log(\tan(c+dx))}{d}-\frac{3Ba^2b}{2d\tan^2(c+dx)}-3Bab^2x-\end{cases}$$

```
[In] integrate(cot(d*x+c)**5*(a+b*tan(d*x+c))**3*(B*tan(d*x+c)+C*tan(d*x+c)**2),
x)
```

```
[Out] Piecewise((nan, Eq(c, 0) & Eq(d, 0)), (x*(a + b*tan(c))**3*(B*tan(c) + C*ta
n(c)**2)*cot(c)**5, Eq(d, 0)), (nan, Eq(c, -d*x)), (B*a**3*x + B*a**3/(d*ta
n(c + d*x)) - B*a**3/(3*d*tan(c + d*x)**3) + 3*B*a**2*b*log(tan(c + d*x)**2
```

```
 + 1)/(2*d) - 3*B*a**2*b*log(tan(c + d*x))/d - 3*B*a**2*b/(2*d*tan(c + d*x)
**2) - 3*B*a*b**2*x - 3*B*a*b**2/(d*tan(c + d*x)) - B*b**3*log(tan(c + d*x)
**2 + 1)/(2*d) + B*b**3*log(tan(c + d*x))/d + C*a**3*log(tan(c + d*x)**2 +
1)/(2*d) - C*a**3*log(tan(c + d*x))/d - C*a**3/(2*d*tan(c + d*x)**2) - 3*C*
a**2*b*x - 3*C*a**2*b/(d*tan(c + d*x)) - 3*C*a*b**2*log(tan(c + d*x)**2 + 1
)/(2*d) + 3*C*a*b**2*log(tan(c + d*x))/d + C*b**3*x, True))
```

Maxima [A] (verification not implemented)

none

Time = 0.34 (sec) , antiderivative size = 180, normalized size of antiderivative = 1.17

$$\int \cot^5(c+dx)(a+b\tan(c+dx))^3 \left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx = \frac{6\,(Ba^3-3\,Ca^2b-3\,Bab^2+Cb^3)(dx+c)+3\,(Ca^3+3\,Ba^2b-3\,Cab^2-Bb^3)\log\left(\tan\left(dx+c\right)^2+1\right)-6}{}$$

```
[In] integrate(cot(d*x+c)^5*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="maxima")
```

```
[Out] 1/6*(6*(B*a^3 - 3*C*a^2*b - 3*B*a*b^2 + C*b^3)*(d*x + c) + 3*(C*a^3 + 3*B*a
^2*b - 3*C*a*b^2 - B*b^3)*log(tan(d*x + c)^2 + 1) - 6*(C*a^3 + 3*B*a^2*b -
3*C*a*b^2 - B*b^3)*log(tan(d*x + c)) - (2*B*a^3 - 6*(B*a^3 - 3*C*a^2*b - 3*
B*a*b^2)*tan(d*x + c)^2 + 3*(C*a^3 + 3*B*a^2*b)*tan(d*x + c))/tan(d*x + c)^
3)/d
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 390 vs. 2(148) = 296.

Time = 1.43 (sec) , antiderivative size = 390, normalized size of antiderivative = 2.53

$$\int \cot^5(c+dx)(a+b\tan(c+dx))^3 \left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx = \frac{Ba^3\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^3-3\,Ca^3\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^2-9\,Ba^2b\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^2-15\,Ba^3\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)+3}{}$$

```
[In] integrate(cot(d*x+c)^5*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="giac")
```

```
[Out] 1/24*(B*a^3*tan(1/2*d*x + 1/2*c)^3 - 3*C*a^3*tan(1/2*d*x + 1/2*c)^2 - 9*B*a
^2*b*tan(1/2*d*x + 1/2*c)^2 - 15*B*a^3*tan(1/2*d*x + 1/2*c) + 36*C*a^2*b*ta
n(1/2*d*x + 1/2*c) + 36*B*a*b^2*tan(1/2*d*x + 1/2*c) + 24*(B*a^3 - 3*C*a^2*
b - 3*B*a*b^2 + C*b^3)*(d*x + c) + 24*(C*a^3 + 3*B*a^2*b - 3*C*a*b^2 - B*b^
```

```
3)*log(tan(1/2*d*x + 1/2*c)^2 + 1) - 24*(C*a^3 + 3*B*a^2*b - 3*C*a*b^2 - B*
b^3)*log(abs(tan(1/2*d*x + 1/2*c))) + (44*C*a^3*tan(1/2*d*x + 1/2*c)^3 + 13
2*B*a^2*b*tan(1/2*d*x + 1/2*c)^3 - 132*C*a*b^2*tan(1/2*d*x + 1/2*c)^3 - 44*
B*b^3*tan(1/2*d*x + 1/2*c)^3 + 15*B*a^3*tan(1/2*d*x + 1/2*c)^2 - 36*C*a^2*b
*tan(1/2*d*x + 1/2*c)^2 - 36*B*a*b^2*tan(1/2*d*x + 1/2*c)^2 - 3*C*a^3*tan(1
/2*d*x + 1/2*c) - 9*B*a^2*b*tan(1/2*d*x + 1/2*c) - B*a^3)/tan(1/2*d*x + 1/2
*c)^3)/d
```

Mupad [B] (verification not implemented)

Time = 8.51 (sec) , antiderivative size = 169, normalized size of antiderivative = 1.10

$$\int \cot^5(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{\ln(\tan(c+dx))\,(-C\,a^3-3\,B\,a^2\,b+3\,C\,a\,b^2+B\,b^3)}{d}$$
$$-\frac{\cot(c+dx)^3\left(\tan(c+dx)\,\left(\frac{C\,a^3}{2}+\frac{3\,B\,b\,a^2}{2}\right)+\frac{B\,a^3}{3}+\tan(c+dx)^2\,(-B\,a^3+3\,C\,a^2\,b+3\,B\,a\,b^2)\right)}{d}$$
$$-\frac{\ln(\tan(c+dx)-\mathrm{i})\,(B+C\,1\mathrm{i})\,(a+b\,1\mathrm{i})^3\,1\mathrm{i}}{2\,d}$$
$$+\frac{\ln(\tan(c+dx)+1\mathrm{i})\,(B-C\,1\mathrm{i})\,(a-b\,1\mathrm{i})^3\,1\mathrm{i}}{2\,d}$$

```
[In] int(cot(c + d*x)^5*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))
^3,x)
```

```
[Out] (log(tan(c + d*x))*(B*b^3 - C*a^3 - 3*B*a^2*b + 3*C*a*b^2))/d - (cot(c + d*
x)^3*(tan(c + d*x)*((C*a^3)/2 + (3*B*a^2*b)/2) + (B*a^3)/3 + tan(c + d*x)^2
*(3*B*a*b^2 - B*a^3 + 3*C*a^2*b)))/d - (log(tan(c + d*x) - 1i)*(B + C*1i)*(
a + b*1i)^3*1i)/(2*d) + (log(tan(c + d*x) + 1i)*(B - C*1i)*(a - b*1i)^3*1i)
/(2*d)
```

3.23 $\int \cot^6(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan$

Optimal result 209
Rubi [A] (verified) 209
Mathematica [C] (verified) 213
Maple [A] (verified) 213
Fricas [A] (verification not implemented) 214
Sympy [B] (verification not implemented) 214
Maxima [A] (verification not implemented) 215
Giac [B] (verification not implemented) 215
Mupad [B] (verification not implemented) 216

Optimal result

Integrand size = 40, antiderivative size = 191

$$\begin{aligned}
&\int \cot^6(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx \\
&= \left(3a^2bB-b^3B+a^3C-3ab^2C\right)x+\frac{(3a^2bB-b^3B+a^3C-3ab^2C)\cot(c+dx)}{d} \\
&\quad+\frac{a(2a^2B-5b^2B-6abC)\cot^2(c+dx)}{4d}-\frac{a^2(3bB+2aC)\cot^3(c+dx)}{6d} \\
&\quad+\frac{(a^3B-3ab^2B-3a^2bC+b^3C)\log(\sin(c+dx))}{d} \\
&\quad-\frac{aB\cot^4(c+dx)(a+b\tan(c+dx))^2}{4d}
\end{aligned}$$

```
[Out] (3*B*a^2*b-B*b^3+C*a^3-3*C*a*b^2)*x+(3*B*a^2*b-B*b^3+C*a^3-3*C*a*b^2)*cot(d
*x+c)/d+1/4*a*(2*B*a^2-5*B*b^2-6*C*a*b)*cot(d*x+c)^2/d-1/6*a^2*(3*B*b+2*C*a
)*cot(d*x+c)^3/d+(B*a^3-3*B*a*b^2-3*C*a^2*b+C*b^3)*ln(sin(d*x+c))/d-1/4*a*B
*cot(d*x+c)^4*(a+b*tan(d*x+c))^2/d
```

Rubi [A] (verified)

Time = 0.63 (sec) , antiderivative size = 191, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 7, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.175, Rules used

$= \{3713, 3686, 3716, 3709, 3610, 3612, 3556\}$

$$\int \cot^6(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$= \frac{a(2a^2B-6abC-5b^2B)\cot^2(c+dx)}{4d}-\frac{a^2(2aC+3bB)\cot^3(c+dx)}{6d}$$
$$+\frac{(a^3C+3a^2bB-3ab^2C-b^3B)\cot(c+dx)}{d}$$
$$+\frac{(a^3B-3a^2bC-3ab^2B+b^3C)\log(\sin(c+dx))}{d}$$
$$+x\left(a^3C+3a^2bB-3ab^2C-b^3B\right)-\frac{aB\cot^4(c+dx)(a+b\tan(c+dx))^2}{4d}$$

```
[In] Int[Cot[c + d*x]^6*(a + b*Tan[c + d*x])^3*(B*Tan[c + d*x] + C*Tan[c + d*x]^
2),x]
```

```
[Out] (3*a^2*b*B - b^3*B + a^3*C - 3*a*b^2*C)*x + ((3*a^2*b*B - b^3*B + a^3*C - 3
*a*b^2*C)*Cot[c + d*x])/d + (a*(2*a^2*B - 5*b^2*B - 6*a*b*C)*Cot[c + d*x]^2
)/(4*d) - (a^2*(3*b*B + 2*a*C)*Cot[c + d*x]^3)/(6*d) + ((a^3*B - 3*a*b^2*B
- 3*a^2*b*C + b^3*C)*Log[Sin[c + d*x]])/d - (a*B*Cot[c + d*x]^4*(a + b*Tan[
c + d*x])^2)/(4*d)
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3610

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(b*c - a*d)*((a + b*Tan[e + f*x])^(m + 1)/
(f*(m + 1)*(a^2 + b^2))), x] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])
^(m + 1)*Simp[a*c + b*d - (b*c - a*d)*Tan[e + f*x], x], x], x] /; FreeQ[{a,
 b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && LtQ[m, -1]
```

Rule 3612

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[(a*c + b*d)*(x/(a^2 + b^2)), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3686

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Si
```

```
mp[(b*c - a*d)*(B*c - A*d)*(a + b*Tan[e + f*x])^(m - 1)*((c + d*Tan[e + f*x
])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dist[1/(d*(n + 1)*(c^2 + d^2)),
 Int[(a + b*Tan[e + f*x])^(m - 2)*(c + d*Tan[e + f*x])^(n + 1)*Simp[a*A*d*(
b*d*(m - 1) - a*c*(n + 1)) + (b*B*c - (A*b + a*B)*d)*(b*c*(m - 1) + a*d*(n
+ 1)) - d*((a*A - b*B)*(b*c - a*d) + (A*b + a*B)*(a*c + b*d))*(n + 1)*Tan[e
 + f*x] - b*(d*(A*b*c + a*B*c - a*A*d)*(m + n) - b*B*(c^2*(m - 1) - d^2*(n
+ 1)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] &&
NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 1] &&
 LtQ[n, -1] && (IntegerQ[m] || IntegersQ[2*m, 2*n])
```

Rule 3709

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 -
 a*b*B + a^2*C)*((a + b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1)*(a^2 + b^2))), x
] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[b*B + a*(A -
C) - (A*b - a*B - b*C)*Tan[e + f*x], x], x], x] /; FreeQ[{a, b, e, f, A, B,
 C}, x] && NeQ[A*b^2 - a*b*B + a^2*C, 0] && LtQ[m, -1] && NeQ[a^2 + b^2, 0]
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rule 3716

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.
)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(-(b*c - a*d))*(c^2*C - B*c*d + A*d^2)*((c
+ d*Tan[e + f*x])^(n + 1)/(d^2*f*(n + 1)*(c^2 + d^2))), x] + Dist[1/(d*(c^2
 + d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)*Simp[a*d*(A*c - c*C + B*d) + b*(
c^2*C - B*c*d + A*d^2) + d*(A*b*c + a*B*c - b*c*C - a*A*d + b*B*d + a*C*d)*
Tan[e + f*x] + b*C*(c^2 + d^2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c
, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] && LtQ[n,
 -1]
```

Rubi steps

$$\text{integral} = \int \cot^5(c+dx)(a+b\tan(c+dx))^3(B+C\tan(c+dx))\,dx$$

$$= -\frac{aB\cot^4(c+dx)(a+b\tan(c+dx))^2}{4d} + \frac{1}{4}\int \cot^4(c+dx)(a + b\tan(c+dx))\left(2a(3bB+2aC) - 4\left(a^2B - b^2B - 2abC\right)\tan(c+dx) - 2b(aB-2bC)\tan^2(c+dx)\right)dx$$

$$= -\frac{a^2(3bB+2aC)\cot^3(c+dx)}{6d} - \frac{aB\cot^4(c+dx)(a+b\tan(c+dx))^2}{4d} + \frac{1}{4}\int \cot^3(c+dx)\left(-2a\left(2a^2B-5b^2B-6abC\right) - 4\left(3a^2bB-b^3B+a^3C-3ab^2C\right)\tan(c+dx) - 2b^2(aB-2bC)\tan^2(c+dx)\right)dx$$

$$= \frac{a(2a^2B-5b^2B-6abC)\cot^2(c+dx)}{4d} - \frac{a^2(3bB+2aC)\cot^3(c+dx)}{6d} - \frac{aB\cot^4(c+dx)(a+b\tan(c+dx))^2}{4d} + \frac{1}{4}\int \cot^2(c+dx)\left(-4\left(3a^2bB-b^3B+a^3C-3ab^2C\right) + 4\left(a^3B-3ab^2B-3a^2bC+b^3C\right)\tan(c+dx)\right)dx$$

$$= \frac{(3a^2bB-b^3B+a^3C-3ab^2C)\cot(c+dx)}{d} + \frac{a(2a^2B-5b^2B-6abC)\cot^2(c+dx)}{4d} - \frac{a^2(3bB+2aC)\cot^3(c+dx)}{6d} - \frac{aB\cot^4(c+dx)(a+b\tan(c+dx))^2}{4d} + \frac{1}{4}\int \cot(c+dx)\left(4\left(a^3B-3ab^2B-3a^2bC+b^3C\right) + 4\left(3a^2bB-b^3B+a^3C-3ab^2C\right)\tan(c+dx)\right)dx$$

$$= \left(3a^2bB-b^3B+a^3C-3ab^2C\right)x + \frac{(3a^2bB-b^3B+a^3C-3ab^2C)\cot(c+dx)}{d} + \frac{a(2a^2B-5b^2B-6abC)\cot^2(c+dx)}{4d} - \frac{a^2(3bB+2aC)\cot^3(c+dx)}{6d} - \frac{aB\cot^4(c+dx)(a+b\tan(c+dx))^2}{4d} + \left(a^3B-3ab^2B-3a^2bC+b^3C\right)\int \cot(c+dx)\,dx$$

$$= \left(3a^2bB-b^3B+a^3C-3ab^2C\right)x + \frac{(3a^2bB-b^3B+a^3C-3ab^2C)\cot(c+dx)}{d} + \frac{a(2a^2B-5b^2B-6abC)\cot^2(c+dx)}{4d} - \frac{a^2(3bB+2aC)\cot^3(c+dx)}{6d} + \frac{(a^3B-3ab^2B-3a^2bC+b^3C)\log(\sin(c+dx))}{d} - \frac{aB\cot^4(c+dx)(a+b\tan(c+dx))^2}{4d}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 0.81 (sec) , antiderivative size = 199, normalized size of antiderivative = 1.04

$$\int \cot^6(c+dx)(a+b\tan(c+dx))^3 \left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$= \frac{12(3a^2bB-b^3B+a^3C-3ab^2C)\cot(c+dx)+6a(a^2B-3b^2B-3abC)\cot^2(c+dx)-4a^2(3bB+aC)\,\mathrm{c}}{}$$

```
[In] Integrate[Cot[c + d*x]^6*(a + b*Tan[c + d*x])^3*(B*Tan[c + d*x] + C*Tan[c +
 d*x]^2),x]
```

```
[Out] (12*(3*a^2*b*B - b^3*B + a^3*C - 3*a*b^2*C)*Cot[c + d*x] + 6*a*(a^2*B - 3*b
^2*B - 3*a*b*C)*Cot[c + d*x]^2 - 4*a^2*(3*b*B + a*C)*Cot[c + d*x]^3 - 3*a^3
*B*Cot[c + d*x]^4 - 6*(a + I*b)^3*(B + I*C)*Log[I - Tan[c + d*x]] + 12*(a^3
*B - 3*a*b^2*B - 3*a^2*b*C + b^3*C)*Log[Tan[c + d*x]] - 6*(a - I*b)^3*(B -
I*C)*Log[I + Tan[c + d*x]])/(12*d)
```

Maple [A] (verified)

Time = 0.31 (sec) , antiderivative size = 209, normalized size of antiderivative = 1.09

method	result
parallelrisch	$\frac{6\left(-B\,a^3+3Ba\,b^2+3C\,a^2b-C\,b^3\right)\ln\left(\sec(dx+c)^2\right)+12\left(B\,a^3-3Ba\,b^2-3C\,a^2b+C\,b^3\right)\ln(\tan(dx+c))-3B\,a^3\cot(dx+c)^4}{}$
derivativedivides	$\frac{-\frac{-3B\,a^2b+B\,b^3-C\,a^3+3Ca\,b^2}{\tan(dx+c)}+\left(B\,a^3-3Ba\,b^2-3C\,a^2b+C\,b^3\right)\ln(\tan(dx+c))-\frac{B\,a^3}{4\tan(dx+c)^4}-\frac{a^2(3Bb+Ca)}{3\tan(dx+c)^3}+\frac{a\left(B\,a^2-3B\,b\right.}{2\tan(dx}}{d}$
default	$\frac{-\frac{-3B\,a^2b+B\,b^3-C\,a^3+3Ca\,b^2}{\tan(dx+c)}+\left(B\,a^3-3Ba\,b^2-3C\,a^2b+C\,b^3\right)\ln(\tan(dx+c))-\frac{B\,a^3}{4\tan(dx+c)^4}-\frac{a^2(3Bb+Ca)}{3\tan(dx+c)^3}+\frac{a\left(B\,a^2-3B\,b\right.}{2\tan(dx}}{d}$
norman	$\frac{\frac{\left(3B\,a^2b-B\,b^3+C\,a^3-3Ca\,b^2\right)\tan(dx+c)^4}{d}+\left(3B\,a^2b-B\,b^3+C\,a^3-3Ca\,b^2\right)x\tan(dx+c)^5-\frac{B\,a^3\tan(dx+c)}{4d}+\frac{a\left(B\,a^2-3B\,b^2-3\right.}{2}}{\tan(dx+c)^5}$
risch	$\frac{6iC\,a^2bc}{d}+\frac{6iBa\,b^2c}{d}-iB\,a^3x+3iC\,a^2bx+3B\,a^2bx-B\,b^3x+C\,a^3x-3Ca\,b^2x-\frac{2iC\,b^3c}{d}$

```
[In] int(cot(d*x+c)^6*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,method=
_RETURNVERBOSE)
```

```
[Out] 1/12*(6*(-B*a^3+3*B*a*b^2+3*C*a^2*b-C*b^3)*ln(sec(d*x+c)^2)+12*(B*a^3-3*B*a
*b^2-3*C*a^2*b+C*b^3)*ln(tan(d*x+c))-3*B*a^3*cot(d*x+c)^4+4*(-3*B*a^2*b-C*a
^3)*cot(d*x+c)^3+6*a*cot(d*x+c)^2*(B*a^2-3*B*b^2-3*C*a*b)+12*cot(d*x+c)*(3*
B*a^2*b-B*b^3+C*a^3-3*C*a*b^2)+36*d*(B*a^2*b-1/3*B*b^3+1/3*C*a^3-C*a*b^2)*x
)/d
```

Fricas [A] (verification not implemented)

none

Time = 0.26 (sec) , antiderivative size = 225, normalized size of antiderivative = 1.18

$$\int \cot^6(c+dx)(a+b\tan(c+dx))^3 \left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$= \frac{6\,(Ba^3-3\,Ca^2b-3\,Bab^2+Cb^3)\log\left(\frac{\tan(dx+c)^2}{\tan(dx+c)^2+1}\right)\tan{(dx+c)}^4+3\,(3\,Ba^3-6\,Ca^2b-6\,Bab^2+4\,(Ca^3+$$

```
[In] integrate(cot(d*x+c)^6*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="fricas")
```

```
[Out] 1/12*(6*(B*a^3 - 3*C*a^2*b - 3*B*a*b^2 + C*b^3)*log(tan(d*x + c)^2/(tan(d*x
 + c)^2 + 1))*tan(d*x + c)^4 + 3*(3*B*a^3 - 6*C*a^2*b - 6*B*a*b^2 + 4*(C*a^
3 + 3*B*a^2*b - 3*C*a*b^2 - B*b^3)*d*x)*tan(d*x + c)^4 - 3*B*a^3 + 12*(C*a^
3 + 3*B*a^2*b - 3*C*a*b^2 - B*b^3)*tan(d*x + c)^3 + 6*(B*a^3 - 3*C*a^2*b -
3*B*a*b^2)*tan(d*x + c)^2 - 4*(C*a^3 + 3*B*a^2*b)*tan(d*x + c))/(d*tan(d*x
+ c)^4)
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 391 vs. 2(187) = 374.

Time = 5.46 (sec) , antiderivative size = 391, normalized size of antiderivative = 2.05

$$\int \cot^6(c+dx)(a+b\tan(c+dx))^3 \left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$= \begin{cases} \text{NaN} \\ x(a+b\tan{(c)})^3 \left(B\tan{(c)}+C\tan^2{(c)}\right)\cot^6{(c)} \\ \text{NaN} \\ -\frac{Ba^3\log{(\tan^2{(c+dx)}+1)}}{2d}+\frac{Ba^3\log{(\tan{(c+dx)})}}{d}+\frac{Ba^3}{2d\tan^2{(c+dx)}}-\frac{Ba^3}{4d\tan^4{(c+dx)}}+3Ba^2bx+\frac{3Ba^2b}{d\tan{(c+dx)}}-\frac{Ba^2b}{d\tan^3{(c+dx)}} \end{cases}$$

```
[In] integrate(cot(d*x+c)**6*(a+b*tan(d*x+c))**3*(B*tan(d*x+c)+C*tan(d*x+c)**2),
x)
```

```
[Out] Piecewise((nan, Eq(c, 0) & Eq(d, 0)), (x*(a + b*tan(c))**3*(B*tan(c) + C*ta
n(c)**2)*cot(c)**6, Eq(d, 0)), (nan, Eq(c, -d*x)), (-B*a**3*log(tan(c + d*x
)**2 + 1)/(2*d) + B*a**3*log(tan(c + d*x))/d + B*a**3/(2*d*tan(c + d*x)**2)
 - B*a**3/(4*d*tan(c + d*x)**4) + 3*B*a**2*b*x + 3*B*a**2*b/(d*tan(c + d*x)
) - B*a**2*b/(d*tan(c + d*x)**3) + 3*B*a*b**2*log(tan(c + d*x)**2 + 1)/(2*d
) - 3*B*a*b**2*log(tan(c + d*x))/d - 3*B*a*b**2/(2*d*tan(c + d*x)**2) - B*b
**3*x - B*b**3/(d*tan(c + d*x)) + C*a**3*x + C*a**3/(d*tan(c + d*x)) - C*a*
```

```
*3/(3*d*tan(c + d*x)**3) + 3*C*a**2*b*log(tan(c + d*x)**2 + 1)/(2*d) - 3*C*
a**2*b*log(tan(c + d*x))/d - 3*C*a**2*b/(2*d*tan(c + d*x)**2) - 3*C*a*b**2*
x - 3*C*a*b**2/(d*tan(c + d*x)) - C*b**3*log(tan(c + d*x)**2 + 1)/(2*d) + C
*b**3*log(tan(c + d*x))/d, True))
```

Maxima [A] (verification not implemented)

none

Time = 0.32 (sec) , antiderivative size = 215, normalized size of antiderivative = 1.13

$$\int \cot^6(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{12\,(Ca^3+3\,Ba^2b-3\,Cab^2-Bb^3)(dx+c)-6\,(Ba^3-3\,Ca^2b-3\,Bab^2+Cb^3)\log\left(\tan\left(dx+c\right)^2+1\right)+\cdots}{\cdots}$$

```
[In] integrate(cot(d*x+c)^6*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="maxima")
```

```
[Out] 1/12*(12*(C*a^3 + 3*B*a^2*b - 3*C*a*b^2 - B*b^3)*(d*x + c) - 6*(B*a^3 - 3*C
*a^2*b - 3*B*a*b^2 + C*b^3)*log(tan(d*x + c)^2 + 1) + 12*(B*a^3 - 3*C*a^2*b
 - 3*B*a*b^2 + C*b^3)*log(tan(d*x + c)) - (3*B*a^3 - 12*(C*a^3 + 3*B*a^2*b
- 3*C*a*b^2 - B*b^3)*tan(d*x + c)^3 - 6*(B*a^3 - 3*C*a^2*b - 3*B*a*b^2)*tan
(d*x + c)^2 + 4*(C*a^3 + 3*B*a^2*b)*tan(d*x + c))/tan(d*x + c)^4)/d
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 528 vs. 2(185) = 370.

Time = 1.48 (sec) , antiderivative size = 528, normalized size of antiderivative = 2.76

$$\int \cot^6(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx=$$
$$-\frac{3\,Ba^3\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^4-8\,Ca^3\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^3-24\,Ba^2b\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^3-36\,Ba^3\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)\cdots}{\cdots}$$

```
[In] integrate(cot(d*x+c)^6*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="giac")
```

```
[Out] -1/192*(3*B*a^3*tan(1/2*d*x + 1/2*c)^4 - 8*C*a^3*tan(1/2*d*x + 1/2*c)^3 - 2
4*B*a^2*b*tan(1/2*d*x + 1/2*c)^3 - 36*B*a^3*tan(1/2*d*x + 1/2*c)^2 + 72*C*a
^2*b*tan(1/2*d*x + 1/2*c)^2 + 72*B*a*b^2*tan(1/2*d*x + 1/2*c)^2 + 120*C*a^3
*tan(1/2*d*x + 1/2*c) + 360*B*a^2*b*tan(1/2*d*x + 1/2*c) - 288*C*a*b^2*tan(
1/2*d*x + 1/2*c) - 96*B*b^3*tan(1/2*d*x + 1/2*c) - 192*(C*a^3 + 3*B*a^2*b -
 3*C*a*b^2 - B*b^3)*(d*x + c) + 192*(B*a^3 - 3*C*a^2*b - 3*B*a*b^2 + C*b^3)
```

```
*log(tan(1/2*d*x + 1/2*c)^2 + 1) - 192*(B*a^3 - 3*C*a^2*b - 3*B*a*b^2 + C*b
^3)*log(abs(tan(1/2*d*x + 1/2*c))) + (400*B*a^3*tan(1/2*d*x + 1/2*c)^4 - 12
00*C*a^2*b*tan(1/2*d*x + 1/2*c)^4 - 1200*B*a*b^2*tan(1/2*d*x + 1/2*c)^4 + 4
00*C*b^3*tan(1/2*d*x + 1/2*c)^4 - 120*C*a^3*tan(1/2*d*x + 1/2*c)^3 - 360*B*
a^2*b*tan(1/2*d*x + 1/2*c)^3 + 288*C*a*b^2*tan(1/2*d*x + 1/2*c)^3 + 96*B*b^
3*tan(1/2*d*x + 1/2*c)^3 - 36*B*a^3*tan(1/2*d*x + 1/2*c)^2 + 72*C*a^2*b*tan
(1/2*d*x + 1/2*c)^2 + 72*B*a*b^2*tan(1/2*d*x + 1/2*c)^2 + 8*C*a^3*tan(1/2*d
*x + 1/2*c) + 24*B*a^2*b*tan(1/2*d*x + 1/2*c) + 3*B*a^3)/tan(1/2*d*x + 1/2*
c)^4)/d
```

Mupad [B] (verification not implemented)

Time = 8.47 (sec) , antiderivative size = 204, normalized size of antiderivative = 1.07

$$\int \cot^6(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$= \frac{\ln\left(\tan(c+d\,x)\right)\,\left(B\,a^3-3\,C\,a^2\,b-3\,B\,a\,b^2+C\,b^3\right)}{d}$$
$$-\frac{\cot(c+d\,x)^4\left(\tan(c+d\,x)\,\left(\frac{C\,a^3}{3}+B\,b\,a^2\right)+\frac{B\,a^3}{4}+\tan(c+d\,x)^2\left(-\frac{B\,a^3}{2}+\frac{3\,C\,a^2\,b}{2}+\frac{3\,B\,a\,b^2}{2}\right)+\tan(c+\right.}{d}$$
$$-\frac{\ln\left(\tan(c+d\,x)+1\mathrm{i}\right)\,\left(B-C\,1\mathrm{i}\right)\,\left(b+a\,1\mathrm{i}\right)^3\,1\mathrm{i}}{2\,d}$$
$$-\frac{\ln\left(\tan(c+d\,x)-\mathrm{i}\right)\,\left(B+C\,1\mathrm{i}\right)\,\left(-b+a\,1\mathrm{i}\right)^3\,1\mathrm{i}}{2\,d}$$

```
[In] int(cot(c + d*x)^6*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))
^3,x)
```

```
[Out] (log(tan(c + d*x))*(B*a^3 + C*b^3 - 3*B*a*b^2 - 3*C*a^2*b))/d - (cot(c + d*
x)^4*(tan(c + d*x)*((C*a^3)/3 + B*a^2*b) + (B*a^3)/4 + tan(c + d*x)^2*((3*B
*a*b^2)/2 - (B*a^3)/2 + (3*C*a^2*b)/2) + tan(c + d*x)^3*(B*b^3 - C*a^3 - 3*
B*a^2*b + 3*C*a*b^2)))/d - (log(tan(c + d*x) + 1i)*(B - C*1i)*(a*1i + b)^3*
1i)/(2*d) - (log(tan(c + d*x) - 1i)*(B + C*1i)*(a*1i - b)^3*1i)/(2*d)
```

3.24 $\int \cot^7(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)dx$

Optimal result 217
Rubi [A] (verified) 217
Mathematica [C] (verified) 221
Maple [A] (verified) 222
Fricas [A] (verification not implemented) 222
Sympy [A] (verification not implemented) 223
Maxima [A] (verification not implemented) 223
Giac [B] (verification not implemented) 224
Mupad [B] (verification not implemented) 225

Optimal result

Integrand size = 40, antiderivative size = 233

$$\begin{aligned}
&\int \cot^7(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx \\
&= -\left(\left(a^3B-3ab^2B-3a^2bC+b^3C\right)x\right)-\frac{\left(a^3B-3ab^2B-3a^2bC+b^3C\right)\cot(c+dx)}{d} \\
&\quad+\frac{\left(3a^2bB-b^3B+a^3C-3ab^2C\right)\cot^2(c+dx)}{2d}+\frac{a\left(5a^2B-12b^2B-15abC\right)\cot^3(c+dx)}{15d} \\
&\quad-\frac{a^2(7bB+5aC)\cot^4(c+dx)}{20d}+\frac{\left(3a^2bB-b^3B+a^3C-3ab^2C\right)\log(\sin(c+dx))}{d} \\
&\quad-\frac{aB\cot^5(c+dx)(a+b\tan(c+dx))^2}{5d}
\end{aligned}$$

```
[Out] -(B*a^3-3*B*a*b^2-3*C*a^2*b+C*b^3)*x-(B*a^3-3*B*a*b^2-3*C*a^2*b+C*b^3)*cot(
d*x+c)/d+1/2*(3*B*a^2*b-B*b^3+C*a^3-3*C*a*b^2)*cot(d*x+c)^2/d+1/15*a*(5*B*a
^2-12*B*b^2-15*C*a*b)*cot(d*x+c)^3/d-1/20*a^2*(7*B*b+5*C*a)*cot(d*x+c)^4/d+
(3*B*a^2*b-B*b^3+C*a^3-3*C*a*b^2)*ln(sin(d*x+c))/d-1/5*a*B*cot(d*x+c)^5*(a+
b*tan(d*x+c))^2/d
```

Rubi [A] (verified)

Time = 0.70 (sec) , antiderivative size = 233, normalized size of antiderivative = 1.00, number of steps used = 8, number of rules used = 7, $\frac{\text{number of rules}}{\text{integrand size}} = 0.175$, Rules used

$= \{3713, 3686, 3716, 3709, 3610, 3612, 3556\}$

$$\int \cot^7(c+dx)(a+b\tan(c+dx))^3 \left(B\tan(c+dx)+C\tan^2(c+dx)\right) dx$$
$$= \frac{a(5a^2B-15abC-12b^2B)\cot^3(c+dx)}{15d} - \frac{a^2(5aC+7bB)\cot^4(c+dx)}{20d} + \frac{(a^3C+3a^2bB-3ab^2C-b^3B)\cot^2(c+dx)}{2d} - \frac{(a^3B-3a^2bC-3ab^2B+b^3C)\cot(c+dx)}{d} + \frac{(a^3C+3a^2bB-3ab^2C-b^3B)\log(\sin(c+dx))}{d} - x\left(a^3B-3a^2bC-3ab^2B+b^3C\right) - \frac{aB\cot^5(c+dx)(a+b\tan(c+dx))^2}{5d}$$

```
[In] Int[Cot[c + d*x]^7*(a + b*Tan[c + d*x])^3*(B*Tan[c + d*x] + C*Tan[c + d*x]^
2),x]
```

```
[Out] -((a^3*B - 3*a*b^2*B - 3*a^2*b*C + b^3*C)*x) - ((a^3*B - 3*a*b^2*B - 3*a^2*
b*C + b^3*C)*Cot[c + d*x])/d + ((3*a^2*b*B - b^3*B + a^3*C - 3*a*b^2*C)*Cot
[c + d*x]^2)/(2*d) + (a*(5*a^2*B - 12*b^2*B - 15*a*b*C)*Cot[c + d*x]^3)/(15
*d) - (a^2*(7*b*B + 5*a*C)*Cot[c + d*x]^4)/(20*d) + ((3*a^2*b*B - b^3*B + a
^3*C - 3*a*b^2*C)*Log[Sin[c + d*x]])/d - (a*B*Cot[c + d*x]^5*(a + b*Tan[c +
 d*x])^2)/(5*d)
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3610

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(b*c - a*d)*((a + b*Tan[e + f*x])^(m + 1)/
(f*(m + 1)*(a^2 + b^2))), x] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])
^(m + 1)*Simp[a*c + b*d - (b*c - a*d)*Tan[e + f*x], x], x], x] /; FreeQ[{a,
 b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && LtQ[m, -1]
```

Rule 3612

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[(a*c + b*d)*(x/(a^2 + b^2)), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3686

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Si
mp[(b*c - a*d)*(B*c - A*d)*(a + b*Tan[e + f*x])^(m - 1)*((c + d*Tan[e + f*x
])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dist[1/(d*(n + 1)*(c^2 + d^2)),
 Int[(a + b*Tan[e + f*x])^(m - 2)*(c + d*Tan[e + f*x])^(n + 1)*Simp[a*A*d*(
b*d*(m - 1) - a*c*(n + 1)) + (b*B*c - (A*b + a*B)*d)*(b*c*(m - 1) + a*d*(n
+ 1)) - d*((a*A - b*B)*(b*c - a*d) + (A*b + a*B)*(a*c + b*d))*(n + 1)*Tan[e
 + f*x] - b*(d*(A*b*c + a*B*c - a*A*d)*(m + n) - b*B*(c^2*(m - 1) - d^2*(n
+ 1)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] &&
NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 1] &&
 LtQ[n, -1] && (IntegerQ[m] || IntegersQ[2*m, 2*n])
```

Rule 3709

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 -
 a*b*B + a^2*C)*((a + b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1)*(a^2 + b^2))), x
] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[b*B + a*(A -
C) - (A*b - a*B - b*C)*Tan[e + f*x], x], x], x] /; FreeQ[{a, b, e, f, A, B,
 C}, x] && NeQ[A*b^2 - a*b*B + a^2*C, 0] && LtQ[m, -1] && NeQ[a^2 + b^2, 0]
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rule 3716

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.
)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(-(b*c - a*d))*(c^2*C - B*c*d + A*d^2)*((c
+ d*Tan[e + f*x])^(n + 1)/(d^2*f*(n + 1)*(c^2 + d^2))), x] + Dist[1/(d*(c^2
 + d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)*Simp[a*d*(A*c - c*C + B*d) + b*(
c^2*C - B*c*d + A*d^2) + d*(A*b*c + a*B*c - b*c*C - a*A*d + b*B*d + a*C*d)*
Tan[e + f*x] + b*C*(c^2 + d^2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c
, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] && LtQ[n,
 -1]
```

Rubi steps

$$\text{integral} = \int \cot^6(c+dx)(a+b\tan(c+dx))^3(B+C\tan(c+dx))\,dx$$

$$= -\frac{aB\cot^5(c+dx)(a+b\tan(c+dx))^2}{5d}+\frac{1}{5}\int \cot^5(c+dx)(a+b\tan(c+dx))\left(a(7bB+5aC)-5\left(a^2B-b^2B-2abC\right)\tan(c+dx)-b(3aB-5bC)\tan^2(c+dx)\right)\,dx$$

$$= -\frac{a^2(7bB+5aC)\cot^4(c+dx)}{20d}-\frac{aB\cot^5(c+dx)(a+b\tan(c+dx))^2}{5d}+\frac{1}{5}\int \cot^4(c+dx)\left(-a\left(5a^2B-12b^2B-15abC\right)-5\left(3a^2bB-b^3B+a^3C-3ab^2C\right)\tan(c+dx)-b^2(3aB-5bC)\tan^2(c+dx)\right)\,dx$$

$$= \frac{a(5a^2B-12b^2B-15abC)\cot^3(c+dx)}{15d}-\frac{a^2(7bB+5aC)\cot^4(c+dx)}{20d}-\frac{aB\cot^5(c+dx)(a+b\tan(c+dx))^2}{5d}+\frac{1}{5}\int \cot^3(c+dx)\left(-5\left(3a^2bB-b^3B+a^3C-3ab^2C\right)+5\left(a^3B-3ab^2B-3a^2bC+b^3C\right)\tan(c+dx)\right)\,dx$$

$$= \frac{(3a^2bB-b^3B+a^3C-3ab^2C)\cot^2(c+dx)}{2d}+\frac{a(5a^2B-12b^2B-15abC)\cot^3(c+dx)}{15d}-\frac{a^2(7bB+5aC)\cot^4(c+dx)}{20d}-\frac{aB\cot^5(c+dx)(a+b\tan(c+dx))^2}{5d}+\frac{1}{5}\int \cot^2(c+dx)\left(5\left(a^3B-3ab^2B-3a^2bC+b^3C\right)+5\left(3a^2bB-b^3B+a^3C-3ab^2C\right)\tan(c+dx)\right)\,dx$$

$$= -\frac{(a^3B-3ab^2B-3a^2bC+b^3C)\cot(c+dx)}{d}+\frac{(3a^2bB-b^3B+a^3C-3ab^2C)\cot^2(c+dx)}{2d}+\frac{a(5a^2B-12b^2B-15abC)\cot^3(c+dx)}{15d}-\frac{a^2(7bB+5aC)\cot^4(c+dx)}{20d}-\frac{aB\cot^5(c+dx)(a+b\tan(c+dx))^2}{5d}+\frac{1}{5}\int \cot(c+dx)\left(5\left(3a^2bB-b^3B+a^3C-3ab^2C\right)-5\left(a^3B-3ab^2B-3a^2bC+b^3C\right)\tan(c+dx)\right)\,dx$$

$$
\begin{aligned}
&= -\left(\left(a^3B - 3ab^2B - 3a^2bC + b^3C\right)x\right) - \frac{(a^3B - 3ab^2B - 3a^2bC + b^3C)\cot(c+dx)}{d} \\
&\quad + \frac{(3a^2bB - b^3B + a^3C - 3ab^2C)\cot^2(c+dx)}{2d} \\
&\quad + \frac{a(5a^2B - 12b^2B - 15abC)\cot^3(c+dx)}{15d} \\
&\quad - \frac{a^2(7bB + 5aC)\cot^4(c+dx)}{20d} - \frac{aB\cot^5(c+dx)(a+b\tan(c+dx))^2}{5d} \\
&\quad + \left(3a^2bB - b^3B + a^3C - 3ab^2C\right)\int \cot(c+dx)\,dx
\end{aligned}
$$

$$
\begin{aligned}
&= -\left(\left(a^3B - 3ab^2B - 3a^2bC + b^3C\right)x\right) - \frac{(a^3B - 3ab^2B - 3a^2bC + b^3C)\cot(c+dx)}{d} \\
&\quad + \frac{(3a^2bB - b^3B + a^3C - 3ab^2C)\cot^2(c+dx)}{2d} \\
&\quad + \frac{a(5a^2B - 12b^2B - 15abC)\cot^3(c+dx)}{15d} - \frac{a^2(7bB + 5aC)\cot^4(c+dx)}{20d} \\
&\quad + \frac{(3a^2bB - b^3B + a^3C - 3ab^2C)\log(\sin(c+dx))}{d} \\
&\quad - \frac{aB\cot^5(c+dx)(a+b\tan(c+dx))^2}{5d}
\end{aligned}
$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 1.24 (sec) , antiderivative size = 237, normalized size of antiderivative = 1.02

$$
\int \cot^7(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx) + C\tan^2(c+dx)\right)\,dx
$$

$$
= \frac{-60(a^3B - 3ab^2B - 3a^2bC + b^3C)\cot(c+dx) + 30(3a^2bB - b^3B + a^3C - 3ab^2C)\cot^2(c+dx) + 20a(a
$$

```
[In] Integrate[Cot[c + d*x]^7*(a + b*Tan[c + d*x])^3*(B*Tan[c + d*x] + C*Tan[c +
 d*x]^2),x]
```

```
[Out] (-60*(a^3*B - 3*a*b^2*B - 3*a^2*b*C + b^3*C)*Cot[c + d*x] + 30*(3*a^2*b*B -
 b^3*B + a^3*C - 3*a*b^2*C)*Cot[c + d*x]^2 + 20*a*(a^2*B - 3*b^2*B - 3*a*b*
C)*Cot[c + d*x]^3 - 15*a^2*(3*b*B + a*C)*Cot[c + d*x]^4 - 12*a^3*B*Cot[c +
d*x]^5 + (30*I)*(a + I*b)^3*(B + I*C)*Log[I - Tan[c + d*x]] + 60*(3*a^2*b*B
 - b^3*B + a^3*C - 3*a*b^2*C)*Log[Tan[c + d*x]] + 30*(I*a + b)^3*(B - I*C)*
Log[I + Tan[c + d*x]])/(60*d)
```

Maple [A] (verified)

Time = 0.36 (sec) , antiderivative size = 243, normalized size of antiderivative = 1.04

method	result
parallelrisch	$\frac{\left(-90B\,a^2b+30B\,b^3-30C\,a^3+90Ca\,b^2\right)\ln\left(\sec(dx+c)^2\right)+\left(180B\,a^2b-60B\,b^3+60C\,a^3-180Ca\,b^2\right)\ln(\tan(dx+c))-12B\,a^3}{}$
derivativedivides	$\frac{-\frac{-3B\,a^2b+B\,b^3-C\,a^3+3Ca\,b^2}{2\tan(dx+c)^2}+\left(3B\,a^2b-B\,b^3+C\,a^3-3Ca\,b^2\right)\ln(\tan(dx+c))-\frac{B\,a^3-3Ba\,b^2-3C\,a^2b+C\,b^3}{\tan(dx+c)}-\frac{B\,a^3}{5\tan(dx+c)^5}-}{}$
default	$\frac{-\frac{-3B\,a^2b+B\,b^3-C\,a^3+3Ca\,b^2}{2\tan(dx+c)^2}+\left(3B\,a^2b-B\,b^3+C\,a^3-3Ca\,b^2\right)\ln(\tan(dx+c))-\frac{B\,a^3-3Ba\,b^2-3C\,a^2b+C\,b^3}{\tan(dx+c)}-\frac{B\,a^3}{5\tan(dx+c)^5}-}{}$
norman	$\frac{\left(-B\,a^3+3Ba\,b^2+3C\,a^2b-C\,b^3\right)x\tan(dx+c)^6+\frac{\left(3B\,a^2b-B\,b^3+C\,a^3-3Ca\,b^2\right)\tan(dx+c)^4}{2d}-\frac{\left(B\,a^3-3Ba\,b^2-3C\,a^2b+C\,b^3\right)\tan(}{d}}{\tan(dx+c)^6}$
risch	$-B\,a^3x+3Ba\,b^2x+3C\,a^2bx-C\,b^3x-\frac{6iB\,a^2bc}{d}-\frac{2i\left(-60C\,a^2b-60Ba\,b^2+15C\,b^3+23B\,a^3-70B\,a^3\mathrm{e}^{2}\right.}{}$

```
[In] int(cot(d*x+c)^7*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,method=
_RETURNVERBOSE)
```

```
[Out] 1/60*((-90*B*a^2*b+30*B*b^3-30*C*a^3+90*C*a*b^2)*ln(sec(d*x+c)^2)+(180*B*a^
2*b-60*B*b^3+60*C*a^3-180*C*a*b^2)*ln(tan(d*x+c))-12*B*a^3*cot(d*x+c)^5+(-4
5*B*a^2*b-15*C*a^3)*cot(d*x+c)^4+20*a*cot(d*x+c)^3*(B*a^2-3*B*b^2-3*C*a*b)+
(90*B*a^2*b-30*B*b^3+30*C*a^3-90*C*a*b^2)*cot(d*x+c)^2+(-60*B*a^3+180*B*a*b
^2+180*C*a^2*b-60*C*b^3)*cot(d*x+c)-60*d*x*(B*a^3-3*B*a*b^2-3*C*a^2*b+C*b^3
))/d
```

Fricas [A] (verification not implemented)

none

Time = 0.26 (sec) , antiderivative size = 266, normalized size of antiderivative = 1.14

$$\int \cot^7(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{30\left(Ca^3+3\,Ba^2b-3\,Cab^2-Bb^3\right)\log\left(\frac{\tan(dx+c)^2}{\tan(dx+c)^2+1}\right)\tan\left(dx+c\right)^5+15\left(3\,Ca^3+9\,Ba^2b-6\,Cab^2-2\,Bb^3\right.}{}$$

```
[In] integrate(cot(d*x+c)^7*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="fricas")
```

```
[Out] 1/60*(30*(C*a^3 + 3*B*a^2*b - 3*C*a*b^2 - B*b^3)*log(tan(d*x + c)^2/(tan(d*
x + c)^2 + 1))*tan(d*x + c)^5 + 15*(3*C*a^3 + 9*B*a^2*b - 6*C*a*b^2 - 2*B*b
^3 - 4*(B*a^3 - 3*C*a^2*b - 3*B*a*b^2 + C*b^3)*d*x)*tan(d*x + c)^5 - 60*(B*
a^3 - 3*C*a^2*b - 3*B*a*b^2 + C*b^3)*tan(d*x + c)^4 - 12*B*a^3 + 30*(C*a^3
+ 3*B*a^2*b - 3*C*a*b^2 - B*b^3)*tan(d*x + c)^3 + 20*(B*a^3 - 3*C*a^2*b - 3
```

```
*B*a*b^2)*tan(d*x + c)^2 - 15*(C*a^3 + 3*B*a^2*b)*tan(d*x + c))/(d*tan(d*x
+ c)^5)
```

Sympy [A] (verification not implemented)

Time = 12.36 (sec) , antiderivative size = 462, normalized size of antiderivative = 1.98

$$\int \cot^7(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\begin{cases}\text{NaN}\\ x(a+b\tan(c))^3\left(B\tan(c)+C\tan^2(c)\right)\cot^7(c)\\ \text{NaN}\\ -Ba^3x-\frac{Ba^3}{d\tan(c+dx)}+\frac{Ba^3}{3d\tan^3(c+dx)}-\frac{Ba^3}{5d\tan^5(c+dx)}-\frac{3Ba^2b\log(\tan^2(c+dx)+1)}{2d}+\frac{3Ba^2b\log(\tan(c+dx))}{d}+\frac{3Ba^2b}{2d\tan^2(c+}\end{cases}$$

```
[In] integrate(cot(d*x+c)**7*(a+b*tan(d*x+c))**3*(B*tan(d*x+c)+C*tan(d*x+c)**2),
x)
```

```
[Out] Piecewise((nan, Eq(c, 0) & Eq(d, 0)), (x*(a + b*tan(c))**3*(B*tan(c) + C*ta
n(c)**2)*cot(c)**7, Eq(d, 0)), (nan, Eq(c, -d*x)), (-B*a**3*x - B*a**3/(d*t
an(c + d*x)) + B*a**3/(3*d*tan(c + d*x)**3) - B*a**3/(5*d*tan(c + d*x)**5)
- 3*B*a**2*b*log(tan(c + d*x)**2 + 1)/(2*d) + 3*B*a**2*b*log(tan(c + d*x))/
d + 3*B*a**2*b/(2*d*tan(c + d*x)**2) - 3*B*a**2*b/(4*d*tan(c + d*x)**4) + 3
*B*a*b**2*x + 3*B*a*b**2/(d*tan(c + d*x)) - B*a*b**2/(d*tan(c + d*x)**3) +
B*b**3*log(tan(c + d*x)**2 + 1)/(2*d) - B*b**3*log(tan(c + d*x))/d - B*b**3
/(2*d*tan(c + d*x)**2) - C*a**3*log(tan(c + d*x)**2 + 1)/(2*d) + C*a**3*log
(tan(c + d*x))/d + C*a**3/(2*d*tan(c + d*x)**2) - C*a**3/(4*d*tan(c + d*x)*
*4) + 3*C*a**2*b*x + 3*C*a**2*b/(d*tan(c + d*x)) - C*a**2*b/(d*tan(c + d*x)
**3) + 3*C*a*b**2*log(tan(c + d*x)**2 + 1)/(2*d) - 3*C*a*b**2*log(tan(c + d
*x))/d - 3*C*a*b**2/(2*d*tan(c + d*x)**2) - C*b**3*x - C*b**3/(d*tan(c + d*
x)), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.38 (sec) , antiderivative size = 250, normalized size of antiderivative = 1.07

$$\int \cot^7(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx=$$
$$-\frac{60\,(Ba^3-3\,Ca^2b-3\,Bab^2+Cb^3)(dx+c)+30\,(Ca^3+3\,Ba^2b-3\,Cab^2-Bb^3)\log\left(\tan(dx+c)^2+1\right.}{}$$

```
[In] integrate(cot(d*x+c)^7*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="maxima")
```

```
[Out] -1/60*(60*(B*a^3 - 3*C*a^2*b - 3*B*a*b^2 + C*b^3)*(d*x + c) + 30*(C*a^3 + 3
*B*a^2*b - 3*C*a*b^2 - B*b^3)*log(tan(d*x + c)^2 + 1) - 60*(C*a^3 + 3*B*a^2
*b - 3*C*a*b^2 - B*b^3)*log(tan(d*x + c)) + (60*(B*a^3 - 3*C*a^2*b - 3*B*a*
b^2 + C*b^3)*tan(d*x + c)^4 + 12*B*a^3 - 30*(C*a^3 + 3*B*a^2*b - 3*C*a*b^2
- B*b^3)*tan(d*x + c)^3 - 20*(B*a^3 - 3*C*a^2*b - 3*B*a*b^2)*tan(d*x + c)^2
 + 15*(C*a^3 + 3*B*a^2*b)*tan(d*x + c))/tan(d*x + c)^5)/d
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 670 vs. $2(225) = 450$.

Time = 1.57 (sec) , antiderivative size = 670, normalized size of antiderivative = 2.88

$$\int \cot^7(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$

$$= \frac{6\,Ba^3\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^5 - 15\,Ca^3\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^4 - 45\,Ba^2b\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^4 - 70\,Ba^3\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^3}{}$$

```
[In] integrate(cot(d*x+c)^7*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="giac")
```

```
[Out] 1/960*(6*B*a^3*tan(1/2*d*x + 1/2*c)^5 - 15*C*a^3*tan(1/2*d*x + 1/2*c)^4 - 4
5*B*a^2*b*tan(1/2*d*x + 1/2*c)^4 - 70*B*a^3*tan(1/2*d*x + 1/2*c)^3 + 120*C*
a^2*b*tan(1/2*d*x + 1/2*c)^3 + 120*B*a*b^2*tan(1/2*d*x + 1/2*c)^3 + 180*C*a
^3*tan(1/2*d*x + 1/2*c)^2 + 540*B*a^2*b*tan(1/2*d*x + 1/2*c)^2 - 360*C*a*b^
2*tan(1/2*d*x + 1/2*c)^2 - 120*B*b^3*tan(1/2*d*x + 1/2*c)^2 + 660*B*a^3*tan
(1/2*d*x + 1/2*c) - 1800*C*a^2*b*tan(1/2*d*x + 1/2*c) - 1800*B*a*b^2*tan(1/
2*d*x + 1/2*c) + 480*C*b^3*tan(1/2*d*x + 1/2*c) - 960*(B*a^3 - 3*C*a^2*b -
3*B*a*b^2 + C*b^3)*(d*x + c) - 960*(C*a^3 + 3*B*a^2*b - 3*C*a*b^2 - B*b^3)*
log(tan(1/2*d*x + 1/2*c)^2 + 1) + 960*(C*a^3 + 3*B*a^2*b - 3*C*a*b^2 - B*b^
3)*log(abs(tan(1/2*d*x + 1/2*c))) - (2192*C*a^3*tan(1/2*d*x + 1/2*c)^5 + 65
76*B*a^2*b*tan(1/2*d*x + 1/2*c)^5 - 6576*C*a*b^2*tan(1/2*d*x + 1/2*c)^5 - 2
192*B*b^3*tan(1/2*d*x + 1/2*c)^5 + 660*B*a^3*tan(1/2*d*x + 1/2*c)^4 - 1800*
C*a^2*b*tan(1/2*d*x + 1/2*c)^4 - 1800*B*a*b^2*tan(1/2*d*x + 1/2*c)^4 + 480*
C*b^3*tan(1/2*d*x + 1/2*c)^4 - 180*C*a^3*tan(1/2*d*x + 1/2*c)^3 - 540*B*a^2
*b*tan(1/2*d*x + 1/2*c)^3 + 360*C*a*b^2*tan(1/2*d*x + 1/2*c)^3 + 120*B*b^3*
tan(1/2*d*x + 1/2*c)^3 - 70*B*a^3*tan(1/2*d*x + 1/2*c)^2 + 120*C*a^2*b*tan(
1/2*d*x + 1/2*c)^2 + 120*B*a*b^2*tan(1/2*d*x + 1/2*c)^2 + 15*C*a^3*tan(1/2*
d*x + 1/2*c) + 45*B*a^2*b*tan(1/2*d*x + 1/2*c) + 6*B*a^3)/tan(1/2*d*x + 1/2
*c)^5)/d
```

Mupad [B] (verification not implemented)

Time = 8.43 (sec) , antiderivative size = 238, normalized size of antiderivative = 1.02

$$\int \cot^7(c+dx)(a+b\tan(c+dx))^3 \left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx =$$
$$-\frac{\cot(c+d\,x)^5\left(\tan(c+d\,x)\,\left(\frac{C\,a^3}{4}+\frac{3\,B\,b\,a^2}{4}\right)+\frac{B\,a^3}{5}+\tan(c+d\,x)^2\left(-\frac{B\,a^3}{3}+C\,a^2\,b+B\,a\,b^2\right)+\tan(c+\dots}{d}$$
$$-\frac{\ln\left(\tan(c+d\,x)\right)\,\left(-C\,a^3-3\,B\,a^2\,b+3\,C\,a\,b^2+B\,b^3\right)}{d}$$
$$+\frac{\ln\left(\tan(c+d\,x)-\mathrm{i}\right)\,\left(B+C\,1\mathrm{i}\right)\,\left(a+b\,1\mathrm{i}\right)^3\,1\mathrm{i}}{2\,d}$$
$$-\frac{\ln\left(\tan(c+d\,x)+1\mathrm{i}\right)\,\left(B-C\,1\mathrm{i}\right)\,\left(a-b\,1\mathrm{i}\right)^3\,1\mathrm{i}}{2\,d}$$

```
[In] int(cot(c + d*x)^7*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))
^3,x)
```

```
[Out] (log(tan(c + d*x) - 1i)*(B + C*1i)*(a + b*1i)^3*1i)/(2*d) - (log(tan(c + d*
x))*(B*b^3 - C*a^3 - 3*B*a^2*b + 3*C*a*b^2))/d - (cot(c + d*x)^5*(tan(c + d
*x)*((C*a^3)/4 + (3*B*a^2*b)/4) + (B*a^3)/5 + tan(c + d*x)^2*(B*a*b^2 - (B*
a^3)/3 + C*a^2*b) + tan(c + d*x)^4*(B*a^3 + C*b^3 - 3*B*a*b^2 - 3*C*a^2*b)
+ tan(c + d*x)^3*((B*b^3)/2 - (C*a^3)/2 - (3*B*a^2*b)/2 + (3*C*a*b^2)/2)))/
d - (log(tan(c + d*x) + 1i)*(B - C*1i)*(a - b*1i)^3*1i)/(2*d)
```

3.25 $\int \frac{\tan^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx$

Optimal result 226
Rubi [A] (verified) 226
Mathematica [C] (verified) 229
Maple [A] (verified) 229
Fricas [A] (verification not implemented) 230
Sympy [C] (verification not implemented) 230
Maxima [A] (verification not implemented) 231
Giac [A] (verification not implemented) 232
Mupad [B] (verification not implemented) 232

Optimal result

Integrand size = 40, antiderivative size = 127

$$\begin{aligned}\int \frac{\tan^2(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{a+b\tan(c+dx)}\,dx &= -\frac{(bB-aC)x}{a^2+b^2}+\frac{(aB+bC)\log(\cos(c+dx))}{(a^2+b^2)\,d}\\ &\quad-\frac{a^3(bB-aC)\log(a+b\tan(c+dx))}{b^3\,(a^2+b^2)\,d}+\frac{(bB-aC)\tan(c+dx)}{b^2d}+\frac{C\tan^2(c+dx)}{2bd}\end{aligned}$$

```
[Out] -(B*b-C*a)*x/(a^2+b^2)+(B*a+C*b)*ln(cos(d*x+c))/(a^2+b^2)/d-a^3*(B*b-C*a)*l
n(a+b*tan(d*x+c))/b^3/(a^2+b^2)/d+(B*b-C*a)*tan(d*x+c)/b^2/d+1/2*C*tan(d*x+
c)^2/b/d
```

Rubi [A] (verified)

Time = 0.57 (sec) , antiderivative size = 127, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 7, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.175, Rules used = {3713, 3688, 3728, 3707, 3698, 31, 3556}

$$\begin{aligned}\int \frac{\tan^2(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{a+b\tan(c+dx)}\,dx &= \frac{(aB+bC)\log(\cos(c+dx))}{d\,(a^2+b^2)}-\frac{x(bB-aC)}{a^2+b^2}-\frac{a^3(bB-aC)\log(a+b\tan(c+dx))}{b^3d\,(a^2+b^2)}\\ &\quad+\frac{(bB-aC)\tan(c+dx)}{b^2d}+\frac{C\tan^2(c+dx)}{2bd}\end{aligned}$$

```
[In] Int[(Tan[c + d*x]^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c + d*x
]),x]
```

```
[Out] -(((b*B - a*C)*x)/(a^2 + b^2)) + ((a*B + b*C)*Log[Cos[c + d*x]])/((a^2 + b^
2)*d) - (a^3*(b*B - a*C)*Log[a + b*Tan[c + d*x]])/(b^3*(a^2 + b^2)*d) + ((b
*B - a*C)*Tan[c + d*x])/(b^2*d) + (C*Tan[c + d*x]^2)/(2*b*d)
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3688

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Si
mp[b*B*(a + b*Tan[e + f*x])^(m - 1)*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(m +
 n))), x] + Dist[1/(d*(m + n)), Int[(a + b*Tan[e + f*x])^(m - 2)*(c + d*Tan
[e + f*x])^n*Simp[a^2*A*d*(m + n) - b*B*(b*c*(m - 1) + a*d*(n + 1)) + d*(m
+ n)*(2*a*A*b + B*(a^2 - b^2))*Tan[e + f*x] - (b*B*(b*c - a*d)*(m - 1) - b*
(A*b + a*B)*d*(m + n))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e,
f, A, B, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2,
 0] && GtQ[m, 1] && (IntegerQ[m] || IntegersQ[2*m, 2*n]) &&  !(IGtQ[n, 1] &
& ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3698

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3707

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[(a*A + b*B - a
*C)*(x/(a^2 + b^2)), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
```

```
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{\tan^3(c+dx)(B+C\tan(c+dx))}{a+b\tan(c+dx)}\,dx \\
&= \frac{C\tan^2(c+dx)}{2bd} + \frac{\int \frac{\tan(c+dx)\left(-2aC-2bC\tan(c+dx)+2(bB-aC)\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx}{2b} \\
&= \frac{(bB-aC)\tan(c+dx)}{b^2d} + \frac{C\tan^2(c+dx)}{2bd} \\
&\quad + \frac{\int \frac{-2a(bB-aC)-2b^2B\tan(c+dx)-2\left(abB-a^2C+b^2C\right)\tan^2(c+dx)}{a+b\tan(c+dx)}\,dx}{2b^2} \\
&= -\frac{(bB-aC)x}{a^2+b^2} + \frac{(bB-aC)\tan(c+dx)}{b^2d} + \frac{C\tan^2(c+dx)}{2bd} \\
&\quad - \frac{\left(a^3(bB-aC)\right)\int \frac{1+\tan^2(c+dx)}{a+b\tan(c+dx)}\,dx}{b^2\left(a^2+b^2\right)} - \frac{(aB+bC)\int \tan(c+dx)\,dx}{a^2+b^2} \\
&= -\frac{(bB-aC)x}{a^2+b^2} + \frac{(aB+bC)\log(\cos(c+dx))}{\left(a^2+b^2\right)d} + \frac{(bB-aC)\tan(c+dx)}{b^2d} \\
&\quad + \frac{C\tan^2(c+dx)}{2bd} - \frac{\left(a^3(bB-aC)\right)\operatorname{Subst}\left(\int \frac{1}{a+x}\,dx, x, b\tan(c+dx)\right)}{b^3\left(a^2+b^2\right)d} \\
&= -\frac{(bB-aC)x}{a^2+b^2} + \frac{(aB+bC)\log(\cos(c+dx))}{\left(a^2+b^2\right)d} \\
&\quad - \frac{a^3(bB-aC)\log(a+b\tan(c+dx))}{b^3\left(a^2+b^2\right)d} + \frac{(bB-aC)\tan(c+dx)}{b^2d} + \frac{C\tan^2(c+dx)}{2bd}
\end{aligned}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 1.58 (sec) , antiderivative size = 138, normalized size of antiderivative = 1.09

$$\int \frac{\tan^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx$$
$$= \frac{-\frac{b(B+iC)\log(i-\tan(c+dx))}{a+ib}-\frac{b(B-iC)\log(i+\tan(c+dx))}{a-ib}+\frac{2a^3(-bB+aC)\log(a+b\tan(c+dx))}{b^2(a^2+b^2)}+\frac{2(bB-aC)\tan(c+dx)}{b}+C\tan^2(c}{2bd}$$

```
[In] Integrate[(Tan[c + d*x]^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c
 + d*x]),x]
```

```
[Out] (-((b*(B + I*C)*Log[I - Tan[c + d*x]])/(a + I*b)) - (b*(B - I*C)*Log[I + Ta
n[c + d*x]])/(a - I*b) + (2*a^3*(-(b*B) + a*C)*Log[a + b*Tan[c + d*x]])/(b^
2*(a^2 + b^2)) + (2*(b*B - a*C)*Tan[c + d*x])/b + C*Tan[c + d*x]^2)/(2*b*d)
```

Maple [A] (verified)

Time = 0.12 (sec) , antiderivative size = 127, normalized size of antiderivative = 1.00

method	result
derivativedivides	$\frac{\frac{\frac{C\tan(dx+c)^2 b}{2}+B\tan(dx+c)b-C\tan(dx+c)a}{b^2}+\frac{\frac{(-Ba-Cb)\ln\left(1+\tan(dx+c)^2\right)}{2}+(-Bb+Ca)\arctan(\tan(dx+c))}{a^2+b^2}-\frac{a^3(Bb-Ca)\ln(a+}{b^3\left(a^2+\right.}}{d}$
default	$\frac{\frac{\frac{C\tan(dx+c)^2 b}{2}+B\tan(dx+c)b-C\tan(dx+c)a}{b^2}+\frac{\frac{(-Ba-Cb)\ln\left(1+\tan(dx+c)^2\right)}{2}+(-Bb+Ca)\arctan(\tan(dx+c))}{a^2+b^2}-\frac{a^3(Bb-Ca)\ln(a+}{b^3\left(a^2+\right.}}{d}$
norman	$\frac{(Bb-Ca)\tan(dx+c)}{b^2 d}-\frac{(Bb-Ca)x}{a^2+b^2}+\frac{C\tan(dx+c)^2}{2bd}-\frac{(Ba+Cb)\ln\left(1+\tan(dx+c)^2\right)}{2d(a^2+b^2)}-\frac{a^3(Bb-Ca)\ln(a+b\tan(}{b^3(a^2+b^2)d}$
parallelrisch	$-\frac{2Bx\,b^4d-2Cxa\,b^3d-C\tan(dx+c)^2a^2b^2-C\tan(dx+c)^2b^4+B\ln\left(1+\tan(dx+c)^2\right)a\,b^3+2B\ln(a+b\tan(dx+c))a^3b-2}{2}$
risch	$\frac{2iC\,a^2c}{b^3d}-\frac{xC}{ib-a}-\frac{2iBac}{b^2d}-\frac{2iCc}{bd}-\frac{2iCx}{b}-\frac{2iBax}{b^2}-\frac{ixB}{ib-a}-\frac{2ia^4Cx}{(a^2+b^2)b^3}+\frac{2iC\,a^2x}{b^3}+\frac{2ia^3Bc}{(a^2+b^2)b^2d}-\frac{}{(a}$

```
[In] int(tan(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x,method=_R
ETURNVERBOSE)
```

```
[Out] 1/d*(1/b^2*(1/2*C*tan(d*x+c)^2*b+B*tan(d*x+c)*b-C*tan(d*x+c)*a)+1/(a^2+b^2)
*(1/2*(-B*a-C*b)*ln(1+tan(d*x+c)^2)+(-B*b+C*a)*arctan(tan(d*x+c)))-1/b^3*a^
3*(B*b-C*a)/(a^2+b^2)*ln(a+b*tan(d*x+c)))
```

Fricas [A] (verification not implemented)

none

Time = 0.27 (sec) , antiderivative size = 190, normalized size of antiderivative = 1.50

$$\int \frac{\tan^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx = \frac{2\left(Cab^3-Bb^4\right)dx+\left(Ca^2b^2+Cb^4\right)\tan\left(dx+c\right)^2+\left(Ca^4-Ba^3b\right)\log\left(\frac{b^2\tan(dx+c)^2+2\,ab\tan(dx+c)+a^2}{\tan(dx+c)^2+1}\right)-(Ca^4}{2\left(a^2b^3+b^5\right)d}$$

```
[In] integrate(tan(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, al
gorithm="fricas")
```

```
[Out] 1/2*(2*(C*a*b^3 - B*b^4)*d*x + (C*a^2*b^2 + C*b^4)*tan(d*x + c)^2 + (C*a^4
- B*a^3*b)*log((b^2*tan(d*x + c)^2 + 2*a*b*tan(d*x + c) + a^2)/(tan(d*x + c
)^2 + 1)) - (C*a^4 - B*a^3*b - B*a*b^3 - C*b^4)*log(1/(tan(d*x + c)^2 + 1))
 - 2*(C*a^3*b - B*a^2*b^2 + C*a*b^3 - B*b^4)*tan(d*x + c))/((a^2*b^3 + b^5)
*d)
```

Sympy [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 0.93 (sec) , antiderivative size = 1306, normalized size of antiderivative = 10.28

$$\int \frac{\tan^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx = \text{Too large to display}$$

```
[In] integrate(tan(d*x+c)**2*(B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c)),x)
```

```
[Out] Piecewise((zoo*x*(B*tan(c) + C*tan(c)**2)*tan(c), Eq(a, 0) & Eq(b, 0) & Eq(
d, 0)), ((-B*log(tan(c + d*x)**2 + 1)/(2*d) + B*tan(c + d*x)**2/(2*d) + C*x
 + C*tan(c + d*x)**3/(3*d) - C*tan(c + d*x)/d)/a, Eq(b, 0)), (-3*B*d*x*tan(
c + d*x)/(2*b*d*tan(c + d*x) - 2*I*b*d) + 3*I*B*d*x/(2*b*d*tan(c + d*x) - 2
*I*b*d) + I*B*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(2*b*d*tan(c + d*x) - 2
*I*b*d) + B*log(tan(c + d*x)**2 + 1)/(2*b*d*tan(c + d*x) - 2*I*b*d) + 2*B*t
an(c + d*x)**2/(2*b*d*tan(c + d*x) - 2*I*b*d) + 3*B/(2*b*d*tan(c + d*x) - 2
*I*b*d) - 3*I*C*d*x*tan(c + d*x)/(2*b*d*tan(c + d*x) - 2*I*b*d) - 3*C*d*x/(
2*b*d*tan(c + d*x) - 2*I*b*d) - 2*C*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(
2*b*d*tan(c + d*x) - 2*I*b*d) + 2*I*C*log(tan(c + d*x)**2 + 1)/(2*b*d*tan(c
 + d*x) - 2*I*b*d) + C*tan(c + d*x)**3/(2*b*d*tan(c + d*x) - 2*I*b*d) + I*C
*tan(c + d*x)**2/(2*b*d*tan(c + d*x) - 2*I*b*d) + 3*I*C/(2*b*d*tan(c + d*x)
 - 2*I*b*d), Eq(a, -I*b)), (-3*B*d*x*tan(c + d*x)/(2*b*d*tan(c + d*x) + 2*I
*b*d) - 3*I*B*d*x/(2*b*d*tan(c + d*x) + 2*I*b*d) - I*B*log(tan(c + d*x)**2
+ 1)*tan(c + d*x)/(2*b*d*tan(c + d*x) + 2*I*b*d) + B*log(tan(c + d*x)**2 +
```

```
1)/(2*b*d*tan(c + d*x) + 2*I*b*d) + 2*B*tan(c + d*x)**2/(2*b*d*tan(c + d*x)
 + 2*I*b*d) + 3*B/(2*b*d*tan(c + d*x) + 2*I*b*d) + 3*I*C*d*x*tan(c + d*x)/(
2*b*d*tan(c + d*x) + 2*I*b*d) - 3*C*d*x/(2*b*d*tan(c + d*x) + 2*I*b*d) - 2*
C*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(2*b*d*tan(c + d*x) + 2*I*b*d) - 2*
I*C*log(tan(c + d*x)**2 + 1)/(2*b*d*tan(c + d*x) + 2*I*b*d) + C*tan(c + d*x
)**3/(2*b*d*tan(c + d*x) + 2*I*b*d) - I*C*tan(c + d*x)**2/(2*b*d*tan(c + d*
x) + 2*I*b*d) - 3*I*C/(2*b*d*tan(c + d*x) + 2*I*b*d), Eq(a, I*b)), (x*(B*ta
n(c) + C*tan(c)**2)*tan(c)**2/(a + b*tan(c)), Eq(d, 0)), (-2*B*a**3*b*log(a
/b + tan(c + d*x))/(2*a**2*b**3*d + 2*b**5*d) + 2*B*a**2*b**2*tan(c + d*x)/
(2*a**2*b**3*d + 2*b**5*d) - B*a*b**3*log(tan(c + d*x)**2 + 1)/(2*a**2*b**3
*d + 2*b**5*d) - 2*B*b**4*d*x/(2*a**2*b**3*d + 2*b**5*d) + 2*B*b**4*tan(c +
 d*x)/(2*a**2*b**3*d + 2*b**5*d) + 2*C*a**4*log(a/b + tan(c + d*x))/(2*a**2
*b**3*d + 2*b**5*d) - 2*C*a**3*b*tan(c + d*x)/(2*a**2*b**3*d + 2*b**5*d) +
C*a**2*b**2*tan(c + d*x)**2/(2*a**2*b**3*d + 2*b**5*d) + 2*C*a*b**3*d*x/(2*
a**2*b**3*d + 2*b**5*d) - 2*C*a*b**3*tan(c + d*x)/(2*a**2*b**3*d + 2*b**5*d
) - C*b**4*log(tan(c + d*x)**2 + 1)/(2*a**2*b**3*d + 2*b**5*d) + C*b**4*tan
(c + d*x)**2/(2*a**2*b**3*d + 2*b**5*d), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.30 (sec) , antiderivative size = 130, normalized size of antiderivative = 1.02

$$\int \frac{\tan^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx = \frac{\frac{2\,(Ca-Bb)(dx+c)}{a^2+b^2}+\frac{2\,(Ca^4-Ba^3b)\log(b\tan(dx+c)+a)}{a^2b^3+b^5}-\frac{(Ba+Cb)\log\left(\tan(dx+c)^2+1\right)}{a^2+b^2}+\frac{Cb\tan(dx+c)^2-2\,(Ca-Bb)\tan(dx+c)}{b^2}}{2\,d}$$

```
[In] integrate(tan(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, al
gorithm="maxima")
```

```
[Out] 1/2*(2*(C*a - B*b)*(d*x + c)/(a^2 + b^2) + 2*(C*a^4 - B*a^3*b)*log(b*tan(d*
x + c) + a)/(a^2*b^3 + b^5) - (B*a + C*b)*log(tan(d*x + c)^2 + 1)/(a^2 + b^
2) + (C*b*tan(d*x + c)^2 - 2*(C*a - B*b)*tan(d*x + c))/b^2)/d
```

Giac [A] (verification not implemented)

none

Time = 0.65 (sec) , antiderivative size = 135, normalized size of antiderivative = 1.06

$$\int \frac{\tan^2(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{a+b\tan(c+dx)}\,dx$$
$$= \frac{\frac{2\,(Ca-Bb)(dx+c)}{a^2+b^2} - \frac{(Ba+Cb)\log\left(\tan(dx+c)^2+1\right)}{a^2+b^2} + \frac{2\,(Ca^4-Ba^3b)\log(|b\tan(dx+c)+a|)}{a^2b^3+b^5} + \frac{Cb\tan(dx+c)^2-2\,Ca\tan(dx+c)+2\,Bb\tan(dx\cdots}{b^2}}{2\,d}$$

```
[In] integrate(tan(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, al
gorithm="giac")
```

```
[Out] 1/2*(2*(C*a - B*b)*(d*x + c)/(a^2 + b^2) - (B*a + C*b)*log(tan(d*x + c)^2 +
 1)/(a^2 + b^2) + 2*(C*a^4 - B*a^3*b)*log(abs(b*tan(d*x + c) + a))/(a^2*b^3
 + b^5) + (C*b*tan(d*x + c)^2 - 2*C*a*tan(d*x + c) + 2*B*b*tan(d*x + c))/b^
2)/d
```

Mupad [B] (verification not implemented)

Time = 8.20 (sec) , antiderivative size = 144, normalized size of antiderivative = 1.13

$$\int \frac{\tan^2(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{a+b\tan(c+dx)}\,dx$$
$$= \frac{\tan(c+d\,x)\,\left(\frac{B}{b}-\frac{C\,a}{b^2}\right)}{d} - \frac{\ln\left(\tan(c+d\,x)-\mathrm{i}\right)\,\left(-C+B\,1\mathrm{i}\right)}{2\,d\,\left(-b+a\,1\mathrm{i}\right)}$$
$$+ \frac{\ln\left(a+b\tan(c+d\,x)\right)\,\left(C\,a^4-B\,a^3\,b\right)}{d\,\left(a^2\,b^3+b^5\right)}$$
$$- \frac{\ln\left(\tan(c+d\,x)+1\mathrm{i}\right)\,\left(B-C\,1\mathrm{i}\right)}{2\,d\,\left(a-b\,1\mathrm{i}\right)} + \frac{C\tan(c+d\,x)^2}{2\,b\,d}$$

```
[In] int((tan(c + d*x)^2*(B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c + d*x
)),x)
```

```
[Out] (tan(c + d*x)*(B/b - (C*a)/b^2))/d - (log(tan(c + d*x) - 1i)*(B*1i - C))/(2
*d*(a*1i - b)) + (log(a + b*tan(c + d*x))*(C*a^4 - B*a^3*b))/(d*(b^5 + a^2*
b^3)) - (log(tan(c + d*x) + 1i)*(B - C*1i))/(2*d*(a - b*1i)) + (C*tan(c + d
*x)^2)/(2*b*d)
```

3.26 $\int \frac{\tan(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx$

Optimal result 233
Rubi [A] (verified) 233
Mathematica [C] (verified) 235
Maple [A] (verified) 236
Fricas [A] (verification not implemented) 236
Sympy [C] (verification not implemented) 237
Maxima [A] (verification not implemented) 238
Giac [A] (verification not implemented) 238
Mupad [B] (verification not implemented) 238

Optimal result

Integrand size = 38, antiderivative size = 101

$$\int \frac{\tan(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx = -\frac{(aB+bC)x}{a^2+b^2}-\frac{(bB-aC)\log(\cos(c+dx))}{(a^2+b^2)\,d}+\frac{a^2(bB-aC)\log(a+b\tan(c+dx))}{b^2\,(a^2+b^2)\,d}+\frac{C\tan(c+dx)}{bd}$$

```
[Out] -(B*a+C*b)*x/(a^2+b^2)-(B*b-C*a)*ln(cos(d*x+c))/(a^2+b^2)/d+a^2*(B*b-C*a)*l
n(a+b*tan(d*x+c))/b^2/(a^2+b^2)/d+C*tan(d*x+c)/b/d
```

Rubi [A] (verified)

Time = 0.31 (sec) , antiderivative size = 101, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.158, Rules used = {3713, 3687, 3707, 3698, 31, 3556}

$$\int \frac{\tan(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx = \frac{a^2(bB-aC)\log(a+b\tan(c+dx))}{b^2d\,(a^2+b^2)}-\frac{(bB-aC)\log(\cos(c+dx))}{d\,(a^2+b^2)}-\frac{x(aB+bC)}{a^2+b^2}+\frac{C\tan(c+dx)}{bd}$$

```
[In] Int[(Tan[c + d*x]*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c + d*x])
,x]
```

[Out] -(((a*B + b*C)*x)/(a^2 + b^2)) - ((b*B - a*C)*Log[Cos[c + d*x]])/((a^2 + b^2)*d) + (a^2*(b*B - a*C)*Log[a + b*Tan[c + d*x]])/(b^2*(a^2 + b^2)*d) + (C*Tan[c + d*x])/(b*d)

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3687

```
Int[(((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^2*((A_.) + (B_.)*tan[(e_.) + (
f_.)*(x_)]))/((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[b^2
*B*(Tan[e + f*x]/(d*f)), x] + Dist[1/d, Int[(a^2*A*d - b^2*B*c + (2*a*A*b +
 B*(a^2 - b^2))*d*Tan[e + f*x] + (A*b^2*d - b*B*(b*c - 2*a*d))*Tan[e + f*x]
^2)/(c + d*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] && N
eQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0]
```

Rule 3698

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3707

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[(a*A + b*B - a
*C)*(x/(a^2 + b^2)), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\text{integral} = \int \frac{\tan^2(c+dx)(B+C\tan(c+dx))}{a+b\tan(c+dx)}\,dx$$

$$= \frac{C\tan(c+dx)}{bd} + \frac{\int \frac{-aC-bC\tan(c+dx)+(bB-aC)\tan^2(c+dx)}{a+b\tan(c+dx)}\,dx}{b}$$

$$= -\frac{(aB+bC)x}{a^2+b^2} + \frac{C\tan(c+dx)}{bd} + \frac{(bB-aC)\int \tan(c+dx)\,dx}{a^2+b^2} + \frac{(a^2(bB-aC))\int \frac{1+\tan^2(c+dx)}{a+b\tan(c+dx)}\,dx}{b\,(a^2+b^2)}$$

$$= -\frac{(aB+bC)x}{a^2+b^2} - \frac{(bB-aC)\log(\cos(c+dx))}{(a^2+b^2)\,d} + \frac{C\tan(c+dx)}{bd} + \frac{(a^2(bB-aC))\,\text{Subst}\left(\int \frac{1}{a+x}\,dx, x, b\tan(c+dx)\right)}{b^2\,(a^2+b^2)\,d}$$

$$= -\frac{(aB+bC)x}{a^2+b^2} - \frac{(bB-aC)\log(\cos(c+dx))}{(a^2+b^2)\,d} + \frac{a^2(bB-aC)\log(a+b\tan(c+dx))}{b^2\,(a^2+b^2)\,d} + \frac{C\tan(c+dx)}{bd}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 0.66 (sec) , antiderivative size = 118, normalized size of antiderivative = 1.17

$$\int \frac{\tan(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{a+b\tan(c+dx)}\,dx = \frac{\frac{i(B+iC)\log(i-\tan(c+dx))}{a+ib} - \frac{(iB+C)\log(i+\tan(c+dx))}{a-ib} + \frac{2a^2(bB-aC)\log(a+b\tan(c+dx))}{b^2(a^2+b^2)} + \frac{2C\tan(c+dx)}{b}}{2d}$$

```
[In] Integrate[(Tan[c + d*x]*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c +
 d*x]),x]

[Out] ((I*(B + I*C)*Log[I - Tan[c + d*x]])/(a + I*b) - ((I*B + C)*Log[I + Tan[c +
 d*x]])/(a - I*b) + (2*a^2*(b*B - a*C)*Log[a + b*Tan[c + d*x]])/(b^2*(a^2 +
 b^2)) + (2*C*Tan[c + d*x])/b)/(2*d)
```

Maple [A] (verified)

Time = 0.13 (sec) , antiderivative size = 101, normalized size of antiderivative = 1.00

method	result
derivativedivides	$\frac{\frac{\tan(dx+c)C}{b}+\frac{\frac{(Bb-Ca)\ln\left(1+\tan(dx+c)^2\right)}{2}+(-Ba-Cb)\arctan(\tan(dx+c))}{a^2+b^2}+\frac{a^2(Bb-Ca)\ln(a+b\tan(dx+c))}{b^2\left(a^2+b^2\right)}}{d}$
default	$\frac{\frac{\tan(dx+c)C}{b}+\frac{\frac{(Bb-Ca)\ln\left(1+\tan(dx+c)^2\right)}{2}+(-Ba-Cb)\arctan(\tan(dx+c))}{a^2+b^2}+\frac{a^2(Bb-Ca)\ln(a+b\tan(dx+c))}{b^2\left(a^2+b^2\right)}}{d}$
norman	$\frac{C\tan(dx+c)}{bd}-\frac{(Ba+Cb)x}{a^2+b^2}+\frac{a^2(Bb-Ca)\ln(a+b\tan(dx+c))}{b^2(a^2+b^2)d}+\frac{(Bb-Ca)\ln\left(1+\tan(dx+c)^2\right)}{2d(a^2+b^2)}$
parallelrisch	$\frac{-2Ba\,b^2dx-2C\,b^3dx+B\ln\left(1+\tan(dx+c)^2\right)b^3+2B\ln(a+b\tan(dx+c))a^2b-C\ln\left(1+\tan(dx+c)^2\right)a\,b^2-2C\ln(a+b\tan(dx}{2d(a^2+b^2)b^2}$
risch	$\frac{xB}{ib-a}-\frac{ixC}{ib-a}-\frac{2ia^2Bx}{b(a^2+b^2)}-\frac{2ia^2Bc}{bd(a^2+b^2)}+\frac{2ia^3Cx}{b^2(a^2+b^2)}+\frac{2ia^3Cc}{b^2d(a^2+b^2)}+\frac{2iBx}{b}+\frac{2iBc}{bd}-\frac{2iCax}{b^2}-\frac{2iCac}{b^2d}+$

```
[In] int(tan(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x,method=_RET
URNVERBOSE)
```

```
[Out] 1/d*(tan(d*x+c)*C/b+1/(a^2+b^2)*(1/2*(B*b-C*a)*ln(1+tan(d*x+c)^2)+(-B*a-C*b
)*arctan(tan(d*x+c)))+1/b^2*a^2*(B*b-C*a)/(a^2+b^2)*ln(a+b*tan(d*x+c)))
```

Fricas [A] (verification not implemented)

none

Time = 0.28 (sec) , antiderivative size = 149, normalized size of antiderivative = 1.48

$$\int \frac{\tan(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx = -\frac{2\left(Bab^2+Cb^3\right)dx+\left(Ca^3-Ba^2b\right)\log\left(\frac{b^2\tan(dx+c)^2+2\,ab\tan(dx+c)+a^2}{\tan(dx+c)^2+1}\right)-\left(Ca^3-Ba^2b+Cab^2-Bb^3\right)\log\left(\right.}{2\left(a^2b^2+b^4\right)d}$$

```
[In] integrate(tan(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, algo
rithm="fricas")
```

```
[Out] -1/2*(2*(B*a*b^2 + C*b^3)*d*x + (C*a^3 - B*a^2*b)*log((b^2*tan(d*x + c)^2 +
 2*a*b*tan(d*x + c) + a^2)/(tan(d*x + c)^2 + 1)) - (C*a^3 - B*a^2*b + C*a*b
^2 - B*b^3)*log(1/(tan(d*x + c)^2 + 1)) - 2*(C*a^2*b + C*b^3)*tan(d*x + c))
/((a^2*b^2 + b^4)*d)
```

Sympy [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 0.67 (sec) , antiderivative size = 1020, normalized size of antiderivative = 10.10

$$\int \frac{\tan(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx = \text{Too large to display}$$

```
[In] integrate(tan(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c)),x)
```

```
[Out] Piecewise((zoo*x*(B*tan(c) + C*tan(c)**2), Eq(a, 0) & Eq(b, 0) & Eq(d, 0)),
 ((-B*x + B*tan(c + d*x)/d - C*log(tan(c + d*x)**2 + 1)/(2*d) + C*tan(c + d
*x)**2/(2*d))/a, Eq(b, 0)), (I*B*d*x*tan(c + d*x)/(2*b*d*tan(c + d*x) - 2*I
*b*d) + B*d*x/(2*b*d*tan(c + d*x) - 2*I*b*d) + B*log(tan(c + d*x)**2 + 1)*t
an(c + d*x)/(2*b*d*tan(c + d*x) - 2*I*b*d) - I*B*log(tan(c + d*x)**2 + 1)/(
2*b*d*tan(c + d*x) - 2*I*b*d) - I*B/(2*b*d*tan(c + d*x) - 2*I*b*d) - 3*C*d*
x*tan(c + d*x)/(2*b*d*tan(c + d*x) - 2*I*b*d) + 3*I*C*d*x/(2*b*d*tan(c + d*
x) - 2*I*b*d) + I*C*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(2*b*d*tan(c + d*
x) - 2*I*b*d) + C*log(tan(c + d*x)**2 + 1)/(2*b*d*tan(c + d*x) - 2*I*b*d) +
 2*C*tan(c + d*x)**2/(2*b*d*tan(c + d*x) - 2*I*b*d) + 3*C/(2*b*d*tan(c + d*
x) - 2*I*b*d), Eq(a, -I*b)), (-I*B*d*x*tan(c + d*x)/(2*b*d*tan(c + d*x) + 2
*I*b*d) + B*d*x/(2*b*d*tan(c + d*x) + 2*I*b*d) + B*log(tan(c + d*x)**2 + 1)
*tan(c + d*x)/(2*b*d*tan(c + d*x) + 2*I*b*d) + I*B*log(tan(c + d*x)**2 + 1)
/(2*b*d*tan(c + d*x) + 2*I*b*d) + I*B/(2*b*d*tan(c + d*x) + 2*I*b*d) - 3*C*
d*x*tan(c + d*x)/(2*b*d*tan(c + d*x) + 2*I*b*d) - 3*I*C*d*x/(2*b*d*tan(c +
d*x) + 2*I*b*d) - I*C*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(2*b*d*tan(c +
d*x) + 2*I*b*d) + C*log(tan(c + d*x)**2 + 1)/(2*b*d*tan(c + d*x) + 2*I*b*d)
 + 2*C*tan(c + d*x)**2/(2*b*d*tan(c + d*x) + 2*I*b*d) + 3*C/(2*b*d*tan(c +
d*x) + 2*I*b*d), Eq(a, I*b)), (x*(B*tan(c) + C*tan(c)**2)*tan(c)/(a + b*tan
(c)), Eq(d, 0)), (2*B*a**2*b*log(a/b + tan(c + d*x))/(2*a**2*b**2*d + 2*b**
4*d) - 2*B*a*b**2*d*x/(2*a**2*b**2*d + 2*b**4*d) + B*b**3*log(tan(c + d*x)*
*2 + 1)/(2*a**2*b**2*d + 2*b**4*d) - 2*C*a**3*log(a/b + tan(c + d*x))/(2*a*
*2*b**2*d + 2*b**4*d) + 2*C*a**2*b*tan(c + d*x)/(2*a**2*b**2*d + 2*b**4*d)
- C*a*b**2*log(tan(c + d*x)**2 + 1)/(2*a**2*b**2*d + 2*b**4*d) - 2*C*b**3*d
*x/(2*a**2*b**2*d + 2*b**4*d) + 2*C*b**3*tan(c + d*x)/(2*a**2*b**2*d + 2*b*
*4*d), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.29 (sec) , antiderivative size = 109, normalized size of antiderivative = 1.08

$$\int \frac{\tan(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{a+b\tan(c+dx)}\,dx = -\frac{\frac{2\,(Ba+Cb)(dx+c)}{a^2+b^2}+\frac{2\,(Ca^3-Ba^2b)\log(b\tan(dx+c)+a)}{a^2b^2+b^4}+\frac{(Ca-Bb)\log\left(\tan(dx+c)^2+1\right)}{a^2+b^2}-\frac{2\,C\tan(dx+c)}{b}}{2\,d}$$

```
[In] integrate(tan(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, algo
rithm="maxima")
```

```
[Out] -1/2*(2*(B*a + C*b)*(d*x + c)/(a^2 + b^2) + 2*(C*a^3 - B*a^2*b)*log(b*tan(d
*x + c) + a)/(a^2*b^2 + b^4) + (C*a - B*b)*log(tan(d*x + c)^2 + 1)/(a^2 + b
^2) - 2*C*tan(d*x + c)/b)/d
```

Giac [A] (verification not implemented)

none

Time = 0.49 (sec) , antiderivative size = 110, normalized size of antiderivative = 1.09

$$\int \frac{\tan(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{a+b\tan(c+dx)}\,dx = -\frac{\frac{2\,(Ba+Cb)(dx+c)}{a^2+b^2}+\frac{(Ca-Bb)\log\left(\tan(dx+c)^2+1\right)}{a^2+b^2}+\frac{2\,(Ca^3-Ba^2b)\log(|b\tan(dx+c)+a|)}{a^2b^2+b^4}-\frac{2\,C\tan(dx+c)}{b}}{2\,d}$$

```
[In] integrate(tan(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, algo
rithm="giac")
```

```
[Out] -1/2*(2*(B*a + C*b)*(d*x + c)/(a^2 + b^2) + (C*a - B*b)*log(tan(d*x + c)^2
+ 1)/(a^2 + b^2) + 2*(C*a^3 - B*a^2*b)*log(abs(b*tan(d*x + c) + a))/(a^2*b^
2 + b^4) - 2*C*tan(d*x + c)/b)/d
```

Mupad [B] (verification not implemented)

Time = 8.25 (sec) , antiderivative size = 117, normalized size of antiderivative = 1.16

$$\int \frac{\tan(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{a+b\tan(c+dx)}\,dx = \frac{C\tan(c+d\,x)}{b\,d}+\frac{\ln\left(\tan(c+d\,x)+1\mathrm{i}\right)\,\left(B-C\,1\mathrm{i}\right)}{2\,d\,\left(b+a\,1\mathrm{i}\right)} - \frac{\ln\left(a+b\tan(c+d\,x)\right)\,\left(C\,a^3-B\,a^2\,b\right)}{d\,\left(a^2\,b^2+b^4\right)}+\frac{\ln\left(\tan(c+d\,x)-\mathrm{i}\right)\,\left(-C+B\,1\mathrm{i}\right)}{2\,d\,\left(a+b\,1\mathrm{i}\right)}$$

[In] int((tan(c + d*x)*(B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c + d*x)),x)

[Out] (log(tan(c + d*x) + 1i)*(B - C*1i))/(2*d*(a*1i + b)) - (log(a + b*tan(c + d*x))*(C*a^3 - B*a^2*b))/(d*(b^4 + a^2*b^2)) + (C*tan(c + d*x))/(b*d) + (log(tan(c + d*x) - 1i)*(B*1i - C))/(2*d*(a + b*1i))

3.27 $\int \frac{B\tan(c+dx)+C\tan^2(c+dx)}{a+b\tan(c+dx)}\,dx$

Optimal result 240
Rubi [A] (verified) 240
Mathematica [C] (verified) 242
Maple [A] (verified) 242
Fricas [A] (verification not implemented) 243
Sympy [C] (verification not implemented) 243
Maxima [A] (verification not implemented) 244
Giac [A] (verification not implemented) 244
Mupad [B] (verification not implemented) 245

Optimal result

Integrand size = 32, antiderivative size = 85

$$\int \frac{B\tan(c+dx)+C\tan^2(c+dx)}{a+b\tan(c+dx)}\,dx = \frac{(bB-aC)x}{a^2+b^2} - \frac{(aB+bC)\log(\cos(c+dx))}{(a^2+b^2)\,d} - \frac{a(bB-aC)\log(a+b\tan(c+dx))}{b\,(a^2+b^2)\,d}$$

```
[Out] (B*b-C*a)*x/(a^2+b^2)-(B*a+C*b)*ln(cos(d*x+c))/(a^2+b^2)/d-a*(B*b-C*a)*ln(a
+b*tan(d*x+c))/b/(a^2+b^2)/d
```

Rubi [A] (verified)

Time = 0.21 (sec) , antiderivative size = 85, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 4, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.125, Rules used = {1643, 649, 209, 266}

$$\int \frac{B\tan(c+dx)+C\tan^2(c+dx)}{a+b\tan(c+dx)}\,dx = -\frac{a(bB-aC)\log(a+b\tan(c+dx))}{bd\,(a^2+b^2)} - \frac{(aB+bC)\log(\cos(c+dx))}{d\,(a^2+b^2)} + \frac{x(bB-aC)}{a^2+b^2}$$

```
[In] Int[(B*Tan[c + d*x] + C*Tan[c + d*x]^2)/(a + b*Tan[c + d*x]),x]
```

```
[Out] ((b*B - a*C)*x)/(a^2 + b^2) - ((a*B + b*C)*Log[Cos[c + d*x]])/((a^2 + b^2)*
d) - (a*(b*B - a*C)*Log[a + b*Tan[c + d*x]])/(b*(a^2 + b^2)*d)
```

Rule 209

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[b, 2]))*A
rcTan[Rt[b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && PosQ[a/b] && (GtQ[a
```

```
, 0] || GtQ[b, 0])
```

Rule 266

```
Int[(x_)^(m_.)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> Simp[Log[RemoveConten
t[a + b*x^n, x]]/(b*n), x] /; FreeQ[{a, b, m, n}, x] && EqQ[m, n - 1]
```

Rule 649

```
Int[((d_) + (e_.)*(x_))/((a_) + (c_.)*(x_)^2), x_Symbol] :> Dist[d, Int[1/(
a + c*x^2), x], x] + Dist[e, Int[x/(a + c*x^2), x], x] /; FreeQ[{a, c, d, e
}, x] &&  !NiceSqrtQ[(-a)*c]
```

Rule 1643

```
Int[(Pq_)*((d_) + (e_.)*(x_))^(m_.)*((a_) + (c_.)*(x_)^2)^(p_.), x_Symbol]
:> Int[ExpandIntegrand[(d + e*x)^m*Pq*(a + c*x^2)^p, x], x] /; FreeQ[{a, c,
 d, e, m}, x] && PolyQ[Pq, x] && IGtQ[p, -2]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \frac{\text{Subst}\left(\int \frac{x(B+Cx)}{(a+bx)(1+x^2)}\,dx, x, \tan(c+dx)\right)}{d}\\
&= \frac{\text{Subst}\left(\int \left(\frac{a(-bB+aC)}{(a^2+b^2)(a+bx)}+\frac{bB-aC+(aB+bC)x}{(a^2+b^2)(1+x^2)}\right)dx, x, \tan(c+dx)\right)}{d}\\
&= -\frac{a(bB-aC)\log(a+b\tan(c+dx))}{b(a^2+b^2)d}+\frac{\text{Subst}\left(\int \frac{bB-aC+(aB+bC)x}{1+x^2}\,dx, x, \tan(c+dx)\right)}{(a^2+b^2)d}\\
&= -\frac{a(bB-aC)\log(a+b\tan(c+dx))}{b(a^2+b^2)d}+\frac{(bB-aC)\text{Subst}\left(\int \frac{1}{1+x^2}\,dx, x, \tan(c+dx)\right)}{(a^2+b^2)d}\\
&\quad+\frac{(aB+bC)\text{Subst}\left(\int \frac{x}{1+x^2}\,dx, x, \tan(c+dx)\right)}{(a^2+b^2)d}\\
&= \frac{(bB-aC)x}{a^2+b^2}-\frac{(aB+bC)\log(\cos(c+dx))}{(a^2+b^2)d}-\frac{a(bB-aC)\log(a+b\tan(c+dx))}{b(a^2+b^2)d}
\end{aligned}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 0.20 (sec) , antiderivative size = 98, normalized size of antiderivative = 1.15

$$\int \frac{B\tan(c+dx)+C\tan^2(c+dx)}{a+b\tan(c+dx)}\,dx$$
$$= \frac{(a-ib)b(B+iC)\log(i-\tan(c+dx))+(a+ib)b(B-iC)\log(i+\tan(c+dx))+2a(-bB+aC)\log(a+}{2b\left(a^2+b^2\right)d}$$

```
[In] Integrate[(B*Tan[c + d*x] + C*Tan[c + d*x]^2)/(a + b*Tan[c + d*x]),x]
```

```
[Out] ((a - I*b)*b*(B + I*C)*Log[I - Tan[c + d*x]] + (a + I*b)*b*(B - I*C)*Log[I
+ Tan[c + d*x]] + 2*a*(-(b*B) + a*C)*Log[a + b*Tan[c + d*x]])/(2*b*(a^2 + b
^2)*d)
```

Maple [A] (verified)

Time = 0.08 (sec) , antiderivative size = 87, normalized size of antiderivative = 1.02

method	result
derivativedivides	$\frac{\frac{\frac{(Ba+Cb)\ln\left(1+\tan(dx+c)^2\right)}{2}+(Bb-Ca)\arctan(\tan(dx+c))}{a^2+b^2}-\frac{a(Bb-Ca)\ln(a+b\tan(dx+c))}{\left(a^2+b^2\right)b}}{d}$
default	$\frac{\frac{\frac{(Ba+Cb)\ln\left(1+\tan(dx+c)^2\right)}{2}+(Bb-Ca)\arctan(\tan(dx+c))}{a^2+b^2}-\frac{a(Bb-Ca)\ln(a+b\tan(dx+c))}{\left(a^2+b^2\right)b}}{d}$
norman	$\frac{(Bb-Ca)x}{a^2+b^2}+\frac{(Ba+Cb)\ln\left(1+\tan(dx+c)^2\right)}{2d(a^2+b^2)}-\frac{a(Bb-Ca)\ln(a+b\tan(dx+c))}{b(a^2+b^2)d}$
parallelrisch	$\frac{2B\,b^2dx-2Cabdx+B\ln\left(1+\tan(dx+c)^2\right)ab-2B\ln(a+b\tan(dx+c))ab+C\ln\left(1+\tan(dx+c)^2\right)b^2+2C\ln(a+b\tan(dx+c))a}{2(a^2+b^2)bd}$
risch	$\frac{ixB}{ib-a}+\frac{xC}{ib-a}+\frac{2iCx}{b}+\frac{2iCc}{bd}+\frac{2iaBx}{a^2+b^2}+\frac{2iaBc}{(a^2+b^2)d}-\frac{2ia^2Cx}{(a^2+b^2)b}-\frac{2ia^2Cc}{(a^2+b^2)bd}-\frac{\ln\left(e^{2i(dx+c)}+1\right)C}{bd}-\frac{a\ln(}{}$

```
[In] int((B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x,method=_RETURNVERBOSE)
```

```
[Out] 1/d*(1/(a^2+b^2)*(1/2*(B*a+C*b)*ln(1+tan(d*x+c)^2)+(B*b-C*a)*arctan(tan(d*x
+c)))-a*(B*b-C*a)/(a^2+b^2)/b*ln(a+b*tan(d*x+c)))
```

Fricas [A] (verification not implemented)

none

Time = 0.26 (sec) , antiderivative size = 110, normalized size of antiderivative = 1.29

$$\int \frac{B\tan(c+dx)+C\tan^2(c+dx)}{a+b\tan(c+dx)}\,dx = -\frac{2\,(Cab-Bb^2)dx-(Ca^2-Bab)\log\left(\frac{b^2\tan(dx+c)^2+2\,ab\tan(dx+c)+a^2}{\tan(dx+c)^2+1}\right)+(Ca^2+Cb^2)\log\left(\frac{1}{\tan(dx+c)^2+1}\right)}{2\,(a^2b+b^3)d}$$

```
[In] integrate((B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, algorithm="fric
as")
```

```
[Out] -1/2*(2*(C*a*b - B*b^2)*d*x - (C*a^2 - B*a*b)*log((b^2*tan(d*x + c)^2 + 2*a
*b*tan(d*x + c) + a^2)/(tan(d*x + c)^2 + 1)) + (C*a^2 + C*b^2)*log(1/(tan(d
*x + c)^2 + 1)))/((a^2*b + b^3)*d)
```

Sympy [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 0.56 (sec) , antiderivative size = 711, normalized size of antiderivative = 8.36

$$\int \frac{B\tan(c+dx)+C\tan^2(c+dx)}{a+b\tan(c+dx)}\,dx = \begin{cases} \frac{\tilde{\infty}x(B\tan(c)+C\tan^2(c))}{\tan(c)} \\ \frac{\frac{B\log(\tan^2(c+dx)+1)}{2d}-Cx+\frac{C\tan(c+dx)}{d}}{a} \\ \frac{Bdx\tan(c+dx)}{2bd\tan(c+dx)-2ibd}-\frac{iBdx}{2bd\tan(c+dx)-2ibd}-\frac{B}{2bd\tan(c+dx)-2ibd}+\frac{iCdx\tan(c+dx)}{2bd\tan(c+dx)-2ibd}+\frac{Cdx}{2bd\tan(c+dx)-2ibd}+\frac{C\log(\tan^2(c\cdots}{2bd\tan\cdots} \\ \frac{Bdx\tan(c+dx)}{2bd\tan(c+dx)+2ibd}+\frac{iBdx}{2bd\tan(c+dx)+2ibd}-\frac{B}{2bd\tan(c+dx)+2ibd}-\frac{iCdx\tan(c+dx)}{2bd\tan(c+dx)+2ibd}+\frac{Cdx}{2bd\tan(c+dx)+2ibd}+\frac{C\log(\tan^2(c\cdots}{2bd\tan\cdots} \\ \frac{x(B\tan(c)+C\tan^2(c))}{a+b\tan(c)} \\ -\frac{2Bab\log(\frac{a}{b}+\tan(c+dx))}{2a^2bd+2b^3d}+\frac{Bab\log(\tan^2(c+dx)+1)}{2a^2bd+2b^3d}+\frac{2Bb^2dx}{2a^2bd+2b^3d}+\frac{2Ca^2\log(\frac{a}{b}+\tan(c+dx))}{2a^2bd+2b^3d}-\frac{2Cabdx}{2a^2bd+2b^3d}+\frac{Cb^2\log(\tan^2\cdots}{2a^2bd+\cdots} \end{cases}$$

```
[In] integrate((B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c)),x)
```

```
[Out] Piecewise((zoo*x*(B*tan(c) + C*tan(c)**2)/tan(c), Eq(a, 0) & Eq(b, 0) & Eq(
d, 0)), ((B*log(tan(c + d*x)**2 + 1)/(2*d) - C*x + C*tan(c + d*x)/d)/a, Eq(
b, 0)), (B*d*x*tan(c + d*x)/(2*b*d*tan(c + d*x) - 2*I*b*d) - I*B*d*x/(2*b*d
*tan(c + d*x) - 2*I*b*d) - B/(2*b*d*tan(c + d*x) - 2*I*b*d) + I*C*d*x*tan(c
 + d*x)/(2*b*d*tan(c + d*x) - 2*I*b*d) + C*d*x/(2*b*d*tan(c + d*x) - 2*I*b*
```

```
d) + C*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(2*b*d*tan(c + d*x) - 2*I*b*d)
 - I*C*log(tan(c + d*x)**2 + 1)/(2*b*d*tan(c + d*x) - 2*I*b*d) - I*C/(2*b*d
*tan(c + d*x) - 2*I*b*d), Eq(a, -I*b)), (B*d*x*tan(c + d*x)/(2*b*d*tan(c +
d*x) + 2*I*b*d) + I*B*d*x/(2*b*d*tan(c + d*x) + 2*I*b*d) - B/(2*b*d*tan(c +
 d*x) + 2*I*b*d) - I*C*d*x*tan(c + d*x)/(2*b*d*tan(c + d*x) + 2*I*b*d) + C*
d*x/(2*b*d*tan(c + d*x) + 2*I*b*d) + C*log(tan(c + d*x)**2 + 1)*tan(c + d*x
)/(2*b*d*tan(c + d*x) + 2*I*b*d) + I*C*log(tan(c + d*x)**2 + 1)/(2*b*d*tan(
c + d*x) + 2*I*b*d) + I*C/(2*b*d*tan(c + d*x) + 2*I*b*d), Eq(a, I*b)), (x*(
B*tan(c) + C*tan(c)**2)/(a + b*tan(c)), Eq(d, 0)), (-2*B*a*b*log(a/b + tan(
c + d*x))/(2*a**2*b*d + 2*b**3*d) + B*a*b*log(tan(c + d*x)**2 + 1)/(2*a**2*
b*d + 2*b**3*d) + 2*B*b**2*d*x/(2*a**2*b*d + 2*b**3*d) + 2*C*a**2*log(a/b +
 tan(c + d*x))/(2*a**2*b*d + 2*b**3*d) - 2*C*a*b*d*x/(2*a**2*b*d + 2*b**3*d
) + C*b**2*log(tan(c + d*x)**2 + 1)/(2*a**2*b*d + 2*b**3*d), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.31 (sec) , antiderivative size = 94, normalized size of antiderivative = 1.11

$$\int \frac{B\tan(c+dx)+C\tan^2(c+dx)}{a+b\tan(c+dx)}\,dx = -\frac{\frac{2\,(Ca-Bb)(dx+c)}{a^2+b^2}-\frac{2\,(Ca^2-Bab)\log(b\tan(dx+c)+a)}{a^2b+b^3}-\frac{(Ba+Cb)\log\left(\tan(dx+c)^2+1\right)}{a^2+b^2}}{2\,d}$$

```
[In] integrate((B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, algorithm="maxi
ma")
```

```
[Out] -1/2*(2*(C*a - B*b)*(d*x + c)/(a^2 + b^2) - 2*(C*a^2 - B*a*b)*log(b*tan(d*x
 + c) + a)/(a^2*b + b^3) - (B*a + C*b)*log(tan(d*x + c)^2 + 1)/(a^2 + b^2))
/d
```

Giac [A] (verification not implemented)

none

Time = 0.50 (sec) , antiderivative size = 95, normalized size of antiderivative = 1.12

$$\int \frac{B\tan(c+dx)+C\tan^2(c+dx)}{a+b\tan(c+dx)}\,dx = -\frac{\frac{2\,(Ca-Bb)(dx+c)}{a^2+b^2}-\frac{(Ba+Cb)\log\left(\tan(dx+c)^2+1\right)}{a^2+b^2}-\frac{2\,(Ca^2-Bab)\log(|b\tan(dx+c)+a|)}{a^2b+b^3}}{2\,d}$$

```
[In] integrate((B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, algorithm="giac
")
```

```
[Out] -1/2*(2*(C*a - B*b)*(d*x + c)/(a^2 + b^2) - (B*a + C*b)*log(tan(d*x + c)^2
+ 1)/(a^2 + b^2) - 2*(C*a^2 - B*a*b)*log(abs(b*tan(d*x + c) + a))/(a^2*b +
b^3))/d
```

Mupad [B] (verification not implemented)

Time = 8.91 (sec) , antiderivative size = 100, normalized size of antiderivative = 1.18

$$\int \frac{B\tan(c+dx)+C\tan^2(c+dx)}{a+b\tan(c+dx)}\,dx = \frac{\ln\left(\tan(c+d\,x)-\mathrm{i}\right)\,\left(-C+B\,1\mathrm{i}\right)}{2\,d\,\left(-b+a\,1\mathrm{i}\right)} + \frac{\ln\left(\tan(c+d\,x)+1\mathrm{i}\right)\,\left(B-C\,1\mathrm{i}\right)}{2\,d\,\left(a-b\,1\mathrm{i}\right)} - \frac{a\,\ln\left(a+b\tan(c+d\,x)\right)\,\left(B\,b-C\,a\right)}{b\,d\,\left(a^2+b^2\right)}$$

```
[In] int((B*tan(c + d*x) + C*tan(c + d*x)^2)/(a + b*tan(c + d*x)),x)
```

```
[Out] (log(tan(c + d*x) - 1i)*(B*1i - C))/(2*d*(a*1i - b)) + (log(tan(c + d*x) +
1i)*(B - C*1i))/(2*d*(a - b*1i)) - (a*log(a + b*tan(c + d*x))*(B*b - C*a))/
(b*d*(a^2 + b^2))
```

3.28 $\int \frac{\cot(c+dx)(B\tan(c+dx)+C\tan^2(c+dx))}{a+b\tan(c+dx)}\,dx$

Optimal result 246
Rubi [A] (verified) 246
Mathematica [A] (verified) 247
Maple [A] (verified) 248
Fricas [A] (verification not implemented) 248
Sympy [C] (verification not implemented) 249
Maxima [A] (verification not implemented) 249
Giac [A] (verification not implemented) 250
Mupad [B] (verification not implemented) 250

Optimal result

Integrand size = 38, antiderivative size = 58

$$\int \frac{\cot(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx = \frac{(aB+bC)x}{a^2+b^2}+\frac{(bB-aC)\log(a\cos(c+dx)+b\sin(c+dx))}{(a^2+b^2)\,d}$$

```
[Out] (B*a+C*b)*x/(a^2+b^2)+(B*b-C*a)*ln(a*cos(d*x+c)+b*sin(d*x+c))/(a^2+b^2)/d
```

Rubi [A] (verified)

Time = 0.16 (sec) , antiderivative size = 58, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 3, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.079, Rules used = {3713, 3612, 3611}

$$\int \frac{\cot(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx = \frac{(bB-aC)\log(a\cos(c+dx)+b\sin(c+dx))}{d\,(a^2+b^2)}+\frac{x(aB+bC)}{a^2+b^2}$$

```
[In] Int[(Cot[c + d*x]*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c + d*x])
,x]
```

```
[Out] ((a*B + b*C)*x)/(a^2 + b^2) + ((b*B - a*C)*Log[a*Cos[c + d*x] + b*Sin[c + d
*x]])/((a^2 + b^2)*d)
```

Rule 3611

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c/(b*f))*Log[RemoveContent[a*Cos[e + f*x] + b*Si
```

```
n[e + f*x], x]], x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3612

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[(a*c + b*d)*(x/(a^2 + b^2)), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\begin{aligned}\text{integral} &= \int \frac{B+C\tan(c+dx)}{a+b\tan(c+dx)}\,dx\\ &= \frac{(aB+bC)x}{a^2+b^2}+\frac{(bB-aC)\int \frac{b-a\tan(c+dx)}{a+b\tan(c+dx)}\,dx}{a^2+b^2}\\ &= \frac{(aB+bC)x}{a^2+b^2}+\frac{(bB-aC)\log(a\cos(c+dx)+b\sin(c+dx))}{(a^2+b^2)\,d}\end{aligned}$$

Mathematica [A] (verified)

Time = 0.16 (sec) , antiderivative size = 67, normalized size of antiderivative = 1.16

$$\int \frac{\cot(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx = \frac{-2(aB+bC)\arctan(\cot(c+dx))+(bB-aC)\left(2\log(b+a\cot(c+dx))-\log\left(\csc^2(c+dx)\right)\right)}{2\left(a^2+b^2\right)d}$$

```
[In] Integrate[(Cot[c + d*x]*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c +
 d*x]),x]
```

```
[Out] (-2*(a*B + b*C)*ArcTan[Cot[c + d*x]] + (b*B - a*C)*(2*Log[b + a*Cot[c + d*x
]] - Log[Csc[c + d*x]^2]))/(2*(a^2 + b^2)*d)
```

Maple [A] (verified)

Time = 0.26 (sec) , antiderivative size = 66, normalized size of antiderivative = 1.14

method	result
parallelrisch	$\frac{(2Bb-2Ca)\ln(a+b\tan(dx+c))+(-Bb+Ca)\ln\left(\sec(dx+c)^2\right)+2dx(Ba+Cb)}{2d(a^2+b^2)}$
derivativedivides	$\frac{\frac{\frac{(-Bb+Ca)\ln\left(1+\tan(dx+c)^2\right)}{2}+(Ba+Cb)\arctan(\tan(dx+c))}{a^2+b^2}+\frac{(Bb-Ca)\ln(a+b\tan(dx+c))}{a^2+b^2}}{d}$
default	$\frac{\frac{\frac{(-Bb+Ca)\ln\left(1+\tan(dx+c)^2\right)}{2}+(Ba+Cb)\arctan(\tan(dx+c))}{a^2+b^2}+\frac{(Bb-Ca)\ln(a+b\tan(dx+c))}{a^2+b^2}}{d}$
norman	$\frac{(Ba+Cb)x}{a^2+b^2}+\frac{(Bb-Ca)\ln(a+b\tan(dx+c))}{d(a^2+b^2)}-\frac{(Bb-Ca)\ln\left(1+\tan(dx+c)^2\right)}{2d(a^2+b^2)}$
risch	$-\frac{xB}{ib-a}+\frac{ixC}{ib-a}-\frac{2iBbx}{a^2+b^2}+\frac{2iCax}{a^2+b^2}-\frac{2iBbc}{d(a^2+b^2)}+\frac{2iCac}{d(a^2+b^2)}+\frac{\ln\left(e^{2i(dx+c)}-\frac{ib+a}{ib-a}\right)Bb}{d(a^2+b^2)}-\frac{\ln\left(e^{2i(dx+c)}-\frac{ib+a}{ib-a}\right)}{d(a^2+b^2)}$

```
[In] int(cot(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x,method=_RET
URNVERBOSE)
```

```
[Out] 1/2*((2*B*b-2*C*a)*ln(a+b*tan(d*x+c))+(-B*b+C*a)*ln(sec(d*x+c)^2)+2*d*x*(B*
a+C*b))/d/(a^2+b^2)
```

Fricas [A] (verification not implemented)

none

Time = 0.26 (sec) , antiderivative size = 76, normalized size of antiderivative = 1.31

$$\int \frac{\cot(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx$$
$$=\frac{2\,(Ba+Cb)dx-(Ca-Bb)\log\left(\frac{b^2\tan(dx+c)^2+2\,ab\tan(dx+c)+a^2}{\tan(dx+c)^2+1}\right)}{2\,(a^2+b^2)d}$$

```
[In] integrate(cot(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, algo
rithm="fricas")
```

```
[Out] 1/2*(2*(B*a + C*b)*d*x - (C*a - B*b)*log((b^2*tan(d*x + c)^2 + 2*a*b*tan(d*
x + c) + a^2)/(tan(d*x + c)^2 + 1)))/((a^2 + b^2)*d)
```

Sympy [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 1.20 (sec) , antiderivative size = 541, normalized size of antiderivative = 9.33

$$\int \frac{\cot(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx$$
$$= \begin{cases} \frac{\tilde{\infty}x(B\tan(c)+C\tan^2(c))\cot(c)}{\tan(c)} \\ \frac{Bx+\frac{C\log\left(\tan^2(c+dx)+1\right)}{2d}}{a} \\ \frac{iBdx\tan(c+dx)}{2bd\tan(c+dx)-2ibd}+\frac{Bdx}{2bd\tan(c+dx)-2ibd}+\frac{iB}{2bd\tan(c+dx)-2ibd}+\frac{Cdx\tan(c+dx)}{2bd\tan(c+dx)-2ibd}-\frac{iCdx}{2bd\tan(c+dx)-2ibd}-\frac{C}{2bd\tan(c+dx)} \\ -\frac{iBdx\tan(c+dx)}{2bd\tan(c+dx)+2ibd}+\frac{Bdx}{2bd\tan(c+dx)+2ibd}-\frac{iB}{2bd\tan(c+dx)+2ibd}+\frac{Cdx\tan(c+dx)}{2bd\tan(c+dx)+2ibd}+\frac{iCdx}{2bd\tan(c+dx)+2ibd}-\frac{C}{2bd\tan(c+} \\ \frac{x(B\tan(c)+C\tan^2(c))\cot(c)}{a+b\tan(c)} \\ \frac{2Badx}{2a^2d+2b^2d}+\frac{2Bb\log\left(\frac{a}{b}+\tan(c+dx)\right)}{2a^2d+2b^2d}-\frac{Bb\log\left(\tan^2(c+dx)+1\right)}{2a^2d+2b^2d}-\frac{2Ca\log\left(\frac{a}{b}+\tan(c+dx)\right)}{2a^2d+2b^2d}+\frac{Ca\log\left(\tan^2(c+dx)+1\right)}{2a^2d+2b^2d}+\frac{2Cbdx}{2a^2d+2b^2} \end{cases}$$

```
[In] integrate(cot(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c)),x)
```

```
[Out] Piecewise((zoo*x*(B*tan(c) + C*tan(c)**2)*cot(c)/tan(c), Eq(a, 0) & Eq(b, 0
) & Eq(d, 0)), ((B*x + C*log(tan(c + d*x)**2 + 1)/(2*d))/a, Eq(b, 0)), (I*B
*d*x*tan(c + d*x)/(2*b*d*tan(c + d*x) - 2*I*b*d) + B*d*x/(2*b*d*tan(c + d*x
) - 2*I*b*d) + I*B/(2*b*d*tan(c + d*x) - 2*I*b*d) + C*d*x*tan(c + d*x)/(2*b
*d*tan(c + d*x) - 2*I*b*d) - I*C*d*x/(2*b*d*tan(c + d*x) - 2*I*b*d) - C/(2*
b*d*tan(c + d*x) - 2*I*b*d), Eq(a, -I*b)), (-I*B*d*x*tan(c + d*x)/(2*b*d*ta
n(c + d*x) + 2*I*b*d) + B*d*x/(2*b*d*tan(c + d*x) + 2*I*b*d) - I*B/(2*b*d*t
an(c + d*x) + 2*I*b*d) + C*d*x*tan(c + d*x)/(2*b*d*tan(c + d*x) + 2*I*b*d)
+ I*C*d*x/(2*b*d*tan(c + d*x) + 2*I*b*d) - C/(2*b*d*tan(c + d*x) + 2*I*b*d)
, Eq(a, I*b)), (x*(B*tan(c) + C*tan(c)**2)*cot(c)/(a + b*tan(c)), Eq(d, 0))
, (2*B*a*d*x/(2*a**2*d + 2*b**2*d) + 2*B*b*log(a/b + tan(c + d*x))/(2*a**2*
d + 2*b**2*d) - B*b*log(tan(c + d*x)**2 + 1)/(2*a**2*d + 2*b**2*d) - 2*C*a*
log(a/b + tan(c + d*x))/(2*a**2*d + 2*b**2*d) + C*a*log(tan(c + d*x)**2 + 1
)/(2*a**2*d + 2*b**2*d) + 2*C*b*d*x/(2*a**2*d + 2*b**2*d), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.44 (sec) , antiderivative size = 88, normalized size of antiderivative = 1.52

$$\int \frac{\cot(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx$$
$$= \frac{\frac{2(Ba+Cb)(dx+c)}{a^2+b^2}-\frac{2(Ca-Bb)\log(b\tan(dx+c)+a)}{a^2+b^2}+\frac{(Ca-Bb)\log\left(\tan(dx+c)^2+1\right)}{a^2+b^2}}{2d}$$

```
[In] integrate(cot(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, algo
rithm="maxima")
```

```
[Out] 1/2*(2*(B*a + C*b)*(d*x + c)/(a^2 + b^2) - 2*(C*a - B*b)*log(b*tan(d*x + c)
 + a)/(a^2 + b^2) + (C*a - B*b)*log(tan(d*x + c)^2 + 1)/(a^2 + b^2))/d
```

Giac [A] (verification not implemented)

none

Time = 0.67 (sec) , antiderivative size = 94, normalized size of antiderivative = 1.62

$$\int \frac{\cot(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx = \frac{\frac{2\,(Ba+Cb)(dx+c)}{a^2+b^2}+\frac{(Ca-Bb)\log\left(\tan(dx+c)^2+1\right)}{a^2+b^2}-\frac{2\,(Cab-Bb^2)\log(|b\tan(dx+c)+a|)}{a^2b+b^3}}{2\,d}$$

```
[In] integrate(cot(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, algo
rithm="giac")
```

```
[Out] 1/2*(2*(B*a + C*b)*(d*x + c)/(a^2 + b^2) + (C*a - B*b)*log(tan(d*x + c)^2 +
 1)/(a^2 + b^2) - 2*(C*a*b - B*b^2)*log(abs(b*tan(d*x + c) + a))/(a^2*b + b
^3))/d
```

Mupad [B] (verification not implemented)

Time = 9.09 (sec) , antiderivative size = 93, normalized size of antiderivative = 1.60

$$\int \frac{\cot(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx = \frac{\ln\left(a+b\tan(c+d\,x)\right)\,\left(B\,b-C\,a\right)}{d\,\left(a^2+b^2\right)}-\frac{\ln\left(\tan(c+d\,x)+1\mathrm{i}\right)\,\left(B-C\,1\mathrm{i}\right)}{2\,d\,\left(b+a\,1\mathrm{i}\right)} - \frac{\ln\left(\tan(c+d\,x)-\mathrm{i}\right)\,\left(-C+B\,1\mathrm{i}\right)}{2\,d\,\left(a+b\,1\mathrm{i}\right)}$$

```
[In] int((cot(c + d*x)*(B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c + d*x))
,x)
```

```
[Out] (log(a + b*tan(c + d*x))*(B*b - C*a))/(d*(a^2 + b^2)) - (log(tan(c + d*x) +
 1i)*(B - C*1i))/(2*d*(a*1i + b)) - (log(tan(c + d*x) - 1i)*(B*1i - C))/(2*
d*(a + b*1i))
```

3.29 $\int \frac{\cot^2(c+dx)(B\tan(c+dx)+C\tan^2(c+dx))}{a+b\tan(c+dx)}\,dx$

Optimal result 251
Rubi [A] (verified) 251
Mathematica [C] (verified) 253
Maple [A] (verified) 253
Fricas [A] (verification not implemented) 254
Sympy [C] (verification not implemented) 254
Maxima [A] (verification not implemented) 255
Giac [A] (verification not implemented) 255
Mupad [B] (verification not implemented) 256

Optimal result

Integrand size = 40, antiderivative size = 80

$$\int \frac{\cot^2(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{a+b\tan(c+dx)}\,dx = -\frac{(bB-aC)x}{a^2+b^2}+\frac{B\log(\sin(c+dx))}{ad}-\frac{b(bB-aC)\log(a\cos(c+dx)+b\sin(c+dx))}{a\,(a^2+b^2)\,d}$$

```
[Out] -(B*b-C*a)*x/(a^2+b^2)+B*ln(sin(d*x+c))/a/d-b*(B*b-C*a)*ln(a*cos(d*x+c)+b*s
in(d*x+c))/a/(a^2+b^2)/d
```

Rubi [A] (verified)

Time = 0.25 (sec) , antiderivative size = 80, normalized size of antiderivative = 1.00, number of steps used = 4, number of rules used = 4, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.100, Rules used = {3713, 3692, 3611, 3556}

$$\int \frac{\cot^2(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{a+b\tan(c+dx)}\,dx = -\frac{b(bB-aC)\log(a\cos(c+dx)+b\sin(c+dx))}{ad\,(a^2+b^2)}-\frac{x(bB-aC)}{a^2+b^2}+\frac{B\log(\sin(c+dx))}{ad}$$

```
[In] Int[(Cot[c + d*x]^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c + d*x
]),x]
```

```
[Out] -(((b*B - a*C)*x)/(a^2 + b^2)) + (B*Log[Sin[c + d*x]])/(a*d) - (b*(b*B - a*
C)*Log[a*Cos[c + d*x] + b*Sin[c + d*x]])/(a*(a^2 + b^2)*d)
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3611

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c/(b*f))*Log[RemoveContent[a*Cos[e + f*x] + b*Si
n[e + f*x], x]], x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3692

```
Int[((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)])/(((a_) + (b_.)*tan[(e_.) + (f_.
)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])), x_Symbol] :> Simp[(B*(b*
c + a*d) + A*(a*c - b*d))*(x/((a^2 + b^2)*(c^2 + d^2))), x] + (Dist[b*((A*b
 - a*B)/((b*c - a*d)*(a^2 + b^2))), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e +
 f*x]), x], x] + Dist[d*((B*c - A*d)/((b*c - a*d)*(c^2 + d^2))), Int[(d - c
*Tan[e + f*x])/(c + d*Tan[e + f*x]), x], x]) /; FreeQ[{a, b, c, d, e, f, A,
 B}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0]
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\text{integral} = \int \frac{\cot(c+dx)(B+C\tan(c+dx))}{a+b\tan(c+dx)}\,dx$$

$$= -\frac{(bB-aC)x}{a^2+b^2} + \frac{B\int \cot(c+dx)\,dx}{a} - \frac{(b(bB-aC))\int \frac{b-a\tan(c+dx)}{a+b\tan(c+dx)}\,dx}{a\,(a^2+b^2)}$$

$$= -\frac{(bB-aC)x}{a^2+b^2} + \frac{B\log(\sin(c+dx))}{ad} - \frac{b(bB-aC)\log(a\cos(c+dx)+b\sin(c+dx))}{a\,(a^2+b^2)\,d}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 0.37 (sec) , antiderivative size = 113, normalized size of antiderivative = 1.41

$$\int \frac{\cot^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx$$
$$=-\frac{\frac{(B+iC)\log(i-\tan(c+dx))}{a+ib}-\frac{2B\log(\tan(c+dx))}{a}+\frac{(B-iC)\log(i+\tan(c+dx))}{a-ib}+\frac{2b(bB-aC)\log(a+b\tan(c+dx))}{a(a^2+b^2)}}{2d}$$

```
[In] Integrate[(Cot[c + d*x]^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c
 + d*x]),x]
```

```
[Out] -1/2*(((B + I*C)*Log[I - Tan[c + d*x]])/(a + I*b) - (2*B*Log[Tan[c + d*x]])
/a + ((B - I*C)*Log[I + Tan[c + d*x]])/(a - I*b) + (2*b*(b*B - a*C)*Log[a +
 b*Tan[c + d*x]])/(a*(a^2 + b^2)))/d
```

Maple [A] (verified)

Time = 0.28 (sec) , antiderivative size = 95, normalized size of antiderivative = 1.19

method	result
parallelrisch	$\frac{\left(-2B\,b^2+2Cab\right)\ln(a+b\tan(dx+c))+\left(-B\,a^2-Cab\right)\ln\left(\sec(dx+c)^2\right)+2B\left(a^2+b^2\right)\ln(\tan(dx+c))-2adx(Bb-Ca)}{2(a^2+b^2)ad}$
derivativedivides	$\frac{\frac{B\ln(\tan(dx+c))}{a}+\frac{\frac{(-Ba-Cb)\ln\left(1+\tan(dx+c)^2\right)}{2}+(-Bb+Ca)\arctan(\tan(dx+c))}{a^2+b^2}-\frac{(Bb-Ca)b\ln(a+b\tan(dx+c))}{\left(a^2+b^2\right)a}}{d}$
default	$\frac{\frac{B\ln(\tan(dx+c))}{a}+\frac{\frac{(-Ba-Cb)\ln\left(1+\tan(dx+c)^2\right)}{2}+(-Bb+Ca)\arctan(\tan(dx+c))}{a^2+b^2}-\frac{(Bb-Ca)b\ln(a+b\tan(dx+c))}{\left(a^2+b^2\right)a}}{d}$
norman	$-\frac{(Bb-Ca)x}{a^2+b^2}+\frac{B\ln(\tan(dx+c))}{ad}-\frac{(Ba+Cb)\ln\left(1+\tan(dx+c)^2\right)}{2d(a^2+b^2)}-\frac{(Bb-Ca)b\ln(a+b\tan(dx+c))}{(a^2+b^2)ad}$
risch	$-\frac{ixB}{ib-a}-\frac{xC}{ib-a}+\frac{2ib^2Bx}{(a^2+b^2)a}+\frac{2ib^2Bc}{(a^2+b^2)ad}-\frac{2ibCx}{a^2+b^2}-\frac{2ibCc}{(a^2+b^2)d}-\frac{2iBx}{a}-\frac{2iBc}{ad}-\frac{b^2\ln\left(\mathrm{e}^{2i(dx+c)}-\frac{ib+a}{ib-a}\right)}{(a^2+b^2)ad}$

```
[In] int(cot(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x,method=_R
ETURNVERBOSE)
```

```
[Out] 1/2*((-2*B*b^2+2*C*a*b)*ln(a+b*tan(d*x+c))+(-B*a^2-C*a*b)*ln(sec(d*x+c)^2)+
2*B*(a^2+b^2)*ln(tan(d*x+c))-2*a*d*x*(B*b-C*a))/(a^2+b^2)/a/d
```

Fricas [A] (verification not implemented)

none

Time = 0.27 (sec) , antiderivative size = 118, normalized size of antiderivative = 1.48

$$\int \frac{\cot^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx = \frac{2\left(Ca^2-Bab\right)dx+\left(Ba^2+Bb^2\right)\log\left(\frac{\tan(dx+c)^2}{\tan(dx+c)^2+1}\right)+\left(Cab-Bb^2\right)\log\left(\frac{b^2\tan(dx+c)^2+2\,ab\tan(dx+c)+a^2}{\tan(dx+c)^2+1}\right)}{2\left(a^3+ab^2\right)d}$$

```
[In] integrate(cot(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, al
gorithm="fricas")
```

```
[Out] 1/2*(2*(C*a^2 - B*a*b)*d*x + (B*a^2 + B*b^2)*log(tan(d*x + c)^2/(tan(d*x +
c)^2 + 1)) + (C*a*b - B*b^2)*log((b^2*tan(d*x + c)^2 + 2*a*b*tan(d*x + c) +
 a^2)/(tan(d*x + c)^2 + 1)))/((a^3 + a*b^2)*d)
```

Sympy [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 2.17 (sec) , antiderivative size = 966, normalized size of antiderivative = 12.08

$$\int \frac{\cot^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx = \text{Too large to display}$$

```
[In] integrate(cot(d*x+c)**2*(B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c)),x)
```

```
[Out] Piecewise((zoo*x*(B*tan(c) + C*tan(c)**2)*cot(c)**2/tan(c), Eq(a, 0) & Eq(b
, 0) & Eq(d, 0)), ((-B*log(tan(c + d*x)**2 + 1)/(2*d) + B*log(tan(c + d*x))
/d + C*x)/a, Eq(b, 0)), ((-B*x - B/(d*tan(c + d*x)) - C*log(tan(c + d*x)**2
 + 1)/(2*d) + C*log(tan(c + d*x))/d)/b, Eq(a, 0)), (B*d*x*tan(c + d*x)/(2*b
*d*tan(c + d*x) - 2*I*b*d) - I*B*d*x/(2*b*d*tan(c + d*x) - 2*I*b*d) - I*B*l
og(tan(c + d*x)**2 + 1)*tan(c + d*x)/(2*b*d*tan(c + d*x) - 2*I*b*d) - B*log
(tan(c + d*x)**2 + 1)/(2*b*d*tan(c + d*x) - 2*I*b*d) + 2*I*B*log(tan(c + d*
x))*tan(c + d*x)/(2*b*d*tan(c + d*x) - 2*I*b*d) + 2*B*log(tan(c + d*x))/(2*
b*d*tan(c + d*x) - 2*I*b*d) + B/(2*b*d*tan(c + d*x) - 2*I*b*d) + I*C*d*x*ta
n(c + d*x)/(2*b*d*tan(c + d*x) - 2*I*b*d) + C*d*x/(2*b*d*tan(c + d*x) - 2*I
*b*d) + I*C/(2*b*d*tan(c + d*x) - 2*I*b*d), Eq(a, -I*b)), (B*d*x*tan(c + d*
x)/(2*b*d*tan(c + d*x) + 2*I*b*d) + I*B*d*x/(2*b*d*tan(c + d*x) + 2*I*b*d)
+ I*B*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(2*b*d*tan(c + d*x) + 2*I*b*d)
- B*log(tan(c + d*x)**2 + 1)/(2*b*d*tan(c + d*x) + 2*I*b*d) - 2*I*B*log(tan
(c + d*x))*tan(c + d*x)/(2*b*d*tan(c + d*x) + 2*I*b*d) + 2*B*log(tan(c + d*
x))/(2*b*d*tan(c + d*x) + 2*I*b*d) + B/(2*b*d*tan(c + d*x) + 2*I*b*d) - I*C
*d*x*tan(c + d*x)/(2*b*d*tan(c + d*x) + 2*I*b*d) + C*d*x/(2*b*d*tan(c + d*x
```

```
) + 2*I*b*d) - I*C/(2*b*d*tan(c + d*x) + 2*I*b*d), Eq(a, I*b)), (x*(B*tan(c
) + C*tan(c)**2)*cot(c)**2/(a + b*tan(c)), Eq(d, 0)), (-B*a**2*log(tan(c +
d*x)**2 + 1)/(2*a**3*d + 2*a*b**2*d) + 2*B*a**2*log(tan(c + d*x))/(2*a**3*d
 + 2*a*b**2*d) - 2*B*a*b*d*x/(2*a**3*d + 2*a*b**2*d) - 2*B*b**2*log(a/b + t
an(c + d*x))/(2*a**3*d + 2*a*b**2*d) + 2*B*b**2*log(tan(c + d*x))/(2*a**3*d
 + 2*a*b**2*d) + 2*C*a**2*d*x/(2*a**3*d + 2*a*b**2*d) + 2*C*a*b*log(a/b + t
an(c + d*x))/(2*a**3*d + 2*a*b**2*d) - C*a*b*log(tan(c + d*x)**2 + 1)/(2*a*
*3*d + 2*a*b**2*d), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.38 (sec) , antiderivative size = 107, normalized size of antiderivative = 1.34

$$\int \frac{\cot^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx = \frac{\frac{2\,(Ca-Bb)(dx+c)}{a^2+b^2}+\frac{2\,(Cab-Bb^2)\log(b\tan(dx+c)+a)}{a^3+ab^2}-\frac{(Ba+Cb)\log\left(\tan(dx+c)^2+1\right)}{a^2+b^2}+\frac{2\,B\log(\tan(dx+c))}{a}}{2\,d}$$

```
[In] integrate(cot(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, al
gorithm="maxima")
```

```
[Out] 1/2*(2*(C*a - B*b)*(d*x + c)/(a^2 + b^2) + 2*(C*a*b - B*b^2)*log(b*tan(d*x
+ c) + a)/(a^3 + a*b^2) - (B*a + C*b)*log(tan(d*x + c)^2 + 1)/(a^2 + b^2) +
 2*B*log(tan(d*x + c))/a)/d
```

Giac [A] (verification not implemented)

none

Time = 0.80 (sec) , antiderivative size = 113, normalized size of antiderivative = 1.41

$$\int \frac{\cot^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx = \frac{\frac{2\,(Ca-Bb)(dx+c)}{a^2+b^2}-\frac{(Ba+Cb)\log\left(\tan(dx+c)^2+1\right)}{a^2+b^2}+\frac{2\,(Cab^2-Bb^3)\log(|b\tan(dx+c)+a|)}{a^3b+ab^3}+\frac{2\,B\log(|\tan(dx+c)|)}{a}}{2\,d}$$

```
[In] integrate(cot(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, al
gorithm="giac")
```

```
[Out] 1/2*(2*(C*a - B*b)*(d*x + c)/(a^2 + b^2) - (B*a + C*b)*log(tan(d*x + c)^2 +
 1)/(a^2 + b^2) + 2*(C*a*b^2 - B*b^3)*log(abs(b*tan(d*x + c) + a))/(a^3*b +
 a*b^3) + 2*B*log(abs(tan(d*x + c)))/a)/d
```

Mupad [B] (verification not implemented)

Time = 9.20 (sec) , antiderivative size = 115, normalized size of antiderivative = 1.44

$$\int \frac{\cot^2(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{a+b\tan(c+dx)}\,dx$$
$$=\frac{B\,\ln(\tan(c+d\,x))}{a\,d}-\frac{\ln(\tan(c+d\,x)-\mathrm{i})\,\left(-C+B\,1\mathrm{i}\right)}{2\,d\,\left(-b+a\,1\mathrm{i}\right)}$$
$$-\frac{\ln(\tan(c+d\,x)+1\mathrm{i})\,\left(B-C\,1\mathrm{i}\right)}{2\,d\,\left(a-b\,1\mathrm{i}\right)}-\frac{b\,\ln(a+b\tan(c+d\,x))\,\left(B\,b-C\,a\right)}{a\,d\,\left(a^2+b^2\right)}$$

```
[In] int((cot(c + d*x)^2*(B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c + d*x
)),x)
```

```
[Out] (B*log(tan(c + d*x)))/(a*d) - (log(tan(c + d*x) - 1i)*(B*1i - C))/(2*d*(a*1
i - b)) - (log(tan(c + d*x) + 1i)*(B - C*1i))/(2*d*(a - b*1i)) - (b*log(a +
 b*tan(c + d*x))*(B*b - C*a))/(a*d*(a^2 + b^2))
```

3.30 $\int \frac{\cot^3(c+dx)(B\tan(c+dx)+C\tan^2(c+dx))}{a+b\tan(c+dx)}\,dx$

Optimal result 257
Rubi [A] (verified) 257
Mathematica [C] (verified) 259
Maple [A] (verified) 260
Fricas [A] (verification not implemented) 260
Sympy [C] (verification not implemented) 261
Maxima [A] (verification not implemented) 262
Giac [A] (verification not implemented) 263
Mupad [B] (verification not implemented) 263

Optimal result

Integrand size = 40, antiderivative size = 103

$$\int \frac{\cot^3(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx = -\frac{(aB+bC)x}{a^2+b^2}-\frac{B\cot(c+dx)}{ad}-\frac{(bB-aC)\log(\sin(c+dx))}{a^2d}+\frac{b^2(bB-aC)\log(a\cos(c+dx)+b\sin(c+dx))}{a^2\left(a^2+b^2\right)d}$$

```
[Out] -(B*a+C*b)*x/(a^2+b^2)-B*cot(d*x+c)/a/d-(B*b-C*a)*ln(sin(d*x+c))/a^2/d+b^2*
(B*b-C*a)*ln(a*cos(d*x+c)+b*sin(d*x+c))/a^2/(a^2+b^2)/d
```

Rubi [A] (verified)

Time = 0.41 (sec) , antiderivative size = 103, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.125, Rules used = {3713, 3690, 3732, 3611, 3556}

$$\int \frac{\cot^3(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx = \frac{b^2(bB-aC)\log(a\cos(c+dx)+b\sin(c+dx))}{a^2d\left(a^2+b^2\right)}-\frac{x(aB+bC)}{a^2+b^2}-\frac{(bB-aC)\log(\sin(c+dx))}{a^2d}-\frac{B\cot(c+dx)}{ad}$$

```
[In] Int[(Cot[c + d*x]^3*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c + d*x
]),x]
```

```
[Out] -(((a*B + b*C)*x)/(a^2 + b^2)) - (B*Cot[c + d*x])/(a*d) - ((b*B - a*C)*Log[
Sin[c + d*x]])/(a^2*d) + (b^2*(b*B - a*C)*Log[a*Cos[c + d*x] + b*Sin[c + d*
x]])/(a^2*(a^2 + b^2)*d)
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3611

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c/(b*f))*Log[RemoveContent[a*Cos[e + f*x] + b*Si
n[e + f*x], x]], x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3690

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Si
mp[b*(A*b - a*B)*(a + b*Tan[e + f*x])^(m + 1)*((c + d*Tan[e + f*x])^(n + 1)
/(f*(m + 1)*(b*c - a*d)*(a^2 + b^2))), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^
2 + b^2)), Int[(a + b*Tan[e + f*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[b*B
*(b*c*(m + 1) + a*d*(n + 1)) + A*(a*(b*c - a*d)*(m + 1) - b^2*d*(m + n + 2)
) - (A*b - a*B)*(b*c - a*d)*(m + 1)*Tan[e + f*x] - b*d*(A*b - a*B)*(m + n +
 2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, n}, x] &&
NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &
& (IntegerQ[m] || IntegersQ[2*m, 2*n]) &&  !(ILtQ[n, -1] && ( !IntegerQ[m]
|| (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rule 3732

```
Int[((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^
2)/(((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])), x_Symbol] :> Simp[(a*(A*c - c*C + B*d) + b*(B*c - A*d + C*d))*(x/
((a^2 + b^2)*(c^2 + d^2))), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/((b*c - a*d)
*(a^2 + b^2)), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] - Dist
[(c^2*C - B*c*d + A*d^2)/((b*c - a*d)*(c^2 + d^2)), Int[(d - c*Tan[e + f*x]
```

```
)/(c + d*Tan[e + f*x]), x], x]) /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] &&
 NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{\cot^2(c+dx)(B+C\tan(c+dx))}{a+b\tan(c+dx)}\,dx \\
&= -\frac{B\cot(c+dx)}{ad} - \frac{\int \frac{\cot(c+dx)(bB-aC+aB\tan(c+dx)+bB\tan^2(c+dx))}{a+b\tan(c+dx)}\,dx}{a} \\
&= -\frac{(aB+bC)x}{a^2+b^2} - \frac{B\cot(c+dx)}{ad} - \frac{(bB-aC)\int\cot(c+dx)\,dx}{a^2} \\
&\quad + \frac{(b^2(bB-aC))\int\frac{b-a\tan(c+dx)}{a+b\tan(c+dx)}\,dx}{a^2(a^2+b^2)} \\
&= -\frac{(aB+bC)x}{a^2+b^2} - \frac{B\cot(c+dx)}{ad} - \frac{(bB-aC)\log(\sin(c+dx))}{a^2d} \\
&\quad + \frac{b^2(bB-aC)\log(a\cos(c+dx)+b\sin(c+dx))}{a^2(a^2+b^2)\,d}
\end{aligned}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 0.98 (sec) , antiderivative size = 138, normalized size of antiderivative = 1.34

$$\int \frac{\cot^3(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx$$
$$= \frac{-\frac{2B\cot(c+dx)}{a} + \frac{i(B+iC)\log(i-\tan(c+dx))}{a+ib} + \frac{2(-bB+aC)\log(\tan(c+dx))}{a^2} - \frac{(iB+C)\log(i+\tan(c+dx))}{a-ib} + \frac{2b^2(bB-aC)\log(a+b\tan(c}{a^2(a^2+b^2)}}{2d}$$

```
[In] Integrate[(Cot[c + d*x]^3*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c
 + d*x]),x]
```

```
[Out] ((-2*B*Cot[c + d*x])/a + (I*(B + I*C)*Log[I - Tan[c + d*x]])/(a + I*b) + (2
*(-(b*B) + a*C)*Log[Tan[c + d*x]])/a^2 - ((I*B + C)*Log[I + Tan[c + d*x]])/
(a - I*b) + (2*b^2*(b*B - a*C)*Log[a + b*Tan[c + d*x]])/(a^2*(a^2 + b^2)))/
(2*d)
```

Maple [A] (verified)

Time = 0.32 (sec) , antiderivative size = 122, normalized size of antiderivative = 1.18

method	result
parallelrisch	$\frac{(2Bb^3-2Cab^2)\ln(a+b\tan(dx+c))+(Ba^2b-Ca^3)\ln\left(\sec(dx+c)^2\right)-2(a^2+b^2)(Bb-Ca)\ln(\tan(dx+c))-2a(B(a^2+b^2)\cdots}{2a^2d(a^2+b^2)}$
derivativedivides	$\frac{-\frac{B}{a\tan(dx+c)}+\frac{(-Bb+Ca)\ln(\tan(dx+c))}{a^2}+\frac{\frac{(Bb-Ca)\ln\left(1+\tan(dx+c)^2\right)}{2}+(-Ba-Cb)\arctan(\tan(dx+c))}{a^2+b^2}+\frac{(Bb-Ca)b^2\ln(a+b\tan(d\cdots}{\left(a^2+b^2\right)a^2}}{d}$
default	$\frac{-\frac{B}{a\tan(dx+c)}+\frac{(-Bb+Ca)\ln(\tan(dx+c))}{a^2}+\frac{\frac{(Bb-Ca)\ln\left(1+\tan(dx+c)^2\right)}{2}+(-Ba-Cb)\arctan(\tan(dx+c))}{a^2+b^2}+\frac{(Bb-Ca)b^2\ln(a+b\tan(d\cdots}{\left(a^2+b^2\right)a^2}}{d}$
norman	$\frac{-\frac{B\tan(dx+c)}{ad}-\frac{(Ba+Cb)x\tan(dx+c)^2}{a^2+b^2}}{\tan(dx+c)^2}+\frac{(Bb-Ca)b^2\ln(a+b\tan(dx+c))}{a^2d(a^2+b^2)}-\frac{(Bb-Ca)\ln(\tan(dx+c))}{a^2d}+\frac{(Bb-Ca)\ln\left(1\cdots\right.}{2d(a^2\cdots}$
risch	$\frac{xB}{ib-a}-\frac{ixC}{ib-a}+\frac{2iBbx}{a^2}+\frac{2iBbc}{a^2d}-\frac{2iCx}{a}-\frac{2iCc}{ad}-\frac{2ib^3Bx}{a^2(a^2+b^2)}-\frac{2ib^3Bc}{a^2d(a^2+b^2)}+\frac{2ib^2Cx}{a(a^2+b^2)}+\frac{2ib^2Cc}{ad(a^2+b^2)}-\cdots$

```
[In] int(cot(d*x+c)^3*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x,method=_R
ETURNVERBOSE)
```

```
[Out] 1/2*((2*B*b^3-2*C*a*b^2)*ln(a+b*tan(d*x+c))+(B*a^2*b-C*a^3)*ln(sec(d*x+c)^2
)-2*(a^2+b^2)*(B*b-C*a)*ln(tan(d*x+c))-2*a*(B*(a^2+b^2)*cot(d*x+c)+a*d*x*(B
*a+C*b)))/a^2/d/(a^2+b^2)
```

Fricas [A] (verification not implemented)

none

Time = 0.27 (sec) , antiderivative size = 177, normalized size of antiderivative = 1.72

$$\int \frac{\cot^3(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{a+b\tan(c+dx)}\,dx = -\frac{2\,Ba^3+2\,Bab^2+2\,(Ba^3+Ca^2b)dx\tan{(dx+c)}-(Ca^3-Ba^2b+Cab^2-Bb^3)\log\left(\frac{\tan(dx+c)^2}{\tan(dx+c)^2+1}\right)\tan{(\cdots}}{2\,(a^4+a^2b^2)d\tan{(dx+c)}}$$

```
[In] integrate(cot(d*x+c)^3*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, al
gorithm="fricas")
```

```
[Out] -1/2*(2*B*a^3 + 2*B*a*b^2 + 2*(B*a^3 + C*a^2*b)*d*x*tan(d*x + c) - (C*a^3 -
 B*a^2*b + C*a*b^2 - B*b^3)*log(tan(d*x + c)^2/(tan(d*x + c)^2 + 1))*tan(d*
x + c) + (C*a*b^2 - B*b^3)*log((b^2*tan(d*x + c)^2 + 2*a*b*tan(d*x + c) + a
^2)/(tan(d*x + c)^2 + 1))*tan(d*x + c))/((a^4 + a^2*b^2)*d*tan(d*x + c))
```

Sympy [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 3.72 (sec) , antiderivative size = 2067, normalized size of antiderivative = 20.07

$$\int \frac{\cot^3(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx = \text{Too large to display}$$

```
[In] integrate(cot(d*x+c)**3*(B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c)),x)
```

```
[Out] Piecewise((nan, Eq(a, 0) & Eq(b, 0) & Eq(c, 0) & Eq(d, 0)), ((-B*x - B/(d*t
an(c + d*x)) - C*log(tan(c + d*x)**2 + 1)/(2*d) + C*log(tan(c + d*x))/d)/a,
 Eq(b, 0)), ((B*log(tan(c + d*x)**2 + 1)/(2*d) - B*log(tan(c + d*x))/d - B/
(2*d*tan(c + d*x)**2) - C*x - C/(d*tan(c + d*x)))/b, Eq(a, 0)), (-3*B*d*x*t
an(c + d*x)**2/(2*a*d*tan(c + d*x)**2 + 2*I*a*d*tan(c + d*x)) - 3*I*B*d*x*t
an(c + d*x)/(2*a*d*tan(c + d*x)**2 + 2*I*a*d*tan(c + d*x)) - I*B*log(tan(c
+ d*x)**2 + 1)*tan(c + d*x)**2/(2*a*d*tan(c + d*x)**2 + 2*I*a*d*tan(c + d*x
)) + B*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(2*a*d*tan(c + d*x)**2 + 2*I*a
*d*tan(c + d*x)) + 2*I*B*log(tan(c + d*x))*tan(c + d*x)**2/(2*a*d*tan(c + d
*x)**2 + 2*I*a*d*tan(c + d*x)) - 2*B*log(tan(c + d*x))*tan(c + d*x)/(2*a*d*
tan(c + d*x)**2 + 2*I*a*d*tan(c + d*x)) - 3*B*tan(c + d*x)/(2*a*d*tan(c + d
*x)**2 + 2*I*a*d*tan(c + d*x)) - 2*I*B/(2*a*d*tan(c + d*x)**2 + 2*I*a*d*tan
(c + d*x)) + I*C*d*x*tan(c + d*x)**2/(2*a*d*tan(c + d*x)**2 + 2*I*a*d*tan(c
 + d*x)) - C*d*x*tan(c + d*x)/(2*a*d*tan(c + d*x)**2 + 2*I*a*d*tan(c + d*x)
) - C*log(tan(c + d*x)**2 + 1)*tan(c + d*x)**2/(2*a*d*tan(c + d*x)**2 + 2*I
*a*d*tan(c + d*x)) - I*C*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(2*a*d*tan(c
 + d*x)**2 + 2*I*a*d*tan(c + d*x)) + 2*C*log(tan(c + d*x))*tan(c + d*x)**2/
(2*a*d*tan(c + d*x)**2 + 2*I*a*d*tan(c + d*x)) + 2*I*C*log(tan(c + d*x))*ta
n(c + d*x)/(2*a*d*tan(c + d*x)**2 + 2*I*a*d*tan(c + d*x)) + I*C*tan(c + d*x
)/(2*a*d*tan(c + d*x)**2 + 2*I*a*d*tan(c + d*x)), Eq(b, -I*a)), (-3*B*d*x*t
an(c + d*x)**2/(2*a*d*tan(c + d*x)**2 - 2*I*a*d*tan(c + d*x)) + 3*I*B*d*x*t
an(c + d*x)/(2*a*d*tan(c + d*x)**2 - 2*I*a*d*tan(c + d*x)) + I*B*log(tan(c
+ d*x)**2 + 1)*tan(c + d*x)**2/(2*a*d*tan(c + d*x)**2 - 2*I*a*d*tan(c + d*x
)) + B*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(2*a*d*tan(c + d*x)**2 - 2*I*a
*d*tan(c + d*x)) - 2*I*B*log(tan(c + d*x))*tan(c + d*x)**2/(2*a*d*tan(c + d
*x)**2 - 2*I*a*d*tan(c + d*x)) - 2*B*log(tan(c + d*x))*tan(c + d*x)/(2*a*d*
tan(c + d*x)**2 - 2*I*a*d*tan(c + d*x)) - 3*B*tan(c + d*x)/(2*a*d*tan(c + d
*x)**2 - 2*I*a*d*tan(c + d*x)) + 2*I*B/(2*a*d*tan(c + d*x)**2 - 2*I*a*d*tan
(c + d*x)) - I*C*d*x*tan(c + d*x)**2/(2*a*d*tan(c + d*x)**2 - 2*I*a*d*tan(c
 + d*x)) - C*d*x*tan(c + d*x)/(2*a*d*tan(c + d*x)**2 - 2*I*a*d*tan(c + d*x)
) - C*log(tan(c + d*x)**2 + 1)*tan(c + d*x)**2/(2*a*d*tan(c + d*x)**2 - 2*I
*a*d*tan(c + d*x)) + I*C*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(2*a*d*tan(c
 + d*x)**2 - 2*I*a*d*tan(c + d*x)) + 2*C*log(tan(c + d*x))*tan(c + d*x)**2/
(2*a*d*tan(c + d*x)**2 - 2*I*a*d*tan(c + d*x)) - 2*I*C*log(tan(c + d*x))*ta
n(c + d*x)/(2*a*d*tan(c + d*x)**2 - 2*I*a*d*tan(c + d*x)) - I*C*tan(c + d*x
```

```
)/(2*a*d*tan(c + d*x)**2 - 2*I*a*d*tan(c + d*x)), Eq(b, I*a)), (nan, Eq(c,
-d*x)), (x*(B*tan(c) + C*tan(c)**2)*cot(c)**3/(a + b*tan(c)), Eq(d, 0)), (-
2*B*a**3*d*x*tan(c + d*x)/(2*a**4*d*tan(c + d*x) + 2*a**2*b**2*d*tan(c + d*
x)) - 2*B*a**3/(2*a**4*d*tan(c + d*x) + 2*a**2*b**2*d*tan(c + d*x)) + B*a**
2*b*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(2*a**4*d*tan(c + d*x) + 2*a**2*b
**2*d*tan(c + d*x)) - 2*B*a**2*b*log(tan(c + d*x))*tan(c + d*x)/(2*a**4*d*t
an(c + d*x) + 2*a**2*b**2*d*tan(c + d*x)) - 2*B*a*b**2/(2*a**4*d*tan(c + d*
x) + 2*a**2*b**2*d*tan(c + d*x)) + 2*B*b**3*log(a/b + tan(c + d*x))*tan(c +
 d*x)/(2*a**4*d*tan(c + d*x) + 2*a**2*b**2*d*tan(c + d*x)) - 2*B*b**3*log(t
an(c + d*x))*tan(c + d*x)/(2*a**4*d*tan(c + d*x) + 2*a**2*b**2*d*tan(c + d*
x)) - C*a**3*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(2*a**4*d*tan(c + d*x) +
 2*a**2*b**2*d*tan(c + d*x)) + 2*C*a**3*log(tan(c + d*x))*tan(c + d*x)/(2*a
**4*d*tan(c + d*x) + 2*a**2*b**2*d*tan(c + d*x)) - 2*C*a**2*b*d*x*tan(c + d
*x)/(2*a**4*d*tan(c + d*x) + 2*a**2*b**2*d*tan(c + d*x)) - 2*C*a*b**2*log(a
/b + tan(c + d*x))*tan(c + d*x)/(2*a**4*d*tan(c + d*x) + 2*a**2*b**2*d*tan(
c + d*x)) + 2*C*a*b**2*log(tan(c + d*x))*tan(c + d*x)/(2*a**4*d*tan(c + d*x
) + 2*a**2*b**2*d*tan(c + d*x)), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.42 (sec) , antiderivative size = 131, normalized size of antiderivative = 1.27

$$\int \frac{\cot^3(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx = -\frac{\frac{2\,(Ba+Cb)(dx+c)}{a^2+b^2}+\frac{2\,(Cab^2-Bb^3)\log(b\tan(dx+c)+a)}{a^4+a^2b^2}+\frac{(Ca-Bb)\log\left(\tan(dx+c)^2+1\right)}{a^2+b^2}-\frac{2\,(Ca-Bb)\log(\tan(dx+c))}{a^2}+\frac{2\,B}{a\tan(dx+c)}}{2\,d}$$

```
[In] integrate(cot(d*x+c)^3*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, al
gorithm="maxima")
```

```
[Out] -1/2*(2*(B*a + C*b)*(d*x + c)/(a^2 + b^2) + 2*(C*a*b^2 - B*b^3)*log(b*tan(d
*x + c) + a)/(a^4 + a^2*b^2) + (C*a - B*b)*log(tan(d*x + c)^2 + 1)/(a^2 + b
^2) - 2*(C*a - B*b)*log(tan(d*x + c))/a^2 + 2*B/(a*tan(d*x + c)))/d
```

Giac [A] (verification not implemented)

none

Time = 1.07 (sec) , antiderivative size = 157, normalized size of antiderivative = 1.52

$$\int \frac{\cot^3(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx = -\frac{\frac{2\,(Ba+Cb)(dx+c)}{a^2+b^2}+\frac{(Ca-Bb)\log\left(\tan(dx+c)^2+1\right)}{a^2+b^2}+\frac{2\,(Cab^3-Bb^4)\log(|b\tan(dx+c)+a|)}{a^4b+a^2b^3}-\frac{2\,(Ca-Bb)\log(|\tan(dx+c)|)}{a^2}+\frac{2\,(Ca\tan}{}}{2\,d}$$

```
[In] integrate(cot(d*x+c)^3*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, al
gorithm="giac")
```

```
[Out] -1/2*(2*(B*a + C*b)*(d*x + c)/(a^2 + b^2) + (C*a - B*b)*log(tan(d*x + c)^2
+ 1)/(a^2 + b^2) + 2*(C*a*b^3 - B*b^4)*log(abs(b*tan(d*x + c) + a))/(a^4*b
+ a^2*b^3) - 2*(C*a - B*b)*log(abs(tan(d*x + c)))/a^2 + 2*(C*a*tan(d*x + c)
 - B*b*tan(d*x + c) + B*a)/(a^2*tan(d*x + c)))/d
```

Mupad [B] (verification not implemented)

Time = 9.88 (sec) , antiderivative size = 140, normalized size of antiderivative = 1.36

$$\int \frac{\cot^3(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx = \frac{\ln\left(a+b\tan(c+d\,x)\right)\,\left(B\,b^3-C\,a\,b^2\right)}{d\,\left(a^4+a^2\,b^2\right)}-\frac{\ln\left(\tan(c+d\,x)\right)\,\left(B\,b-C\,a\right)}{a^2\,d}+\frac{\ln\left(\tan(c+d\,x)+1\mathrm{i}\right)\,\left(B-C\,1\mathrm{i}\right)}{2\,d\,\left(b+a\,1\mathrm{i}\right)}-\frac{B\cot(c+d\,x)}{a\,d}+\frac{\ln\left(\tan(c+d\,x)-\mathrm{i}\right)\,\left(-C+B\,1\mathrm{i}\right)}{2\,d\,\left(a+b\,1\mathrm{i}\right)}$$

```
[In] int((cot(c + d*x)^3*(B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c + d*x
)),x)
```

```
[Out] (log(a + b*tan(c + d*x))*(B*b^3 - C*a*b^2))/(d*(a^4 + a^2*b^2)) - (log(tan(
c + d*x))*(B*b - C*a))/(a^2*d) + (log(tan(c + d*x) + 1i)*(B - C*1i))/(2*d*(
a*1i + b)) - (B*cot(c + d*x))/(a*d) + (log(tan(c + d*x) - 1i)*(B*1i - C))/(
2*d*(a + b*1i))
```

3.31 $\int \frac{\cot^4(c+dx)(B\tan(c+dx)+C\tan^2(c+dx))}{a+b\tan(c+dx)}\,dx$

Optimal result 264
Rubi [A] (verified) 264
Mathematica [C] (verified) 267
Maple [A] (verified) 267
Fricas [A] (verification not implemented) 268
Sympy [C] (verification not implemented) 268
Maxima [A] (verification not implemented) 270
Giac [A] (verification not implemented) 270
Mupad [B] (verification not implemented) 271

Optimal result

Integrand size = 40, antiderivative size = 137

$$\begin{aligned}\int \frac{\cot^4(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx &= \frac{(bB-aC)x}{a^2+b^2}+\frac{(bB-aC)\cot(c+dx)}{a^2d}-\frac{B\cot^2(c+dx)}{2ad}\\ &\quad-\frac{(a^2B-b^2B+abC)\log(\sin(c+dx))}{a^3d}-\frac{b^3(bB-aC)\log(a\cos(c+dx)+b\sin(c+dx))}{a^3\left(a^2+b^2\right)d}\end{aligned}$$

```
[Out] (B*b-C*a)*x/(a^2+b^2)+(B*b-C*a)*cot(d*x+c)/a^2/d-1/2*B*cot(d*x+c)^2/a/d-(B*
a^2-B*b^2+C*a*b)*ln(sin(d*x+c))/a^3/d-b^3*(B*b-C*a)*ln(a*cos(d*x+c)+b*sin(d
*x+c))/a^3/(a^2+b^2)/d
```

Rubi [A] (verified)

Time = 0.72 (sec) , antiderivative size = 137, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.150, Rules used = {3713, 3690, 3730, 3732, 3611, 3556}

$$\begin{aligned}\int \frac{\cot^4(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx &= \frac{x(bB-aC)}{a^2+b^2}+\frac{(bB-aC)\cot(c+dx)}{a^2d}-\frac{(a^2B+abC-b^2B)\log(\sin(c+dx))}{a^3d}\\ &\quad-\frac{b^3(bB-aC)\log(a\cos(c+dx)+b\sin(c+dx))}{a^3d\left(a^2+b^2\right)}-\frac{B\cot^2(c+dx)}{2ad}\end{aligned}$$

```
[In] Int[(Cot[c + d*x]^4*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c + d*x
]),x]
```

```
[Out] ((b*B - a*C)*x)/(a^2 + b^2) + ((b*B - a*C)*Cot[c + d*x])/(a^2*d) - (B*Cot[c
 + d*x]^2)/(2*a*d) - ((a^2*B - b^2*B + a*b*C)*Log[Sin[c + d*x]])/(a^3*d) -
(b^3*(b*B - a*C)*Log[a*Cos[c + d*x] + b*Sin[c + d*x]])/(a^3*(a^2 + b^2)*d)
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3611

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c/(b*f))*Log[RemoveContent[a*Cos[e + f*x] + b*Si
n[e + f*x], x]], x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3690

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Si
mp[b*(A*b - a*B)*(a + b*Tan[e + f*x])^(m + 1)*((c + d*Tan[e + f*x])^(n + 1)
/(f*(m + 1)*(b*c - a*d)*(a^2 + b^2))), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^
2 + b^2)), Int[(a + b*Tan[e + f*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[b*B
*(b*c*(m + 1) + a*d*(n + 1)) + A*(a*(b*c - a*d)*(m + 1) - b^2*d*(m + n + 2)
) - (A*b - a*B)*(b*c - a*d)*(m + 1)*Tan[e + f*x] - b*d*(A*b - a*B)*(m + n +
 2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, n}, x] &&
NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &
& (IntegerQ[m] || IntegersQ[2*m, 2*n]) &&  !(ILtQ[n, -1] && ( !IntegerQ[m]
|| (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rule 3730

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 - a*(b*B - a*C))*(a + b*Tan[e +
 f*x])^(m + 1)*((c + d*Tan[e + f*x])^(n + 1)/(f*(m + 1)*(b*c - a*d)*(a^2 +
b^2))), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
```

```
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3732

```
Int[((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^
2)/(((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])), x_Symbol] :> Simp[(a*(A*c - c*C + B*d) + b*(B*c - A*d + C*d))*(x/
((a^2 + b^2)*(c^2 + d^2))), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/((b*c - a*d)
*(a^2 + b^2)), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] - Dist
[(c^2*C - B*c*d + A*d^2)/((b*c - a*d)*(c^2 + d^2)), Int[(d - c*Tan[e + f*x]
)/(c + d*Tan[e + f*x]), x], x]) /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] &&
 NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0]
```

Rubi steps

$$\text{integral} = \int \frac{\cot^3(c+dx)(B+C\tan(c+dx))}{a+b\tan(c+dx)}\,dx$$

$$= -\frac{B\cot^2(c+dx)}{2ad} - \frac{\int \frac{\cot^2(c+dx)(2(bB-aC)+2aB\tan(c+dx)+2bB\tan^2(c+dx))}{a+b\tan(c+dx)}\,dx}{2a}$$

$$= \frac{(bB-aC)\cot(c+dx)}{a^2d} - \frac{B\cot^2(c+dx)}{2ad} + \frac{\int \frac{\cot(c+dx)(-2(a^2B-b^2B+abC)-2a^2C\tan(c+dx)+2b(bB-aC)\tan^2(c+dx))}{a+b\tan(c+dx)}\,dx}{2a^2}$$

$$= \frac{(bB-aC)x}{a^2+b^2} + \frac{(bB-aC)\cot(c+dx)}{a^2d} - \frac{B\cot^2(c+dx)}{2ad} - \frac{(b^3(bB-aC))\int \frac{b-a\tan(c+dx)}{a+b\tan(c+dx)}\,dx}{a^3(a^2+b^2)} - \frac{(a^2B-b^2B+abC)\int \cot(c+dx)\,dx}{a^3}$$

$$= \frac{(bB-aC)x}{a^2+b^2} + \frac{(bB-aC)\cot(c+dx)}{a^2d} - \frac{B\cot^2(c+dx)}{2ad} - \frac{(a^2B-b^2B+abC)\log(\sin(c+dx))}{a^3d} - \frac{b^3(bB-aC)\log(a\cos(c+dx)+b\sin(c+dx))}{a^3(a^2+b^2)d}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 1.52 (sec) , antiderivative size = 163, normalized size of antiderivative = 1.19

$$\int \frac{\cot^4(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx$$
$$=\frac{\frac{2(bB-aC)\cot(c+dx)}{a^2}-\frac{B\cot^2(c+dx)}{a}+\frac{(B+iC)\log(i-\tan(c+dx))}{a+ib}-\frac{2(a^2B-b^2B+abC)\log(\tan(c+dx))}{a^3}+\frac{(B-iC)\log(i+\tan(c+dx))}{a-ib}}{2d}$$

```
[In] Integrate[(Cot[c + d*x]^4*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c
 + d*x]),x]
```

```
[Out] ((2*(b*B - a*C)*Cot[c + d*x])/a^2 - (B*Cot[c + d*x]^2)/a + ((B + I*C)*Log[I
 - Tan[c + d*x]])/(a + I*b) - (2*(a^2*B - b^2*B + a*b*C)*Log[Tan[c + d*x]])
/a^3 + ((B - I*C)*Log[I + Tan[c + d*x]])/(a - I*b) + (2*b^3*(-(b*B) + a*C)*
Log[a + b*Tan[c + d*x]])/(a^3*(a^2 + b^2)))/(2*d)
```

Maple [A] (verified)

Time = 0.35 (sec) , antiderivative size = 152, normalized size of antiderivative = 1.11

method	result
derivativedivides	$\frac{-\frac{B}{2a\tan(dx+c)^2}-\frac{-Bb+Ca}{a^2\tan(dx+c)}+\frac{\left(-Ba^2+Bb^2-Cab\right)\ln(\tan(dx+c))}{a^3}+\frac{\frac{(Ba+Cb)\ln\left(1+\tan(dx+c)^2\right)}{2}+(Bb-Ca)\arctan(\tan(dx+c)}{a^2+b^2}}{d}$
default	$\frac{-\frac{B}{2a\tan(dx+c)^2}-\frac{-Bb+Ca}{a^2\tan(dx+c)}+\frac{\left(-Ba^2+Bb^2-Cab\right)\ln(\tan(dx+c))}{a^3}+\frac{\frac{(Ba+Cb)\ln\left(1+\tan(dx+c)^2\right)}{2}+(Bb-Ca)\arctan(\tan(dx+c)}{a^2+b^2}}{d}$
parallelrisch	$\frac{(-2Bb^4+2Cab^3)\ln(a+b\tan(dx+c))+(Ba^4+Ca^3b)\ln\left(\sec(dx+c)^2\right)+(-2Ba^4+2Bb^4-2Ca^3b-2Cab^3)\ln(\tan(dx+}{2(a^2+b^2)a^3d}$
norman	$\frac{\frac{(Bb-Ca)\tan(dx+c)^2}{a^2d}+\frac{(Bb-Ca)x\tan(dx+c)^3}{a^2+b^2}-\frac{B\tan(dx+c)}{2ad}}{\tan(dx+c)^3}+\frac{(Ba+Cb)\ln\left(1+\tan(dx+c)^2\right)}{2d(a^2+b^2)}-\frac{(Ba^2-Bb^2+Cab)\ln(t}{a^3d}$
risch	$-\frac{2iBb^2c}{a^3d}+\frac{xC}{ib-a}-\frac{2ib^3Cx}{(a^2+b^2)a^2}+\frac{ixB}{ib-a}-\frac{2iBb^2x}{a^3}+\frac{2iCbc}{a^2d}+\frac{2iBx}{a}+\frac{2ib^4Bx}{(a^2+b^2)a^3}+\frac{2iBc}{ad}-\frac{2i(iBa\,e^{2i(d}}{}$

```
[In] int(cot(d*x+c)^4*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x,method=_R
ETURNVERBOSE)
```

```
[Out] 1/d*(-1/2/a*B/tan(d*x+c)^2-(-B*b+C*a)/a^2/tan(d*x+c)+1/a^3*(-B*a^2+B*b^2-C*
a*b)*ln(tan(d*x+c))+1/(a^2+b^2)*(1/2*(B*a+C*b)*ln(1+tan(d*x+c)^2)+(B*b-C*a)
*arctan(tan(d*x+c)))-(B*b-C*a)*b^3/(a^2+b^2)/a^3*ln(a+b*tan(d*x+c)))
```

Fricas [A] (verification not implemented)

none

Time = 0.28 (sec) , antiderivative size = 234, normalized size of antiderivative = 1.71

$$\int \frac{\cot^4(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{a+b\tan(c+dx)}\,dx = -\frac{Ba^4+Ba^2b^2+(Ba^4+Ca^3b+Cab^3-Bb^4)\log\left(\frac{\tan(dx+c)^2}{\tan(dx+c)^2+1}\right)\tan(dx+c)^2-(Cab^3-Bb^4)\log\left(\frac{b^2\tan(d}{}\right.}{}$$

```
[In] integrate(cot(d*x+c)^4*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, al
gorithm="fricas")
```

```
[Out] -1/2*(B*a^4 + B*a^2*b^2 + (B*a^4 + C*a^3*b + C*a*b^3 - B*b^4)*log(tan(d*x +
 c)^2/(tan(d*x + c)^2 + 1))*tan(d*x + c)^2 - (C*a*b^3 - B*b^4)*log((b^2*tan
(d*x + c)^2 + 2*a*b*tan(d*x + c) + a^2)/(tan(d*x + c)^2 + 1))*tan(d*x + c)^
2 + (B*a^4 + B*a^2*b^2 + 2*(C*a^4 - B*a^3*b)*d*x)*tan(d*x + c)^2 + 2*(C*a^4
 - B*a^3*b + C*a^2*b^2 - B*a*b^3)*tan(d*x + c))/((a^5 + a^3*b^2)*d*tan(d*x
+ c)^2)
```

Sympy [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 10.29 (sec) , antiderivative size = 2596, normalized size of antiderivative = 18.95

$$\int \frac{\cot^4(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{a+b\tan(c+dx)}\,dx = \text{Too large to display}$$

```
[In] integrate(cot(d*x+c)**4*(B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c)),x)
```

```
[Out] Piecewise((nan, Eq(a, 0) & Eq(b, 0) & Eq(c, 0) & Eq(d, 0)), ((B*log(tan(c +
 d*x)**2 + 1)/(2*d) - B*log(tan(c + d*x))/d - B/(2*d*tan(c + d*x)**2) - C*x
 - C/(d*tan(c + d*x)))/a, Eq(b, 0)), ((B*x + B/(d*tan(c + d*x)) - B/(3*d*ta
n(c + d*x)**3) + C*log(tan(c + d*x)**2 + 1)/(2*d) - C*log(tan(c + d*x))/d -
 C/(2*d*tan(c + d*x)**2))/b, Eq(a, 0)), (-3*I*B*d*x*tan(c + d*x)**3/(2*a*d*
tan(c + d*x)**3 + 2*I*a*d*tan(c + d*x)**2) + 3*B*d*x*tan(c + d*x)**2/(2*a*d
*tan(c + d*x)**3 + 2*I*a*d*tan(c + d*x)**2) + 2*B*log(tan(c + d*x)**2 + 1)*
tan(c + d*x)**3/(2*a*d*tan(c + d*x)**3 + 2*I*a*d*tan(c + d*x)**2) + 2*I*B*l
og(tan(c + d*x)**2 + 1)*tan(c + d*x)**2/(2*a*d*tan(c + d*x)**3 + 2*I*a*d*ta
n(c + d*x)**2) - 4*B*log(tan(c + d*x))*tan(c + d*x)**3/(2*a*d*tan(c + d*x)*
*3 + 2*I*a*d*tan(c + d*x)**2) - 4*I*B*log(tan(c + d*x))*tan(c + d*x)**2/(2*
a*d*tan(c + d*x)**3 + 2*I*a*d*tan(c + d*x)**2) - 3*I*B*tan(c + d*x)**2/(2*a
*d*tan(c + d*x)**3 + 2*I*a*d*tan(c + d*x)**2) + B*tan(c + d*x)/(2*a*d*tan(c
```

```
 + d*x)**3 + 2*I*a*d*tan(c + d*x)**2) - I*B/(2*a*d*tan(c + d*x)**3 + 2*I*a*
d*tan(c + d*x)**2) - 3*C*d*x*tan(c + d*x)**3/(2*a*d*tan(c + d*x)**3 + 2*I*a
*d*tan(c + d*x)**2) - 3*I*C*d*x*tan(c + d*x)**2/(2*a*d*tan(c + d*x)**3 + 2*
I*a*d*tan(c + d*x)**2) - I*C*log(tan(c + d*x)**2 + 1)*tan(c + d*x)**3/(2*a*
d*tan(c + d*x)**3 + 2*I*a*d*tan(c + d*x)**2) + C*log(tan(c + d*x)**2 + 1)*t
an(c + d*x)**2/(2*a*d*tan(c + d*x)**3 + 2*I*a*d*tan(c + d*x)**2) + 2*I*C*lo
g(tan(c + d*x))*tan(c + d*x)**3/(2*a*d*tan(c + d*x)**3 + 2*I*a*d*tan(c + d*
x)**2) - 2*C*log(tan(c + d*x))*tan(c + d*x)**2/(2*a*d*tan(c + d*x)**3 + 2*I
*a*d*tan(c + d*x)**2) - 3*C*tan(c + d*x)**2/(2*a*d*tan(c + d*x)**3 + 2*I*a*
d*tan(c + d*x)**2) - 2*I*C*tan(c + d*x)/(2*a*d*tan(c + d*x)**3 + 2*I*a*d*ta
n(c + d*x)**2), Eq(b, -I*a)), (3*I*B*d*x*tan(c + d*x)**3/(2*a*d*tan(c + d*x
)**3 - 2*I*a*d*tan(c + d*x)**2) + 3*B*d*x*tan(c + d*x)**2/(2*a*d*tan(c + d*
x)**3 - 2*I*a*d*tan(c + d*x)**2) + 2*B*log(tan(c + d*x)**2 + 1)*tan(c + d*x
)**3/(2*a*d*tan(c + d*x)**3 - 2*I*a*d*tan(c + d*x)**2) - 2*I*B*log(tan(c +
d*x)**2 + 1)*tan(c + d*x)**2/(2*a*d*tan(c + d*x)**3 - 2*I*a*d*tan(c + d*x)*
*2) - 4*B*log(tan(c + d*x))*tan(c + d*x)**3/(2*a*d*tan(c + d*x)**3 - 2*I*a*
d*tan(c + d*x)**2) + 4*I*B*log(tan(c + d*x))*tan(c + d*x)**2/(2*a*d*tan(c +
 d*x)**3 - 2*I*a*d*tan(c + d*x)**2) + 3*I*B*tan(c + d*x)**2/(2*a*d*tan(c +
d*x)**3 - 2*I*a*d*tan(c + d*x)**2) + B*tan(c + d*x)/(2*a*d*tan(c + d*x)**3
- 2*I*a*d*tan(c + d*x)**2) + I*B/(2*a*d*tan(c + d*x)**3 - 2*I*a*d*tan(c + d
*x)**2) - 3*C*d*x*tan(c + d*x)**3/(2*a*d*tan(c + d*x)**3 - 2*I*a*d*tan(c +
d*x)**2) + 3*I*C*d*x*tan(c + d*x)**2/(2*a*d*tan(c + d*x)**3 - 2*I*a*d*tan(c
 + d*x)**2) + I*C*log(tan(c + d*x)**2 + 1)*tan(c + d*x)**3/(2*a*d*tan(c + d
*x)**3 - 2*I*a*d*tan(c + d*x)**2) + C*log(tan(c + d*x)**2 + 1)*tan(c + d*x)
**2/(2*a*d*tan(c + d*x)**3 - 2*I*a*d*tan(c + d*x)**2) - 2*I*C*log(tan(c + d
*x))*tan(c + d*x)**3/(2*a*d*tan(c + d*x)**3 - 2*I*a*d*tan(c + d*x)**2) - 2*
C*log(tan(c + d*x))*tan(c + d*x)**2/(2*a*d*tan(c + d*x)**3 - 2*I*a*d*tan(c
+ d*x)**2) - 3*C*tan(c + d*x)**2/(2*a*d*tan(c + d*x)**3 - 2*I*a*d*tan(c + d
*x)**2) + 2*I*C*tan(c + d*x)/(2*a*d*tan(c + d*x)**3 - 2*I*a*d*tan(c + d*x)*
*2), Eq(b, I*a)), (nan, Eq(c, -d*x)), (x*(B*tan(c) + C*tan(c)**2)*cot(c)**4
/(a + b*tan(c)), Eq(d, 0)), (B*a**4*log(tan(c + d*x)**2 + 1)*tan(c + d*x)**
2/(2*a**5*d*tan(c + d*x)**2 + 2*a**3*b**2*d*tan(c + d*x)**2) - 2*B*a**4*log
(tan(c + d*x))*tan(c + d*x)**2/(2*a**5*d*tan(c + d*x)**2 + 2*a**3*b**2*d*ta
n(c + d*x)**2) - B*a**4/(2*a**5*d*tan(c + d*x)**2 + 2*a**3*b**2*d*tan(c + d
*x)**2) + 2*B*a**3*b*d*x*tan(c + d*x)**2/(2*a**5*d*tan(c + d*x)**2 + 2*a**3
*b**2*d*tan(c + d*x)**2) + 2*B*a**3*b*tan(c + d*x)/(2*a**5*d*tan(c + d*x)**
2 + 2*a**3*b**2*d*tan(c + d*x)**2) - B*a**2*b**2/(2*a**5*d*tan(c + d*x)**2
+ 2*a**3*b**2*d*tan(c + d*x)**2) + 2*B*a*b**3*tan(c + d*x)/(2*a**5*d*tan(c
+ d*x)**2 + 2*a**3*b**2*d*tan(c + d*x)**2) - 2*B*b**4*log(a/b + tan(c + d*x
))*tan(c + d*x)**2/(2*a**5*d*tan(c + d*x)**2 + 2*a**3*b**2*d*tan(c + d*x)**
2) + 2*B*b**4*log(tan(c + d*x))*tan(c + d*x)**2/(2*a**5*d*tan(c + d*x)**2 +
 2*a**3*b**2*d*tan(c + d*x)**2) - 2*C*a**4*d*x*tan(c + d*x)**2/(2*a**5*d*ta
n(c + d*x)**2 + 2*a**3*b**2*d*tan(c + d*x)**2) - 2*C*a**4*tan(c + d*x)/(2*a
**5*d*tan(c + d*x)**2 + 2*a**3*b**2*d*tan(c + d*x)**2) + C*a**3*b*log(tan(c
 + d*x)**2 + 1)*tan(c + d*x)**2/(2*a**5*d*tan(c + d*x)**2 + 2*a**3*b**2*d*t
```

```
an(c + d*x)**2) - 2*C*a**3*b*log(tan(c + d*x))*tan(c + d*x)**2/(2*a**5*d*ta
n(c + d*x)**2 + 2*a**3*b**2*d*tan(c + d*x)**2) - 2*C*a**2*b**2*tan(c + d*x)
/(2*a**5*d*tan(c + d*x)**2 + 2*a**3*b**2*d*tan(c + d*x)**2) + 2*C*a*b**3*lo
g(a/b + tan(c + d*x))*tan(c + d*x)**2/(2*a**5*d*tan(c + d*x)**2 + 2*a**3*b*
*2*d*tan(c + d*x)**2) - 2*C*a*b**3*log(tan(c + d*x))*tan(c + d*x)**2/(2*a**
5*d*tan(c + d*x)**2 + 2*a**3*b**2*d*tan(c + d*x)**2), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.31 (sec) , antiderivative size = 158, normalized size of antiderivative = 1.15

$$\int \frac{\cot^4(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx = -\frac{\frac{2(Ca-Bb)(dx+c)}{a^2+b^2} - \frac{2\left(Cab^3-Bb^4\right)\log(b\tan(dx+c)+a)}{a^5+a^3b^2} - \frac{(Ba+Cb)\log\left(\tan(dx+c)^2+1\right)}{a^2+b^2} + \frac{2\left(Ba^2+Cab-Bb^2\right)\log(\tan(dx+c))}{a^3} + \frac{Ba+\cdots}{}}{2\,d}$$

```
[In] integrate(cot(d*x+c)^4*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, al
gorithm="maxima")
```

```
[Out] -1/2*(2*(C*a - B*b)*(d*x + c)/(a^2 + b^2) - 2*(C*a*b^3 - B*b^4)*log(b*tan(d
*x + c) + a)/(a^5 + a^3*b^2) - (B*a + C*b)*log(tan(d*x + c)^2 + 1)/(a^2 + b
^2) + 2*(B*a^2 + C*a*b - B*b^2)*log(tan(d*x + c))/a^3 + (B*a + 2*(C*a - B*b
)*tan(d*x + c))/(a^2*tan(d*x + c)^2))/d
```

Giac [A] (verification not implemented)

none

Time = 1.36 (sec) , antiderivative size = 214, normalized size of antiderivative = 1.56

$$\int \frac{\cot^4(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx = -\frac{\frac{2(Ca-Bb)(dx+c)}{a^2+b^2} - \frac{(Ba+Cb)\log\left(\tan(dx+c)^2+1\right)}{a^2+b^2} - \frac{2\left(Cab^4-Bb^5\right)\log(|b\tan(dx+c)+a|)}{a^5b+a^3b^3} + \frac{2\left(Ba^2+Cab-Bb^2\right)\log(|\tan(dx+c)|)}{a^3} - \frac{3\cdots}{}}{2\,d}$$

```
[In] integrate(cot(d*x+c)^4*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, al
gorithm="giac")
```

```
[Out] -1/2*(2*(C*a - B*b)*(d*x + c)/(a^2 + b^2) - (B*a + C*b)*log(tan(d*x + c)^2
+ 1)/(a^2 + b^2) - 2*(C*a*b^4 - B*b^5)*log(abs(b*tan(d*x + c) + a))/(a^5*b
+ a^3*b^3) + 2*(B*a^2 + C*a*b - B*b^2)*log(abs(tan(d*x + c)))/a^3 - (3*B*a^
2*tan(d*x + c)^2 + 3*C*a*b*tan(d*x + c)^2 - 3*B*b^2*tan(d*x + c)^2 - 2*C*a^
2*tan(d*x + c) + 2*B*a*b*tan(d*x + c) - B*a^2)/(a^3*tan(d*x + c)^2))/d
```

Mupad [B] (verification not implemented)

Time = 10.24 (sec) , antiderivative size = 175, normalized size of antiderivative = 1.28

$$\int \frac{\cot^4(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{a+b\tan(c+dx)}\,dx$$
$$= -\frac{\cot(c+d\,x)^2\,\left(\frac{B}{2\,a}-\frac{\tan(c+d\,x)\,(B\,b-C\,a)}{a^2}\right)}{d}+\frac{\ln\left(\tan(c+d\,x)-\mathrm{i}\right)\,(-C+B\,1\mathrm{i})}{2\,d\,(-b+a\,1\mathrm{i})}$$
$$-\frac{\ln\left(\tan(c+d\,x)\right)\,(B\,a^2+C\,a\,b-B\,b^2)}{a^3\,d}$$
$$-\frac{\ln\left(a+b\tan(c+d\,x)\right)\,(B\,b^4-C\,a\,b^3)}{d\,(a^5+a^3\,b^2)}+\frac{\ln\left(\tan(c+d\,x)+1\mathrm{i}\right)\,(B-C\,1\mathrm{i})}{2\,d\,(a-b\,1\mathrm{i})}$$

```
[In] int((cot(c + d*x)^4*(B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c + d*x
)),x)
```

```
[Out] (log(tan(c + d*x) - 1i)*(B*1i - C))/(2*d*(a*1i - b)) - (cot(c + d*x)^2*(B/(
2*a) - (tan(c + d*x)*(B*b - C*a))/a^2))/d - (log(tan(c + d*x))*(B*a^2 - B*b
^2 + C*a*b))/(a^3*d) - (log(a + b*tan(c + d*x))*(B*b^4 - C*a*b^3))/(d*(a^5
+ a^3*b^2)) + (log(tan(c + d*x) + 1i)*(B - C*1i))/(2*d*(a - b*1i))
```

3.32 $\int \frac{\tan^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx$

Optimal result 272
Rubi [A] (verified) 272
Mathematica [C] (verified) 275
Maple [A] (verified) 276
Fricas [B] (verification not implemented) 276
Sympy [C] (verification not implemented) 277
Maxima [A] (verification not implemented) 279
Giac [A] (verification not implemented) 280
Mupad [B] (verification not implemented) 280

Optimal result

Integrand size = 40, antiderivative size = 208

$$\begin{aligned}\int \frac{\tan^2(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^2}\,dx &= -\frac{(2abB-a^2C+b^2C)\,x}{(a^2+b^2)^2}+\frac{(a^2B-b^2B+2abC)\log(\cos(c+dx))}{(a^2+b^2)^2\,d}\\ &\quad+\frac{a^2(a^2bB+3b^3B-2a^3C-4ab^2C)\log(a+b\tan(c+dx))}{b^3\,(a^2+b^2)^2\,d}\\ &\quad-\frac{(abB-2a^2C-b^2C)\tan(c+dx)}{b^2\,(a^2+b^2)\,d}+\frac{a(bB-aC)\tan^2(c+dx)}{b\,(a^2+b^2)\,d(a+b\tan(c+dx))}\end{aligned}$$

```
[Out] -(2*B*a*b-C*a^2+C*b^2)*x/(a^2+b^2)^2+(B*a^2-B*b^2+2*C*a*b)*ln(cos(d*x+c))/(
a^2+b^2)^2/d+a^2*(B*a^2*b+3*B*b^3-2*C*a^3-4*C*a*b^2)*ln(a+b*tan(d*x+c))/b^3
/(a^2+b^2)^2/d-(B*a*b-2*C*a^2-C*b^2)*tan(d*x+c)/b^2/(a^2+b^2)/d+a*(B*b-C*a)
*tan(d*x+c)^2/b/(a^2+b^2)/d/(a+b*tan(d*x+c))
```

Rubi [A] (verified)

Time = 0.64 (sec) , antiderivative size = 208, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 7, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.175, Rules used

= {3713, 3686, 3728, 3707, 3698, 31, 3556}

$$\int \frac{\tan^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx$$
$$= \frac{a(bB-aC)\tan^2(c+dx)}{bd\left(a^2+b^2\right)(a+b\tan(c+dx))} - \frac{\left(-2a^2C+abB-b^2C\right)\tan(c+dx)}{b^2d\left(a^2+b^2\right)}$$
$$+ \frac{\left(a^2B+2abC-b^2B\right)\log(\cos(c+dx))}{d\left(a^2+b^2\right)^2} - \frac{x\left(a^2(-C)+2abB+b^2C\right)}{\left(a^2+b^2\right)^2}$$
$$+ \frac{a^2\left(-2a^3C+a^2bB-4ab^2C+3b^3B\right)\log(a+b\tan(c+dx))}{b^3d\left(a^2+b^2\right)^2}$$

```
[In] Int[(Tan[c + d*x]^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c + d*x
])^2,x]
```

```
[Out] -(((2*a*b*B - a^2*C + b^2*C)*x)/(a^2 + b^2)^2) + ((a^2*B - b^2*B + 2*a*b*C)
*Log[Cos[c + d*x]])/((a^2 + b^2)^2*d) + (a^2*(a^2*b*B + 3*b^3*B - 2*a^3*C -
 4*a*b^2*C)*Log[a + b*Tan[c + d*x]])/(b^3*(a^2 + b^2)^2*d) - ((a*b*B - 2*a^
2*C - b^2*C)*Tan[c + d*x])/(b^2*(a^2 + b^2)*d) + (a*(b*B - a*C)*Tan[c + d*x
]^2)/(b*(a^2 + b^2)*d*(a + b*Tan[c + d*x]))
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3686

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Si
mp[(b*c - a*d)*(B*c - A*d)*(a + b*Tan[e + f*x])^(m - 1)*((c + d*Tan[e + f*x
])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dist[1/(d*(n + 1)*(c^2 + d^2)),
 Int[(a + b*Tan[e + f*x])^(m - 2)*(c + d*Tan[e + f*x])^(n + 1)*Simp[a*A*d*(
b*d*(m - 1) - a*c*(n + 1)) + (b*B*c - (A*b + a*B)*d)*(b*c*(m - 1) + a*d*(n
+ 1)) - d*((a*A - b*B)*(b*c - a*d) + (A*b + a*B)*(a*c + b*d))*(n + 1)*Tan[e
 + f*x] - b*(d*(A*b*c + a*B*c - a*A*d)*(m + n) - b*B*(c^2*(m - 1) - d^2*(n
+ 1)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] &&
NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 1] &&
 LtQ[n, -1] && (IntegerQ[m] || IntegersQ[2*m, 2*n])
```

Rule 3698

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3707

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[(a*A + b*B - a
*C)*(x/(a^2 + b^2)), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$
\begin{aligned}
\text{integral} &= \int \frac{\tan^3(c+dx)(B+C\tan(c+dx))}{(a+b\tan(c+dx))^2}\,dx \\
&= \frac{a(bB-aC)\tan^2(c+dx)}{b\left(a^2+b^2\right)d(a+b\tan(c+dx))} \\
&\quad + \frac{\int \frac{\tan(c+dx)\left(-2a(bB-aC)+b(bB-aC)\tan(c+dx)-\left(abB-2a^2C-b^2C\right)\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx}{b\left(a^2+b^2\right)}
\end{aligned}
$$

$$= -\frac{(abB - 2a^2C - b^2C)\tan(c+dx)}{b^2(a^2+b^2)d} + \frac{a(bB - aC)\tan^2(c+dx)}{b(a^2+b^2)d(a+b\tan(c+dx))} + \frac{\int \frac{a(abB-2a^2C-b^2C) - b^2(aB+bC)\tan(c+dx) + (a^2+b^2)(bB-2aC)\tan^2(c+dx)}{a+b\tan(c+dx)}dx}{b^2(a^2+b^2)}$$

$$= -\frac{(2abB - a^2C + b^2C)x}{(a^2+b^2)^2} - \frac{(abB - 2a^2C - b^2C)\tan(c+dx)}{b^2(a^2+b^2)d} + \frac{a(bB - aC)\tan^2(c+dx)}{b(a^2+b^2)d(a+b\tan(c+dx))} - \frac{(a^2B - b^2B + 2abC)\int\tan(c+dx)\,dx}{(a^2+b^2)^2} + \frac{(a^2(a^2bB + 3b^3B - 2a^3C - 4ab^2C))\int\frac{1+\tan^2(c+dx)}{a+b\tan(c+dx)}dx}{b^2(a^2+b^2)^2}$$

$$= -\frac{(2abB - a^2C + b^2C)x}{(a^2+b^2)^2} + \frac{(a^2B - b^2B + 2abC)\log(\cos(c+dx))}{(a^2+b^2)^2 d} - \frac{(abB - 2a^2C - b^2C)\tan(c+dx)}{b^2(a^2+b^2)d} + \frac{a(bB - aC)\tan^2(c+dx)}{b(a^2+b^2)d(a+b\tan(c+dx))} + \frac{(a^2(a^2bB + 3b^3B - 2a^3C - 4ab^2C))\,\text{Subst}\left(\int\frac{1}{a+x}dx, x, b\tan(c+dx)\right)}{b^3(a^2+b^2)^2 d}$$

$$= -\frac{(2abB - a^2C + b^2C)x}{(a^2+b^2)^2} + \frac{(a^2B - b^2B + 2abC)\log(\cos(c+dx))}{(a^2+b^2)^2 d} + \frac{a^2(a^2bB + 3b^3B - 2a^3C - 4ab^2C)\log(a+b\tan(c+dx))}{b^3(a^2+b^2)^2 d} - \frac{(abB - 2a^2C - b^2C)\tan(c+dx)}{b^2(a^2+b^2)d} + \frac{a(bB - aC)\tan^2(c+dx)}{b(a^2+b^2)d(a+b\tan(c+dx))}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 6.09 (sec) , antiderivative size = 193, normalized size of antiderivative = 0.93

$$\int \frac{\tan^2(c+dx)\left(B\tan(c+dx) + C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}dx = -\frac{\frac{(B+iC)\log(i-\tan(c+dx))}{(a+ib)^2} + \frac{(B-iC)\log(i+\tan(c+dx))}{(a-ib)^2} + \frac{2a^2(-a^2bB-3b^3B+2a^3C+4ab^2C)\log(a+b\tan(c+dx))}{b^3(a^2+b^2)^2} + \frac{2a^2(-abB+2a^2C\cdots}{b^3(a^2+b^2)(a+b\tan\cdots}}{2d}$$

```
[In] Integrate[(Tan[c + d*x]^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c
 + d*x])^2,x]
```

```
[Out] -1/2*(((B + I*C)*Log[I - Tan[c + d*x]])/(a + I*b)^2 + ((B - I*C)*Log[I + Ta
n[c + d*x]])/(a - I*b)^2 + (2*a^2*(-(a^2*b*B) - 3*b^3*B + 2*a^3*C + 4*a*b^2
*C)*Log[a + b*Tan[c + d*x]])/(b^3*(a^2 + b^2)^2) + (2*a^2*(-(a*b*B) + 2*a^2
*C + b^2*C))/(b^3*(a^2 + b^2)*(a + b*Tan[c + d*x])) - (2*C*Tan[c + d*x]^2)/
(b*(a + b*Tan[c + d*x])))/d
```

Maple [A] (verified)

Time = 0.19 (sec) , antiderivative size = 172, normalized size of antiderivative = 0.83

method	result
derivativedivides	$\frac{\frac{\tan(dx+c)C}{b^2}+\frac{\frac{(-B a^2+B b^2-2Cab)\ln(1+\tan(dx+c)^2)}{2}+(-2Bab+C a^2-C b^2)\arctan(\tan(dx+c))}{(a^2+b^2)^2}+\frac{a^2(B a^2b+3B b^3-2C a^3-4Ca b}{b^3(a^2+b^2)}}{d}$
default	$\frac{\frac{\tan(dx+c)C}{b^2}+\frac{\frac{(-B a^2+B b^2-2Cab)\ln(1+\tan(dx+c)^2)}{2}+(-2Bab+C a^2-C b^2)\arctan(\tan(dx+c))}{(a^2+b^2)^2}+\frac{a^2(B a^2b+3B b^3-2C a^3-4Ca b}{b^3(a^2+b^2)}}{d}$
norman	$\frac{\frac{C\tan(dx+c)^2}{bd}+\frac{(B a^2b-2C a^3-Ca b^2)a}{d b^3(a^2+b^2)}-\frac{a(2Bab-C a^2+C b^2)x}{a^4+2a^2b^2+b^4}-\frac{b(2Bab-C a^2+C b^2)x\tan(dx+c)}{a^4+2a^2b^2+b^4}}{a+b\tan(dx+c)}+\frac{a^2(B a^2b+3B b^3-2C a}{(a^4+2}$
parallelrisch	$-\frac{4C a^6+2C a^2b^4-2B a^3b^3+6C a^4b^2-2B a^5b+B\ln(1+\tan(dx+c)^2)\tan(dx+c)a^2b^4-2B\ln(a+b\tan(dx+c))\tan(dx+c)}{}$
risch	$\frac{2i(-B a^3b\,e^{2i(dx+c)}+2C a^4e^{2i(dx+c)}-C b^4e^{2i(dx+c)}-2iC a^3b\,e^{2i(dx+c)}-2iCa b^3e^{2i(dx+c)}-B a^3b+2C a^4+2C a^2b^2+C b^4}{(e^{2i(dx+c)}+1)(ib+a)(-ib+a)^2(-ib\,e^{2i(dx+c)}+a\,e^{2i(dx+c)}+ib+a)b^2d}$

```
[In] int(tan(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x,method=
_RETURNVERBOSE)
```

```
[Out] 1/d*(tan(d*x+c)*C/b^2+1/(a^2+b^2)^2*(1/2*(-B*a^2+B*b^2-2*C*a*b)*ln(1+tan(d*
x+c)^2)+(-2*B*a*b+C*a^2-C*b^2)*arctan(tan(d*x+c)))+1/b^3*a^2*(B*a^2*b+3*B*b
^3-2*C*a^3-4*C*a*b^2)/(a^2+b^2)^2*ln(a+b*tan(d*x+c))+1/b^3*a^3*(B*b-C*a)/(a
^2+b^2)/(a+b*tan(d*x+c)))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 434 vs. 2(209) = 418.

Time = 0.31 (sec) , antiderivative size = 434, normalized size of antiderivative = 2.09

$$\int \frac{\tan^2(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^2}\,dx =$$

$$-\frac{2\,Ca^4b^2-2\,Ba^3b^3-2\,(Ca^3b^3-2\,Ba^2b^4-Cab^5)dx-2\,(Ca^4b^2+2\,Ca^2b^4+Cb^6)\tan\left(dx+c\right)^2+(2\,Ca^6}{}$$

```
[In] integrate(tan(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x,
algorithm="fricas")
```

```
[Out] -1/2*(2*C*a^4*b^2 - 2*B*a^3*b^3 - 2*(C*a^3*b^3 - 2*B*a^2*b^4 - C*a*b^5)*d*x
 - 2*(C*a^4*b^2 + 2*C*a^2*b^4 + C*b^6)*tan(d*x + c)^2 + (2*C*a^6 - B*a^5*b
+ 4*C*a^4*b^2 - 3*B*a^3*b^3 + (2*C*a^5*b - B*a^4*b^2 + 4*C*a^3*b^3 - 3*B*a^
2*b^4)*tan(d*x + c))*log((b^2*tan(d*x + c)^2 + 2*a*b*tan(d*x + c) + a^2)/(t
```

```
an(d*x + c)^2 + 1)) - (2*C*a^6 - B*a^5*b + 4*C*a^4*b^2 - 2*B*a^3*b^3 + 2*C*
a^2*b^4 - B*a*b^5 + (2*C*a^5*b - B*a^4*b^2 + 4*C*a^3*b^3 - 2*B*a^2*b^4 + 2*
C*a*b^5 - B*b^6)*tan(d*x + c))*log(1/(tan(d*x + c)^2 + 1)) - 2*(2*C*a^5*b -
 B*a^4*b^2 + 2*C*a^3*b^3 + C*a*b^5 + (C*a^2*b^4 - 2*B*a*b^5 - C*b^6)*d*x)*t
an(d*x + c))/((a^4*b^4 + 2*a^2*b^6 + b^8)*d*tan(d*x + c) + (a^5*b^3 + 2*a^3
*b^5 + a*b^7)*d)
```

Sympy [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 1.41 (sec) , antiderivative size = 4541, normalized size of antiderivative = 21.83

$$\int \frac{\tan^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx = \text{Too large to display}$$

```
[In] integrate(tan(d*x+c)**2*(B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c))**2,
x)
```

```
[Out] Piecewise((zoo*x*(B*tan(c) + C*tan(c)**2), Eq(a, 0) & Eq(b, 0) & Eq(d, 0)),
 ((-B*log(tan(c + d*x)**2 + 1)/(2*d) + B*tan(c + d*x)**2/(2*d) + C*x + C*ta
n(c + d*x)**3/(3*d) - C*tan(c + d*x)/d)/a**2, Eq(b, 0)), (3*I*B*d*x*tan(c +
 d*x)**2/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) +
6*B*d*x*tan(c + d*x)/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) -
4*b**2*d) - 3*I*B*d*x/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) -
 4*b**2*d) + 2*B*log(tan(c + d*x)**2 + 1)*tan(c + d*x)**2/(4*b**2*d*tan(c +
 d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) - 4*I*B*log(tan(c + d*x)**2
+ 1)*tan(c + d*x)/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b
**2*d) - 2*B*log(tan(c + d*x)**2 + 1)/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*
d*tan(c + d*x) - 4*b**2*d) - 5*I*B*tan(c + d*x)/(4*b**2*d*tan(c + d*x)**2 -
 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) - 4*B/(4*b**2*d*tan(c + d*x)**2 - 8*I*
b**2*d*tan(c + d*x) - 4*b**2*d) - 9*C*d*x*tan(c + d*x)**2/(4*b**2*d*tan(c +
 d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + 18*I*C*d*x*tan(c + d*x)/(4
*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + 9*C*d*x/(4*
b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + 4*I*C*log(ta
n(c + d*x)**2 + 1)*tan(c + d*x)**2/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*t
an(c + d*x) - 4*b**2*d) + 8*C*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(4*b**2
*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) - 4*I*C*log(tan(c
+ d*x)**2 + 1)/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2
*d) + 4*C*tan(c + d*x)**3/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*
x) - 4*b**2*d) + 19*C*tan(c + d*x)/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*t
an(c + d*x) - 4*b**2*d) - 14*I*C/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan
(c + d*x) - 4*b**2*d), Eq(a, -I*b)), (-3*I*B*d*x*tan(c + d*x)**2/(4*b**2*d*
tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + 6*B*d*x*tan(c + d*x
)/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + 3*I*B*d
*x/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + 2*B*lo
```

g(tan(c + d*x)**2 + 1)*tan(c + d*x)**2/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2
*d*tan(c + d*x) - 4*b**2*d) + 4*I*B*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(
4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) - 2*B*log(ta
n(c + d*x)**2 + 1)/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*
b**2*d) + 5*I*B*tan(c + d*x)/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c +
 d*x) - 4*b**2*d) - 4*B/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x)
 - 4*b**2*d) - 9*C*d*x*tan(c + d*x)**2/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2
*d*tan(c + d*x) - 4*b**2*d) - 18*I*C*d*x*tan(c + d*x)/(4*b**2*d*tan(c + d*x
)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + 9*C*d*x/(4*b**2*d*tan(c + d*x)
2 + 8*I*b2*d*tan(c + d*x) - 4*b**2*d) - 4*I*C*log(tan(c + d*x)**2 + 1)*
tan(c + d*x)**2/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**
2*d) + 8*C*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(4*b**2*d*tan(c + d*x)**2
+ 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + 4*I*C*log(tan(c + d*x)**2 + 1)/(4*b
2*d*tan(c + d*x)2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + 4*C*tan(c + d
*x)**3/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + 19
*C*tan(c + d*x)/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**
2*d) + 14*I*C/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*
d), Eq(a, I*b)), (x*(B*tan(c) + C*tan(c)**2)*tan(c)**2/(a + b*tan(c))**2, E
q(d, 0)), (2*B*a**5*b*log(a/b + tan(c + d*x))/(2*a**5*b**3*d + 2*a**4*b**4*
d*tan(c + d*x) + 4*a**3*b**5*d + 4*a**2*b**6*d*tan(c + d*x) + 2*a*b**7*d +
2*b**8*d*tan(c + d*x)) + 2*B*a**5*b/(2*a**5*b**3*d + 2*a**4*b**4*d*tan(c +
d*x) + 4*a**3*b**5*d + 4*a**2*b**6*d*tan(c + d*x) + 2*a*b**7*d + 2*b**8*d*t
an(c + d*x)) + 2*B*a**4*b**2*log(a/b + tan(c + d*x))*tan(c + d*x)/(2*a**5*b
3*d + 2*a4*b**4*d*tan(c + d*x) + 4*a**3*b**5*d + 4*a**2*b**6*d*tan(c +
d*x) + 2*a*b**7*d + 2*b**8*d*tan(c + d*x)) + 6*B*a**3*b**3*log(a/b + tan(c
+ d*x))/(2*a**5*b**3*d + 2*a**4*b**4*d*tan(c + d*x) + 4*a**3*b**5*d + 4*a**
2*b**6*d*tan(c + d*x) + 2*a*b**7*d + 2*b**8*d*tan(c + d*x)) - B*a**3*b**3*l
og(tan(c + d*x)**2 + 1)/(2*a**5*b**3*d + 2*a**4*b**4*d*tan(c + d*x) + 4*a**
3*b**5*d + 4*a**2*b**6*d*tan(c + d*x) + 2*a*b**7*d + 2*b**8*d*tan(c + d*x))
 + 2*B*a**3*b**3/(2*a**5*b**3*d + 2*a**4*b**4*d*tan(c + d*x) + 4*a**3*b**5*
d + 4*a**2*b**6*d*tan(c + d*x) + 2*a*b**7*d + 2*b**8*d*tan(c + d*x)) - 4*B*
a**2*b**4*d*x/(2*a**5*b**3*d + 2*a**4*b**4*d*tan(c + d*x) + 4*a**3*b**5*d +
 4*a**2*b**6*d*tan(c + d*x) + 2*a*b**7*d + 2*b**8*d*tan(c + d*x)) + 6*B*a**
2*b**4*log(a/b + tan(c + d*x))*tan(c + d*x)/(2*a**5*b**3*d + 2*a**4*b**4*d*
tan(c + d*x) + 4*a**3*b**5*d + 4*a**2*b**6*d*tan(c + d*x) + 2*a*b**7*d + 2*
b**8*d*tan(c + d*x)) - B*a**2*b**4*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(2
*a**5*b**3*d + 2*a**4*b**4*d*tan(c + d*x) + 4*a**3*b**5*d + 4*a**2*b**6*d*t
an(c + d*x) + 2*a*b**7*d + 2*b**8*d*tan(c + d*x)) - 4*B*a*b**5*d*x*tan(c +
d*x)/(2*a**5*b**3*d + 2*a**4*b**4*d*tan(c + d*x) + 4*a**3*b**5*d + 4*a**2*b
6*d*tan(c + d*x) + 2*a*b7*d + 2*b**8*d*tan(c + d*x)) + B*a*b**5*log(tan
(c + d*x)**2 + 1)/(2*a**5*b**3*d + 2*a**4*b**4*d*tan(c + d*x) + 4*a**3*b**5
*d + 4*a**2*b**6*d*tan(c + d*x) + 2*a*b**7*d + 2*b**8*d*tan(c + d*x)) + B*b
6*log(tan(c + d*x)2 + 1)*tan(c + d*x)/(2*a**5*b**3*d + 2*a**4*b**4*d*ta
n(c + d*x) + 4*a**3*b**5*d + 4*a**2*b**6*d*tan(c + d*x) + 2*a*b**7*d + 2*b*
*8*d*tan(c + d*x)) - 4*C*a**6*log(a/b + tan(c + d*x))/(2*a**5*b**3*d + 2*a*

```
*4*b**4*d*tan(c + d*x) + 4*a**3*b**5*d + 4*a**2*b**6*d*tan(c + d*x) + 2*a*b
**7*d + 2*b**8*d*tan(c + d*x)) - 4*C*a**6/(2*a**5*b**3*d + 2*a**4*b**4*d*ta
n(c + d*x) + 4*a**3*b**5*d + 4*a**2*b**6*d*tan(c + d*x) + 2*a*b**7*d + 2*b*
*8*d*tan(c + d*x)) - 4*C*a**5*b*log(a/b + tan(c + d*x))*tan(c + d*x)/(2*a**
5*b**3*d + 2*a**4*b**4*d*tan(c + d*x) + 4*a**3*b**5*d + 4*a**2*b**6*d*tan(c
 + d*x) + 2*a*b**7*d + 2*b**8*d*tan(c + d*x)) - 8*C*a**4*b**2*log(a/b + tan
(c + d*x))/(2*a**5*b**3*d + 2*a**4*b**4*d*tan(c + d*x) + 4*a**3*b**5*d + 4*
a**2*b**6*d*tan(c + d*x) + 2*a*b**7*d + 2*b**8*d*tan(c + d*x)) + 2*C*a**4*b
**2*tan(c + d*x)**2/(2*a**5*b**3*d + 2*a**4*b**4*d*tan(c + d*x) + 4*a**3*b*
*5*d + 4*a**2*b**6*d*tan(c + d*x) + 2*a*b**7*d + 2*b**8*d*tan(c + d*x)) - 6
*C*a**4*b**2/(2*a**5*b**3*d + 2*a**4*b**4*d*tan(c + d*x) + 4*a**3*b**5*d +
4*a**2*b**6*d*tan(c + d*x) + 2*a*b**7*d + 2*b**8*d*tan(c + d*x)) + 2*C*a**3
*b**3*d*x/(2*a**5*b**3*d + 2*a**4*b**4*d*tan(c + d*x) + 4*a**3*b**5*d + 4*a
**2*b**6*d*tan(c + d*x) + 2*a*b**7*d + 2*b**8*d*tan(c + d*x)) - 8*C*a**3*b*
*3*log(a/b + tan(c + d*x))*tan(c + d*x)/(2*a**5*b**3*d + 2*a**4*b**4*d*tan(
c + d*x) + 4*a**3*b**5*d + 4*a**2*b**6*d*tan(c + d*x) + 2*a*b**7*d + 2*b**8
*d*tan(c + d*x)) + 2*C*a**2*b**4*d*x*tan(c + d*x)/(2*a**5*b**3*d + 2*a**4*b
**4*d*tan(c + d*x) + 4*a**3*b**5*d + 4*a**2*b**6*d*tan(c + d*x) + 2*a*b**7*
d + 2*b**8*d*tan(c + d*x)) - 2*C*a**2*b**4*log(tan(c + d*x)**2 + 1)/(2*a**5
*b**3*d + 2*a**4*b**4*d*tan(c + d*x) + 4*a**3*b**5*d + 4*a**2*b**6*d*tan(c
+ d*x) + 2*a*b**7*d + 2*b**8*d*tan(c + d*x)) + 4*C*a**2*b**4*tan(c + d*x)**
2/(2*a**5*b**3*d + 2*a**4*b**4*d*tan(c + d*x) + 4*a**3*b**5*d + 4*a**2*b**6
*d*tan(c + d*x) + 2*a*b**7*d + 2*b**8*d*tan(c + d*x)) - 2*C*a**2*b**4/(2*a*
*5*b**3*d + 2*a**4*b**4*d*tan(c + d*x) + 4*a**3*b**5*d + 4*a**2*b**6*d*tan(
c + d*x) + 2*a*b**7*d + 2*b**8*d*tan(c + d*x)) - 2*C*a*b**5*d*x/(2*a**5*b**
3*d + 2*a**4*b**4*d*tan(c + d*x) + 4*a**3*b**5*d + 4*a**2*b**6*d*tan(c + d*
x) + 2*a*b**7*d + 2*b**8*d*tan(c + d*x)) - 2*C*a*b**5*log(tan(c + d*x)**2 +
 1)*tan(c + d*x)/(2*a**5*b**3*d + 2*a**4*b**4*d*tan(c + d*x) + 4*a**3*b**5*
d + 4*a**2*b**6*d*tan(c + d*x) + 2*a*b**7*d + 2*b**8*d*tan(c + d*x)) - 2*C*
b**6*d*x*tan(c + d*x)/(2*a**5*b**3*d + 2*a**4*b**4*d*tan(c + d*x) + 4*a**3*
b**5*d + 4*a**2*b**6*d*tan(c + d*x) + 2*a*b**7*d + 2*b**8*d*tan(c + d*x)) +
 2*C*b**6*tan(c + d*x)**2/(2*a**5*b**3*d + 2*a**4*b**4*d*tan(c + d*x) + 4*a
**3*b**5*d + 4*a**2*b**6*d*tan(c + d*x) + 2*a*b**7*d + 2*b**8*d*tan(c + d*x
)), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.32 (sec) , antiderivative size = 220, normalized size of antiderivative = 1.06

$$\int \frac{\tan^2(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^2}\,dx$$

$$= \frac{\frac{2\,(Ca^2-2\,Bab-Cb^2)(dx+c)}{a^4+2\,a^2b^2+b^4} - \frac{2\,(2\,Ca^5-Ba^4b+4\,Ca^3b^2-3\,Ba^2b^3)\log(b\tan(dx+c)+a)}{a^4b^3+2\,a^2b^5+b^7} - \frac{(Ba^2+2\,Cab-Bb^2)\log\left(\tan(dx+c)^2+1\right)}{a^4+2\,a^2b^2+b^4} - \frac{}{a^3b^3}}{2\,d}$$

```
[In] integrate(tan(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x,
algorithm="maxima")
```

```
[Out] 1/2*(2*(C*a^2 - 2*B*a*b - C*b^2)*(d*x + c)/(a^4 + 2*a^2*b^2 + b^4) - 2*(2*C
*a^5 - B*a^4*b + 4*C*a^3*b^2 - 3*B*a^2*b^3)*log(b*tan(d*x + c) + a)/(a^4*b^
3 + 2*a^2*b^5 + b^7) - (B*a^2 + 2*C*a*b - B*b^2)*log(tan(d*x + c)^2 + 1)/(a
^4 + 2*a^2*b^2 + b^4) - 2*(C*a^4 - B*a^3*b)/(a^3*b^3 + a*b^5 + (a^2*b^4 + b
^6)*tan(d*x + c)) + 2*C*tan(d*x + c)/b^2)/d
```

Giac [A] (verification not implemented)

none

Time = 0.69 (sec) , antiderivative size = 290, normalized size of antiderivative = 1.39

$$\int \frac{\tan^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx = \frac{\frac{2\,(Ca^2-2\,Bab-Cb^2)(dx+c)}{a^4+2\,a^2b^2+b^4}-\frac{(Ba^2+2\,Cab-Bb^2)\log\left(\tan(dx+c)^2+1\right)}{a^4+2\,a^2b^2+b^4}-\frac{2\,(2\,Ca^5-Ba^4b+4\,Ca^3b^2-3\,Ba^2b^3)\log(|b\tan(dx+c)+a|)}{a^4b^3+2\,a^2b^5+b^7}+\frac{2\,C\,t}{}}{2\,d}$$

```
[In] integrate(tan(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x,
algorithm="giac")
```

```
[Out] 1/2*(2*(C*a^2 - 2*B*a*b - C*b^2)*(d*x + c)/(a^4 + 2*a^2*b^2 + b^4) - (B*a^2
 + 2*C*a*b - B*b^2)*log(tan(d*x + c)^2 + 1)/(a^4 + 2*a^2*b^2 + b^4) - 2*(2*
C*a^5 - B*a^4*b + 4*C*a^3*b^2 - 3*B*a^2*b^3)*log(abs(b*tan(d*x + c) + a))/(
a^4*b^3 + 2*a^2*b^5 + b^7) + 2*C*tan(d*x + c)/b^2 + 2*(2*C*a^5*b*tan(d*x +
c) - B*a^4*b^2*tan(d*x + c) + 4*C*a^3*b^3*tan(d*x + c) - 3*B*a^2*b^4*tan(d*
x + c) + C*a^6 + 3*C*a^4*b^2 - 2*B*a^3*b^3)/((a^4*b^3 + 2*a^2*b^5 + b^7)*(b
*tan(d*x + c) + a)))/d
```

Mupad [B] (verification not implemented)

Time = 9.24 (sec) , antiderivative size = 210, normalized size of antiderivative = 1.01

$$\begin{aligned}\int \frac{\tan^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx &= \frac{C\tan(c+d\,x)}{b^2\,d}-\frac{\ln\left(a+b\tan(c+d\,x)\right)\,\left(2\,C\,a^5-B\,a^4\,b+4\,C\,a^3\,b^2-3\,B\,a^2\,b^3\right)}{d\,\left(a^4\,b^3+2\,a^2\,b^5+b^7\right)}\\ &\quad-\frac{\ln\left(\tan(c+d\,x)-\mathrm{i}\right)\,\left(B+C\,1\mathrm{i}\right)}{2\,d\,\left(a^2+a\,b\,2\mathrm{i}-b^2\right)}-\frac{\ln\left(\tan(c+d\,x)+1\mathrm{i}\right)\,\left(C+B\,1\mathrm{i}\right)}{2\,d\,\left(a^2\,1\mathrm{i}+2\,a\,b-b^2\,1\mathrm{i}\right)}\\ &\quad-\frac{a^2\,\left(C\,a^2-B\,a\,b\right)}{b\,d\,\left(\tan\left(c+d\,x\right)\,b^3+a\,b^2\right)\,\left(a^2+b^2\right)}\end{aligned}$$

```
[In] int((tan(c + d*x)^2*(B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c + d*x
))^2,x)
```

```
[Out] (C*tan(c + d*x))/(b^2*d) - (log(a + b*tan(c + d*x))*(2*C*a^5 - 3*B*a^2*b^3
+ 4*C*a^3*b^2 - B*a^4*b))/(d*(b^7 + 2*a^2*b^5 + a^4*b^3)) - (log(tan(c + d*
x) - 1i)*(B + C*1i))/(2*d*(a*b*2i + a^2 - b^2)) - (log(tan(c + d*x) + 1i)*(
B*1i + C))/(2*d*(2*a*b + a^2*1i - b^2*1i)) - (a^2*(C*a^2 - B*a*b))/(b*d*(a*
b^2 + b^3*tan(c + d*x))*(a^2 + b^2))
```

3.33 $\int \frac{\tan(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx$

Optimal result 282
Rubi [A] (verified) 282
Mathematica [C] (verified) 284
Maple [A] (verified) 285
Fricas [B] (verification not implemented) 285
Sympy [C] (verification not implemented) 286
Maxima [A] (verification not implemented) 288
Giac [A] (verification not implemented) 288
Mupad [B] (verification not implemented) 289

Optimal result

Integrand size = 38, antiderivative size = 157

$$\int \frac{\tan(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx = -\frac{(a^2B-b^2B+2abC)\,x}{(a^2+b^2)^2}-\frac{(2abB-a^2C+b^2C)\log(\cos(c+dx))}{(a^2+b^2)^2\,d} - \frac{a(2b^3B-a^3C-3ab^2C)\log(a+b\tan(c+dx))}{b^2\left(a^2+b^2\right)^2 d}-\frac{a^2(bB-aC)}{b^2\left(a^2+b^2\right)d(a+b\tan(c+dx))}$$

```
[Out] -(B*a^2-B*b^2+2*C*a*b)*x/(a^2+b^2)^2-(2*B*a*b-C*a^2+C*b^2)*ln(cos(d*x+c))/(
a^2+b^2)^2/d-a*(2*B*b^3-C*a^3-3*C*a*b^2)*ln(a+b*tan(d*x+c))/b^2/(a^2+b^2)^2
/d-a^2*(B*b-C*a)/b^2/(a^2+b^2)/d/(a+b*tan(d*x+c))
```

Rubi [A] (verified)

Time = 0.36 (sec) , antiderivative size = 157, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.158, Rules used = {3713, 3685, 3707, 3698, 31, 3556}

$$\int \frac{\tan(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx = -\frac{a^2(bB-aC)}{b^2d\left(a^2+b^2\right)(a+b\tan(c+dx))}-\frac{(a^2(-C)+2abB+b^2C)\log(\cos(c+dx))}{d\left(a^2+b^2\right)^2} - \frac{x(a^2B+2abC-b^2B)}{(a^2+b^2)^2}-\frac{a(a^3(-C)-3ab^2C+2b^3B)\log(a+b\tan(c+dx))}{b^2d\left(a^2+b^2\right)^2}$$

```
[In] Int[(Tan[c + d*x]*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c + d*x])
^2,x]
```

```
[Out] -(((a^2*B - b^2*B + 2*a*b*C)*x)/(a^2 + b^2)^2) - ((2*a*b*B - a^2*C + b^2*C)
*Log[Cos[c + d*x]])/((a^2 + b^2)^2*d) - (a*(2*b^3*B - a^3*C - 3*a*b^2*C)*Lo
g[a + b*Tan[c + d*x]])/(b^2*(a^2 + b^2)^2*d) - (a^2*(b*B - a*C))/(b^2*(a^2
+ b^2)*d*(a + b*Tan[c + d*x]))
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3685

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^2*((A_.) + (B_.)*tan[(e_.) + (f
_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Simp[
(-(B*c - A*d))*(b*c - a*d)^2*((c + d*Tan[e + f*x])^(n + 1)/(f*d^2*(n + 1)*(
c^2 + d^2))), x] + Dist[1/(d*(c^2 + d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)
*Simp[B*(b*c - a*d)^2 + A*d*(a^2*c - b^2*c + 2*a*b*d) + d*(B*(a^2*c - b^2*c
 + 2*a*b*d) + A*(2*a*b*c - a^2*d + b^2*d))*Tan[e + f*x] + b^2*B*(c^2 + d^2)
*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] && NeQ[b*
c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[n, -1]
```

Rule 3698

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3707

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[(a*A + b*B - a
*C)*(x/(a^2 + b^2)), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
```

```
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\text{integral} = \int \frac{\tan^2(c+dx)(B+C\tan(c+dx))}{(a+b\tan(c+dx))^2}\,dx$$

$$= -\frac{a^2(bB-aC)}{b^2(a^2+b^2)\,d(a+b\tan(c+dx))} + \frac{\int \frac{-a(bB-aC)+b(bB-aC)\tan(c+dx)+(a^2+b^2)C\tan^2(c+dx)}{a+b\tan(c+dx)}\,dx}{b(a^2+b^2)}$$

$$= -\frac{(a^2B-b^2B+2abC)\,x}{(a^2+b^2)^2} - \frac{a^2(bB-aC)}{b^2(a^2+b^2)\,d(a+b\tan(c+dx))} + \frac{(2abB-a^2C+b^2C)\int \tan(c+dx)\,dx}{(a^2+b^2)^2} - \frac{(a(2b^3B-a^3C-3ab^2C))\int \frac{1+\tan^2(c+dx)}{a+b\tan(c+dx)}\,dx}{b(a^2+b^2)^2}$$

$$= -\frac{(a^2B-b^2B+2abC)\,x}{(a^2+b^2)^2} - \frac{(2abB-a^2C+b^2C)\log(\cos(c+dx))}{(a^2+b^2)^2\,d} - \frac{a^2(bB-aC)}{b^2(a^2+b^2)\,d(a+b\tan(c+dx))} - \frac{(a(2b^3B-a^3C-3ab^2C))\,\text{Subst}\left(\int \frac{1}{a+x}\,dx, x, b\tan(c+dx)\right)}{b^2(a^2+b^2)^2\,d}$$

$$= -\frac{(a^2B-b^2B+2abC)\,x}{(a^2+b^2)^2} - \frac{(2abB-a^2C+b^2C)\log(\cos(c+dx))}{(a^2+b^2)^2\,d} - \frac{a(2b^3B-a^3C-3ab^2C)\log(a+b\tan(c+dx))}{b^2(a^2+b^2)^2\,d} - \frac{a^2(bB-aC)}{b^2(a^2+b^2)\,d(a+b\tan(c+dx))}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 2.90 (sec) , antiderivative size = 146, normalized size of antiderivative = 0.93

$$\int \frac{\tan(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx$$

$$= \frac{\frac{i(B+iC)\log(i-\tan(c+dx))}{(a+ib)^2} - \frac{i(B-iC)\log(i+\tan(c+dx))}{(a-ib)^2} + \frac{2a\left(\left(-2bB+3aC+\frac{a^3C}{b^2}\right)\log(a+b\tan(c+dx))+\frac{a(a^2+b^2)(-bB+aC)}{b^2(a+b\tan(c+dx))}\right)}{(a^2+b^2)^2}}{2d}$$

```
[In] Integrate[(Tan[c + d*x]*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c +
 d*x])^2,x]
```

```
[Out] ((I*(B + I*C)*Log[I - Tan[c + d*x]])/(a + I*b)^2 - (I*(B - I*C)*Log[I + Tan
[c + d*x]])/(a - I*b)^2 + (2*a*((-2*b*B + 3*a*C + (a^3*C)/b^2)*Log[a + b*Ta
n[c + d*x]] + (a*(a^2 + b^2)*(-(b*B) + a*C))/(b^2*(a + b*Tan[c + d*x]))))/(
a^2 + b^2)^2)/(2*d)
```

Maple [A] (verified)

Time = 0.13 (sec) , antiderivative size = 155, normalized size of antiderivative = 0.99

method	result
derivativedivides	$\frac{\frac{\frac{(2Bab-Ca^2+Cb^2)\ln(1+\tan(dx+c)^2)}{2}+(-Ba^2+Bb^2-2Cab)\arctan(\tan(dx+c))}{(a^2+b^2)^2}-\frac{a^2(Bb-Ca)}{b^2(a^2+b^2)(a+b\tan(dx+c))}-\frac{a(2Bb^3-Ca^3-}{}}{d}$
default	$\frac{\frac{\frac{(2Bab-Ca^2+Cb^2)\ln(1+\tan(dx+c)^2)}{2}+(-Ba^2+Bb^2-2Cab)\arctan(\tan(dx+c))}{(a^2+b^2)^2}-\frac{a^2(Bb-Ca)}{b^2(a^2+b^2)(a+b\tan(dx+c))}-\frac{a(2Bb^3-Ca^3-}{}}{d}$
norman	$\frac{-\frac{a(Ba^2-Bb^2+2Cab)x}{a^4+2a^2b^2+b^4}-\frac{b(Ba^2-Bb^2+2Cab)x\tan(dx+c)}{a^4+2a^2b^2+b^4}-\frac{(Bab-Ca^2)a}{db^2(a^2+b^2)}}{a+b\tan(dx+c)}+\frac{(2Bab-Ca^2+Cb^2)\ln(1+\tan(dx+c)^2)}{2d(a^4+2a^2b^2+b^4)}-$
parallelrisch	$2Ca^5-2Ba^4b+2B\ln(1+\tan(dx+c)^2)\tan(dx+c)ab^4-4B\ln(a+b\tan(dx+c))\tan(dx+c)ab^4-C\ln(1+\tan(dx+c)^2)\,\mathrm{ta}$
risch	$\frac{xB}{2iba-a^2+b^2}+\frac{4iabBx}{a^4+2a^2b^2+b^4}-\frac{6ia^2Cx}{a^4+2a^2b^2+b^4}+\frac{2iCc}{b^2d}+\frac{2ia^2B}{(ib+a)d(-ib+a)^2(-ib\,e^{2i(dx+c)}+a\,e^{2i(dx+c)}+ib+a)}-$

```
[In] int(tan(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x,method=_R
ETURNVERBOSE)
```

```
[Out] 1/d*(1/(a^2+b^2)^2*(1/2*(2*B*a*b-C*a^2+C*b^2)*ln(1+tan(d*x+c)^2)+(-B*a^2+B*
b^2-2*C*a*b)*arctan(tan(d*x+c)))-a^2*(B*b-C*a)/b^2/(a^2+b^2)/(a+b*tan(d*x+c
))-a*(2*B*b^3-C*a^3-3*C*a*b^2)/(a^2+b^2)^2/b^2*ln(a+b*tan(d*x+c)))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 311 vs. $2(153) = 306$.

Time = 0.29 (sec) , antiderivative size = 311, normalized size of antiderivative = 1.98

$$\int \frac{\tan(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx$$

$$=\frac{2\,Ca^3b^2-2\,Ba^2b^3-2\,(Ba^3b^2+2\,Ca^2b^3-Bab^4)dx+(Ca^5+3\,Ca^3b^2-2\,Ba^2b^3+(Ca^4b+3\,Ca^2b^3-2\,}{}$$

```
[In] integrate(tan(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x, al
gorithm="fricas")
```

```
[Out] 1/2*(2*C*a^3*b^2 - 2*B*a^2*b^3 - 2*(B*a^3*b^2 + 2*C*a^2*b^3 - B*a*b^4)*d*x
+ (C*a^5 + 3*C*a^3*b^2 - 2*B*a^2*b^3 + (C*a^4*b + 3*C*a^2*b^3 - 2*B*a*b^4)*
tan(d*x + c))*log((b^2*tan(d*x + c)^2 + 2*a*b*tan(d*x + c) + a^2)/(tan(d*x
+ c)^2 + 1)) - (C*a^5 + 2*C*a^3*b^2 + C*a*b^4 + (C*a^4*b + 2*C*a^2*b^3 + C*
b^5)*tan(d*x + c))*log(1/(tan(d*x + c)^2 + 1)) - 2*(C*a^4*b - B*a^3*b^2 + (
B*a^2*b^3 + 2*C*a*b^4 - B*b^5)*d*x)*tan(d*x + c))/((a^4*b^3 + 2*a^2*b^5 + b
^7)*d*tan(d*x + c) + (a^5*b^2 + 2*a^3*b^4 + a*b^6)*d)
```

Sympy [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 1.08 (sec) , antiderivative size = 3497, normalized size of antiderivative = 22.27

$$\int \frac{\tan(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx = \text{Too large to display}$$

```
[In] integrate(tan(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c))**2,x)
```

```
[Out] Piecewise((zoo*x*(B*tan(c) + C*tan(c)**2)/tan(c), Eq(a, 0) & Eq(b, 0) & Eq(
d, 0)), ((-B*x + B*tan(c + d*x)/d - C*log(tan(c + d*x)**2 + 1)/(2*d) + C*ta
n(c + d*x)**2/(2*d))/a**2, Eq(b, 0)), (B*d*x*tan(c + d*x)**2/(4*b**2*d*tan(
c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) - 2*I*B*d*x*tan(c + d*x)/
(4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) - B*d*x/(4*
b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) - 3*B*tan(c +
d*x)/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + 2*I*
B/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + 3*I*C*d
*x*tan(c + d*x)**2/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*
b**2*d) + 6*C*d*x*tan(c + d*x)/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c
 + d*x) - 4*b**2*d) - 3*I*C*d*x/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(
c + d*x) - 4*b**2*d) + 2*C*log(tan(c + d*x)**2 + 1)*tan(c + d*x)**2/(4*b**2
*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) - 4*I*C*log(tan(c
+ d*x)**2 + 1)*tan(c + d*x)/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c +
d*x) - 4*b**2*d) - 2*C*log(tan(c + d*x)**2 + 1)/(4*b**2*d*tan(c + d*x)**2 -
 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) - 5*I*C*tan(c + d*x)/(4*b**2*d*tan(c +
 d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) - 4*C/(4*b**2*d*tan(c + d*x)
**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d), Eq(a, -I*b)), (B*d*x*tan(c + d*x
)**2/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + 2*I*
B*d*x*tan(c + d*x)/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*
b**2*d) - B*d*x/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**
2*d) - 3*B*tan(c + d*x)/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x)
 - 4*b**2*d) - 2*I*B/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) -
4*b**2*d) - 3*I*C*d*x*tan(c + d*x)**2/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*
```

```
d*tan(c + d*x) - 4*b**2*d) + 6*C*d*x*tan(c + d*x)/(4*b**2*d*tan(c + d*x)**2
 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + 3*I*C*d*x/(4*b**2*d*tan(c + d*x)**
2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + 2*C*log(tan(c + d*x)**2 + 1)*tan(
c + d*x)**2/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d)
 + 4*I*C*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(4*b**2*d*tan(c + d*x)**2 +
8*I*b**2*d*tan(c + d*x) - 4*b**2*d) - 2*C*log(tan(c + d*x)**2 + 1)/(4*b**2*
d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + 5*I*C*tan(c + d*x
)/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) - 4*C/(4*
b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d), Eq(a, I*b)),
(x*(B*tan(c) + C*tan(c)**2)*tan(c)/(a + b*tan(c))**2, Eq(d, 0)), (-2*B*a**4
*b/(2*a**5*b**2*d + 2*a**4*b**3*d*tan(c + d*x) + 4*a**3*b**4*d + 4*a**2*b**
5*d*tan(c + d*x) + 2*a*b**6*d + 2*b**7*d*tan(c + d*x)) - 2*B*a**3*b**2*d*x/
(2*a**5*b**2*d + 2*a**4*b**3*d*tan(c + d*x) + 4*a**3*b**4*d + 4*a**2*b**5*d
*tan(c + d*x) + 2*a*b**6*d + 2*b**7*d*tan(c + d*x)) - 2*B*a**2*b**3*d*x*tan
(c + d*x)/(2*a**5*b**2*d + 2*a**4*b**3*d*tan(c + d*x) + 4*a**3*b**4*d + 4*a
**2*b**5*d*tan(c + d*x) + 2*a*b**6*d + 2*b**7*d*tan(c + d*x)) - 4*B*a**2*b*
*3*log(a/b + tan(c + d*x))/(2*a**5*b**2*d + 2*a**4*b**3*d*tan(c + d*x) + 4*
a**3*b**4*d + 4*a**2*b**5*d*tan(c + d*x) + 2*a*b**6*d + 2*b**7*d*tan(c + d*
x)) + 2*B*a**2*b**3*log(tan(c + d*x)**2 + 1)/(2*a**5*b**2*d + 2*a**4*b**3*d
*tan(c + d*x) + 4*a**3*b**4*d + 4*a**2*b**5*d*tan(c + d*x) + 2*a*b**6*d + 2
*b**7*d*tan(c + d*x)) - 2*B*a**2*b**3/(2*a**5*b**2*d + 2*a**4*b**3*d*tan(c
+ d*x) + 4*a**3*b**4*d + 4*a**2*b**5*d*tan(c + d*x) + 2*a*b**6*d + 2*b**7*d
*tan(c + d*x)) + 2*B*a*b**4*d*x/(2*a**5*b**2*d + 2*a**4*b**3*d*tan(c + d*x)
 + 4*a**3*b**4*d + 4*a**2*b**5*d*tan(c + d*x) + 2*a*b**6*d + 2*b**7*d*tan(c
 + d*x)) - 4*B*a*b**4*log(a/b + tan(c + d*x))*tan(c + d*x)/(2*a**5*b**2*d +
 2*a**4*b**3*d*tan(c + d*x) + 4*a**3*b**4*d + 4*a**2*b**5*d*tan(c + d*x) +
2*a*b**6*d + 2*b**7*d*tan(c + d*x)) + 2*B*a*b**4*log(tan(c + d*x)**2 + 1)*t
an(c + d*x)/(2*a**5*b**2*d + 2*a**4*b**3*d*tan(c + d*x) + 4*a**3*b**4*d + 4
*a**2*b**5*d*tan(c + d*x) + 2*a*b**6*d + 2*b**7*d*tan(c + d*x)) + 2*B*b**5*
d*x*tan(c + d*x)/(2*a**5*b**2*d + 2*a**4*b**3*d*tan(c + d*x) + 4*a**3*b**4*
d + 4*a**2*b**5*d*tan(c + d*x) + 2*a*b**6*d + 2*b**7*d*tan(c + d*x)) + 2*C*
a**5*log(a/b + tan(c + d*x))/(2*a**5*b**2*d + 2*a**4*b**3*d*tan(c + d*x) +
4*a**3*b**4*d + 4*a**2*b**5*d*tan(c + d*x) + 2*a*b**6*d + 2*b**7*d*tan(c +
d*x)) + 2*C*a**5/(2*a**5*b**2*d + 2*a**4*b**3*d*tan(c + d*x) + 4*a**3*b**4*
d + 4*a**2*b**5*d*tan(c + d*x) + 2*a*b**6*d + 2*b**7*d*tan(c + d*x)) + 2*C*
a**4*b*log(a/b + tan(c + d*x))*tan(c + d*x)/(2*a**5*b**2*d + 2*a**4*b**3*d*
tan(c + d*x) + 4*a**3*b**4*d + 4*a**2*b**5*d*tan(c + d*x) + 2*a*b**6*d + 2*
b**7*d*tan(c + d*x)) + 6*C*a**3*b**2*log(a/b + tan(c + d*x))/(2*a**5*b**2*d
 + 2*a**4*b**3*d*tan(c + d*x) + 4*a**3*b**4*d + 4*a**2*b**5*d*tan(c + d*x)
+ 2*a*b**6*d + 2*b**7*d*tan(c + d*x)) - C*a**3*b**2*log(tan(c + d*x)**2 + 1
)/(2*a**5*b**2*d + 2*a**4*b**3*d*tan(c + d*x) + 4*a**3*b**4*d + 4*a**2*b**5
*d*tan(c + d*x) + 2*a*b**6*d + 2*b**7*d*tan(c + d*x)) + 2*C*a**3*b**2/(2*a*
*5*b**2*d + 2*a**4*b**3*d*tan(c + d*x) + 4*a**3*b**4*d + 4*a**2*b**5*d*tan(
c + d*x) + 2*a*b**6*d + 2*b**7*d*tan(c + d*x)) - 4*C*a**2*b**3*d*x/(2*a**5*
b**2*d + 2*a**4*b**3*d*tan(c + d*x) + 4*a**3*b**4*d + 4*a**2*b**5*d*tan(c +
```

```
d*x) + 2*a*b**6*d + 2*b**7*d*tan(c + d*x)) + 6*C*a**2*b**3*log(a/b + tan(c
 + d*x))*tan(c + d*x)/(2*a**5*b**2*d + 2*a**4*b**3*d*tan(c + d*x) + 4*a**3*
b**4*d + 4*a**2*b**5*d*tan(c + d*x) + 2*a*b**6*d + 2*b**7*d*tan(c + d*x)) -
 C*a**2*b**3*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(2*a**5*b**2*d + 2*a**4*
b**3*d*tan(c + d*x) + 4*a**3*b**4*d + 4*a**2*b**5*d*tan(c + d*x) + 2*a*b**6
*d + 2*b**7*d*tan(c + d*x)) - 4*C*a*b**4*d*x*tan(c + d*x)/(2*a**5*b**2*d +
2*a**4*b**3*d*tan(c + d*x) + 4*a**3*b**4*d + 4*a**2*b**5*d*tan(c + d*x) + 2
*a*b**6*d + 2*b**7*d*tan(c + d*x)) + C*a*b**4*log(tan(c + d*x)**2 + 1)/(2*a
**5*b**2*d + 2*a**4*b**3*d*tan(c + d*x) + 4*a**3*b**4*d + 4*a**2*b**5*d*tan
(c + d*x) + 2*a*b**6*d + 2*b**7*d*tan(c + d*x)) + C*b**5*log(tan(c + d*x)**
2 + 1)*tan(c + d*x)/(2*a**5*b**2*d + 2*a**4*b**3*d*tan(c + d*x) + 4*a**3*b*
*4*d + 4*a**2*b**5*d*tan(c + d*x) + 2*a*b**6*d + 2*b**7*d*tan(c + d*x)), Tr
ue))
```

Maxima [A] (verification not implemented)

none

Time = 0.33 (sec) , antiderivative size = 197, normalized size of antiderivative = 1.25

$$\int \frac{\tan(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx = -\frac{\frac{2\,(Ba^2+2\,Cab-Bb^2)(dx+c)}{a^4+2\,a^2b^2+b^4}-\frac{2\,(Ca^4+3\,Ca^2b^2-2\,Bab^3)\log(b\tan(dx+c)+a)}{a^4b^2+2\,a^2b^4+b^6}+\frac{(Ca^2-2\,Bab-Cb^2)\log\left(\tan(dx+c)^2+1\right)}{a^4+2\,a^2b^2+b^4}-\frac{2\,(C}{a^3b^2+ab^4+(}}{2\,d}$$

```
[In] integrate(tan(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x, al
gorithm="maxima")
```

```
[Out] -1/2*(2*(B*a^2 + 2*C*a*b - B*b^2)*(d*x + c)/(a^4 + 2*a^2*b^2 + b^4) - 2*(C*
a^4 + 3*C*a^2*b^2 - 2*B*a*b^3)*log(b*tan(d*x + c) + a)/(a^4*b^2 + 2*a^2*b^4
 + b^6) + (C*a^2 - 2*B*a*b - C*b^2)*log(tan(d*x + c)^2 + 1)/(a^4 + 2*a^2*b^
2 + b^4) - 2*(C*a^3 - B*a^2*b)/(a^3*b^2 + a*b^4 + (a^2*b^3 + b^5)*tan(d*x +
 c)))/d
```

Giac [A] (verification not implemented)

none

Time = 0.56 (sec) , antiderivative size = 244, normalized size of antiderivative = 1.55

$$\int \frac{\tan(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx = -\frac{\frac{2\,(Ba^2+2\,Cab-Bb^2)(dx+c)}{a^4+2\,a^2b^2+b^4}+\frac{(Ca^2-2\,Bab-Cb^2)\log\left(\tan(dx+c)^2+1\right)}{a^4+2\,a^2b^2+b^4}-\frac{2\,(Ca^4+3\,Ca^2b^2-2\,Bab^3)\log(|b\tan(dx+c)+a|)}{a^4b^2+2\,a^2b^4+b^6}+\frac{2\,(Ca^4\tan(}{}}{2\,d}$$

```
[In] integrate(tan(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x, al
gorithm="giac")
```

```
[Out] -1/2*(2*(B*a^2 + 2*C*a*b - B*b^2)*(d*x + c)/(a^4 + 2*a^2*b^2 + b^4) + (C*a^
2 - 2*B*a*b - C*b^2)*log(tan(d*x + c)^2 + 1)/(a^4 + 2*a^2*b^2 + b^4) - 2*(C
*a^4 + 3*C*a^2*b^2 - 2*B*a*b^3)*log(abs(b*tan(d*x + c) + a))/(a^4*b^2 + 2*a
^2*b^4 + b^6) + 2*(C*a^4*tan(d*x + c) + 3*C*a^2*b^2*tan(d*x + c) - 2*B*a*b^
3*tan(d*x + c) + B*a^4 + 2*C*a^3*b - B*a^2*b^2)/((a^4*b + 2*a^2*b^3 + b^5)*
(b*tan(d*x + c) + a)))/d
```

Mupad [B] (verification not implemented)

Time = 8.65 (sec) , antiderivative size = 165, normalized size of antiderivative = 1.05

$$\int \frac{\tan(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^2}\,dx$$
$$= \frac{\ln\left(\tan(c+d\,x)+1\mathrm{i}\right)\,\left(C+B\,1\mathrm{i}\right)}{2\,d\,\left(-a^2+a\,b\,2\mathrm{i}+b^2\right)} + \frac{\ln\left(\tan(c+d\,x)-\mathrm{i}\right)\,\left(B+C\,1\mathrm{i}\right)}{2\,d\,\left(-a^2\,1\mathrm{i}+2\,a\,b+b^2\,1\mathrm{i}\right)}$$
$$- \frac{a^2\,(B\,b-C\,a)}{b^2\,d\,\left(a^2+b^2\right)\,\left(a+b\tan\left(c+d\,x\right)\right)}$$
$$+ \frac{a\,\ln\left(a+b\tan(c+d\,x)\right)\,\left(C\,a^3+3\,C\,a\,b^2-2\,B\,b^3\right)}{b^2\,d\,\left(a^2+b^2\right)^2}$$

```
[In] int((tan(c + d*x)*(B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c + d*x))
^2,x)
```

```
[Out] (log(tan(c + d*x) + 1i)*(B*1i + C))/(2*d*(a*b*2i - a^2 + b^2)) + (log(tan(c
 + d*x) - 1i)*(B + C*1i))/(2*d*(2*a*b - a^2*1i + b^2*1i)) - (a^2*(B*b - C*a
))/(b^2*d*(a^2 + b^2)*(a + b*tan(c + d*x))) + (a*log(a + b*tan(c + d*x))*(C
*a^3 - 2*B*b^3 + 3*C*a*b^2))/(b^2*d*(a^2 + b^2)^2)
```

3.34 $\int \frac{B\tan(c+dx)+C\tan^2(c+dx)}{(a+b\tan(c+dx))^2}\,dx$

Optimal result 290
Rubi [A] (verified) 290
Mathematica [C] (verified) 292
Maple [A] (verified) 292
Fricas [A] (verification not implemented) 293
Sympy [C] (verification not implemented) 293
Maxima [A] (verification not implemented) 295
Giac [B] (verification not implemented) 295
Mupad [B] (verification not implemented) 296

Optimal result

Integrand size = 32, antiderivative size = 115

$$\int \frac{B\tan(c+dx)+C\tan^2(c+dx)}{(a+b\tan(c+dx))^2}\,dx = \frac{(2abB-a^2C+b^2C)\,x}{(a^2+b^2)^2}-\frac{(a^2B-b^2B+2abC)\log(a\cos(c+dx)+b\sin(c+dx))}{(a^2+b^2)^2\,d}+\frac{a(bB-aC)}{b\,(a^2+b^2)\,d(a+b\tan(c+dx))}$$

```
[Out] (2*B*a*b-C*a^2+C*b^2)*x/(a^2+b^2)^2-(B*a^2-B*b^2+2*C*a*b)*ln(a*cos(d*x+c)+b
*sin(d*x+c))/(a^2+b^2)^2/d+a*(B*b-C*a)/b/(a^2+b^2)/d/(a+b*tan(d*x+c))
```

Rubi [A] (verified)

Time = 0.17 (sec) , antiderivative size = 115, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 3, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.094, Rules used = {3709, 3612, 3611}

$$\int \frac{B\tan(c+dx)+C\tan^2(c+dx)}{(a+b\tan(c+dx))^2}\,dx = \frac{a(bB-aC)}{bd\,(a^2+b^2)\,(a+b\tan(c+dx))}-\frac{(a^2B+2abC-b^2B)\log(a\cos(c+dx)+b\sin(c+dx))}{d\,(a^2+b^2)^2}+\frac{x(a^2(-C)+2abB+b^2C)}{(a^2+b^2)^2}$$

```
[In] Int[(B*Tan[c + d*x] + C*Tan[c + d*x]^2)/(a + b*Tan[c + d*x])^2,x]
```

```
[Out] ((2*a*b*B - a^2*C + b^2*C)*x)/(a^2 + b^2)^2 - ((a^2*B - b^2*B + 2*a*b*C)*Lo
g[a*Cos[c + d*x] + b*Sin[c + d*x]])/((a^2 + b^2)^2*d) + (a*(b*B - a*C))/(b*
(a^2 + b^2)*d*(a + b*Tan[c + d*x]))
```

Rule 3611

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c/(b*f))*Log[RemoveContent[a*Cos[e + f*x] + b*Si
n[e + f*x], x]], x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3612

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[(a*c + b*d)*(x/(a^2 + b^2)), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3709

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 -
 a*b*B + a^2*C)*((a + b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1)*(a^2 + b^2))), x
] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[b*B + a*(A -
C) - (A*b - a*B - b*C)*Tan[e + f*x], x], x], x] /; FreeQ[{a, b, e, f, A, B,
 C}, x] && NeQ[A*b^2 - a*b*B + a^2*C, 0] && LtQ[m, -1] && NeQ[a^2 + b^2, 0]
```

Rubi steps

$$
\begin{aligned}
\text{integral} &= \frac{a(bB-aC)}{b\left(a^2+b^2\right)d(a+b\tan(c+dx))}+\frac{\int\frac{bB-aC+(aB+bC)\tan(c+dx)}{a+b\tan(c+dx)}\,dx}{a^2+b^2}\\
&= \frac{\left(2abB-a^2C+b^2C\right)x}{\left(a^2+b^2\right)^2}+\frac{a(bB-aC)}{b\left(a^2+b^2\right)d(a+b\tan(c+dx))}\\
&\quad-\frac{\left(a^2B-b^2B+2abC\right)\int\frac{b-a\tan(c+dx)}{a+b\tan(c+dx)}\,dx}{\left(a^2+b^2\right)^2}\\
&= \frac{\left(2abB-a^2C+b^2C\right)x}{\left(a^2+b^2\right)^2}-\frac{\left(a^2B-b^2B+2abC\right)\log(a\cos(c+dx)+b\sin(c+dx))}{\left(a^2+b^2\right)^2d}\\
&\quad+\frac{a(bB-aC)}{b\left(a^2+b^2\right)d(a+b\tan(c+dx))}
\end{aligned}
$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 2.30 (sec) , antiderivative size = 140, normalized size of antiderivative = 1.22

$$\int \frac{B\tan(c+dx)+C\tan^2(c+dx)}{(a+b\tan(c+dx))^2}\,dx$$
$$=\frac{\frac{(B+iC)\log(i-\tan(c+dx))}{(a+ib)^2}+\frac{(B-iC)\log(i+\tan(c+dx))}{(a-ib)^2}+\frac{2\left((-a^2B+b^2B-2abC)\log(a+b\tan(c+dx))-\frac{a(a^2+b^2)(-bB+aC)}{b(a+b\tan(c+dx))}\right)}{(a^2+b^2)^2}}{2d}$$

```
[In] Integrate[(B*Tan[c + d*x] + C*Tan[c + d*x]^2)/(a + b*Tan[c + d*x])^2,x]
```

```
[Out] (((B + I*C)*Log[I - Tan[c + d*x]])/(a + I*b)^2 + ((B - I*C)*Log[I + Tan[c +
 d*x]])/(a - I*b)^2 + (2*((-(a^2*B) + b^2*B - 2*a*b*C)*Log[a + b*Tan[c + d*
x]] - (a*(a^2 + b^2)*(-(b*B) + a*C))/(b*(a + b*Tan[c + d*x]))))/(a^2 + b^2)
^2)/(2*d)
```

Maple [A] (verified)

Time = 0.09 (sec) , antiderivative size = 145, normalized size of antiderivative = 1.26

method	result
derivativedivides	$\frac{\frac{\frac{(Ba^2-Bb^2+2Cab)\ln(1+\tan(dx+c)^2)}{2}+(2Bab-Ca^2+Cb^2)\arctan(\tan(dx+c))}{(a^2+b^2)^2}+\frac{a(Bb-Ca)}{(a^2+b^2)b(a+b\tan(dx+c))}-\frac{(Ba^2-Bb^2+2Cab)\ldots}{(a^2+b\ldots}}{d}$
default	$\frac{\frac{\frac{(Ba^2-Bb^2+2Cab)\ln(1+\tan(dx+c)^2)}{2}+(2Bab-Ca^2+Cb^2)\arctan(\tan(dx+c))}{(a^2+b^2)^2}+\frac{a(Bb-Ca)}{(a^2+b^2)b(a+b\tan(dx+c))}-\frac{(Ba^2-Bb^2+2Cab)\ldots}{(a^2+b\ldots}}{d}$
norman	$\frac{\frac{a(2Bab-Ca^2+Cb^2)x}{a^4+2a^2b^2+b^4}+\frac{b(2Bab-Ca^2+Cb^2)x\tan(dx+c)}{a^4+2a^2b^2+b^4}+\frac{a(Bb-Ca)}{(a^2+b^2)bd}}{a+b\tan(dx+c)}+\frac{(Ba^2-Bb^2+2Cab)\ln(1+\tan(dx+c)^2)}{2d(a^4+2a^2b^2+b^4)}-\frac{(Ba\ldots}{}$
parallelrisch	$2Ba^3b+2Bab^3-2Ca^2b^2-2Ca^4+2Cxab^3d+4Bx\tan(dx+c)ab^3d-2Cx\tan(dx+c)a^2b^2d-B\ln(1+\tan(dx+c)^2)ab^3-2B\ldots$
risch	$\frac{ixB}{2iba-a^2+b^2}+\frac{xC}{2iba-a^2+b^2}+\frac{2ia^2Bx}{a^4+2a^2b^2+b^4}-\frac{2iBb^2x}{a^4+2a^2b^2+b^4}+\frac{4iCabx}{a^4+2a^2b^2+b^4}+\frac{2ia^2Bc}{(a^4+2a^2b^2+b^4)d}-\frac{2iB\ldots}{d(a^4+2a\ldots}$

```
[In] int((B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x,method=_RETURNVERBOS
E)
```

```
[Out] 1/d*(1/(a^2+b^2)^2*(1/2*(B*a^2-B*b^2+2*C*a*b)*ln(1+tan(d*x+c)^2)+(2*B*a*b-C
*a^2+C*b^2)*arctan(tan(d*x+c)))+a*(B*b-C*a)/(a^2+b^2)/b/(a+b*tan(d*x+c))-(B
*a^2-B*b^2+2*C*a*b)/(a^2+b^2)^2*ln(a+b*tan(d*x+c)))
```

Fricas [A] (verification not implemented)

none

Time = 0.26 (sec) , antiderivative size = 221, normalized size of antiderivative = 1.92

$$\int \frac{B\tan(c+dx)+C\tan^2(c+dx)}{(a+b\tan(c+dx))^2}\,dx =$$
$$-\frac{2\,Ca^2b-2\,Bab^2+2\,(Ca^3-2\,Ba^2b-Cab^2)dx+(Ba^3+2\,Ca^2b-Bab^2+(Ba^2b+2\,Cab^2-Bb^3)\tan}{2\,((a^4b+2\,a^2b^3+b^5)d\tan(dx}$$

```
[In] integrate((B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x, algorithm="fr
icas")
```

```
[Out] -1/2*(2*C*a^2*b - 2*B*a*b^2 + 2*(C*a^3 - 2*B*a^2*b - C*a*b^2)*d*x + (B*a^3
+ 2*C*a^2*b - B*a*b^2 + (B*a^2*b + 2*C*a*b^2 - B*b^3)*tan(d*x + c))*log((b^
2*tan(d*x + c)^2 + 2*a*b*tan(d*x + c) + a^2)/(tan(d*x + c)^2 + 1)) - 2*(C*a
^3 - B*a^2*b - (C*a^2*b - 2*B*a*b^2 - C*b^3)*d*x)*tan(d*x + c))/((a^4*b + 2
*a^2*b^3 + b^5)*d*tan(d*x + c) + (a^5 + 2*a^3*b^2 + a*b^4)*d)
```

Sympy [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 0.87 (sec) , antiderivative size = 2995, normalized size of antiderivative = 26.04

$$\int \frac{B\tan(c+dx)+C\tan^2(c+dx)}{(a+b\tan(c+dx))^2}\,dx = \text{Too large to display}$$

```
[In] integrate((B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c))**2,x)
```

```
[Out] Piecewise((zoo*x*(B*tan(c) + C*tan(c)**2)/tan(c)**2, Eq(a, 0) & Eq(b, 0) &
Eq(d, 0)), ((B*log(tan(c + d*x)**2 + 1)/(2*d) - C*x + C*tan(c + d*x)/d)/a**
2, Eq(b, 0)), (I*B*d*x*tan(c + d*x)**2/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**2
*d*tan(c + d*x) - 4*b**2*d) + 2*B*d*x*tan(c + d*x)/(4*b**2*d*tan(c + d*x)**
2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) - I*B*d*x/(4*b**2*d*tan(c + d*x)**2
 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + I*B*tan(c + d*x)/(4*b**2*d*tan(c +
 d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + C*d*x*tan(c + d*x)**2/(4*b
**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) - 2*I*C*d*x*tan
(c + d*x)/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) -
 C*d*x/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) - 3*
C*tan(c + d*x)/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2
*d) + 2*I*C/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d)
, Eq(a, -I*b)), (-I*B*d*x*tan(c + d*x)**2/(4*b**2*d*tan(c + d*x)**2 + 8*I*b
**2*d*tan(c + d*x) - 4*b**2*d) + 2*B*d*x*tan(c + d*x)/(4*b**2*d*tan(c + d*x
)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + I*B*d*x/(4*b**2*d*tan(c + d*x)
```

```
**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) - I*B*tan(c + d*x)/(4*b**2*d*tan(
c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + C*d*x*tan(c + d*x)**2/(
4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + 2*I*C*d*x*
tan(c + d*x)/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d
) - C*d*x/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) -
 3*C*tan(c + d*x)/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b
**2*d) - 2*I*C/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2
*d), Eq(a, I*b)), (x*(B*tan(c) + C*tan(c)**2)/(a + b*tan(c))**2, Eq(d, 0)),
 (-2*B*a**3*b*log(a/b + tan(c + d*x))/(2*a**5*b*d + 2*a**4*b**2*d*tan(c + d
*x) + 4*a**3*b**3*d + 4*a**2*b**4*d*tan(c + d*x) + 2*a*b**5*d + 2*b**6*d*ta
n(c + d*x)) + B*a**3*b*log(tan(c + d*x)**2 + 1)/(2*a**5*b*d + 2*a**4*b**2*d
*tan(c + d*x) + 4*a**3*b**3*d + 4*a**2*b**4*d*tan(c + d*x) + 2*a*b**5*d + 2
*b**6*d*tan(c + d*x)) + 2*B*a**3*b/(2*a**5*b*d + 2*a**4*b**2*d*tan(c + d*x)
 + 4*a**3*b**3*d + 4*a**2*b**4*d*tan(c + d*x) + 2*a*b**5*d + 2*b**6*d*tan(c
 + d*x)) + 4*B*a**2*b**2*d*x/(2*a**5*b*d + 2*a**4*b**2*d*tan(c + d*x) + 4*a
**3*b**3*d + 4*a**2*b**4*d*tan(c + d*x) + 2*a*b**5*d + 2*b**6*d*tan(c + d*x
)) - 2*B*a**2*b**2*log(a/b + tan(c + d*x))*tan(c + d*x)/(2*a**5*b*d + 2*a**
4*b**2*d*tan(c + d*x) + 4*a**3*b**3*d + 4*a**2*b**4*d*tan(c + d*x) + 2*a*b*
*5*d + 2*b**6*d*tan(c + d*x)) + B*a**2*b**2*log(tan(c + d*x)**2 + 1)*tan(c
+ d*x)/(2*a**5*b*d + 2*a**4*b**2*d*tan(c + d*x) + 4*a**3*b**3*d + 4*a**2*b*
*4*d*tan(c + d*x) + 2*a*b**5*d + 2*b**6*d*tan(c + d*x)) + 4*B*a*b**3*d*x*ta
n(c + d*x)/(2*a**5*b*d + 2*a**4*b**2*d*tan(c + d*x) + 4*a**3*b**3*d + 4*a**
2*b**4*d*tan(c + d*x) + 2*a*b**5*d + 2*b**6*d*tan(c + d*x)) + 2*B*a*b**3*lo
g(a/b + tan(c + d*x))/(2*a**5*b*d + 2*a**4*b**2*d*tan(c + d*x) + 4*a**3*b**
3*d + 4*a**2*b**4*d*tan(c + d*x) + 2*a*b**5*d + 2*b**6*d*tan(c + d*x)) - B*
a*b**3*log(tan(c + d*x)**2 + 1)/(2*a**5*b*d + 2*a**4*b**2*d*tan(c + d*x) +
4*a**3*b**3*d + 4*a**2*b**4*d*tan(c + d*x) + 2*a*b**5*d + 2*b**6*d*tan(c +
d*x)) + 2*B*a*b**3/(2*a**5*b*d + 2*a**4*b**2*d*tan(c + d*x) + 4*a**3*b**3*d
 + 4*a**2*b**4*d*tan(c + d*x) + 2*a*b**5*d + 2*b**6*d*tan(c + d*x)) + 2*B*b
**4*log(a/b + tan(c + d*x))*tan(c + d*x)/(2*a**5*b*d + 2*a**4*b**2*d*tan(c
+ d*x) + 4*a**3*b**3*d + 4*a**2*b**4*d*tan(c + d*x) + 2*a*b**5*d + 2*b**6*d
*tan(c + d*x)) - B*b**4*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(2*a**5*b*d +
 2*a**4*b**2*d*tan(c + d*x) + 4*a**3*b**3*d + 4*a**2*b**4*d*tan(c + d*x) +
2*a*b**5*d + 2*b**6*d*tan(c + d*x)) - 2*C*a**4/(2*a**5*b*d + 2*a**4*b**2*d*
tan(c + d*x) + 4*a**3*b**3*d + 4*a**2*b**4*d*tan(c + d*x) + 2*a*b**5*d + 2*
b**6*d*tan(c + d*x)) - 2*C*a**3*b*d*x/(2*a**5*b*d + 2*a**4*b**2*d*tan(c + d
*x) + 4*a**3*b**3*d + 4*a**2*b**4*d*tan(c + d*x) + 2*a*b**5*d + 2*b**6*d*ta
n(c + d*x)) - 2*C*a**2*b**2*d*x*tan(c + d*x)/(2*a**5*b*d + 2*a**4*b**2*d*ta
n(c + d*x) + 4*a**3*b**3*d + 4*a**2*b**4*d*tan(c + d*x) + 2*a*b**5*d + 2*b*
*6*d*tan(c + d*x)) - 4*C*a**2*b**2*log(a/b + tan(c + d*x))/(2*a**5*b*d + 2*
a**4*b**2*d*tan(c + d*x) + 4*a**3*b**3*d + 4*a**2*b**4*d*tan(c + d*x) + 2*a
*b**5*d + 2*b**6*d*tan(c + d*x)) + 2*C*a**2*b**2*log(tan(c + d*x)**2 + 1)/(
2*a**5*b*d + 2*a**4*b**2*d*tan(c + d*x) + 4*a**3*b**3*d + 4*a**2*b**4*d*tan
(c + d*x) + 2*a*b**5*d + 2*b**6*d*tan(c + d*x)) - 2*C*a**2*b**2/(2*a**5*b*d
 + 2*a**4*b**2*d*tan(c + d*x) + 4*a**3*b**3*d + 4*a**2*b**4*d*tan(c + d*x)
```

```
+ 2*a*b**5*d + 2*b**6*d*tan(c + d*x)) + 2*C*a*b**3*d*x/(2*a**5*b*d + 2*a**4
*b**2*d*tan(c + d*x) + 4*a**3*b**3*d + 4*a**2*b**4*d*tan(c + d*x) + 2*a*b**
5*d + 2*b**6*d*tan(c + d*x)) - 4*C*a*b**3*log(a/b + tan(c + d*x))*tan(c + d
*x)/(2*a**5*b*d + 2*a**4*b**2*d*tan(c + d*x) + 4*a**3*b**3*d + 4*a**2*b**4*
d*tan(c + d*x) + 2*a*b**5*d + 2*b**6*d*tan(c + d*x)) + 2*C*a*b**3*log(tan(c
 + d*x)**2 + 1)*tan(c + d*x)/(2*a**5*b*d + 2*a**4*b**2*d*tan(c + d*x) + 4*a
**3*b**3*d + 4*a**2*b**4*d*tan(c + d*x) + 2*a*b**5*d + 2*b**6*d*tan(c + d*x
)) + 2*C*b**4*d*x*tan(c + d*x)/(2*a**5*b*d + 2*a**4*b**2*d*tan(c + d*x) + 4
*a**3*b**3*d + 4*a**2*b**4*d*tan(c + d*x) + 2*a*b**5*d + 2*b**6*d*tan(c + d
*x)), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.32 (sec) , antiderivative size = 185, normalized size of antiderivative = 1.61

$$\int \frac{B\tan(c+dx)+C\tan^2(c+dx)}{(a+b\tan(c+dx))^2}\,dx = -\frac{\frac{2\,(Ca^2-2\,Bab-Cb^2)(dx+c)}{a^4+2\,a^2b^2+b^4}+\frac{2\,(Ba^2+2\,Cab-Bb^2)\log(b\tan(dx+c)+a)}{a^4+2\,a^2b^2+b^4}-\frac{(Ba^2+2\,Cab-Bb^2)\log\left(\tan(dx+c)^2+1\right)}{a^4+2\,a^2b^2+b^4}+\frac{2\,(Ca^2}{a^3b+ab^3+(a^2b^2}}{2\,d}$$

```
[In] integrate((B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x, algorithm="ma
xima")
```

```
[Out] -1/2*(2*(C*a^2 - 2*B*a*b - C*b^2)*(d*x + c)/(a^4 + 2*a^2*b^2 + b^4) + 2*(B*
a^2 + 2*C*a*b - B*b^2)*log(b*tan(d*x + c) + a)/(a^4 + 2*a^2*b^2 + b^4) - (B
*a^2 + 2*C*a*b - B*b^2)*log(tan(d*x + c)^2 + 1)/(a^4 + 2*a^2*b^2 + b^4) + 2
*(C*a^2 - B*a*b)/(a^3*b + a*b^3 + (a^2*b^2 + b^4)*tan(d*x + c)))/d
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 241 vs. 2(117) = 234.

Time = 0.54 (sec) , antiderivative size = 241, normalized size of antiderivative = 2.10

$$\int \frac{B\tan(c+dx)+C\tan^2(c+dx)}{(a+b\tan(c+dx))^2}\,dx = -\frac{\frac{2\,(Ca^2-2\,Bab-Cb^2)(dx+c)}{a^4+2\,a^2b^2+b^4}-\frac{(Ba^2+2\,Cab-Bb^2)\log\left(\tan(dx+c)^2+1\right)}{a^4+2\,a^2b^2+b^4}+\frac{2\,(Ba^2b+2\,Cab^2-Bb^3)\log(|b\tan(dx+c)+a|)}{a^4b+2\,a^2b^3+b^5}-\frac{2\,(Ba^2b^2\tan}{}}{2\,d}$$

```
[In] integrate((B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x, algorithm="gi
ac")
```

```
[Out] -1/2*(2*(C*a^2 - 2*B*a*b - C*b^2)*(d*x + c)/(a^4 + 2*a^2*b^2 + b^4) - (B*a^
2 + 2*C*a*b - B*b^2)*log(tan(d*x + c)^2 + 1)/(a^4 + 2*a^2*b^2 + b^4) + 2*(B
*a^2*b + 2*C*a*b^2 - B*b^3)*log(abs(b*tan(d*x + c) + a))/(a^4*b + 2*a^2*b^3
 + b^5) - 2*(B*a^2*b^2*tan(d*x + c) + 2*C*a*b^3*tan(d*x + c) - B*b^4*tan(d*
x + c) - C*a^4 + 2*B*a^3*b + C*a^2*b^2)/((a^4*b + 2*a^2*b^3 + b^5)*(b*tan(d
*x + c) + a)))/d
```

Mupad [B] (verification not implemented)

Time = 8.72 (sec) , antiderivative size = 163, normalized size of antiderivative = 1.42

$$\int \frac{B\tan(c+dx)+C\tan^2(c+dx)}{(a+b\tan(c+dx))^2}\,dx = \frac{a\,(B\,b-C\,a)}{b\,d\,\left(a^2+b^2\right)\,\left(a+b\tan\left(c+d\,x\right)\right)} + \frac{\ln\left(\tan(c+d\,x)-\mathrm{i}\right)\,\left(B+C\,1\mathrm{i}\right)}{2\,d\,\left(a^2+a\,b\,2\mathrm{i}-b^2\right)} + \frac{\ln\left(\tan(c+d\,x)+1\mathrm{i}\right)\,\left(C+B\,1\mathrm{i}\right)}{2\,d\,\left(a^2\,1\mathrm{i}+2\,a\,b-b^2\,1\mathrm{i}\right)} - \frac{\ln\left(a+b\tan(c+d\,x)\right)\,\left(\frac{B}{a^2+b^2}-\frac{2\,b\,(B\,b-C\,a)}{{\left(a^2+b^2\right)}^2}\right)}{d}$$

```
[In] int((B*tan(c + d*x) + C*tan(c + d*x)^2)/(a + b*tan(c + d*x))^2,x)
```

```
[Out] (log(tan(c + d*x) - 1i)*(B + C*1i))/(2*d*(a*b*2i + a^2 - b^2)) - (log(a + b
*tan(c + d*x))*(B/(a^2 + b^2) - (2*b*(B*b - C*a))/(a^2 + b^2)^2))/d + (log(
tan(c + d*x) + 1i)*(B*1i + C))/(2*d*(2*a*b + a^2*1i - b^2*1i)) + (a*(B*b -
C*a))/(b*d*(a^2 + b^2)*(a + b*tan(c + d*x)))
```

3.35 $\int \frac{\cot(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx$

Optimal result 297
Rubi [A] (verified) 297
Mathematica [C] (verified) 299
Maple [A] (verified) 299
Fricas [A] (verification not implemented) 300
Sympy [C] (verification not implemented) 300
Maxima [A] (verification not implemented) 302
Giac [B] (verification not implemented) 302
Mupad [B] (verification not implemented) 303

Optimal result

Integrand size = 38, antiderivative size = 111

$$\begin{aligned}&\int \frac{\cot(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx\\&=\frac{(a^2B-b^2B+2abC)\,x}{(a^2+b^2)^2}+\frac{(2abB-a^2C+b^2C)\log(a\cos(c+dx)+b\sin(c+dx))}{(a^2+b^2)^2\,d}\\&\quad-\frac{bB-aC}{(a^2+b^2)\,d(a+b\tan(c+dx))}\end{aligned}$$

```
[Out] (B*a^2-B*b^2+2*C*a*b)*x/(a^2+b^2)^2+(2*B*a*b-C*a^2+C*b^2)*ln(a*cos(d*x+c)+b
*sin(d*x+c))/(a^2+b^2)^2/d+(-B*b+C*a)/(a^2+b^2)/d/(a+b*tan(d*x+c))
```

Rubi [A] (verified)

Time = 0.23 (sec) , antiderivative size = 111, normalized size of antiderivative = 1.00, number of steps used = 4, number of rules used = 4, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.105, Rules used = {3713, 3610, 3612, 3611}

$$\begin{aligned}&\int \frac{\cot(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx\\&=-\frac{bB-aC}{d\,(a^2+b^2)\,(a+b\tan(c+dx))}\\&\quad+\frac{(a^2(-C)+2abB+b^2C)\log(a\cos(c+dx)+b\sin(c+dx))}{d\,(a^2+b^2)^2}+\frac{x(a^2B+2abC-b^2B)}{(a^2+b^2)^2}\end{aligned}$$

```
[In] Int[(Cot[c + d*x]*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c + d*x])
^2,x]
```

```
[Out] ((a^2*B - b^2*B + 2*a*b*C)*x)/(a^2 + b^2)^2 + ((2*a*b*B - a^2*C + b^2*C)*Lo
g[a*Cos[c + d*x] + b*Sin[c + d*x]])/((a^2 + b^2)^2*d) - (b*B - a*C)/((a^2 +
 b^2)*d*(a + b*Tan[c + d*x]))
```

Rule 3610

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(b*c - a*d)*((a + b*Tan[e + f*x])^(m + 1)/
(f*(m + 1)*(a^2 + b^2))), x] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])
^(m + 1)*Simp[a*c + b*d - (b*c - a*d)*Tan[e + f*x], x], x], x] /; FreeQ[{a,
 b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && LtQ[m, -1]
```

Rule 3611

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c/(b*f))*Log[RemoveContent[a*Cos[e + f*x] + b*Si
n[e + f*x], x]], x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3612

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[(a*c + b*d)*(x/(a^2 + b^2)), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$
\begin{aligned}
\text{integral} &= \int \frac{B + C\tan(c+dx)}{(a+b\tan(c+dx))^2}\,dx \\
&= -\frac{bB - aC}{(a^2+b^2)\,d(a+b\tan(c+dx))} + \frac{\int \frac{aB+bC-(bB-aC)\tan(c+dx)}{a+b\tan(c+dx)}\,dx}{a^2+b^2} \\
&= \frac{(a^2B - b^2B + 2abC)\,x}{(a^2+b^2)^2} - \frac{bB-aC}{(a^2+b^2)\,d(a+b\tan(c+dx))} \\
&\quad + \frac{(2abB - a^2C + b^2C)\int \frac{b-a\tan(c+dx)}{a+b\tan(c+dx)}\,dx}{(a^2+b^2)^2}
\end{aligned}
$$

$$= \frac{(a^2B - b^2B + 2abC)\,x}{(a^2+b^2)^2} + \frac{(2abB - a^2C + b^2C)\log(a\cos(c+dx) + b\sin(c+dx))}{(a^2+b^2)^2\,d} - \frac{bB - aC}{(a^2+b^2)\,d(a + b\tan(c+dx))}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 2.45 (sec) , antiderivative size = 190, normalized size of antiderivative = 1.71

$$\int \frac{\cot(c+dx)\left(B\tan(c+dx) + C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx$$

$$= \frac{\frac{C((-ia-b)\log(i-\tan(c+dx)) + i(a+ib)\log(i+\tan(c+dx)) + 2b\log(a+b\tan(c+dx)))}{a^2+b^2} - (bB-aC)\left(\frac{i\log(i-\tan(c+dx))}{(a+ib)^2} - \frac{i\log(i+\tan(c+}{(a-ib)^2}\right.}{2bd}$$

```
[In] Integrate[(Cot[c + d*x]*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c +
 d*x])^2,x]
```

```
[Out] ((C*(((-I)*a - b)*Log[I - Tan[c + d*x]] + I*(a + I*b)*Log[I + Tan[c + d*x]]
 + 2*b*Log[a + b*Tan[c + d*x]]))/(a^2 + b^2) - (b*B - a*C)*((I*Log[I - Tan[
c + d*x]])/(a + I*b)^2 - (I*Log[I + Tan[c + d*x]])/(a - I*b)^2 + (2*b*(-2*a
*Log[a + b*Tan[c + d*x]] + (a^2 + b^2)/(a + b*Tan[c + d*x])))/(a^2 + b^2)^2
))/(2*b*d)
```

Maple [A] (verified)

Time = 0.29 (sec) , antiderivative size = 141, normalized size of antiderivative = 1.27

method	result
derivativedivides	$\frac{\frac{\frac{(-2Bab+Ca^2-Cb^2)\ln(1+\tan(dx+c)^2)}{2} + (Ba^2-Bb^2+2Cab)\arctan(\tan(dx+c))}{(a^2+b^2)^2} - \frac{Bb-Ca}{(a^2+b^2)(a+b\tan(dx+c))} + \frac{(2Bab-Ca^2+Cb}{(a^2}}{d}$
default	$\frac{\frac{\frac{(-2Bab+Ca^2-Cb^2)\ln(1+\tan(dx+c)^2)}{2} + (Ba^2-Bb^2+2Cab)\arctan(\tan(dx+c))}{(a^2+b^2)^2} - \frac{Bb-Ca}{(a^2+b^2)(a+b\tan(dx+c))} + \frac{(2Bab-Ca^2+Cb}{(a^2}}{d}$
parallelrisch	$\frac{2a(Bab-\frac{1}{2}Ca^2+\frac{1}{2}Cb^2)(a+b\tan(dx+c))\ln(a+b\tan(dx+c)) - a(Bab-\frac{1}{2}Ca^2+\frac{1}{2}Cb^2)(a+b\tan(dx+c))\ln(\sec(dx+c)^2}{(a+b\tan(dx+c))da(a^2+b^2}$
norman	$\frac{\frac{a(Ba^2-Bb^2+2Cab)x}{a^4+2a^2b^2+b^4} + \frac{b(Ba^2-Bb^2+2Cab)x\tan(dx+c)}{a^4+2a^2b^2+b^4} + \frac{(Bb-Ca)b\tan(dx+c)}{ad(a^2+b^2)}}{a+b\tan(dx+c)} + \frac{(2Bab-Ca^2+Cb^2)\ln(a+b\tan(dx+c)}{d(a^4+2a^2b^2+b^4)}$
risch	$-\frac{xB}{2iba-a^2+b^2} + \frac{ixC}{2iba-a^2+b^2} - \frac{4iabBx}{a^4+2a^2b^2+b^4} + \frac{2ia^2Cx}{a^4+2a^2b^2+b^4} - \frac{2iCb^2x}{a^4+2a^2b^2+b^4} - \frac{4iabBc}{d(a^4+2a^2b^2+b^4)} + \frac{}{d(a^4}$

```
[In] int(cot(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x,method=_R
ETURNVERBOSE)
```

```
[Out] 1/d*(1/(a^2+b^2)^2*(1/2*(-2*B*a*b+C*a^2-C*b^2)*ln(1+tan(d*x+c)^2)+(B*a^2-B*
b^2+2*C*a*b)*arctan(tan(d*x+c)))-(B*b-C*a)/(a^2+b^2)/(a+b*tan(d*x+c))+(2*B*
a*b-C*a^2+C*b^2)/(a^2+b^2)^2*ln(a+b*tan(d*x+c)))
```

Fricas [A] (verification not implemented)

none

Time = 0.28 (sec) , antiderivative size = 222, normalized size of antiderivative = 2.00

$$\int \frac{\cot(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx = \frac{2\,Cab^2-2\,Bb^3+2\,(Ba^3+2\,Ca^2b-Bab^2)dx-(Ca^3-2\,Ba^2b-Cab^2+(Ca^2b-2\,Bab^2-Cb^3)\tan\left(dx\right.}{2\left((a^4b+2\,a^2b^3+b^5)d\tan\left(dx+c\right.\right.}$$

```
[In] integrate(cot(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x, al
gorithm="fricas")
```

```
[Out] 1/2*(2*C*a*b^2 - 2*B*b^3 + 2*(B*a^3 + 2*C*a^2*b - B*a*b^2)*d*x - (C*a^3 - 2
*B*a^2*b - C*a*b^2 + (C*a^2*b - 2*B*a*b^2 - C*b^3)*tan(d*x + c))*log((b^2*t
an(d*x + c)^2 + 2*a*b*tan(d*x + c) + a^2)/(tan(d*x + c)^2 + 1)) - 2*(C*a^2*
b - B*a*b^2 - (B*a^2*b + 2*C*a*b^2 - B*b^3)*d*x)*tan(d*x + c))/((a^4*b + 2*
a^2*b^3 + b^5)*d*tan(d*x + c) + (a^5 + 2*a^3*b^2 + a*b^4)*d)
```

Sympy [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 2.07 (sec) , antiderivative size = 2895, normalized size of antiderivative = 26.08

$$\int \frac{\cot(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx = \text{Too large to display}$$

```
[In] integrate(cot(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c))**2,x)
```

```
[Out] Piecewise((zoo*x*(B*tan(c) + C*tan(c)**2)*cot(c)/tan(c)**2, Eq(a, 0) & Eq(b
, 0) & Eq(d, 0)), ((B*x + C*log(tan(c + d*x)**2 + 1)/(2*d))/a**2, Eq(b, 0))
, (-B*d*x*tan(c + d*x)**2/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*
x) - 4*b**2*d) + 2*I*B*d*x*tan(c + d*x)/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**
2*d*tan(c + d*x) - 4*b**2*d) + B*d*x/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d
*tan(c + d*x) - 4*b**2*d) - B*tan(c + d*x)/(4*b**2*d*tan(c + d*x)**2 - 8*I*
b**2*d*tan(c + d*x) - 4*b**2*d) + 2*I*B/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**
2*d*tan(c + d*x) - 4*b**2*d) + I*C*d*x*tan(c + d*x)**2/(4*b**2*d*tan(c + d*
```

x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + 2*C*d*x*tan(c + d*x)/(4*b**2*
d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) - I*C*d*x/(4*b**2*d
*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + I*C*tan(c + d*x)/(
4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d), Eq(a, -I*b)
), (-B*d*x*tan(c + d*x)**2/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d
*x) - 4*b**2*d) - 2*I*B*d*x*tan(c + d*x)/(4*b**2*d*tan(c + d*x)**2 + 8*I*b*
*2*d*tan(c + d*x) - 4*b**2*d) + B*d*x/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*
d*tan(c + d*x) - 4*b**2*d) - B*tan(c + d*x)/(4*b**2*d*tan(c + d*x)**2 + 8*I
*b**2*d*tan(c + d*x) - 4*b**2*d) - 2*I*B/(4*b**2*d*tan(c + d*x)**2 + 8*I*b*
*2*d*tan(c + d*x) - 4*b**2*d) - I*C*d*x*tan(c + d*x)**2/(4*b**2*d*tan(c + d
*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + 2*C*d*x*tan(c + d*x)/(4*b**2
*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + I*C*d*x/(4*b**2*
d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) - I*C*tan(c + d*x)/
(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d), Eq(a, I*b)
), (x*(B*tan(c) + C*tan(c)**2)*cot(c)/(a + b*tan(c))**2, Eq(d, 0)), (2*B*a*
*3*d*x/(2*a**5*d + 2*a**4*b*d*tan(c + d*x) + 4*a**3*b**2*d + 4*a**2*b**3*d*
tan(c + d*x) + 2*a*b**4*d + 2*b**5*d*tan(c + d*x)) + 2*B*a**2*b*d*x*tan(c +
 d*x)/(2*a**5*d + 2*a**4*b*d*tan(c + d*x) + 4*a**3*b**2*d + 4*a**2*b**3*d*t
an(c + d*x) + 2*a*b**4*d + 2*b**5*d*tan(c + d*x)) + 4*B*a**2*b*log(a/b + ta
n(c + d*x))/(2*a**5*d + 2*a**4*b*d*tan(c + d*x) + 4*a**3*b**2*d + 4*a**2*b*
*3*d*tan(c + d*x) + 2*a*b**4*d + 2*b**5*d*tan(c + d*x)) - 2*B*a**2*b*log(ta
n(c + d*x)**2 + 1)/(2*a**5*d + 2*a**4*b*d*tan(c + d*x) + 4*a**3*b**2*d + 4*
a**2*b**3*d*tan(c + d*x) + 2*a*b**4*d + 2*b**5*d*tan(c + d*x)) - 2*B*a**2*b
/(2*a**5*d + 2*a**4*b*d*tan(c + d*x) + 4*a**3*b**2*d + 4*a**2*b**3*d*tan(c
+ d*x) + 2*a*b**4*d + 2*b**5*d*tan(c + d*x)) - 2*B*a*b**2*d*x/(2*a**5*d + 2
*a**4*b*d*tan(c + d*x) + 4*a**3*b**2*d + 4*a**2*b**3*d*tan(c + d*x) + 2*a*b
4*d + 2*b5*d*tan(c + d*x)) + 4*B*a*b**2*log(a/b + tan(c + d*x))*tan(c +
 d*x)/(2*a**5*d + 2*a**4*b*d*tan(c + d*x) + 4*a**3*b**2*d + 4*a**2*b**3*d*t
an(c + d*x) + 2*a*b**4*d + 2*b**5*d*tan(c + d*x)) - 2*B*a*b**2*log(tan(c +
d*x)**2 + 1)*tan(c + d*x)/(2*a**5*d + 2*a**4*b*d*tan(c + d*x) + 4*a**3*b**2
*d + 4*a**2*b**3*d*tan(c + d*x) + 2*a*b**4*d + 2*b**5*d*tan(c + d*x)) - 2*B
*b**3*d*x*tan(c + d*x)/(2*a**5*d + 2*a**4*b*d*tan(c + d*x) + 4*a**3*b**2*d
+ 4*a**2*b**3*d*tan(c + d*x) + 2*a*b**4*d + 2*b**5*d*tan(c + d*x)) - 2*B*b*
*3/(2*a**5*d + 2*a**4*b*d*tan(c + d*x) + 4*a**3*b**2*d + 4*a**2*b**3*d*tan(
c + d*x) + 2*a*b**4*d + 2*b**5*d*tan(c + d*x)) - 2*C*a**3*log(a/b + tan(c +
 d*x))/(2*a**5*d + 2*a**4*b*d*tan(c + d*x) + 4*a**3*b**2*d + 4*a**2*b**3*d*
tan(c + d*x) + 2*a*b**4*d + 2*b**5*d*tan(c + d*x)) + C*a**3*log(tan(c + d*x
)**2 + 1)/(2*a**5*d + 2*a**4*b*d*tan(c + d*x) + 4*a**3*b**2*d + 4*a**2*b**3
*d*tan(c + d*x) + 2*a*b**4*d + 2*b**5*d*tan(c + d*x)) + 2*C*a**3/(2*a**5*d
+ 2*a**4*b*d*tan(c + d*x) + 4*a**3*b**2*d + 4*a**2*b**3*d*tan(c + d*x) + 2*
a*b**4*d + 2*b**5*d*tan(c + d*x)) + 4*C*a**2*b*d*x/(2*a**5*d + 2*a**4*b*d*t
an(c + d*x) + 4*a**3*b**2*d + 4*a**2*b**3*d*tan(c + d*x) + 2*a*b**4*d + 2*b
5*d*tan(c + d*x)) - 2*C*a2*b*log(a/b + tan(c + d*x))*tan(c + d*x)/(2*a*
*5*d + 2*a**4*b*d*tan(c + d*x) + 4*a**3*b**2*d + 4*a**2*b**3*d*tan(c + d*x)
 + 2*a*b**4*d + 2*b**5*d*tan(c + d*x)) + C*a**2*b*log(tan(c + d*x)**2 + 1)*

```
tan(c + d*x)/(2*a**5*d + 2*a**4*b*d*tan(c + d*x) + 4*a**3*b**2*d + 4*a**2*b
**3*d*tan(c + d*x) + 2*a*b**4*d + 2*b**5*d*tan(c + d*x)) + 4*C*a*b**2*d*x*t
an(c + d*x)/(2*a**5*d + 2*a**4*b*d*tan(c + d*x) + 4*a**3*b**2*d + 4*a**2*b*
*3*d*tan(c + d*x) + 2*a*b**4*d + 2*b**5*d*tan(c + d*x)) + 2*C*a*b**2*log(a/
b + tan(c + d*x))/(2*a**5*d + 2*a**4*b*d*tan(c + d*x) + 4*a**3*b**2*d + 4*a
**2*b**3*d*tan(c + d*x) + 2*a*b**4*d + 2*b**5*d*tan(c + d*x)) - C*a*b**2*lo
g(tan(c + d*x)**2 + 1)/(2*a**5*d + 2*a**4*b*d*tan(c + d*x) + 4*a**3*b**2*d
+ 4*a**2*b**3*d*tan(c + d*x) + 2*a*b**4*d + 2*b**5*d*tan(c + d*x)) + 2*C*a*
b**2/(2*a**5*d + 2*a**4*b*d*tan(c + d*x) + 4*a**3*b**2*d + 4*a**2*b**3*d*ta
n(c + d*x) + 2*a*b**4*d + 2*b**5*d*tan(c + d*x)) + 2*C*b**3*log(a/b + tan(c
 + d*x))*tan(c + d*x)/(2*a**5*d + 2*a**4*b*d*tan(c + d*x) + 4*a**3*b**2*d +
 4*a**2*b**3*d*tan(c + d*x) + 2*a*b**4*d + 2*b**5*d*tan(c + d*x)) - C*b**3*
log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(2*a**5*d + 2*a**4*b*d*tan(c + d*x) +
 4*a**3*b**2*d + 4*a**2*b**3*d*tan(c + d*x) + 2*a*b**4*d + 2*b**5*d*tan(c +
 d*x)), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.37 (sec) , antiderivative size = 177, normalized size of antiderivative = 1.59

$$\int \frac{\cot(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx = \frac{\frac{2\,(Ba^2+2\,Cab-Bb^2)(dx+c)}{a^4+2\,a^2b^2+b^4} - \frac{2\,(Ca^2-2\,Bab-Cb^2)\log(b\tan(dx+c)+a)}{a^4+2\,a^2b^2+b^4} + \frac{(Ca^2-2\,Bab-Cb^2)\log\left(\tan(dx+c)^2+1\right)}{a^4+2\,a^2b^2+b^4} + \frac{2\,(Ca-Bb)}{a^3+ab^2+(a^2b+b^3)\,\mathrm{ta}}}{2\,d}$$

```
[In] integrate(cot(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x, al
gorithm="maxima")
```

```
[Out] 1/2*(2*(B*a^2 + 2*C*a*b - B*b^2)*(d*x + c)/(a^4 + 2*a^2*b^2 + b^4) - 2*(C*a
^2 - 2*B*a*b - C*b^2)*log(b*tan(d*x + c) + a)/(a^4 + 2*a^2*b^2 + b^4) + (C*
a^2 - 2*B*a*b - C*b^2)*log(tan(d*x + c)^2 + 1)/(a^4 + 2*a^2*b^2 + b^4) + 2*
(C*a - B*b)/(a^3 + a*b^2 + (a^2*b + b^3)*tan(d*x + c)))/d
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 234 vs. 2(111) = 222.

Time = 0.87 (sec) , antiderivative size = 234, normalized size of antiderivative = 2.11

$$\int \frac{\cot(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx = \frac{\frac{2\,(Ba^2+2\,Cab-Bb^2)(dx+c)}{a^4+2\,a^2b^2+b^4} + \frac{(Ca^2-2\,Bab-Cb^2)\log\left(\tan(dx+c)^2+1\right)}{a^4+2\,a^2b^2+b^4} - \frac{2\,(Ca^2b-2\,Bab^2-Cb^3)\log(|b\tan(dx+c)+a|)}{a^4b+2\,a^2b^3+b^5} + \frac{2\,(Ca^2b\tan(dx+c}{}}{2\,d}$$

```
[In] integrate(cot(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x, al
gorithm="giac")
```

```
[Out] 1/2*(2*(B*a^2 + 2*C*a*b - B*b^2)*(d*x + c)/(a^4 + 2*a^2*b^2 + b^4) + (C*a^2
 - 2*B*a*b - C*b^2)*log(tan(d*x + c)^2 + 1)/(a^4 + 2*a^2*b^2 + b^4) - 2*(C*
a^2*b - 2*B*a*b^2 - C*b^3)*log(abs(b*tan(d*x + c) + a))/(a^4*b + 2*a^2*b^3
+ b^5) + 2*(C*a^2*b*tan(d*x + c) - 2*B*a*b^2*tan(d*x + c) - C*b^3*tan(d*x +
 c) + 2*C*a^3 - 3*B*a^2*b - B*b^3)/((a^4 + 2*a^2*b^2 + b^4)*(b*tan(d*x + c)
 + a)))/d
```

Mupad [B] (verification not implemented)

Time = 8.62 (sec) , antiderivative size = 153, normalized size of antiderivative = 1.38

$$\int \frac{\cot(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx = \frac{\ln\left(a+b\tan(c+d\,x)\right)\,\left(-C\,a^2+2\,B\,a\,b+C\,b^2\right)}{d\,{\left(a^2+b^2\right)}^2}-\frac{B\,b-C\,a}{d\,\left(a^2+b^2\right)\,\left(a+b\tan\left(c+d\,x\right)\right)} - \frac{\ln\left(\tan(c+d\,x)+1\mathrm{i}\right)\,\left(C+B\,1\mathrm{i}\right)}{2\,d\,\left(-a^2+a\,b\,2\mathrm{i}+b^2\right)}-\frac{\ln\left(\tan(c+d\,x)-\mathrm{i}\right)\,\left(B+C\,1\mathrm{i}\right)}{2\,d\,\left(-a^2\,1\mathrm{i}+2\,a\,b+b^2\,1\mathrm{i}\right)}$$

```
[In] int((cot(c + d*x)*(B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c + d*x))
^2,x)
```

```
[Out] (log(a + b*tan(c + d*x))*(C*b^2 - C*a^2 + 2*B*a*b))/(d*(a^2 + b^2)^2) - (B*
b - C*a)/(d*(a^2 + b^2)*(a + b*tan(c + d*x))) - (log(tan(c + d*x) + 1i)*(B*
1i + C))/(2*d*(a*b*2i - a^2 + b^2)) - (log(tan(c + d*x) - 1i)*(B + C*1i))/(
2*d*(2*a*b - a^2*1i + b^2*1i))
```

3.36 $\int \frac{\cot^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx$

Optimal result . . . 304
Rubi [A] (verified) . . . 304
Mathematica [C] (verified) . . . 306
Maple [A] (verified) . . . 307
Fricas [B] (verification not implemented) . . . 307
Sympy [C] (verification not implemented) . . . 308
Maxima [A] (verification not implemented) . . . 310
Giac [B] (verification not implemented) . . . 311
Mupad [B] (verification not implemented) . . . 311

Optimal result

Integrand size = 40, antiderivative size = 137

$$\begin{aligned}
&\int \frac{\cot^2(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^2}\,dx \\
&= -\frac{(2abB-a^2C+b^2C)\,x}{(a^2+b^2)^2}+\frac{B\log(\sin(c+dx))}{a^2d} \\
&\quad -\frac{b(3a^2bB+b^3B-2a^3C)\log(a\cos(c+dx)+b\sin(c+dx))}{a^2\,(a^2+b^2)^2\,d} \\
&\quad +\frac{b(bB-aC)}{a\,(a^2+b^2)\,d(a+b\tan(c+dx))}
\end{aligned}$$

```
[Out] -(2*B*a*b-C*a^2+C*b^2)*x/(a^2+b^2)^2+B*ln(sin(d*x+c))/a^2/d-b*(3*B*a^2*b+B*
b^3-2*C*a^3)*ln(a*cos(d*x+c)+b*sin(d*x+c))/a^2/(a^2+b^2)^2/d+b*(B*b-C*a)/a/
(a^2+b^2)/d/(a+b*tan(d*x+c))
```

Rubi [A] (verified)

Time = 0.46 (sec) , antiderivative size = 137, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.125, Rules used = {3713, 3690, 3732, 3611, 3556}

$$\begin{aligned}
&\int \frac{\cot^2(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^2}\,dx \\
&= \frac{b(bB-aC)}{ad\,(a^2+b^2)\,(a+b\tan(c+dx))}-\frac{x(a^2(-C)+2abB+b^2C)}{(a^2+b^2)^2}+\frac{B\log(\sin(c+dx))}{a^2d} \\
&\quad -\frac{b(-2a^3C+3a^2bB+b^3B)\log(a\cos(c+dx)+b\sin(c+dx))}{a^2d\,(a^2+b^2)^2}
\end{aligned}$$

```
[In] Int[(Cot[c + d*x]^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c + d*x
])^2,x]
```

```
[Out] -((((2*a*b*B - a^2*C + b^2*C)*x)/(a^2 + b^2)^2) + (B*Log[Sin[c + d*x]])/(a^2
*d) - (b*(3*a^2*b*B + b^3*B - 2*a^3*C)*Log[a*Cos[c + d*x] + b*Sin[c + d*x]]
)/(a^2*(a^2 + b^2)^2*d) + (b*(b*B - a*C))/(a*(a^2 + b^2)*d*(a + b*Tan[c + d
*x]))
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3611

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c/(b*f))*Log[RemoveContent[a*Cos[e + f*x] + b*Si
n[e + f*x], x]], x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3690

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Si
mp[b*(A*b - a*B)*(a + b*Tan[e + f*x])^(m + 1)*((c + d*Tan[e + f*x])^(n + 1)
/(f*(m + 1)*(b*c - a*d)*(a^2 + b^2))), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^
2 + b^2)), Int[(a + b*Tan[e + f*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[b*B
*(b*c*(m + 1) + a*d*(n + 1)) + A*(a*(b*c - a*d)*(m + 1) - b^2*d*(m + n + 2)
) - (A*b - a*B)*(b*c - a*d)*(m + 1)*Tan[e + f*x] - b*d*(A*b - a*B)*(m + n +
 2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, n}, x] &&
NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &
& (IntegerQ[m] || IntegersQ[2*m, 2*n]) &&  !(ILtQ[n, -1] && ( !IntegerQ[m]
|| (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rule 3732

```
Int[((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^
2)/(((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
```

```
*(x_)])), x_Symbol] :> Simp[(a*(A*c - c*C + B*d) + b*(B*c - A*d + C*d))*(x/
((a^2 + b^2)*(c^2 + d^2))), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/((b*c - a*d)
*(a^2 + b^2)), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] - Dist
[(c^2*C - B*c*d + A*d^2)/((b*c - a*d)*(c^2 + d^2)), Int[(d - c*Tan[e + f*x]
)/(c + d*Tan[e + f*x]), x], x]) /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] &&
 NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{\cot(c+dx)(B+C\tan(c+dx))}{(a+b\tan(c+dx))^2}\,dx \\
&= \frac{b(bB-aC)}{a\left(a^2+b^2\right)d(a+b\tan(c+dx))}+\frac{\int \frac{\cot(c+dx)\left(\left(a^2+b^2\right)B-a(bB-aC)\tan(c+dx)+b(bB-aC)\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx}{a\left(a^2+b^2\right)} \\
&= -\frac{\left(2abB-a^2C+b^2C\right)x}{\left(a^2+b^2\right)^2}+\frac{b(bB-aC)}{a\left(a^2+b^2\right)d(a+b\tan(c+dx))} \\
&\quad +\frac{B\int\cot(c+dx)\,dx}{a^2}-\frac{\left(b\left(3a^2bB+b^3B-2a^3C\right)\right)\int\frac{b-a\tan(c+dx)}{a+b\tan(c+dx)}\,dx}{a^2\left(a^2+b^2\right)^2} \\
&= -\frac{\left(2abB-a^2C+b^2C\right)x}{\left(a^2+b^2\right)^2}+\frac{B\log(\sin(c+dx))}{a^2d} \\
&\quad -\frac{b\left(3a^2bB+b^3B-2a^3C\right)\log(a\cos(c+dx)+b\sin(c+dx))}{a^2\left(a^2+b^2\right)^2d} \\
&\quad +\frac{b(bB-aC)}{a\left(a^2+b^2\right)d(a+b\tan(c+dx))}
\end{aligned}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 2.61 (sec) , antiderivative size = 159, normalized size of antiderivative = 1.16

$$\int \frac{\cot^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx = -\frac{\frac{(B+iC)\log(i-\tan(c+dx))}{(a+ib)^2}-\frac{2B\log(\tan(c+dx))}{a^2}+\frac{(B-iC)\log(i+\tan(c+dx))}{(a-ib)^2}+\frac{2b\left(3a^2bB+b^3B-2a^3C\right)\log(a+b\tan(c+dx))}{a^2\left(a^2+b^2\right)^2}+\frac{2}{a\left(a^2+b\right.}}{2d}$$

```
[In] Integrate[(Cot[c + d*x]^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c
 + d*x])^2,x]
```

```
[Out] -1/2*(((B + I*C)*Log[I - Tan[c + d*x]])/(a + I*b)^2 - (2*B*Log[Tan[c + d*x]
])/a^2 + ((B - I*C)*Log[I + Tan[c + d*x]])/(a - I*b)^2 + (2*b*(3*a^2*b*B +
b^3*B - 2*a^3*C)*Log[a + b*Tan[c + d*x]])/(a^2*(a^2 + b^2)^2) + (2*b*(-(b*B
) + a*C))/(a*(a^2 + b^2)*(a + b*Tan[c + d*x])))/d
```

Maple [A] (verified)

Time = 0.40 (sec) , antiderivative size = 163, normalized size of antiderivative = 1.19

method	result
derivativedivides	$\frac{\frac{B\ln(\tan(dx+c))}{a^2}+\frac{\frac{\left(-Ba^2+Bb^2-2Cab\right)\ln\left(1+\tan(dx+c)^2\right)}{2}+\left(-2Bab+Ca^2-Cb^2\right)\arctan(\tan(dx+c))}{\left(a^2+b^2\right)^2}-\frac{b\left(3Ba^2b+Bb^3-2Ca^3\right)\ln}{\left(a^2+b^2\right)^2}}{d}$
default	$\frac{\frac{B\ln(\tan(dx+c))}{a^2}+\frac{\frac{\left(-Ba^2+Bb^2-2Cab\right)\ln\left(1+\tan(dx+c)^2\right)}{2}+\left(-2Bab+Ca^2-Cb^2\right)\arctan(\tan(dx+c))}{\left(a^2+b^2\right)^2}-\frac{b\left(3Ba^2b+Bb^3-2Ca^3\right)\ln}{\left(a^2+b^2\right)^2}}{d}$
parallelrisch	$-6\left(Ba^2b+\frac{1}{3}Bb^3-\frac{2}{3}Ca^3\right)b(a+b\tan(dx+c))\ln(a+b\tan(dx+c))-a^2(a+b\tan(dx+c))\left(Ba^2-Bb^2+2Cab\right)\ln\left(\sec(dx+c\right.$
norman	$\frac{-\frac{a\left(2Bab-Ca^2+Cb^2\right)x\tan(dx+c)}{a^4+2a^2b^2+b^4}-\frac{b\left(2Bab-Ca^2+Cb^2\right)x\tan(dx+c)^2}{a^4+2a^2b^2+b^4}-\frac{\left(Bb^2-Cab\right)b\tan(dx+c)^2}{da^2\left(a^2+b^2\right)}}{\tan(dx+c)(a+b\tan(dx+c))}+\frac{B\ln(\tan(dx+c))}{a^2d}-$
risch	$-\frac{4iCabx}{a^4+2a^2b^2+b^4}-\frac{xC}{2iba-a^2+b^2}-\frac{ixB}{2iba-a^2+b^2}+\frac{2ib^4Bx}{(a^4+2a^2b^2+b^4)a^2}+\frac{2ib^4Bc}{(a^4+2a^2b^2+b^4)a^2d}-\frac{2iBc}{a^2d}-\frac{4i}{d(a^4+}$

```
[In] int(cot(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x,method=
_RETURNVERBOSE)
```

```
[Out] 1/d*(1/a^2*B*ln(tan(d*x+c))+1/(a^2+b^2)^2*(1/2*(-B*a^2+B*b^2-2*C*a*b)*ln(1+
tan(d*x+c)^2)+(-2*B*a*b+C*a^2-C*b^2)*arctan(tan(d*x+c)))-b*(3*B*a^2*b+B*b^3
-2*C*a^3)/(a^2+b^2)^2/a^2*ln(a+b*tan(d*x+c))+(B*b-C*a)*b/(a^2+b^2)/a/(a+b*t
an(d*x+c)))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 323 vs. $2(137) = 274$.

Time = 0.31 (sec) , antiderivative size = 323, normalized size of antiderivative = 2.36

$$\int \frac{\cot^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx =$$

$$-\frac{2Ca^2b^3-2Bab^4-2(Ca^5-2Ba^4b-Ca^3b^2)dx-(Ba^5+2Ba^3b^2+Bab^4+(Ba^4b+2Ba^2b^3+Bb^5)}{}$$

```
[In] integrate(cot(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x,
algorithm="fricas")
```

```
[Out] -1/2*(2*C*a^2*b^3 - 2*B*a*b^4 - 2*(C*a^5 - 2*B*a^4*b - C*a^3*b^2)*d*x - (B*
a^5 + 2*B*a^3*b^2 + B*a*b^4 + (B*a^4*b + 2*B*a^2*b^3 + B*b^5)*tan(d*x + c))
*log(tan(d*x + c)^2/(tan(d*x + c)^2 + 1)) - (2*C*a^4*b - 3*B*a^3*b^2 - B*a*
b^4 + (2*C*a^3*b^2 - 3*B*a^2*b^3 - B*b^5)*tan(d*x + c))*log((b^2*tan(d*x +
```

```
c)^2 + 2*a*b*tan(d*x + c) + a^2)/(tan(d*x + c)^2 + 1)) - 2*(C*a^3*b^2 - B*a
^2*b^3 + (C*a^4*b - 2*B*a^3*b^2 - C*a^2*b^3)*d*x)*tan(d*x + c))/((a^6*b + 2
*a^4*b^3 + a^2*b^5)*d*tan(d*x + c) + (a^7 + 2*a^5*b^2 + a^3*b^4)*d)
```

Sympy [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 3.79 (sec) , antiderivative size = 4502, normalized size of antiderivative = 32.86

$$\int \frac{\cot^2(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^2}\,dx = \text{Too large to display}$$

```
[In] integrate(cot(d*x+c)**2*(B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c))**2,
x)
```

```
[Out] Piecewise((zoo*x*(B*tan(c) + C*tan(c)**2)*cot(c)**2/tan(c)**2, Eq(a, 0) & E
q(b, 0) & Eq(d, 0)), ((-B*log(tan(c + d*x)**2 + 1)/(2*d) + B*log(tan(c + d*
x))/d + C*x)/a**2, Eq(b, 0)), ((B*log(tan(c + d*x)**2 + 1)/(2*d) - B*log(ta
n(c + d*x))/d - B/(2*d*tan(c + d*x)**2) - C*x - C/(d*tan(c + d*x)))/b**2, E
q(a, 0)), (3*I*B*d*x*tan(c + d*x)**2/(4*a**2*d*tan(c + d*x)**2 + 8*I*a**2*d
*tan(c + d*x) - 4*a**2*d) - 6*B*d*x*tan(c + d*x)/(4*a**2*d*tan(c + d*x)**2
+ 8*I*a**2*d*tan(c + d*x) - 4*a**2*d) - 3*I*B*d*x/(4*a**2*d*tan(c + d*x)**2
 + 8*I*a**2*d*tan(c + d*x) - 4*a**2*d) - 2*B*log(tan(c + d*x)**2 + 1)*tan(c
 + d*x)**2/(4*a**2*d*tan(c + d*x)**2 + 8*I*a**2*d*tan(c + d*x) - 4*a**2*d)
- 4*I*B*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(4*a**2*d*tan(c + d*x)**2 + 8
*I*a**2*d*tan(c + d*x) - 4*a**2*d) + 2*B*log(tan(c + d*x)**2 + 1)/(4*a**2*d
*tan(c + d*x)**2 + 8*I*a**2*d*tan(c + d*x) - 4*a**2*d) + 4*B*log(tan(c + d*
x))*tan(c + d*x)**2/(4*a**2*d*tan(c + d*x)**2 + 8*I*a**2*d*tan(c + d*x) - 4
*a**2*d) + 8*I*B*log(tan(c + d*x))*tan(c + d*x)/(4*a**2*d*tan(c + d*x)**2 +
 8*I*a**2*d*tan(c + d*x) - 4*a**2*d) - 4*B*log(tan(c + d*x))/(4*a**2*d*tan(
c + d*x)**2 + 8*I*a**2*d*tan(c + d*x) - 4*a**2*d) + 3*I*B*tan(c + d*x)/(4*a
**2*d*tan(c + d*x)**2 + 8*I*a**2*d*tan(c + d*x) - 4*a**2*d) - 4*B/(4*a**2*d
*tan(c + d*x)**2 + 8*I*a**2*d*tan(c + d*x) - 4*a**2*d) + C*d*x*tan(c + d*x)
**2/(4*a**2*d*tan(c + d*x)**2 + 8*I*a**2*d*tan(c + d*x) - 4*a**2*d) + 2*I*C
*d*x*tan(c + d*x)/(4*a**2*d*tan(c + d*x)**2 + 8*I*a**2*d*tan(c + d*x) - 4*a
**2*d) - C*d*x/(4*a**2*d*tan(c + d*x)**2 + 8*I*a**2*d*tan(c + d*x) - 4*a**2
*d) + C*tan(c + d*x)/(4*a**2*d*tan(c + d*x)**2 + 8*I*a**2*d*tan(c + d*x) -
4*a**2*d) + 2*I*C/(4*a**2*d*tan(c + d*x)**2 + 8*I*a**2*d*tan(c + d*x) - 4*a
**2*d), Eq(b, -I*a)), (-3*I*B*d*x*tan(c + d*x)**2/(4*a**2*d*tan(c + d*x)**2
 - 8*I*a**2*d*tan(c + d*x) - 4*a**2*d) - 6*B*d*x*tan(c + d*x)/(4*a**2*d*tan
(c + d*x)**2 - 8*I*a**2*d*tan(c + d*x) - 4*a**2*d) + 3*I*B*d*x/(4*a**2*d*ta
n(c + d*x)**2 - 8*I*a**2*d*tan(c + d*x) - 4*a**2*d) - 2*B*log(tan(c + d*x)*
*2 + 1)*tan(c + d*x)**2/(4*a**2*d*tan(c + d*x)**2 - 8*I*a**2*d*tan(c + d*x)
 - 4*a**2*d) + 4*I*B*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(4*a**2*d*tan(c
+ d*x)**2 - 8*I*a**2*d*tan(c + d*x) - 4*a**2*d) + 2*B*log(tan(c + d*x)**2 +
```

```
 1)/(4*a**2*d*tan(c + d*x)**2 - 8*I*a**2*d*tan(c + d*x) - 4*a**2*d) + 4*B*l
og(tan(c + d*x))*tan(c + d*x)**2/(4*a**2*d*tan(c + d*x)**2 - 8*I*a**2*d*tan
(c + d*x) - 4*a**2*d) - 8*I*B*log(tan(c + d*x))*tan(c + d*x)/(4*a**2*d*tan(
c + d*x)**2 - 8*I*a**2*d*tan(c + d*x) - 4*a**2*d) - 4*B*log(tan(c + d*x))/(
4*a**2*d*tan(c + d*x)**2 - 8*I*a**2*d*tan(c + d*x) - 4*a**2*d) - 3*I*B*tan(
c + d*x)/(4*a**2*d*tan(c + d*x)**2 - 8*I*a**2*d*tan(c + d*x) - 4*a**2*d) -
4*B/(4*a**2*d*tan(c + d*x)**2 - 8*I*a**2*d*tan(c + d*x) - 4*a**2*d) + C*d*x
*tan(c + d*x)**2/(4*a**2*d*tan(c + d*x)**2 - 8*I*a**2*d*tan(c + d*x) - 4*a*
*2*d) - 2*I*C*d*x*tan(c + d*x)/(4*a**2*d*tan(c + d*x)**2 - 8*I*a**2*d*tan(c
 + d*x) - 4*a**2*d) - C*d*x/(4*a**2*d*tan(c + d*x)**2 - 8*I*a**2*d*tan(c +
d*x) - 4*a**2*d) + C*tan(c + d*x)/(4*a**2*d*tan(c + d*x)**2 - 8*I*a**2*d*ta
n(c + d*x) - 4*a**2*d) - 2*I*C/(4*a**2*d*tan(c + d*x)**2 - 8*I*a**2*d*tan(c
 + d*x) - 4*a**2*d), Eq(b, I*a)), (zoo*(-B*log(tan(c + d*x)**2 + 1)/(2*d) +
 B*log(tan(c + d*x))/d + C*x), Eq(b, -a/tan(c + d*x))), (x*(B*tan(c) + C*ta
n(c)**2)*cot(c)**2/(a + b*tan(c))**2, Eq(d, 0)), (-B*a**5*log(tan(c + d*x)*
*2 + 1)/(2*a**7*d + 2*a**6*b*d*tan(c + d*x) + 4*a**5*b**2*d + 4*a**4*b**3*d
*tan(c + d*x) + 2*a**3*b**4*d + 2*a**2*b**5*d*tan(c + d*x)) + 2*B*a**5*log(
tan(c + d*x))/(2*a**7*d + 2*a**6*b*d*tan(c + d*x) + 4*a**5*b**2*d + 4*a**4*
b**3*d*tan(c + d*x) + 2*a**3*b**4*d + 2*a**2*b**5*d*tan(c + d*x)) - 4*B*a**
4*b*d*x/(2*a**7*d + 2*a**6*b*d*tan(c + d*x) + 4*a**5*b**2*d + 4*a**4*b**3*d
*tan(c + d*x) + 2*a**3*b**4*d + 2*a**2*b**5*d*tan(c + d*x)) - B*a**4*b*log(
tan(c + d*x)**2 + 1)*tan(c + d*x)/(2*a**7*d + 2*a**6*b*d*tan(c + d*x) + 4*a
**5*b**2*d + 4*a**4*b**3*d*tan(c + d*x) + 2*a**3*b**4*d + 2*a**2*b**5*d*tan
(c + d*x)) + 2*B*a**4*b*log(tan(c + d*x))*tan(c + d*x)/(2*a**7*d + 2*a**6*b
*d*tan(c + d*x) + 4*a**5*b**2*d + 4*a**4*b**3*d*tan(c + d*x) + 2*a**3*b**4*
d + 2*a**2*b**5*d*tan(c + d*x)) - 4*B*a**3*b**2*d*x*tan(c + d*x)/(2*a**7*d
+ 2*a**6*b*d*tan(c + d*x) + 4*a**5*b**2*d + 4*a**4*b**3*d*tan(c + d*x) + 2*
a**3*b**4*d + 2*a**2*b**5*d*tan(c + d*x)) - 6*B*a**3*b**2*log(a/b + tan(c +
 d*x))/(2*a**7*d + 2*a**6*b*d*tan(c + d*x) + 4*a**5*b**2*d + 4*a**4*b**3*d*
tan(c + d*x) + 2*a**3*b**4*d + 2*a**2*b**5*d*tan(c + d*x)) + B*a**3*b**2*lo
g(tan(c + d*x)**2 + 1)/(2*a**7*d + 2*a**6*b*d*tan(c + d*x) + 4*a**5*b**2*d
+ 4*a**4*b**3*d*tan(c + d*x) + 2*a**3*b**4*d + 2*a**2*b**5*d*tan(c + d*x))
+ 4*B*a**3*b**2*log(tan(c + d*x))/(2*a**7*d + 2*a**6*b*d*tan(c + d*x) + 4*a
**5*b**2*d + 4*a**4*b**3*d*tan(c + d*x) + 2*a**3*b**4*d + 2*a**2*b**5*d*tan
(c + d*x)) + 2*B*a**3*b**2/(2*a**7*d + 2*a**6*b*d*tan(c + d*x) + 4*a**5*b**
2*d + 4*a**4*b**3*d*tan(c + d*x) + 2*a**3*b**4*d + 2*a**2*b**5*d*tan(c + d*
x)) - 6*B*a**2*b**3*log(a/b + tan(c + d*x))*tan(c + d*x)/(2*a**7*d + 2*a**6
*b*d*tan(c + d*x) + 4*a**5*b**2*d + 4*a**4*b**3*d*tan(c + d*x) + 2*a**3*b**
4*d + 2*a**2*b**5*d*tan(c + d*x)) + B*a**2*b**3*log(tan(c + d*x)**2 + 1)*ta
n(c + d*x)/(2*a**7*d + 2*a**6*b*d*tan(c + d*x) + 4*a**5*b**2*d + 4*a**4*b**
3*d*tan(c + d*x) + 2*a**3*b**4*d + 2*a**2*b**5*d*tan(c + d*x)) + 4*B*a**2*b
**3*log(tan(c + d*x))*tan(c + d*x)/(2*a**7*d + 2*a**6*b*d*tan(c + d*x) + 4*
a**5*b**2*d + 4*a**4*b**3*d*tan(c + d*x) + 2*a**3*b**4*d + 2*a**2*b**5*d*ta
n(c + d*x)) - 2*B*a*b**4*log(a/b + tan(c + d*x))/(2*a**7*d + 2*a**6*b*d*tan
(c + d*x) + 4*a**5*b**2*d + 4*a**4*b**3*d*tan(c + d*x) + 2*a**3*b**4*d + 2*
```

```
a**2*b**5*d*tan(c + d*x)) + 2*B*a*b**4*log(tan(c + d*x))/(2*a**7*d + 2*a**6
*b*d*tan(c + d*x) + 4*a**5*b**2*d + 4*a**4*b**3*d*tan(c + d*x) + 2*a**3*b**
4*d + 2*a**2*b**5*d*tan(c + d*x)) + 2*B*a*b**4/(2*a**7*d + 2*a**6*b*d*tan(c
 + d*x) + 4*a**5*b**2*d + 4*a**4*b**3*d*tan(c + d*x) + 2*a**3*b**4*d + 2*a*
*2*b**5*d*tan(c + d*x)) - 2*B*b**5*log(a/b + tan(c + d*x))*tan(c + d*x)/(2*
a**7*d + 2*a**6*b*d*tan(c + d*x) + 4*a**5*b**2*d + 4*a**4*b**3*d*tan(c + d*
x) + 2*a**3*b**4*d + 2*a**2*b**5*d*tan(c + d*x)) + 2*B*b**5*log(tan(c + d*x
))*tan(c + d*x)/(2*a**7*d + 2*a**6*b*d*tan(c + d*x) + 4*a**5*b**2*d + 4*a**
4*b**3*d*tan(c + d*x) + 2*a**3*b**4*d + 2*a**2*b**5*d*tan(c + d*x)) + 2*C*a
**5*d*x/(2*a**7*d + 2*a**6*b*d*tan(c + d*x) + 4*a**5*b**2*d + 4*a**4*b**3*d
*tan(c + d*x) + 2*a**3*b**4*d + 2*a**2*b**5*d*tan(c + d*x)) + 2*C*a**4*b*d*
x*tan(c + d*x)/(2*a**7*d + 2*a**6*b*d*tan(c + d*x) + 4*a**5*b**2*d + 4*a**4
*b**3*d*tan(c + d*x) + 2*a**3*b**4*d + 2*a**2*b**5*d*tan(c + d*x)) + 4*C*a*
*4*b*log(a/b + tan(c + d*x))/(2*a**7*d + 2*a**6*b*d*tan(c + d*x) + 4*a**5*b
**2*d + 4*a**4*b**3*d*tan(c + d*x) + 2*a**3*b**4*d + 2*a**2*b**5*d*tan(c +
d*x)) - 2*C*a**4*b*log(tan(c + d*x)**2 + 1)/(2*a**7*d + 2*a**6*b*d*tan(c +
d*x) + 4*a**5*b**2*d + 4*a**4*b**3*d*tan(c + d*x) + 2*a**3*b**4*d + 2*a**2*
b**5*d*tan(c + d*x)) - 2*C*a**4*b/(2*a**7*d + 2*a**6*b*d*tan(c + d*x) + 4*a
**5*b**2*d + 4*a**4*b**3*d*tan(c + d*x) + 2*a**3*b**4*d + 2*a**2*b**5*d*tan
(c + d*x)) - 2*C*a**3*b**2*d*x/(2*a**7*d + 2*a**6*b*d*tan(c + d*x) + 4*a**5
*b**2*d + 4*a**4*b**3*d*tan(c + d*x) + 2*a**3*b**4*d + 2*a**2*b**5*d*tan(c
+ d*x)) + 4*C*a**3*b**2*log(a/b + tan(c + d*x))*tan(c + d*x)/(2*a**7*d + 2*
a**6*b*d*tan(c + d*x) + 4*a**5*b**2*d + 4*a**4*b**3*d*tan(c + d*x) + 2*a**3
*b**4*d + 2*a**2*b**5*d*tan(c + d*x)) - 2*C*a**3*b**2*log(tan(c + d*x)**2 +
 1)*tan(c + d*x)/(2*a**7*d + 2*a**6*b*d*tan(c + d*x) + 4*a**5*b**2*d + 4*a*
*4*b**3*d*tan(c + d*x) + 2*a**3*b**4*d + 2*a**2*b**5*d*tan(c + d*x)) - 2*C*
a**2*b**3*d*x*tan(c + d*x)/(2*a**7*d + 2*a**6*b*d*tan(c + d*x) + 4*a**5*b**
2*d + 4*a**4*b**3*d*tan(c + d*x) + 2*a**3*b**4*d + 2*a**2*b**5*d*tan(c + d*
x)) - 2*C*a**2*b**3/(2*a**7*d + 2*a**6*b*d*tan(c + d*x) + 4*a**5*b**2*d + 4
*a**4*b**3*d*tan(c + d*x) + 2*a**3*b**4*d + 2*a**2*b**5*d*tan(c + d*x)), Tr
ue))
```

Maxima [A] (verification not implemented)

none

Time = 0.35 (sec) , antiderivative size = 208, normalized size of antiderivative = 1.52

$$\int \frac{\cot^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx$$
$$= \frac{\frac{2\,(Ca^2-2\,Bab-Cb^2)(dx+c)}{a^4+2\,a^2b^2+b^4}+\frac{2\,(2\,Ca^3b-3\,Ba^2b^2-Bb^4)\log(b\tan(dx+c)+a)}{a^6+2\,a^4b^2+a^2b^4}-\frac{(Ba^2+2\,Cab-Bb^2)\log\left(\tan(dx+c)^2+1\right)}{a^4+2\,a^2b^2+b^4}-\frac{2\,(Cab}{a^4+a^2b^2+(a^3b+}}{2\,d}$$

```
[In] integrate(cot(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x,
algorithm="maxima")
```

```
[Out] 1/2*(2*(C*a^2 - 2*B*a*b - C*b^2)*(d*x + c)/(a^4 + 2*a^2*b^2 + b^4) + 2*(2*C
*a^3*b - 3*B*a^2*b^2 - B*b^4)*log(b*tan(d*x + c) + a)/(a^6 + 2*a^4*b^2 + a^
2*b^4) - (B*a^2 + 2*C*a*b - B*b^2)*log(tan(d*x + c)^2 + 1)/(a^4 + 2*a^2*b^2
 + b^4) - 2*(C*a*b - B*b^2)/(a^4 + a^2*b^2 + (a^3*b + a*b^3)*tan(d*x + c))
+ 2*B*log(tan(d*x + c))/a^2)/d
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 279 vs. $2(137) = 274$.

Time = 1.18 (sec) , antiderivative size = 279, normalized size of antiderivative = 2.04

$$\int \frac{\cot^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx = \frac{\frac{2\left(Ca^2-2Bab-Cb^2\right)(dx+c)}{a^4+2a^2b^2+b^4}-\frac{\left(Ba^2+2Cab-Bb^2\right)\log\left(\tan(dx+c)^2+1\right)}{a^4+2a^2b^2+b^4}+\frac{2\left(2Ca^3b^2-3Ba^2b^3-Bb^5\right)\log(|b\tan(dx+c)+a|)}{a^6b+2a^4b^3+a^2b^5}+\frac{2B\log(|\tan}{a^2}}{2d}$$

```
[In] integrate(cot(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x,
algorithm="giac")
```

```
[Out] 1/2*(2*(C*a^2 - 2*B*a*b - C*b^2)*(d*x + c)/(a^4 + 2*a^2*b^2 + b^4) - (B*a^2
 + 2*C*a*b - B*b^2)*log(tan(d*x + c)^2 + 1)/(a^4 + 2*a^2*b^2 + b^4) + 2*(2*
C*a^3*b^2 - 3*B*a^2*b^3 - B*b^5)*log(abs(b*tan(d*x + c) + a))/(a^6*b + 2*a^
4*b^3 + a^2*b^5) + 2*B*log(abs(tan(d*x + c)))/a^2 - 2*(2*C*a^3*b^2*tan(d*x
+ c) - 3*B*a^2*b^3*tan(d*x + c) - B*b^5*tan(d*x + c) + 3*C*a^4*b - 4*B*a^3*
b^2 + C*a^2*b^3 - 2*B*a*b^4)/((a^6 + 2*a^4*b^2 + a^2*b^4)*(b*tan(d*x + c) +
 a)))/d
```

Mupad [B] (verification not implemented)

Time = 10.18 (sec) , antiderivative size = 180, normalized size of antiderivative = 1.31

$$\begin{aligned}\int \frac{\cot^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx &= \frac{B\,\ln\left(\tan(c+d\,x)\right)}{a^2\,d}-\frac{\ln\left(\tan(c+d\,x)-\mathrm{i}\right)\,\left(B+C\,1\mathrm{i}\right)}{2\,d\,\left(a^2+a\,b\,2\mathrm{i}-b^2\right)}\\ &\quad-\frac{\ln\left(\tan(c+d\,x)+1\mathrm{i}\right)\,\left(C+B\,1\mathrm{i}\right)}{2\,d\,\left(a^2\,1\mathrm{i}+2\,a\,b-b^2\,1\mathrm{i}\right)}+\frac{B\,b^2-C\,a\,b}{a\,d\,\left(a^2+b^2\right)\,\left(a+b\tan\left(c+d\,x\right)\right)}\\ &\quad-\frac{b\,\ln\left(a+b\tan(c+d\,x)\right)\,\left(-2\,C\,a^3+3\,B\,a^2\,b+B\,b^3\right)}{a^2\,d\,{\left(a^2+b^2\right)}^2}\end{aligned}$$

```
[In] int((cot(c + d*x)^2*(B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c + d*x
))^2,x)
```

```
[Out] (B*log(tan(c + d*x)))/(a^2*d) - (log(tan(c + d*x) - 1i)*(B + C*1i))/(2*d*(a
*b*2i + a^2 - b^2)) - (log(tan(c + d*x) + 1i)*(B*1i + C))/(2*d*(2*a*b + a^2
*1i - b^2*1i)) + (B*b^2 - C*a*b)/(a*d*(a^2 + b^2)*(a + b*tan(c + d*x))) - (
b*log(a + b*tan(c + d*x))*(B*b^3 - 2*C*a^3 + 3*B*a^2*b))/(a^2*d*(a^2 + b^2)
^2)
```

3.37 $\int \frac{\cot^3(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx$

Optimal result 313
Rubi [A] (verified) 313
Mathematica [C] (verified) 316
Maple [A] (verified) 317
Fricas [B] (verification not implemented) 317
Sympy [C] (verification not implemented) 318
Maxima [A] (verification not implemented) 322
Giac [A] (verification not implemented) 323
Mupad [B] (verification not implemented) 323

Optimal result

Integrand size = 40, antiderivative size = 192

$$\begin{aligned}
&\int \frac{\cot^3(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx \\
&= -\frac{(a^2B-b^2B+2abC)\,x}{(a^2+b^2)^2}-\frac{(2bB-aC)\log(\sin(c+dx))}{a^3d} \\
&\quad+\frac{b^2(4a^2bB+2b^3B-3a^3C-ab^2C)\log(a\cos(c+dx)+b\sin(c+dx))}{a^3\left(a^2+b^2\right)^2d} \\
&\quad-\frac{b(a^2B+2b^2B-abC)}{a^2\left(a^2+b^2\right)d(a+b\tan(c+dx))}-\frac{B\cot(c+dx)}{ad(a+b\tan(c+dx))}
\end{aligned}$$

```
[Out] -(B*a^2-B*b^2+2*C*a*b)*x/(a^2+b^2)^2-(2*B*b-C*a)*ln(sin(d*x+c))/a^3/d+b^2*(
4*B*a^2*b+2*B*b^3-3*C*a^3-C*a*b^2)*ln(a*cos(d*x+c)+b*sin(d*x+c))/a^3/(a^2+b
^2)^2/d-b*(B*a^2+2*B*b^2-C*a*b)/a^2/(a^2+b^2)/d/(a+b*tan(d*x+c))-B*cot(d*x+
c)/a/d/(a+b*tan(d*x+c))
```

Rubi [A] (verified)

Time = 0.72 (sec) , antiderivative size = 192, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.150, Rules used

$= \{3713, 3690, 3730, 3732, 3611, 3556\}$

$$\int \frac{\cot^3(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx$$
$$= -\frac{(2bB-aC)\log(\sin(c+dx))}{a^3d} - \frac{b(a^2B-abC+2b^2B)}{a^2d\,(a^2+b^2)\,(a+b\tan(c+dx))} - \frac{x(a^2B+2abC-b^2B)}{(a^2+b^2)^2} + \frac{b^2(-3a^3C+4a^2bB-ab^2C+2b^3B)\log(a\cos(c+dx)+b\sin(c+dx))}{a^3d\,(a^2+b^2)^2} - \frac{B\cot(c+dx)}{ad(a+b\tan(c+dx))}$$

```
[In] Int[(Cot[c + d*x]^3*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c + d*x
])^2,x]
```

```
[Out] -(((a^2*B - b^2*B + 2*a*b*C)*x)/(a^2 + b^2)^2) - ((2*b*B - a*C)*Log[Sin[c +
 d*x]])/(a^3*d) + (b^2*(4*a^2*b*B + 2*b^3*B - 3*a^3*C - a*b^2*C)*Log[a*Cos[
c + d*x] + b*Sin[c + d*x]])/(a^3*(a^2 + b^2)^2*d) - (b*(a^2*B + 2*b^2*B - a
*b*C))/(a^2*(a^2 + b^2)*d*(a + b*Tan[c + d*x])) - (B*Cot[c + d*x])/(a*d*(a
+ b*Tan[c + d*x]))
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3611

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c/(b*f))*Log[RemoveContent[a*Cos[e + f*x] + b*Si
n[e + f*x], x]], x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3690

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Si
mp[b*(A*b - a*B)*(a + b*Tan[e + f*x])^(m + 1)*((c + d*Tan[e + f*x])^(n + 1)
/(f*(m + 1)*(b*c - a*d)*(a^2 + b^2))), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^
2 + b^2)), Int[(a + b*Tan[e + f*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[b*B
*(b*c*(m + 1) + a*d*(n + 1)) + A*(a*(b*c - a*d)*(m + 1) - b^2*d*(m + n + 2)
) - (A*b - a*B)*(b*c - a*d)*(m + 1)*Tan[e + f*x] - b*d*(A*b - a*B)*(m + n +
 2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, n}, x] &&
NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &
& (IntegerQ[m] || IntegersQ[2*m, 2*n]) &&  !(ILtQ[n, -1] && ( !IntegerQ[m]
```

```
|| (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rule 3730

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 - a*(b*B - a*C))*(a + b*Tan[e +
 f*x])^(m + 1)*((c + d*Tan[e + f*x])^(n + 1)/(f*(m + 1)*(b*c - a*d)*(a^2 +
b^2))), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3732

```
Int[((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^
2)/(((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])), x_Symbol] :> Simp[(a*(A*c - c*C + B*d) + b*(B*c - A*d + C*d))*(x/
((a^2 + b^2)*(c^2 + d^2))), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/((b*c - a*d)
*(a^2 + b^2)), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] - Dist
[(c^2*C - B*c*d + A*d^2)/((b*c - a*d)*(c^2 + d^2)), Int[(d - c*Tan[e + f*x]
)/(c + d*Tan[e + f*x]), x], x]) /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] &&
 NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0]
```

Rubi steps

$$\text{integral} = \int \frac{\cot^2(c+dx)(B+C\tan(c+dx))}{(a+b\tan(c+dx))^2}\,dx$$

$$= -\frac{B\cot(c+dx)}{ad(a+b\tan(c+dx))} - \frac{\int \frac{\cot(c+dx)\left(2bB-aC+aB\tan(c+dx)+2bB\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx}{a}$$

$$= -\frac{b(a^2B + 2b^2B - abC)}{a^2(a^2+b^2)\,d(a+b\tan(c+dx))} - \frac{B\cot(c+dx)}{ad(a+b\tan(c+dx))} - \frac{\int \frac{\cot(c+dx)\left((a^2+b^2)(2bB-aC)+a^2(aB+bC)\tan(c+dx)+b(a^2B+2b^2B-abC)\tan^2(c+dx)\right)}{a+b\tan(c+dx)}dx}{a^2(a^2+b^2)}$$

$$= -\frac{(a^2B - b^2B + 2abC)\,x}{(a^2+b^2)^2} - \frac{b(a^2B + 2b^2B - abC)}{a^2(a^2+b^2)\,d(a+b\tan(c+dx))} - \frac{B\cot(c+dx)}{ad(a+b\tan(c+dx))} - \frac{(2bB-aC)\int\cot(c+dx)\,dx}{a^3} + \frac{(b^2(4a^2bB+2b^3B-3a^3C-ab^2C))\int\frac{b-a\tan(c+dx)}{a+b\tan(c+dx)}dx}{a^3(a^2+b^2)^2}$$

$$= -\frac{(a^2B - b^2B + 2abC)\,x}{(a^2+b^2)^2} - \frac{(2bB-aC)\log(\sin(c+dx))}{a^3d} + \frac{b^2(4a^2bB+2b^3B-3a^3C-ab^2C)\log(a\cos(c+dx)+b\sin(c+dx))}{a^3(a^2+b^2)^2\,d} - \frac{b(a^2B + 2b^2B - abC)}{a^2(a^2+b^2)\,d(a+b\tan(c+dx))} - \frac{B\cot(c+dx)}{ad(a+b\tan(c+dx))}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 3.81 (sec) , antiderivative size = 193, normalized size of antiderivative = 1.01

$$\int \frac{\cot^3(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^2}\,dx$$
$$= \frac{-\frac{2B\cot(c+dx)}{a^2} + \frac{i(B+iC)\log(i-\tan(c+dx))}{(a+ib)^2} + \frac{2(-2bB+aC)\log(\tan(c+dx))}{a^3} - \frac{(iB+C)\log(i+\tan(c+dx))}{(a-ib)^2} - \frac{2b^2(-4a^2bB-2b^3B+3a^3C\cdots}{a^3(a^2\cdots}}{2d}$$

```
[In] Integrate[(Cot[c + d*x]^3*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c
 + d*x])^2,x]
```

```
[Out] ((-2*B*Cot[c + d*x])/a^2 + (I*(B + I*C)*Log[I - Tan[c + d*x]])/(a + I*b)^2
+ (2*(-2*b*B + a*C)*Log[Tan[c + d*x]])/a^3 - ((I*B + C)*Log[I + Tan[c + d*x
]])/(a - I*b)^2 - (2*b^2*(-4*a^2*b*B - 2*b^3*B + 3*a^3*C + a*b^2*C)*Log[a +
 b*Tan[c + d*x]])/(a^3*(a^2 + b^2)^2) + (2*b^2*(-(b*B) + a*C))/(a^2*(a^2 +
b^2)*(a + b*Tan[c + d*x])))/(2*d)
```

Maple [A] (verified)

Time = 0.54 (sec) , antiderivative size = 196, normalized size of antiderivative = 1.02

method	result
derivativedivides	$\frac{-\frac{B}{a^{2}\tan(dx+c)}+\frac{(-2Bb+Ca)\ln(\tan(dx+c))}{a^{3}}+\frac{\frac{\left(2Bab-C\,a^{2}+C\,b^{2}\right)\ln\left(1+\tan(dx+c)^{2}\right)}{2}+\left(-B\,a^{2}+B\,b^{2}-2Cab\right)\arctan(\tan(dx+c))}{\left(a^{2}+b^{2}\right)^{2}}+}{d}$
default	$\frac{-\frac{B}{a^{2}\tan(dx+c)}+\frac{(-2Bb+Ca)\ln(\tan(dx+c))}{a^{3}}+\frac{\frac{\left(2Bab-C\,a^{2}+C\,b^{2}\right)\ln\left(1+\tan(dx+c)^{2}\right)}{2}+\left(-B\,a^{2}+B\,b^{2}-2Cab\right)\arctan(\tan(dx+c))}{\left(a^{2}+b^{2}\right)^{2}}+}{d}$
parallelrisch	$\frac{4b^{2}(a+b\tan(dx+c))\left(B\,a^{2}b+\frac{1}{2}B\,b^{3}-\frac{3}{4}C\,a^{3}-\frac{1}{4}Ca\,b^{2}\right)\ln(a+b\tan(dx+c))+a^{3}\left(Bab-\frac{1}{2}C\,a^{2}+\frac{1}{2}C\,b^{2}\right)(a+b\tan(dx+c))\ln}{}$
norman	$\frac{\frac{\left(B\,a^{2}b+2B\,b^{3}-Ca\,b^{2}\right)b\tan(dx+c)^{3}}{d\,a^{3}\left(a^{2}+b^{2}\right)}-\frac{B\tan(dx+c)}{ad}-\frac{a\left(B\,a^{2}-B\,b^{2}+2Cab\right)x\tan(dx+c)^{2}}{a^{4}+2a^{2}b^{2}+b^{4}}-\frac{b\left(B\,a^{2}-B\,b^{2}+2Cab\right)x\tan(dx+c)^{3}}{a^{4}+2a^{2}b^{2}+b^{4}}}{\tan(dx+c)^{2}(a+b\tan(dx+c))}+$
risch	$\frac{xB}{2iba-a^{2}+b^{2}}-\frac{4ib^{5}Bc}{a^{3}d(a^{4}+2a^{2}b^{2}+b^{4})}+\frac{2ib^{4}Cx}{a^{2}(a^{4}+2a^{2}b^{2}+b^{4})}-\frac{4ib^{5}Bx}{a^{3}(a^{4}+2a^{2}b^{2}+b^{4})}-\frac{ixC}{2iba-a^{2}+b^{2}}-\frac{2i\left(-2iB\,a^{3}b\,\mathrm{e}^{2}\right.}{}$

```
[In] int(cot(d*x+c)^3*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x,method=
_RETURNVERBOSE)
```

```
[Out] 1/d*(-1/a^2*B/tan(d*x+c)+(-2*B*b+C*a)/a^3*ln(tan(d*x+c))+1/(a^2+b^2)^2*(1/2
*(2*B*a*b-C*a^2+C*b^2)*ln(1+tan(d*x+c)^2)+(-B*a^2+B*b^2-2*C*a*b)*arctan(tan
(d*x+c)))+b^2*(4*B*a^2*b+2*B*b^3-3*C*a^3-C*a*b^2)/(a^2+b^2)^2/a^3*ln(a+b*ta
n(d*x+c))-(B*b-C*a)*b^2/(a^2+b^2)/a^2/(a+b*tan(d*x+c)))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 465 vs. 2(190) = 380.

Time = 0.32 (sec) , antiderivative size = 465, normalized size of antiderivative = 2.42

$$\int \frac{\cot^3(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx =$$

$$-\frac{2\,Ba^6+4\,Ba^4b^2+2\,Ba^2b^4+2\,(Ca^3b^3-Ba^2b^4+(Ba^5b+2\,Ca^4b^2-Ba^3b^3)dx)\tan{(dx+c)}^2-\big((Ca^5}{}$$

```
[In] integrate(cot(d*x+c)^3*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x,
algorithm="fricas")
```

```
[Out] -1/2*(2*B*a^6 + 4*B*a^4*b^2 + 2*B*a^2*b^4 + 2*(C*a^3*b^3 - B*a^2*b^4 + (B*a
^5*b + 2*C*a^4*b^2 - B*a^3*b^3)*d*x)*tan(d*x + c)^2 - ((C*a^5*b - 2*B*a^4*b
^2 + 2*C*a^3*b^3 - 4*B*a^2*b^4 + C*a*b^5 - 2*B*b^6)*tan(d*x + c)^2 + (C*a^6
 - 2*B*a^5*b + 2*C*a^4*b^2 - 4*B*a^3*b^3 + C*a^2*b^4 - 2*B*a*b^5)*tan(d*x +
```

```
c))*log(tan(d*x + c)^2/(tan(d*x + c)^2 + 1)) + ((3*C*a^3*b^3 - 4*B*a^2*b^4
 + C*a*b^5 - 2*B*b^6)*tan(d*x + c)^2 + (3*C*a^4*b^2 - 4*B*a^3*b^3 + C*a^2*b
^4 - 2*B*a*b^5)*tan(d*x + c))*log((b^2*tan(d*x + c)^2 + 2*a*b*tan(d*x + c)
+ a^2)/(tan(d*x + c)^2 + 1)) + 2*(B*a^5*b + 2*B*a^3*b^3 - C*a^2*b^4 + 2*B*a
*b^5 + (B*a^6 + 2*C*a^5*b - B*a^4*b^2)*d*x)*tan(d*x + c))/((a^7*b + 2*a^5*b
^3 + a^3*b^5)*d*tan(d*x + c)^2 + (a^8 + 2*a^6*b^2 + a^4*b^4)*d*tan(d*x + c)
)
```

Sympy [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 6.03 (sec) , antiderivative size = 8143, normalized size of antiderivative = 42.41

$$\int \frac{\cot^3(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx = \text{Too large to display}$$

```
[In] integrate(cot(d*x+c)**3*(B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c))**2,
x)
```

```
[Out] Piecewise(((-B*x - B/(d*tan(c + d*x)) - C*log(tan(c + d*x)**2 + 1)/(2*d) +
C*log(tan(c + d*x))/d)/a**2, Eq(b, 0)), ((B*x + B/(d*tan(c + d*x)) - B/(3*d
*tan(c + d*x)**3) + C*log(tan(c + d*x)**2 + 1)/(2*d) - C*log(tan(c + d*x))/
d - C/(2*d*tan(c + d*x)**2))/b**2, Eq(a, 0)), (-9*B*d*x*tan(c + d*x)**3/(4*
a**2*d*tan(c + d*x)**3 + 8*I*a**2*d*tan(c + d*x)**2 - 4*a**2*d*tan(c + d*x)
) - 18*I*B*d*x*tan(c + d*x)**2/(4*a**2*d*tan(c + d*x)**3 + 8*I*a**2*d*tan(c
 + d*x)**2 - 4*a**2*d*tan(c + d*x)) + 9*B*d*x*tan(c + d*x)/(4*a**2*d*tan(c
+ d*x)**3 + 8*I*a**2*d*tan(c + d*x)**2 - 4*a**2*d*tan(c + d*x)) - 4*I*B*log
(tan(c + d*x)**2 + 1)*tan(c + d*x)**3/(4*a**2*d*tan(c + d*x)**3 + 8*I*a**2*
d*tan(c + d*x)**2 - 4*a**2*d*tan(c + d*x)) + 8*B*log(tan(c + d*x)**2 + 1)*t
an(c + d*x)**2/(4*a**2*d*tan(c + d*x)**3 + 8*I*a**2*d*tan(c + d*x)**2 - 4*a
**2*d*tan(c + d*x)) + 4*I*B*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(4*a**2*d
*tan(c + d*x)**3 + 8*I*a**2*d*tan(c + d*x)**2 - 4*a**2*d*tan(c + d*x)) + 8*
I*B*log(tan(c + d*x))*tan(c + d*x)**3/(4*a**2*d*tan(c + d*x)**3 + 8*I*a**2*
d*tan(c + d*x)**2 - 4*a**2*d*tan(c + d*x)) - 16*B*log(tan(c + d*x))*tan(c +
 d*x)**2/(4*a**2*d*tan(c + d*x)**3 + 8*I*a**2*d*tan(c + d*x)**2 - 4*a**2*d*
tan(c + d*x)) - 8*I*B*log(tan(c + d*x))*tan(c + d*x)/(4*a**2*d*tan(c + d*x)
**3 + 8*I*a**2*d*tan(c + d*x)**2 - 4*a**2*d*tan(c + d*x)) - 9*B*tan(c + d*x
)**2/(4*a**2*d*tan(c + d*x)**3 + 8*I*a**2*d*tan(c + d*x)**2 - 4*a**2*d*tan(
c + d*x)) - 14*I*B*tan(c + d*x)/(4*a**2*d*tan(c + d*x)**3 + 8*I*a**2*d*tan(
c + d*x)**2 - 4*a**2*d*tan(c + d*x)) + 4*B/(4*a**2*d*tan(c + d*x)**3 + 8*I*
a**2*d*tan(c + d*x)**2 - 4*a**2*d*tan(c + d*x)) + 3*I*C*d*x*tan(c + d*x)**3
/(4*a**2*d*tan(c + d*x)**3 + 8*I*a**2*d*tan(c + d*x)**2 - 4*a**2*d*tan(c +
d*x)) - 6*C*d*x*tan(c + d*x)**2/(4*a**2*d*tan(c + d*x)**3 + 8*I*a**2*d*tan(
c + d*x)**2 - 4*a**2*d*tan(c + d*x)) - 3*I*C*d*x*tan(c + d*x)/(4*a**2*d*tan
(c + d*x)**3 + 8*I*a**2*d*tan(c + d*x)**2 - 4*a**2*d*tan(c + d*x)) - 2*C*lo
```

g(tan(c + d*x)**2 + 1)*tan(c + d*x)**3/(4*a**2*d*tan(c + d*x)**3 + 8*I*a**2
*d*tan(c + d*x)**2 - 4*a**2*d*tan(c + d*x)) - 4*I*C*log(tan(c + d*x)**2 + 1
)*tan(c + d*x)**2/(4*a**2*d*tan(c + d*x)**3 + 8*I*a**2*d*tan(c + d*x)**2 -
4*a**2*d*tan(c + d*x)) + 2*C*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(4*a**2*
d*tan(c + d*x)**3 + 8*I*a**2*d*tan(c + d*x)**2 - 4*a**2*d*tan(c + d*x)) + 4
*C*log(tan(c + d*x))*tan(c + d*x)**3/(4*a**2*d*tan(c + d*x)**3 + 8*I*a**2*d
*tan(c + d*x)**2 - 4*a**2*d*tan(c + d*x)) + 8*I*C*log(tan(c + d*x))*tan(c +
 d*x)**2/(4*a**2*d*tan(c + d*x)**3 + 8*I*a**2*d*tan(c + d*x)**2 - 4*a**2*d*
tan(c + d*x)) - 4*C*log(tan(c + d*x))*tan(c + d*x)/(4*a**2*d*tan(c + d*x)**
3 + 8*I*a**2*d*tan(c + d*x)**2 - 4*a**2*d*tan(c + d*x)) + 3*I*C*tan(c + d*x
)**2/(4*a**2*d*tan(c + d*x)**3 + 8*I*a**2*d*tan(c + d*x)**2 - 4*a**2*d*tan(
c + d*x)) - 4*C*tan(c + d*x)/(4*a**2*d*tan(c + d*x)**3 + 8*I*a**2*d*tan(c +
 d*x)**2 - 4*a**2*d*tan(c + d*x)), Eq(b, -I*a)), (-9*B*d*x*tan(c + d*x)**3/
(4*a**2*d*tan(c + d*x)**3 - 8*I*a**2*d*tan(c + d*x)**2 - 4*a**2*d*tan(c + d
*x)) + 18*I*B*d*x*tan(c + d*x)**2/(4*a**2*d*tan(c + d*x)**3 - 8*I*a**2*d*ta
n(c + d*x)**2 - 4*a**2*d*tan(c + d*x)) + 9*B*d*x*tan(c + d*x)/(4*a**2*d*tan
(c + d*x)**3 - 8*I*a**2*d*tan(c + d*x)**2 - 4*a**2*d*tan(c + d*x)) + 4*I*B*
log(tan(c + d*x)**2 + 1)*tan(c + d*x)**3/(4*a**2*d*tan(c + d*x)**3 - 8*I*a*
*2*d*tan(c + d*x)**2 - 4*a**2*d*tan(c + d*x)) + 8*B*log(tan(c + d*x)**2 + 1
)*tan(c + d*x)**2/(4*a**2*d*tan(c + d*x)**3 - 8*I*a**2*d*tan(c + d*x)**2 -
4*a**2*d*tan(c + d*x)) - 4*I*B*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(4*a**
2*d*tan(c + d*x)**3 - 8*I*a**2*d*tan(c + d*x)**2 - 4*a**2*d*tan(c + d*x)) -
 8*I*B*log(tan(c + d*x))*tan(c + d*x)**3/(4*a**2*d*tan(c + d*x)**3 - 8*I*a*
*2*d*tan(c + d*x)**2 - 4*a**2*d*tan(c + d*x)) - 16*B*log(tan(c + d*x))*tan(
c + d*x)**2/(4*a**2*d*tan(c + d*x)**3 - 8*I*a**2*d*tan(c + d*x)**2 - 4*a**2
*d*tan(c + d*x)) + 8*I*B*log(tan(c + d*x))*tan(c + d*x)/(4*a**2*d*tan(c + d
*x)**3 - 8*I*a**2*d*tan(c + d*x)**2 - 4*a**2*d*tan(c + d*x)) - 9*B*tan(c +
d*x)**2/(4*a**2*d*tan(c + d*x)**3 - 8*I*a**2*d*tan(c + d*x)**2 - 4*a**2*d*t
an(c + d*x)) + 14*I*B*tan(c + d*x)/(4*a**2*d*tan(c + d*x)**3 - 8*I*a**2*d*t
an(c + d*x)**2 - 4*a**2*d*tan(c + d*x)) + 4*B/(4*a**2*d*tan(c + d*x)**3 - 8
*I*a**2*d*tan(c + d*x)**2 - 4*a**2*d*tan(c + d*x)) - 3*I*C*d*x*tan(c + d*x)
3/(4*a2*d*tan(c + d*x)**3 - 8*I*a**2*d*tan(c + d*x)**2 - 4*a**2*d*tan(c
 + d*x)) - 6*C*d*x*tan(c + d*x)**2/(4*a**2*d*tan(c + d*x)**3 - 8*I*a**2*d*t
an(c + d*x)**2 - 4*a**2*d*tan(c + d*x)) + 3*I*C*d*x*tan(c + d*x)/(4*a**2*d*
tan(c + d*x)**3 - 8*I*a**2*d*tan(c + d*x)**2 - 4*a**2*d*tan(c + d*x)) - 2*C
*log(tan(c + d*x)**2 + 1)*tan(c + d*x)**3/(4*a**2*d*tan(c + d*x)**3 - 8*I*a
2*d*tan(c + d*x)2 - 4*a**2*d*tan(c + d*x)) + 4*I*C*log(tan(c + d*x)**2
+ 1)*tan(c + d*x)**2/(4*a**2*d*tan(c + d*x)**3 - 8*I*a**2*d*tan(c + d*x)**2
 - 4*a**2*d*tan(c + d*x)) + 2*C*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(4*a*
*2*d*tan(c + d*x)**3 - 8*I*a**2*d*tan(c + d*x)**2 - 4*a**2*d*tan(c + d*x))
+ 4*C*log(tan(c + d*x))*tan(c + d*x)**3/(4*a**2*d*tan(c + d*x)**3 - 8*I*a**
2*d*tan(c + d*x)**2 - 4*a**2*d*tan(c + d*x)) - 8*I*C*log(tan(c + d*x))*tan(
c + d*x)**2/(4*a**2*d*tan(c + d*x)**3 - 8*I*a**2*d*tan(c + d*x)**2 - 4*a**2
*d*tan(c + d*x)) - 4*C*log(tan(c + d*x))*tan(c + d*x)/(4*a**2*d*tan(c + d*x
)**3 - 8*I*a**2*d*tan(c + d*x)**2 - 4*a**2*d*tan(c + d*x)) - 3*I*C*tan(c +

d*x)**2/(4*a**2*d*tan(c + d*x)**3 - 8*I*a**2*d*tan(c + d*x)**2 - 4*a**2*d*t
an(c + d*x)) - 4*C*tan(c + d*x)/(4*a**2*d*tan(c + d*x)**3 - 8*I*a**2*d*tan(
c + d*x)**2 - 4*a**2*d*tan(c + d*x)), Eq(b, I*a)), (zoo*(-B*x - B/(d*tan(c
+ d*x)) - C*log(tan(c + d*x)**2 + 1)/(2*d) + C*log(tan(c + d*x))/d), Eq(b,
-a/tan(c + d*x))), (nan, Eq(c, -d*x)), (x*(B*tan(c) + C*tan(c)**2)*cot(c)**
3/(a + b*tan(c))**2, Eq(d, 0)), (-2*B*a**6*d*x*tan(c + d*x)/(2*a**8*d*tan(c
 + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*x) + 4*a**5*
b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*d*tan(c +
 d*x)**2) - 2*B*a**6/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 +
4*a**6*b**2*d*tan(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*
tan(c + d*x) + 2*a**3*b**5*d*tan(c + d*x)**2) - 2*B*a**5*b*d*x*tan(c + d*x)
2/(2*a8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan
(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*
a**3*b**5*d*tan(c + d*x)**2) + 2*B*a**5*b*log(tan(c + d*x)**2 + 1)*tan(c +
d*x)/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*ta
n(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2
*a**3*b**5*d*tan(c + d*x)**2) - 4*B*a**5*b*log(tan(c + d*x))*tan(c + d*x)/(
2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c +
d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*
b**5*d*tan(c + d*x)**2) - 2*B*a**5*b*tan(c + d*x)/(2*a**8*d*tan(c + d*x) +
2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*x) + 4*a**5*b**3*d*tan
(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*d*tan(c + d*x)**2)
+ 2*B*a**4*b**2*d*x*tan(c + d*x)/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c
+ d*x)**2 + 4*a**6*b**2*d*tan(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*
a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*d*tan(c + d*x)**2) + 2*B*a**4*b**2*l
og(tan(c + d*x)**2 + 1)*tan(c + d*x)**2/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d
*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)*
*2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*d*tan(c + d*x)**2) - 4*B*a**4
*b**2*log(tan(c + d*x))*tan(c + d*x)**2/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d
*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)*
*2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*d*tan(c + d*x)**2) - 4*B*a**4
*b**2/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*t
an(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) +
2*a**3*b**5*d*tan(c + d*x)**2) + 2*B*a**3*b**3*d*x*tan(c + d*x)**2/(2*a**8*
d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*x) +
4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*d*
tan(c + d*x)**2) + 8*B*a**3*b**3*log(a/b + tan(c + d*x))*tan(c + d*x)/(2*a*
*8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*x)
 + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5
*d*tan(c + d*x)**2) - 8*B*a**3*b**3*log(tan(c + d*x))*tan(c + d*x)/(2*a**8*
d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*x) +
4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*d*
tan(c + d*x)**2) - 6*B*a**3*b**3*tan(c + d*x)/(2*a**8*d*tan(c + d*x) + 2*a*
*7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*x) + 4*a**5*b**3*d*tan(c +
 d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*d*tan(c + d*x)**2) + 8*

B*a**2*b**4*log(a/b + tan(c + d*x))*tan(c + d*x)**2/(2*a**8*d*tan(c + d*x)
+ 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*x) + 4*a**5*b**3*d*t
an(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*d*tan(c + d*x)**2
) - 8*B*a**2*b**4*log(tan(c + d*x))*tan(c + d*x)**2/(2*a**8*d*tan(c + d*x)
+ 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*x) + 4*a**5*b**3*d*t
an(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*d*tan(c + d*x)**2
) - 2*B*a**2*b**4/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a
6*b2*d*tan(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan
(c + d*x) + 2*a**3*b**5*d*tan(c + d*x)**2) + 4*B*a*b**5*log(a/b + tan(c + d
*x))*tan(c + d*x)/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a
6*b2*d*tan(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan
(c + d*x) + 2*a**3*b**5*d*tan(c + d*x)**2) - 4*B*a*b**5*log(tan(c + d*x))*t
an(c + d*x)/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b*
*2*d*tan(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d
*x) + 2*a**3*b**5*d*tan(c + d*x)**2) - 4*B*a*b**5*tan(c + d*x)/(2*a**8*d*ta
n(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*x) + 4*a*
*5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*d*tan(
c + d*x)**2) + 4*B*b**6*log(a/b + tan(c + d*x))*tan(c + d*x)**2/(2*a**8*d*t
an(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*x) + 4*a
5*b3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*d*tan
(c + d*x)**2) - 4*B*b**6*log(tan(c + d*x))*tan(c + d*x)**2/(2*a**8*d*tan(c
+ d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*x) + 4*a**5*b
3*d*tan(c + d*x)2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*d*tan(c +
d*x)**2) - C*a**6*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(2*a**8*d*tan(c + d
*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*x) + 4*a**5*b**3
*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*d*tan(c + d*x
)**2) + 2*C*a**6*log(tan(c + d*x))*tan(c + d*x)/(2*a**8*d*tan(c + d*x) + 2*
a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*x) + 4*a**5*b**3*d*tan(c
 + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*d*tan(c + d*x)**2) -
4*C*a**5*b*d*x*tan(c + d*x)/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x
)**2 + 4*a**6*b**2*d*tan(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*
b**4*d*tan(c + d*x) + 2*a**3*b**5*d*tan(c + d*x)**2) - C*a**5*b*log(tan(c +
 d*x)**2 + 1)*tan(c + d*x)**2/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d
*x)**2 + 4*a**6*b**2*d*tan(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**
4*b**4*d*tan(c + d*x) + 2*a**3*b**5*d*tan(c + d*x)**2) + 2*C*a**5*b*log(tan
(c + d*x))*tan(c + d*x)**2/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)
2 + 4*a6*b**2*d*tan(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b
4*d*tan(c + d*x) + 2*a3*b**5*d*tan(c + d*x)**2) - 4*C*a**4*b**2*d*x*tan
(c + d*x)**2/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b
2*d*tan(c + d*x) + 4*a5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c +
d*x) + 2*a**3*b**5*d*tan(c + d*x)**2) - 6*C*a**4*b**2*log(a/b + tan(c + d*x
))*tan(c + d*x)/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**
6*b**2*d*tan(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c
 + d*x) + 2*a**3*b**5*d*tan(c + d*x)**2) + C*a**4*b**2*log(tan(c + d*x)**2
+ 1)*tan(c + d*x)/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a

```
**6*b**2*d*tan(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan
(c + d*x) + 2*a**3*b**5*d*tan(c + d*x)**2) + 4*C*a**4*b**2*log(tan(c + d*x)
)*tan(c + d*x)/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6
*b**2*d*tan(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c
+ d*x) + 2*a**3*b**5*d*tan(c + d*x)**2) + 2*C*a**4*b**2*tan(c + d*x)/(2*a**
8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*x)
+ 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*
d*tan(c + d*x)**2) - 6*C*a**3*b**3*log(a/b + tan(c + d*x))*tan(c + d*x)**2/
(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c +
 d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3
*b**5*d*tan(c + d*x)**2) + C*a**3*b**3*log(tan(c + d*x)**2 + 1)*tan(c + d*x
)**2/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*ta
n(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2
*a**3*b**5*d*tan(c + d*x)**2) + 4*C*a**3*b**3*log(tan(c + d*x))*tan(c + d*x
)**2/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*ta
n(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2
*a**3*b**5*d*tan(c + d*x)**2) - 2*C*a**2*b**4*log(a/b + tan(c + d*x))*tan(c
 + d*x)/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d
*tan(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x)
+ 2*a**3*b**5*d*tan(c + d*x)**2) + 2*C*a**2*b**4*log(tan(c + d*x))*tan(c +
d*x)/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*ta
n(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2
*a**3*b**5*d*tan(c + d*x)**2) + 2*C*a**2*b**4*tan(c + d*x)/(2*a**8*d*tan(c
+ d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*x) + 4*a**5*b
**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*d*tan(c +
d*x)**2) - 2*C*a*b**5*log(a/b + tan(c + d*x))*tan(c + d*x)**2/(2*a**8*d*tan
(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*x) + 4*a**
5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*d*tan(c
 + d*x)**2) + 2*C*a*b**5*log(tan(c + d*x))*tan(c + d*x)**2/(2*a**8*d*tan(c
+ d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*x) + 4*a**5*b
**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*d*tan(c +
d*x)**2), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.35 (sec) , antiderivative size = 262, normalized size of antiderivative = 1.36

$$\int \frac{\cot^3(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^2}\,dx = -\frac{\frac{2\,(Ba^2+2\,Cab-Bb^2)(dx+c)}{a^4+2\,a^2b^2+b^4}+\frac{2\,(3\,Ca^3b^2-4\,Ba^2b^3+Cab^4-2\,Bb^5)\log(b\tan(dx+c)+a)}{a^7+2\,a^5b^2+a^3b^4}+\frac{(Ca^2-2\,Bab-Cb^2)\log\left(\tan(dx+c)^2+1\right)}{a^4+2\,a^2b^2+b^4}+\frac{2\,(\cdots}{(a\cdots}}{2\,d}$$

```
[In] integrate(cot(d*x+c)^3*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x,
```

```
algorithm="maxima")
```

```
[Out] -1/2*(2*(B*a^2 + 2*C*a*b - B*b^2)*(d*x + c)/(a^4 + 2*a^2*b^2 + b^4) + 2*(3*
C*a^3*b^2 - 4*B*a^2*b^3 + C*a*b^4 - 2*B*b^5)*log(b*tan(d*x + c) + a)/(a^7 +
 2*a^5*b^2 + a^3*b^4) + (C*a^2 - 2*B*a*b - C*b^2)*log(tan(d*x + c)^2 + 1)/(
a^4 + 2*a^2*b^2 + b^4) + 2*(B*a^3 + B*a*b^2 + (B*a^2*b - C*a*b^2 + 2*B*b^3)
*tan(d*x + c))/((a^4*b + a^2*b^3)*tan(d*x + c)^2 + (a^5 + a^3*b^2)*tan(d*x
+ c)) - 2*(C*a - 2*B*b)*log(tan(d*x + c))/a^3)/d
```

Giac [A] (verification not implemented)

none

Time = 1.23 (sec) , antiderivative size = 362, normalized size of antiderivative = 1.89

$$\int \frac{\cot^3(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx = -\frac{\frac{2\left(Ba^2+2\,Cab-Bb^2\right)(dx+c)}{a^4+2\,a^2b^2+b^4}+\frac{\left(Ca^2-2\,Bab-Cb^2\right)\log\left(\tan(dx+c)^2+1\right)}{a^4+2\,a^2b^2+b^4}+\frac{2\left(3\,Ca^3b^3-4\,Ba^2b^4+Cab^5-2\,Bb^6\right)\log(|b\tan(dx+c)+a|)}{a^7b+2\,a^5b^3+a^3b^5}+}{}$$

```
[In] integrate(cot(d*x+c)^3*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x,
algorithm="giac")
```

```
[Out] -1/2*(2*(B*a^2 + 2*C*a*b - B*b^2)*(d*x + c)/(a^4 + 2*a^2*b^2 + b^4) + (C*a^
2 - 2*B*a*b - C*b^2)*log(tan(d*x + c)^2 + 1)/(a^4 + 2*a^2*b^2 + b^4) + 2*(3
*C*a^3*b^3 - 4*B*a^2*b^4 + C*a*b^5 - 2*B*b^6)*log(abs(b*tan(d*x + c) + a))/
(a^7*b + 2*a^5*b^3 + a^3*b^5) + (C*a^4*b*tan(d*x + c)^2 - 2*B*a^3*b^2*tan(d
*x + c)^2 - C*a^2*b^3*tan(d*x + c)^2 + C*a^5*tan(d*x + c) - 3*C*a^3*b^2*tan
(d*x + c) + 6*B*a^2*b^3*tan(d*x + c) - 2*C*a*b^4*tan(d*x + c) + 4*B*b^5*tan
(d*x + c) + 2*B*a^5 + 4*B*a^3*b^2 + 2*B*a*b^4)/((a^6 + 2*a^4*b^2 + a^2*b^4)
*(b*tan(d*x + c)^2 + a*tan(d*x + c))) - 2*(C*a - 2*B*b)*log(abs(tan(d*x + c
)))/a^3)/d
```

Mupad [B] (verification not implemented)

Time = 11.13 (sec) , antiderivative size = 230, normalized size of antiderivative = 1.20

$$\begin{aligned}\int \frac{\cot^3(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx &= \frac{b^2\,\ln\left(a+b\tan(c+d\,x)\right)\,\left(-3\,C\,a^3+4\,B\,a^2\,b-C\,a\,b^2+2\,B\,b^3\right)}{a^3\,d\,\left(a^2+b^2\right)^2} \\ &\quad -\frac{\ln\left(\tan(c+d\,x)\right)\,\left(2\,B\,b-C\,a\right)}{a^3\,d}+\frac{\ln\left(\tan(c+d\,x)+1\mathrm{i}\right)\,\left(C+B\,1\mathrm{i}\right)}{2\,d\,\left(-a^2+a\,b\,2\mathrm{i}+b^2\right)} \\ &\quad +\frac{\ln\left(\tan(c+d\,x)-\mathrm{i}\right)\,\left(B+C\,1\mathrm{i}\right)}{2\,d\,\left(-a^2\,1\mathrm{i}+2\,a\,b+b^2\,1\mathrm{i}\right)}-\frac{\frac{B}{a}+\frac{\tan(c+d\,x)\,\left(B\,a^2\,b-C\,a\,b^2+2\,B\,b^3\right)}{a^2\,\left(a^2+b^2\right)}}{d\,\left(b\tan\left(c+d\,x\right)^2+a\tan\left(c+d\,x\right)\right)}\end{aligned}$$

```
[In] int((cot(c + d*x)^3*(B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c + d*x
))^2,x)
```

```
[Out] (log(tan(c + d*x) + 1i)*(B*1i + C))/(2*d*(a*b*2i - a^2 + b^2)) - (log(tan(c
 + d*x))*(2*B*b - C*a))/(a^3*d) - (B/a + (tan(c + d*x)*(2*B*b^3 + B*a^2*b -
 C*a*b^2))/(a^2*(a^2 + b^2)))/(d*(a*tan(c + d*x) + b*tan(c + d*x)^2)) + (lo
g(tan(c + d*x) - 1i)*(B + C*1i))/(2*d*(2*a*b - a^2*1i + b^2*1i)) + (b^2*log
(a + b*tan(c + d*x))*(2*B*b^3 - 3*C*a^3 + 4*B*a^2*b - C*a*b^2))/(a^3*d*(a^2
 + b^2)^2)
```

3.38 $\int \frac{\tan^3(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx$

Optimal result 325
Rubi [A] (verified) 326
Mathematica [C] (verified) 330
Maple [A] (verified) 330
Fricas [B] (verification not implemented) 331
Sympy [F(-2)] 332
Maxima [A] (verification not implemented) 332
Giac [A] (verification not implemented) 333
Mupad [B] (verification not implemented) 333

Optimal result

Integrand size = 40, antiderivative size = 331

$$\begin{aligned}
&\int \frac{\tan^3(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx \\
&= \frac{(a^3B-3ab^2B+3a^2bC-b^3C)\,x}{(a^2+b^2)^3}+\frac{(3a^2bB-b^3B-a^3C+3ab^2C)\log(\cos(c+dx))}{(a^2+b^2)^3\,d} \\
&\quad+\frac{a^2(a^4bB+3a^2b^3B+6b^5B-3a^5C-9a^3b^2C-10ab^4C)\log(a+b\tan(c+dx))}{b^4\,(a^2+b^2)^3\,d} \\
&\quad-\frac{(a^3bB+3ab^3B-3a^4C-6a^2b^2C-b^4C)\tan(c+dx)}{b^3\,(a^2+b^2)^2\,d} \\
&\quad+\frac{a(bB-aC)\tan^3(c+dx)}{2b\,(a^2+b^2)\,d(a+b\tan(c+dx))^2}+\frac{a(a^2bB+5b^3B-3a^3C-7ab^2C)\tan^2(c+dx)}{2b^2\,(a^2+b^2)^2\,d(a+b\tan(c+dx))}
\end{aligned}$$

```
[Out] (B*a^3-3*B*a*b^2+3*C*a^2*b-C*b^3)*x/(a^2+b^2)^3+(3*B*a^2*b-B*b^3-C*a^3+3*C*
a*b^2)*ln(cos(d*x+c))/(a^2+b^2)^3/d+a^2*(B*a^4*b+3*B*a^2*b^3+6*B*b^5-3*C*a^
5-9*C*a^3*b^2-10*C*a*b^4)*ln(a+b*tan(d*x+c))/b^4/(a^2+b^2)^3/d-(B*a^3*b+3*B
*a*b^3-3*C*a^4-6*C*a^2*b^2-C*b^4)*tan(d*x+c)/b^3/(a^2+b^2)^2/d+1/2*a*(B*b-C
*a)*tan(d*x+c)^3/b/(a^2+b^2)/d/(a+b*tan(d*x+c))^2+1/2*a*(B*a^2*b+5*B*b^3-3*
C*a^3-7*C*a*b^2)*tan(d*x+c)^2/b^2/(a^2+b^2)^2/d/(a+b*tan(d*x+c))
```

Rubi [A] (verified)

Time = 0.97 (sec) , antiderivative size = 331, normalized size of antiderivative = 1.00, number of steps used = 8, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}} = 0.200$, Rules used = {3713, 3686, 3726, 3728, 3707, 3698, 31, 3556}

$$\begin{aligned}
&\int \frac{\tan^3(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx \\
&= \frac{a(bB-aC)\tan^3(c+dx)}{2bd\,(a^2+b^2)\,(a+b\tan(c+dx))^2}+\frac{a(-3a^3C+a^2bB-7ab^2C+5b^3B)\tan^2(c+dx)}{2b^2d\,(a^2+b^2)^2\,(a+b\tan(c+dx))} \\
&\quad+\frac{(a^3(-C)+3a^2bB+3ab^2C-b^3B)\log(\cos(c+dx))}{d\,(a^2+b^2)^3} \\
&\quad+\frac{x(a^3B+3a^2bC-3ab^2B-b^3C)}{(a^2+b^2)^3} \\
&\quad-\frac{(-3a^4C+a^3bB-6a^2b^2C+3ab^3B-b^4C)\tan(c+dx)}{b^3d\,(a^2+b^2)^2} \\
&\quad+\frac{a^2(-3a^5C+a^4bB-9a^3b^2C+3a^2b^3B-10ab^4C+6b^5B)\log(a+b\tan(c+dx))}{b^4d\,(a^2+b^2)^3}
\end{aligned}$$

```
[In] Int[(Tan[c + d*x]^3*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c + d*x
])^3,x]
```

```
[Out] ((a^3*B - 3*a*b^2*B + 3*a^2*b*C - b^3*C)*x)/(a^2 + b^2)^3 + ((3*a^2*b*B - b
^3*B - a^3*C + 3*a*b^2*C)*Log[Cos[c + d*x]])/((a^2 + b^2)^3*d) + (a^2*(a^4*
b*B + 3*a^2*b^3*B + 6*b^5*B - 3*a^5*C - 9*a^3*b^2*C - 10*a*b^4*C)*Log[a + b
*Tan[c + d*x]])/(b^4*(a^2 + b^2)^3*d) - ((a^3*b*B + 3*a*b^3*B - 3*a^4*C - 6
*a^2*b^2*C - b^4*C)*Tan[c + d*x])/(b^3*(a^2 + b^2)^2*d) + (a*(b*B - a*C)*Ta
n[c + d*x]^3)/(2*b*(a^2 + b^2)*d*(a + b*Tan[c + d*x])^2) + (a*(a^2*b*B + 5*
b^3*B - 3*a^3*C - 7*a*b^2*C)*Tan[c + d*x]^2)/(2*b^2*(a^2 + b^2)^2*d*(a + b*
Tan[c + d*x]))
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3686

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Si
```

```
mp[(b*c - a*d)*(B*c - A*d)*(a + b*Tan[e + f*x])^(m - 1)*((c + d*Tan[e + f*x
])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dist[1/(d*(n + 1)*(c^2 + d^2)),
 Int[(a + b*Tan[e + f*x])^(m - 2)*(c + d*Tan[e + f*x])^(n + 1)*Simp[a*A*d*(
b*d*(m - 1) - a*c*(n + 1)) + (b*B*c - (A*b + a*B)*d)*(b*c*(m - 1) + a*d*(n
+ 1)) - d*((a*A - b*B)*(b*c - a*d) + (A*b + a*B)*(a*c + b*d))*(n + 1)*Tan[e
 + f*x] - b*(d*(A*b*c + a*B*c - a*A*d)*(m + n) - b*B*(c^2*(m - 1) - d^2*(n
+ 1)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] &&
NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 1] &&
 LtQ[n, -1] && (IntegerQ[m] || IntegersQ[2*m, 2*n])
```

Rule 3698

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3707

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[(a*A + b*B - a
*C)*(x/(a^2 + b^2)), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rule 3726

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*d^2 + c*(c*C - B*d))*(a + b*Tan[e +
 f*x])^m*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
```

```
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\text{integral} = \int \frac{\tan^4(c+dx)(B+C\tan(c+dx))}{(a+b\tan(c+dx))^3}\,dx$$

$$= \frac{a(bB-aC)\tan^3(c+dx)}{2b(a^2+b^2)d(a+b\tan(c+dx))^2} + \frac{\int \frac{\tan^2(c+dx)\left(-3a(bB-aC)+2b(bB-aC)\tan(c+dx)-\left(abB-3a^2C-2b^2C\right)\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx}{2b(a^2+b^2)}$$

$$= \frac{a(bB-aC)\tan^3(c+dx)}{2b(a^2+b^2)d(a+b\tan(c+dx))^2} + \frac{a(a^2bB+5b^3B-3a^3C-7ab^2C)\tan^2(c+dx)}{2b^2(a^2+b^2)^2d(a+b\tan(c+dx))} + \frac{\int \frac{\tan(c+dx)\left(-2a\left(a^2bB+5b^3B-3a^3C-7ab^2C\right)-2b^2\left(a^2B-b^2B+2abC\right)\tan(c+dx)-2\left(a^3bB+3ab^3B-3a^4C-6a^2b^2C-b^4C\right)\tan^2(c}{a+b\tan(c+dx)}}{2b^2(a^2+b^2)^2}$$

$$= -\frac{(a^3bB+3ab^3B-3a^4C-6a^2b^2C-b^4C)\tan(c+dx)}{b^3(a^2+b^2)^2d} + \frac{a(bB-aC)\tan^3(c+dx)}{2b(a^2+b^2)d(a+b\tan(c+dx))^2} + \frac{a(a^2bB+5b^3B-3a^3C-7ab^2C)\tan^2(c+dx)}{2b^2(a^2+b^2)^2d(a+b\tan(c+dx))} + \frac{\int \frac{2a\left(a^3bB+3ab^3B-3a^4C-6a^2b^2C-b^4C\right)-2b^3\left(2abB-a^2C+b^2C\right)\tan(c+dx)+2\left(a^2+b^2\right)^2(bB-3aC)\tan^2(c+dx)}{a+b\tan(c+dx)}\,dx}{2b^3(a^2+b^2)^2}$$

$$= \frac{\left(a^3B-3ab^2B+3a^2bC-b^3C\right)x}{\left(a^2+b^2\right)^3} - \frac{\left(a^3bB+3ab^3B-3a^4C-6a^2b^2C-b^4C\right)\tan(c+dx)}{b^3\left(a^2+b^2\right)^2 d} + \frac{a(bB-aC)\tan^3(c+dx)}{2b\left(a^2+b^2\right)d(a+b\tan(c+dx))^2} + \frac{a\left(a^2bB+5b^3B-3a^3C-7ab^2C\right)\tan^2(c+dx)}{2b^2\left(a^2+b^2\right)^2 d(a+b\tan(c+dx))} - \frac{\left(3a^2bB-b^3B-a^3C+3ab^2C\right)\int\tan(c+dx)\,dx}{\left(a^2+b^2\right)^3} + \frac{\left(a^2\left(a^4bB+3a^2b^3B+6b^5B-3a^5C-9a^3b^2C-10ab^4C\right)\right)\int\frac{1+\tan^2(c+dx)}{a+b\tan(c+dx)}\,dx}{b^3\left(a^2+b^2\right)^3}$$

$$= \frac{\left(a^3B-3ab^2B+3a^2bC-b^3C\right)x}{\left(a^2+b^2\right)^3} + \frac{\left(3a^2bB-b^3B-a^3C+3ab^2C\right)\log(\cos(c+dx))}{\left(a^2+b^2\right)^3 d} - \frac{\left(a^3bB+3ab^3B-3a^4C-6a^2b^2C-b^4C\right)\tan(c+dx)}{b^3\left(a^2+b^2\right)^2 d} + \frac{a(bB-aC)\tan^3(c+dx)}{2b\left(a^2+b^2\right)d(a+b\tan(c+dx))^2} + \frac{a\left(a^2bB+5b^3B-3a^3C-7ab^2C\right)\tan^2(c+dx)}{2b^2\left(a^2+b^2\right)^2 d(a+b\tan(c+dx))} + \frac{\left(a^2\left(a^4bB+3a^2b^3B+6b^5B-3a^5C-9a^3b^2C-10ab^4C\right)\right)\text{Subst}\left(\int\frac{1}{a+x}\,dx,x,b\tan(c+dx)\right)}{b^4\left(a^2+b^2\right)^3 d}$$

$$= \frac{\left(a^3B-3ab^2B+3a^2bC-b^3C\right)x}{\left(a^2+b^2\right)^3} + \frac{\left(3a^2bB-b^3B-a^3C+3ab^2C\right)\log(\cos(c+dx))}{\left(a^2+b^2\right)^3 d} + \frac{a^2\left(a^4bB+3a^2b^3B+6b^5B-3a^5C-9a^3b^2C-10ab^4C\right)\log(a+b\tan(c+dx))}{b^4\left(a^2+b^2\right)^3 d} - \frac{\left(a^3bB+3ab^3B-3a^4C-6a^2b^2C-b^4C\right)\tan(c+dx)}{b^3\left(a^2+b^2\right)^2 d} + \frac{a(bB-aC)\tan^3(c+dx)}{2b\left(a^2+b^2\right)d(a+b\tan(c+dx))^2} + \frac{a\left(a^2bB+5b^3B-3a^3C-7ab^2C\right)\tan^2(c+dx)}{2b^2\left(a^2+b^2\right)^2 d(a+b\tan(c+dx))}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 5.21 (sec) , antiderivative size = 275, normalized size of antiderivative = 0.83

$$\int \frac{\tan^3(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx$$
$$= \frac{\frac{(B+iC)\log(i-\tan(c+dx))}{(-ia+b)^3}+\frac{(B-iC)\log(i+\tan(c+dx))}{(ia+b)^3}+\frac{2a^2(a^4bB+3a^2b^3B+6b^5B-3a^5C-9a^3b^2C-10ab^4C)\log(a+b\tan(c+dx))}{b^4(a^2+b^2)^3}+\frac{a^3(}{b^4(a^2}}{2d}$$

```
[In] Integrate[(Tan[c + d*x]^3*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c
 + d*x])^3,x]
```

```
[Out] (((B + I*C)*Log[I - Tan[c + d*x]])/((-I)*a + b)^3 + ((B - I*C)*Log[I + Tan[
c + d*x]])/(I*a + b)^3 + (2*a^2*(a^4*b*B + 3*a^2*b^3*B + 6*b^5*B - 3*a^5*C
- 9*a^3*b^2*C - 10*a*b^4*C)*Log[a + b*Tan[c + d*x]])/(b^4*(a^2 + b^2)^3) +
(a^3*(-(a*b*B) + 3*a^2*C + 2*b^2*C))/(b^4*(a^2 + b^2)*(a + b*Tan[c + d*x])^
2) + (2*C*Tan[c + d*x]^3)/(b*(a + b*Tan[c + d*x])^2) - (2*a^2*(-2*a^3*b*B -
 4*a*b^3*B + 6*a^4*C + 11*a^2*b^2*C + 3*b^4*C))/(b^4*(a^2 + b^2)^2*(a + b*T
an[c + d*x])))/(2*d)
```

Maple [A] (verified)

Time = 0.22 (sec) , antiderivative size = 263, normalized size of antiderivative = 0.79

method	result
derivativedivides	$\frac{\frac{\tan(dx+c)C}{b^3}+\frac{\frac{\left(-3Ba^2b+Bb^3+Ca^3-3Cab^2\right)\ln\left(1+\tan(dx+c)^2\right)}{2}+\left(Ba^3-3Bab^2+3Ca^2b-Cb^3\right)\arctan(\tan(dx+c))}{\left(a^2+b^2\right)^3}+\frac{a^2\left(Ba^4b+3\right.}{}}{d}$
default	$\frac{\frac{\tan(dx+c)C}{b^3}+\frac{\frac{\left(-3Ba^2b+Bb^3+Ca^3-3Cab^2\right)\ln\left(1+\tan(dx+c)^2\right)}{2}+\left(Ba^3-3Bab^2+3Ca^2b-Cb^3\right)\arctan(\tan(dx+c))}{\left(a^2+b^2\right)^3}+\frac{a^2\left(Ba^4b+3\right.}{}}{d}$
norman	$\frac{\frac{C\tan(dx+c)^3}{bd}+\frac{\left(Ba^3-3Bab^2+3Ca^2b-Cb^3\right)a^2x}{\left(a^4+2a^2b^2+b^4\right)\left(a^2+b^2\right)}+\frac{b^2\left(Ba^3-3Bab^2+3Ca^2b-Cb^3\right)x\tan(dx+c)^2}{\left(a^4+2a^2b^2+b^4\right)\left(a^2+b^2\right)}+\frac{a\left(2Ba^4b+4Ba^2b^3-6Ca^5-11\right.}{db^3\left(a^4+2a^2\right.}}{(a+b\tan(dx}$
parallelrisch	Expression too large to display
risch	Expression too large to display

```
[In] int(tan(d*x+c)^3*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x,method=
_RETURNVERBOSE)
```

```
[Out] 1/d*(tan(d*x+c)*C/b^3+1/(a^2+b^2)^3*(1/2*(-3*B*a^2*b+B*b^3+C*a^3-3*C*a*b^2)
*ln(1+tan(d*x+c)^2)+(B*a^3-3*B*a*b^2+3*C*a^2*b-C*b^3)*arctan(tan(d*x+c)))+1
/b^4*a^2*(B*a^4*b+3*B*a^2*b^3+6*B*b^5-3*C*a^5-9*C*a^3*b^2-10*C*a*b^4)/(a^2+
```

```
b^2)^3*ln(a+b*tan(d*x+c))-1/2/b^4*a^4*(B*b-C*a)/(a^2+b^2)/(a+b*tan(d*x+c))^
2+1/b^4*a^3*(2*B*a^2*b+4*B*b^3-3*C*a^3-5*C*a*b^2)/(a^2+b^2)^2/(a+b*tan(d*x+
c)))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 890 vs. $2(328) = 656$.

Time = 0.34 (sec) , antiderivative size = 890, normalized size of antiderivative = 2.69

$$\int \frac{\tan^3(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx =$$

$$-\frac{3\,Ca^7b^2 - Ba^6b^3 + 9\,Ca^5b^4 - 7\,Ba^4b^5 - 2\left(Ca^6b^3 + 3\,Ca^4b^5 + 3\,Ca^2b^7 + Cb^9\right)\tan\left(dx+c\right)^3 - 2\left(Ba^5b^4\right.}{}$$

```
[In] integrate(tan(d*x+c)^3*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x,
algorithm="fricas")
```

```
[Out] -1/2*(3*C*a^7*b^2 - B*a^6*b^3 + 9*C*a^5*b^4 - 7*B*a^4*b^5 - 2*(C*a^6*b^3 +
3*C*a^4*b^5 + 3*C*a^2*b^7 + C*b^9)*tan(d*x + c)^3 - 2*(B*a^5*b^4 + 3*C*a^4*
b^5 - 3*B*a^3*b^6 - C*a^2*b^7)*d*x - (9*C*a^7*b^2 - 3*B*a^6*b^3 + 23*C*a^5*
b^4 - 9*B*a^4*b^5 + 12*C*a^3*b^6 + 4*C*a*b^8 + 2*(B*a^3*b^6 + 3*C*a^2*b^7 -
 3*B*a*b^8 - C*b^9)*d*x)*tan(d*x + c)^2 + (3*C*a^9 - B*a^8*b + 9*C*a^7*b^2
- 3*B*a^6*b^3 + 10*C*a^5*b^4 - 6*B*a^4*b^5 + (3*C*a^7*b^2 - B*a^6*b^3 + 9*C
*a^5*b^4 - 3*B*a^4*b^5 + 10*C*a^3*b^6 - 6*B*a^2*b^7)*tan(d*x + c)^2 + 2*(3*
C*a^8*b - B*a^7*b^2 + 9*C*a^6*b^3 - 3*B*a^5*b^4 + 10*C*a^4*b^5 - 6*B*a^3*b^
6)*tan(d*x + c))*log((b^2*tan(d*x + c)^2 + 2*a*b*tan(d*x + c) + a^2)/(tan(d
*x + c)^2 + 1)) - (3*C*a^9 - B*a^8*b + 9*C*a^7*b^2 - 3*B*a^6*b^3 + 9*C*a^5*
b^4 - 3*B*a^4*b^5 + 3*C*a^3*b^6 - B*a^2*b^7 + (3*C*a^7*b^2 - B*a^6*b^3 + 9*
C*a^5*b^4 - 3*B*a^4*b^5 + 9*C*a^3*b^6 - 3*B*a^2*b^7 + 3*C*a*b^8 - B*b^9)*ta
n(d*x + c)^2 + 2*(3*C*a^8*b - B*a^7*b^2 + 9*C*a^6*b^3 - 3*B*a^5*b^4 + 9*C*a
^4*b^5 - 3*B*a^3*b^6 + 3*C*a^2*b^7 - B*a*b^8)*tan(d*x + c))*log(1/(tan(d*x
+ c)^2 + 1)) - 2*(3*C*a^8*b - B*a^7*b^2 + 6*C*a^6*b^3 - 3*B*a^5*b^4 - 2*C*a
^4*b^5 + 4*B*a^3*b^6 + C*a^2*b^7 + 2*(B*a^4*b^5 + 3*C*a^3*b^6 - 3*B*a^2*b^7
 - C*a*b^8)*d*x)*tan(d*x + c))/((a^6*b^6 + 3*a^4*b^8 + 3*a^2*b^10 + b^12)*d
*tan(d*x + c)^2 + 2*(a^7*b^5 + 3*a^5*b^7 + 3*a^3*b^9 + a*b^11)*d*tan(d*x +
c) + (a^8*b^4 + 3*a^6*b^6 + 3*a^4*b^8 + a^2*b^10)*d)
```

Sympy [F(-2)]

Exception generated.

$$\int \frac{\tan^3(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx = \text{Exception raised: AttributeError}$$

```
[In] integrate(tan(d*x+c)**3*(B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c))**3,
x)
```

```
[Out] Exception raised: AttributeError >> 'NoneType' object has no attribute 'pri
mitive'
```

Maxima [A] (verification not implemented)

none

Time = 0.34 (sec) , antiderivative size = 389, normalized size of antiderivative = 1.18

$$\int \frac{\tan^3(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx$$
$$= \frac{\frac{2\,(Ba^3+3\,Ca^2b-3\,Bab^2-Cb^3)(dx+c)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6} - \frac{2\,(3\,Ca^7-Ba^6b+9\,Ca^5b^2-3\,Ba^4b^3+10\,Ca^3b^4-6\,Ba^2b^5)\log(b\tan(dx+c)+a)}{a^6b^4+3\,a^4b^6+3\,a^2b^8+b^{10}} + \frac{(Ca^3-3\,Ba^2b-3\,Cab}{a^6+3\,a}}{2\,d}$$

```
[In] integrate(tan(d*x+c)^3*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x,
algorithm="maxima")
```

```
[Out] 1/2*(2*(B*a^3 + 3*C*a^2*b - 3*B*a*b^2 - C*b^3)*(d*x + c)/(a^6 + 3*a^4*b^2 +
 3*a^2*b^4 + b^6) - 2*(3*C*a^7 - B*a^6*b + 9*C*a^5*b^2 - 3*B*a^4*b^3 + 10*C
*a^3*b^4 - 6*B*a^2*b^5)*log(b*tan(d*x + c) + a)/(a^6*b^4 + 3*a^4*b^6 + 3*a^
2*b^8 + b^10) + (C*a^3 - 3*B*a^2*b - 3*C*a*b^2 + B*b^3)*log(tan(d*x + c)^2
+ 1)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) - (5*C*a^7 - 3*B*a^6*b + 9*C*a^5*b
^2 - 7*B*a^4*b^3 + 2*(3*C*a^6*b - 2*B*a^5*b^2 + 5*C*a^4*b^3 - 4*B*a^3*b^4)*
tan(d*x + c))/(a^6*b^4 + 2*a^4*b^6 + a^2*b^8 + (a^4*b^6 + 2*a^2*b^8 + b^10)
*tan(d*x + c)^2 + 2*(a^5*b^5 + 2*a^3*b^7 + a*b^9)*tan(d*x + c)) + 2*C*tan(d
*x + c)/b^3)/d
```

Giac [A] (verification not implemented)

none

Time = 1.02 (sec) , antiderivative size = 505, normalized size of antiderivative = 1.53

$$\int \frac{\tan^3(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^3}\,dx$$
$$= \frac{\frac{2\,(Ba^3+3\,Ca^2b-3\,Bab^2-Cb^3)(dx+c)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6} + \frac{(Ca^3-3\,Ba^2b-3\,Cab^2+Bb^3)\log\left(\tan(dx+c)^2+1\right)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6} - \frac{2\,(3\,Ca^7-Ba^6b+9\,Ca^5b^2-3\,Ba^4b^3+10\,Ca^3b^4}{a^6b^4+3\,a^4b^6+3\,a^2b}}{}$$

```
[In] integrate(tan(d*x+c)^3*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x,
algorithm="giac")
```

```
[Out] 1/2*(2*(B*a^3 + 3*C*a^2*b - 3*B*a*b^2 - C*b^3)*(d*x + c)/(a^6 + 3*a^4*b^2 +
 3*a^2*b^4 + b^6) + (C*a^3 - 3*B*a^2*b - 3*C*a*b^2 + B*b^3)*log(tan(d*x + c
)^2 + 1)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) - 2*(3*C*a^7 - B*a^6*b + 9*C*a
^5*b^2 - 3*B*a^4*b^3 + 10*C*a^3*b^4 - 6*B*a^2*b^5)*log(abs(b*tan(d*x + c) +
 a))/(a^6*b^4 + 3*a^4*b^6 + 3*a^2*b^8 + b^10) + 2*C*tan(d*x + c)/b^3 + (9*C
*a^7*b^2*tan(d*x + c)^2 - 3*B*a^6*b^3*tan(d*x + c)^2 + 27*C*a^5*b^4*tan(d*x
 + c)^2 - 9*B*a^4*b^5*tan(d*x + c)^2 + 30*C*a^3*b^6*tan(d*x + c)^2 - 18*B*a
^2*b^7*tan(d*x + c)^2 + 12*C*a^8*b*tan(d*x + c) - 2*B*a^7*b^2*tan(d*x + c)
+ 38*C*a^6*b^3*tan(d*x + c) - 6*B*a^5*b^4*tan(d*x + c) + 50*C*a^4*b^5*tan(d
*x + c) - 28*B*a^3*b^6*tan(d*x + c) + 4*C*a^9 + 13*C*a^7*b^2 + B*a^6*b^3 +
21*C*a^5*b^4 - 11*B*a^4*b^5)/((a^6*b^4 + 3*a^4*b^6 + 3*a^2*b^8 + b^10)*(b*t
an(d*x + c) + a)^2))/d
```

Mupad [B] (verification not implemented)

Time = 9.47 (sec) , antiderivative size = 335, normalized size of antiderivative = 1.01

$$\int \frac{\tan^3(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^3}\,dx$$
$$= \frac{C\tan(c+d\,x)}{b^3\,d} + \frac{\ln\left(\tan(c+d\,x)-\mathrm{i}\right)\,(-C+B\,1\mathrm{i})}{2\,d\,(-a^3-a^2\,b\,3\mathrm{i}+3\,a\,b^2+b^3\,1\mathrm{i})} + \frac{\ln\left(\tan(c+d\,x)+1\mathrm{i}\right)\,(B-C\,1\mathrm{i})}{2\,d\,(-a^3\,1\mathrm{i}-3\,a^2\,b+a\,b^2\,3\mathrm{i}+b^3)}$$
$$- \frac{\frac{5\,C\,a^7-3\,B\,a^6\,b+9\,C\,a^5\,b^2-7\,B\,a^4\,b^3}{2\,b\,(a^4+2\,a^2\,b^2+b^4)} + \frac{\tan(c+d\,x)\,(3\,C\,a^6-2\,B\,a^5\,b+5\,C\,a^4\,b^2-4\,B\,a^3\,b^3)}{a^4+2\,a^2\,b^2+b^4}}{d\,\left(a^2\,b^3+2\,a\,b^4\tan\left(c+d\,x\right)+b^5\tan\left(c+d\,x\right)^2\right)}$$
$$+ \frac{a^2\,\ln\left(a+b\tan(c+d\,x)\right)\,(-3\,C\,a^5+B\,a^4\,b-9\,C\,a^3\,b^2+3\,B\,a^2\,b^3-10\,C\,a\,b^4+6\,B\,b^5)}{b^4\,d\,\left(a^2+b^2\right)^3}$$

```
[In] int((tan(c + d*x)^3*(B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c + d*x
))^3,x)
```

```
[Out] (log(tan(c + d*x) - 1i)*(B*1i - C))/(2*d*(3*a*b^2 - a^2*b*3i - a^3 + b^3*1i
)) - ((5*C*a^7 - 7*B*a^4*b^3 + 9*C*a^5*b^2 - 3*B*a^6*b)/(2*b*(a^4 + b^4 + 2
*a^2*b^2)) + (tan(c + d*x)*(3*C*a^6 - 4*B*a^3*b^3 + 5*C*a^4*b^2 - 2*B*a^5*b
))/(a^4 + b^4 + 2*a^2*b^2))/(d*(a^2*b^3 + b^5*tan(c + d*x)^2 + 2*a*b^4*tan(
c + d*x))) + (log(tan(c + d*x) + 1i)*(B - C*1i))/(2*d*(a*b^2*3i - 3*a^2*b -
 a^3*1i + b^3)) + (C*tan(c + d*x))/(b^3*d) + (a^2*log(a + b*tan(c + d*x))*(
6*B*b^5 - 3*C*a^5 + 3*B*a^2*b^3 - 9*C*a^3*b^2 + B*a^4*b - 10*C*a*b^4))/(b^4
*d*(a^2 + b^2)^3)
```

3.39 $\int \frac{\tan^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx$

Optimal result . . . 335
Rubi [A] (verified) . . . 335
Mathematica [C] (verified) . . . 339
Maple [A] (verified) . . . 339
Fricas [B] (verification not implemented) . . . 340
Sympy [F(-2)] . . . 341
Maxima [A] (verification not implemented) . . . 341
Giac [A] (verification not implemented) . . . 341
Mupad [B] (verification not implemented) . . . 342

Optimal result

Integrand size = 40, antiderivative size = 250

$$\begin{aligned}&\int \frac{\tan^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx\\&=-\frac{(3a^2bB-b^3B-a^3C+3ab^2C)\,x}{(a^2+b^2)^3}+\frac{(a^3B-3ab^2B+3a^2bC-b^3C)\log(\cos(c+dx))}{(a^2+b^2)^3\,d}\\&\quad+\frac{a(a^2b^3B-3b^5B+a^5C+3a^3b^2C+6ab^4C)\log(a+b\tan(c+dx))}{b^3\,(a^2+b^2)^3\,d}\\&\quad+\frac{a(bB-aC)\tan^2(c+dx)}{2b\,(a^2+b^2)\,d(a+b\tan(c+dx))^2}-\frac{a^2(2b^3B-a^3C-3ab^2C)}{b^3\,(a^2+b^2)^2\,d(a+b\tan(c+dx))}\end{aligned}$$

```
[Out] -(3*B*a^2*b-B*b^3-C*a^3+3*C*a*b^2)*x/(a^2+b^2)^3+(B*a^3-3*B*a*b^2+3*C*a^2*b
-C*b^3)*ln(cos(d*x+c))/(a^2+b^2)^3/d+a*(B*a^2*b^3-3*B*b^5+C*a^5+3*C*a^3*b^2
+6*C*a*b^4)*ln(a+b*tan(d*x+c))/b^3/(a^2+b^2)^3/d+1/2*a*(B*b-C*a)*tan(d*x+c)
^2/b/(a^2+b^2)/d/(a+b*tan(d*x+c))^2-a^2*(2*B*b^3-C*a^3-3*C*a*b^2)/b^3/(a^2+
b^2)^2/d/(a+b*tan(d*x+c))
```

Rubi [A] (verified)

Time = 0.64 (sec) , antiderivative size = 250, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 7, $\frac{\text{number of rules}}{\text{integrand size}} = 0.175$, Rules used

$= \{3713, 3686, 3716, 3707, 3698, 31, 3556\}$

$$\int \frac{\tan^2(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^3}\,dx$$
$$= \frac{a(bB-aC)\tan^2(c+dx)}{2bd\,(a^2+b^2)\,(a+b\tan(c+dx))^2} - \frac{a^2(a^3(-C)-3ab^2C+2b^3B)}{b^3d\,(a^2+b^2)^2\,(a+b\tan(c+dx))}$$
$$+ \frac{(a^3B+3a^2bC-3ab^2B-b^3C)\log(\cos(c+dx))}{d\,(a^2+b^2)^3}$$
$$- \frac{x(a^3(-C)+3a^2bB+3ab^2C-b^3B)}{(a^2+b^2)^3}$$
$$+ \frac{a(a^5C+3a^3b^2C+a^2b^3B+6ab^4C-3b^5B)\log(a+b\tan(c+dx))}{b^3d\,(a^2+b^2)^3}$$

```
[In] Int[(Tan[c + d*x]^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c + d*x
])^3,x]
```

```
[Out] -(((3*a^2*b*B - b^3*B - a^3*C + 3*a*b^2*C)*x)/(a^2 + b^2)^3) + ((a^3*B - 3*
a*b^2*B + 3*a^2*b*C - b^3*C)*Log[Cos[c + d*x]])/((a^2 + b^2)^3*d) + (a*(a^2
*b^3*B - 3*b^5*B + a^5*C + 3*a^3*b^2*C + 6*a*b^4*C)*Log[a + b*Tan[c + d*x]]
)/(b^3*(a^2 + b^2)^3*d) + (a*(b*B - a*C)*Tan[c + d*x]^2)/(2*b*(a^2 + b^2)*d
*(a + b*Tan[c + d*x])^2) - (a^2*(2*b^3*B - a^3*C - 3*a*b^2*C))/(b^3*(a^2 +
b^2)^2*d*(a + b*Tan[c + d*x]))
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3686

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Si
mp[(b*c - a*d)*(B*c - A*d)*(a + b*Tan[e + f*x])^(m - 1)*((c + d*Tan[e + f*x
])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dist[1/(d*(n + 1)*(c^2 + d^2)),
 Int[(a + b*Tan[e + f*x])^(m - 2)*(c + d*Tan[e + f*x])^(n + 1)*Simp[a*A*d*(
b*d*(m - 1) - a*c*(n + 1)) + (b*B*c - (A*b + a*B)*d)*(b*c*(m - 1) + a*d*(n
+ 1)) - d*((a*A - b*B)*(b*c - a*d) + (A*b + a*B)*(a*c + b*d))*(n + 1)*Tan[e
 + f*x] - b*(d*(A*b*c + a*B*c - a*A*d)*(m + n) - b*B*(c^2*(m - 1) - d^2*(n
+ 1)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] &&
NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 1] &&
```

```
LtQ[n, -1] && (IntegerQ[m] || IntegersQ[2*m, 2*n])
```

Rule 3698

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3707

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[(a*A + b*B - a
*C)*(x/(a^2 + b^2)), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rule 3716

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.
)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(-(b*c - a*d))*(c^2*C - B*c*d + A*d^2)*((c
+ d*Tan[e + f*x])^(n + 1)/(d^2*f*(n + 1)*(c^2 + d^2))), x] + Dist[1/(d*(c^2
 + d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)*Simp[a*d*(A*c - c*C + B*d) + b*(
c^2*C - B*c*d + A*d^2) + d*(A*b*c + a*B*c - b*c*C - a*A*d + b*B*d + a*C*d)*
Tan[e + f*x] + b*C*(c^2 + d^2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c
, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] && LtQ[n,
 -1]
```

Rubi steps

$$\text{integral} = \int \frac{\tan^3(c+dx)(B+C\tan(c+dx))}{(a+b\tan(c+dx))^3}\,dx$$

$$
\begin{aligned}
&= \frac{a(bB-aC)\tan^2(c+dx)}{2b(a^2+b^2)\,d(a+b\tan(c+dx))^2} \\
&\quad+ \frac{\int \frac{\tan(c+dx)(-2a(bB-aC)+2b(bB-aC)\tan(c+dx)+2(a^2+b^2)C\tan^2(c+dx))}{(a+b\tan(c+dx))^2}\,dx}{2b(a^2+b^2)} \\
&= \frac{a(bB-aC)\tan^2(c+dx)}{2b(a^2+b^2)\,d(a+b\tan(c+dx))^2} - \frac{a^2(2b^3B-a^3C-3ab^2C)}{b^3(a^2+b^2)^2\,d(a+b\tan(c+dx))} \\
&\quad+ \frac{\int \frac{-2a(2b^3B-a^3C-3ab^2C)-2b^2(a^2B-b^2B+2abC)\tan(c+dx)+2(a^2+b^2)^2C\tan^2(c+dx)}{a+b\tan(c+dx)}\,dx}{2b^2(a^2+b^2)^2} \\
&= -\frac{(3a^2bB-b^3B-a^3C+3ab^2C)\,x}{(a^2+b^2)^3} + \frac{a(bB-aC)\tan^2(c+dx)}{2b(a^2+b^2)\,d(a+b\tan(c+dx))^2} \\
&\quad- \frac{a^2(2b^3B-a^3C-3ab^2C)}{b^3(a^2+b^2)^2\,d(a+b\tan(c+dx))} \\
&\quad- \frac{(a^3B-3ab^2B+3a^2bC-b^3C)\int\tan(c+dx)\,dx}{(a^2+b^2)^3} \\
&\quad+ \frac{(a(a^2b^3B-3b^5B+a^5C+3a^3b^2C+6ab^4C))\int\frac{1+\tan^2(c+dx)}{a+b\tan(c+dx)}\,dx}{b^2(a^2+b^2)^3} \\
&= -\frac{(3a^2bB-b^3B-a^3C+3ab^2C)\,x}{(a^2+b^2)^3} + \frac{(a^3B-3ab^2B+3a^2bC-b^3C)\log(\cos(c+dx))}{(a^2+b^2)^3\,d} \\
&\quad+ \frac{a(bB-aC)\tan^2(c+dx)}{2b(a^2+b^2)\,d(a+b\tan(c+dx))^2} - \frac{a^2(2b^3B-a^3C-3ab^2C)}{b^3(a^2+b^2)^2\,d(a+b\tan(c+dx))} \\
&\quad+ \frac{(a(a^2b^3B-3b^5B+a^5C+3a^3b^2C+6ab^4C))\,\mathrm{Subst}\left(\int\frac{1}{a+x}\,dx,x,b\tan(c+dx)\right)}{b^3(a^2+b^2)^3\,d} \\
&= -\frac{(3a^2bB-b^3B-a^3C+3ab^2C)\,x}{(a^2+b^2)^3} \\
&\quad+ \frac{(a^3B-3ab^2B+3a^2bC-b^3C)\log(\cos(c+dx))}{(a^2+b^2)^3\,d} \\
&\quad+ \frac{a(a^2b^3B-3b^5B+a^5C+3a^3b^2C+6ab^4C)\log(a+b\tan(c+dx))}{b^3(a^2+b^2)^3\,d} \\
&\quad+ \frac{a(bB-aC)\tan^2(c+dx)}{2b(a^2+b^2)\,d(a+b\tan(c+dx))^2} - \frac{a^2(2b^3B-a^3C-3ab^2C)}{b^3(a^2+b^2)^2\,d(a+b\tan(c+dx))}
\end{aligned}
$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 6.39 (sec) , antiderivative size = 235, normalized size of antiderivative = 0.94

$$
\begin{aligned}
&\int \frac{\tan^2(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^3}\,dx \\
&= -\frac{(B+iC)\log(i-\tan(c+dx))}{2(a+ib)^3d}-\frac{(B-iC)\log(i+\tan(c+dx))}{2(a-ib)^3d} \\
&\quad+\frac{a(a^2b^3B-3b^5B+a^5C+3a^3b^2C+6ab^4C)\log(a+b\tan(c+dx))}{b^3\,(a^2+b^2)^3\,d} \\
&\quad+\frac{a^3(bB-aC)}{2b^3\,(a^2+b^2)\,d(a+b\tan(c+dx))^2}-\frac{a^2(a^2bB+3b^3B-2a^3C-4ab^2C)}{b^3\,(a^2+b^2)^2\,d(a+b\tan(c+dx))}
\end{aligned}
$$

```
[In] Integrate[(Tan[c + d*x]^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c
 + d*x])^3,x]
```

```
[Out] -1/2*((B + I*C)*Log[I - Tan[c + d*x]])/((a + I*b)^3*d) - ((B - I*C)*Log[I +
 Tan[c + d*x]])/(2*(a - I*b)^3*d) + (a*(a^2*b^3*B - 3*b^5*B + a^5*C + 3*a^3
*b^2*C + 6*a*b^4*C)*Log[a + b*Tan[c + d*x]])/(b^3*(a^2 + b^2)^3*d) + (a^3*(
b*B - a*C))/(2*b^3*(a^2 + b^2)*d*(a + b*Tan[c + d*x])^2) - (a^2*(a^2*b*B +
3*b^3*B - 2*a^3*C - 4*a*b^2*C))/(b^3*(a^2 + b^2)^2*d*(a + b*Tan[c + d*x]))
```

Maple [A] (verified)

Time = 0.15 (sec) , antiderivative size = 242, normalized size of antiderivative = 0.97

method	result
derivativedivides	$\frac{\frac{\frac{\left(-B\,a^3+3Ba\,b^2-3C\,a^2b+C\,b^3\right)\ln\left(1+\tan(dx+c)^2\right)}{2}+\left(-3B\,a^2b+B\,b^3+C\,a^3-3Ca\,b^2\right)\arctan(\tan(dx+c))}{\left(a^2+b^2\right)^3}+\frac{a\left(B\,a^2b^3-3B\,b^5+C\,a^5 \cdots\right.}{\cdots}}{d}$
default	$\frac{\frac{\frac{\left(-B\,a^3+3Ba\,b^2-3C\,a^2b+C\,b^3\right)\ln\left(1+\tan(dx+c)^2\right)}{2}+\left(-3B\,a^2b+B\,b^3+C\,a^3-3Ca\,b^2\right)\arctan(\tan(dx+c))}{\left(a^2+b^2\right)^3}+\frac{a\left(B\,a^2b^3-3B\,b^5+C\,a^5 \cdots\right.}{\cdots}}{d}$
norman	$\frac{-\frac{a^2\left(B\,a^3b+5Ba\,b^3-3C\,a^4-7C\,a^2b^2\right)}{2d\,b^3\left(a^4+2a^2b^2+b^4\right)}-\frac{\left(3B\,a^2b-B\,b^3-C\,a^3+3Ca\,b^2\right)a^2x}{\left(a^4+2a^2b^2+b^4\right)\left(a^2+b^2\right)}-\frac{b^2\left(3B\,a^2b-B\,b^3-C\,a^3+3Ca\,b^2\right)x\tan(dx+c)^2}{\left(a^4+2a^2b^2+b^4\right)\left(a^2+b^2\right)}-\frac{a\left(B \cdots\right.}{\cdots}}{(a+b\tan(dx+c))^2}$
risch	$-\frac{6ia^4Cx}{(a^6+3a^4b^2+3a^2b^4+b^6)b}-\frac{xC}{3ia^2b-ib^3-a^3+3a\,b^2}+\frac{2iCc}{d\,b^3}+\frac{2iCx}{b^3}-\frac{2ia^3Bc}{(a^6+3a^4b^2+3a^2b^4+b^6)d}-\frac{2i \cdots}{(a^6+3a^4b^2+ \cdots}$
parallelrisch	Expression too large to display

```
[In] int(tan(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x,method=
_RETURNVERBOSE)
```

```
[Out] 1/d*(1/(a^2+b^2)^3*(1/2*(-B*a^3+3*B*a*b^2-3*C*a^2*b+C*b^3)*ln(1+tan(d*x+c)^
2)+(-3*B*a^2*b+B*b^3+C*a^3-3*C*a*b^2)*arctan(tan(d*x+c)))+a*(B*a^2*b^3-3*B*
b^5+C*a^5+3*C*a^3*b^2+6*C*a*b^4)/(a^2+b^2)^3/b^3*ln(a+b*tan(d*x+c))-a^2*(B*
a^2*b+3*B*b^3-2*C*a^3-4*C*a*b^2)/b^3/(a^2+b^2)^2/(a+b*tan(d*x+c))+1/2*a^3*(
B*b-C*a)/b^3/(a^2+b^2)/(a+b*tan(d*x+c))^2)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 666 vs. $2(243) = 486$.

Time = 0.31 (sec) , antiderivative size = 666, normalized size of antiderivative = 2.66

$$\int \frac{\tan^2(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^3}\,dx = \frac{Ca^6b^2+Ba^5b^3+7\,Ca^4b^4-5\,Ba^3b^5+2\,(Ca^5b^3-3\,Ba^4b^4-3\,Ca^3b^5+Ba^2b^6)dx-(3\,Ca^6b^2-Ba^5b^3+9\,C}{}$$

```
[In] integrate(tan(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x,
algorithm="fricas")
```

```
[Out] 1/2*(C*a^6*b^2 + B*a^5*b^3 + 7*C*a^4*b^4 - 5*B*a^3*b^5 + 2*(C*a^5*b^3 - 3*B
*a^4*b^4 - 3*C*a^3*b^5 + B*a^2*b^6)*d*x - (3*C*a^6*b^2 - B*a^5*b^3 + 9*C*a^
4*b^4 - 7*B*a^3*b^5 - 2*(C*a^3*b^5 - 3*B*a^2*b^6 - 3*C*a*b^7 + B*b^8)*d*x)*
tan(d*x + c)^2 + (C*a^8 + 3*C*a^6*b^2 + B*a^5*b^3 + 6*C*a^4*b^4 - 3*B*a^3*b
^5 + (C*a^6*b^2 + 3*C*a^4*b^4 + B*a^3*b^5 + 6*C*a^2*b^6 - 3*B*a*b^7)*tan(d*
x + c)^2 + 2*(C*a^7*b + 3*C*a^5*b^3 + B*a^4*b^4 + 6*C*a^3*b^5 - 3*B*a^2*b^6
)*tan(d*x + c))*log((b^2*tan(d*x + c)^2 + 2*a*b*tan(d*x + c) + a^2)/(tan(d*
x + c)^2 + 1)) - (C*a^8 + 3*C*a^6*b^2 + 3*C*a^4*b^4 + C*a^2*b^6 + (C*a^6*b^
2 + 3*C*a^4*b^4 + 3*C*a^2*b^6 + C*b^8)*tan(d*x + c)^2 + 2*(C*a^7*b + 3*C*a^
5*b^3 + 3*C*a^3*b^5 + C*a*b^7)*tan(d*x + c))*log(1/(tan(d*x + c)^2 + 1)) -
2*(C*a^7*b + 3*C*a^5*b^3 - 3*B*a^4*b^4 - 4*C*a^3*b^5 + 3*B*a^2*b^6 - 2*(C*a
^4*b^4 - 3*B*a^3*b^5 - 3*C*a^2*b^6 + B*a*b^7)*d*x)*tan(d*x + c))/((a^6*b^5
+ 3*a^4*b^7 + 3*a^2*b^9 + b^11)*d*tan(d*x + c)^2 + 2*(a^7*b^4 + 3*a^5*b^6 +
 3*a^3*b^8 + a*b^10)*d*tan(d*x + c) + (a^8*b^3 + 3*a^6*b^5 + 3*a^4*b^7 + a^
2*b^9)*d)
```

Sympy [F(-2)]

Exception generated.

$$\int \frac{\tan^2(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^3}\,dx = \text{Exception raised: AttributeError}$$

```
[In] integrate(tan(d*x+c)**2*(B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c))**3,
x)
```

```
[Out] Exception raised: AttributeError >> 'NoneType' object has no attribute 'pri
mitive'
```

Maxima [A] (verification not implemented)

none

Time = 0.37 (sec) , antiderivative size = 366, normalized size of antiderivative = 1.46

$$\int \frac{\tan^2(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^3}\,dx$$
$$= \frac{\frac{2\,(Ca^3-3\,Ba^2b-3\,Cab^2+Bb^3)(dx+c)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6} + \frac{2\,(Ca^6+3\,Ca^4b^2+Ba^3b^3+6\,Ca^2b^4-3\,Bab^5)\log(b\tan(dx+c)+a)}{a^6b^3+3\,a^4b^5+3\,a^2b^7+b^9} - \frac{(Ba^3+3\,Ca^2b-3\,Bab^2-Cb^3)\log\ldots}{a^6+3\,a^4b^2+3\,a^2b^4\ldots}}{2\,d}$$

```
[In] integrate(tan(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x,
algorithm="maxima")
```

```
[Out] 1/2*(2*(C*a^3 - 3*B*a^2*b - 3*C*a*b^2 + B*b^3)*(d*x + c)/(a^6 + 3*a^4*b^2 +
 3*a^2*b^4 + b^6) + 2*(C*a^6 + 3*C*a^4*b^2 + B*a^3*b^3 + 6*C*a^2*b^4 - 3*B*
a*b^5)*log(b*tan(d*x + c) + a)/(a^6*b^3 + 3*a^4*b^5 + 3*a^2*b^7 + b^9) - (B
*a^3 + 3*C*a^2*b - 3*B*a*b^2 - C*b^3)*log(tan(d*x + c)^2 + 1)/(a^6 + 3*a^4*
b^2 + 3*a^2*b^4 + b^6) + (3*C*a^6 - B*a^5*b + 7*C*a^4*b^2 - 5*B*a^3*b^3 + 2
*(2*C*a^5*b - B*a^4*b^2 + 4*C*a^3*b^3 - 3*B*a^2*b^4)*tan(d*x + c))/(a^6*b^3
 + 2*a^4*b^5 + a^2*b^7 + (a^4*b^5 + 2*a^2*b^7 + b^9)*tan(d*x + c)^2 + 2*(a^
5*b^4 + 2*a^3*b^6 + a*b^8)*tan(d*x + c)))/d
```

Giac [A] (verification not implemented)

none

Time = 0.83 (sec) , antiderivative size = 458, normalized size of antiderivative = 1.83

$$\int \frac{\tan^2(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^3}\,dx$$
$$= \frac{\frac{2\,(Ca^3-3\,Ba^2b-3\,Cab^2+Bb^3)(dx+c)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6} - \frac{(Ba^3+3\,Ca^2b-3\,Bab^2-Cb^3)\log\left(\tan(dx+c)^2+1\right)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6} + \frac{2\,(Ca^6+3\,Ca^4b^2+Ba^3b^3+6\,Ca^2b^4-3\,Bab^5)\log\ldots}{a^6b^3+3\,a^4b^5+3\,a^2b^7+b^9}\ldots}{\ldots}$$

```
[In] integrate(tan(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x,
algorithm="giac")
```

```
[Out] 1/2*(2*(C*a^3 - 3*B*a^2*b - 3*C*a*b^2 + B*b^3)*(d*x + c)/(a^6 + 3*a^4*b^2 +
 3*a^2*b^4 + b^6) - (B*a^3 + 3*C*a^2*b - 3*B*a*b^2 - C*b^3)*log(tan(d*x + c
)^2 + 1)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) + 2*(C*a^6 + 3*C*a^4*b^2 + B*a
^3*b^3 + 6*C*a^2*b^4 - 3*B*a*b^5)*log(abs(b*tan(d*x + c) + a))/(a^6*b^3 + 3
*a^4*b^5 + 3*a^2*b^7 + b^9) - (3*C*a^6*b*tan(d*x + c)^2 + 9*C*a^4*b^3*tan(d
*x + c)^2 + 3*B*a^3*b^4*tan(d*x + c)^2 + 18*C*a^2*b^5*tan(d*x + c)^2 - 9*B*
a*b^6*tan(d*x + c)^2 + 2*C*a^7*tan(d*x + c) + 2*B*a^6*b*tan(d*x + c) + 6*C*
a^5*b^2*tan(d*x + c) + 14*B*a^4*b^3*tan(d*x + c) + 28*C*a^3*b^4*tan(d*x + c
) - 12*B*a^2*b^5*tan(d*x + c) + B*a^7 - C*a^6*b + 9*B*a^5*b^2 + 11*C*a^4*b^
3 - 4*B*a^3*b^4)/((a^6*b^2 + 3*a^4*b^4 + 3*a^2*b^6 + b^8)*(b*tan(d*x + c) +
 a)^2))/d
```

Mupad [B] (verification not implemented)

Time = 8.52 (sec) , antiderivative size = 307, normalized size of antiderivative = 1.23

$$\int \frac{\tan^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx = \frac{\frac{3\,C\,a^6-B\,a^5\,b+7\,C\,a^4\,b^2-5\,B\,a^3\,b^3}{2\,b^3\,\left(a^4+2\,a^2\,b^2+b^4\right)}-\frac{a^2\,\tan(c+d\,x)\,\left(-2\,C\,a^3+B\,a^2\,b-4\,C\,a\,b^2+3\,B\,b^3\right)}{b^2\,\left(a^4+2\,a^2\,b^2+b^4\right)}}{d\,\left(a^2+2\,a\,b\,\tan\left(c+d\,x\right)+b^2\,\tan\left(c+d\,x\right)^2\right)} + \frac{\ln\left(\tan(c+d\,x)-\mathrm{i}\right)\,\left(-C+B\,1\mathrm{i}\right)}{2\,d\,\left(-a^3\,1\mathrm{i}+3\,a^2\,b+a\,b^2\,3\mathrm{i}-b^3\right)} + \frac{\ln\left(\tan(c+d\,x)+1\mathrm{i}\right)\,\left(B-C\,1\mathrm{i}\right)}{2\,d\,\left(-a^3+a^2\,b\,3\mathrm{i}+3\,a\,b^2-b^3\,1\mathrm{i}\right)} + \frac{a\,\ln\left(a+b\tan(c+d\,x)\right)\,\left(C\,a^5+3\,C\,a^3\,b^2+B\,a^2\,b^3+6\,C\,a\,b^4-3\,B\,b^5\right)}{b^3\,d\,\left(a^2+b^2\right)^3}$$

```
[In] int((tan(c + d*x)^2*(B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c + d*x
))^3,x)
```

```
[Out] ((3*C*a^6 - 5*B*a^3*b^3 + 7*C*a^4*b^2 - B*a^5*b)/(2*b^3*(a^4 + b^4 + 2*a^2*
b^2)) - (a^2*tan(c + d*x)*(3*B*b^3 - 2*C*a^3 + B*a^2*b - 4*C*a*b^2))/(b^2*(
a^4 + b^4 + 2*a^2*b^2)))/(d*(a^2 + b^2*tan(c + d*x)^2 + 2*a*b*tan(c + d*x))
) + (log(tan(c + d*x) - 1i)*(B*1i - C))/(2*d*(a*b^2*3i + 3*a^2*b - a^3*1i -
 b^3)) + (log(tan(c + d*x) + 1i)*(B - C*1i))/(2*d*(3*a*b^2 + a^2*b*3i - a^3
 - b^3*1i)) + (a*log(a + b*tan(c + d*x))*(C*a^5 - 3*B*b^5 + B*a^2*b^3 + 3*C
*a^3*b^2 + 6*C*a*b^4))/(b^3*d*(a^2 + b^2)^3)
```

3.40 $\int \frac{\tan(c+dx)(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^3}\,dx$

Optimal result . . . 343
Rubi [A] (verified) . . . 343
Mathematica [C] (verified) . . . 346
Maple [A] (verified) . . . 346
Fricas [B] (verification not implemented) . . . 347
Sympy [F(-2)] . . . 347
Maxima [A] (verification not implemented) . . . 348
Giac [B] (verification not implemented) . . . 348
Mupad [B] (verification not implemented) . . . 349

Optimal result

Integrand size = 38, antiderivative size = 189

$$\begin{aligned}\int \frac{\tan(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^3}\,dx &= -\frac{(a^3B-3ab^2B+3a^2bC-b^3C)\,x}{(a^2+b^2)^3}\\ &\quad-\frac{(3a^2bB-b^3B-a^3C+3ab^2C)\log(a\cos(c+dx)+b\sin(c+dx))}{(a^2+b^2)^3\,d}\\ &\quad-\frac{a^2(bB-aC)}{2b^2\,(a^2+b^2)\,d(a+b\tan(c+dx))^2}+\frac{a(2b^3B-a^3C-3ab^2C)}{b^2\,(a^2+b^2)^2\,d(a+b\tan(c+dx))}\end{aligned}$$

```
[Out] -(B*a^3-3*B*a*b^2+3*C*a^2*b-C*b^3)*x/(a^2+b^2)^3-(3*B*a^2*b-B*b^3-C*a^3+3*C
*a*b^2)*ln(a*cos(d*x+c)+b*sin(d*x+c))/(a^2+b^2)^3/d-1/2*a^2*(B*b-C*a)/b^2/(
a^2+b^2)/d/(a+b*tan(d*x+c))^2+a*(2*B*b^3-C*a^3-3*C*a*b^2)/b^2/(a^2+b^2)^2/d
/(a+b*tan(d*x+c))
```

Rubi [A] (verified)

Time = 0.47 (sec) , antiderivative size = 189, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.132, Rules used

$= \{3713, 3685, 3709, 3612, 3611\}$

$$\int \frac{\tan(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx$$
$$= -\frac{a^2(bB-aC)}{2b^2d\left(a^2+b^2\right)(a+b\tan(c+dx))^2} + \frac{a(a^3(-C)-3ab^2C+2b^3B)}{b^2d\left(a^2+b^2\right)^2(a+b\tan(c+dx))}$$
$$- \frac{(a^3(-C)+3a^2bB+3ab^2C-b^3B)\log(a\cos(c+dx)+b\sin(c+dx))}{d\left(a^2+b^2\right)^3}$$
$$- \frac{x(a^3B+3a^2bC-3ab^2B-b^3C)}{\left(a^2+b^2\right)^3}$$

```
[In] Int[(Tan[c + d*x]*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c + d*x])
^3,x]
```

```
[Out] -(((a^3*B - 3*a*b^2*B + 3*a^2*b*C - b^3*C)*x)/(a^2 + b^2)^3) - ((3*a^2*b*B
- b^3*B - a^3*C + 3*a*b^2*C)*Log[a*Cos[c + d*x] + b*Sin[c + d*x]])/((a^2 +
b^2)^3*d) - (a^2*(b*B - a*C))/(2*b^2*(a^2 + b^2)*d*(a + b*Tan[c + d*x])^2)
+ (a*(2*b^3*B - a^3*C - 3*a*b^2*C))/(b^2*(a^2 + b^2)^2*d*(a + b*Tan[c + d*x
]))
```

Rule 3611

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c/(b*f))*Log[RemoveContent[a*Cos[e + f*x] + b*Si
n[e + f*x], x]], x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3612

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[(a*c + b*d)*(x/(a^2 + b^2)), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3685

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^2*((A_.) + (B_.)*tan[(e_.) + (f
_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Simp[
(-(B*c - A*d))*(b*c - a*d)^2*((c + d*Tan[e + f*x])^(n + 1)/(f*d^2*(n + 1)*(
c^2 + d^2))), x] + Dist[1/(d*(c^2 + d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)
*Simp[B*(b*c - a*d)^2 + A*d*(a^2*c - b^2*c + 2*a*b*d) + d*(B*(a^2*c - b^2*c
 + 2*a*b*d) + A*(2*a*b*c - a^2*d + b^2*d))*Tan[e + f*x] + b^2*B*(c^2 + d^2)
*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] && NeQ[b*
c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[n, -1]
```

Rule 3709

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 -
 a*b*B + a^2*C)*((a + b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1)*(a^2 + b^2))), x
] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[b*B + a*(A -
C) - (A*b - a*B - b*C)*Tan[e + f*x], x], x], x] /; FreeQ[{a, b, e, f, A, B,
 C}, x] && NeQ[A*b^2 - a*b*B + a^2*C, 0] && LtQ[m, -1] && NeQ[a^2 + b^2, 0]
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\text{integral} = \int \frac{\tan^2(c+dx)(B+C\tan(c+dx))}{(a+b\tan(c+dx))^3}\,dx$$

$$= -\frac{a^2(bB-aC)}{2b^2(a^2+b^2)\,d(a+b\tan(c+dx))^2} + \frac{\int \frac{-a(bB-aC)+b(bB-aC)\tan(c+dx)+(a^2+b^2)C\tan^2(c+dx)}{(a+b\tan(c+dx))^2}\,dx}{b(a^2+b^2)}$$

$$= -\frac{a^2(bB-aC)}{2b^2(a^2+b^2)\,d(a+b\tan(c+dx))^2} + \frac{a(2b^3B-a^3C-3ab^2C)}{b^2(a^2+b^2)^2\,d(a+b\tan(c+dx))} + \frac{\int \frac{-b(a^2B-b^2B+2abC)+b(2abB-a^2C+b^2C)\tan(c+dx)}{a+b\tan(c+dx)}\,dx}{b(a^2+b^2)^2}$$

$$= -\frac{(a^3B-3ab^2B+3a^2bC-b^3C)\,x}{(a^2+b^2)^3} - \frac{a^2(bB-aC)}{2b^2(a^2+b^2)\,d(a+b\tan(c+dx))^2} + \frac{a(2b^3B-a^3C-3ab^2C)}{b^2(a^2+b^2)^2\,d(a+b\tan(c+dx))} - \frac{(3a^2bB-b^3B-a^3C+3ab^2C)\int \frac{b-a\tan(c+dx)}{a+b\tan(c+dx)}\,dx}{(a^2+b^2)^3}$$

$$= -\frac{(a^3B-3ab^2B+3a^2bC-b^3C)\,x}{(a^2+b^2)^3} - \frac{(3a^2bB-b^3B-a^3C+3ab^2C)\log(a\cos(c+dx)+b\sin(c+dx))}{(a^2+b^2)^3\,d} - \frac{a^2(bB-aC)}{2b^2(a^2+b^2)\,d(a+b\tan(c+dx))^2} + \frac{a(2b^3B-a^3C-3ab^2C)}{b^2(a^2+b^2)^2\,d(a+b\tan(c+dx))}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 6.24 (sec) , antiderivative size = 288, normalized size of antiderivative = 1.52

$$\int \frac{\tan(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx$$

$$= \frac{-\frac{bB+aC}{b(a+b\tan(c+dx))^2}-\frac{2C\tan(c+dx)}{(a+b\tan(c+dx))^2}+C\left(\frac{i\log(i-\tan(c+dx))}{(a+ib)^2}-\frac{i\log(i+\tan(c+dx))}{(a-ib)^2}+\frac{2b\left(-2a\log(a+b\tan(c+dx))+\frac{a^2+b^2}{a+b\tan(c+dx)}\right.}{(a^2+b^2)^2}\right.}{2}$$

```
[In] Integrate[(Tan[c + d*x]*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c +
 d*x])^3,x]
```

```
[Out] (-((b*B + a*C)/(b*(a + b*Tan[c + d*x])^2)) - (2*C*Tan[c + d*x])/(a + b*Tan[
c + d*x])^2 + C*((I*Log[I - Tan[c + d*x]])/(a + I*b)^2 - (I*Log[I + Tan[c +
 d*x]])/(a - I*b)^2 + (2*b*(-2*a*Log[a + b*Tan[c + d*x]] + (a^2 + b^2)/(a +
 b*Tan[c + d*x])))/(a^2 + b^2)^2) + (b*B - a*C)*((I*Log[I - Tan[c + d*x]])/
(a + I*b)^3 - Log[I + Tan[c + d*x]]/(I*a + b)^3 + (b*((-6*a^2 + 2*b^2)*Log[
a + b*Tan[c + d*x]] + ((a^2 + b^2)*(5*a^2 + b^2 + 4*a*b*Tan[c + d*x]))/(a +
 b*Tan[c + d*x])^2))/(a^2 + b^2)^3))/(2*b*d)
```

Maple [A] (verified)

Time = 0.11 (sec) , antiderivative size = 223, normalized size of antiderivative = 1.18

method	result
derivativedivides	$\frac{\frac{\frac{\left(3B\,a^2b-B\,b^3-C\,a^3+3Ca\,b^2\right)\ln\left(1+\tan(dx+c)^2\right)}{2}+\left(-B\,a^3+3Ba\,b^2-3C\,a^2b+C\,b^3\right)\arctan(\tan(dx+c))}{\left(a^2+b^2\right)^3}-\frac{a^2(Bb-Ca)}{2b^2\left(a^2+b^2\right)(a+b\tan(dx+c)}}{d}$
default	$\frac{\frac{\frac{\left(3B\,a^2b-B\,b^3-C\,a^3+3Ca\,b^2\right)\ln\left(1+\tan(dx+c)^2\right)}{2}+\left(-B\,a^3+3Ba\,b^2-3C\,a^2b+C\,b^3\right)\arctan(\tan(dx+c))}{\left(a^2+b^2\right)^3}-\frac{a^2(Bb-Ca)}{2b^2\left(a^2+b^2\right)(a+b\tan(dx+c)}}{d}$
norman	$\frac{-\frac{\left(2Ba\,b^3-C\,a^4-3C\,a^2b^2\right)\tan(dx+c)^2}{2ad\left(a^4+2a^2b^2+b^4\right)}-\frac{a\left(B\,a^3-Ba\,b^2+2C\,a^2b\right)}{2db\left(a^4+2a^2b^2+b^4\right)}-\frac{\left(B\,a^3-3Ba\,b^2+3C\,a^2b-C\,b^3\right)a^2x}{\left(a^4+2a^2b^2+b^4\right)\left(a^2+b^2\right)}-\frac{b^2\left(B\,a^3-3Ba\,b^2+3C\,a^2b-\right.}{\left(a^4+2a^2b^2+b^4\right)}}{(a+b\tan(dx+c))^2}$
risch	$\frac{xB}{3ia^2b-ib^3-a^3+3a\,b^2}-\frac{ixC}{3ia^2b-ib^3-a^3+3a\,b^2}+\frac{6iB\,a^2bx}{a^6+3a^4b^2+3a^2b^4+b^6}-\frac{2iB\,b^3x}{a^6+3a^4b^2+3a^2b^4+b^6}-\frac{2iC\,a^3x}{a^6+3a^4b^2+3a^2b}$
parallelrisch	$\frac{-C\,a^7+4B\tan(dx+c)a^3b^4+4B\tan(dx+c)a\,b^6-2C\tan(dx+c)a^6b-8C\tan(dx+c)a^4b^3-6C\tan(dx+c)a^2b^5-B\ln\left(1+\tan\right.}{}$

```
[In] int(tan(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x,method=_R
ETURNVERBOSE)
```

```
[Out] 1/d*(1/(a^2+b^2)^3*(1/2*(3*B*a^2*b-B*b^3-C*a^3+3*C*a*b^2)*ln(1+tan(d*x+c)^2
)+(-B*a^3+3*B*a*b^2-3*C*a^2*b+C*b^3)*arctan(tan(d*x+c)))-1/2*a^2*(B*b-C*a)/
b^2/(a^2+b^2)/(a+b*tan(d*x+c))^2-(3*B*a^2*b-B*b^3-C*a^3+3*C*a*b^2)/(a^2+b^2
)^3*ln(a+b*tan(d*x+c))+a*(2*B*b^3-C*a^3-3*C*a*b^2)/(a^2+b^2)^2/b^2/(a+b*tan
(d*x+c)))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 478 vs. $2(184) = 368$.

Time = 0.27 (sec) , antiderivative size = 478, normalized size of antiderivative = 2.53

$$\int \frac{\tan(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx$$
$$= \frac{Ca^5 - 3\,Ba^4b - 5\,Ca^3b^2 + 3\,Ba^2b^3 - 2\,(Ba^5 + 3\,Ca^4b - 3\,Ba^3b^2 - Ca^2b^3)dx + (Ca^5 + Ba^4b + 7\,Ca^3b^2 -}{}$$

```
[In] integrate(tan(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x, al
gorithm="fricas")
```

```
[Out] 1/2*(C*a^5 - 3*B*a^4*b - 5*C*a^3*b^2 + 3*B*a^2*b^3 - 2*(B*a^5 + 3*C*a^4*b -
 3*B*a^3*b^2 - C*a^2*b^3)*d*x + (C*a^5 + B*a^4*b + 7*C*a^3*b^2 - 5*B*a^2*b^
3 - 2*(B*a^3*b^2 + 3*C*a^2*b^3 - 3*B*a*b^4 - C*b^5)*d*x)*tan(d*x + c)^2 + (
C*a^5 - 3*B*a^4*b - 3*C*a^3*b^2 + B*a^2*b^3 + (C*a^3*b^2 - 3*B*a^2*b^3 - 3*
C*a*b^4 + B*b^5)*tan(d*x + c)^2 + 2*(C*a^4*b - 3*B*a^3*b^2 - 3*C*a^2*b^3 +
B*a*b^4)*tan(d*x + c))*log((b^2*tan(d*x + c)^2 + 2*a*b*tan(d*x + c) + a^2)/
(tan(d*x + c)^2 + 1)) + 2*(B*a^5 + 3*C*a^4*b - 3*B*a^3*b^2 - 3*C*a^2*b^3 +
2*B*a*b^4 - 2*(B*a^4*b + 3*C*a^3*b^2 - 3*B*a^2*b^3 - C*a*b^4)*d*x)*tan(d*x
+ c))/((a^6*b^2 + 3*a^4*b^4 + 3*a^2*b^6 + b^8)*d*tan(d*x + c)^2 + 2*(a^7*b
+ 3*a^5*b^3 + 3*a^3*b^5 + a*b^7)*d*tan(d*x + c) + (a^8 + 3*a^6*b^2 + 3*a^4*
b^4 + a^2*b^6)*d)
```

Sympy [F(-2)]

Exception generated.

$$\int \frac{\tan(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx = \text{Exception raised: AttributeError}$$

```
[In] integrate(tan(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c))**3,x)
```

```
[Out] Exception raised: AttributeError >> 'NoneType' object has no attribute 'pri
mitive'
```

Maxima [A] (verification not implemented)

none

Time = 0.37 (sec) , antiderivative size = 333, normalized size of antiderivative = 1.76

$$\int \frac{\tan(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx =$$

$$-\frac{\frac{2\,(Ba^3+3\,Ca^2b-3\,Bab^2-Cb^3)(dx+c)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}-\frac{2\,(Ca^3-3\,Ba^2b-3\,Cab^2+Bb^3)\log(b\tan(dx+c)+a)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}+\frac{(Ca^3-3\,Ba^2b-3\,Cab^2+Bb^3)\log\left(\tan(dx+c\right.}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}}{2\,d}$$

```
[In] integrate(tan(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x, al
gorithm="maxima")
```

```
[Out] -1/2*(2*(B*a^3 + 3*C*a^2*b - 3*B*a*b^2 - C*b^3)*(d*x + c)/(a^6 + 3*a^4*b^2
+ 3*a^2*b^4 + b^6) - 2*(C*a^3 - 3*B*a^2*b - 3*C*a*b^2 + B*b^3)*log(b*tan(d*
x + c) + a)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) + (C*a^3 - 3*B*a^2*b - 3*C*
a*b^2 + B*b^3)*log(tan(d*x + c)^2 + 1)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6)
+ (C*a^5 + B*a^4*b + 5*C*a^3*b^2 - 3*B*a^2*b^3 + 2*(C*a^4*b + 3*C*a^2*b^3 -
 2*B*a*b^4)*tan(d*x + c))/(a^6*b^2 + 2*a^4*b^4 + a^2*b^6 + (a^4*b^4 + 2*a^2
*b^6 + b^8)*tan(d*x + c)^2 + 2*(a^5*b^3 + 2*a^3*b^5 + a*b^7)*tan(d*x + c)))
/d
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 410 vs. 2(184) = 368.

Time = 0.68 (sec) , antiderivative size = 410, normalized size of antiderivative = 2.17

$$\int \frac{\tan(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx =$$

$$-\frac{\frac{2\,(Ba^3+3\,Ca^2b-3\,Bab^2-Cb^3)(dx+c)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}+\frac{(Ca^3-3\,Ba^2b-3\,Cab^2+Bb^3)\log\left(\tan(dx+c)^2+1\right)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}-\frac{2\,(Ca^3b-3\,Ba^2b^2-3\,Cab^3+Bb^4)\log(|b\tan(}{a^6b+3\,a^4b^3+3\,a^2b^5+b^7}}{}$$

```
[In] integrate(tan(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x, al
gorithm="giac")
```

```
[Out] -1/2*(2*(B*a^3 + 3*C*a^2*b - 3*B*a*b^2 - C*b^3)*(d*x + c)/(a^6 + 3*a^4*b^2
+ 3*a^2*b^4 + b^6) + (C*a^3 - 3*B*a^2*b - 3*C*a*b^2 + B*b^3)*log(tan(d*x +
c)^2 + 1)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) - 2*(C*a^3*b - 3*B*a^2*b^2 -
3*C*a*b^3 + B*b^4)*log(abs(b*tan(d*x + c) + a))/(a^6*b + 3*a^4*b^3 + 3*a^2*
b^5 + b^7) + (3*C*a^3*b^4*tan(d*x + c)^2 - 9*B*a^2*b^5*tan(d*x + c)^2 - 9*C
*a*b^6*tan(d*x + c)^2 + 3*B*b^7*tan(d*x + c)^2 + 2*C*a^6*b*tan(d*x + c) + 1
4*C*a^4*b^3*tan(d*x + c) - 22*B*a^3*b^4*tan(d*x + c) - 12*C*a^2*b^5*tan(d*x
 + c) + 2*B*a*b^6*tan(d*x + c) + C*a^7 + B*a^6*b + 9*C*a^5*b^2 - 11*B*a^4*b
^3 - 4*C*a^3*b^4)/((a^6*b^2 + 3*a^4*b^4 + 3*a^2*b^6 + b^8)*(b*tan(d*x + c)
+ a)^2))/d
```

Mupad [B] (verification not implemented)

Time = 8.77 (sec) , antiderivative size = 280, normalized size of antiderivative = 1.48

$$\int \frac{\tan(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^3}\,dx$$
$$= \frac{\ln\left(a+b\tan(c+d\,x)\right)\,\left(C\,a^3-3\,B\,a^2\,b-3\,C\,a\,b^2+B\,b^3\right)}{d\,{\left(a^2+b^2\right)}^3}$$
$$-\frac{\ln\left(\tan(c+d\,x)-\mathrm{i}\right)\,\left(-C+B\,1\mathrm{i}\right)}{2\,d\,\left(-a^3-a^2\,b\,3\mathrm{i}+3\,a\,b^2+b^3\,1\mathrm{i}\right)}-\frac{\ln\left(\tan(c+d\,x)+1\mathrm{i}\right)\,\left(B-C\,1\mathrm{i}\right)}{2\,d\,\left(-a^3\,1\mathrm{i}-3\,a^2\,b+a\,b^2\,3\mathrm{i}+b^3\right)}$$
$$-\frac{\frac{a\,\left(C\,a^4+B\,a^3\,b+5\,C\,a^2\,b^2-3\,B\,a\,b^3\right)}{2\,b^2\,\left(a^4+2\,a^2\,b^2+b^4\right)}+\frac{\tan(c+d\,x)\,\left(C\,a^4+3\,C\,a^2\,b^2-2\,B\,a\,b^3\right)}{b\,\left(a^4+2\,a^2\,b^2+b^4\right)}}{d\,\left(a^2+2\,a\,b\tan\left(c+d\,x\right)+b^2\tan{\left(c+d\,x\right)}^2\right)}$$

```
[In] int((tan(c + d*x)*(B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c + d*x))
^3,x)
```

```
[Out] (log(a + b*tan(c + d*x))*(B*b^3 + C*a^3 - 3*B*a^2*b - 3*C*a*b^2))/(d*(a^2 +
 b^2)^3) - (log(tan(c + d*x) - 1i)*(B*1i - C))/(2*d*(3*a*b^2 - a^2*b*3i - a
^3 + b^3*1i)) - (log(tan(c + d*x) + 1i)*(B - C*1i))/(2*d*(a*b^2*3i - 3*a^2*
b - a^3*1i + b^3)) - ((a*(C*a^4 + 5*C*a^2*b^2 - 3*B*a*b^3 + B*a^3*b))/(2*b^
2*(a^4 + b^4 + 2*a^2*b^2)) + (tan(c + d*x)*(C*a^4 + 3*C*a^2*b^2 - 2*B*a*b^3
))/(b*(a^4 + b^4 + 2*a^2*b^2)))/(d*(a^2 + b^2*tan(c + d*x)^2 + 2*a*b*tan(c
+ d*x)))
```

3.41 $\int \frac{B\tan(c+dx)+C\tan^2(c+dx)}{(a+b\tan(c+dx))^3}\,dx$

Optimal result 350
Rubi [A] (verified) 350
Mathematica [C] (verified) 352
Maple [A] (verified) 353
Fricas [B] (verification not implemented) 353
Sympy [F(-2)] 354
Maxima [A] (verification not implemented) 354
Giac [B] (verification not implemented) 355
Mupad [B] (verification not implemented) 355

Optimal result

Integrand size = 32, antiderivative size = 179

$$\int \frac{B\tan(c+dx)+C\tan^2(c+dx)}{(a+b\tan(c+dx))^3}\,dx = \frac{(3a^2bB-b^3B-a^3C+3ab^2C)\,x}{(a^2+b^2)^3} - \frac{(a^3B-3ab^2B+3a^2bC-b^3C)\log(a\cos(c+dx)+b\sin(c+dx))}{(a^2+b^2)^3\,d} + \frac{a(bB-aC)}{2b\,(a^2+b^2)\,d(a+b\tan(c+dx))^2} + \frac{a^2B-b^2B+2abC}{(a^2+b^2)^2\,d(a+b\tan(c+dx))}$$

```
[Out] (3*B*a^2*b-B*b^3-C*a^3+3*C*a*b^2)*x/(a^2+b^2)^3-(B*a^3-3*B*a*b^2+3*C*a^2*b-
C*b^3)*ln(a*cos(d*x+c)+b*sin(d*x+c))/(a^2+b^2)^3/d+1/2*a*(B*b-C*a)/b/(a^2+b
^2)/d/(a+b*tan(d*x+c))^2+(B*a^2-B*b^2+2*C*a*b)/(a^2+b^2)^2/d/(a+b*tan(d*x+c
))
```

Rubi [A] (verified)

Time = 0.30 (sec) , antiderivative size = 179, normalized size of antiderivative = 1.00, number of steps used = 4, number of rules used = 4, $\frac{\text{number of rules}}{\text{integrand size}} = 0.125$, Rules used

= {3709, 3610, 3612, 3611}

$$\int \frac{B\tan(c+dx)+C\tan^2(c+dx)}{(a+b\tan(c+dx))^3}\,dx = \frac{a(bB-aC)}{2bd\,(a^2+b^2)\,(a+b\tan(c+dx))^2}+\frac{a^2B+2abC-b^2B}{d\,(a^2+b^2)^2\,(a+b\tan(c+dx))} - \frac{(a^3B+3a^2bC-3ab^2B-b^3C)\log(a\cos(c+dx)+b\sin(c+dx))}{d\,(a^2+b^2)^3} + \frac{x(a^3(-C)+3a^2bB+3ab^2C-b^3B)}{(a^2+b^2)^3}$$

```
[In] Int[(B*Tan[c + d*x] + C*Tan[c + d*x]^2)/(a + b*Tan[c + d*x])^3,x]
```

```
[Out] ((3*a^2*b*B - b^3*B - a^3*C + 3*a*b^2*C)*x)/(a^2 + b^2)^3 - ((a^3*B - 3*a*b
^2*B + 3*a^2*b*C - b^3*C)*Log[a*Cos[c + d*x] + b*Sin[c + d*x]])/((a^2 + b^2
)^3*d) + (a*(b*B - a*C))/(2*b*(a^2 + b^2)*d*(a + b*Tan[c + d*x])^2) + (a^2*
B - b^2*B + 2*a*b*C)/((a^2 + b^2)^2*d*(a + b*Tan[c + d*x]))
```

Rule 3610

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(b*c - a*d)*((a + b*Tan[e + f*x])^(m + 1)/
(f*(m + 1)*(a^2 + b^2))), x] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])
^(m + 1)*Simp[a*c + b*d - (b*c - a*d)*Tan[e + f*x], x], x], x] /; FreeQ[{a,
 b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && LtQ[m, -1]
```

Rule 3611

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c/(b*f))*Log[RemoveContent[a*Cos[e + f*x] + b*Si
n[e + f*x], x]], x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3612

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[(a*c + b*d)*(x/(a^2 + b^2)), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3709

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 -
 a*b*B + a^2*C)*((a + b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1)*(a^2 + b^2))), x
```

```
] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[b*B + a*(A -
C) - (A*b - a*B - b*C)*Tan[e + f*x], x], x], x] /; FreeQ[{a, b, e, f, A, B,
 C}, x] && NeQ[A*b^2 - a*b*B + a^2*C, 0] && LtQ[m, -1] && NeQ[a^2 + b^2, 0]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \frac{a(bB-aC)}{2b\left(a^2+b^2\right)d(a+b\tan(c+dx))^2}+\frac{\int\frac{bB-aC+(aB+bC)\tan(c+dx)}{(a+b\tan(c+dx))^2}\,dx}{a^2+b^2}\\
&= \frac{a(bB-aC)}{2b\left(a^2+b^2\right)d(a+b\tan(c+dx))^2}+\frac{a^2B-b^2B+2abC}{\left(a^2+b^2\right)^2d(a+b\tan(c+dx))}\\
&\quad+\frac{\int\frac{2abB-a^2C+b^2C+\left(a^2B-b^2B+2abC\right)\tan(c+dx)}{a+b\tan(c+dx)}\,dx}{\left(a^2+b^2\right)^2}\\
&= \frac{\left(3a^2bB-b^3B-a^3C+3ab^2C\right)x}{\left(a^2+b^2\right)^3}+\frac{a(bB-aC)}{2b\left(a^2+b^2\right)d(a+b\tan(c+dx))^2}\\
&\quad+\frac{a^2B-b^2B+2abC}{\left(a^2+b^2\right)^2d(a+b\tan(c+dx))}\\
&\quad-\frac{\left(a^3B-3ab^2B+3a^2bC-b^3C\right)\int\frac{b-a\tan(c+dx)}{a+b\tan(c+dx)}\,dx}{\left(a^2+b^2\right)^3}\\
&= \frac{\left(3a^2bB-b^3B-a^3C+3ab^2C\right)x}{\left(a^2+b^2\right)^3}\\
&\quad-\frac{\left(a^3B-3ab^2B+3a^2bC-b^3C\right)\log(a\cos(c+dx)+b\sin(c+dx))}{\left(a^2+b^2\right)^3d}\\
&\quad+\frac{a(bB-aC)}{2b\left(a^2+b^2\right)d(a+b\tan(c+dx))^2}+\frac{a^2B-b^2B+2abC}{\left(a^2+b^2\right)^2d(a+b\tan(c+dx))}
\end{aligned}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 4.35 (sec) , antiderivative size = 188, normalized size of antiderivative = 1.05

$$\int\frac{B\tan(c+dx)+C\tan^2(c+dx)}{(a+b\tan(c+dx))^3}\,dx$$

$$=\frac{\frac{(B+iC)\log(i-\tan(c+dx))}{(a+ib)^3}+\frac{(B-iC)\log(i+\tan(c+dx))}{(a-ib)^3}-\frac{2\left(a^3B-3ab^2B+3a^2bC-b^3C\right)\log(a+b\tan(c+dx))}{\left(a^2+b^2\right)^3}+\frac{a(bB-aC)}{b(a^2+b^2)(a+b\tan(c+dx))^2}}{2d}$$

```
[In] Integrate[(B*Tan[c + d*x] + C*Tan[c + d*x]^2)/(a + b*Tan[c + d*x])^3,x]
[Out] (((B + I*C)*Log[I - Tan[c + d*x]])/(a + I*b)^3 + ((B - I*C)*Log[I + Tan[c +
 d*x]])/(a - I*b)^3 - (2*(a^3*B - 3*a*b^2*B + 3*a^2*b*C - b^3*C)*Log[a + b*
```

```
Tan[c + d*x]])/(a^2 + b^2)^3 + (a*(b*B - a*C))/(b*(a^2 + b^2)*(a + b*Tan[c
+ d*x])^2) + (2*(a^2*B - b^2*B + 2*a*b*C))/((a^2 + b^2)^2*(a + b*Tan[c + d*
x])))/(2*d)
```

Maple [A] (verified)

Time = 0.14 (sec) , antiderivative size = 213, normalized size of antiderivative = 1.19

method	result
derivativedivides	$\frac{\frac{\frac{(B a^3-3Ba b^2+3C a^2b-C b^3)\ln(1+\tan(dx+c)^2)}{2}+(3B a^2b-B b^3-C a^3+3Ca b^2)\arctan(\tan(dx+c))}{(a^2+b^2)^3}+\frac{a(Bb-Ca)}{2(a^2+b^2)b(a+b\tan(dx+c))}}{d}$
default	$\frac{\frac{\frac{(B a^3-3Ba b^2+3C a^2b-C b^3)\ln(1+\tan(dx+c)^2)}{2}+(3B a^2b-B b^3-C a^3+3Ca b^2)\arctan(\tan(dx+c))}{(a^2+b^2)^3}+\frac{a(Bb-Ca)}{2(a^2+b^2)b(a+b\tan(dx+c))}}{d}$
norman	$\frac{\frac{(B a^2b^2-B b^4+2Ca b^3)\tan(dx+c)}{db(a^4+2a^2b^2+b^4)}+\frac{(3B a^2b-B b^3-C a^3+3Ca b^2)a^2x}{(a^4+2a^2b^2+b^4)(a^2+b^2)}+\frac{b^2(3B a^2b-B b^3-C a^3+3Ca b^2)x\tan(dx+c)^2}{(a^4+2a^2b^2+b^4)(a^2+b^2)}+\frac{a(3B a}{2}}{(a+b\tan(dx+c))^2}$
risch	$\frac{ixB}{3ia^2b-ib^3-a^3+3a b^2}+\frac{xC}{3ia^2b-ib^3-a^3+3a b^2}+\frac{2ia^3Bx}{a^6+3a^4b^2+3a^2b^4+b^6}-\frac{6ia b^2Bx}{a^6+3a^4b^2+3a^2b^4+b^6}+\frac{6ia^2bCx}{a^6+3a^4b^2+3a}$
parallelrisch	$\frac{2B a^3b^4-b^6Ba+3C a^2b^5+3B a^5b^2+4C\tan(dx+c)a^3b^4+4C\tan(dx+c)a b^6-C\ln(1+\tan(dx+c)^2)\tan(dx+c)^2b^7+2C\ln}{}$

```
[In] int((B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x,method=_RETURNVERBOS
E)
```

```
[Out] 1/d*(1/(a^2+b^2)^3*(1/2*(B*a^3-3*B*a*b^2+3*C*a^2*b-C*b^3)*ln(1+tan(d*x+c)^2
)+(3*B*a^2*b-B*b^3-C*a^3+3*C*a*b^2)*arctan(tan(d*x+c)))+1/2*a*(B*b-C*a)/(a^
2+b^2)/b/(a+b*tan(d*x+c))^2+(B*a^2-B*b^2+2*C*a*b)/(a^2+b^2)^2/(a+b*tan(d*x+
c))-(B*a^3-3*B*a*b^2+3*C*a^2*b-C*b^3)/(a^2+b^2)^3*ln(a+b*tan(d*x+c)))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 488 vs. $2(176) = 352$.

Time = 0.28 (sec) , antiderivative size = 488, normalized size of antiderivative = 2.73

$$\int \frac{B\tan(c+dx)+C\tan^2(c+dx)}{(a+b\tan(c+dx))^3}\,dx = -\frac{3\,Ca^4b-5\,Ba^3b^2-3\,Ca^2b^3+Bab^4+2\,(Ca^5-3\,Ba^4b-3\,Ca^3b^2+Ba^2b^3)dx-(Ca^4b-3\,Ba^3b^2-5\,C}{}$$

```
[In] integrate((B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x, algorithm="fr
icas")
```

```
[Out] -1/2*(3*C*a^4*b - 5*B*a^3*b^2 - 3*C*a^2*b^3 + B*a*b^4 + 2*(C*a^5 - 3*B*a^4*
b - 3*C*a^3*b^2 + B*a^2*b^3)*d*x - (C*a^4*b - 3*B*a^3*b^2 - 5*C*a^2*b^3 + 3
*B*a*b^4 - 2*(C*a^3*b^2 - 3*B*a^2*b^3 - 3*C*a*b^4 + B*b^5)*d*x)*tan(d*x + c
)^2 + (B*a^5 + 3*C*a^4*b - 3*B*a^3*b^2 - C*a^2*b^3 + (B*a^3*b^2 + 3*C*a^2*b
^3 - 3*B*a*b^4 - C*b^5)*tan(d*x + c)^2 + 2*(B*a^4*b + 3*C*a^3*b^2 - 3*B*a^2
*b^3 - C*a*b^4)*tan(d*x + c))*log((b^2*tan(d*x + c)^2 + 2*a*b*tan(d*x + c)
+ a^2)/(tan(d*x + c)^2 + 1)) - 2*(C*a^5 - 2*B*a^4*b - 3*C*a^3*b^2 + 3*B*a^2
*b^3 + 2*C*a*b^4 - B*b^5 - 2*(C*a^4*b - 3*B*a^3*b^2 - 3*C*a^2*b^3 + B*a*b^4
)*d*x)*tan(d*x + c))/((a^6*b^2 + 3*a^4*b^4 + 3*a^2*b^6 + b^8)*d*tan(d*x + c
)^2 + 2*(a^7*b + 3*a^5*b^3 + 3*a^3*b^5 + a*b^7)*d*tan(d*x + c) + (a^8 + 3*a
^6*b^2 + 3*a^4*b^4 + a^2*b^6)*d)
```

Sympy [F(-2)]

Exception generated.

$$\int \frac{B\tan(c+dx)+C\tan^2(c+dx)}{(a+b\tan(c+dx))^3}\,dx = \text{Exception raised: AttributeError}$$

```
[In] integrate((B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c))**3,x)
```

```
[Out] Exception raised: AttributeError >> 'NoneType' object has no attribute 'pri
mitive'
```

Maxima [A] (verification not implemented)

none

Time = 0.39 (sec) , antiderivative size = 330, normalized size of antiderivative = 1.84

$$\int \frac{B\tan(c+dx)+C\tan^2(c+dx)}{(a+b\tan(c+dx))^3}\,dx = -\frac{\frac{2\,(Ca^3-3\,Ba^2b-3\,Cab^2+Bb^3)(dx+c)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}+\frac{2\,(Ba^3+3\,Ca^2b-3\,Bab^2-Cb^3)\log(b\tan(dx+c)+a)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}-\frac{(Ba^3+3\,Ca^2b-3\,Bab^2-Cb^3)\log\left(\tan(dx+c\right.}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}}{2\,d}$$

```
[In] integrate((B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x, algorithm="ma
xima")
```

```
[Out] -1/2*(2*(C*a^3 - 3*B*a^2*b - 3*C*a*b^2 + B*b^3)*(d*x + c)/(a^6 + 3*a^4*b^2
+ 3*a^2*b^4 + b^6) + 2*(B*a^3 + 3*C*a^2*b - 3*B*a*b^2 - C*b^3)*log(b*tan(d*
x + c) + a)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) - (B*a^3 + 3*C*a^2*b - 3*B*
a*b^2 - C*b^3)*log(tan(d*x + c)^2 + 1)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6)
+ (C*a^4 - 3*B*a^3*b - 3*C*a^2*b^2 + B*a*b^3 - 2*(B*a^2*b^2 + 2*C*a*b^3 - B
*b^4)*tan(d*x + c))/(a^6*b + 2*a^4*b^3 + a^2*b^5 + (a^4*b^3 + 2*a^2*b^5 + b
^7)*tan(d*x + c)^2 + 2*(a^5*b^2 + 2*a^3*b^4 + a*b^6)*tan(d*x + c)))/d
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 410 vs. $2(176) = 352$.

Time = 0.69 (sec) , antiderivative size = 410, normalized size of antiderivative = 2.29

$$\int \frac{B\tan(c+dx)+C\tan^2(c+dx)}{(a+b\tan(c+dx))^3}\,dx = -\frac{\frac{2\,(Ca^3-3\,Ba^2b-3\,Cab^2+Bb^3)(dx+c)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}-\frac{(Ba^3+3\,Ca^2b-3\,Bab^2-Cb^3)\log\left(\tan(dx+c)^2+1\right)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}+\frac{2\,(Ba^3b+3\,Ca^2b^2-3\,Bab^3-Cb^4)\log(|b\tan}{a^6b+3\,a^4b^3+3\,a^2b^5+b^7}}{}$$

```
[In] integrate((B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x, algorithm="gi
ac")
```

```
[Out] -1/2*(2*(C*a^3 - 3*B*a^2*b - 3*C*a*b^2 + B*b^3)*(d*x + c)/(a^6 + 3*a^4*b^2
+ 3*a^2*b^4 + b^6) - (B*a^3 + 3*C*a^2*b - 3*B*a*b^2 - C*b^3)*log(tan(d*x +
c)^2 + 1)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) + 2*(B*a^3*b + 3*C*a^2*b^2 -
3*B*a*b^3 - C*b^4)*log(abs(b*tan(d*x + c) + a))/(a^6*b + 3*a^4*b^3 + 3*a^2*
b^5 + b^7) - (3*B*a^3*b^3*tan(d*x + c)^2 + 9*C*a^2*b^4*tan(d*x + c)^2 - 9*B
*a*b^5*tan(d*x + c)^2 - 3*C*b^6*tan(d*x + c)^2 + 8*B*a^4*b^2*tan(d*x + c) +
 22*C*a^3*b^3*tan(d*x + c) - 18*B*a^2*b^4*tan(d*x + c) - 2*C*a*b^5*tan(d*x
+ c) - 2*B*b^6*tan(d*x + c) - C*a^6 + 6*B*a^5*b + 11*C*a^4*b^2 - 7*B*a^3*b^
3 - B*a*b^5)/((a^6*b + 3*a^4*b^3 + 3*a^2*b^5 + b^7)*(b*tan(d*x + c) + a)^2)
)/d
```

Mupad [B] (verification not implemented)

Time = 8.65 (sec) , antiderivative size = 282, normalized size of antiderivative = 1.58

$$\int \frac{B\tan(c+dx)+C\tan^2(c+dx)}{(a+b\tan(c+dx))^3}\,dx = \frac{\frac{\tan(c+d\,x)\,\left(B\,a^2\,b+2\,C\,a\,b^2-B\,b^3\right)}{a^4+2\,a^2\,b^2+b^4}-\frac{C\,a^4-3\,B\,a^3\,b-3\,C\,a^2\,b^2+B\,a\,b^3}{2\,b\,\left(a^4+2\,a^2\,b^2+b^4\right)}}{d\,\left(a^2+2\,a\,b\tan\left(c+d\,x\right)+b^2\tan\left(c+d\,x\right)^2\right)} - \frac{\ln\left(a+b\tan(c+d\,x)\right)\,\left(\frac{B\,a+3\,C\,b}{\left(a^2+b^2\right)^2}-\frac{4\,b^2\,\left(B\,a+C\,b\right)}{\left(a^2+b^2\right)^3}\right)}{d} - \frac{\ln\left(\tan(c+d\,x)-\mathrm{i}\right)\,\left(-C+B\,1\mathrm{i}\right)}{2\,d\,\left(-a^3\,1\mathrm{i}+3\,a^2\,b+a\,b^2\,3\mathrm{i}-b^3\right)} - \frac{\ln\left(\tan(c+d\,x)+1\mathrm{i}\right)\,\left(B-C\,1\mathrm{i}\right)}{2\,d\,\left(-a^3+a^2\,b\,3\mathrm{i}+3\,a\,b^2-b^3\,1\mathrm{i}\right)}$$

```
[In] int((B*tan(c + d*x) + C*tan(c + d*x)^2)/(a + b*tan(c + d*x))^3,x)
```

```
[Out] ((tan(c + d*x)*(B*a^2*b - B*b^3 + 2*C*a*b^2))/(a^4 + b^4 + 2*a^2*b^2) - (C*
a^4 - 3*C*a^2*b^2 + B*a*b^3 - 3*B*a^3*b)/(2*b*(a^4 + b^4 + 2*a^2*b^2)))/(d*
(a^2 + b^2*tan(c + d*x)^2 + 2*a*b*tan(c + d*x))) - (log(a + b*tan(c + d*x))
```

```
*((B*a + 3*C*b)/(a^2 + b^2)^2 - (4*b^2*(B*a + C*b))/(a^2 + b^2)^3))/d - (lo
g(tan(c + d*x) - 1i)*(B*1i - C))/(2*d*(a*b^2*3i + 3*a^2*b - a^3*1i - b^3))
- (log(tan(c + d*x) + 1i)*(B - C*1i))/(2*d*(3*a*b^2 + a^2*b*3i - a^3 - b^3*
1i))
```

3.42 $\int \frac{\cot(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx$

Optimal result . . . 357
Rubi [A] (verified) . . . 357
Mathematica [C] (verified) . . . 359
Maple [A] (verified) . . . 360
Fricas [B] (verification not implemented) . . . 361
Sympy [F(-2)] . . . 361
Maxima [A] (verification not implemented) . . . 362
Giac [B] (verification not implemented) . . . 362
Mupad [B] (verification not implemented) . . . 363

Optimal result

Integrand size = 38, antiderivative size = 175

$$
\begin{aligned}
&\int \frac{\cot(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx \\
&= \frac{(a^3B-3ab^2B+3a^2bC-b^3C)\,x}{(a^2+b^2)^3} \\
&\quad + \frac{(3a^2bB-b^3B-a^3C+3ab^2C)\log(a\cos(c+dx)+b\sin(c+dx))}{(a^2+b^2)^3\,d} \\
&\quad - \frac{bB-aC}{2\,(a^2+b^2)\,d(a+b\tan(c+dx))^2} - \frac{2abB-a^2C+b^2C}{(a^2+b^2)^2\,d(a+b\tan(c+dx))}
\end{aligned}
$$

```
[Out] (B*a^3-3*B*a*b^2+3*C*a^2*b-C*b^3)*x/(a^2+b^2)^3+(3*B*a^2*b-B*b^3-C*a^3+3*C*
a*b^2)*ln(a*cos(d*x+c)+b*sin(d*x+c))/(a^2+b^2)^3/d+1/2*(-B*b+C*a)/(a^2+b^2)
/d/(a+b*tan(d*x+c))^2+(-2*B*a*b+C*a^2-C*b^2)/(a^2+b^2)^2/d/(a+b*tan(d*x+c))
```

Rubi [A] (verified)

Time = 0.38 (sec) , antiderivative size = 175, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 4, $\frac{\text{number of rules}}{\text{integrand size}} = 0.105$, Rules used

= {3713, 3610, 3612, 3611}

$$\int \frac{\cot(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx$$
$$= -\frac{bB-aC}{2d\left(a^2+b^2\right)(a+b\tan(c+dx))^2} - \frac{a^2(-C)+2abB+b^2C}{d\left(a^2+b^2\right)^2(a+b\tan(c+dx))}$$
$$+ \frac{\left(a^3(-C)+3a^2bB+3ab^2C-b^3B\right)\log(a\cos(c+dx)+b\sin(c+dx))}{d\left(a^2+b^2\right)^3}$$
$$+ \frac{x(a^3B+3a^2bC-3ab^2B-b^3C)}{\left(a^2+b^2\right)^3}$$

```
[In] Int[(Cot[c + d*x]*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c + d*x])
^3,x]
```

```
[Out] ((a^3*B - 3*a*b^2*B + 3*a^2*b*C - b^3*C)*x)/(a^2 + b^2)^3 + ((3*a^2*b*B - b
^3*B - a^3*C + 3*a*b^2*C)*Log[a*Cos[c + d*x] + b*Sin[c + d*x]])/((a^2 + b^2
)^3*d) - (b*B - a*C)/(2*(a^2 + b^2)*d*(a + b*Tan[c + d*x])^2) - (2*a*b*B -
a^2*C + b^2*C)/((a^2 + b^2)^2*d*(a + b*Tan[c + d*x]))
```

Rule 3610

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(b*c - a*d)*((a + b*Tan[e + f*x])^(m + 1)/
(f*(m + 1)*(a^2 + b^2))), x] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])
^(m + 1)*Simp[a*c + b*d - (b*c - a*d)*Tan[e + f*x], x], x], x] /; FreeQ[{a,
 b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && LtQ[m, -1]
```

Rule 3611

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c/(b*f))*Log[RemoveContent[a*Cos[e + f*x] + b*Si
n[e + f*x], x]], x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3612

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[(a*c + b*d)*(x/(a^2 + b^2)), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
```

```
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \int \frac{B + C\tan(c+dx)}{(a+b\tan(c+dx))^3}\,dx \\
&= -\frac{bB-aC}{2(a^2+b^2)\,d(a+b\tan(c+dx))^2} + \frac{\int \frac{aB+bC-(bB-aC)\tan(c+dx)}{(a+b\tan(c+dx))^2}\,dx}{a^2+b^2} \\
&= -\frac{bB-aC}{2(a^2+b^2)\,d(a+b\tan(c+dx))^2} - \frac{2abB-a^2C+b^2C}{(a^2+b^2)^2\,d(a+b\tan(c+dx))} \\
&\quad + \frac{\int \frac{a^2B-b^2B+2abC-(2abB-a^2C+b^2C)\tan(c+dx)}{a+b\tan(c+dx)}\,dx}{(a^2+b^2)^2} \\
&= \frac{(a^3B-3ab^2B+3a^2bC-b^3C)\,x}{(a^2+b^2)^3} - \frac{bB-aC}{2(a^2+b^2)\,d(a+b\tan(c+dx))^2} \\
&\quad - \frac{2abB-a^2C+b^2C}{(a^2+b^2)^2\,d(a+b\tan(c+dx))} \\
&\quad + \frac{(3a^2bB-b^3B-a^3C+3ab^2C)\int \frac{b-a\tan(c+dx)}{a+b\tan(c+dx)}\,dx}{(a^2+b^2)^3} \\
&= \frac{(a^3B-3ab^2B+3a^2bC-b^3C)\,x}{(a^2+b^2)^3} \\
&\quad + \frac{(3a^2bB-b^3B-a^3C+3ab^2C)\log(a\cos(c+dx)+b\sin(c+dx))}{(a^2+b^2)^3\,d} \\
&\quad - \frac{bB-aC}{2(a^2+b^2)\,d(a+b\tan(c+dx))^2} - \frac{2abB-a^2C+b^2C}{(a^2+b^2)^2\,d(a+b\tan(c+dx))}
\end{aligned}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 4.74 (sec) , antiderivative size = 243, normalized size of antiderivative = 1.39

$$\int \frac{\cot(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^3}\,dx =$$

$$-\frac{C\left(\frac{i\log(i-\tan(c+dx))}{(a+ib)^2} - \frac{i\log(i+\tan(c+dx))}{(a-ib)^2} + \frac{2b\left(-2a\log(a+b\tan(c+dx))+\frac{a^2+b^2}{a+b\tan(c+dx)}\right)}{(a^2+b^2)^2}\right) + (bB-aC)\left(\frac{i\log(i-\tan(c+dx}{(a+ib)^3}\right.}{2bd}$$

```
[In] Integrate[(Cot[c + d*x]*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c +
 d*x])^3,x]
```

```
[Out] -1/2*(C*((I*Log[I - Tan[c + d*x]])/(a + I*b)^2 - (I*Log[I + Tan[c + d*x]])/
(a - I*b)^2 + (2*b*(-2*a*Log[a + b*Tan[c + d*x]] + (a^2 + b^2)/(a + b*Tan[c
 + d*x])))/(a^2 + b^2)^2) + (b*B - a*C)*((I*Log[I - Tan[c + d*x]])/(a + I*b
)^3 - Log[I + Tan[c + d*x]]/(I*a + b)^3 + (b*((-6*a^2 + 2*b^2)*Log[a + b*Ta
n[c + d*x]] + ((a^2 + b^2)*(5*a^2 + b^2 + 4*a*b*Tan[c + d*x]))/(a + b*Tan[c
 + d*x])^2))/(a^2 + b^2)^3))/(b*d)
```

Maple [A] (verified)

Time = 0.50 (sec) , antiderivative size = 208, normalized size of antiderivative = 1.19

method	result
derivativedivides	$\frac{\frac{\frac{\left(-3B\,a^2b+B\,b^3+C\,a^3-3Ca\,b^2\right)\ln\left(1+\tan(dx+c)^2\right)}{2}+\left(B\,a^3-3Ba\,b^2+3C\,a^2b-C\,b^3\right)\arctan(\tan(dx+c))}{\left(a^2+b^2\right)^3}+\frac{\left(3B\,a^2b-B\,b^3-C\,a^3+3Ca\,b\right.}{\left(a^2+b^2\right.}}{d}$
default	$\frac{\frac{\frac{\left(-3B\,a^2b+B\,b^3+C\,a^3-3Ca\,b^2\right)\ln\left(1+\tan(dx+c)^2\right)}{2}+\left(B\,a^3-3Ba\,b^2+3C\,a^2b-C\,b^3\right)\arctan(\tan(dx+c))}{\left(a^2+b^2\right)^3}+\frac{\left(3B\,a^2b-B\,b^3-C\,a^3+3Ca\,b\right.}{\left(a^2+b^2\right.}}{d}$
parallelrisch	$6a\left(B\,a^2b-\frac{1}{3}B\,b^3-\frac{1}{3}C\,a^3+Ca\,b^2\right)(a+b\tan(dx+c))^2\ln(a+b\tan(dx+c))-3a\left(B\,a^2b-\frac{1}{3}B\,b^3-\frac{1}{3}C\,a^3+Ca\,b^2\right)(a+b\tan(dx+$
norman	$\frac{\frac{\left(B\,a^3-3Ba\,b^2+3C\,a^2b-C\,b^3\right)a^2x}{\left(a^4+2a^2b^2+b^4\right)\left(a^2+b^2\right)}+\frac{b^2\left(B\,a^3-3Ba\,b^2+3C\,a^2b-C\,b^3\right)x\tan(dx+c)^2}{\left(a^4+2a^2b^2+b^4\right)\left(a^2+b^2\right)}-\frac{3B\,a^2b^2+B\,b^4-2C\,a^3b}{2bd\left(a^4+2a^2b^2+b^4\right)}+\frac{b\left(2Ba\,b^2-C\,a^2b+C\,b^3\right.}{2da\left(a^4+2a^2b^2\right.}}{(a+b\tan(dx+c))^2}$
risch	$-\frac{xB}{3ia^2b-ib^3-a^3+3a\,b^2}+\frac{ixC}{3ia^2b-ib^3-a^3+3a\,b^2}-\frac{6iB\,a^2bx}{a^6+3a^4b^2+3a^2b^4+b^6}+\frac{2iB\,b^3x}{a^6+3a^4b^2+3a^2b^4+b^6}+\frac{2iC\,a^3x}{a^6+3a^4b^2+3a}$

```
[In] int(cot(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x,method=_R
ETURNVERBOSE)
```

```
[Out] 1/d*(1/(a^2+b^2)^3*(1/2*(-3*B*a^2*b+B*b^3+C*a^3-3*C*a*b^2)*ln(1+tan(d*x+c)^
2)+(B*a^3-3*B*a*b^2+3*C*a^2*b-C*b^3)*arctan(tan(d*x+c)))+(3*B*a^2*b-B*b^3-C
*a^3+3*C*a*b^2)/(a^2+b^2)^3*ln(a+b*tan(d*x+c))-1/2*(B*b-C*a)/(a^2+b^2)/(a+b
*tan(d*x+c))^2-(2*B*a*b-C*a^2+C*b^2)/(a^2+b^2)^2/(a+b*tan(d*x+c)))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 482 vs. $2(171) = 342$.

Time = 0.28 (sec) , antiderivative size = 482, normalized size of antiderivative = 2.75

$$\int \frac{\cot(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx$$
$$= \frac{5\,Ca^3b^2 - 7\,Ba^2b^3 - Cab^4 - Bb^5 + 2\,(Ba^5 + 3\,Ca^4b - 3\,Ba^3b^2 - Ca^2b^3)dx - (3\,Ca^3b^2 - 5\,Ba^2b^3 - 3\,Ca}{}$$

```
[In] integrate(cot(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x, al
gorithm="fricas")
```

```
[Out] 1/2*(5*C*a^3*b^2 - 7*B*a^2*b^3 - C*a*b^4 - B*b^5 + 2*(B*a^5 + 3*C*a^4*b - 3
*B*a^3*b^2 - C*a^2*b^3)*d*x - (3*C*a^3*b^2 - 5*B*a^2*b^3 - 3*C*a*b^4 + B*b^
5 - 2*(B*a^3*b^2 + 3*C*a^2*b^3 - 3*B*a*b^4 - C*b^5)*d*x)*tan(d*x + c)^2 - (
C*a^5 - 3*B*a^4*b - 3*C*a^3*b^2 + B*a^2*b^3 + (C*a^3*b^2 - 3*B*a^2*b^3 - 3*
C*a*b^4 + B*b^5)*tan(d*x + c)^2 + 2*(C*a^4*b - 3*B*a^3*b^2 - 3*C*a^2*b^3 +
B*a*b^4)*tan(d*x + c))*log((b^2*tan(d*x + c)^2 + 2*a*b*tan(d*x + c) + a^2)/
(tan(d*x + c)^2 + 1)) - 2*(2*C*a^4*b - 3*B*a^3*b^2 - 3*C*a^2*b^3 + 3*B*a*b^
4 + C*b^5 - 2*(B*a^4*b + 3*C*a^3*b^2 - 3*B*a^2*b^3 - C*a*b^4)*d*x)*tan(d*x
+ c))/((a^6*b^2 + 3*a^4*b^4 + 3*a^2*b^6 + b^8)*d*tan(d*x + c)^2 + 2*(a^7*b
+ 3*a^5*b^3 + 3*a^3*b^5 + a*b^7)*d*tan(d*x + c) + (a^8 + 3*a^6*b^2 + 3*a^4*
b^4 + a^2*b^6)*d)
```

Sympy [F(-2)]

Exception generated.

$$\int \frac{\cot(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx = \text{Exception raised: AttributeError}$$

```
[In] integrate(cot(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c))**3,x)
```

```
[Out] Exception raised: AttributeError >> 'NoneType' object has no attribute 'pri
mitive'
```

Maxima [A] (verification not implemented)

none

Time = 0.32 (sec) , antiderivative size = 321, normalized size of antiderivative = 1.83

$$\int \frac{\cot(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx$$
$$=\frac{\frac{2\,(Ba^3+3\,Ca^2b-3\,Bab^2-Cb^3)(dx+c)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}-\frac{2\,(Ca^3-3\,Ba^2b-3\,Cab^2+Bb^3)\log(b\tan(dx+c)+a)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}+\frac{(Ca^3-3\,Ba^2b-3\,Cab^2+Bb^3)\log\left(\tan(dx+c)^2\right.}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}}{2\,d}$$

```
[In] integrate(cot(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x, al
gorithm="maxima")
```

```
[Out] 1/2*(2*(B*a^3 + 3*C*a^2*b - 3*B*a*b^2 - C*b^3)*(d*x + c)/(a^6 + 3*a^4*b^2 +
 3*a^2*b^4 + b^6) - 2*(C*a^3 - 3*B*a^2*b - 3*C*a*b^2 + B*b^3)*log(b*tan(d*x
 + c) + a)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) + (C*a^3 - 3*B*a^2*b - 3*C*a
*b^2 + B*b^3)*log(tan(d*x + c)^2 + 1)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) +
 (3*C*a^3 - 5*B*a^2*b - C*a*b^2 - B*b^3 + 2*(C*a^2*b - 2*B*a*b^2 - C*b^3)*t
an(d*x + c))/(a^6 + 2*a^4*b^2 + a^2*b^4 + (a^4*b^2 + 2*a^2*b^4 + b^6)*tan(d
*x + c)^2 + 2*(a^5*b + 2*a^3*b^3 + a*b^5)*tan(d*x + c)))/d
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 409 vs. 2(171) = 342.

Time = 1.25 (sec) , antiderivative size = 409, normalized size of antiderivative = 2.34

$$\int \frac{\cot(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx$$
$$=\frac{\frac{2\,(Ba^3+3\,Ca^2b-3\,Bab^2-Cb^3)(dx+c)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}+\frac{(Ca^3-3\,Ba^2b-3\,Cab^2+Bb^3)\log\left(\tan(dx+c)^2+1\right)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}-\frac{2\,(Ca^3b-3\,Ba^2b^2-3\,Cab^3+Bb^4)\log(|b\tan(dx+}{a^6b+3\,a^4b^3+3\,a^2b^5+b^7}}{}$$

```
[In] integrate(cot(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x, al
gorithm="giac")
```

```
[Out] 1/2*(2*(B*a^3 + 3*C*a^2*b - 3*B*a*b^2 - C*b^3)*(d*x + c)/(a^6 + 3*a^4*b^2 +
 3*a^2*b^4 + b^6) + (C*a^3 - 3*B*a^2*b - 3*C*a*b^2 + B*b^3)*log(tan(d*x + c
)^2 + 1)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) - 2*(C*a^3*b - 3*B*a^2*b^2 - 3
*C*a*b^3 + B*b^4)*log(abs(b*tan(d*x + c) + a))/(a^6*b + 3*a^4*b^3 + 3*a^2*b
^5 + b^7) + (3*C*a^3*b^2*tan(d*x + c)^2 - 9*B*a^2*b^3*tan(d*x + c)^2 - 9*C*
a*b^4*tan(d*x + c)^2 + 3*B*b^5*tan(d*x + c)^2 + 8*C*a^4*b*tan(d*x + c) - 22
*B*a^3*b^2*tan(d*x + c) - 18*C*a^2*b^3*tan(d*x + c) + 2*B*a*b^4*tan(d*x + c
) - 2*C*b^5*tan(d*x + c) + 6*C*a^5 - 14*B*a^4*b - 7*C*a^3*b^2 - 3*B*a^2*b^3
 - C*a*b^4 - B*b^5)/((a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6)*(b*tan(d*x + c) +
a)^2))/d
```

Mupad [B] (verification not implemented)

Time = 8.56 (sec) , antiderivative size = 279, normalized size of antiderivative = 1.59

$$\int \frac{\cot(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^3}\,dx$$
$$= \frac{\ln\left(a+b\tan(c+d\,x)\right)\,\left(\frac{3\,B\,b-C\,a}{{\left(a^2+b^2\right)}^2}-\frac{4\,b^2\,\left(B\,b-C\,a\right)}{{\left(a^2+b^2\right)}^3}\right)}{d} - \frac{\frac{-3\,C\,a^3+5\,B\,a^2\,b+C\,a\,b^2+B\,b^3}{2\,\left(a^4+2\,a^2\,b^2+b^4\right)}+\frac{\tan(c+d\,x)\,\left(-C\,a^2\,b+2\,B\,a\,b^2+C\,b^3\right)}{a^4+2\,a^2\,b^2+b^4}}{d\,\left(a^2+2\,a\,b\tan\left(c+d\,x\right)+b^2\tan{\left(c+d\,x\right)}^2\right)} + \frac{\ln\left(\tan(c+d\,x)-\mathrm{i}\right)\,\left(-C+B\,1\mathrm{i}\right)}{2\,d\,\left(-a^3-a^2\,b\,3\mathrm{i}+3\,a\,b^2+b^3\,1\mathrm{i}\right)} + \frac{\ln\left(\tan(c+d\,x)+1\mathrm{i}\right)\,\left(B-C\,1\mathrm{i}\right)}{2\,d\,\left(-a^3\,1\mathrm{i}-3\,a^2\,b+a\,b^2\,3\mathrm{i}+b^3\right)}$$

```
[In] int((cot(c + d*x)*(B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c + d*x))
^3,x)
```

```
[Out] (log(a + b*tan(c + d*x))*((3*B*b - C*a)/(a^2 + b^2)^2 - (4*b^2*(B*b - C*a))
/(a^2 + b^2)^3))/d - ((B*b^3 - 3*C*a^3 + 5*B*a^2*b + C*a*b^2)/(2*(a^4 + b^4
 + 2*a^2*b^2)) + (tan(c + d*x)*(C*b^3 + 2*B*a*b^2 - C*a^2*b))/(a^4 + b^4 +
2*a^2*b^2))/(d*(a^2 + b^2*tan(c + d*x)^2 + 2*a*b*tan(c + d*x))) + (log(tan(
c + d*x) - 1i)*(B*1i - C))/(2*d*(3*a*b^2 - a^2*b*3i - a^3 + b^3*1i)) + (log
(tan(c + d*x) + 1i)*(B - C*1i))/(2*d*(a*b^2*3i - 3*a^2*b - a^3*1i + b^3))
```

3.43 $\int \frac{\cot^2(c+dx)(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^3}\,dx$

Optimal result 364
Rubi [A] (verified) 364
Mathematica [C] (verified) 367
Maple [A] (verified) 368
Fricas [B] (verification not implemented) 368
Sympy [F(-2)] 369
Maxima [A] (verification not implemented) 369
Giac [B] (verification not implemented) 370
Mupad [B] (verification not implemented) 370

Optimal result

Integrand size = 40, antiderivative size = 215

$$\begin{aligned}
&\int \frac{\cot^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx \\
&= -\frac{(3a^2bB-b^3B-a^3C+3ab^2C)\,x}{(a^2+b^2)^3}+\frac{B\log(\sin(c+dx))}{a^3d} \\
&\quad -\frac{b(6a^4bB+3a^2b^3B+b^5B-3a^5C+a^3b^2C)\log(a\cos(c+dx)+b\sin(c+dx))}{a^3\left(a^2+b^2\right)^3d} \\
&\quad +\frac{b(bB-aC)}{2a\left(a^2+b^2\right)d(a+b\tan(c+dx))^2}+\frac{b(3a^2bB+b^3B-2a^3C)}{a^2\left(a^2+b^2\right)^2d(a+b\tan(c+dx))}
\end{aligned}$$

```
[Out] -(3*B*a^2*b-B*b^3-C*a^3+3*C*a*b^2)*x/(a^2+b^2)^3+B*ln(sin(d*x+c))/a^3/d-b*(
6*B*a^4*b+3*B*a^2*b^3+B*b^5-3*C*a^5+C*a^3*b^2)*ln(a*cos(d*x+c)+b*sin(d*x+c)
)/a^3/(a^2+b^2)^3/d+1/2*b*(B*b-C*a)/a/(a^2+b^2)/d/(a+b*tan(d*x+c))^2+b*(3*B
*a^2*b+B*b^3-2*C*a^3)/a^2/(a^2+b^2)^2/d/(a+b*tan(d*x+c))
```

Rubi [A] (verified)

Time = 0.80 (sec) , antiderivative size = 215, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.150, Rules used

= {3713, 3690, 3730, 3732, 3611, 3556}

$$\int \frac{\cot^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx$$
$$=\frac{B\log(\sin(c+dx))}{a^3d}+\frac{b(bB-aC)}{2ad\left(a^2+b^2\right)(a+b\tan(c+dx))^2}$$
$$+\frac{b(-2a^3C+3a^2bB+b^3B)}{a^2d\left(a^2+b^2\right)^2(a+b\tan(c+dx))}-\frac{x(a^3(-C)+3a^2bB+3ab^2C-b^3B)}{\left(a^2+b^2\right)^3}$$
$$-\frac{b(-3a^5C+6a^4bB+a^3b^2C+3a^2b^3B+b^5B)\log(a\cos(c+dx)+b\sin(c+dx))}{a^3d\left(a^2+b^2\right)^3}$$

```
[In] Int[(Cot[c + d*x]^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c + d*x
])^3,x]
```

```
[Out] -(((3*a^2*b*B - b^3*B - a^3*C + 3*a*b^2*C)*x)/(a^2 + b^2)^3) + (B*Log[Sin[c
 + d*x]])/(a^3*d) - (b*(6*a^4*b*B + 3*a^2*b^3*B + b^5*B - 3*a^5*C + a^3*b^2
*C)*Log[a*Cos[c + d*x] + b*Sin[c + d*x]])/(a^3*(a^2 + b^2)^3*d) + (b*(b*B -
 a*C))/(2*a*(a^2 + b^2)*d*(a + b*Tan[c + d*x])^2) + (b*(3*a^2*b*B + b^3*B -
 2*a^3*C))/(a^2*(a^2 + b^2)^2*d*(a + b*Tan[c + d*x]))
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3611

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c/(b*f))*Log[RemoveContent[a*Cos[e + f*x] + b*Si
n[e + f*x], x]], x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3690

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Si
mp[b*(A*b - a*B)*(a + b*Tan[e + f*x])^(m + 1)*((c + d*Tan[e + f*x])^(n + 1)
/(f*(m + 1)*(b*c - a*d)*(a^2 + b^2))), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^
2 + b^2)), Int[(a + b*Tan[e + f*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[b*B
*(b*c*(m + 1) + a*d*(n + 1)) + A*(a*(b*c - a*d)*(m + 1) - b^2*d*(m + n + 2)
) - (A*b - a*B)*(b*c - a*d)*(m + 1)*Tan[e + f*x] - b*d*(A*b - a*B)*(m + n +
 2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, n}, x] &&
NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &
& (IntegerQ[m] || IntegersQ[2*m, 2*n]) &&  !(ILtQ[n, -1] && ( !IntegerQ[m]
|| (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rule 3730

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 - a*(b*B - a*C))*(a + b*Tan[e +
 f*x])^(m + 1)*((c + d*Tan[e + f*x])^(n + 1)/(f*(m + 1)*(b*c - a*d)*(a^2 +
b^2))), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3732

```
Int[((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^
2)/(((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])), x_Symbol] :> Simp[(a*(A*c - c*C + B*d) + b*(B*c - A*d + C*d))*(x/
((a^2 + b^2)*(c^2 + d^2))), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/((b*c - a*d)
*(a^2 + b^2)), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] - Dist
[(c^2*C - B*c*d + A*d^2)/((b*c - a*d)*(c^2 + d^2)), Int[(d - c*Tan[e + f*x]
)/(c + d*Tan[e + f*x]), x], x]) /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] &&
 NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0]
```

Rubi steps

$$\text{integral} = \int \frac{\cot(c+dx)(B+C\tan(c+dx))}{(a+b\tan(c+dx))^3}\,dx$$

$$= \frac{b(bB-aC)}{2a\left(a^2+b^2\right)d(a+b\tan(c+dx))^2} + \frac{\int \frac{\cot(c+dx)\left(2\left(a^2+b^2\right)B-2a(bB-aC)\tan(c+dx)+2b(bB-aC)\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx}{2a\left(a^2+b^2\right)}$$

$$= \frac{b(bB - aC)}{2a\left(a^2+b^2\right)d(a+b\tan(c+dx))^2} + \frac{b(3a^2bB + b^3B - 2a^3C)}{a^2\left(a^2+b^2\right)^2 d(a+b\tan(c+dx))}$$
$$+ \frac{\int \frac{\cot(c+dx)\left(2(a^2+b^2)^2B - 2a^2(2abB - a^2C + b^2C)\tan(c+dx) + 2b(3a^2bB + b^3B - 2a^3C)\tan^2(c+dx)\right)}{a+b\tan(c+dx)}dx}{2a^2\left(a^2+b^2\right)^2}$$

$$= -\frac{(3a^2bB - b^3B - a^3C + 3ab^2C)\,x}{\left(a^2+b^2\right)^3} + \frac{b(bB - aC)}{2a\left(a^2+b^2\right)d(a+b\tan(c+dx))^2}$$
$$+ \frac{b(3a^2bB + b^3B - 2a^3C)}{a^2\left(a^2+b^2\right)^2 d(a+b\tan(c+dx))} + \frac{B\int\cot(c+dx)\,dx}{a^3}$$
$$- \frac{\left(b(6a^4bB + 3a^2b^3B + b^5B - 3a^5C + a^3b^2C)\right)\int\frac{b-a\tan(c+dx)}{a+b\tan(c+dx)}dx}{a^3\left(a^2+b^2\right)^3}$$

$$= -\frac{(3a^2bB - b^3B - a^3C + 3ab^2C)\,x}{\left(a^2+b^2\right)^3} + \frac{B\log(\sin(c+dx))}{a^3d}$$
$$- \frac{b(6a^4bB + 3a^2b^3B + b^5B - 3a^5C + a^3b^2C)\log(a\cos(c+dx) + b\sin(c+dx))}{a^3\left(a^2+b^2\right)^3 d}$$
$$+ \frac{b(bB - aC)}{2a\left(a^2+b^2\right)d(a+b\tan(c+dx))^2} + \frac{b(3a^2bB + b^3B - 2a^3C)}{a^2\left(a^2+b^2\right)^2 d(a+b\tan(c+dx))}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 3.31 (sec) , antiderivative size = 223, normalized size of antiderivative = 1.04

$$\int \frac{\cot^2(c+dx)\left(B\tan(c+dx) + C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}dx$$
$$= \frac{-\frac{(B+iC)\log(i-\tan(c+dx))}{(a+ib)^3} + \frac{2B\log(\tan(c+dx))}{a^3} - \frac{(B-iC)\log(i+\tan(c+dx))}{(a-ib)^3} - \frac{2b(6a^4bB+3a^2b^3B+b^5B-3a^5C+a^3b^2C)\log(a+b\tan(c}{a^3(a^2+b^2)^3}}{2d}$$

```
[In] Integrate[(Cot[c + d*x]^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c
 + d*x])^3,x]
```

```
[Out] (-(((B + I*C)*Log[I - Tan[c + d*x]])/(a + I*b)^3) + (2*B*Log[Tan[c + d*x]])
/a^3 - ((B - I*C)*Log[I + Tan[c + d*x]])/(a - I*b)^3 - (2*b*(6*a^4*b*B + 3*
a^2*b^3*B + b^5*B - 3*a^5*C + a^3*b^2*C)*Log[a + b*Tan[c + d*x]])/(a^3*(a^2
 + b^2)^3) + (b*(b*B - a*C))/(a*(a^2 + b^2)*(a + b*Tan[c + d*x])^2) + (2*b*
(3*a^2*b*B + b^3*B - 2*a^3*C))/(a^2*(a^2 + b^2)^2*(a + b*Tan[c + d*x])))/(2
*d)
```

Maple [A] (verified)

Time = 0.66 (sec) , antiderivative size = 243, normalized size of antiderivative = 1.13

method	result
derivativedivides	$\frac{\frac{B\ln(\tan(dx+c))}{a^3}+\frac{\frac{\left(-B\,a^3+3Ba\,b^2-3C\,a^2b+C\,b^3\right)\ln\left(1+\tan(dx+c)^2\right)}{2}+\left(-3B\,a^2b+B\,b^3+C\,a^3-3C\,a\,b^2\right)\arctan(\tan(dx+c))}{\left(a^2+b^2\right)^3}+\frac{b(3B}{(a^2+b^2}}{d}$
default	$\frac{\frac{B\ln(\tan(dx+c))}{a^3}+\frac{\frac{\left(-B\,a^3+3Ba\,b^2-3C\,a^2b+C\,b^3\right)\ln\left(1+\tan(dx+c)^2\right)}{2}+\left(-3B\,a^2b+B\,b^3+C\,a^3-3C\,a\,b^2\right)\arctan(\tan(dx+c))}{\left(a^2+b^2\right)^3}+\frac{b(3B}{(a^2+b^2}}{d}$
parallelrisch	$-12b(B\,a^4b+\frac{1}{2}B\,a^2b^3+\frac{1}{6}B\,b^5-\frac{1}{2}C\,a^5+\frac{1}{6}C\,a^3b^2)(a+b\tan(dx+c))^2\ln(a+b\tan(dx+c))-a^3(a+b\tan(dx+c))^2(B\,a^3-3Ba$
norman	$\frac{-\frac{b^2\left(3B\,a^2b-B\,b^3-C\,a^3+3Ca\,b^2\right)x\tan(dx+c)^3}{\left(a^4+2a^2b^2+b^4\right)\left(a^2+b^2\right)}-\frac{b\left(4B\,a^2b^2+2B\,b^4-3C\,a^3b-Ca\,b^3\right)\tan(dx+c)^2}{d\,a^2\left(a^4+2a^2b^2+b^4\right)}-\frac{b^2\left(7B\,a^2b^2+3B\,b^4-5C\,a^3b-Ca\right.}{2d\,a^3\left(a^4+2a^2b^2+b^4\right.}}{\tan(dx+c)(a+b\tan(dx-}$
risch	$\frac{2iC\,b^3x}{a^6+3a^4b^2+3a^2b^4+b^6}-\frac{xC}{3ia^2b-ib^3-a^3+3a\,b^2}+\frac{6ib^4Bx}{(a^6+3a^4b^2+3a^2b^4+b^6)a}+\frac{2iC\,b^3c}{d(a^6+3a^4b^2+3a^2b^4+b^6)}+\frac{6}{(a^6+3a^4b^2}$

```
[In] int(cot(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x,method=
_RETURNVERBOSE)
```

```
[Out] 1/d*(1/a^3*B*ln(tan(d*x+c))+1/(a^2+b^2)^3*(1/2*(-B*a^3+3*B*a*b^2-3*C*a^2*b+
C*b^3)*ln(1+tan(d*x+c)^2)+(-3*B*a^2*b+B*b^3+C*a^3-3*C*a*b^2)*arctan(tan(d*x
+c)))+b*(3*B*a^2*b+B*b^3-2*C*a^3)/(a^2+b^2)^2/a^2/(a+b*tan(d*x+c))-b*(6*B*a
^4*b+3*B*a^2*b^3+B*b^5-3*C*a^5+C*a^3*b^2)/(a^2+b^2)^3/a^3*ln(a+b*tan(d*x+c)
)+1/2*(B*b-C*a)*b/(a^2+b^2)/a/(a+b*tan(d*x+c))^2)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 683 vs. 2(213) = 426.

Time = 0.33 (sec) , antiderivative size = 683, normalized size of antiderivative = 3.18

$$\int \frac{\cot^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx =$$
$$-\frac{7\,Ca^5b^3-9\,Ba^4b^4+Ca^3b^5-3\,Ba^2b^6-2\left(Ca^8-3\,Ba^7b-3\,Ca^6b^2+Ba^5b^3\right)dx-\left(5\,Ca^5b^3-7\,Ba^4b^4-\right.}{}$$

```
[In] integrate(cot(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x,
algorithm="fricas")
```

```
[Out] -1/2*(7*C*a^5*b^3 - 9*B*a^4*b^4 + C*a^3*b^5 - 3*B*a^2*b^6 - 2*(C*a^8 - 3*B*
a^7*b - 3*C*a^6*b^2 + B*a^5*b^3)*d*x - (5*C*a^5*b^3 - 7*B*a^4*b^4 - C*a^3*b
```

```
^5 - B*a^2*b^6 + 2*(C*a^6*b^2 - 3*B*a^5*b^3 - 3*C*a^4*b^4 + B*a^3*b^5)*d*x)
*tan(d*x + c)^2 - (B*a^8 + 3*B*a^6*b^2 + 3*B*a^4*b^4 + B*a^2*b^6 + (B*a^6*b
^2 + 3*B*a^4*b^4 + 3*B*a^2*b^6 + B*b^8)*tan(d*x + c)^2 + 2*(B*a^7*b + 3*B*a
^5*b^3 + 3*B*a^3*b^5 + B*a*b^7)*tan(d*x + c))*log(tan(d*x + c)^2/(tan(d*x +
 c)^2 + 1)) - (3*C*a^7*b - 6*B*a^6*b^2 - C*a^5*b^3 - 3*B*a^4*b^4 - B*a^2*b^
6 + (3*C*a^5*b^3 - 6*B*a^4*b^4 - C*a^3*b^5 - 3*B*a^2*b^6 - B*b^8)*tan(d*x +
 c)^2 + 2*(3*C*a^6*b^2 - 6*B*a^5*b^3 - C*a^4*b^4 - 3*B*a^3*b^5 - B*a*b^7)*t
an(d*x + c))*log((b^2*tan(d*x + c)^2 + 2*a*b*tan(d*x + c) + a^2)/(tan(d*x +
 c)^2 + 1)) - 2*(3*C*a^6*b^2 - 4*B*a^5*b^3 - 3*C*a^4*b^4 + 3*B*a^3*b^5 + B*
a*b^7 + 2*(C*a^7*b - 3*B*a^6*b^2 - 3*C*a^5*b^3 + B*a^4*b^4)*d*x)*tan(d*x +
c))/((a^9*b^2 + 3*a^7*b^4 + 3*a^5*b^6 + a^3*b^8)*d*tan(d*x + c)^2 + 2*(a^10
*b + 3*a^8*b^3 + 3*a^6*b^5 + a^4*b^7)*d*tan(d*x + c) + (a^11 + 3*a^9*b^2 +
3*a^7*b^4 + a^5*b^6)*d)
```

Sympy [F(-2)]

Exception generated.

$$\int \frac{\cot^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx = \text{Exception raised: AttributeError}$$

```
[In] integrate(cot(d*x+c)**2*(B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c))**3,
x)
```

```
[Out] Exception raised: AttributeError >> 'NoneType' object has no attribute 'pri
mitive'
```

Maxima [A] (verification not implemented)

none

Time = 0.34 (sec) , antiderivative size = 372, normalized size of antiderivative = 1.73

$$\int \frac{\cot^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx = \frac{\frac{2\,(Ca^3-3\,Ba^2b-3\,Cab^2+Bb^3)(dx+c)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}+\frac{2\,(3\,Ca^5b-6\,Ba^4b^2-Ca^3b^3-3\,Ba^2b^4-Bb^6)\log(b\tan(dx+c)+a)}{a^9+3\,a^7b^2+3\,a^5b^4+a^3b^6}-\frac{(Ba^3+3\,Ca^2b-3\,Bab^2-Cb^3)\,\mathrm{lo}}{a^6+3\,a^4b^2+3\,a^2b^4}}{2\,d}$$

```
[In] integrate(cot(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x,
algorithm="maxima")
```

```
[Out] 1/2*(2*(C*a^3 - 3*B*a^2*b - 3*C*a*b^2 + B*b^3)*(d*x + c)/(a^6 + 3*a^4*b^2 +
 3*a^2*b^4 + b^6) + 2*(3*C*a^5*b - 6*B*a^4*b^2 - C*a^3*b^3 - 3*B*a^2*b^4 -
B*b^6)*log(b*tan(d*x + c) + a)/(a^9 + 3*a^7*b^2 + 3*a^5*b^4 + a^3*b^6) - (B
*a^3 + 3*C*a^2*b - 3*B*a*b^2 - C*b^3)*log(tan(d*x + c)^2 + 1)/(a^6 + 3*a^4*
```

```
b^2 + 3*a^2*b^4 + b^6) - (5*C*a^4*b - 7*B*a^3*b^2 + C*a^2*b^3 - 3*B*a*b^4 +
 2*(2*C*a^3*b^2 - 3*B*a^2*b^3 - B*b^5)*tan(d*x + c))/(a^8 + 2*a^6*b^2 + a^4
*b^4 + (a^6*b^2 + 2*a^4*b^4 + a^2*b^6)*tan(d*x + c)^2 + 2*(a^7*b + 2*a^5*b^
3 + a^3*b^5)*tan(d*x + c)) + 2*B*log(tan(d*x + c))/a^3)/d
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 479 vs. $2(213) = 426$.

Time = 1.24 (sec) , antiderivative size = 479, normalized size of antiderivative = 2.23

$$\int \frac{\cot^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx$$
$$= \frac{\frac{2\left(Ca^3-3Ba^2b-3Cab^2+Bb^3\right)(dx+c)}{a^6+3a^4b^2+3a^2b^4+b^6} - \frac{\left(Ba^3+3Ca^2b-3Bab^2-Cb^3\right)\log\left(\tan(dx+c)^2+1\right)}{a^6+3a^4b^2+3a^2b^4+b^6} + \frac{2\left(3Ca^5b^2-6Ba^4b^3-Ca^3b^4-3Ba^2b^5-Bb^7\right)\log}{a^9b+3a^7b^3+3a^5b^5+a^3b^7}}{}$$

```
[In] integrate(cot(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x,
algorithm="giac")
```

```
[Out] 1/2*(2*(C*a^3 - 3*B*a^2*b - 3*C*a*b^2 + B*b^3)*(d*x + c)/(a^6 + 3*a^4*b^2 +
 3*a^2*b^4 + b^6) - (B*a^3 + 3*C*a^2*b - 3*B*a*b^2 - C*b^3)*log(tan(d*x + c
)^2 + 1)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) + 2*(3*C*a^5*b^2 - 6*B*a^4*b^3
 - C*a^3*b^4 - 3*B*a^2*b^5 - B*b^7)*log(abs(b*tan(d*x + c) + a))/(a^9*b + 3
*a^7*b^3 + 3*a^5*b^5 + a^3*b^7) + 2*B*log(abs(tan(d*x + c)))/a^3 - (9*C*a^5
*b^3*tan(d*x + c)^2 - 18*B*a^4*b^4*tan(d*x + c)^2 - 3*C*a^3*b^5*tan(d*x + c
)^2 - 9*B*a^2*b^6*tan(d*x + c)^2 - 3*B*b^8*tan(d*x + c)^2 + 22*C*a^6*b^2*ta
n(d*x + c) - 42*B*a^5*b^3*tan(d*x + c) - 2*C*a^4*b^4*tan(d*x + c) - 26*B*a^
3*b^5*tan(d*x + c) - 8*B*a*b^7*tan(d*x + c) + 14*C*a^7*b - 25*B*a^6*b^2 + 3
*C*a^5*b^3 - 19*B*a^4*b^4 + C*a^3*b^5 - 6*B*a^2*b^6)/((a^9 + 3*a^7*b^2 + 3*
a^5*b^4 + a^3*b^6)*(b*tan(d*x + c) + a)^2))/d
```

Mupad [B] (verification not implemented)

Time = 10.87 (sec) , antiderivative size = 315, normalized size of antiderivative = 1.47

$$\int \frac{\cot^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx$$
$$= \frac{\frac{-5Ca^3b+7Ba^2b^2-Cab^3+3Bb^4}{2a\left(a^4+2a^2b^2+b^4\right)} + \frac{\tan(c+dx)\left(-2Ca^3b^2+3Ba^2b^3+Bb^5\right)}{a^2\left(a^4+2a^2b^2+b^4\right)}}{d\left(a^2+2ab\tan(c+dx)+b^2\tan(c+dx)^2\right)} + \frac{B\ln(\tan(c+dx))}{a^3d}$$
$$+ \frac{\ln(\tan(c+dx)-\mathrm{i})\,(-C+B\,1\mathrm{i})}{2d\left(-a^3\,1\mathrm{i}+3a^2b+ab^2\,3\mathrm{i}-b^3\right)} + \frac{\ln(\tan(c+dx)+1\mathrm{i})\,(B-C\,1\mathrm{i})}{2d\left(-a^3+a^2b\,3\mathrm{i}+3ab^2-b^3\,1\mathrm{i}\right)}$$
$$- \frac{b\ln(a+b\tan(c+dx))\left(-3Ca^5+6Ba^4b+Ca^3b^2+3Ba^2b^3+Bb^5\right)}{a^3d\left(a^2+b^2\right)^3}$$

```
[In] int((cot(c + d*x)^2*(B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c + d*x
))^3,x)
```

```
[Out] ((3*B*b^4 + 7*B*a^2*b^2 - C*a*b^3 - 5*C*a^3*b)/(2*a*(a^4 + b^4 + 2*a^2*b^2)
) + (tan(c + d*x)*(B*b^5 + 3*B*a^2*b^3 - 2*C*a^3*b^2))/(a^2*(a^4 + b^4 + 2*
a^2*b^2)))/(d*(a^2 + b^2*tan(c + d*x)^2 + 2*a*b*tan(c + d*x))) + (B*log(tan
(c + d*x)))/(a^3*d) + (log(tan(c + d*x) - 1i)*(B*1i - C))/(2*d*(a*b^2*3i +
3*a^2*b - a^3*1i - b^3)) + (log(tan(c + d*x) + 1i)*(B - C*1i))/(2*d*(3*a*b^
2 + a^2*b*3i - a^3 - b^3*1i)) - (b*log(a + b*tan(c + d*x))*(B*b^5 - 3*C*a^5
 + 3*B*a^2*b^3 + C*a^3*b^2 + 6*B*a^4*b))/(a^3*d*(a^2 + b^2)^3)
```

3.44 $\int \frac{\cot^3(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx$

Optimal result 372
Rubi [A] (verified) 373
Mathematica [C] (verified) 376
Maple [A] (verified) 377
Fricas [B] (verification not implemented) 377
Sympy [F(-2)] 378
Maxima [A] (verification not implemented) 378
Giac [A] (verification not implemented) 379
Mupad [B] (verification not implemented) 380

Optimal result

Integrand size = 40, antiderivative size = 287

$$\begin{aligned}
&\int \frac{\cot^3(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx\\
&=-\frac{(a^3B-3ab^2B+3a^2bC-b^3C)\,x}{(a^2+b^2)^3}-\frac{(3bB-aC)\log(\sin(c+dx))}{a^4d}\\
&\quad+\frac{b^2(10a^4bB+9a^2b^3B+3b^5B-6a^5C-3a^3b^2C-ab^4C)\log(a\cos(c+dx)+b\sin(c+dx))}{a^4\,(a^2+b^2)^3\,d}\\
&\quad-\frac{b(2a^2B+3b^2B-abC)}{2a^2\,(a^2+b^2)\,d(a+b\tan(c+dx))^2}-\frac{B\cot(c+dx)}{ad(a+b\tan(c+dx))^2}\\
&\quad-\frac{b(a^4B+6a^2b^2B+3b^4B-3a^3bC-ab^3C)}{a^3\,(a^2+b^2)^2\,d(a+b\tan(c+dx))}
\end{aligned}$$

```
[Out] -(B*a^3-3*B*a*b^2+3*C*a^2*b-C*b^3)*x/(a^2+b^2)^3-(3*B*b-C*a)*ln(sin(d*x+c))
/a^4/d+b^2*(10*B*a^4*b+9*B*a^2*b^3+3*B*b^5-6*C*a^5-3*C*a^3*b^2-C*a*b^4)*ln(
a*cos(d*x+c)+b*sin(d*x+c))/a^4/(a^2+b^2)^3/d-1/2*b*(2*B*a^2+3*B*b^2-C*a*b)/
a^2/(a^2+b^2)/d/(a+b*tan(d*x+c))^2-B*cot(d*x+c)/a/d/(a+b*tan(d*x+c))^2-b*(B
*a^4+6*B*a^2*b^2+3*B*b^4-3*C*a^3*b-C*a*b^3)/a^3/(a^2+b^2)^2/d/(a+b*tan(d*x+
c))
```

Rubi [A] (verified)

Time = 1.09 (sec) , antiderivative size = 287, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.150, Rules used = {3713, 3690, 3730, 3732, 3611, 3556}

$$\int \frac{\cot^3(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx$$
$$= -\frac{(3bB-aC)\log(\sin(c+dx))}{a^4d}-\frac{b(2a^2B-abC+3b^2B)}{2a^2d\,(a^2+b^2)\,(a+b\tan(c+dx))^2}$$
$$-\frac{x(a^3B+3a^2bC-3ab^2B-b^3C)}{(a^2+b^2)^3}-\frac{b(a^4B-3a^3bC+6a^2b^2B-ab^3C+3b^4B)}{a^3d\,(a^2+b^2)^2\,(a+b\tan(c+dx))}$$
$$+\frac{b^2(-6a^5C+10a^4bB-3a^3b^2C+9a^2b^3B-ab^4C+3b^5B)\log(a\cos(c+dx)+b\sin(c+dx))}{a^4d\,(a^2+b^2)^3}$$
$$-\frac{B\cot(c+dx)}{ad(a+b\tan(c+dx))^2}$$

```
[In] Int[(Cot[c + d*x]^3*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c + d*x
])^3,x]
```

```
[Out] -(((a^3*B - 3*a*b^2*B + 3*a^2*b*C - b^3*C)*x)/(a^2 + b^2)^3) - ((3*b*B - a*
C)*Log[Sin[c + d*x]])/(a^4*d) + (b^2*(10*a^4*b*B + 9*a^2*b^3*B + 3*b^5*B -
6*a^5*C - 3*a^3*b^2*C - a*b^4*C)*Log[a*Cos[c + d*x] + b*Sin[c + d*x]])/(a^4
*(a^2 + b^2)^3*d) - (b*(2*a^2*B + 3*b^2*B - a*b*C))/(2*a^2*(a^2 + b^2)*d*(a
 + b*Tan[c + d*x])^2) - (B*Cot[c + d*x])/(a*d*(a + b*Tan[c + d*x])^2) - (b*
(a^4*B + 6*a^2*b^2*B + 3*b^4*B - 3*a^3*b*C - a*b^3*C))/(a^3*(a^2 + b^2)^2*d
*(a + b*Tan[c + d*x]))
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3611

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c/(b*f))*Log[RemoveContent[a*Cos[e + f*x] + b*Si
n[e + f*x], x]], x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3690

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Si
mp[b*(A*b - a*B)*(a + b*Tan[e + f*x])^(m + 1)*((c + d*Tan[e + f*x])^(n + 1)
/(f*(m + 1)*(b*c - a*d)*(a^2 + b^2))), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^
```

```
2 + b^2)), Int[(a + b*Tan[e + f*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[b*B
*(b*c*(m + 1) + a*d*(n + 1)) + A*(a*(b*c - a*d)*(m + 1) - b^2*d*(m + n + 2)
) - (A*b - a*B)*(b*c - a*d)*(m + 1)*Tan[e + f*x] - b*d*(A*b - a*B)*(m + n +
 2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, n}, x] &&
NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &
& (IntegerQ[m] || IntegersQ[2*m, 2*n]) &&  !(ILtQ[n, -1] && ( !IntegerQ[m]
|| (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3713

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rule 3730

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 - a*(b*B - a*C))*(a + b*Tan[e +
 f*x])^(m + 1)*((c + d*Tan[e + f*x])^(n + 1)/(f*(m + 1)*(b*c - a*d)*(a^2 +
b^2))), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3732

```
Int[((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^
2)/(((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])), x_Symbol] :> Simp[(a*(A*c - c*C + B*d) + b*(B*c - A*d + C*d))*(x/
((a^2 + b^2)*(c^2 + d^2))), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/((b*c - a*d)
*(a^2 + b^2)), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] - Dist
[(c^2*C - B*c*d + A*d^2)/((b*c - a*d)*(c^2 + d^2)), Int[(d - c*Tan[e + f*x]
)/(c + d*Tan[e + f*x]), x], x]) /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] &&
 NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0]
```

Rubi steps

$$\text{integral} = \int \frac{\cot^2(c+dx)(B+C\tan(c+dx))}{(a+b\tan(c+dx))^3}\,dx$$

$$= -\frac{B\cot(c+dx)}{ad(a+b\tan(c+dx))^2} - \frac{\int \frac{\cot(c+dx)(3bB-aC+aB\tan(c+dx)+3bB\tan^2(c+dx))}{(a+b\tan(c+dx))^3}\,dx}{a}$$

$$= -\frac{b(2a^2B+3b^2B-abC)}{2a^2(a^2+b^2)\,d(a+b\tan(c+dx))^2} - \frac{B\cot(c+dx)}{ad(a+b\tan(c+dx))^2} - \frac{\int \frac{\cot(c+dx)(2(a^2+b^2)(3bB-aC)+2a^2(aB+bC)\tan(c+dx)+2b(2a^2B+3b^2B-abC)\tan^2(c+dx))}{(a+b\tan(c+dx))^2}\,dx}{2a^2(a^2+b^2)}$$

$$= -\frac{b(2a^2B+3b^2B-abC)}{2a^2(a^2+b^2)\,d(a+b\tan(c+dx))^2} - \frac{B\cot(c+dx)}{ad(a+b\tan(c+dx))^2} - \frac{b(a^4B+6a^2b^2B+3b^4B-3a^3bC-ab^3C)}{a^3(a^2+b^2)^2\,d(a+b\tan(c+dx))} - \frac{\int \frac{\cot(c+dx)\left(2(a^2+b^2)^2(3bB-aC)+2a^3(a^2B-b^2B+2abC)\tan(c+dx)+2b(a^4B+6a^2b^2B+3b^4B-3a^3bC-ab^3C)\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,d}{2a^3(a^2+b^2)^2}$$

$$= -\frac{(a^3B-3ab^2B+3a^2bC-b^3C)\,x}{(a^2+b^2)^3} - \frac{b(2a^2B+3b^2B-abC)}{2a^2(a^2+b^2)\,d(a+b\tan(c+dx))^2} - \frac{B\cot(c+dx)}{ad(a+b\tan(c+dx))^2} - \frac{b(a^4B+6a^2b^2B+3b^4B-3a^3bC-ab^3C)}{a^3(a^2+b^2)^2\,d(a+b\tan(c+dx))} - \frac{(3bB-aC)\int\cot(c+dx)\,dx}{a^4} + \frac{(b^2(10a^4bB+9a^2b^3B+3b^5B-6a^5C-3a^3b^2C-ab^4C))\int\frac{b-a\tan(c+dx)}{a+b\tan(c+dx)}\,dx}{a^4(a^2+b^2)^3}$$

$$= -\frac{(a^3B-3ab^2B+3a^2bC-b^3C)\,x}{(a^2+b^2)^3} - \frac{(3bB-aC)\log(\sin(c+dx))}{a^4d} + \frac{b^2(10a^4bB+9a^2b^3B+3b^5B-6a^5C-3a^3b^2C-ab^4C)\log(a\cos(c+dx)+b\sin(c+dx))}{a^4(a^2+b^2)^3\,d} - \frac{b(2a^2B+3b^2B-abC)}{2a^2(a^2+b^2)\,d(a+b\tan(c+dx))^2} - \frac{B\cot(c+dx)}{ad(a+b\tan(c+dx))^2} - \frac{b(a^4B+6a^2b^2B+3b^4B-3a^3bC-ab^3C)}{a^3(a^2+b^2)^2\,d(a+b\tan(c+dx))}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 6.45 (sec) , antiderivative size = 288, normalized size of antiderivative = 1.00

$$
\begin{aligned}
&\int \frac{\cot^3(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx \\
&= -\frac{B\cot(c+dx)}{a^3d}+\frac{(B+iC)\log(i-\tan(c+dx))}{2(ia-b)^3d} \\
&\quad -\frac{(3bB-aC)\log(\tan(c+dx))}{a^4d}-\frac{(iB+C)\log(i+\tan(c+dx))}{2(a-ib)^3d} \\
&\quad +\frac{b^2(10a^4bB+9a^2b^3B+3b^5B-6a^5C-3a^3b^2C-ab^4C)\log(a+b\tan(c+dx))}{a^4\left(a^2+b^2\right)^3d} \\
&\quad -\frac{b^2(bB-aC)}{2a^2\left(a^2+b^2\right)d(a+b\tan(c+dx))^2}-\frac{b^2(4a^2bB+2b^3B-3a^3C-ab^2C)}{a^3\left(a^2+b^2\right)^2d(a+b\tan(c+dx))}
\end{aligned}
$$

```
[In] Integrate[(Cot[c + d*x]^3*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c
 + d*x])^3,x]
```

```
[Out] -((B*Cot[c + d*x])/(a^3*d)) + ((B + I*C)*Log[I - Tan[c + d*x]])/(2*(I*a - b
)^3*d) - ((3*b*B - a*C)*Log[Tan[c + d*x]])/(a^4*d) - ((I*B + C)*Log[I + Tan
[c + d*x]])/(2*(a - I*b)^3*d) + (b^2*(10*a^4*b*B + 9*a^2*b^3*B + 3*b^5*B -
6*a^5*C - 3*a^3*b^2*C - a*b^4*C)*Log[a + b*Tan[c + d*x]])/(a^4*(a^2 + b^2)^
3*d) - (b^2*(b*B - a*C))/(2*a^2*(a^2 + b^2)*d*(a + b*Tan[c + d*x])^2) - (b^
2*(4*a^2*b*B + 2*b^3*B - 3*a^3*C - a*b^2*C))/(a^3*(a^2 + b^2)^2*d*(a + b*Ta
n[c + d*x]))
```

Maple [A] (verified)

Time = 1.03 (sec) , antiderivative size = 289, normalized size of antiderivative = 1.01

method	result
derivativedivides	$\frac{-\frac{B}{a^3\tan(dx+c)}+\frac{(-3Bb+Ca)\ln(\tan(dx+c))}{a^4}+\frac{\frac{\left(3B\,a^2b-B\,b^3-C\,a^3+3Ca\,b^2\right)\ln\left(1+\tan(dx+c)^2\right)}{2}+\left(-B\,a^3+3Ba\,b^2-3C\,a^2b+C\,b^3\right)}{\left(a^2+b^2\right)^3}}{}$
default	$\frac{-\frac{B}{a^3\tan(dx+c)}+\frac{(-3Bb+Ca)\ln(\tan(dx+c))}{a^4}+\frac{\frac{\left(3B\,a^2b-B\,b^3-C\,a^3+3Ca\,b^2\right)\ln\left(1+\tan(dx+c)^2\right)}{2}+\left(-B\,a^3+3Ba\,b^2-3C\,a^2b+C\,b^3\right)}{\left(a^2+b^2\right)^3}}{}$
parallelrisch	$\frac{20b^2(a+b\tan(dx+c))^2\left(B\,a^4b+\frac{9}{10}B\,a^2b^3+\frac{3}{10}B\,b^5-\frac{3}{5}C\,a^5-\frac{3}{10}C\,a^3b^2-\frac{1}{10}Ca\,b^4\right)\ln(a+b\tan(dx+c))+3a^4\left(B\,a^2b-\frac{1}{3}B\,b\right.}{}$
norman	$\frac{\frac{b\left(3B\,a^4b+11B\,a^2b^3+6B\,b^5-4C\,a^3b^2-2Ca\,b^4\right)\tan(dx+c)^3}{d\,a^3\left(a^4+2a^2b^2+b^4\right)}-\frac{B\tan(dx+c)}{ad}+\frac{b^2\left(4B\,a^4b+17B\,a^2b^3+9B\,b^5-7C\,a^3b^2-3Ca\,b^4\right)\tan(d}{2a^4d\left(a^4+2a^2b^2+b^4\right)}}{\tan(}$
risch	Expression too large to display

```
[In] int(cot(d*x+c)^3*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x,method=
_RETURNVERBOSE)
```

```
[Out] 1/d*(-1/a^3*B/tan(d*x+c)+(-3*B*b+C*a)/a^4*ln(tan(d*x+c))+1/(a^2+b^2)^3*(1/2
*(3*B*a^2*b-B*b^3-C*a^3+3*C*a*b^2)*ln(1+tan(d*x+c)^2)+(-B*a^3+3*B*a*b^2-3*C
*a^2*b+C*b^3)*arctan(tan(d*x+c)))-b^2*(4*B*a^2*b+2*B*b^3-3*C*a^3-C*a*b^2)/(
a^2+b^2)^2/a^3/(a+b*tan(d*x+c))+b^2*(10*B*a^4*b+9*B*a^2*b^3+3*B*b^5-6*C*a^5
-3*C*a^3*b^2-C*a*b^4)/(a^2+b^2)^3/a^4*ln(a+b*tan(d*x+c))-1/2*(B*b-C*a)*b^2/
(a^2+b^2)/a^2/(a+b*tan(d*x+c))^2)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 917 vs. 2(283) = 566.

Time = 0.37 (sec) , antiderivative size = 917, normalized size of antiderivative = 3.20

$$\int \frac{\cot^3(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx =$$

$$-\frac{2\,Ba^9+6\,Ba^7b^2+6\,Ba^5b^4+2\,Ba^3b^6+(7\,Ca^5b^4-9\,Ba^4b^5+Ca^3b^6-3\,Ba^2b^7+2\,(Ba^7b^2+3\,Ca^6b^3-}{}$$

```
[In] integrate(cot(d*x+c)^3*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x,
algorithm="fricas")
```

```
[Out] -1/2*(2*B*a^9 + 6*B*a^7*b^2 + 6*B*a^5*b^4 + 2*B*a^3*b^6 + (7*C*a^5*b^4 - 9*
B*a^4*b^5 + C*a^3*b^6 - 3*B*a^2*b^7 + 2*(B*a^7*b^2 + 3*C*a^6*b^3 - 3*B*a^5*
b^4 - C*a^4*b^5)*d*x)*tan(d*x + c)^3 + 2*(B*a^7*b^2 + 4*C*a^6*b^3 - 2*B*a^5
```

```
*b^4 - 3*C*a^4*b^5 + 6*B*a^3*b^6 - C*a^2*b^7 + 3*B*a*b^8 + 2*(B*a^8*b + 3*C
*a^7*b^2 - 3*B*a^6*b^3 - C*a^5*b^4)*d*x)*tan(d*x + c)^2 - ((C*a^7*b^2 - 3*B
*a^6*b^3 + 3*C*a^5*b^4 - 9*B*a^4*b^5 + 3*C*a^3*b^6 - 9*B*a^2*b^7 + C*a*b^8
- 3*B*b^9)*tan(d*x + c)^3 + 2*(C*a^8*b - 3*B*a^7*b^2 + 3*C*a^6*b^3 - 9*B*a^
5*b^4 + 3*C*a^4*b^5 - 9*B*a^3*b^6 + C*a^2*b^7 - 3*B*a*b^8)*tan(d*x + c)^2 +
 (C*a^9 - 3*B*a^8*b + 3*C*a^7*b^2 - 9*B*a^6*b^3 + 3*C*a^5*b^4 - 9*B*a^4*b^5
 + C*a^3*b^6 - 3*B*a^2*b^7)*tan(d*x + c))*log(tan(d*x + c)^2/(tan(d*x + c)^
2 + 1)) + ((6*C*a^5*b^4 - 10*B*a^4*b^5 + 3*C*a^3*b^6 - 9*B*a^2*b^7 + C*a*b^
8 - 3*B*b^9)*tan(d*x + c)^3 + 2*(6*C*a^6*b^3 - 10*B*a^5*b^4 + 3*C*a^4*b^5 -
 9*B*a^3*b^6 + C*a^2*b^7 - 3*B*a*b^8)*tan(d*x + c)^2 + (6*C*a^7*b^2 - 10*B*
a^6*b^3 + 3*C*a^5*b^4 - 9*B*a^4*b^5 + C*a^3*b^6 - 3*B*a^2*b^7)*tan(d*x + c)
)*log((b^2*tan(d*x + c)^2 + 2*a*b*tan(d*x + c) + a^2)/(tan(d*x + c)^2 + 1))
 + (4*B*a^8*b + 12*B*a^6*b^3 - 9*C*a^5*b^4 + 23*B*a^4*b^5 - 3*C*a^3*b^6 + 9
*B*a^2*b^7 + 2*(B*a^9 + 3*C*a^8*b - 3*B*a^7*b^2 - C*a^6*b^3)*d*x)*tan(d*x +
 c))/((a^10*b^2 + 3*a^8*b^4 + 3*a^6*b^6 + a^4*b^8)*d*tan(d*x + c)^3 + 2*(a^
11*b + 3*a^9*b^3 + 3*a^7*b^5 + a^5*b^7)*d*tan(d*x + c)^2 + (a^12 + 3*a^10*b
^2 + 3*a^8*b^4 + a^6*b^6)*d*tan(d*x + c))
```

Sympy [F(-2)]

Exception generated.

$$\int \frac{\cot^3(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^3}\,dx = \text{Exception raised: AttributeError}$$

```
[In] integrate(cot(d*x+c)**3*(B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c))**3,
x)
```

```
[Out] Exception raised: AttributeError >> 'NoneType' object has no attribute 'pri
mitive'
```

Maxima [A] (verification not implemented)

none

Time = 0.34 (sec) , antiderivative size = 454, normalized size of antiderivative = 1.58

$$\int \frac{\cot^3(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^3}\,dx =$$

$$-\frac{\frac{2\,(Ba^3+3\,Ca^2b-3\,Bab^2-Cb^3)(dx+c)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}+\frac{2\,(6\,Ca^5b^2-10\,Ba^4b^3+3\,Ca^3b^4-9\,Ba^2b^5+Cab^6-3\,Bb^7)\log(b\tan(dx+c)+a)}{a^{10}+3\,a^8b^2+3\,a^6b^4+a^4b^6}+\frac{(Ca^3-3\,Ba^2b-3\,C}{a^6+}}{}$$

```
[In] integrate(cot(d*x+c)^3*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x,
algorithm="maxima")
```

```
[Out] -1/2*(2*(B*a^3 + 3*C*a^2*b - 3*B*a*b^2 - C*b^3)*(d*x + c)/(a^6 + 3*a^4*b^2
+ 3*a^2*b^4 + b^6) + 2*(6*C*a^5*b^2 - 10*B*a^4*b^3 + 3*C*a^3*b^4 - 9*B*a^2*
b^5 + C*a*b^6 - 3*B*b^7)*log(b*tan(d*x + c) + a)/(a^10 + 3*a^8*b^2 + 3*a^6*
b^4 + a^4*b^6) + (C*a^3 - 3*B*a^2*b - 3*C*a*b^2 + B*b^3)*log(tan(d*x + c)^2
 + 1)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) + (2*B*a^6 + 4*B*a^4*b^2 + 2*B*a^
2*b^4 + 2*(B*a^4*b^2 - 3*C*a^3*b^3 + 6*B*a^2*b^4 - C*a*b^5 + 3*B*b^6)*tan(d
*x + c)^2 + (4*B*a^5*b - 7*C*a^4*b^2 + 17*B*a^3*b^3 - 3*C*a^2*b^4 + 9*B*a*b
^5)*tan(d*x + c))/((a^7*b^2 + 2*a^5*b^4 + a^3*b^6)*tan(d*x + c)^3 + 2*(a^8*
b + 2*a^6*b^3 + a^4*b^5)*tan(d*x + c)^2 + (a^9 + 2*a^7*b^2 + a^5*b^4)*tan(d
*x + c)) - 2*(C*a - 3*B*b)*log(tan(d*x + c))/a^4)/d
```

Giac [A] (verification not implemented)

none

Time = 1.58 (sec) , antiderivative size = 560, normalized size of antiderivative = 1.95

$$\int \frac{\cot^3(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^3}\,dx =$$

$$-\frac{\frac{2\,(Ba^3+3\,Ca^2b-3\,Bab^2-Cb^3)(dx+c)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}+\frac{(Ca^3-3\,Ba^2b-3\,Cab^2+Bb^3)\log\left(\tan(dx+c)^2+1\right)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}+\frac{2\,(6\,Ca^5b^3-10\,Ba^4b^4+3\,Ca^3b^5-9\,Ba^2b^6+}{a^{10}b+3\,a^8b^3+3\,a}}{}$$

```
[In] integrate(cot(d*x+c)^3*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x,
algorithm="giac")
```

```
[Out] -1/2*(2*(B*a^3 + 3*C*a^2*b - 3*B*a*b^2 - C*b^3)*(d*x + c)/(a^6 + 3*a^4*b^2
+ 3*a^2*b^4 + b^6) + (C*a^3 - 3*B*a^2*b - 3*C*a*b^2 + B*b^3)*log(tan(d*x +
c)^2 + 1)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) + 2*(6*C*a^5*b^3 - 10*B*a^4*b
^4 + 3*C*a^3*b^5 - 9*B*a^2*b^6 + C*a*b^7 - 3*B*b^8)*log(abs(b*tan(d*x + c)
+ a))/(a^10*b + 3*a^8*b^3 + 3*a^6*b^5 + a^4*b^7) - (18*C*a^5*b^4*tan(d*x +
c)^2 - 30*B*a^4*b^5*tan(d*x + c)^2 + 9*C*a^3*b^6*tan(d*x + c)^2 - 27*B*a^2*
b^7*tan(d*x + c)^2 + 3*C*a*b^8*tan(d*x + c)^2 - 9*B*b^9*tan(d*x + c)^2 + 42
*C*a^6*b^3*tan(d*x + c) - 68*B*a^5*b^4*tan(d*x + c) + 26*C*a^4*b^5*tan(d*x
+ c) - 66*B*a^3*b^6*tan(d*x + c) + 8*C*a^2*b^7*tan(d*x + c) - 22*B*a*b^8*ta
n(d*x + c) + 25*C*a^7*b^2 - 39*B*a^6*b^3 + 19*C*a^5*b^4 - 41*B*a^4*b^5 + 6*
C*a^3*b^6 - 14*B*a^2*b^7)/((a^10 + 3*a^8*b^2 + 3*a^6*b^4 + a^4*b^6)*(b*tan(
d*x + c) + a)^2) - 2*(C*a - 3*B*b)*log(abs(tan(d*x + c)))/a^4 + 2*(C*a*tan(
d*x + c) - 3*B*b*tan(d*x + c) + B*a)/(a^4*tan(d*x + c)))/d
```

Mupad [B] (verification not implemented)

Time = 13.89 (sec) , antiderivative size = 380, normalized size of antiderivative = 1.32

$$\int \frac{\cot^3(c+dx)\,(B\tan(c+dx)+C\tan^2(c+dx))}{(a+b\tan(c+dx))^3}\,dx$$
$$= \frac{b^2\,\ln(a+b\tan(c+d\,x))\,(-6\,C\,a^5+10\,B\,a^4\,b-3\,C\,a^3\,b^2+9\,B\,a^2\,b^3-C\,a\,b^4+3\,B\,b^5)}{a^4\,d\,(a^2+b^2)^3}$$
$$-\frac{\ln(\tan(c+d\,x)-\mathrm{i})\,(-C+B\,1\mathrm{i})}{2\,d\,(-a^3-a^2\,b\,3\mathrm{i}+3\,a\,b^2+b^3\,1\mathrm{i})}$$
$$-\frac{\ln(\tan(c+d\,x))\,(3\,B\,b-C\,a)}{a^4\,d}-\frac{\ln(\tan(c+d\,x)+1\mathrm{i})\,(B-C\,1\mathrm{i})}{2\,d\,(-a^3\,1\mathrm{i}-3\,a^2\,b+a\,b^2\,3\mathrm{i}+b^3)}$$
$$-\frac{\frac{B}{a}+\frac{\tan(c+d\,x)^2\,(B\,a^4\,b^2-3\,C\,a^3\,b^3+6\,B\,a^2\,b^4-C\,a\,b^5+3\,B\,b^6)}{a^3\,(a^4+2\,a^2\,b^2+b^4)}+\frac{\tan(c+d\,x)\,(4\,B\,a^4\,b-7\,C\,a^3\,b^2+17\,B\,a^2\,b^3-3\,C\,a\,b^4+9\,B\,b^5)}{2\,a^2\,(a^4+2\,a^2\,b^2+b^4)}}{d\,\left(a^2\tan(c+d\,x)+2\,a\,b\tan(c+d\,x)^2+b^2\tan(c+d\,x)^3\right)}$$

```
[In] int((cot(c + d*x)^3*(B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c + d*x
))^3,x)
```

```
[Out] (b^2*log(a + b*tan(c + d*x))*(3*B*b^5 - 6*C*a^5 + 9*B*a^2*b^3 - 3*C*a^3*b^2
 + 10*B*a^4*b - C*a*b^4))/(a^4*d*(a^2 + b^2)^3) - (log(tan(c + d*x) - 1i)*(
B*1i - C))/(2*d*(3*a*b^2 - a^2*b*3i - a^3 + b^3*1i)) - (log(tan(c + d*x))*(
3*B*b - C*a))/(a^4*d) - (log(tan(c + d*x) + 1i)*(B - C*1i))/(2*d*(a*b^2*3i
- 3*a^2*b - a^3*1i + b^3)) - (B/a + (tan(c + d*x)^2*(3*B*b^6 + 6*B*a^2*b^4
+ B*a^4*b^2 - 3*C*a^3*b^3 - C*a*b^5))/(a^3*(a^4 + b^4 + 2*a^2*b^2)) + (tan(
c + d*x)*(9*B*b^5 + 17*B*a^2*b^3 - 7*C*a^3*b^2 + 4*B*a^4*b - 3*C*a*b^4))/(2
*a^2*(a^4 + b^4 + 2*a^2*b^2)))/(d*(a^2*tan(c + d*x) + b^2*tan(c + d*x)^3 +
2*a*b*tan(c + d*x)^2))
```

3.45 $\int \tan^2(c+dx)(b\tan(c+dx))^n (A+B\tan(c+dx)+C$

Optimal result 381
Rubi [A] (verified) 381
Mathematica [A] (verified) 383
Maple [F] 383
Fricas [F] 384
Sympy [F] 384
Maxima [F] 384
Giac [F] 385
Mupad [F(-1)] 385

Optimal result

Integrand size = 39, antiderivative size = 132

$$\begin{aligned}\int \tan^2(c+dx)(b\tan(c+dx))^n \left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx &= \frac{C(b\tan(c+dx))^{3+n}}{b^3d(3+n)} \\ &+ \frac{(A-C)\,\text{Hypergeometric2F1}\left(1,\frac{3+n}{2},\frac{5+n}{2},-\tan^2(c+dx)\right)(b\tan(c+dx))^{3+n}}{b^3d(3+n)} \\ &+ \frac{B\,\text{Hypergeometric2F1}\left(1,\frac{4+n}{2},\frac{6+n}{2},-\tan^2(c+dx)\right)(b\tan(c+dx))^{4+n}}{b^4d(4+n)}\end{aligned}$$

```
[Out] C*(b*tan(d*x+c))^(3+n)/b^3/d/(3+n)+(A-C)*hypergeom([1, 3/2+1/2*n],[5/2+1/2*
n],-tan(d*x+c)^2)*(b*tan(d*x+c))^(3+n)/b^3/d/(3+n)+B*hypergeom([1, 2+1/2*n]
,[3+1/2*n],-tan(d*x+c)^2)*(b*tan(d*x+c))^(4+n)/b^4/d/(4+n)
```

Rubi [A] (verified)

Time = 0.21 (sec) , antiderivative size = 132, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.128, Rules used = {16, 3711, 3619, 3557, 371}

$$\begin{aligned}\int \tan^2(c+dx)(b\tan(c+dx))^n \left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx &= \frac{(A-C)(b\tan(c+dx))^{n+3}\,\text{Hypergeometric2F1}\left(1,\frac{n+3}{2},\frac{n+5}{2},-\tan^2(c+dx)\right)}{b^3d(n+3)} \\ &+ \frac{B(b\tan(c+dx))^{n+4}\,\text{Hypergeometric2F1}\left(1,\frac{n+4}{2},\frac{n+6}{2},-\tan^2(c+dx)\right)}{b^4d(n+4)} \\ &+ \frac{C(b\tan(c+dx))^{n+3}}{b^3d(n+3)}\end{aligned}$$

```
[In] Int[Tan[c + d*x]^2*(b*Tan[c + d*x])^n*(A + B*Tan[c + d*x] + C*Tan[c + d*x]^
2),x]
```

```
[Out] (C*(b*Tan[c + d*x])^(3 + n))/(b^3*d*(3 + n)) + ((A - C)*Hypergeometric2F1[1
, (3 + n)/2, (5 + n)/2, -Tan[c + d*x]^2]*(b*Tan[c + d*x])^(3 + n))/(b^3*d*(
3 + n)) + (B*Hypergeometric2F1[1, (4 + n)/2, (6 + n)/2, -Tan[c + d*x]^2]*(b
*Tan[c + d*x])^(4 + n))/(b^4*d*(4 + n))
```

Rule 16

```
Int[(u_.)*(v_)^(m_.)*((b_)*(v_))^(n_), x_Symbol] :> Dist[1/b^m, Int[u*(b*v)
^(m + n), x], x] /; FreeQ[{b, n}, x] && IntegerQ[m]
```

Rule 371

```
Int[((c_.)*(x_))^(m_.)*((a_) + (b_.)*(x_)^(n_))^(p_), x_Symbol] :> Simp[a^p
*((c*x)^(m + 1)/(c*(m + 1)))*Hypergeometric2F1[-p, (m + 1)/n, (m + 1)/n + 1
, (-b)*(x^n/a)], x] /; FreeQ[{a, b, c, m, n, p}, x] &&  !IGtQ[p, 0] && (ILt
Q[p, 0] || GtQ[a, 0])
```

Rule 3557

```
Int[((b_.)*tan[(c_.) + (d_.)*(x_)])^(n_), x_Symbol] :> Dist[b/d, Subst[Int[
x^n/(b^2 + x^2), x], x, b*Tan[c + d*x]], x] /; FreeQ[{b, c, d, n}, x] &&  !
IntegerQ[n]
```

Rule 3619

```
Int[((b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) + (f_.)*(x
_)]), x_Symbol] :> Dist[c, Int[(b*Tan[e + f*x])^m, x], x] + Dist[d/b, Int[(
b*Tan[e + f*x])^(m + 1), x], x] /; FreeQ[{b, c, d, e, f, m}, x] && NeQ[c^2
+ d^2, 0] &&  !IntegerQ[2*m]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rubi steps

$$\begin{aligned}\text{integral} &= \frac{\int (b\tan(c+dx))^{2+n}\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx}{b^2}\\ &= \frac{C(b\tan(c+dx))^{3+n}}{b^3d(3+n)}+\frac{\int (b\tan(c+dx))^{2+n}(A-C+B\tan(c+dx))\,dx}{b^2}\end{aligned}$$

$$= \frac{C(b\tan(c+dx))^{3+n}}{b^3d(3+n)} + \frac{B\int (b\tan(c+dx))^{3+n}\,dx}{b^3} + \frac{(A-C)\int (b\tan(c+dx))^{2+n}\,dx}{b^2}$$

$$= \frac{C(b\tan(c+dx))^{3+n}}{b^3d(3+n)} + \frac{B\text{Subst}\left(\int \frac{x^{3+n}}{b^2+x^2}\,dx, x, b\tan(c+dx)\right)}{b^2d} + \frac{(A-C)\text{Subst}\left(\int \frac{x^{2+n}}{b^2+x^2}\,dx, x, b\tan(c+dx)\right)}{bd}$$

$$= \frac{C(b\tan(c+dx))^{3+n}}{b^3d(3+n)} + \frac{(A-C)\,\text{Hypergeometric2F1}\left(1, \frac{3+n}{2}, \frac{5+n}{2}, -\tan^2(c+dx)\right)(b\tan(c+dx))^{3+n}}{b^3d(3+n)} + \frac{B\,\text{Hypergeometric2F1}\left(1, \frac{4+n}{2}, \frac{6+n}{2}, -\tan^2(c+dx)\right)(b\tan(c+dx))^{4+n}}{b^4d(4+n)}$$

Mathematica [A] (verified)

Time = 0.63 (sec) , antiderivative size = 110, normalized size of antiderivative = 0.83

$$\int \tan^2(c+dx)(b\tan(c+dx))^n \left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$= \frac{\tan^3(c+dx)(b\tan(c+dx))^n \left(C(4+n)+(A-C)(4+n)\,\text{Hypergeometric2F1}\left(1, \frac{3+n}{2}, \frac{5+n}{2}, -\tan^2(c+d\right.\right.}{d(3+n)(4+n)}$$

```
[In] Integrate[Tan[c + d*x]^2*(b*Tan[c + d*x])^n*(A + B*Tan[c + d*x] + C*Tan[c +
 d*x]^2),x]
```

```
[Out] (Tan[c + d*x]^3*(b*Tan[c + d*x])^n*(C*(4 + n) + (A - C)*(4 + n)*Hypergeomet
ric2F1[1, (3 + n)/2, (5 + n)/2, -Tan[c + d*x]^2] + B*(3 + n)*Hypergeometric
2F1[1, (4 + n)/2, (6 + n)/2, -Tan[c + d*x]^2]*Tan[c + d*x]))/(d*(3 + n)*(4
+ n))
```

Maple [F]

$$\int \tan(dx+c)^2 (b\tan(dx+c))^n \left(A+B\tan(dx+c)+C\tan(dx+c)^2\right) dx$$

```
[In] int(tan(d*x+c)^2*(b*tan(d*x+c))^n*(A+B*tan(d*x+c)+C*tan(d*x+c)^2),x)
```

```
[Out] int(tan(d*x+c)^2*(b*tan(d*x+c))^n*(A+B*tan(d*x+c)+C*tan(d*x+c)^2),x)
```

Fricas [F]

$$\int \tan^2(c+dx)(b\tan(c+dx))^n \left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\int \left(C\tan\left(dx+c\right)^2+B\tan\left(dx+c\right)+A\right)\left(b\tan\left(dx+c\right)\right)^n\tan\left(dx+c\right)^2\,dx$$

```
[In] integrate(tan(d*x+c)^2*(b*tan(d*x+c))^n*(A+B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="fricas")
```

```
[Out] integral((C*tan(d*x + c)^4 + B*tan(d*x + c)^3 + A*tan(d*x + c)^2)*(b*tan(d*
x + c))^n, x)
```

Sympy [F]

$$\int \tan^2(c+dx)(b\tan(c+dx))^n \left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\int \left(b\tan\left(c+dx\right)\right)^n\left(A+B\tan\left(c+dx\right)+C\tan^2\left(c+dx\right)\right)\tan^2\left(c+dx\right)dx$$

```
[In] integrate(tan(d*x+c)**2*(b*tan(d*x+c))**n*(A+B*tan(d*x+c)+C*tan(d*x+c)**2),
x)
```

```
[Out] Integral((b*tan(c + d*x))**n*(A + B*tan(c + d*x) + C*tan(c + d*x)**2)*tan(c
 + d*x)**2, x)
```

Maxima [F]

$$\int \tan^2(c+dx)(b\tan(c+dx))^n \left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\int \left(C\tan\left(dx+c\right)^2+B\tan\left(dx+c\right)+A\right)\left(b\tan\left(dx+c\right)\right)^n\tan\left(dx+c\right)^2\,dx$$

```
[In] integrate(tan(d*x+c)^2*(b*tan(d*x+c))^n*(A+B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="maxima")
```

```
[Out] integrate((C*tan(d*x + c)^2 + B*tan(d*x + c) + A)*(b*tan(d*x + c))^n*tan(d*
x + c)^2, x)
```

Giac [F]

$$\int \tan^2(c+dx)(b\tan(c+dx))^n \left(A+B\tan(c+dx)+C\tan^2(c+dx)\right) dx$$
$$= \int \left(C\tan\left(dx+c\right)^2+B\tan\left(dx+c\right)+A\right)\left(b\tan\left(dx+c\right)\right)^n\tan\left(dx+c\right)^2 dx$$

```
[In] integrate(tan(d*x+c)^2*(b*tan(d*x+c))^n*(A+B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="giac")
[Out] integrate((C*tan(d*x + c)^2 + B*tan(d*x + c) + A)*(b*tan(d*x + c))^n*tan(d*
x + c)^2, x)
```

Mupad [F(-1)]

Timed out.

$$\int \tan^2(c+dx)(b\tan(c+dx))^n \left(A+B\tan(c+dx)+C\tan^2(c+dx)\right) dx$$
$$= \int \tan(c+d\,x)^2\,(b\tan(c+d\,x))^n\,\left(C\tan(c+d\,x)^2+B\tan(c+d\,x)+A\right) dx$$

```
[In] int(tan(c + d*x)^2*(b*tan(c + d*x))^n*(A + B*tan(c + d*x) + C*tan(c + d*x)^
2),x)
[Out] int(tan(c + d*x)^2*(b*tan(c + d*x))^n*(A + B*tan(c + d*x) + C*tan(c + d*x)^
2), x)
```

3.46 $\int \tan^m(c+dx)(b\tan(c+dx))^n\left(A+B\tan(c+dx)+C\right.$

Optimal result 386
Rubi [A] (verified) 386
Mathematica [A] (verified) 389
Maple [F] 389
Fricas [F] 389
Sympy [F] 390
Maxima [F] 390
Giac [F] 390
Mupad [F(-1)] 391

Optimal result

Integrand size = 39, antiderivative size = 154

$$\int \tan^m(c+dx)(b\tan(c+dx))^n\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{C\tan^{1+m}(c+dx)(b\tan(c+dx))^n}{d(1+m+n)}$$
$$+\frac{(A-C)\,\text{Hypergeometric2F1}\left(1,\frac{1}{2}(1+m+n),\frac{1}{2}(3+m+n),-\tan^2(c+dx)\right)\tan^{1+m}(c+dx)(b\tan(c+}{d(1+m+n)}$$
$$+\frac{B\,\text{Hypergeometric2F1}\left(1,\frac{1}{2}(2+m+n),\frac{1}{2}(4+m+n),-\tan^2(c+dx)\right)\tan^{2+m}(c+dx)(b\tan(c+dx))^n}{d(2+m+n)}$$

```
[Out] C*tan(d*x+c)^(1+m)*(b*tan(d*x+c))^n/d/(1+m+n)+(A-C)*hypergeom([1, 1/2+1/2*m
+1/2*n],[3/2+1/2*m+1/2*n],-tan(d*x+c)^2)*tan(d*x+c)^(1+m)*(b*tan(d*x+c))^n/
d/(1+m+n)+B*hypergeom([1, 1+1/2*m+1/2*n],[2+1/2*m+1/2*n],-tan(d*x+c)^2)*tan
(d*x+c)^(2+m)*(b*tan(d*x+c))^n/d/(2+m+n)
```

Rubi [A] (verified)

Time = 0.17 (sec) , antiderivative size = 154, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.128, Rules used

$= \{20, 3711, 3619, 3557, 371\}$

$$\int \tan^m(c+dx)(b\tan(c+dx))^n \left(A+B\tan(c+dx)+C\tan^2(c+dx)\right) dx$$
$$= \frac{(A-C)\tan^{m+1}(c+dx)(b\tan(c+dx))^n \text{Hypergeometric2F1}\left(1, \frac{1}{2}(m+n+1), \frac{1}{2}(m+n+3), -\tan^2(c+\right.}{d(m+n+1)}$$
$$+ \frac{B\tan^{m+2}(c+dx)(b\tan(c+dx))^n \text{Hypergeometric2F1}\left(1, \frac{1}{2}(m+n+2), \frac{1}{2}(m+n+4), -\tan^2(c+dx)\right.}{d(m+n+2)}$$
$$+ \frac{C\tan^{m+1}(c+dx)(b\tan(c+dx))^n}{d(m+n+1)}$$

```
[In] Int[Tan[c + d*x]^m*(b*Tan[c + d*x])^n*(A + B*Tan[c + d*x] + C*Tan[c + d*x]^
2),x]
```

```
[Out] (C*Tan[c + d*x]^(1 + m)*(b*Tan[c + d*x])^n)/(d*(1 + m + n)) + ((A - C)*Hype
rgeometric2F1[1, (1 + m + n)/2, (3 + m + n)/2, -Tan[c + d*x]^2]*Tan[c + d*x
]^(1 + m)*(b*Tan[c + d*x])^n)/(d*(1 + m + n)) + (B*Hypergeometric2F1[1, (2
+ m + n)/2, (4 + m + n)/2, -Tan[c + d*x]^2]*Tan[c + d*x]^(2 + m)*(b*Tan[c +
 d*x])^n)/(d*(2 + m + n))
```

Rule 20

```
Int[(u_.)*((a_.)*(v_))^(m_)*((b_.)*(v_))^(n_), x_Symbol] :> Dist[b^IntPart[
n]*((b*v)^FracPart[n]/(a^IntPart[n]*(a*v)^FracPart[n])), Int[u*(a*v)^(m + n
), x], x] /; FreeQ[{a, b, m, n}, x] &&  !IntegerQ[m] &&  !IntegerQ[n] &&  !
IntegerQ[m + n]
```

Rule 371

```
Int[((c_.)*(x_))^(m_.)*((a_) + (b_.)*(x_)^(n_))^(p_), x_Symbol] :> Simp[a^p
*((c*x)^(m + 1)/(c*(m + 1)))*Hypergeometric2F1[-p, (m + 1)/n, (m + 1)/n + 1
, (-b)*(x^n/a)], x] /; FreeQ[{a, b, c, m, n, p}, x] &&  !IGtQ[p, 0] && (ILt
Q[p, 0] || GtQ[a, 0])
```

Rule 3557

```
Int[((b_.)*tan[(c_.) + (d_.)*(x_)])^(n_), x_Symbol] :> Dist[b/d, Subst[Int[
x^n/(b^2 + x^2), x], x, b*Tan[c + d*x]], x] /; FreeQ[{b, c, d, n}, x] &&  !
IntegerQ[n]
```

Rule 3619

```
Int[((b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) + (f_.)*(x
_)]), x_Symbol] :> Dist[c, Int[(b*Tan[e + f*x])^m, x], x] + Dist[d/b, Int[(
b*Tan[e + f*x])^(m + 1), x], x] /; FreeQ[{b, c, d, e, f, m}, x] && NeQ[c^2
+ d^2, 0] &&  !IntegerQ[2*m]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rubi steps

$$\text{integral} = \left(\tan^{-n}(c+dx)(b\tan(c+dx))^n\right)\int \tan^{m+n}(c+dx)\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$

$$=\frac{C\tan^{1+m}(c+dx)(b\tan(c+dx))^n}{d(1+m+n)}+\left(\tan^{-n}(c+dx)(b\tan(c+dx))^n\right)\int \tan^{m+n}(c+dx)(A-C+B\tan(c+dx))\,dx$$

$$=\frac{C\tan^{1+m}(c+dx)(b\tan(c+dx))^n}{d(1+m+n)}+\left(B\tan^{-n}(c+dx)(b\tan(c+dx))^n\right)\int \tan^{1+m+n}(c+dx)\,dx+\left((A-C)\tan^{-n}(c+dx)(b\tan(c+dx))^n\right)\int \tan^{m+n}(c+dx)\,dx$$

$$=\frac{C\tan^{1+m}(c+dx)(b\tan(c+dx))^n}{d(1+m+n)}+\frac{(B\tan^{-n}(c+dx)(b\tan(c+dx))^n)\operatorname{Subst}\left(\int\frac{x^{1+m+n}}{1+x^2}\,dx,x,\tan(c+dx)\right)}{d}+\frac{((A-C)\tan^{-n}(c+dx)(b\tan(c+dx))^n)\operatorname{Subst}\left(\int\frac{x^{m+n}}{1+x^2}\,dx,x,\tan(c+dx)\right)}{d}$$

$$=\frac{C\tan^{1+m}(c+dx)(b\tan(c+dx))^n}{d(1+m+n)}+\frac{(A-C)\,\text{Hypergeometric2F1}\left(1,\frac{1}{2}(1+m+n),\frac{1}{2}(3+m+n),-\tan^2(c+dx)\right)\tan^{1+m}(c+dx)(b}{d(1+m+n)}+\frac{B\,\text{Hypergeometric2F1}\left(1,\frac{1}{2}(2+m+n),\frac{1}{2}(4+m+n),-\tan^2(c+dx)\right)\tan^{2+m}(c+dx)(b\tan(c}{d(2+m+n)}$$

Mathematica [A] (verified)

Time = 0.42 (sec) , antiderivative size = 115, normalized size of antiderivative = 0.75

$$\int \tan^m(c+dx)(b\tan(c+dx))^n\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{\tan^{1+m}(c+dx)(b\tan(c+dx))^n\left(\frac{C}{1+m+n}+\frac{(A-C)\,\text{Hypergeometric2F1}\left(1,\frac{1}{2}(1+m+n),\frac{1}{2}(3+m+n),-\tan^2(c+dx)\right)}{1+m+n}+\frac{B\,\text{Hyperg}}{}\right.}{d}$$

```
[In] Integrate[Tan[c + d*x]^m*(b*Tan[c + d*x])^n*(A + B*Tan[c + d*x] + C*Tan[c +
 d*x]^2),x]
```

```
[Out] (Tan[c + d*x]^(1 + m)*(b*Tan[c + d*x])^n*(C/(1 + m + n) + ((A - C)*Hypergeo
metric2F1[1, (1 + m + n)/2, (3 + m + n)/2, -Tan[c + d*x]^2])/(1 + m + n) +
(B*Hypergeometric2F1[1, (2 + m + n)/2, (4 + m + n)/2, -Tan[c + d*x]^2]*Tan[
c + d*x])/(2 + m + n)))/d
```

Maple [F]

$$\int \tan(dx+c)^m\,(b\tan(dx+c))^n\left(A+B\tan(dx+c)+C\tan(dx+c)^2\right)dx$$

```
[In] int(tan(d*x+c)^m*(b*tan(d*x+c))^n*(A+B*tan(d*x+c)+C*tan(d*x+c)^2),x)
```

```
[Out] int(tan(d*x+c)^m*(b*tan(d*x+c))^n*(A+B*tan(d*x+c)+C*tan(d*x+c)^2),x)
```

Fricas [F]

$$\int \tan^m(c+dx)(b\tan(c+dx))^n\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\int \left(C\tan(dx+c)^2+B\tan(dx+c)+A\right)(b\tan(dx+c))^n\tan(dx+c)^m\,dx$$

```
[In] integrate(tan(d*x+c)^m*(b*tan(d*x+c))^n*(A+B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="fricas")
```

```
[Out] integral((C*tan(d*x + c)^2 + B*tan(d*x + c) + A)*(b*tan(d*x + c))^n*tan(d*x
 + c)^m, x)
```

Sympy [F]

$$\int \tan^m(c+dx)(b\tan(c+dx))^n \left(A+B\tan(c+dx)+C\tan^2(c+dx)\right) dx$$
$$= \int (b\tan(c+dx))^n \left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)\tan^m(c+dx)\,dx$$

```
[In] integrate(tan(d*x+c)**m*(b*tan(d*x+c))**n*(A+B*tan(d*x+c)+C*tan(d*x+c)**2),
x)
```

```
[Out] Integral((b*tan(c + d*x))**n*(A + B*tan(c + d*x) + C*tan(c + d*x)**2)*tan(c
 + d*x)**m, x)
```

Maxima [F]

$$\int \tan^m(c+dx)(b\tan(c+dx))^n \left(A+B\tan(c+dx)+C\tan^2(c+dx)\right) dx$$
$$= \int \left(C\tan(dx+c)^2+B\tan(dx+c)+A\right)(b\tan(dx+c))^n\tan(dx+c)^m\,dx$$

```
[In] integrate(tan(d*x+c)^m*(b*tan(d*x+c))^n*(A+B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="maxima")
```

```
[Out] integrate((C*tan(d*x + c)^2 + B*tan(d*x + c) + A)*(b*tan(d*x + c))^n*tan(d*
x + c)^m, x)
```

Giac [F]

$$\int \tan^m(c+dx)(b\tan(c+dx))^n \left(A+B\tan(c+dx)+C\tan^2(c+dx)\right) dx$$
$$= \int \left(C\tan(dx+c)^2+B\tan(dx+c)+A\right)(b\tan(dx+c))^n\tan(dx+c)^m\,dx$$

```
[In] integrate(tan(d*x+c)^m*(b*tan(d*x+c))^n*(A+B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="giac")
```

```
[Out] integrate((C*tan(d*x + c)^2 + B*tan(d*x + c) + A)*(b*tan(d*x + c))^n*tan(d*
x + c)^m, x)
```

Mupad [F(-1)]

Timed out.

$$\int \tan^m(c+dx)(b\tan(c+dx))^n \left(A+B\tan(c+dx)+C\tan^2(c+dx)\right) dx$$
$$= \int \tan(c+d\,x)^m\,(b\tan(c+d\,x))^n\,\left(C\tan(c+d\,x)^2+B\tan(c+d\,x)+A\right) dx$$

```
[In] int(tan(c + d*x)^m*(b*tan(c + d*x))^n*(A + B*tan(c + d*x) + C*tan(c + d*x)^
2),x)
[Out] int(tan(c + d*x)^m*(b*tan(c + d*x))^n*(A + B*tan(c + d*x) + C*tan(c + d*x)^
2), x)
```

3.47 $\int \tan^m(c+dx)\sqrt{b\tan(c+dx)}(A+B\tan(c+dx)+C$

Optimal result 392
Rubi [A] (verified) 392
Mathematica [A] (verified) 395
Maple [F] 395
Fricas [F] 395
Sympy [F] 396
Maxima [F(-1)] 396
Giac [F] 396
Mupad [F(-1)] 397

Optimal result

Integrand size = 41, antiderivative size = 170

$$\int \tan^m(c+dx)\sqrt{b\tan(c+dx)}\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{2C\tan^{1+m}(c+dx)\sqrt{b\tan(c+dx)}}{d(3+2m)}$$
$$+\frac{2(A-C)\,\text{Hypergeometric2F1}\left(1,\frac{1}{4}(3+2m),\frac{1}{4}(7+2m),-\tan^2(c+dx)\right)\tan^{1+m}(c+dx)\sqrt{b\tan(c+dx}}{d(3+2m)}$$
$$+\frac{2B\,\text{Hypergeometric2F1}\left(1,\frac{1}{4}(5+2m),\frac{1}{4}(9+2m),-\tan^2(c+dx)\right)\tan^{2+m}(c+dx)\sqrt{b\tan(c+dx)}}{d(5+2m)}$$

```
[Out] 2*C*(b*tan(d*x+c))^(1/2)*tan(d*x+c)^(1+m)/d/(3+2*m)+2*(A-C)*hypergeom([1, 3
/4+1/2*m],[7/4+1/2*m],-tan(d*x+c)^2)*(b*tan(d*x+c))^(1/2)*tan(d*x+c)^(1+m)/
d/(3+2*m)+2*B*hypergeom([1, 5/4+1/2*m],[9/4+1/2*m],-tan(d*x+c)^2)*(b*tan(d*
x+c))^(1/2)*tan(d*x+c)^(2+m)/d/(5+2*m)
```

Rubi [A] (verified)

Time = 0.18 (sec) , antiderivative size = 170, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}} = 0.122$, Rules used

$= \{20, 3711, 3619, 3557, 371\}$

$$\int \tan^m(c+dx)\sqrt{b\tan(c+dx)}\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{2(A-C)\sqrt{b\tan(c+dx)}\tan^{m+1}(c+dx)\,\text{Hypergeometric2F1}\left(1,\frac{1}{4}(2m+3),\frac{1}{4}(2m+7),-\tan^2(c+dx)\right)}{d(2m+3)}$$
$$+\frac{2B\sqrt{b\tan(c+dx)}\tan^{m+2}(c+dx)\,\text{Hypergeometric2F1}\left(1,\frac{1}{4}(2m+5),\frac{1}{4}(2m+9),-\tan^2(c+dx)\right)}{d(2m+5)}$$
$$+\frac{2C\sqrt{b\tan(c+dx)}\tan^{m+1}(c+dx)}{d(2m+3)}$$

```
[In] Int[Tan[c + d*x]^m*Sqrt[b*Tan[c + d*x]]*(A + B*Tan[c + d*x] + C*Tan[c + d*x
]^2),x]
```

```
[Out] (2*C*Tan[c + d*x]^(1 + m)*Sqrt[b*Tan[c + d*x]])/(d*(3 + 2*m)) + (2*(A - C)*
Hypergeometric2F1[1, (3 + 2*m)/4, (7 + 2*m)/4, -Tan[c + d*x]^2]*Tan[c + d*x
]^(1 + m)*Sqrt[b*Tan[c + d*x]])/(d*(3 + 2*m)) + (2*B*Hypergeometric2F1[1, (
5 + 2*m)/4, (9 + 2*m)/4, -Tan[c + d*x]^2]*Tan[c + d*x]^(2 + m)*Sqrt[b*Tan[c
 + d*x]])/(d*(5 + 2*m))
```

Rule 20

```
Int[(u_.)*((a_.)*(v_))^(m_)*((b_.)*(v_))^(n_), x_Symbol] :> Dist[b^IntPart[
n]*((b*v)^FracPart[n]/(a^IntPart[n]*(a*v)^FracPart[n])), Int[u*(a*v)^(m + n
), x], x] /; FreeQ[{a, b, m, n}, x] &&  !IntegerQ[m] &&  !IntegerQ[n] &&  !
IntegerQ[m + n]
```

Rule 371

```
Int[((c_.)*(x_))^(m_.)*((a_) + (b_.)*(x_)^(n_))^(p_), x_Symbol] :> Simp[a^p
*((c*x)^(m + 1)/(c*(m + 1)))*Hypergeometric2F1[-p, (m + 1)/n, (m + 1)/n + 1
, (-b)*(x^n/a)], x] /; FreeQ[{a, b, c, m, n, p}, x] &&  !IGtQ[p, 0] && (ILt
Q[p, 0] || GtQ[a, 0])
```

Rule 3557

```
Int[((b_.)*tan[(c_.) + (d_.)*(x_)])^(n_), x_Symbol] :> Dist[b/d, Subst[Int[
x^n/(b^2 + x^2), x], x, b*Tan[c + d*x]], x] /; FreeQ[{b, c, d, n}, x] &&  !
IntegerQ[n]
```

Rule 3619

```
Int[((b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) + (f_.)*(x
_)]), x_Symbol] :> Dist[c, Int[(b*Tan[e + f*x])^m, x], x] + Dist[d/b, Int[(
b*Tan[e + f*x])^(m + 1), x], x] /; FreeQ[{b, c, d, e, f, m}, x] && NeQ[c^2
+ d^2, 0] &&  !IntegerQ[2*m]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rubi steps

$$\text{integral} = \frac{\sqrt{b\tan(c+dx)}\int \tan^{\frac{1}{2}+m}(c+dx)\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx}{\sqrt{\tan(c+dx)}}$$

$$= \frac{2C\tan^{1+m}(c+dx)\sqrt{b\tan(c+dx)}}{d(3+2m)} + \frac{\sqrt{b\tan(c+dx)}\int \tan^{\frac{1}{2}+m}(c+dx)(A-C+B\tan(c+dx))\,dx}{\sqrt{\tan(c+dx)}}$$

$$= \frac{2C\tan^{1+m}(c+dx)\sqrt{b\tan(c+dx)}}{d(3+2m)} + \frac{\left(B\sqrt{b\tan(c+dx)}\right)\int \tan^{\frac{3}{2}+m}(c+dx)\,dx}{\sqrt{\tan(c+dx)}} + \frac{\left((A-C)\sqrt{b\tan(c+dx)}\right)\int \tan^{\frac{1}{2}+m}(c+dx)\,dx}{\sqrt{\tan(c+dx)}}$$

$$= \frac{2C\tan^{1+m}(c+dx)\sqrt{b\tan(c+dx)}}{d(3+2m)} + \frac{\left(B\sqrt{b\tan(c+dx)}\right)\text{Subst}\left(\int \frac{x^{\frac{3}{2}+m}}{1+x^2}\,dx, x, \tan(c+dx)\right)}{d\sqrt{\tan(c+dx)}} + \frac{\left((A-C)\sqrt{b\tan(c+dx)}\right)\text{Subst}\left(\int \frac{x^{\frac{1}{2}+m}}{1+x^2}\,dx, x, \tan(c+dx)\right)}{d\sqrt{\tan(c+dx)}}$$

$$= \frac{2C\tan^{1+m}(c+dx)\sqrt{b\tan(c+dx)}}{d(3+2m)} + \frac{2(A-C)\,\text{Hypergeometric2F1}\left(1,\frac{1}{4}(3+2m),\frac{1}{4}(7+2m),-\tan^2(c+dx)\right)\tan^{1+m}(c+dx)\sqrt{b\tan(c+dx)}}{d(3+2m)} + \frac{2B\,\text{Hypergeometric2F1}\left(1,\frac{1}{4}(5+2m),\frac{1}{4}(9+2m),-\tan^2(c+dx)\right)\tan^{2+m}(c+dx)\sqrt{b\tan(c+dx)}}{d(5+2m)}$$

Mathematica [A] (verified)

Time = 0.55 (sec) , antiderivative size = 133, normalized size of antiderivative = 0.78

$$\int \tan^m(c+dx)\sqrt{b\tan(c+dx)}\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\frac{2\tan^{1+m}(c+dx)\sqrt{b\tan(c+dx)}\left(C(5+2m)+(A-C)(5+2m)\,\text{Hypergeometric2F1}\left(1,\frac{1}{4}(3+2m),\frac{1}{4}(7\right.\right.}{d(3+}$$

```
[In] Integrate[Tan[c + d*x]^m*Sqrt[b*Tan[c + d*x]]*(A + B*Tan[c + d*x] + C*Tan[c
 + d*x]^2),x]
```

```
[Out] (2*Tan[c + d*x]^(1 + m)*Sqrt[b*Tan[c + d*x]]*(C*(5 + 2*m) + (A - C)*(5 + 2*
m)*Hypergeometric2F1[1, (3 + 2*m)/4, (7 + 2*m)/4, -Tan[c + d*x]^2] + B*(3 +
 2*m)*Hypergeometric2F1[1, (5 + 2*m)/4, (9 + 2*m)/4, -Tan[c + d*x]^2]*Tan[c
 + d*x]))/(d*(3 + 2*m)*(5 + 2*m))
```

Maple [F]

$$\int \tan\left(dx+c\right)^{m}\sqrt{b\tan\left(dx+c\right)}\left(A+B\tan\left(dx+c\right)+C\tan\left(dx+c\right)^{2}\right)dx$$

```
[In] int(tan(d*x+c)^m*(b*tan(d*x+c))^(1/2)*(A+B*tan(d*x+c)+C*tan(d*x+c)^2),x)
```

```
[Out] int(tan(d*x+c)^m*(b*tan(d*x+c))^(1/2)*(A+B*tan(d*x+c)+C*tan(d*x+c)^2),x)
```

Fricas [F]

$$\int \tan^m(c+dx)\sqrt{b\tan(c+dx)}\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\int \left(C\tan\left(dx+c\right)^{2}+B\tan\left(dx+c\right)+A\right)\sqrt{b\tan\left(dx+c\right)}\tan\left(dx+c\right)^{m}\,dx$$

```
[In] integrate(tan(d*x+c)^m*(b*tan(d*x+c))^(1/2)*(A+B*tan(d*x+c)+C*tan(d*x+c)^2)
,x, algorithm="fricas")
```

```
[Out] integral((C*tan(d*x + c)^2 + B*tan(d*x + c) + A)*sqrt(b*tan(d*x + c))*tan(d
*x + c)^m, x)
```

Sympy [F]

$$\int \tan^m(c+dx)\sqrt{b\tan(c+dx)}\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\int \sqrt{b\tan\left(c+dx\right)}\left(A+B\tan\left(c+dx\right)+C\tan^2\left(c+dx\right)\right)\tan^m\left(c+dx\right)dx$$

```
[In] integrate(tan(d*x+c)**m*(b*tan(d*x+c))**(1/2)*(A+B*tan(d*x+c)+C*tan(d*x+c)*
*2),x)
```

```
[Out] Integral(sqrt(b*tan(c + d*x))*(A + B*tan(c + d*x) + C*tan(c + d*x)**2)*tan(
c + d*x)**m, x)
```

Maxima [F(-1)]

Timed out.

$$\int \tan^m(c+dx)\sqrt{b\tan(c+dx)}\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx=\text{Timed out}$$

```
[In] integrate(tan(d*x+c)^m*(b*tan(d*x+c))^(1/2)*(A+B*tan(d*x+c)+C*tan(d*x+c)^2)
,x, algorithm="maxima")
```

```
[Out] Timed out
```

Giac [F]

$$\int \tan^m(c+dx)\sqrt{b\tan(c+dx)}\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\int \left(C\tan\left(dx+c\right)^2+B\tan\left(dx+c\right)+A\right)\sqrt{b\tan\left(dx+c\right)}\tan\left(dx+c\right)^m\,dx$$

```
[In] integrate(tan(d*x+c)^m*(b*tan(d*x+c))^(1/2)*(A+B*tan(d*x+c)+C*tan(d*x+c)^2)
,x, algorithm="giac")
```

```
[Out] integrate((C*tan(d*x + c)^2 + B*tan(d*x + c) + A)*sqrt(b*tan(d*x + c))*tan(
d*x + c)^m, x)
```

Mupad [F(-1)]

Timed out.

$$\int \tan^m(c+dx)\sqrt{b\tan(c+dx)}\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx$$
$$=\int \tan(c+d\,x)^m\,\sqrt{b\tan\left(c+d\,x\right)}\,\left(C\tan(c+d\,x)^2+B\tan(c+d\,x)+A\right)\,dx$$

```
[In] int(tan(c + d*x)^m*(b*tan(c + d*x))^(1/2)*(A + B*tan(c + d*x) + C*tan(c + d
*x)^2),x)
[Out] int(tan(c + d*x)^m*(b*tan(c + d*x))^(1/2)*(A + B*tan(c + d*x) + C*tan(c + d
*x)^2), x)
```

3.48 $\int \frac{\tan^m(c+dx)\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)}{\sqrt{b\tan(c+dx)}}\,dx$

Optimal result 398
Rubi [A] (verified) 399
Mathematica [A] (verified) 401
Maple [F] 401
Fricas [F] 401
Sympy [F] 402
Maxima [F(-1)] 402
Giac [F(-1)] 402
Mupad [F(-1)] 403

Optimal result

Integrand size = 41, antiderivative size = 170

$$\int \frac{\tan^m(c+dx)\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)}{\sqrt{b\tan(c+dx)}}\,dx = \frac{2C\tan^{1+m}(c+dx)}{d(1+2m)\sqrt{b\tan(c+dx)}} + \frac{2(A-C)\,\text{Hypergeometric2F1}\left(1,\frac{1}{4}(1+2m),\frac{1}{4}(5+2m),-\tan^2(c+dx)\right)\tan^{1+m}(c+dx)}{d(1+2m)\sqrt{b\tan(c+dx)}} + \frac{2B\,\text{Hypergeometric2F1}\left(1,\frac{1}{4}(3+2m),\frac{1}{4}(7+2m),-\tan^2(c+dx)\right)\tan^{2+m}(c+dx)}{d(3+2m)\sqrt{b\tan(c+dx)}}$$

```
[Out] 2*C*tan(d*x+c)^(1+m)/d/(1+2*m)/(b*tan(d*x+c))^(1/2)+2*(A-C)*hypergeom([1, 1
/4+1/2*m],[5/4+1/2*m],-tan(d*x+c)^2)*tan(d*x+c)^(1+m)/d/(1+2*m)/(b*tan(d*x+
c))^(1/2)+2*B*hypergeom([1, 3/4+1/2*m],[7/4+1/2*m],-tan(d*x+c)^2)*tan(d*x+c
)^(2+m)/d/(3+2*m)/(b*tan(d*x+c))^(1/2)
```

Rubi [A] (verified)

Time = 0.17 (sec) , antiderivative size = 170, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.122, Rules used = {20, 3711, 3619, 3557, 371}

$$\int \frac{\tan^m(c+dx)\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)}{\sqrt{b\tan(c+dx)}}\,dx$$
$$=\frac{2(A-C)\tan^{m+1}(c+dx)\,\text{Hypergeometric2F1}\left(1,\frac{1}{4}(2m+1),\frac{1}{4}(2m+5),-\tan^2(c+dx)\right)}{d(2m+1)\sqrt{b\tan(c+dx)}}$$
$$+\frac{2B\tan^{m+2}(c+dx)\,\text{Hypergeometric2F1}\left(1,\frac{1}{4}(2m+3),\frac{1}{4}(2m+7),-\tan^2(c+dx)\right)}{d(2m+3)\sqrt{b\tan(c+dx)}}$$
$$+\frac{2C\tan^{m+1}(c+dx)}{d(2m+1)\sqrt{b\tan(c+dx)}}$$

```
[In] Int[(Tan[c + d*x]^m*(A + B*Tan[c + d*x] + C*Tan[c + d*x]^2))/Sqrt[b*Tan[c +
 d*x]],x]
```

```
[Out] (2*C*Tan[c + d*x]^(1 + m))/(d*(1 + 2*m)*Sqrt[b*Tan[c + d*x]]) + (2*(A - C)*
Hypergeometric2F1[1, (1 + 2*m)/4, (5 + 2*m)/4, -Tan[c + d*x]^2]*Tan[c + d*x
]^(1 + m))/(d*(1 + 2*m)*Sqrt[b*Tan[c + d*x]]) + (2*B*Hypergeometric2F1[1, (
3 + 2*m)/4, (7 + 2*m)/4, -Tan[c + d*x]^2]*Tan[c + d*x]^(2 + m))/(d*(3 + 2*m
)*Sqrt[b*Tan[c + d*x]])
```

Rule 20

```
Int[(u_.)*((a_.)*(v_))^(m_)*((b_.)*(v_))^(n_), x_Symbol] :> Dist[b^IntPart[
n]*((b*v)^FracPart[n]/(a^IntPart[n]*(a*v)^FracPart[n])), Int[u*(a*v)^(m + n
), x], x] /; FreeQ[{a, b, m, n}, x] && !IntegerQ[m] && !IntegerQ[n] && !
IntegerQ[m + n]
```

Rule 371

```
Int[((c_.)*(x_))^(m_.)*((a_) + (b_.)*(x_)^(n_))^(p_), x_Symbol] :> Simp[a^p
*((c*x)^(m + 1)/(c*(m + 1)))*Hypergeometric2F1[-p, (m + 1)/n, (m + 1)/n + 1
, (-b)*(x^n/a)], x] /; FreeQ[{a, b, c, m, n, p}, x] && !IGtQ[p, 0] && (ILt
Q[p, 0] || GtQ[a, 0])
```

Rule 3557

```
Int[((b_.)*tan[(c_.) + (d_.)*(x_)])^(n_), x_Symbol] :> Dist[b/d, Subst[Int[
x^n/(b^2 + x^2), x], x, b*Tan[c + d*x]], x] /; FreeQ[{b, c, d, n}, x] && !
IntegerQ[n]
```

Rule 3619

```
Int[((b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) + (f_.)*(x
_)]), x_Symbol] :> Dist[c, Int[(b*Tan[e + f*x])^m, x], x] + Dist[d/b, Int[(
b*Tan[e + f*x])^(m + 1), x], x] /; FreeQ[{b, c, d, e, f, m}, x] && NeQ[c^2
+ d^2, 0] &&  !IntegerQ[2*m]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rubi steps

$$\text{integral} = \frac{\sqrt{\tan(c+dx)}\int \tan^{-\frac{1}{2}+m}(c+dx)\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx}{\sqrt{b\tan(c+dx)}}$$

$$= \frac{2C\tan^{1+m}(c+dx)}{d(1+2m)\sqrt{b\tan(c+dx)}}+\frac{\sqrt{\tan(c+dx)}\int \tan^{-\frac{1}{2}+m}(c+dx)(A-C+B\tan(c+dx))\,dx}{\sqrt{b\tan(c+dx)}}$$

$$= \frac{2C\tan^{1+m}(c+dx)}{d(1+2m)\sqrt{b\tan(c+dx)}}+\frac{\left(B\sqrt{\tan(c+dx)}\right)\int \tan^{\frac{1}{2}+m}(c+dx)\,dx}{\sqrt{b\tan(c+dx)}}+\frac{\left((A-C)\sqrt{\tan(c+dx)}\right)\int \tan^{-\frac{1}{2}+m}(c+dx)\,dx}{\sqrt{b\tan(c+dx)}}$$

$$= \frac{2C\tan^{1+m}(c+dx)}{d(1+2m)\sqrt{b\tan(c+dx)}}+\frac{\left(B\sqrt{\tan(c+dx)}\right)\text{Subst}\left(\int \frac{x^{\frac{1}{2}+m}}{1+x^2}\,dx,x,\tan(c+dx)\right)}{d\sqrt{b\tan(c+dx)}}+\frac{\left((A-C)\sqrt{\tan(c+dx)}\right)\text{Subst}\left(\int \frac{x^{-\frac{1}{2}+m}}{1+x^2}\,dx,x,\tan(c+dx)\right)}{d\sqrt{b\tan(c+dx)}}$$

$$= \frac{2C\tan^{1+m}(c+dx)}{d(1+2m)\sqrt{b\tan(c+dx)}}+\frac{2(A-C)\,\text{Hypergeometric2F1}\left(1,\frac{1}{4}(1+2m),\frac{1}{4}(5+2m),-\tan^2(c+dx)\right)\tan^{1+m}(c+dx)}{d(1+2m)\sqrt{b\tan(c+dx)}}+\frac{2B\,\text{Hypergeometric2F1}\left(1,\frac{1}{4}(3+2m),\frac{1}{4}(7+2m),-\tan^2(c+dx)\right)\tan^{2+m}(c+dx)}{d(3+2m)\sqrt{b\tan(c+dx)}}$$

Mathematica [A] (verified)

Time = 0.53 (sec) , antiderivative size = 133, normalized size of antiderivative = 0.78

$$\int \frac{\tan^m(c+dx)\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)}{\sqrt{b\tan(c+dx)}}\,dx$$
$$=\frac{2\tan^{1+m}(c+dx)\left(C(3+2m)+(A-C)(3+2m)\,\text{Hypergeometric2F1}\left(1,\tfrac{1}{4}(1+2m),\tfrac{1}{4}(5+2m),-\tan^2(\ldots\right.\right.}{d(1+2m)(3+2m)\ldots}$$

[In] Integrate[(Tan[c + d*x]^m*(A + B*Tan[c + d*x] + C*Tan[c + d*x]^2))/Sqrt[b*T
an[c + d*x]],x]

[Out] (2*Tan[c + d*x]^(1 + m)*(C*(3 + 2*m) + (A - C)*(3 + 2*m)*Hypergeometric2F1[
1, (1 + 2*m)/4, (5 + 2*m)/4, -Tan[c + d*x]^2] + B*(1 + 2*m)*Hypergeometric2
F1[1, (3 + 2*m)/4, (7 + 2*m)/4, -Tan[c + d*x]^2]*Tan[c + d*x]))/(d*(1 + 2*m
)*(3 + 2*m)*Sqrt[b*Tan[c + d*x]])

Maple [F]

$$\int \frac{\tan(dx+c)^m\left(A+B\tan(dx+c)+C\tan(dx+c)^2\right)}{\sqrt{b\tan(dx+c)}}\,dx$$

[In] int(tan(d*x+c)^m*(A+B*tan(d*x+c)+C*tan(d*x+c)^2)/(b*tan(d*x+c))^(1/2),x)

[Out] int(tan(d*x+c)^m*(A+B*tan(d*x+c)+C*tan(d*x+c)^2)/(b*tan(d*x+c))^(1/2),x)

Fricas [F]

$$\int \frac{\tan^m(c+dx)\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)}{\sqrt{b\tan(c+dx)}}\,dx$$
$$=\int \frac{\left(C\tan(dx+c)^2+B\tan(dx+c)+A\right)\tan(dx+c)^m}{\sqrt{b\tan(dx+c)}}\,dx$$

[In] integrate(tan(d*x+c)^m*(A+B*tan(d*x+c)+C*tan(d*x+c)^2)/(b*tan(d*x+c))^(1/2)
,x, algorithm="fricas")

[Out] integral((C*tan(d*x + c)^2 + B*tan(d*x + c) + A)*sqrt(b*tan(d*x + c))*tan(d
*x + c)^m/(b*tan(d*x + c)), x)

Sympy [F]

$$\int \frac{\tan^m(c+dx)\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)}{\sqrt{b\tan(c+dx)}}\,dx = \int \frac{\left(A+B\tan\left(c+dx\right)+C\tan^2\left(c+dx\right)\right)\tan^m\left(c+dx\right)}{\sqrt{b\tan\left(c+dx\right)}}\,dx$$

```
[In] integrate(tan(d*x+c)**m*(A+B*tan(d*x+c)+C*tan(d*x+c)**2)/(b*tan(d*x+c))**(1
/2),x)
```

```
[Out] Integral((A + B*tan(c + d*x) + C*tan(c + d*x)**2)*tan(c + d*x)**m/sqrt(b*ta
n(c + d*x)), x)
```

Maxima [F(-1)]

Timed out.

$$\int \frac{\tan^m(c+dx)\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)}{\sqrt{b\tan(c+dx)}}\,dx = \text{Timed out}$$

```
[In] integrate(tan(d*x+c)^m*(A+B*tan(d*x+c)+C*tan(d*x+c)^2)/(b*tan(d*x+c))^(1/2)
,x, algorithm="maxima")
```

```
[Out] Timed out
```

Giac [F(-1)]

Timed out.

$$\int \frac{\tan^m(c+dx)\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)}{\sqrt{b\tan(c+dx)}}\,dx = \text{Timed out}$$

```
[In] integrate(tan(d*x+c)^m*(A+B*tan(d*x+c)+C*tan(d*x+c)^2)/(b*tan(d*x+c))^(1/2)
,x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{\tan^m(c+dx)\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)}{\sqrt{b\tan(c+dx)}}\,dx$$
$$=\int \frac{\tan(c+d\,x)^m\,\left(C\tan(c+d\,x)^2+B\tan(c+d\,x)+A\right)}{\sqrt{b\tan\left(c+d\,x\right)}}\,dx$$

```
[In] int((tan(c + d*x)^m*(A + B*tan(c + d*x) + C*tan(c + d*x)^2))/(b*tan(c + d*x
))^(1/2),x)
[Out] int((tan(c + d*x)^m*(A + B*tan(c + d*x) + C*tan(c + d*x)^2))/(b*tan(c + d*x
))^(1/2), x)
```

3.49 $\int \frac{\tan^m(c+dx)\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)}{\sqrt{a+b\tan(c+dx)}}\,dx$

Optimal result 404
Rubi [A] (verified) 405
Mathematica [F] 408
Maple [F] 408
Fricas [F] 408
Sympy [F] 409
Maxima [F] 409
Giac [F(-1)] 409
Mupad [F(-1)] 410

Optimal result

Integrand size = 43, antiderivative size = 328

$$\int \frac{\tan^m(c+dx)\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)}{\sqrt{a+b\tan(c+dx)}}\,dx = -\frac{\left(bB+\sqrt{-b^2}(A-C)\right)\operatorname{AppellF1}\left(\frac{1}{2},1,-m,\frac{3}{2},\frac{a+b\tan(c+dx)}{a-\sqrt{-b^2}},1+\frac{b\tan(c+dx)}{a}\right)\tan^m(c+dx)\left(-\frac{b\tan(c+dx)}{a}\right)^{-m}\cdots}{b\left(a-\sqrt{-b^2}\right)d} - \frac{\left(bB-\sqrt{-b^2}(A-C)\right)\operatorname{AppellF1}\left(\frac{1}{2},1,-m,\frac{3}{2},\frac{a+b\tan(c+dx)}{a+\sqrt{-b^2}},1+\frac{b\tan(c+dx)}{a}\right)\tan^m(c+dx)\left(-\frac{b\tan(c+dx)}{a}\right)^{-m}\cdots}{b\left(a+\sqrt{-b^2}\right)d} + \frac{2C\,\text{Hypergeometric2F1}\left(\frac{1}{2},-m,\frac{3}{2},1+\frac{b\tan(c+dx)}{a}\right)\tan^m(c+dx)\left(-\frac{b\tan(c+dx)}{a}\right)^{-m}\sqrt{a+b\tan(c+dx)}}{bd}$$

```
[Out] 2*C*hypergeom([1/2, -m],[3/2],1+b*tan(d*x+c)/a)*(a+b*tan(d*x+c))^(1/2)*tan(
d*x+c)^m/b/d/((-b*tan(d*x+c)/a)^m)-AppellF1(1/2,1,-m,3/2,(a+b*tan(d*x+c))/(
a+(-b^2)^(1/2)),1+b*tan(d*x+c)/a)*(B*b-(A-C)*(-b^2)^(1/2))*(a+b*tan(d*x+c))
^(1/2)*tan(d*x+c)^m/b/d/(a+(-b^2)^(1/2))/((-b*tan(d*x+c)/a)^m)-AppellF1(1/2
,1,-m,3/2,(a+b*tan(d*x+c))/(a-(-b^2)^(1/2)),1+b*tan(d*x+c)/a)*(B*b+(A-C)*(-
b^2)^(1/2))*(a+b*tan(d*x+c))^(1/2)*tan(d*x+c)^m/b/d/(a-(-b^2)^(1/2))/((-b*t
an(d*x+c)/a)^m)
```

Rubi [A] (verified)

Time = 1.93 (sec) , antiderivative size = 328, normalized size of antiderivative = 1.00, number of steps used = 13, number of rules used = 7, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.163, Rules used = {3736, 6852, 1706, 252, 251, 441, 440}

$$\int \frac{\tan^m(c+dx)\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)}{\sqrt{a+b\tan(c+dx)}}\,dx = -\frac{\left(\sqrt{-b^2}(A-C)+bB\right)\tan^m(c+dx)\sqrt{a+b\tan(c+dx)}\left(-\frac{b\tan(c+dx)}{a}\right)^{-m}\text{AppellF1}\left(\frac{1}{2},1,-m,\frac{3}{2},\frac{a+b\tan}{a-\sqrt{}}\right.}{bd\left(a-\sqrt{-b^2}\right)} - \frac{\left(bB-\sqrt{-b^2}(A-C)\right)\tan^m(c+dx)\sqrt{a+b\tan(c+dx)}\left(-\frac{b\tan(c+dx)}{a}\right)^{-m}\text{AppellF1}\left(\frac{1}{2},1,-m,\frac{3}{2},\frac{a+b\tan}{a+\sqrt{}}\right.}{bd\left(a+\sqrt{-b^2}\right)} + \frac{2C\tan^m(c+dx)\sqrt{a+b\tan(c+dx)}\left(-\frac{b\tan(c+dx)}{a}\right)^{-m}\text{Hypergeometric2F1}\left(\frac{1}{2},-m,\frac{3}{2},\frac{b\tan(c+dx)}{a}+1\right)}{bd}$$

```
[In] Int[(Tan[c + d*x]^m*(A + B*Tan[c + d*x] + C*Tan[c + d*x]^2))/Sqrt[a + b*Tan
[c + d*x]],x]
```

```
[Out] -(((b*B + Sqrt[-b^2]*(A - C))*AppellF1[1/2, 1, -m, 3/2, (a + b*Tan[c + d*x]
)/(a - Sqrt[-b^2]), 1 + (b*Tan[c + d*x])/a]*Tan[c + d*x]^m*Sqrt[a + b*Tan[c
 + d*x]])/(b*(a - Sqrt[-b^2])*d*(-((b*Tan[c + d*x])/a))^m)) - ((b*B - Sqrt[
-b^2]*(A - C))*AppellF1[1/2, 1, -m, 3/2, (a + b*Tan[c + d*x])/(a + Sqrt[-b^
2]), 1 + (b*Tan[c + d*x])/a]*Tan[c + d*x]^m*Sqrt[a + b*Tan[c + d*x]])/(b*(a
 + Sqrt[-b^2])*d*(-((b*Tan[c + d*x])/a))^m) + (2*C*Hypergeometric2F1[1/2, -
m, 3/2, 1 + (b*Tan[c + d*x])/a]*Tan[c + d*x]^m*Sqrt[a + b*Tan[c + d*x]])/(b
*d*(-((b*Tan[c + d*x])/a))^m)
```

Rule 251

```
Int[((a_) + (b_.)*(x_)^(n_))^(p_), x_Symbol] :> Simp[a^p*x*Hypergeometric2F
1[-p, 1/n, 1/n + 1, (-b)*(x^n/a)], x] /; FreeQ[{a, b, n, p}, x] && !IGtQ[p
, 0] && !IntegerQ[1/n] && !ILtQ[Simplify[1/n + p], 0] && (IntegerQ[p] ||
GtQ[a, 0])
```

Rule 252

```
Int[((a_) + (b_.)*(x_)^(n_))^(p_), x_Symbol] :> Dist[a^IntPart[p]*((a + b*x
^n)^FracPart[p]/(1 + b*(x^n/a))^FracPart[p]), Int[(1 + b*(x^n/a))^p, x], x]
 /; FreeQ[{a, b, n, p}, x] && !IGtQ[p, 0] && !IntegerQ[1/n] && !ILtQ[Sim
plify[1/n + p], 0] && !(IntegerQ[p] || GtQ[a, 0])
```

Rule 440

```
Int[((a_) + (b_.)*(x_)^(n_))^(p_)*((c_) + (d_.)*(x_)^(n_))^(q_), x_Symbol]
:> Simp[a^p*c^q*x*AppellF1[1/n, -p, -q, 1 + 1/n, (-b)*(x^n/a), (-d)*(x^n/c)
```

```
], x] /; FreeQ[{a, b, c, d, n, p, q}, x] && NeQ[b*c - a*d, 0] && NeQ[n, -1]
 && (IntegerQ[p] || GtQ[a, 0]) && (IntegerQ[q] || GtQ[c, 0])
```

Rule 441

```
Int[((a_) + (b_.)*(x_)^(n_))^(p_)*((c_) + (d_.)*(x_)^(n_))^(q_), x_Symbol]
:> Dist[a^IntPart[p]*((a + b*x^n)^FracPart[p]/(1 + b*(x^n/a))^FracPart[p]),
 Int[(1 + b*(x^n/a))^p*(c + d*x^n)^q, x], x] /; FreeQ[{a, b, c, d, n, p, q}
, x] && NeQ[b*c - a*d, 0] && NeQ[n, -1] &&  !(IntegerQ[p] || GtQ[a, 0])
```

Rule 1706

```
Int[(Px_)*((d_) + (e_.)*(x_)^2)^(q_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^
(p_.), x_Symbol] :> Int[ExpandIntegrand[Px*(d + e*x^2)^q*(a + b*x^2 + c*x^4
)^p, x], x] /; FreeQ[{a, b, c, d, e, q}, x] && PolyQ[Px, x^2] && NeQ[b^2 -
4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] && IntegerQ[p]
```

Rule 3736

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[(a + b*ff*x)^m*(c + d*ff*x)^n*((A + B*ff*x + C*ff^2*x^2
)/(1 + ff^2*x^2)), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6852

```
Int[(u_.)*((a_.)*(v_)^(m_.))^(p_), x_Symbol] :> Dist[a^IntPart[p]*((a*v^m)^
FracPart[p]/v^(m*FracPart[p])), Int[u*v^(m*p), x], x] /; FreeQ[{a, m, p}, x
] &&  !IntegerQ[p] &&  !FreeQ[v, x] &&  !(EqQ[a, 1] && EqQ[m, 1]) &&  !(EqQ
[v, x] && EqQ[m, 1])
```

Rubi steps

$$\text{integral} = \frac{\text{Subst}\left(\int \frac{x^m(A+Bx+Cx^2)}{\sqrt{a+bx}(1+x^2)}\,dx, x, \tan(c+dx)\right)}{d}$$

$$= \frac{2\text{Subst}\left(\int \frac{\left(\frac{-a+x^2}{b}\right)^m (Ab^2+(a-x^2)(-bB+C(a-x^2)))}{a^2+b^2-2ax^2+x^4}\,dx, x, \sqrt{a+b\tan(c+dx)}\right)}{bd}$$

$$= \frac{(2\tan^m(c+dx)(b\tan(c+dx))^{-m})\,\text{Subst}\left(\int \frac{(-a+x^2)^m (Ab^2+(a-x^2)(-bB+C(a-x^2)))}{a^2+b^2-2ax^2+x^4}\,dx, x, \sqrt{a+b\tan(c-\right.}{bd}$$

$$= \frac{\left(2\tan^m(c+dx)(b\tan(c+dx))^{-m}\right)\text{Subst}\left(\int\left(C(-a+x^2)^m+\frac{(-a+x^2)^m(b(Ab-aB-bC)+bBx^2)}{a^2+b^2-2ax^2+x^4}\right)dx,x,\right.}{bd}$$

$$= \frac{\left(2\tan^m(c+dx)(b\tan(c+dx))^{-m}\right)\text{Subst}\left(\int\frac{(-a+x^2)^m(b(Ab-aB-bC)+bBx^2)}{a^2+b^2-2ax^2+x^4}dx,x,\sqrt{a+b\tan(c+dx)}\right.}{bd} + \frac{\left(2C\tan^m(c+dx)(b\tan(c+dx))^{-m}\right)\text{Subst}\left(\int(-a+x^2)^m\,dx,x,\sqrt{a+b\tan(c+dx)}\right)}{bd}$$

$$= \frac{\left(2\tan^m(c+dx)(b\tan(c+dx))^{-m}\right)\text{Subst}\left(\int\left(\frac{\left(bB-\sqrt{-b^2}(A-C)\right)(-a+x^2)^m}{-2a-2\sqrt{-b^2}+2x^2}+\frac{\left(bB+\sqrt{-b^2}(A-C)\right)(-a+x^2)^m}{-2a+2\sqrt{-b^2}+2x^2}\right.\right.}{bd} + \frac{\left(2C\tan^m(c+dx)\left(-\frac{b\tan(c+dx)}{a}\right)^{-m}\right)\text{Subst}\left(\int\left(1-\frac{x^2}{a}\right)^m dx,x,\sqrt{a+b\tan(c+dx)}\right)}{bd}$$

$$= \frac{2C\,\text{Hypergeometric2F1}\left(\frac{1}{2},-m,\frac{3}{2},\frac{a+b\tan(c+dx)}{a}\right)\tan^m(c+dx)\left(-\frac{b\tan(c+dx)}{a}\right)^{-m}\sqrt{a+b\tan(c+d}}{bd} + \frac{\left(2\left(bB-\sqrt{-b^2}(A-C)\right)\tan^m(c+dx)(b\tan(c+dx))^{-m}\right)\text{Subst}\left(\int\frac{(-a+x^2)^m}{-2a-2\sqrt{-b^2}+2x^2}dx,x,\sqrt{a+b}\right.}{bd} + \frac{\left(2\left(bB+\sqrt{-b^2}(A-C)\right)\tan^m(c+dx)(b\tan(c+dx))^{-m}\right)\text{Subst}\left(\int\frac{(-a+x^2)^m}{-2a+2\sqrt{-b^2}+2x^2}dx,x,\sqrt{a+b}\right.}{bd}$$

$$= \frac{2C\,\text{Hypergeometric2F1}\left(\frac{1}{2},-m,\frac{3}{2},\frac{a+b\tan(c+dx)}{a}\right)\tan^m(c+dx)\left(-\frac{b\tan(c+dx)}{a}\right)^{-m}\sqrt{a+b\tan(c+d}}{bd} + \frac{\left(2\left(bB-\sqrt{-b^2}(A-C)\right)\tan^m(c+dx)\left(-\frac{b\tan(c+dx)}{a}\right)^{-m}\right)\text{Subst}\left(\int\frac{\left(1-\frac{x^2}{a}\right)^m}{-2a-2\sqrt{-b^2}+2x^2}dx,x,\sqrt{a+}\right.}{bd} + \frac{\left(2\left(bB+\sqrt{-b^2}(A-C)\right)\tan^m(c+dx)\left(-\frac{b\tan(c+dx)}{a}\right)^{-m}\right)\text{Subst}\left(\int\frac{\left(1-\frac{x^2}{a}\right)^m}{-2a+2\sqrt{-b^2}+2x^2}dx,x,\sqrt{a+}\right.}{bd}$$

$$= -\frac{\left(bB+\sqrt{-b^2}(A-C)\right)\text{AppellF1}\left(\frac{1}{2},1,-m,\frac{3}{2},\frac{a+b\tan(c+dx)}{a-\sqrt{-b^2}},\frac{a+b\tan(c+dx)}{a}\right)\tan^m(c+dx)\left(-\frac{b\tan(c+}{a}\right.}{b\left(a-\sqrt{-b^2}\right)d} - \frac{\left(bB-\sqrt{-b^2}(A-C)\right)\text{AppellF1}\left(\frac{1}{2},1,-m,\frac{3}{2},\frac{a+b\tan(c+dx)}{a+\sqrt{-b^2}},\frac{a+b\tan(c+dx)}{a}\right)\tan^m(c+dx)\left(-\frac{b\tan(c+}{a}\right.}{b\left(a+\sqrt{-b^2}\right)d} + \frac{2C\,\text{Hypergeometric2F1}\left(\frac{1}{2},-m,\frac{3}{2},\frac{a+b\tan(c+dx)}{a}\right)\tan^m(c+dx)\left(-\frac{b\tan(c+dx)}{a}\right)^{-m}\sqrt{a+b\tan(c+}}{bd}$$

Mathematica [F]

$$\int \frac{\tan^m(c+dx)\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)}{\sqrt{a+b\tan(c+dx)}}\,dx = \int \frac{\tan^m(c+dx)\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)}{\sqrt{a+b\tan(c+dx)}}\,dx$$

```
[In] Integrate[(Tan[c + d*x]^m*(A + B*Tan[c + d*x] + C*Tan[c + d*x]^2))/Sqrt[a +
 b*Tan[c + d*x]],x]
[Out] Integrate[(Tan[c + d*x]^m*(A + B*Tan[c + d*x] + C*Tan[c + d*x]^2))/Sqrt[a +
 b*Tan[c + d*x]], x]
```

Maple [F]

$$\int \frac{\tan\left(dx+c\right)^{m}\left(A+B\tan\left(dx+c\right)+C\tan\left(dx+c\right)^{2}\right)}{\sqrt{a+b\tan\left(dx+c\right)}}dx$$

```
[In] int(tan(d*x+c)^m*(A+B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^(1/2),x)
[Out] int(tan(d*x+c)^m*(A+B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^(1/2),x)
```

Fricas [F]

$$\int \frac{\tan^m(c+dx)\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)}{\sqrt{a+b\tan(c+dx)}}\,dx = \int \frac{\left(C\tan\left(dx+c\right)^{2}+B\tan\left(dx+c\right)+A\right)\tan\left(dx+c\right)^{m}}{\sqrt{b\tan\left(dx+c\right)+a}}\,dx$$

```
[In] integrate(tan(d*x+c)^m*(A+B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^(1/
2),x, algorithm="fricas")
[Out] integral((C*tan(d*x + c)^2 + B*tan(d*x + c) + A)*tan(d*x + c)^m/sqrt(b*tan(
d*x + c) + a), x)
```

Sympy [F]

$$\int \frac{\tan^m(c+dx)\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)}{\sqrt{a+b\tan(c+dx)}}\,dx = \int \frac{\left(A+B\tan\left(c+dx\right)+C\tan^2\left(c+dx\right)\right)\tan^m\left(c+dx\right)}{\sqrt{a+b\tan\left(c+dx\right)}}\,dx$$

```
[In] integrate(tan(d*x+c)**m*(A+B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c))**(1/2),x)
[Out] Integral((A + B*tan(c + d*x) + C*tan(c + d*x)**2)*tan(c + d*x)**m/sqrt(a + b*tan(c + d*x)), x)
```

Maxima [F]

$$\int \frac{\tan^m(c+dx)\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)}{\sqrt{a+b\tan(c+dx)}}\,dx = \int \frac{\left(C\tan\left(dx+c\right)^2+B\tan\left(dx+c\right)+A\right)\tan\left(dx+c\right)^m}{\sqrt{b\tan\left(dx+c\right)+a}}\,dx$$

```
[In] integrate(tan(d*x+c)^m*(A+B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^(1/2),x, algorithm="maxima")
[Out] integrate((C*tan(d*x + c)^2 + B*tan(d*x + c) + A)*tan(d*x + c)^m/sqrt(b*tan(d*x + c) + a), x)
```

Giac [F(-1)]

Timed out.

$$\int \frac{\tan^m(c+dx)\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)}{\sqrt{a+b\tan(c+dx)}}\,dx = \text{Timed out}$$

```
[In] integrate(tan(d*x+c)^m*(A+B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^(1/2),x, algorithm="giac")
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{\tan^m(c+dx)\,(A+B\tan(c+dx)+C\tan^2(c+dx))}{\sqrt{a+b\tan(c+dx)}}\,dx$$
$$=\int \frac{\tan(c+d\,x)^m\,\left(C\tan(c+d\,x)^2+B\tan(c+d\,x)+A\right)}{\sqrt{a+b\tan\left(c+d\,x\right)}}\,dx$$

```
[In] int((tan(c + d*x)^m*(A + B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c +
 d*x))^(1/2),x)
[Out] int((tan(c + d*x)^m*(A + B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c +
 d*x))^(1/2), x)
```

3.50 $\int (a+b\tan(e+fx))^3(c+d\tan(e+fx))\left(A+B\tan(e+\right.$

Optimal result 411
Rubi [A] (verified) 412
Mathematica [C] (verified) 415
Maple [A] (verified) 415
Fricas [A] (verification not implemented) 416
Sympy [B] (verification not implemented) 416
Maxima [A] (verification not implemented) 417
Giac [B] (verification not implemented) 418
Mupad [B] (verification not implemented) 425

Optimal result

Integrand size = 43, antiderivative size = 353

$$\int (a+b\tan(e+fx))^3(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\left(a^3(Ac-cC-Bd)-3ab^2(Ac-cC-Bd)-3a^2b(Bc+(A-C)d)+b^3(Bc+(A-C)d)\right)x$$
$$-\frac{(3a^2b(Ac-cC-Bd)-b^3(Ac-cC-Bd)+a^3(Bc+(A-C)d)-3ab^2(Bc+(A-C)d))\log(\cos(e+}{f}$$
$$+\frac{b(2ab(Ac-cC-Bd)+a^2(Bc+(A-C)d)-b^2(Bc+(A-C)d))\tan(e+fx)}{f}$$
$$+\frac{(Abc+aBc-bcC+aAd-bBd-aCd)(a+b\tan(e+fx))^2}{2f}$$
$$+\frac{(Bc+(A-C)d)(a+b\tan(e+fx))^3}{3f}$$
$$-\frac{(aCd-5b(cC+Bd))(a+b\tan(e+fx))^4}{20b^2f}+\frac{Cd\tan(e+fx)(a+b\tan(e+fx))^4}{5bf}$$

```
[Out] (a^3*(A*c-B*d-C*c)-3*a*b^2*(A*c-B*d-C*c)-3*a^2*b*(B*c+(A-C)*d)+b^3*(B*c+(A-
C)*d))*x-(3*a^2*b*(A*c-B*d-C*c)-b^3*(A*c-B*d-C*c)+a^3*(B*c+(A-C)*d)-3*a*b^2
*(B*c+(A-C)*d))*ln(cos(f*x+e))/f+b*(2*a*b*(A*c-B*d-C*c)+a^2*(B*c+(A-C)*d)-b
^2*(B*c+(A-C)*d))*tan(f*x+e)/f+1/2*(A*a*d+A*b*c+B*a*c-B*b*d-C*a*d-C*b*c)*(a
+b*tan(f*x+e))^2/f+1/3*(B*c+(A-C)*d)*(a+b*tan(f*x+e))^3/f-1/20*(C*a*d-5*b*(
B*d+C*c))*(a+b*tan(f*x+e))^4/b^2/f+1/5*C*d*tan(f*x+e)*(a+b*tan(f*x+e))^4/b/
f
```

Rubi [A] (verified)

Time = 0.92 (sec) , antiderivative size = 353, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.116, Rules used = {3718, 3711, 3609, 3606, 3556}

$$\int (a+b\tan(e+fx))^3(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\frac{b\tan(e+fx)\left(a^2(d(A-C)+Bc)+2ab(Ac-Bd-cC)-b^2(d(A-C)+Bc)\right)}{f}$$
$$-\frac{\log(\cos(e+fx))\left(a^3(d(A-C)+Bc)+3a^2b(Ac-Bd-cC)-3ab^2(d(A-C)+Bc)-b^3(Ac-Bd-c\right.}{f}$$
$$+x\left(a^3(Ac-Bd-cC)-3a^2b(d(A-C)+Bc)-3ab^2(Ac-Bd-cC)+b^3(d(A-C)+Bc)\right)$$
$$+\frac{(d(A-C)+Bc)(a+b\tan(e+fx))^3}{3f}$$
$$+\frac{(a+b\tan(e+fx))^2(aAd+aBc-aCd+Abc-bBd-bcC)}{2f}$$
$$-\frac{(aCd-5b(Bd+cC))(a+b\tan(e+fx))^4}{20b^2f}+\frac{Cd\tan(e+fx)(a+b\tan(e+fx))^4}{5bf}$$

```
[In] Int[(a + b*Tan[e + f*x])^3*(c + d*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan
[e + f*x]^2),x]
```

```
[Out] (a^3*(A*c - c*C - B*d) - 3*a*b^2*(A*c - c*C - B*d) - 3*a^2*b*(B*c + (A - C)
*d) + b^3*(B*c + (A - C)*d))*x - ((3*a^2*b*(A*c - c*C - B*d) - b^3*(A*c - c
*C - B*d) + a^3*(B*c + (A - C)*d) - 3*a*b^2*(B*c + (A - C)*d))*Log[Cos[e +
f*x]])/f + (b*(2*a*b*(A*c - c*C - B*d) + a^2*(B*c + (A - C)*d) - b^2*(B*c +
 (A - C)*d))*Tan[e + f*x])/f + ((A*b*c + a*B*c - b*c*C + a*A*d - b*B*d - a*
C*d)*(a + b*Tan[e + f*x])^2)/(2*f) + ((B*c + (A - C)*d)*(a + b*Tan[e + f*x]
)^3)/(3*f) - ((a*C*d - 5*b*(c*C + B*d))*(a + b*Tan[e + f*x])^4)/(20*b^2*f)
+ (C*d*Tan[e + f*x]*(a + b*Tan[e + f*x])^4)/(5*b*f)
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3606

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
 f*x], x], x] + Simp[b*d*(Tan[e + f*x]/f), x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3609

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[d*((a + b*Tan[e + f*x])^m/(f*m)), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rubi steps

$$\text{integral} = \frac{Cd\tan(e+fx)(a+b\tan(e+fx))^4}{5bf} - \frac{\int (a+b\tan(e+fx))^3\left(-5Abc+aCd-5b(Bc+(A-C)d)\tan(e+fx)+(aCd-5b(cC+Bd))\tan^2(\right.}{5b}$$

$$= -\frac{(aCd-5b(cC+Bd))(a+b\tan(e+fx))^4}{20b^2f} + \frac{Cd\tan(e+fx)(a+b\tan(e+fx))^4}{5bf} - \frac{\int (a+b\tan(e+fx))^3(-5b(Ac-cC-Bd)-5b(Bc+(A-C)d)\tan(e+fx))\,dx}{5b}$$

$$= \frac{(Bc+(A-C)d)(a+b\tan(e+fx))^3}{3f} - \frac{(aCd-5b(cC+Bd))(a+b\tan(e+fx))^4}{20b^2f} + \frac{Cd\tan(e+fx)(a+b\tan(e+fx))^4}{5bf} - \frac{\int (a+b\tan(e+fx))^2(5b(bBc+b(A-C)d-a(Ac-cC-Bd))-5b(Abc+aBc-bcC+aA}{5b}$$

$$= \frac{(Abc+aBc-bcC+aAd-bBd-aCd)(a+b\tan(e+fx))^2}{2f} + \frac{(Bc+(A-C)d)(a+b\tan(e+fx))^3}{3f} - \frac{(aCd-5b(cC+Bd))(a+b\tan(e+fx))^4}{20b^2f} + \frac{Cd\tan(e+fx)(a+b\tan(e+fx))^4}{5bf} - \frac{\int(a+b\tan(e+fx))\left(-5b(a^2(Ac-cC-Bd)-b^2(Ac-cC-Bd)-2ab(Bc+(A-C)d))-5\right.}{5b}$$

$$= \left(a^3(Ac-cC-Bd)-3ab^2(Ac-cC-Bd)-3a^2b(Bc+(A-C)d) + b^3(Bc+(A-C)d)\right)x + \frac{b(2ab(Ac-cC-Bd)+a^2(Bc+(A-C)d)-b^2(Bc+(A-C)d))\tan(e+fx)}{f} + \frac{(Abc+aBc-bcC+aAd-bBd-aCd)(a+b\tan(e+fx))^2}{2f} + \frac{(Bc+(A-C)d)(a+b\tan(e+fx))^3}{3f} - \frac{(aCd-5b(cC+Bd))(a+b\tan(e+fx))^4}{20b^2f} + \frac{Cd\tan(e+fx)(a+b\tan(e+fx))^4}{5bf} - \left(-3a^2b(Ac-cC-Bd)+b^3(Ac-cC-Bd) - a^3(Bc+(A-C)d)+3ab^2(Bc+(A-C)d)\right)\int\tan(e+fx)\,dx$$

$$= \left(a^3(Ac-cC-Bd)-3ab^2(Ac-cC-Bd)-3a^2b(Bc+(A-C)d) + b^3(Bc+(A-C)d)\right)x - \frac{(3a^2b(Ac-cC-Bd)-b^3(Ac-cC-Bd)+a^3(Bc+(A-C)d)-3ab^2(Bc+(A-C)d))\log(c}{f} + \frac{b(2ab(Ac-cC-Bd)+a^2(Bc+(A-C)d)-b^2(Bc+(A-C)d))\tan(e+fx)}{f} + \frac{(Abc+aBc-bcC+aAd-bBd-aCd)(a+b\tan(e+fx))^2}{2f} + \frac{(Bc+(A-C)d)(a+b\tan(e+fx))^3}{3f} - \frac{(aCd-5b(cC+Bd))(a+b\tan(e+fx))^4}{20b^2f} + \frac{Cd\tan(e+fx)(a+b\tan(e+fx))^4}{5bf}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 6.42 (sec) , antiderivative size = 300, normalized size of antiderivative = 0.85

$$\int (a+b\tan(e+fx))^3(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\frac{Cd\tan(e+fx)(a+b\tan(e+fx))^4}{5bf}$$
$$-\frac{\frac{(aCd-5b(cC+Bd))(a+b\tan(e+fx))^4}{4bf}-\frac{5(3(Abc-aBc-bcC-aAd-bBd+aCd)((ia-b)^3\log(i-\tan(e+fx))-(ia+b)^3\log(i+\tan(e+fx))+6a}{}}{}$$

[In] Integrate[(a + b*Tan[e + f*x])^3*(c + d*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2),x]

[Out] (C*d*Tan[e + f*x]*(a + b*Tan[e + f*x])^4)/(5*b*f) - (((a*C*d - 5*b*(c*C + B*d))*(a + b*Tan[e + f*x])^4)/(4*b*f) - (5*(3*(A*b*c - a*B*c - b*c*C - a*A*d - b*B*d + a*C*d)*((I*a - b)^3*Log[I - Tan[e + f*x]] - (I*a + b)^3*Log[I + Tan[e + f*x]] + 6*a*b^2*Tan[e + f*x] + b^3*Tan[e + f*x]^2) - (B*c + (A - C)*d)*((3*I)*(a + I*b)^4*Log[I - Tan[e + f*x]] - (3*I)*(a - I*b)^4*Log[I + Tan[e + f*x]] - 6*b^2*(6*a^2 - b^2)*Tan[e + f*x] - 12*a*b^3*Tan[e + f*x]^2 - 2*b^4*Tan[e + f*x]^3)))/(6*f))/(5*b)

Maple [A] (verified)

Time = 0.30 (sec) , antiderivative size = 347, normalized size of antiderivative = 0.98

method	result
parts	$\frac{(A a^3 d+3A a^2 bc+B a^3 c)\ln\left(1+\tan(fx+e)^2\right)}{2f}+\frac{(B b^3 d+3C a b^2 d+C b^3 c)\left(\frac{\tan(fx+e)^4}{4}-\frac{\tan(fx+e)^2}{2}+\frac{\ln\left(1+\tan(fx+e)^2\right)}{2}\right.}{f}$
norman	$(A a^3 c-3A a^2 bd-3Aa b^2 c+A b^3 d-B a^3 d-3B a^2 bc+3Ba b^2 d+B b^3 c-C a^3 c+3$
derivativedivides	$\frac{\left(A a^3 d+3A a^2 bc-3Aa b^2 d-A b^3 c+B a^3 c-3B a^2 bd-3Ba b^2 c+B b^3 d-a^3 Cd-3C a^2 bc+3Ca b^2 d+C b^3 c\right)\ln\left(1+\tan(fx+e)^2\right)}{2}+(A a^3$
default	$\frac{\left(A a^3 d+3A a^2 bc-3Aa b^2 d-A b^3 c+B a^3 c-3B a^2 bd-3Ba b^2 c+B b^3 d-a^3 Cd-3C a^2 bc+3Ca b^2 d+C b^3 c\right)\ln\left(1+\tan(fx+e)^2\right)}{2}+(A a^3$
parallelrisch	$\frac{30C a^3 d\tan(fx+e)^2-30C b^3 c\tan(fx+e)^2+20B b^3 c\tan(fx+e)^3-20C b^3 d\tan(fx+e)^3+15B b^3 d\tan(fx+e)^4+15C b^3 c}{}$
risch	Expression too large to display

[In] int((a+b*tan(f*x+e))^3*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x,method=_RETURNVERBOSE)

```
[Out] 1/2*(A*a^3*d+3*A*a^2*b*c+B*a^3*c)/f*ln(1+tan(f*x+e)^2)+(B*b^3*d+3*C*a*b^2*d
+C*b^3*c)/f*(1/4*tan(f*x+e)^4-1/2*tan(f*x+e)^2+1/2*ln(1+tan(f*x+e)^2))+(A*b
^3*d+3*B*a*b^2*d+B*b^3*c+3*C*a^2*b*d+3*C*a*b^2*c)/f*(1/3*tan(f*x+e)^3-tan(f
*x+e)+arctan(tan(f*x+e)))+(3*A*a^2*b*d+3*A*a*b^2*c+B*a^3*d+3*B*a^2*b*c+C*a^
3*c)/f*(tan(f*x+e)-arctan(tan(f*x+e)))+(3*A*a*b^2*d+A*b^3*c+3*B*a^2*b*d+3*B
*a*b^2*c+C*a^3*d+3*C*a^2*b*c)/f*(1/2*tan(f*x+e)^2-1/2*ln(1+tan(f*x+e)^2))+A
*a^3*c*x+C*b^3*d/f*(1/5*tan(f*x+e)^5-1/3*tan(f*x+e)^3+tan(f*x+e)-arctan(tan
(f*x+e)))
```

Fricas [A] (verification not implemented)

none

Time = 0.27 (sec) , antiderivative size = 415, normalized size of antiderivative = 1.18

$$\int (a+b\tan(e+fx))^3(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\frac{12\,Cb^3d\tan\left(fx+e\right)^5+15\left(Cb^3c+(3\,Cab^2+Bb^3)d\right)\tan\left(fx+e\right)^4+20\left((3\,Cab^2+Bb^3)c+(3\,Ca^2b+3\,B\right.}{}$$

```
[In] integrate((a+b*tan(f*x+e))^3*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x, algorithm="fricas")
```

```
[Out] 1/60*(12*C*b^3*d*tan(f*x + e)^5 + 15*(C*b^3*c + (3*C*a*b^2 + B*b^3)*d)*tan(
f*x + e)^4 + 20*((3*C*a*b^2 + B*b^3)*c + (3*C*a^2*b + 3*B*a*b^2 + (A - C)*b
^3)*d)*tan(f*x + e)^3 + 60*(((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*
b^3)*c - (B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*d)*f*x + 30*((
3*C*a^2*b + 3*B*a*b^2 + (A - C)*b^3)*c + (C*a^3 + 3*B*a^2*b + 3*(A - C)*a*b
^2 - B*b^3)*d)*tan(f*x + e)^2 - 30*((B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 -
(A - C)*b^3)*c + ((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b^3)*d)*log
(1/(tan(f*x + e)^2 + 1)) + 60*((C*a^3 + 3*B*a^2*b + 3*(A - C)*a*b^2 - B*b^3
)*c + (B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*d)*tan(f*x + e))/
f
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1001 vs. 2(316) = 632.

Time = 0.29 (sec) , antiderivative size = 1001, normalized size of antiderivative = 2.84

$$\int (a+b\tan(e+fx))^3(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))**3*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)
**2),x)
```

```
[Out] Piecewise((A*a**3*c*x + A*a**3*d*log(tan(e + f*x)**2 + 1)/(2*f) + 3*A*a**2*
b*c*log(tan(e + f*x)**2 + 1)/(2*f) - 3*A*a**2*b*d*x + 3*A*a**2*b*d*tan(e +
f*x)/f - 3*A*a*b**2*c*x + 3*A*a*b**2*c*tan(e + f*x)/f - 3*A*a*b**2*d*log(ta
n(e + f*x)**2 + 1)/(2*f) + 3*A*a*b**2*d*tan(e + f*x)**2/(2*f) - A*b**3*c*lo
g(tan(e + f*x)**2 + 1)/(2*f) + A*b**3*c*tan(e + f*x)**2/(2*f) + A*b**3*d*x
+ A*b**3*d*tan(e + f*x)**3/(3*f) - A*b**3*d*tan(e + f*x)/f + B*a**3*c*log(t
an(e + f*x)**2 + 1)/(2*f) - B*a**3*d*x + B*a**3*d*tan(e + f*x)/f - 3*B*a**2
*b*c*x + 3*B*a**2*b*c*tan(e + f*x)/f - 3*B*a**2*b*d*log(tan(e + f*x)**2 + 1
)/(2*f) + 3*B*a**2*b*d*tan(e + f*x)**2/(2*f) - 3*B*a*b**2*c*log(tan(e + f*x
)**2 + 1)/(2*f) + 3*B*a*b**2*c*tan(e + f*x)**2/(2*f) + 3*B*a*b**2*d*x + B*a
*b**2*d*tan(e + f*x)**3/f - 3*B*a*b**2*d*tan(e + f*x)/f + B*b**3*c*x + B*b*
*3*c*tan(e + f*x)**3/(3*f) - B*b**3*c*tan(e + f*x)/f + B*b**3*d*log(tan(e +
 f*x)**2 + 1)/(2*f) + B*b**3*d*tan(e + f*x)**4/(4*f) - B*b**3*d*tan(e + f*x
)**2/(2*f) - C*a**3*c*x + C*a**3*c*tan(e + f*x)/f - C*a**3*d*log(tan(e + f*
x)**2 + 1)/(2*f) + C*a**3*d*tan(e + f*x)**2/(2*f) - 3*C*a**2*b*c*log(tan(e
+ f*x)**2 + 1)/(2*f) + 3*C*a**2*b*c*tan(e + f*x)**2/(2*f) + 3*C*a**2*b*d*x
+ C*a**2*b*d*tan(e + f*x)**3/f - 3*C*a**2*b*d*tan(e + f*x)/f + 3*C*a*b**2*c
*x + C*a*b**2*c*tan(e + f*x)**3/f - 3*C*a*b**2*c*tan(e + f*x)/f + 3*C*a*b**
2*d*log(tan(e + f*x)**2 + 1)/(2*f) + 3*C*a*b**2*d*tan(e + f*x)**4/(4*f) - 3
*C*a*b**2*d*tan(e + f*x)**2/(2*f) + C*b**3*c*log(tan(e + f*x)**2 + 1)/(2*f)
 + C*b**3*c*tan(e + f*x)**4/(4*f) - C*b**3*c*tan(e + f*x)**2/(2*f) - C*b**3
*d*x + C*b**3*d*tan(e + f*x)**5/(5*f) - C*b**3*d*tan(e + f*x)**3/(3*f) + C*
b**3*d*tan(e + f*x)/f, Ne(f, 0)), (x*(a + b*tan(e))**3*(c + d*tan(e))*(A +
B*tan(e) + C*tan(e)**2), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.40 (sec) , antiderivative size = 416, normalized size of antiderivative = 1.18

$$\int (a+b\tan(e+fx))^3(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= \frac{12\,Cb^3d\tan\left(fx+e\right)^5+15\left(Cb^3c+(3\,Cab^2+Bb^3)d\right)\tan\left(fx+e\right)^4+20\left((3\,Cab^2+Bb^3)c+(3\,Ca^2b+3\right.}{}$$

```
[In] integrate((a+b*tan(f*x+e))^3*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x, algorithm="maxima")
```

```
[Out] 1/60*(12*C*b^3*d*tan(f*x + e)^5 + 15*(C*b^3*c + (3*C*a*b^2 + B*b^3)*d)*tan(
f*x + e)^4 + 20*((3*C*a*b^2 + B*b^3)*c + (3*C*a^2*b + 3*B*a*b^2 + (A - C)*b
^3)*d)*tan(f*x + e)^3 + 30*((3*C*a^2*b + 3*B*a*b^2 + (A - C)*b^3)*c + (C*a^
3 + 3*B*a^2*b + 3*(A - C)*a*b^2 - B*b^3)*d)*tan(f*x + e)^2 + 60*(((A - C)*a
^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b^3)*c - (B*a^3 + 3*(A - C)*a^2*b - 3*
B*a*b^2 - (A - C)*b^3)*d)*(f*x + e) + 30*((B*a^3 + 3*(A - C)*a^2*b - 3*B*a*
```

```
b^2 - (A - C)*b^3)*c + ((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b^3)*
d)*log(tan(f*x + e)^2 + 1) + 60*((C*a^3 + 3*B*a^2*b + 3*(A - C)*a*b^2 - B*b
^3)*c + (B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*d)*tan(f*x + e)
)/f
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 10353 vs. $2(345) = 690$.

Time = 9.66 (sec) , antiderivative size = 10353, normalized size of antiderivative = 29.33

$$\int (a + b\tan(e + fx))^3 (c + d\tan(e + fx)) \left(A + B\tan(e + fx) + C\tan^2(e + fx)\right) dx$$
$$= \text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))^3*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x, algorithm="giac")
```

```
[Out] 1/60*(60*A*a^3*c*f*x*tan(f*x)^5*tan(e)^5 - 60*C*a^3*c*f*x*tan(f*x)^5*tan(e)
^5 - 180*B*a^2*b*c*f*x*tan(f*x)^5*tan(e)^5 - 180*A*a*b^2*c*f*x*tan(f*x)^5*t
an(e)^5 + 180*C*a*b^2*c*f*x*tan(f*x)^5*tan(e)^5 + 60*B*b^3*c*f*x*tan(f*x)^5
*tan(e)^5 - 60*B*a^3*d*f*x*tan(f*x)^5*tan(e)^5 - 180*A*a^2*b*d*f*x*tan(f*x)
^5*tan(e)^5 + 180*C*a^2*b*d*f*x*tan(f*x)^5*tan(e)^5 + 180*B*a*b^2*d*f*x*tan
(f*x)^5*tan(e)^5 + 60*A*b^3*d*f*x*tan(f*x)^5*tan(e)^5 - 60*C*b^3*d*f*x*tan(
f*x)^5*tan(e)^5 - 30*B*a^3*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e)
 + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^
5 - 90*A*a^2*b*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f
*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5 + 90*C*a^2
*b*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)
^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5 + 90*B*a*b^2*c*log(4*(
tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x
)^2 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5 + 30*A*b^3*c*log(4*(tan(f*x)^2*tan
(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2
 + 1))*tan(f*x)^5*tan(e)^5 - 30*C*b^3*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(
f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x
)^5*tan(e)^5 - 30*A*a^3*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) +
1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5 +
 30*C*a^3*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2
*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5 + 90*B*a^2*b*d*
log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 +
tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5 + 90*A*a*b^2*d*log(4*(tan(f
*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 +
 tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5 - 90*C*a*b^2*d*log(4*(tan(f*x)^2*tan(e)
^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 +
1))*tan(f*x)^5*tan(e)^5 - 30*B*b^3*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x
)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5
```

*tan(e)^5 - 300*A*a^3*c*f*x*tan(f*x)^4*tan(e)^4 + 300*C*a^3*c*f*x*tan(f*x)^
4*tan(e)^4 + 900*B*a^2*b*c*f*x*tan(f*x)^4*tan(e)^4 + 900*A*a*b^2*c*f*x*tan(
f*x)^4*tan(e)^4 - 900*C*a*b^2*c*f*x*tan(f*x)^4*tan(e)^4 - 300*B*b^3*c*f*x*t
an(f*x)^4*tan(e)^4 + 300*B*a^3*d*f*x*tan(f*x)^4*tan(e)^4 + 900*A*a^2*b*d*f*
x*tan(f*x)^4*tan(e)^4 - 900*C*a^2*b*d*f*x*tan(f*x)^4*tan(e)^4 - 900*B*a*b^2
*d*f*x*tan(f*x)^4*tan(e)^4 - 300*A*b^3*d*f*x*tan(f*x)^4*tan(e)^4 + 300*C*b^
3*d*f*x*tan(f*x)^4*tan(e)^4 + 90*C*a^2*b*c*tan(f*x)^5*tan(e)^5 + 90*B*a*b^2
*c*tan(f*x)^5*tan(e)^5 + 30*A*b^3*c*tan(f*x)^5*tan(e)^5 - 45*C*b^3*c*tan(f*
x)^5*tan(e)^5 + 30*C*a^3*d*tan(f*x)^5*tan(e)^5 + 90*B*a^2*b*d*tan(f*x)^5*ta
n(e)^5 + 90*A*a*b^2*d*tan(f*x)^5*tan(e)^5 - 135*C*a*b^2*d*tan(f*x)^5*tan(e)
^5 - 45*B*b^3*d*tan(f*x)^5*tan(e)^5 + 150*B*a^3*c*log(4*(tan(f*x)^2*tan(e)^
2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1
))*tan(f*x)^4*tan(e)^4 + 450*A*a^2*b*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f
*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)
^4*tan(e)^4 - 450*C*a^2*b*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e)
+ 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4
 - 450*B*a*b^2*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f
*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 - 150*A*b^
3*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^
2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 150*C*b^3*c*log(4*(ta
n(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^
2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 150*A*a^3*d*log(4*(tan(f*x)^2*tan(
e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2
+ 1))*tan(f*x)^4*tan(e)^4 - 150*C*a^3*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(
f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x
)^4*tan(e)^4 - 450*B*a^2*b*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e)
 + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^
4 - 450*A*a*b^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(
f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 450*C*a
*b^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(
e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 150*B*b^3*d*log(4*
(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*
x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 - 60*C*a^3*c*tan(f*x)^5*tan(e)^4
- 180*B*a^2*b*c*tan(f*x)^5*tan(e)^4 - 180*A*a*b^2*c*tan(f*x)^5*tan(e)^4 + 1
80*C*a*b^2*c*tan(f*x)^5*tan(e)^4 + 60*B*b^3*c*tan(f*x)^5*tan(e)^4 - 60*B*a^
3*d*tan(f*x)^5*tan(e)^4 - 180*A*a^2*b*d*tan(f*x)^5*tan(e)^4 + 180*C*a^2*b*d
*tan(f*x)^5*tan(e)^4 + 180*B*a*b^2*d*tan(f*x)^5*tan(e)^4 + 60*A*b^3*d*tan(f
*x)^5*tan(e)^4 - 60*C*b^3*d*tan(f*x)^5*tan(e)^4 - 60*C*a^3*c*tan(f*x)^4*tan
(e)^5 - 180*B*a^2*b*c*tan(f*x)^4*tan(e)^5 - 180*A*a*b^2*c*tan(f*x)^4*tan(e)
^5 + 180*C*a*b^2*c*tan(f*x)^4*tan(e)^5 + 60*B*b^3*c*tan(f*x)^4*tan(e)^5 - 6
0*B*a^3*d*tan(f*x)^4*tan(e)^5 - 180*A*a^2*b*d*tan(f*x)^4*tan(e)^5 + 180*C*a
^2*b*d*tan(f*x)^4*tan(e)^5 + 180*B*a*b^2*d*tan(f*x)^4*tan(e)^5 + 60*A*b^3*d
*tan(f*x)^4*tan(e)^5 - 60*C*b^3*d*tan(f*x)^4*tan(e)^5 + 600*A*a^3*c*f*x*tan
(f*x)^3*tan(e)^3 - 600*C*a^3*c*f*x*tan(f*x)^3*tan(e)^3 - 1800*B*a^2*b*c*f*x
*tan(f*x)^3*tan(e)^3 - 1800*A*a*b^2*c*f*x*tan(f*x)^3*tan(e)^3 + 1800*C*a*b^

2*c*f*x*tan(f*x)^3*tan(e)^3 + 600*B*b^3*c*f*x*tan(f*x)^3*tan(e)^3 - 600*B*a^3*d*f*x*tan(f*x)^3*tan(e)^3 - 1800*A*a^2*b*d*f*x*tan(f*x)^3*tan(e)^3 + 1800*C*a^2*b*d*f*x*tan(f*x)^3*tan(e)^3 + 1800*B*a*b^2*d*f*x*tan(f*x)^3*tan(e)^3 + 600*A*b^3*d*f*x*tan(f*x)^3*tan(e)^3 - 600*C*b^3*d*f*x*tan(f*x)^3*tan(e)^3 + 90*C*a^2*b*c*tan(f*x)^5*tan(e)^3 + 90*B*a*b^2*c*tan(f*x)^5*tan(e)^3 + 30*A*b^3*c*tan(f*x)^5*tan(e)^3 - 30*C*b^3*c*tan(f*x)^5*tan(e)^3 + 30*C*a^3*d*tan(f*x)^5*tan(e)^3 + 90*B*a^2*b*d*tan(f*x)^5*tan(e)^3 + 90*A*a*b^2*d*tan(f*x)^5*tan(e)^3 - 90*C*a*b^2*d*tan(f*x)^5*tan(e)^3 - 30*B*b^3*d*tan(f*x)^5*tan(e)^3 - 270*C*a^2*b*c*tan(f*x)^4*tan(e)^4 - 270*B*a*b^2*c*tan(f*x)^4*tan(e)^4 - 90*A*b^3*c*tan(f*x)^4*tan(e)^4 + 165*C*b^3*c*tan(f*x)^4*tan(e)^4 - 90*C*a^3*d*tan(f*x)^4*tan(e)^4 - 270*B*a^2*b*d*tan(f*x)^4*tan(e)^4 - 270*A*a*b^2*d*tan(f*x)^4*tan(e)^4 + 495*C*a*b^2*d*tan(f*x)^4*tan(e)^4 + 165*B*b^3*d*tan(f*x)^4*tan(e)^4 + 90*C*a^2*b*c*tan(f*x)^3*tan(e)^5 + 90*B*a*b^2*c*tan(f*x)^3*tan(e)^5 + 30*A*b^3*c*tan(f*x)^3*tan(e)^5 - 30*C*b^3*c*tan(f*x)^3*tan(e)^5 + 30*C*a^3*d*tan(f*x)^3*tan(e)^5 + 90*B*a^2*b*d*tan(f*x)^3*tan(e)^5 + 90*A*a*b^2*d*tan(f*x)^3*tan(e)^5 - 90*C*a*b^2*d*tan(f*x)^3*tan(e)^5 - 30*B*b^3*d*tan(f*x)^3*tan(e)^5 - 60*C*a*b^2*c*tan(f*x)^5*tan(e)^2 - 20*B*b^3*c*tan(f*x)^5*tan(e)^2 - 60*C*a^2*b*d*tan(f*x)^5*tan(e)^2 - 60*B*a*b^2*d*tan(f*x)^5*tan(e)^2 - 20*A*b^3*d*tan(f*x)^5*tan(e)^2 + 20*C*b^3*d*tan(f*x)^5*tan(e)^2 - 300*B*a^3*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 900*A*a^2*b*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 900*C*a^2*b*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 900*B*a*b^2*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 300*A*b^3*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 300*C*b^3*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 300*A*a^3*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 300*C*a^3*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 900*B*a^2*b*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 900*A*a*b^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 900*C*a*b^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 300*B*b^3*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 240*C*a^3*c*tan(f*x)^4*tan(e)^3 + 720*B*a^2*b*c*tan(f*x)^4*tan(e)^3 + 720*A*a*b^2*c*tan(f*x)^4*tan(e)^3 - 900*C*a*b^2*c*tan(f*x)^4*tan(e)^3 - 300*B*b^3*c*tan(f*x)^4*tan(e)^3 + 240*B*a^3*d*tan(f*x)^4*tan(e)^3 + 720*A*a^2*b*d*tan(f*x)^4*tan(e)^3 - 900*C*a^2*b*d*tan(f*x)^4*tan(e)^3 -

900*B*a*b^2*d*tan(f*x)^4*tan(e)^3 - 300*A*b^3*d*tan(f*x)^4*tan(e)^3 + 300*C
*b^3*d*tan(f*x)^4*tan(e)^3 + 240*C*a^3*c*tan(f*x)^3*tan(e)^4 + 720*B*a^2*b*
c*tan(f*x)^3*tan(e)^4 + 720*A*a*b^2*c*tan(f*x)^3*tan(e)^4 - 900*C*a*b^2*c*t
an(f*x)^3*tan(e)^4 - 300*B*b^3*c*tan(f*x)^3*tan(e)^4 + 240*B*a^3*d*tan(f*x)
^3*tan(e)^4 + 720*A*a^2*b*d*tan(f*x)^3*tan(e)^4 - 900*C*a^2*b*d*tan(f*x)^3*
tan(e)^4 - 900*B*a*b^2*d*tan(f*x)^3*tan(e)^4 - 300*A*b^3*d*tan(f*x)^3*tan(e
)^4 + 300*C*b^3*d*tan(f*x)^3*tan(e)^4 - 60*C*a*b^2*c*tan(f*x)^2*tan(e)^5 -
20*B*b^3*c*tan(f*x)^2*tan(e)^5 - 60*C*a^2*b*d*tan(f*x)^2*tan(e)^5 - 60*B*a*
b^2*d*tan(f*x)^2*tan(e)^5 - 20*A*b^3*d*tan(f*x)^2*tan(e)^5 + 20*C*b^3*d*tan
(f*x)^2*tan(e)^5 + 15*C*b^3*c*tan(f*x)^5*tan(e) + 45*C*a*b^2*d*tan(f*x)^5*t
an(e) + 15*B*b^3*d*tan(f*x)^5*tan(e) - 600*A*a^3*c*f*x*tan(f*x)^2*tan(e)^2
+ 600*C*a^3*c*f*x*tan(f*x)^2*tan(e)^2 + 1800*B*a^2*b*c*f*x*tan(f*x)^2*tan(e
)^2 + 1800*A*a*b^2*c*f*x*tan(f*x)^2*tan(e)^2 - 1800*C*a*b^2*c*f*x*tan(f*x)^
2*tan(e)^2 - 600*B*b^3*c*f*x*tan(f*x)^2*tan(e)^2 + 600*B*a^3*d*f*x*tan(f*x)
^2*tan(e)^2 + 1800*A*a^2*b*d*f*x*tan(f*x)^2*tan(e)^2 - 1800*C*a^2*b*d*f*x*t
an(f*x)^2*tan(e)^2 - 1800*B*a*b^2*d*f*x*tan(f*x)^2*tan(e)^2 - 600*A*b^3*d*f
*x*tan(f*x)^2*tan(e)^2 + 600*C*b^3*d*f*x*tan(f*x)^2*tan(e)^2 - 270*C*a^2*b*
c*tan(f*x)^4*tan(e)^2 - 270*B*a*b^2*c*tan(f*x)^4*tan(e)^2 - 90*A*b^3*c*tan(
f*x)^4*tan(e)^2 + 150*C*b^3*c*tan(f*x)^4*tan(e)^2 - 90*C*a^3*d*tan(f*x)^4*t
an(e)^2 - 270*B*a^2*b*d*tan(f*x)^4*tan(e)^2 - 270*A*a*b^2*d*tan(f*x)^4*tan(
e)^2 + 450*C*a*b^2*d*tan(f*x)^4*tan(e)^2 + 150*B*b^3*d*tan(f*x)^4*tan(e)^2
+ 360*C*a^2*b*c*tan(f*x)^3*tan(e)^3 + 360*B*a*b^2*c*tan(f*x)^3*tan(e)^3 + 1
20*A*b^3*c*tan(f*x)^3*tan(e)^3 - 180*C*b^3*c*tan(f*x)^3*tan(e)^3 + 120*C*a^
3*d*tan(f*x)^3*tan(e)^3 + 360*B*a^2*b*d*tan(f*x)^3*tan(e)^3 + 360*A*a*b^2*d
*tan(f*x)^3*tan(e)^3 - 540*C*a*b^2*d*tan(f*x)^3*tan(e)^3 - 180*B*b^3*d*tan(
f*x)^3*tan(e)^3 - 270*C*a^2*b*c*tan(f*x)^2*tan(e)^4 - 270*B*a*b^2*c*tan(f*x
)^2*tan(e)^4 - 90*A*b^3*c*tan(f*x)^2*tan(e)^4 + 150*C*b^3*c*tan(f*x)^2*tan(
e)^4 - 90*C*a^3*d*tan(f*x)^2*tan(e)^4 - 270*B*a^2*b*d*tan(f*x)^2*tan(e)^4 -
270*A*a*b^2*d*tan(f*x)^2*tan(e)^4 + 450*C*a*b^2*d*tan(f*x)^2*tan(e)^4 + 15
0*B*b^3*d*tan(f*x)^2*tan(e)^4 + 15*C*b^3*c*tan(f*x)*tan(e)^5 + 45*C*a*b^2*d
*tan(f*x)*tan(e)^5 + 15*B*b^3*d*tan(f*x)*tan(e)^5 - 12*C*b^3*d*tan(f*x)^5 +
120*C*a*b^2*c*tan(f*x)^4*tan(e) + 40*B*b^3*c*tan(f*x)^4*tan(e) + 120*C*a^2
*b*d*tan(f*x)^4*tan(e) + 120*B*a*b^2*d*tan(f*x)^4*tan(e) + 40*A*b^3*d*tan(f
*x)^4*tan(e) - 100*C*b^3*d*tan(f*x)^4*tan(e) + 300*B*a^3*c*log(4*(tan(f*x)^
2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan
(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 900*A*a^2*b*c*log(4*(tan(f*x)^2*tan(e)^2
- 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))
*tan(f*x)^2*tan(e)^2 - 900*C*a^2*b*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x
)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2
*tan(e)^2 - 900*B*a*b^2*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) +
1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 -
300*A*b^3*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^
2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 300*C*b^3*c*
log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 +
tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 300*A*a^3*d*log(4*(tan(f*

x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 300*C*a^3*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 900*B*a^2*b*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 900*A*a*b^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 900*C*a*b^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 300*B*b^3*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 360*C*a^3*c*tan(f*x)^3*tan(e)^2 - 1080*B*a^2*b*c*tan(f*x)^3*tan(e)^2 - 1080*A*a*b^2*c*tan(f*x)^3*tan(e)^2 + 1440*C*a*b^2*c*tan(f*x)^3*tan(e)^2 + 480*B*b^3*c*tan(f*x)^3*tan(e)^2 - 360*B*a^3*d*tan(f*x)^3*tan(e)^2 - 1080*A*a^2*b*d*tan(f*x)^3*tan(e)^2 + 1440*C*a^2*b*d*tan(f*x)^3*tan(e)^2 + 1440*B*a*b^2*d*tan(f*x)^3*tan(e)^2 + 480*A*b^3*d*tan(f*x)^3*tan(e)^2 - 600*C*b^3*d*tan(f*x)^3*tan(e)^2 - 360*C*a^3*c*tan(f*x)^2*tan(e)^3 - 1080*B*a^2*b*c*tan(f*x)^2*tan(e)^3 - 1080*A*a*b^2*c*tan(f*x)^2*tan(e)^3 + 1440*C*a*b^2*c*tan(f*x)^2*tan(e)^3 + 480*B*b^3*c*tan(f*x)^2*tan(e)^3 - 360*B*a^3*d*tan(f*x)^2*tan(e)^3 - 1080*A*a^2*b*d*tan(f*x)^2*tan(e)^3 + 1440*C*a^2*b*d*tan(f*x)^2*tan(e)^3 + 1440*B*a*b^2*d*tan(f*x)^2*tan(e)^3 + 480*A*b^3*d*tan(f*x)^2*tan(e)^3 - 600*C*b^3*d*tan(f*x)^2*tan(e)^3 + 120*C*a*b^2*c*tan(f*x)*tan(e)^4 + 40*B*b^3*c*tan(f*x)*tan(e)^4 + 120*C*a^2*b*d*tan(f*x)*tan(e)^4 + 120*B*a*b^2*d*tan(f*x)*tan(e)^4 + 40*A*b^3*d*tan(f*x)*tan(e)^4 - 100*C*b^3*d*tan(f*x)*tan(e)^4 - 12*C*b^3*d*tan(e)^5 - 15*C*b^3*c*tan(f*x)^4 - 45*C*a*b^2*d*tan(f*x)^4 - 15*B*b^3*d*tan(f*x)^4 + 300*A*a^3*c*f*x*tan(f*x)*tan(e) - 300*C*a^3*c*f*x*tan(f*x)*tan(e) - 900*B*a^2*b*c*f*x*tan(f*x)*tan(e) - 900*A*a*b^2*c*f*x*tan(f*x)*tan(e) + 900*C*a*b^2*c*f*x*tan(f*x)*tan(e) + 300*B*b^3*c*f*x*tan(f*x)*tan(e) - 300*B*a^3*d*f*x*tan(f*x)*tan(e) - 900*A*a^2*b*d*f*x*tan(f*x)*tan(e) + 900*C*a^2*b*d*f*x*tan(f*x)*tan(e) + 900*B*a*b^2*d*f*x*tan(f*x)*tan(e) + 300*A*b^3*d*f*x*tan(f*x)*tan(e) - 300*C*b^3*d*f*x*tan(f*x)*tan(e) + 270*C*a^2*b*c*tan(f*x)^3*tan(e) + 270*B*a*b^2*c*tan(f*x)^3*tan(e) + 90*A*b^3*c*tan(f*x)^3*tan(e) - 150*C*b^3*c*tan(f*x)^3*tan(e) + 90*C*a^3*d*tan(f*x)^3*tan(e) + 270*B*a^2*b*d*tan(f*x)^3*tan(e) + 270*A*a*b^2*d*tan(f*x)^3*tan(e) - 450*C*a*b^2*d*tan(f*x)^3*tan(e) - 150*B*b^3*d*tan(f*x)^3*tan(e) - 360*C*a^2*b*c*tan(f*x)^2*tan(e)^2 - 360*B*a*b^2*c*tan(f*x)^2*tan(e)^2 - 120*A*b^3*c*tan(f*x)^2*tan(e)^2 + 180*C*b^3*c*tan(f*x)^2*tan(e)^2 - 120*C*a^3*d*tan(f*x)^2*tan(e)^2 - 360*B*a^2*b*d*tan(f*x)^2*tan(e)^2 - 360*A*a*b^2*d*tan(f*x)^2*tan(e)^2 + 540*C*a*b^2*d*tan(f*x)^2*tan(e)^2 + 180*B*b^3*d*tan(f*x)^2*tan(e)^2 + 270*C*a^2*b*c*tan(f*x)*tan(e)^3 + 270*B*a*b^2*c*tan(f*x)*tan(e)^3 + 90*A*b^3*c*tan(f*x)*tan(e)^3 - 150*C*b^3*c*tan(f*x)*tan(e)^3 + 90*C*a^3*d*tan(f*x)*tan(e)^3 + 270*B*a^2*b*d*tan(f*x)*tan(e)^3 + 270*A*a*b^2*d*tan(f*x)*tan(e)^3 - 450*C*a*b^2*d*tan(f*x)*tan(e)^3 - 150*B*b^3*d*tan(f*x)*tan(e)^3 - 15*C*b^3*c*tan(e)^4 - 45*C*a*b^2*d*tan(e)^4 - 15*B*b^3*d*tan(e)^4 - 60*C*a*b^2*c*tan(f*x)^3 - 20*B*b^3*c*tan(f*x)^3 - 60*C*a^2*b*d*tan(f*x)^3 - 60*B*a*b^2*d*tan(f*x)^3

- 20*A*b^3*d*tan(f*x)^3 + 20*C*b^3*d*tan(f*x)^3 - 150*B*a^3*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) - 450*A*a^2*b*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 450*C*a^2*b*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 450*B*a*b^2*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 150*A*b^3*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) - 150*C*b^3*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) - 150*A*a^3*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 150*C*a^3*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 450*B*a^2*b*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 450*A*a*b^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) - 450*C*a*b^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) - 150*B*b^3*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 240*C*a^3*c*tan(f*x)^2*tan(e) + 720*B*a^2*b*c*tan(f*x)^2*tan(e) + 720*A*a*b^2*c*tan(f*x)^2*tan(e) - 900*C*a*b^2*c*tan(f*x)^2*tan(e) - 300*B*b^3*c*tan(f*x)^2*tan(e) + 240*B*a^3*d*tan(f*x)^2*tan(e) + 720*A*a^2*b*d*tan(f*x)^2*tan(e) - 900*C*a^2*b*d*tan(f*x)^2*tan(e) - 900*B*a*b^2*d*tan(f*x)^2*tan(e) - 300*A*b^3*d*tan(f*x)^2*tan(e) + 300*C*b^3*d*tan(f*x)^2*tan(e) + 240*C*a^3*c*tan(f*x)*tan(e)^2 + 720*B*a^2*b*c*tan(f*x)*tan(e)^2 + 720*A*a*b^2*c*tan(f*x)*tan(e)^2 - 900*C*a*b^2*c*tan(f*x)*tan(e)^2 - 300*B*b^3*c*tan(f*x)*tan(e)^2 + 240*B*a^3*d*tan(f*x)*tan(e)^2 + 720*A*a^2*b*d*tan(f*x)*tan(e)^2 - 900*C*a^2*b*d*tan(f*x)*tan(e)^2 - 900*B*a*b^2*d*tan(f*x)*tan(e)^2 - 300*A*b^3*d*tan(f*x)*tan(e)^2 + 300*C*b^3*d*tan(f*x)*tan(e)^2 - 60*C*a*b^2*c*tan(e)^3 - 20*B*b^3*c*tan(e)^3 - 60*C*a^2*b*d*tan(e)^3 - 60*B*a*b^2*d*tan(e)^3 - 20*A*b^3*d*tan(e)^3 + 20*C*b^3*d*tan(e)^3 - 60*A*a^3*c*f*x + 60*C*a^3*c*f*x + 180*B*a^2*b*c*f*x + 180*A*a*b^2*c*f*x - 180*C*a*b^2*c*f*x - 60*B*b^3*c*f*x + 60*B*a^3*d*f*x + 180*A*a^2*b*d*f*x - 180*C*a^2*b*d*f*x - 180*B*a*b^2*d*f*x - 60*A*b^3*d*f*x + 60*C*b^3*d*f*x - 90*C*a^2*b*c*tan(f*x)^2 - 90*B*a*b^2*c*tan(f*x)^2 - 30*A*b^3*c*tan(f*x)^2 + 30*C*b^3*c*tan(f*x)^2 - 30*C*a^3*d*tan(f*x)^2 - 90*B*a^2*b*d*tan(f*x)^2 - 90*A*a*b^2*d*tan(f*x)^2 + 90*C*a*b^2*d*tan(f*x)^2 + 30*B*b^3*d*tan(f*x)^2 + 270*C*a^2*b*c*tan(f*x)*tan(e) + 270*B*a*b^2*c*tan(f*x)*tan(e) + 90*A*b^3*c*tan(f*x)*tan(e) - 165*C*b^3*c*tan(f*x)*tan(e) + 90*C*a^3*d*tan(f*x)*tan(e) + 270*B*a^2*b*d*tan(f*x)*tan(e) + 270*A*a*b^2*d*tan(f*x)*tan(e) - 495*C*a*b^2*d*tan(f*x)*tan(e) - 165*B*b^3*d*tan(f*x)*tan(e) - 90*C*a^2*b*c*tan(e)^2 - 90*B*a*b^2*c*tan(e)^2 - 30*A*b^3*c*tan(e)^2 + 30*C*b^3*c*tan(e)^2 - 30*C*a^3*d*tan(e)^2 - 90*B*a^2*b*

```
d*tan(e)^2 - 90*A*a*b^2*d*tan(e)^2 + 90*C*a*b^2*d*tan(e)^2 + 30*B*b^3*d*tan
(e)^2 + 30*B*a^3*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan
(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) + 90*A*a^2*b*c*log(4*(tan(f*
x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 +
tan(e)^2 + 1)) - 90*C*a^2*b*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e
) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 90*B*a*b^2*c*lo
g(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + ta
n(f*x)^2 + tan(e)^2 + 1)) - 30*A*b^3*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f
*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) + 30*C*b
^3*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)
^2 + tan(f*x)^2 + tan(e)^2 + 1)) + 30*A*a^3*d*log(4*(tan(f*x)^2*tan(e)^2 -
2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) -
 30*C*a^3*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2
*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 90*B*a^2*b*d*log(4*(tan(f*x)^2*ta
n(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^
2 + 1)) - 90*A*a*b^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/
(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) + 90*C*a*b^2*d*log(4*(ta
n(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^
2 + tan(e)^2 + 1)) + 30*B*b^3*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan
(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 60*C*a^3*c*ta
n(f*x) - 180*B*a^2*b*c*tan(f*x) - 180*A*a*b^2*c*tan(f*x) + 180*C*a*b^2*c*ta
n(f*x) + 60*B*b^3*c*tan(f*x) - 60*B*a^3*d*tan(f*x) - 180*A*a^2*b*d*tan(f*x)
 + 180*C*a^2*b*d*tan(f*x) + 180*B*a*b^2*d*tan(f*x) + 60*A*b^3*d*tan(f*x) -
60*C*b^3*d*tan(f*x) - 60*C*a^3*c*tan(e) - 180*B*a^2*b*c*tan(e) - 180*A*a*b^
2*c*tan(e) + 180*C*a*b^2*c*tan(e) + 60*B*b^3*c*tan(e) - 60*B*a^3*d*tan(e) -
 180*A*a^2*b*d*tan(e) + 180*C*a^2*b*d*tan(e) + 180*B*a*b^2*d*tan(e) + 60*A*
b^3*d*tan(e) - 60*C*b^3*d*tan(e) - 90*C*a^2*b*c - 90*B*a*b^2*c - 30*A*b^3*c
 + 45*C*b^3*c - 30*C*a^3*d - 90*B*a^2*b*d - 90*A*a*b^2*d + 135*C*a*b^2*d +
45*B*b^3*d)/(f*tan(f*x)^5*tan(e)^5 - 5*f*tan(f*x)^4*tan(e)^4 + 10*f*tan(f*x
)^3*tan(e)^3 - 10*f*tan(f*x)^2*tan(e)^2 + 5*f*tan(f*x)*tan(e) - f)
```

Mupad [B] (verification not implemented)

Time = 8.88 (sec) , antiderivative size = 477, normalized size of antiderivative = 1.35

$$\int (a+b\tan(e+fx))^3(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= x\left(A\,a^3\,c+A\,b^3\,d-B\,a^3\,d+B\,b^3\,c-C\,a^3\,c-C\,b^3\,d-3\,A\,a\,b^2\,c-3\,A\,a^2\,b\,d-3\,B\,a^2\,b\,c+3\,B\,a\,b^2\,d+3\,C\,a\,b^2\,c+3\,C\,a^2\,b\,d\right)+\frac{\tan(e+f\,x)^4\left(\frac{B\,b^3\,d}{4}+\frac{C\,b^3\,c}{4}+\frac{3\,C\,a\,b^2\,d}{4}\right)}{f}$$
$$+\frac{\tan(e+f\,x)^3\left(\frac{A\,b^3\,d}{3}+\frac{B\,b^3\,c}{3}-\frac{C\,b^3\,d}{3}+B\,a\,b^2\,d+C\,a\,b^2\,c+C\,a^2\,b\,d\right)}{f}$$
$$+\frac{\tan(e+f\,x)^2\left(\frac{A\,b^3\,c}{2}-\frac{B\,b^3\,d}{2}+\frac{C\,a^3\,d}{2}-\frac{C\,b^3\,c}{2}+\frac{3\,A\,a\,b^2\,d}{2}+\frac{3\,B\,a\,b^2\,c}{2}+\frac{3\,B\,a^2\,b\,d}{2}+\frac{3\,C\,a^2\,b\,c}{2}-\frac{3\,C\,a\,b^2\,d}{2}\right)}{f}$$
$$+\frac{\ln\left(\tan(e+f\,x)^2+1\right)\left(\frac{A\,a^3\,d}{2}-\frac{A\,b^3\,c}{2}+\frac{B\,a^3\,c}{2}+\frac{B\,b^3\,d}{2}-\frac{C\,a^3\,d}{2}+\frac{C\,b^3\,c}{2}+\frac{3\,A\,a^2\,b\,c}{2}-\frac{3\,A\,a\,b^2\,d}{2}-\frac{3\,B\,a\,b^2\,c}{2}-3\,I\right.}{f}$$
$$+\frac{\tan(e+f\,x)\,\left(B\,a^3\,d-A\,b^3\,d-B\,b^3\,c+C\,a^3\,c+C\,b^3\,d+3\,A\,a\,b^2\,c+3\,A\,a^2\,b\,d+3\,B\,a^2\,b\,c-3\,B\,a\,b^2\,d\right.}{f}$$
$$+\frac{C\,b^3\,d\tan(e+f\,x)^5}{5\,f}$$

```
[In] int((a + b*tan(e + f*x))^3*(c + d*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan
(e + f*x)^2),x)
```

```
[Out] x*(A*a^3*c + A*b^3*d - B*a^3*d + B*b^3*c - C*a^3*c - C*b^3*d - 3*A*a*b^2*c
- 3*A*a^2*b*d - 3*B*a^2*b*c + 3*B*a*b^2*d + 3*C*a*b^2*c + 3*C*a^2*b*d) + (t
an(e + f*x)^4*((B*b^3*d)/4 + (C*b^3*c)/4 + (3*C*a*b^2*d)/4))/f + (tan(e + f
*x)^3*((A*b^3*d)/3 + (B*b^3*c)/3 - (C*b^3*d)/3 + B*a*b^2*d + C*a*b^2*c + C*
a^2*b*d))/f + (tan(e + f*x)^2*((A*b^3*c)/2 - (B*b^3*d)/2 + (C*a^3*d)/2 - (C
*b^3*c)/2 + (3*A*a*b^2*d)/2 + (3*B*a*b^2*c)/2 + (3*B*a^2*b*d)/2 + (3*C*a^2*
b*c)/2 - (3*C*a*b^2*d)/2))/f + (log(tan(e + f*x)^2 + 1)*((A*a^3*d)/2 - (A*b
^3*c)/2 + (B*a^3*c)/2 + (B*b^3*d)/2 - (C*a^3*d)/2 + (C*b^3*c)/2 + (3*A*a^2*
b*c)/2 - (3*A*a*b^2*d)/2 - (3*B*a*b^2*c)/2 - (3*B*a^2*b*d)/2 - (3*C*a^2*b*c
)/2 + (3*C*a*b^2*d)/2))/f + (tan(e + f*x)*(B*a^3*d - A*b^3*d - B*b^3*c + C*
a^3*c + C*b^3*d + 3*A*a*b^2*c + 3*A*a^2*b*d + 3*B*a^2*b*c - 3*B*a*b^2*d - 3
*C*a*b^2*c - 3*C*a^2*b*d))/f + (C*b^3*d*tan(e + f*x)^5)/(5*f)
```

3.51 $\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))\left(A+B\tan(e+f\right.$

Optimal result 426
Rubi [A] (verified) 427
Mathematica [C] (verified) 429
Maple [A] (verified) 430
Fricas [A] (verification not implemented) 430
Sympy [B] (verification not implemented) 431
Maxima [A] (verification not implemented) 431
Giac [B] (verification not implemented) 432
Mupad [B] (verification not implemented) 436

Optimal result

Integrand size = 43, antiderivative size = 248

$$
\begin{aligned}
&\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx \\
&= \left(a^2(Ac-cC-Bd)-b^2(Ac-cC-Bd)-2ab(Bc+(A-C)d)\right)x \\
&\quad-\frac{(2ab(Ac-cC-Bd)+a^2(Bc+(A-C)d)-b^2(Bc+(A-C)d))\log(\cos(e+fx))}{f} \\
&\quad+\frac{b(Abc+aBc-bcC+aAd-bBd-aCd)\tan(e+fx)}{f} \\
&\quad+\frac{(Bc+(A-C)d)(a+b\tan(e+fx))^2}{2f} \\
&\quad-\frac{(aCd-4b(cC+Bd))(a+b\tan(e+fx))^3}{12b^2f}+\frac{Cd\tan(e+fx)(a+b\tan(e+fx))^3}{4bf}
\end{aligned}
$$

```
[Out] (a^2*(A*c-B*d-C*c)-b^2*(A*c-B*d-C*c)-2*a*b*(B*c+(A-C)*d))*x-(2*a*b*(A*c-B*d
-C*c)+a^2*(B*c+(A-C)*d)-b^2*(B*c+(A-C)*d))*ln(cos(f*x+e))/f+b*(A*a*d+A*b*c+
B*a*c-B*b*d-C*a*d-C*b*c)*tan(f*x+e)/f+1/2*(B*c+(A-C)*d)*(a+b*tan(f*x+e))^2/
f-1/12*(C*a*d-4*b*(B*d+C*c))*(a+b*tan(f*x+e))^3/b^2/f+1/4*C*d*tan(f*x+e)*(a
+b*tan(f*x+e))^3/b/f
```

Rubi [A] (verified)

Time = 0.50 (sec) , antiderivative size = 248, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}} = 0.116$, Rules used = {3718, 3711, 3609, 3606, 3556}

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=-\frac{\log(\cos(e+fx))\left(a^2(d(A-C)+Bc)+2ab(Ac-Bd-cC)-b^2(d(A-C)+Bc)\right)}{f}$$
$$+x\left(a^2(Ac-Bd-cC)-2ab(d(A-C)+Bc)-b^2(Ac-Bd-cC)\right)$$
$$+\frac{(d(A-C)+Bc)(a+b\tan(e+fx))^2}{2f}$$
$$+\frac{b\tan(e+fx)(aAd+aBc-aCd+Abc-bBd-bcC)}{f}$$
$$-\frac{(aCd-4b(Bd+cC))(a+b\tan(e+fx))^3}{12b^2f}+\frac{Cd\tan(e+fx)(a+b\tan(e+fx))^3}{4bf}$$

```
[In] Int[(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan
[e + f*x]^2),x]
```

```
[Out] (a^2*(A*c - c*C - B*d) - b^2*(A*c - c*C - B*d) - 2*a*b*(B*c + (A - C)*d))*x
 - ((2*a*b*(A*c - c*C - B*d) + a^2*(B*c + (A - C)*d) - b^2*(B*c + (A - C)*d
))*Log[Cos[e + f*x]])/f + (b*(A*b*c + a*B*c - b*c*C + a*A*d - b*B*d - a*C*d
)*Tan[e + f*x])/f + ((B*c + (A - C)*d)*(a + b*Tan[e + f*x])^2)/(2*f) - ((a*
C*d - 4*b*(c*C + B*d))*(a + b*Tan[e + f*x])^3)/(12*b^2*f) + (C*d*Tan[e + f*
x]*(a + b*Tan[e + f*x])^3)/(4*b*f)
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3606

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
 f*x], x], x] + Simp[b*d*(Tan[e + f*x]/f), x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3609

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[d*((a + b*Tan[e + f*x])^m/(f*m)), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
```

```
0] && GtQ[m, 0]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rubi steps

$$\text{integral} = \frac{Cd\tan(e+fx)(a+b\tan(e+fx))^3}{4bf} - \frac{\int (a+b\tan(e+fx))^2\,(-4Abc+aCd-4b(Bc+(A-C)d)\tan(e+fx)+(aCd-4b(cC+Bd))\tan^2(e}{4b}$$

$$= -\frac{(aCd-4b(cC+Bd))(a+b\tan(e+fx))^3}{12b^2f} + \frac{Cd\tan(e+fx)(a+b\tan(e+fx))^3}{4bf} - \frac{\int (a+b\tan(e+fx))^2(-4b(Ac-cC-Bd)-4b(Bc+(A-C)d)\tan(e+fx))\,dx}{4b}$$

$$= \frac{(Bc+(A-C)d)(a+b\tan(e+fx))^2}{2f} - \frac{(aCd-4b(cC+Bd))(a+b\tan(e+fx))^3}{12b^2f} + \frac{Cd\tan(e+fx)(a+b\tan(e+fx))^3}{4bf} - \frac{\int (a+b\tan(e+fx))(4b(bBc+b(A-C)d-a(Ac-cC-Bd))-4b(Abc+aBc-bcC+aAd-}{4b}$$

$$= \left(a^2(Ac-cC-Bd)-b^2(Ac-cC-Bd)-2ab(Bc+(A-C)d)\right)x + \frac{b(Abc+aBc-bcC+aAd-bBd-aCd)\tan(e+fx)}{f} + \frac{(Bc+(A-C)d)(a+b\tan(e+fx))^2}{2f} - \frac{(aCd-4b(cC+Bd))(a+b\tan(e+fx))^3}{12b^2f} + \frac{Cd\tan(e+fx)(a+b\tan(e+fx))^3}{4bf} - \left(-2ab(Ac-cC-Bd)-a^2(Bc+(A-C)d)+b^2(Bc+(A-C)d)\right)\int \tan(e+fx)\,dx$$

$$= \left(a^2(Ac-cC-Bd)-b^2(Ac-cC-Bd)-2ab(Bc+(A-C)d)\right)x - \frac{(2ab(Ac-cC-Bd)+a^2(Bc+(A-C)d)-b^2(Bc+(A-C)d))\log(\cos(e+fx))}{f} + \frac{b(Abc+aBc-bcC+aAd-bBd-aCd)\tan(e+fx)}{f} + \frac{(Bc+(A-C)d)(a+b\tan(e+fx))^2}{2f} - \frac{(aCd-4b(cC+Bd))(a+b\tan(e+fx))^3}{12b^2f} + \frac{Cd\tan(e+fx)(a+b\tan(e+fx))^3}{4bf}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 3.67 (sec) , antiderivative size = 243, normalized size of antiderivative = 0.98

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$

$$= \frac{\frac{(-aCd+4b(cC+Bd))(a+b\tan(e+fx))^3}{b} + 3Cd\tan(e+fx)(a+b\tan(e+fx))^3 - 6(Abc-aBc-bcC-aAd-bB}{}$$

```
[In] Integrate[(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])*(A + B*Tan[e + f*x] +
 C*Tan[e + f*x]^2),x]
```

```
[Out] (((-(a*C*d) + 4*b*(c*C + B*d))*(a + b*Tan[e + f*x])^3)/b + 3*C*d*Tan[e + f*
x]*(a + b*Tan[e + f*x])^3 - 6*(A*b*c - a*B*c - b*c*C - a*A*d - b*B*d + a*C*
d)*(I*((a + I*b)^2*Log[I - Tan[e + f*x]] - (a - I*b)^2*Log[I + Tan[e + f*x]
]) - 2*b^2*Tan[e + f*x]) + 6*(B*c + (A - C)*d)*((I*a - b)^3*Log[I - Tan[e +
 f*x]] - (I*a + b)^3*Log[I + Tan[e + f*x]] + 6*a*b^2*Tan[e + f*x] + b^3*Tan
[e + f*x]^2))/(12*b*f)
```

Maple [A] (verified)

Time = 0.15 (sec) , antiderivative size = 246, normalized size of antiderivative = 0.99

method	result
parts	$\frac{(A\,a^2d+2Aabc+B\,a^2c)\ln\left(1+\tan(fx+e)^2\right)}{2f}+\frac{(B\,b^2d+2Cabd+C\,b^2c)\left(\frac{\tan(fx+e)^3}{3}-\tan(fx+e)+\arctan(\tan(fx+e))\right)}{f}$
norman	$(A\,a^2c-2Aabd-A\,b^2c-B\,a^2d-2Babc+B\,b^2d-C\,a^2c+2Cabd+C\,b^2c)\,x+\frac{(2Aabd-}{}$
derivativedivides	$\frac{dC\,b^2\tan(fx+e)^4}{4}+\frac{B\,b^2d\tan(fx+e)^3}{3}+\frac{2Cabd\tan(fx+e)^3}{3}+\frac{C\,b^2c\tan(fx+e)^3}{3}+\frac{A\,b^2d\tan(fx+e)^2}{2}+Babd\tan(fx+e)^2+\frac{B\,b^2c\tan}{2}$
default	$\frac{dC\,b^2\tan(fx+e)^4}{4}+\frac{B\,b^2d\tan(fx+e)^3}{3}+\frac{2Cabd\tan(fx+e)^3}{3}+\frac{C\,b^2c\tan(fx+e)^3}{3}+\frac{A\,b^2d\tan(fx+e)^2}{2}+Babd\tan(fx+e)^2+\frac{B\,b^2c\tan}{2}$
parallelrisch	$\frac{-12\tan(fx+e)C\,b^2c+3dC\,b^2\tan(fx+e)^4+4B\,b^2d\tan(fx+e)^3+4C\,b^2c\tan(fx+e)^3+6A\,b^2d\tan(fx+e)^2+6B\,b^2c\tan(fx}{}$
risch	Expression too large to display

```
[In] int((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x,m
ethod=_RETURNVERBOSE)
```

```
[Out] 1/2*(A*a^2*d+2*A*a*b*c+B*a^2*c)/f*ln(1+tan(f*x+e)^2)+(B*b^2*d+2*C*a*b*d+C*b
^2*c)/f*(1/3*tan(f*x+e)^3-tan(f*x+e)+arctan(tan(f*x+e)))+(A*b^2*d+2*B*a*b*d
+B*b^2*c+C*a^2*d+2*C*a*b*c)/f*(1/2*tan(f*x+e)^2-1/2*ln(1+tan(f*x+e)^2))+(2*
A*a*b*d+A*b^2*c+B*a^2*d+2*B*a*b*c+C*a^2*c)/f*(tan(f*x+e)-arctan(tan(f*x+e))
)+A*a^2*c*x+d*C*b^2/f*(1/4*tan(f*x+e)^4-1/2*tan(f*x+e)^2+1/2*ln(1+tan(f*x+e
)^2))
```

Fricas [A] (verification not implemented)

none

Time = 0.25 (sec) , antiderivative size = 273, normalized size of antiderivative = 1.10

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\frac{3\,Cb^2d\tan\left(fx+e\right)^4+4\left(Cb^2c+(2\,Cab+Bb^2)d\right)\tan\left(fx+e\right)^3+12\left(((A-C)a^2-2\,Bab-(A-C)b^2)c\right.}{}$$

```
[In] integrate((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x, algorithm="fricas")
```

```
[Out] 1/12*(3*C*b^2*d*tan(f*x + e)^4 + 4*(C*b^2*c + (2*C*a*b + B*b^2)*d)*tan(f*x
+ e)^3 + 12*(((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*c - (B*a^2 + 2*(A - C)*a
*b - B*b^2)*d)*f*x + 6*((2*C*a*b + B*b^2)*c + (C*a^2 + 2*B*a*b + (A - C)*b^
```

```
2)*d)*tan(f*x + e)^2 - 6*((B*a^2 + 2*(A - C)*a*b - B*b^2)*c + ((A - C)*a^2
- 2*B*a*b - (A - C)*b^2)*d)*log(1/(tan(f*x + e)^2 + 1)) + 12*((C*a^2 + 2*B*
a*b + (A - C)*b^2)*c + (B*a^2 + 2*(A - C)*a*b - B*b^2)*d)*tan(f*x + e))/f
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 617 vs. $2(218) = 436$.

Time = 0.21 (sec) , antiderivative size = 617, normalized size of antiderivative = 2.49

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\begin{cases} Aa^2cx+\frac{Aa^2d\log\left(\tan^2(e+fx)+1\right)}{2f}+\frac{Aabc\log\left(\tan^2(e+fx)+1\right)}{f}-2Aabdx+\frac{2Aabd\tan(e+fx)}{f}-Ab^2cx+\frac{Ab^2c\tan(e+fx)}{f} \\ x(a+b\tan(e))^2\left(c+d\tan(e)\right)\left(A+B\tan(e)+C\tan^2(e)\right) \end{cases}$$

```
[In] integrate((a+b*tan(f*x+e))**2*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)
**2),x)
```

```
[Out] Piecewise((A*a**2*c*x + A*a**2*d*log(tan(e + f*x)**2 + 1)/(2*f) + A*a*b*c*l
og(tan(e + f*x)**2 + 1)/f - 2*A*a*b*d*x + 2*A*a*b*d*tan(e + f*x)/f - A*b**2
*c*x + A*b**2*c*tan(e + f*x)/f - A*b**2*d*log(tan(e + f*x)**2 + 1)/(2*f) +
A*b**2*d*tan(e + f*x)**2/(2*f) + B*a**2*c*log(tan(e + f*x)**2 + 1)/(2*f) -
B*a**2*d*x + B*a**2*d*tan(e + f*x)/f - 2*B*a*b*c*x + 2*B*a*b*c*tan(e + f*x)
/f - B*a*b*d*log(tan(e + f*x)**2 + 1)/f + B*a*b*d*tan(e + f*x)**2/f - B*b**
2*c*log(tan(e + f*x)**2 + 1)/(2*f) + B*b**2*c*tan(e + f*x)**2/(2*f) + B*b**
2*d*x + B*b**2*d*tan(e + f*x)**3/(3*f) - B*b**2*d*tan(e + f*x)/f - C*a**2*c
*x + C*a**2*c*tan(e + f*x)/f - C*a**2*d*log(tan(e + f*x)**2 + 1)/(2*f) + C*
a**2*d*tan(e + f*x)**2/(2*f) - C*a*b*c*log(tan(e + f*x)**2 + 1)/f + C*a*b*c
*tan(e + f*x)**2/f + 2*C*a*b*d*x + 2*C*a*b*d*tan(e + f*x)**3/(3*f) - 2*C*a*
b*d*tan(e + f*x)/f + C*b**2*c*x + C*b**2*c*tan(e + f*x)**3/(3*f) - C*b**2*c
*tan(e + f*x)/f + C*b**2*d*log(tan(e + f*x)**2 + 1)/(2*f) + C*b**2*d*tan(e
+ f*x)**4/(4*f) - C*b**2*d*tan(e + f*x)**2/(2*f), Ne(f, 0)), (x*(a + b*tan(
e))**2*(c + d*tan(e))*(A + B*tan(e) + C*tan(e)**2), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.38 (sec) , antiderivative size = 274, normalized size of antiderivative = 1.10

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\frac{3\,Cb^2d\tan(fx+e)^4+4\,(Cb^2c+(2\,Cab+Bb^2)d)\tan(fx+e)^3+6\,((2\,Cab+Bb^2)c+(Ca^2+2\,Bab+(}{}$$

```
[In] integrate((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x, algorithm="maxima")
```

```
[Out] 1/12*(3*C*b^2*d*tan(f*x + e)^4 + 4*(C*b^2*c + (2*C*a*b + B*b^2)*d)*tan(f*x
+ e)^3 + 6*((2*C*a*b + B*b^2)*c + (C*a^2 + 2*B*a*b + (A - C)*b^2)*d)*tan(f*
x + e)^2 + 12*(((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*c - (B*a^2 + 2*(A - C)
*a*b - B*b^2)*d)*(f*x + e) + 6*((B*a^2 + 2*(A - C)*a*b - B*b^2)*c + ((A - C
)*a^2 - 2*B*a*b - (A - C)*b^2)*d)*log(tan(f*x + e)^2 + 1) + 12*((C*a^2 + 2*
B*a*b + (A - C)*b^2)*c + (B*a^2 + 2*(A - C)*a*b - B*b^2)*d)*tan(f*x + e))/f
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 5631 vs. 2(242) = 484.

Time = 4.13 (sec) , antiderivative size = 5631, normalized size of antiderivative = 22.71

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= \text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x, algorithm="giac")
```

```
[Out] 1/12*(12*A*a^2*c*f*x*tan(f*x)^4*tan(e)^4 - 12*C*a^2*c*f*x*tan(f*x)^4*tan(e)
^4 - 24*B*a*b*c*f*x*tan(f*x)^4*tan(e)^4 - 12*A*b^2*c*f*x*tan(f*x)^4*tan(e)^
4 + 12*C*b^2*c*f*x*tan(f*x)^4*tan(e)^4 - 12*B*a^2*d*f*x*tan(f*x)^4*tan(e)^4
 - 24*A*a*b*d*f*x*tan(f*x)^4*tan(e)^4 + 24*C*a*b*d*f*x*tan(f*x)^4*tan(e)^4
+ 12*B*b^2*d*f*x*tan(f*x)^4*tan(e)^4 - 6*B*a^2*c*log(4*(tan(f*x)^2*tan(e)^2
 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)
)*tan(f*x)^4*tan(e)^4 - 12*A*a*b*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*
tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*t
an(e)^4 + 12*C*a*b*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(t
an(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 6*B*
b^2*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e
)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 - 6*A*a^2*d*log(4*(ta
n(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^
2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 6*C*a^2*d*log(4*(tan(f*x)^2*tan(e)
^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 +
1))*tan(f*x)^4*tan(e)^4 + 12*B*a*b*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x
)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4
*tan(e)^4 + 6*A*b^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(
tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 - 6*C
*b^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(
e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 - 48*A*a^2*c*f*x*tan
(f*x)^3*tan(e)^3 + 48*C*a^2*c*f*x*tan(f*x)^3*tan(e)^3 + 96*B*a*b*c*f*x*tan(
f*x)^3*tan(e)^3 + 48*A*b^2*c*f*x*tan(f*x)^3*tan(e)^3 - 48*C*b^2*c*f*x*tan(f
```

*x)^3*tan(e)^3 + 48*B*a^2*d*f*x*tan(f*x)^3*tan(e)^3 + 96*A*a*b*d*f*x*tan(f*x)^3*tan(e)^3 - 96*C*a*b*d*f*x*tan(f*x)^3*tan(e)^3 - 48*B*b^2*d*f*x*tan(f*x)^3*tan(e)^3 + 12*C*a*b*c*tan(f*x)^4*tan(e)^4 + 6*B*b^2*c*tan(f*x)^4*tan(e)^4 + 6*C*a^2*d*tan(f*x)^4*tan(e)^4 + 12*B*a*b*d*tan(f*x)^4*tan(e)^4 + 6*A*b^2*d*tan(f*x)^4*tan(e)^4 - 9*C*b^2*d*tan(f*x)^4*tan(e)^4 + 24*B*a^2*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 48*A*a*b*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 48*C*a*b*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 24*B*b^2*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 24*A*a^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 24*C*a^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 48*B*a*b*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 24*A*b^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 24*C*b^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 12*C*a^2*c*tan(f*x)^4*tan(e)^3 - 24*B*a*b*c*tan(f*x)^4*tan(e)^3 - 12*A*b^2*c*tan(f*x)^4*tan(e)^3 + 12*C*b^2*c*tan(f*x)^4*tan(e)^3 - 12*B*a^2*d*tan(f*x)^4*tan(e)^3 - 24*A*a*b*d*tan(f*x)^4*tan(e)^3 + 24*C*a*b*d*tan(f*x)^4*tan(e)^3 + 12*B*b^2*d*tan(f*x)^4*tan(e)^3 - 12*C*a^2*c*tan(f*x)^3*tan(e)^4 - 24*B*a*b*c*tan(f*x)^3*tan(e)^4 - 12*A*b^2*c*tan(f*x)^3*tan(e)^4 + 12*C*b^2*c*tan(f*x)^3*tan(e)^4 - 12*B*a^2*d*tan(f*x)^3*tan(e)^4 - 24*A*a*b*d*tan(f*x)^3*tan(e)^4 + 24*C*a*b*d*tan(f*x)^3*tan(e)^4 + 12*B*b^2*d*tan(f*x)^3*tan(e)^4 + 72*A*a^2*c*f*x*tan(f*x)^2*tan(e)^2 - 72*C*a^2*c*f*x*tan(f*x)^2*tan(e)^2 - 144*B*a*b*c*f*x*tan(f*x)^2*tan(e)^2 - 72*A*b^2*c*f*x*tan(f*x)^2*tan(e)^2 + 72*C*b^2*c*f*x*tan(f*x)^2*tan(e)^2 - 72*B*a^2*d*f*x*tan(f*x)^2*tan(e)^2 - 144*A*a*b*d*f*x*tan(f*x)^2*tan(e)^2 + 144*C*a*b*d*f*x*tan(f*x)^2*tan(e)^2 + 72*B*b^2*d*f*x*tan(f*x)^2*tan(e)^2 + 12*C*a*b*c*tan(f*x)^4*tan(e)^2 + 6*B*b^2*c*tan(f*x)^4*tan(e)^2 + 6*C*a^2*d*tan(f*x)^4*tan(e)^2 + 12*B*a*b*d*tan(f*x)^4*tan(e)^2 + 6*A*b^2*d*tan(f*x)^4*tan(e)^2 - 6*C*b^2*d*tan(f*x)^4*tan(e)^2 - 24*C*a*b*c*tan(f*x)^3*tan(e)^3 - 12*B*b^2*c*tan(f*x)^3*tan(e)^3 - 12*C*a^2*d*tan(f*x)^3*tan(e)^3 - 24*B*a*b*d*tan(f*x)^3*tan(e)^3 - 12*A*b^2*d*tan(f*x)^3*tan(e)^3 + 24*C*b^2*d*tan(f*x)^3*tan(e)^3 + 12*C*a*b*c*tan(f*x)^2*tan(e)^4 + 6*B*b^2*c*tan(f*x)^2*tan(e)^4 + 6*C*a^2*d*tan(f*x)^2*tan(e)^4 + 12*B*a*b*d*tan(f*x)^2*tan(e)^4 + 6*A*b^2*d*tan(f*x)^2*tan(e)^4 - 6*C*b^2*d*tan(f*x)^2*tan(e)^4 - 4*C*b^2*c*tan(f*x)^4*tan(e) - 8*C*a*b*d*tan(f*x)^4*tan(e) - 4*B*b^2*d*tan(f*x)^4*tan(e) - 36*B*a^2*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 72*A*a*b*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*ta

n(f*x)^2*tan(e)^2 + 72*C*a*b*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(
e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e
)^2 + 36*B*b^2*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f
*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 36*A*a^2
*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2
 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 36*C*a^2*d*log(4*(tan(
f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2
+ tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 72*B*a*b*d*log(4*(tan(f*x)^2*tan(e)^
2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1
))*tan(f*x)^2*tan(e)^2 + 36*A*b^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)
*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*
tan(e)^2 - 36*C*b^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(
tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 36*
C*a^2*c*tan(f*x)^3*tan(e)^2 + 72*B*a*b*c*tan(f*x)^3*tan(e)^2 + 36*A*b^2*c*t
an(f*x)^3*tan(e)^2 - 48*C*b^2*c*tan(f*x)^3*tan(e)^2 + 36*B*a^2*d*tan(f*x)^3
*tan(e)^2 + 72*A*a*b*d*tan(f*x)^3*tan(e)^2 - 96*C*a*b*d*tan(f*x)^3*tan(e)^2
 - 48*B*b^2*d*tan(f*x)^3*tan(e)^2 + 36*C*a^2*c*tan(f*x)^2*tan(e)^3 + 72*B*a
*b*c*tan(f*x)^2*tan(e)^3 + 36*A*b^2*c*tan(f*x)^2*tan(e)^3 - 48*C*b^2*c*tan(
f*x)^2*tan(e)^3 + 36*B*a^2*d*tan(f*x)^2*tan(e)^3 + 72*A*a*b*d*tan(f*x)^2*ta
n(e)^3 - 96*C*a*b*d*tan(f*x)^2*tan(e)^3 - 48*B*b^2*d*tan(f*x)^2*tan(e)^3 -
4*C*b^2*c*tan(f*x)*tan(e)^4 - 8*C*a*b*d*tan(f*x)*tan(e)^4 - 4*B*b^2*d*tan(f
*x)*tan(e)^4 + 3*C*b^2*d*tan(f*x)^4 - 48*A*a^2*c*f*x*tan(f*x)*tan(e) + 48*C
*a^2*c*f*x*tan(f*x)*tan(e) + 96*B*a*b*c*f*x*tan(f*x)*tan(e) + 48*A*b^2*c*f*
x*tan(f*x)*tan(e) - 48*C*b^2*c*f*x*tan(f*x)*tan(e) + 48*B*a^2*d*f*x*tan(f*x
)*tan(e) + 96*A*a*b*d*f*x*tan(f*x)*tan(e) - 96*C*a*b*d*f*x*tan(f*x)*tan(e)
- 48*B*b^2*d*f*x*tan(f*x)*tan(e) - 24*C*a*b*c*tan(f*x)^3*tan(e) - 12*B*b^2*
c*tan(f*x)^3*tan(e) - 12*C*a^2*d*tan(f*x)^3*tan(e) - 24*B*a*b*d*tan(f*x)^3*
tan(e) - 12*A*b^2*d*tan(f*x)^3*tan(e) + 24*C*b^2*d*tan(f*x)^3*tan(e) + 24*C
*a*b*c*tan(f*x)^2*tan(e)^2 + 12*B*b^2*c*tan(f*x)^2*tan(e)^2 + 12*C*a^2*d*ta
n(f*x)^2*tan(e)^2 + 24*B*a*b*d*tan(f*x)^2*tan(e)^2 + 12*A*b^2*d*tan(f*x)^2*
tan(e)^2 - 12*C*b^2*d*tan(f*x)^2*tan(e)^2 - 24*C*a*b*c*tan(f*x)*tan(e)^3 -
12*B*b^2*c*tan(f*x)*tan(e)^3 - 12*C*a^2*d*tan(f*x)*tan(e)^3 - 24*B*a*b*d*ta
n(f*x)*tan(e)^3 - 12*A*b^2*d*tan(f*x)*tan(e)^3 + 24*C*b^2*d*tan(f*x)*tan(e)
^3 + 3*C*b^2*d*tan(e)^4 + 4*C*b^2*c*tan(f*x)^3 + 8*C*a*b*d*tan(f*x)^3 + 4*B
*b^2*d*tan(f*x)^3 + 24*B*a^2*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(
e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e)
+ 48*A*a*b*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^
2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) - 48*C*a*b*c*log(4
*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f
*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) - 24*B*b^2*c*log(4*(tan(f*x)^2*tan(e
)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 +
 1))*tan(f*x)*tan(e) + 24*A*a^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*t
an(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(
e) - 24*C*a^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*
x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) - 48*B*a*b*d*lo

```
g(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + ta
n(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) - 24*A*b^2*d*log(4*(tan(f*x)^2*ta
n(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^
2 + 1))*tan(f*x)*tan(e) + 24*C*b^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x
)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*t
an(e) - 36*C*a^2*c*tan(f*x)^2*tan(e) - 72*B*a*b*c*tan(f*x)^2*tan(e) - 36*A*
b^2*c*tan(f*x)^2*tan(e) + 48*C*b^2*c*tan(f*x)^2*tan(e) - 36*B*a^2*d*tan(f*x
)^2*tan(e) - 72*A*a*b*d*tan(f*x)^2*tan(e) + 96*C*a*b*d*tan(f*x)^2*tan(e) +
48*B*b^2*d*tan(f*x)^2*tan(e) - 36*C*a^2*c*tan(f*x)*tan(e)^2 - 72*B*a*b*c*ta
n(f*x)*tan(e)^2 - 36*A*b^2*c*tan(f*x)*tan(e)^2 + 48*C*b^2*c*tan(f*x)*tan(e)
^2 - 36*B*a^2*d*tan(f*x)*tan(e)^2 - 72*A*a*b*d*tan(f*x)*tan(e)^2 + 96*C*a*b
*d*tan(f*x)*tan(e)^2 + 48*B*b^2*d*tan(f*x)*tan(e)^2 + 4*C*b^2*c*tan(e)^3 +
8*C*a*b*d*tan(e)^3 + 4*B*b^2*d*tan(e)^3 + 12*A*a^2*c*f*x - 12*C*a^2*c*f*x -
 24*B*a*b*c*f*x - 12*A*b^2*c*f*x + 12*C*b^2*c*f*x - 12*B*a^2*d*f*x - 24*A*a
*b*d*f*x + 24*C*a*b*d*f*x + 12*B*b^2*d*f*x + 12*C*a*b*c*tan(f*x)^2 + 6*B*b^
2*c*tan(f*x)^2 + 6*C*a^2*d*tan(f*x)^2 + 12*B*a*b*d*tan(f*x)^2 + 6*A*b^2*d*t
an(f*x)^2 - 6*C*b^2*d*tan(f*x)^2 - 24*C*a*b*c*tan(f*x)*tan(e) - 12*B*b^2*c*
tan(f*x)*tan(e) - 12*C*a^2*d*tan(f*x)*tan(e) - 24*B*a*b*d*tan(f*x)*tan(e) -
 12*A*b^2*d*tan(f*x)*tan(e) + 24*C*b^2*d*tan(f*x)*tan(e) + 12*C*a*b*c*tan(e
)^2 + 6*B*b^2*c*tan(e)^2 + 6*C*a^2*d*tan(e)^2 + 12*B*a*b*d*tan(e)^2 + 6*A*b
^2*d*tan(e)^2 - 6*C*b^2*d*tan(e)^2 - 6*B*a^2*c*log(4*(tan(f*x)^2*tan(e)^2 -
 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))
- 12*A*a*b*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^
2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) + 12*C*a*b*c*log(4*(tan(f*x)^2*tan
(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2
 + 1)) + 6*B*b^2*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan
(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 6*A*a^2*d*log(4*(tan(f*x)^
2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan
(e)^2 + 1)) + 6*C*a^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)
/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) + 12*B*a*b*d*log(4*(tan
(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2
 + tan(e)^2 + 1)) + 6*A*b^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e
) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 6*C*b^2*d*log(4
*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f
*x)^2 + tan(e)^2 + 1)) + 12*C*a^2*c*tan(f*x) + 24*B*a*b*c*tan(f*x) + 12*A*b
^2*c*tan(f*x) - 12*C*b^2*c*tan(f*x) + 12*B*a^2*d*tan(f*x) + 24*A*a*b*d*tan(
f*x) - 24*C*a*b*d*tan(f*x) - 12*B*b^2*d*tan(f*x) + 12*C*a^2*c*tan(e) + 24*B
*a*b*c*tan(e) + 12*A*b^2*c*tan(e) - 12*C*b^2*c*tan(e) + 12*B*a^2*d*tan(e) +
 24*A*a*b*d*tan(e) - 24*C*a*b*d*tan(e) - 12*B*b^2*d*tan(e) + 12*C*a*b*c + 6
*B*b^2*c + 6*C*a^2*d + 12*B*a*b*d + 6*A*b^2*d - 9*C*b^2*d)/(f*tan(f*x)^4*ta
n(e)^4 - 4*f*tan(f*x)^3*tan(e)^3 + 6*f*tan(f*x)^2*tan(e)^2 - 4*f*tan(f*x)*t
an(e) + f)
```

Mupad [B] (verification not implemented)

Time = 8.99 (sec) , antiderivative size = 300, normalized size of antiderivative = 1.21

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= \frac{\tan(e+f\,x)^2\left(\frac{A\,b^2\,d}{2}+\frac{B\,b^2\,c}{2}+\frac{C\,a^2\,d}{2}-\frac{C\,b^2\,d}{2}+B\,a\,b\,d+C\,a\,b\,c\right)}{f}$$
$$-x\left(A\,b^2\,c-A\,a^2\,c+B\,a^2\,d+C\,a^2\,c-B\,b^2\,d-C\,b^2\,c+2\,A\,a\,b\,d+2\,B\,a\,b\,c-2\,C\,a\,b\,d\right)$$
$$-\frac{\ln\left(\tan(e+f\,x)^2+1\right)\,\left(\frac{A\,b^2\,d}{2}-\frac{B\,a^2\,c}{2}-\frac{A\,a^2\,d}{2}+\frac{B\,b^2\,c}{2}+\frac{C\,a^2\,d}{2}-\frac{C\,b^2\,d}{2}-A\,a\,b\,c+B\,a\,b\,d+C\,a\,b\,c\right)}{f}$$
$$+\frac{\tan(e+f\,x)\,\left(A\,b^2\,c+B\,a^2\,d+C\,a^2\,c-B\,b^2\,d-C\,b^2\,c+2\,A\,a\,b\,d+2\,B\,a\,b\,c-2\,C\,a\,b\,d\right)}{f}$$
$$+\frac{\tan(e+f\,x)^3\left(\frac{B\,b^2\,d}{3}+\frac{C\,b^2\,c}{3}+\frac{2\,C\,a\,b\,d}{3}\right)}{f}+\frac{C\,b^2\,d\tan(e+f\,x)^4}{4\,f}$$

```
[In] int((a + b*tan(e + f*x))^2*(c + d*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan
(e + f*x)^2),x)
```

```
[Out] (tan(e + f*x)^2*((A*b^2*d)/2 + (B*b^2*c)/2 + (C*a^2*d)/2 - (C*b^2*d)/2 + B*
a*b*d + C*a*b*c))/f - x*(A*b^2*c - A*a^2*c + B*a^2*d + C*a^2*c - B*b^2*d -
C*b^2*c + 2*A*a*b*d + 2*B*a*b*c - 2*C*a*b*d) - (log(tan(e + f*x)^2 + 1)*((A
*b^2*d)/2 - (B*a^2*c)/2 - (A*a^2*d)/2 + (B*b^2*c)/2 + (C*a^2*d)/2 - (C*b^2*
d)/2 - A*a*b*c + B*a*b*d + C*a*b*c))/f + (tan(e + f*x)*(A*b^2*c + B*a^2*d +
 C*a^2*c - B*b^2*d - C*b^2*c + 2*A*a*b*d + 2*B*a*b*c - 2*C*a*b*d))/f + (tan
(e + f*x)^3*((B*b^2*d)/3 + (C*b^2*c)/3 + (2*C*a*b*d)/3))/f + (C*b^2*d*tan(e
 + f*x)^4)/(4*f)
```

3.52 $\int (a+b\tan(e+fx))(c+d\tan(e+fx))\left(A+B\tan(e+f$

Optimal result 437
Rubi [A] (verified) 437
Mathematica [C] (verified) 439
Maple [A] (verified) 440
Fricas [A] (verification not implemented) 440
Sympy [B] (verification not implemented) 441
Maxima [A] (verification not implemented) 441
Giac [B] (verification not implemented) 442
Mupad [B] (verification not implemented) 444

Optimal result

Integrand size = 41, antiderivative size = 161

$$\begin{aligned}\int (a+b\tan(e+fx))(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx\\ =(a(Ac-cC-Bd)-b(Bc+(A-C)d))x\\ -\frac{(Abc+aBc-bcC+aAd-bBd-aCd)\log(\cos(e+fx))}{f}\\ +\frac{(Ab+aB-bC)d\tan(e+fx)}{f}-\frac{(bcC-3bBd-3aCd)(c+d\tan(e+fx))^2}{6d^2f}\\ +\frac{bC\tan(e+fx)(c+d\tan(e+fx))^2}{3df}\end{aligned}$$

```
[Out] (a*(A*c-B*d-C*c)-b*(B*c+(A-C)*d))*x-(A*a*d+A*b*c+B*a*c-B*b*d-C*a*d-C*b*c)*l
n(cos(f*x+e))/f+(A*b+B*a-C*b)*d*tan(f*x+e)/f-1/6*(-3*B*b*d-3*C*a*d+C*b*c)*(
c+d*tan(f*x+e))^2/d^2/f+1/3*b*C*tan(f*x+e)*(c+d*tan(f*x+e))^2/d/f
```

Rubi [A] (verified)

Time = 0.28 (sec) , antiderivative size = 161, normalized size of antiderivative = 1.00, number of steps used = 4, number of rules used = 4, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.098, Rules used = {3718, 3711, 3606, 3556}

$$\begin{aligned}\int (a+b\tan(e+fx))(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx\\ =-\frac{\log(\cos(e+fx))(aAd+aBc-aCd+Abc-bBd-bcC)}{f}\\ -x(-a(Ac-Bd-cC)+bd(A-C)+bBc)+\frac{d\tan(e+fx)(aB+Ab-bC)}{f}\\ -\frac{(-3aCd-3bBd+bcC)(c+d\tan(e+fx))^2}{6d^2f}+\frac{bC\tan(e+fx)(c+d\tan(e+fx))^2}{3df}\end{aligned}$$

```
[In] Int[(a + b*Tan[e + f*x])*(c + d*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e
 + f*x]^2),x]
```

```
[Out] -((b*B*c + b*(A - C)*d - a*(A*c - c*C - B*d))*x) - ((A*b*c + a*B*c - b*c*C
+ a*A*d - b*B*d - a*C*d)*Log[Cos[e + f*x]])/f + ((A*b + a*B - b*C)*d*Tan[e
+ f*x])/f - ((b*c*C - 3*b*B*d - 3*a*C*d)*(c + d*Tan[e + f*x])^2)/(6*d^2*f)
+ (b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^2)/(3*d*f)
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3606

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
 f*x], x], x] + Simp[b*d*(Tan[e + f*x]/f), x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rubi steps

$$\text{integral} = \frac{bC\tan(e+fx)(c+d\tan(e+fx))^2}{3df} - \frac{\int(c+d\tan(e+fx))\left(bcC-3aAd-3(Ab+aB-bC)d\tan(e+fx)+(bcC-3bBd-3aCd)\tan^2(e+\right.}{3d}$$

$$= -\frac{(bcC-3bBd-3aCd)(c+d\tan(e+fx))^2}{6d^2f}+\frac{bC\tan(e+fx)(c+d\tan(e+fx))^2}{3df} - \frac{\int(c+d\tan(e+fx))(3(bB-a(A-C))d-3(Ab+aB-bC)d\tan(e+fx))\,dx}{3d}$$

$$= -((bBc+b(A-C)d-a(Ac-cC-Bd))x)+\frac{(Ab+aB-bC)d\tan(e+fx)}{f} - \frac{(bcC-3bBd-3aCd)(c+d\tan(e+fx))^2}{6d^2f} + \frac{bC\tan(e+fx)(c+d\tan(e+fx))^2}{3df} - (-aBc+bcC+bBd+aCd-A(bc+ad))\int\tan(e+fx)\,dx$$

$$= -((bBc+b(A-C)d-a(Ac-cC-Bd))x) - \frac{(aBc-bcC-bBd-aCd+A(bc+ad))\log(\cos(e+fx))}{f} + \frac{(Ab+aB-bC)d\tan(e+fx)}{f} - \frac{(bcC-3bBd-3aCd)(c+d\tan(e+fx))^2}{6d^2f} + \frac{bC\tan(e+fx)(c+d\tan(e+fx))^2}{3df}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 1.69 (sec) , antiderivative size = 161, normalized size of antiderivative = 1.00

$$\int(a+b\tan(e+fx))(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \frac{3(a+ib)(A+iB-C)(-ic+d)\log(i-\tan(e+fx))+3(a-ib)(A-iB-C)(ic+d)\log(i+\tan(e+f}{6f}$$

```
[In] Integrate[(a + b*Tan[e + f*x])*(c + d*Tan[e + f*x])*(A + B*Tan[e + f*x] + C
*Tan[e + f*x]^2),x]
```

```
[Out] (3*(a + I*b)*(A + I*B - C)*((-I)*c + d)*Log[I - Tan[e + f*x]] + 3*(a - I*b)
*(A - I*B - C)*(I*c + d)*Log[I + Tan[e + f*x]] + 6*(A*b + a*B - b*C)*d*Tan[
e + f*x] + ((-(b*c*C) + 3*b*B*d + 3*a*C*d)*(c + d*Tan[e + f*x])^2)/d^2 + (2
*b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^2)/d)/(6*f)
```

Maple [A] (verified)

Time = 0.08 (sec) , antiderivative size = 147, normalized size of antiderivative = 0.91

method	result
parts	$\frac{(Aad+Abc+Bac)\ln\left(1+\tan(fx+e)^2\right)}{2f}+\frac{(bdB+Cad+Cbc)\left(\frac{\tan(fx+e)^2}{2}-\frac{\ln\left(1+\tan(fx+e)^2\right)}{2}\right)}{f}+\frac{(Abd+Bad+Bbc+Ca}{}$
norman	$(Aac-Abd-Bad-Bbc-Cac+Cbd)\,x+\frac{(Abd+Bad+Bbc+Cac-Cbd)\tan(fx+e)}{f}+\frac{(bdB+Cad+}{}$
derivativedivides	$\frac{C\tan(fx+e)^3bd}{3}+\frac{B\tan(fx+e)^2bd}{2}+\frac{C\tan(fx+e)^2ad}{2}+\frac{C\tan(fx+e)^2bc}{2}+A\tan(fx+e)bd+B\tan(fx+e)ad+B\tan(fx+e)bc+C$
default	$\frac{C\tan(fx+e)^3bd}{3}+\frac{B\tan(fx+e)^2bd}{2}+\frac{C\tan(fx+e)^2ad}{2}+\frac{C\tan(fx+e)^2bc}{2}+A\tan(fx+e)bd+B\tan(fx+e)ad+B\tan(fx+e)bc+C$
parallelrisch	$\frac{2C\tan(fx+e)^3bd+6Aacfx-6Abdfx-6Badfx-6Bbcfx+3B\tan(fx+e)^2bd-6Cacfx+6Cbdfx+3C\tan(fx+e)^2ad+3C\,\mathrm{ta}}{}$
risch	$-Badx+Cbdx+\frac{2iBace}{f}-\frac{2iBbde}{f}-\frac{2iCade}{f}-\frac{2iCbce}{f}+\frac{2iAade}{f}+\frac{2iAbce}{f}+\frac{2i\left(-3iCad\,e^{2i(fx+e)}-3\right.}{}$

```
[In] int((a+b*tan(f*x+e))*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x,met
hod=_RETURNVERBOSE)
```

```
[Out] 1/2*(A*a*d+A*b*c+B*a*c)/f*ln(1+tan(f*x+e)^2)+(B*b*d+C*a*d+C*b*c)/f*(1/2*tan
(f*x+e)^2-1/2*ln(1+tan(f*x+e)^2))+(A*b*d+B*a*d+B*b*c+C*a*c)/f*(tan(f*x+e)-a
rctan(tan(f*x+e)))+A*a*c*x+C*b*d/f*(1/3*tan(f*x+e)^3-tan(f*x+e)+arctan(tan(
f*x+e)))
```

Fricas [A] (verification not implemented)

none

Time = 0.26 (sec) , antiderivative size = 150, normalized size of antiderivative = 0.93

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\frac{2\,Cbd\tan{(fx+e)}^3+6\,(((A-C)a-Bb)c-(Ba+(A-C)b)d)fx+3\,(Cbc+(Ca+Bb)d)\tan{(fx+e)}}{}$$

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)
,x, algorithm="fricas")
```

```
[Out] 1/6*(2*C*b*d*tan(f*x + e)^3 + 6*(((A - C)*a - B*b)*c - (B*a + (A - C)*b)*d)
*f*x + 3*(C*b*c + (C*a + B*b)*d)*tan(f*x + e)^2 - 3*((B*a + (A - C)*b)*c +
((A - C)*a - B*b)*d)*log(1/(tan(f*x + e)^2 + 1)) + 6*((C*a + B*b)*c + (B*a
+ (A - C)*b)*d)*tan(f*x + e))/f
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 326 vs. $2(148) = 296$.

Time = 0.15 (sec) , antiderivative size = 326, normalized size of antiderivative = 2.02

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= \begin{cases} Aacx + \frac{Aad\log\left(\tan^2(e+fx)+1\right)}{2f} + \frac{Abc\log\left(\tan^2(e+fx)+1\right)}{2f} - Abdx + \frac{Abd\tan(e+fx)}{f} + \frac{Bac\log\left(\tan^2(e+fx)+1\right)}{2f} - Badx \\ x(a+b\tan(e))(c+d\tan(e))\left(A+B\tan(e)+C\tan^2(e)\right) \end{cases}$$

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)**2
),x)
```

```
[Out] Piecewise((A*a*c*x + A*a*d*log(tan(e + f*x)**2 + 1)/(2*f) + A*b*c*log(tan(e
 + f*x)**2 + 1)/(2*f) - A*b*d*x + A*b*d*tan(e + f*x)/f + B*a*c*log(tan(e +
f*x)**2 + 1)/(2*f) - B*a*d*x + B*a*d*tan(e + f*x)/f - B*b*c*x + B*b*c*tan(e
 + f*x)/f - B*b*d*log(tan(e + f*x)**2 + 1)/(2*f) + B*b*d*tan(e + f*x)**2/(2
*f) - C*a*c*x + C*a*c*tan(e + f*x)/f - C*a*d*log(tan(e + f*x)**2 + 1)/(2*f)
 + C*a*d*tan(e + f*x)**2/(2*f) - C*b*c*log(tan(e + f*x)**2 + 1)/(2*f) + C*b
*c*tan(e + f*x)**2/(2*f) + C*b*d*x + C*b*d*tan(e + f*x)**3/(3*f) - C*b*d*ta
n(e + f*x)/f, Ne(f, 0)), (x*(a + b*tan(e))*(c + d*tan(e))*(A + B*tan(e) + C
*tan(e)**2), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.40 (sec) , antiderivative size = 151, normalized size of antiderivative = 0.94

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= \frac{2\,Cbd\tan(fx+e)^3 + 3\,(Cbc + (Ca+Bb)d)\tan(fx+e)^2 + 6\,(((A-C)a-Bb)c - (Ba+(A-C)b)d)(}{}$$

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)
,x, algorithm="maxima")
```

```
[Out] 1/6*(2*C*b*d*tan(f*x + e)^3 + 3*(C*b*c + (C*a + B*b)*d)*tan(f*x + e)^2 + 6*
(((A - C)*a - B*b)*c - (B*a + (A - C)*b)*d)*(f*x + e) + 3*((B*a + (A - C)*b
)*c + ((A - C)*a - B*b)*d)*log(tan(f*x + e)^2 + 1) + 6*((C*a + B*b)*c + (B*
a + (A - C)*b)*d)*tan(f*x + e))/f
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 2475 vs. $2(157) = 314$.

Time = 1.79 (sec) , antiderivative size = 2475, normalized size of antiderivative = 15.37

$$\int (a + b\tan(e + fx))(c + d\tan(e + fx)) \left(A + B\tan(e + fx) + C\tan^2(e + fx)\right) dx$$
$$= \text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)
,x, algorithm="giac")
```

```
[Out] 1/6*(6*A*a*c*f*x*tan(f*x)^3*tan(e)^3 - 6*C*a*c*f*x*tan(f*x)^3*tan(e)^3 - 6*
B*b*c*f*x*tan(f*x)^3*tan(e)^3 - 6*B*a*d*f*x*tan(f*x)^3*tan(e)^3 - 6*A*b*d*f
*x*tan(f*x)^3*tan(e)^3 + 6*C*b*d*f*x*tan(f*x)^3*tan(e)^3 - 3*B*a*c*log(4*(t
an(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)
^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 3*A*b*c*log(4*(tan(f*x)^2*tan(e)^
2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1
))*tan(f*x)^3*tan(e)^3 + 3*C*b*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*ta
n(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan
(e)^3 - 3*A*a*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*
x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 3*C*a*d*l
og(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + t
an(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 3*B*b*d*log(4*(tan(f*x)^2*
tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e
)^2 + 1))*tan(f*x)^3*tan(e)^3 - 18*A*a*c*f*x*tan(f*x)^2*tan(e)^2 + 18*C*a*c
*f*x*tan(f*x)^2*tan(e)^2 + 18*B*b*c*f*x*tan(f*x)^2*tan(e)^2 + 18*B*a*d*f*x*
tan(f*x)^2*tan(e)^2 + 18*A*b*d*f*x*tan(f*x)^2*tan(e)^2 - 18*C*b*d*f*x*tan(f
*x)^2*tan(e)^2 + 3*C*b*c*tan(f*x)^3*tan(e)^3 + 3*C*a*d*tan(f*x)^3*tan(e)^3
+ 3*B*b*d*tan(f*x)^3*tan(e)^3 + 9*B*a*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(
f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x
)^2*tan(e)^2 + 9*A*b*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/
(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 9*
C*b*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e
)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 9*A*a*d*log(4*(tan(
f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2
+ tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 9*C*a*d*log(4*(tan(f*x)^2*tan(e)^2 -
 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*
tan(f*x)^2*tan(e)^2 - 9*B*b*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e
) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)
^2 - 6*C*a*c*tan(f*x)^3*tan(e)^2 - 6*B*b*c*tan(f*x)^3*tan(e)^2 - 6*B*a*d*ta
n(f*x)^3*tan(e)^2 - 6*A*b*d*tan(f*x)^3*tan(e)^2 + 6*C*b*d*tan(f*x)^3*tan(e)
^2 - 6*C*a*c*tan(f*x)^2*tan(e)^3 - 6*B*b*c*tan(f*x)^2*tan(e)^3 - 6*B*a*d*ta
n(f*x)^2*tan(e)^3 - 6*A*b*d*tan(f*x)^2*tan(e)^3 + 6*C*b*d*tan(f*x)^2*tan(e)
^3 + 18*A*a*c*f*x*tan(f*x)*tan(e) - 18*C*a*c*f*x*tan(f*x)*tan(e) - 18*B*b*c
```

```
*f*x*tan(f*x)*tan(e) - 18*B*a*d*f*x*tan(f*x)*tan(e) - 18*A*b*d*f*x*tan(f*x)
*tan(e) + 18*C*b*d*f*x*tan(f*x)*tan(e) + 3*C*b*c*tan(f*x)^3*tan(e) + 3*C*a*
d*tan(f*x)^3*tan(e) + 3*B*b*d*tan(f*x)^3*tan(e) - 3*C*b*c*tan(f*x)^2*tan(e)
^2 - 3*C*a*d*tan(f*x)^2*tan(e)^2 - 3*B*b*d*tan(f*x)^2*tan(e)^2 + 3*C*b*c*ta
n(f*x)*tan(e)^3 + 3*C*a*d*tan(f*x)*tan(e)^3 + 3*B*b*d*tan(f*x)*tan(e)^3 - 2
*C*b*d*tan(f*x)^3 - 9*B*a*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e)
+ 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) - 9
*A*b*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(
e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 9*C*b*c*log(4*(tan(f*x
)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + t
an(e)^2 + 1))*tan(f*x)*tan(e) - 9*A*a*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(
f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x
)*tan(e) + 9*C*a*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan
(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 9*B*b*d*lo
g(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + ta
n(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 12*C*a*c*tan(f*x)^2*tan(e) + 12
*B*b*c*tan(f*x)^2*tan(e) + 12*B*a*d*tan(f*x)^2*tan(e) + 12*A*b*d*tan(f*x)^2
*tan(e) - 18*C*b*d*tan(f*x)^2*tan(e) + 12*C*a*c*tan(f*x)*tan(e)^2 + 12*B*b*
c*tan(f*x)*tan(e)^2 + 12*B*a*d*tan(f*x)*tan(e)^2 + 12*A*b*d*tan(f*x)*tan(e)
^2 - 18*C*b*d*tan(f*x)*tan(e)^2 - 2*C*b*d*tan(e)^3 - 6*A*a*c*f*x + 6*C*a*c*
f*x + 6*B*b*c*f*x + 6*B*a*d*f*x + 6*A*b*d*f*x - 6*C*b*d*f*x - 3*C*b*c*tan(f
*x)^2 - 3*C*a*d*tan(f*x)^2 - 3*B*b*d*tan(f*x)^2 + 3*C*b*c*tan(f*x)*tan(e) +
 3*C*a*d*tan(f*x)*tan(e) + 3*B*b*d*tan(f*x)*tan(e) - 3*C*b*c*tan(e)^2 - 3*C
*a*d*tan(e)^2 - 3*B*b*d*tan(e)^2 + 3*B*a*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*t
an(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) + 3*
A*b*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e
)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 3*C*b*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*
tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) + 3
*A*a*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(
e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 3*C*a*d*log(4*(tan(f*x)^2*tan(e)^2 - 2
*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) -
3*B*b*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan
(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 6*C*a*c*tan(f*x) - 6*B*b*c*tan(f*x) -
 6*B*a*d*tan(f*x) - 6*A*b*d*tan(f*x) + 6*C*b*d*tan(f*x) - 6*C*a*c*tan(e) -
6*B*b*c*tan(e) - 6*B*a*d*tan(e) - 6*A*b*d*tan(e) + 6*C*b*d*tan(e) - 3*C*b*c
 - 3*C*a*d - 3*B*b*d)/(f*tan(f*x)^3*tan(e)^3 - 3*f*tan(f*x)^2*tan(e)^2 + 3*
f*tan(f*x)*tan(e) - f)
```

Mupad [B] (verification not implemented)

Time = 8.42 (sec) , antiderivative size = 153, normalized size of antiderivative = 0.95

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\frac{\ln\left(\tan(e+f\,x)^2+1\right)\,\left(\frac{A\,a\,d}{2}+\frac{A\,b\,c}{2}+\frac{B\,a\,c}{2}-\frac{B\,b\,d}{2}-\frac{C\,a\,d}{2}-\frac{C\,b\,c}{2}\right)}{f}$$
$$-x\,(A\,b\,d-A\,a\,c+B\,a\,d+B\,b\,c+C\,a\,c-C\,b\,d)+\frac{\tan(e+f\,x)^2\,\left(\frac{B\,b\,d}{2}+\frac{C\,a\,d}{2}+\frac{C\,b\,c}{2}\right)}{f}$$
$$+\frac{\tan(e+f\,x)\,\left(A\,b\,d+B\,a\,d+B\,b\,c+C\,a\,c-C\,b\,d\right)}{f}+\frac{C\,b\,d\tan(e+f\,x)^3}{3\,f}$$

```
[In] int((a + b*tan(e + f*x))*(c + d*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e
 + f*x)^2),x)
```

```
[Out] (log(tan(e + f*x)^2 + 1)*((A*a*d)/2 + (A*b*c)/2 + (B*a*c)/2 - (B*b*d)/2 - (
C*a*d)/2 - (C*b*c)/2))/f - x*(A*b*d - A*a*c + B*a*d + B*b*c + C*a*c - C*b*d
) + (tan(e + f*x)^2*((B*b*d)/2 + (C*a*d)/2 + (C*b*c)/2))/f + (tan(e + f*x)*
(A*b*d + B*a*d + B*b*c + C*a*c - C*b*d))/f + (C*b*d*tan(e + f*x)^3)/(3*f)
```

3.53 $\int (c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+$

Optimal result 445
Rubi [A] (verified) 445
Mathematica [A] (verified) 446
Maple [A] (verified) 447
Fricas [A] (verification not implemented) 447
Sympy [B] (verification not implemented) 448
Maxima [A] (verification not implemented) 448
Giac [B] (verification not implemented) 448
Mupad [B] (verification not implemented) 449

Optimal result

Integrand size = 31, antiderivative size = 73

$$\begin{aligned}\int (c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx\\ =(Ac-cC-Bd)x-\frac{(Bc+(A-C)d)\log(\cos(e+fx))}{f}\\ +\frac{Bd\tan(e+fx)}{f}+\frac{C(c+d\tan(e+fx))^2}{2df}\end{aligned}$$

```
[Out] (A*c-B*d-C*c)*x-(B*c+(A-C)*d)*ln(cos(f*x+e))/f+B*d*tan(f*x+e)/f+1/2*C*(c+d*
tan(f*x+e))^2/d/f
```

Rubi [A] (verified)

Time = 0.06 (sec) , antiderivative size = 73, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 3, $\frac{\text{number of rules}}{\text{integrand size}} = 0.097$, Rules used = {3711, 3606, 3556}

$$\begin{aligned}\int (c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx\\ =-\frac{(d(A-C)+Bc)\log(\cos(e+fx))}{f}+x(Ac-Bd-cC)\\ +\frac{Bd\tan(e+fx)}{f}+\frac{C(c+d\tan(e+fx))^2}{2df}\end{aligned}$$

```
[In] Int[(c + d*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] (A*c - c*C - B*d)*x - ((B*c + (A - C)*d)*Log[Cos[e + f*x]])/f + (B*d*Tan[e
+ f*x])/f + (C*(c + d*Tan[e + f*x])^2)/(2*d*f)
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3606

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
 f*x], x], x] + Simp[b*d*(Tan[e + f*x]/f), x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \frac{C(c+d\tan(e+fx))^2}{2df} + \int (A-C+B\tan(e+fx))(c+d\tan(e+fx))\,dx \\
&= (Ac-cC-Bd)x + \frac{Bd\tan(e+fx)}{f} + \frac{C(c+d\tan(e+fx))^2}{2df} \\
&\quad + (Bc+(A-C)d)\int \tan(e+fx)\,dx \\
&= (Ac-cC-Bd)x - \frac{(Bc+(A-C)d)\log(\cos(e+fx))}{f} \\
&\quad + \frac{Bd\tan(e+fx)}{f} + \frac{C(c+d\tan(e+fx))^2}{2df}
\end{aligned}$$

Mathematica [A] (verified)

Time = 0.52 (sec) , antiderivative size = 76, normalized size of antiderivative = 1.04

$$\int (c+d\tan(e+fx)) \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$

$$= \frac{2Acfx - 2(cC+Bd)\arctan(\tan(e+fx)) - 2(Bc+(A-C)d)\log(\cos(e+fx)) + 2(cC+Bd)\tan(e+f}{2f}$$

```
[In] Integrate[(c + d*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] (2*A*c*f*x - 2*(c*C + B*d)*ArcTan[Tan[e + f*x]] - 2*(B*c + (A - C)*d)*Log[C
os[e + f*x]] + 2*(c*C + B*d)*Tan[e + f*x] + C*d*Tan[e + f*x]^2)/(2*f)
```

Maple [A] (verified)

Time = 0.05 (sec) , antiderivative size = 75, normalized size of antiderivative = 1.03

method	result
norman	$(Ac-Bd-Cc)\,x+\frac{(Bd+Cc)\tan(fx+e)}{f}+\frac{Cd\tan(fx+e)^{2}}{2f}+\frac{(Ad+Bc-Cd)\ln\left(1+\tan(fx+e)^{2}\right)}{2f}$
derivativedivides	$\frac{\frac{C\tan(fx+e)^{2}d}{2}+B\tan(fx+e)d+C\tan(fx+e)c+\frac{(Ad+Bc-Cd)\ln\left(1+\tan(fx+e)^{2}\right)}{2}+(Ac-Bd-Cc)\arctan(\tan(fx+e))}{f}$
default	$\frac{\frac{C\tan(fx+e)^{2}d}{2}+B\tan(fx+e)d+C\tan(fx+e)c+\frac{(Ad+Bc-Cd)\ln\left(1+\tan(fx+e)^{2}\right)}{2}+(Ac-Bd-Cc)\arctan(\tan(fx+e))}{f}$
parts	$Acx+\frac{(Ad+Bc)\ln\left(1+\tan(fx+e)^{2}\right)}{2f}+\frac{(Bd+Cc)(\tan(fx+e)-\arctan(\tan(fx+e)))}{f}+\frac{Cd\left(\frac{\tan(fx+e)^{2}}{2}-\frac{\ln(1+\tan}{}\right.}{f}$
parallelrisch	$\frac{2Acfx-2Bdfx-2Ccfx+C\tan(fx+e)^{2}d+A\ln\left(1+\tan(fx+e)^{2}\right)d+B\ln\left(1+\tan(fx+e)^{2}\right)c+2B\tan(fx+e)d-C\ln(1+}{2f}$
risch	$\frac{2iAde}{f}-iCdx+iAdx+Acx-Bdx-Ccx-\frac{2iCde}{f}+iBcx+\frac{2iBce}{f}+\frac{2i(-iCd\,e^{2i(fx+e)}+Bd}{f(}$

```
[In] int((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x,method=_RETURNVERBOS
E)
```

```
[Out] (A*c-B*d-C*c)*x+(B*d+C*c)/f*tan(f*x+e)+1/2*C*d/f*tan(f*x+e)^2+1/2*(A*d+B*c-
C*d)/f*ln(1+tan(f*x+e)^2)
```

Fricas [A] (verification not implemented)

none

Time = 0.24 (sec) , antiderivative size = 74, normalized size of antiderivative = 1.01

$$\int (c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\frac{Cd\tan\left(fx+e\right)^{2}+2\left((A-C)c-Bd\right)fx-\left(Bc+(A-C)d\right)\log\left(\frac{1}{\tan(fx+e)^{2}+1}\right)+2\left(Cc+Bd\right)\tan\left(fx+\right.}{2\,f}$$

```
[In] integrate((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorithm="fr
icas")
```

```
[Out] 1/2*(C*d*tan(f*x + e)^2 + 2*((A - C)*c - B*d)*f*x - (B*c + (A - C)*d)*log(1
/(tan(f*x + e)^2 + 1)) + 2*(C*c + B*d)*tan(f*x + e))/f
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 131 vs. $2(60) = 120$.

Time = 0.12 (sec) , antiderivative size = 131, normalized size of antiderivative = 1.79

$$\int (c+d\tan(e+fx)) \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right) \, dx$$
$$= \begin{cases} Acx + \frac{Ad\log\left(\tan^2(e+fx)+1\right)}{2f} + \frac{Bc\log\left(\tan^2(e+fx)+1\right)}{2f} - Bdx + \frac{Bd\tan(e+fx)}{f} - Ccx + \frac{Cc\tan(e+fx)}{f} - \frac{Cd\log\left(\tan^2(e+ \right.}{2f} \\ x(c+d\tan(e))\left(A+B\tan(e)+C\tan^2(e)\right) \end{cases}$$

```
[In] integrate((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)**2),x)
```

```
[Out] Piecewise((A*c*x + A*d*log(tan(e + f*x)**2 + 1)/(2*f) + B*c*log(tan(e + f*x
)**2 + 1)/(2*f) - B*d*x + B*d*tan(e + f*x)/f - C*c*x + C*c*tan(e + f*x)/f -
 C*d*log(tan(e + f*x)**2 + 1)/(2*f) + C*d*tan(e + f*x)**2/(2*f), Ne(f, 0)),
 (x*(c + d*tan(e))*(A + B*tan(e) + C*tan(e)**2), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.39 (sec) , antiderivative size = 74, normalized size of antiderivative = 1.01

$$\int (c+d\tan(e+fx)) \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right) \, dx$$
$$= \frac{Cd\tan(fx+e)^2 + 2\left((A-C)c-Bd\right)(fx+e) + (Bc+(A-C)d)\log\left(\tan(fx+e)^2+1\right) + 2\left(Cc+Bd\right)}{2f}$$

```
[In] integrate((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorithm="ma
xima")
```

```
[Out] 1/2*(C*d*tan(f*x + e)^2 + 2*((A - C)*c - B*d)*(f*x + e) + (B*c + (A - C)*d)
*log(tan(f*x + e)^2 + 1) + 2*(C*c + B*d)*tan(f*x + e))/f
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 761 vs. $2(71) = 142$.

Time = 0.78 (sec) , antiderivative size = 761, normalized size of antiderivative = 10.42

$$\int (c+d\tan(e+fx)) \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right) \, dx = \text{Too large to display}$$

```
[In] integrate((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorithm="gi
ac")
```

```
[Out] 1/2*(2*A*c*f*x*tan(f*x)^2*tan(e)^2 - 2*C*c*f*x*tan(f*x)^2*tan(e)^2 - 2*B*d*
f*x*tan(f*x)^2*tan(e)^2 - B*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e
) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)
^2 - A*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*ta
n(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + C*d*log(4*(tan(f
*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 +
 tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 4*A*c*f*x*tan(f*x)*tan(e) + 4*C*c*f*x
*tan(f*x)*tan(e) + 4*B*d*f*x*tan(f*x)*tan(e) + C*d*tan(f*x)^2*tan(e)^2 + 2*
B*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^
2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 2*A*d*log(4*(tan(f*x)^2*t
an(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)
^2 + 1))*tan(f*x)*tan(e) - 2*C*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*ta
n(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e
) - 2*C*c*tan(f*x)^2*tan(e) - 2*B*d*tan(f*x)^2*tan(e) - 2*C*c*tan(f*x)*tan(
e)^2 - 2*B*d*tan(f*x)*tan(e)^2 + 2*A*c*f*x - 2*C*c*f*x - 2*B*d*f*x + C*d*ta
n(f*x)^2 + C*d*tan(e)^2 - B*c*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e
) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - A*d*log(4*(tan(
f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2
+ tan(e)^2 + 1)) + C*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/
(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) + 2*C*c*tan(f*x) + 2*B*d
*tan(f*x) + 2*C*c*tan(e) + 2*B*d*tan(e) + C*d)/(f*tan(f*x)^2*tan(e)^2 - 2*f
*tan(f*x)*tan(e) + f)
```

Mupad [B] (verification not implemented)

Time = 8.70 (sec) , antiderivative size = 75, normalized size of antiderivative = 1.03

$$\int (c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \frac{\tan(e+f\,x)\,\left(B\,d+C\,c\right)}{f}-x\,\left(B\,d-A\,c+C\,c\right) + \frac{\ln\left(\tan(e+f\,x)^2+1\right)\,\left(\frac{A\,d}{2}+\frac{B\,c}{2}-\frac{C\,d}{2}\right)}{f}+\frac{C\,d\tan(e+f\,x)^2}{2\,f}$$

```
[In] int((c + d*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)^2),x)
```

```
[Out] (tan(e + f*x)*(B*d + C*c))/f - x*(B*d - A*c + C*c) + (log(tan(e + f*x)^2 +
1)*((A*d)/2 + (B*c)/2 - (C*d)/2))/f + (C*d*tan(e + f*x)^2)/(2*f)
```

3.54 $\int \frac{(c+d\tan(e+fx))(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx$

Optimal result 450
Rubi [A] (verified) 450
Mathematica [C] (verified) 452
Maple [A] (verified) 453
Fricas [A] (verification not implemented) 453
Sympy [C] (verification not implemented) 454
Maxima [A] (verification not implemented) 455
Giac [A] (verification not implemented) 456
Mupad [B] (verification not implemented) 456

Optimal result

Integrand size = 43, antiderivative size = 156

$$\begin{aligned}&\int \frac{(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{a+b\tan(e+fx)}\,dx\\&=\frac{(a(Ac-cC-Bd)+b(Bc+(A-C)d))x}{a^2+b^2}\\&\quad+\frac{(Abc-aBc-bcC-aAd-bBd+aCd)\log(\cos(e+fx))}{(a^2+b^2)\,f}\\&\quad+\frac{(Ab^2-a(bB-aC))\,(bc-ad)\log(a+b\tan(e+fx))}{b^2\,(a^2+b^2)\,f}+\frac{Cd\tan(e+fx)}{bf}\end{aligned}$$

```
[Out] (a*(A*c-B*d-C*c)+b*(B*c+(A-C)*d))*x/(a^2+b^2)+(-A*a*d+A*b*c-B*a*c-B*b*d+C*a
*d-C*b*c)*ln(cos(f*x+e))/(a^2+b^2)/f+(A*b^2-a*(B*b-C*a))*(-a*d+b*c)*ln(a+b*
tan(f*x+e))/b^2/(a^2+b^2)/f+C*d*tan(f*x+e)/b/f
```

Rubi [A] (verified)

Time = 0.39 (sec) , antiderivative size = 155, normalized size of antiderivative = 0.99, number of steps used = 5, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.116, Rules used = {3718, 3707, 3698, 31, 3556}

$$\begin{aligned}&\int \frac{(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{a+b\tan(e+fx)}\,dx\\&=\frac{(bc-ad)\left(Ab^2-a(bB-aC)\right)\log(a+b\tan(e+fx))}{b^2f\,(a^2+b^2)}\\&\quad+\frac{\log(\cos(e+fx))(-aAd-aBc+aCd+Abc-bBd-bcC)}{f\,(a^2+b^2)}\\&\quad+\frac{x(a(Ac-Bd-cC)+bd(A-C)+bBc)}{a^2+b^2}+\frac{Cd\tan(e+fx)}{bf}\end{aligned}$$

```
[In] Int[((c + d*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a + b*T
an[e + f*x]),x]
```

```
[Out] ((b*B*c + b*(A - C)*d + a*(A*c - c*C - B*d))*x)/(a^2 + b^2) + ((A*b*c - a*B
*c - b*c*C - a*A*d - b*B*d + a*C*d)*Log[Cos[e + f*x]])/((a^2 + b^2)*f) + ((
A*b^2 - a*(b*B - a*C))*(b*c - a*d)*Log[a + b*Tan[e + f*x]])/(b^2*(a^2 + b^2
)*f) + (C*d*Tan[e + f*x])/(b*f)
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3698

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3707

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[(a*A + b*B - a
*C)*(x/(a^2 + b^2)), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rubi steps

$$\text{integral}=\frac{Cd\tan(e+fx)}{bf}-\frac{\int\frac{-Abc+aCd-b(Bc+(A-C)d)\tan(e+fx)-(bcC+bBd-aCd)\tan^2(e+fx)}{a+b\tan(e+fx)}\,dx}{b}$$
$$=\frac{(bBc+b(A-C)d+a(Ac-cC-Bd))x}{a^2+b^2}+\frac{Cd\tan(e+fx)}{bf}+\frac{\left(\left(Ab^2-a(bB-aC)\right)(bc-ad)\right)\int\frac{1+\tan^2(e+fx)}{a+b\tan(e+fx)}\,dx}{b\left(a^2+b^2\right)}-\frac{(Abc-aBc-bcC-aAd-bBd+aCd)\int\tan(e+fx)\,dx}{a^2+b^2}$$
$$=\frac{(bBc+b(A-C)d+a(Ac-cC-Bd))x}{a^2+b^2}+\frac{(Abc-aBc-bcC-aAd-bBd+aCd)\log(\cos(e+fx))}{\left(a^2+b^2\right)f}+\frac{Cd\tan(e+fx)}{bf}+\frac{\left(\left(Ab^2-a(bB-aC)\right)(bc-ad)\right)\mathrm{Subst}\left(\int\frac{1}{a+x}\,dx,x,b\tan(e+fx)\right)}{b^2\left(a^2+b^2\right)f}$$
$$=\frac{(bBc+b(A-C)d+a(Ac-cC-Bd))x}{a^2+b^2}+\frac{(Abc-aBc-bcC-aAd-bBd+aCd)\log(\cos(e+fx))}{\left(a^2+b^2\right)f}+\frac{\left(Ab^2-a(bB-aC)\right)(bc-ad)\log(a+b\tan(e+fx))}{b^2\left(a^2+b^2\right)f}+\frac{Cd\tan(e+fx)}{bf}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 1.20 (sec) , antiderivative size = 148, normalized size of antiderivative = 0.95

$$\int\frac{(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{a+b\tan(e+fx)}\,dx$$
$$=\frac{\frac{(A+iB-C)(-ic+d)\log(i-\tan(e+fx))}{a+ib}+\frac{(A-iB-C)(ic+d)\log(i+\tan(e+fx))}{a-ib}+\frac{2\left(Ab^2+a(-bB+aC)\right)(bc-ad)\log(a+b\tan(e+fx))}{b^2(a^2+b^2)}+\frac{2Cd\,\mathrm{t}}{}}{2f}$$

```
[In] Integrate[((c + d*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a
 + b*Tan[e + f*x]),x]
```

```
[Out] (((A + I*B - C)*((-I)*c + d)*Log[I - Tan[e + f*x]])/(a + I*b) + ((A - I*B -
 C)*(I*c + d)*Log[I + Tan[e + f*x]])/(a - I*b) + (2*(A*b^2 + a*(-(b*B) + a*
C))*(b*c - a*d)*Log[a + b*Tan[e + f*x]])/(b^2*(a^2 + b^2)) + (2*C*d*Tan[e +
 f*x])/b)/(2*f)
```

Maple [A] (verified)

Time = 0.15 (sec) , antiderivative size = 173, normalized size of antiderivative = 1.11

method	result
derivativedivides	$\frac{\frac{\tan(fx+e)Cd}{b}+\frac{\left(-Aa\,b^{2}d+A\,b^{3}c+B\,a^{2}bd-Ba\,b^{2}c-a^{3}Cd+C\,a^{2}bc\right)\ln(a+b\tan(fx+e))}{b^{2}\left(a^{2}+b^{2}\right)}+\frac{\frac{(Aad-Abc+Bac+bdB-Cad+Cbc)\ln\left(1+\tan\right.}{2}}{}}{f}$
default	$\frac{\frac{\tan(fx+e)Cd}{b}+\frac{\left(-Aa\,b^{2}d+A\,b^{3}c+B\,a^{2}bd-Ba\,b^{2}c-a^{3}Cd+C\,a^{2}bc\right)\ln(a+b\tan(fx+e))}{b^{2}\left(a^{2}+b^{2}\right)}+\frac{\frac{(Aad-Abc+Bac+bdB-Cad+Cbc)\ln\left(1+\tan\right.}{2}}{}}{f}$
norman	$\frac{(Aac+Abd-Bad+Bbc-Cac-Cbd)x}{a^{2}+b^{2}}+\frac{Cd\tan(fx+e)}{bf}+\frac{(Aad-Abc+Bac+bdB-Cad+Cbc)\ln\left(1+\tan(fx+e)^{2}\right)}{2(a^{2}+b^{2})f}-$
parallelrisch	$2Aa\,b^{2}cfx+2A\,b^{3}dfx-2Ba\,b^{2}dfx+2B\,b^{3}cfx-2Ca\,b^{2}cfx-2C\,b^{3}dfx+A\ln\left(1+\tan(fx+e)^{2}\right)a\,b^{2}d-A\ln\left(1+\tan(fx+e)^{2}\right.$
risch	$-\frac{2iB\,a^{2}de}{bf(a^{2}+b^{2})}+\frac{2ia^{3}Cde}{b^{2}f(a^{2}+b^{2})}-\frac{2iC\,a^{2}ce}{bf(a^{2}+b^{2})}-\frac{xAc}{ib-a}+\frac{xBd}{ib-a}+\frac{xCc}{ib-a}-\frac{2iCade}{b^{2}f}+\frac{2iAade}{f(a^{2}+b^{2})}-\frac{2ibAce}{f(a^{2}+b^{2})}-$

```
[In] int((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e)),x,met
hod=_RETURNVERBOSE)
```

```
[Out] 1/f*(tan(f*x+e)*C*d/b+(-A*a*b^2*d+A*b^3*c+B*a^2*b*d-B*a*b^2*c-C*a^3*d+C*a^2
*b*c)/b^2/(a^2+b^2)*ln(a+b*tan(f*x+e))+1/(a^2+b^2)*(1/2*(A*a*d-A*b*c+B*a*c+
B*b*d-C*a*d+C*b*c)*ln(1+tan(f*x+e)^2)+(A*a*c+A*b*d-B*a*d+B*b*c-C*a*c-C*b*d)
*arctan(tan(f*x+e))))
```

Fricas [A] (verification not implemented)

none

Time = 0.37 (sec) , antiderivative size = 226, normalized size of antiderivative = 1.45

$$\int \frac{(c+d\tan(e+fx))\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx$$

$$=\frac{2\,(((A-C)ab^2+Bb^3)c-(Bab^2-(A-C)b^3)d)fx+2\,(Ca^2b+Cb^3)d\tan{(fx+e)}+((Ca^2b-Bab^2+}{}$$

```
[In] integrate((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
,x, algorithm="fricas")
```

```
[Out] 1/2*(2*(((A - C)*a*b^2 + B*b^3)*c - (B*a*b^2 - (A - C)*b^3)*d)*f*x + 2*(C*a
^2*b + C*b^3)*d*tan(f*x + e) + ((C*a^2*b - B*a*b^2 + A*b^3)*c - (C*a^3 - B*
a^2*b + A*a*b^2)*d)*log((b^2*tan(f*x + e)^2 + 2*a*b*tan(f*x + e) + a^2)/(ta
n(f*x + e)^2 + 1)) - ((C*a^2*b + C*b^3)*c - (C*a^3 - B*a^2*b + C*a*b^2 - B*
b^3)*d)*log(1/(tan(f*x + e)^2 + 1)))/((a^2*b^2 + b^4)*f)
```

Sympy [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 0.96 (sec) , antiderivative size = 2387, normalized size of antiderivative = 15.30

$$\int \frac{(c+d\tan(e+fx))\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx = \text{Too large to display}$$

```
[In] integrate((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e)
),x)
```

```
[Out] Piecewise((zoo*x*(c + d*tan(e))*(A + B*tan(e) + C*tan(e)**2)/tan(e), Eq(a,
0) & Eq(b, 0) & Eq(f, 0)), ((A*c*x + A*d*log(tan(e + f*x)**2 + 1)/(2*f) + B
*c*log(tan(e + f*x)**2 + 1)/(2*f) - B*d*x + B*d*tan(e + f*x)/f - C*c*x + C*
c*tan(e + f*x)/f - C*d*log(tan(e + f*x)**2 + 1)/(2*f) + C*d*tan(e + f*x)**2
/(2*f))/a, Eq(b, 0)), (I*A*c*f*x*tan(e + f*x)/(2*b*f*tan(e + f*x) - 2*I*b*f
) + A*c*f*x/(2*b*f*tan(e + f*x) - 2*I*b*f) + I*A*c/(2*b*f*tan(e + f*x) - 2*
I*b*f) + A*d*f*x*tan(e + f*x)/(2*b*f*tan(e + f*x) - 2*I*b*f) - I*A*d*f*x/(2
*b*f*tan(e + f*x) - 2*I*b*f) - A*d/(2*b*f*tan(e + f*x) - 2*I*b*f) + B*c*f*x
*tan(e + f*x)/(2*b*f*tan(e + f*x) - 2*I*b*f) - I*B*c*f*x/(2*b*f*tan(e + f*x
) - 2*I*b*f) - B*c/(2*b*f*tan(e + f*x) - 2*I*b*f) + I*B*d*f*x*tan(e + f*x)/
(2*b*f*tan(e + f*x) - 2*I*b*f) + B*d*f*x/(2*b*f*tan(e + f*x) - 2*I*b*f) + B
*d*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*b*f*tan(e + f*x) - 2*I*b*f) - I
*B*d*log(tan(e + f*x)**2 + 1)/(2*b*f*tan(e + f*x) - 2*I*b*f) - I*B*d/(2*b*f
*tan(e + f*x) - 2*I*b*f) + I*C*c*f*x*tan(e + f*x)/(2*b*f*tan(e + f*x) - 2*I
*b*f) + C*c*f*x/(2*b*f*tan(e + f*x) - 2*I*b*f) + C*c*log(tan(e + f*x)**2 +
1)*tan(e + f*x)/(2*b*f*tan(e + f*x) - 2*I*b*f) - I*C*c*log(tan(e + f*x)**2
+ 1)/(2*b*f*tan(e + f*x) - 2*I*b*f) - I*C*c/(2*b*f*tan(e + f*x) - 2*I*b*f)
- 3*C*d*f*x*tan(e + f*x)/(2*b*f*tan(e + f*x) - 2*I*b*f) + 3*I*C*d*f*x/(2*b*
f*tan(e + f*x) - 2*I*b*f) + I*C*d*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*
b*f*tan(e + f*x) - 2*I*b*f) + C*d*log(tan(e + f*x)**2 + 1)/(2*b*f*tan(e + f
*x) - 2*I*b*f) + 2*C*d*tan(e + f*x)**2/(2*b*f*tan(e + f*x) - 2*I*b*f) + 3*C
*d/(2*b*f*tan(e + f*x) - 2*I*b*f), Eq(a, -I*b)), (-I*A*c*f*x*tan(e + f*x)/(
2*b*f*tan(e + f*x) + 2*I*b*f) + A*c*f*x/(2*b*f*tan(e + f*x) + 2*I*b*f) - I*
A*c/(2*b*f*tan(e + f*x) + 2*I*b*f) + A*d*f*x*tan(e + f*x)/(2*b*f*tan(e + f*
x) + 2*I*b*f) + I*A*d*f*x/(2*b*f*tan(e + f*x) + 2*I*b*f) - A*d/(2*b*f*tan(e
 + f*x) + 2*I*b*f) + B*c*f*x*tan(e + f*x)/(2*b*f*tan(e + f*x) + 2*I*b*f) +
I*B*c*f*x/(2*b*f*tan(e + f*x) + 2*I*b*f) - B*c/(2*b*f*tan(e + f*x) + 2*I*b*
f) - I*B*d*f*x*tan(e + f*x)/(2*b*f*tan(e + f*x) + 2*I*b*f) + B*d*f*x/(2*b*f
*tan(e + f*x) + 2*I*b*f) + B*d*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*b*f
*tan(e + f*x) + 2*I*b*f) + I*B*d*log(tan(e + f*x)**2 + 1)/(2*b*f*tan(e + f*
x) + 2*I*b*f) + I*B*d/(2*b*f*tan(e + f*x) + 2*I*b*f) - I*C*c*f*x*tan(e + f*
x)/(2*b*f*tan(e + f*x) + 2*I*b*f) + C*c*f*x/(2*b*f*tan(e + f*x) + 2*I*b*f)
+ C*c*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*b*f*tan(e + f*x) + 2*I*b*f)
+ I*C*c*log(tan(e + f*x)**2 + 1)/(2*b*f*tan(e + f*x) + 2*I*b*f) + I*C*c/(2*
```

```
b*f*tan(e + f*x) + 2*I*b*f) - 3*C*d*f*x*tan(e + f*x)/(2*b*f*tan(e + f*x) +
2*I*b*f) - 3*I*C*d*f*x/(2*b*f*tan(e + f*x) + 2*I*b*f) - I*C*d*log(tan(e + f
*x)**2 + 1)*tan(e + f*x)/(2*b*f*tan(e + f*x) + 2*I*b*f) + C*d*log(tan(e + f
*x)**2 + 1)/(2*b*f*tan(e + f*x) + 2*I*b*f) + 2*C*d*tan(e + f*x)**2/(2*b*f*t
an(e + f*x) + 2*I*b*f) + 3*C*d/(2*b*f*tan(e + f*x) + 2*I*b*f), Eq(a, I*b)),
 (x*(c + d*tan(e))*(A + B*tan(e) + C*tan(e)**2)/(a + b*tan(e)), Eq(f, 0)),
(2*A*a*b**2*c*f*x/(2*a**2*b**2*f + 2*b**4*f) - 2*A*a*b**2*d*log(a/b + tan(e
 + f*x))/(2*a**2*b**2*f + 2*b**4*f) + A*a*b**2*d*log(tan(e + f*x)**2 + 1)/(
2*a**2*b**2*f + 2*b**4*f) + 2*A*b**3*c*log(a/b + tan(e + f*x))/(2*a**2*b**2
*f + 2*b**4*f) - A*b**3*c*log(tan(e + f*x)**2 + 1)/(2*a**2*b**2*f + 2*b**4*
f) + 2*A*b**3*d*f*x/(2*a**2*b**2*f + 2*b**4*f) + 2*B*a**2*b*d*log(a/b + tan
(e + f*x))/(2*a**2*b**2*f + 2*b**4*f) - 2*B*a*b**2*c*log(a/b + tan(e + f*x)
)/(2*a**2*b**2*f + 2*b**4*f) + B*a*b**2*c*log(tan(e + f*x)**2 + 1)/(2*a**2*
b**2*f + 2*b**4*f) - 2*B*a*b**2*d*f*x/(2*a**2*b**2*f + 2*b**4*f) + 2*B*b**3
*c*f*x/(2*a**2*b**2*f + 2*b**4*f) + B*b**3*d*log(tan(e + f*x)**2 + 1)/(2*a*
*2*b**2*f + 2*b**4*f) - 2*C*a**3*d*log(a/b + tan(e + f*x))/(2*a**2*b**2*f +
 2*b**4*f) + 2*C*a**2*b*c*log(a/b + tan(e + f*x))/(2*a**2*b**2*f + 2*b**4*f
) + 2*C*a**2*b*d*tan(e + f*x)/(2*a**2*b**2*f + 2*b**4*f) - 2*C*a*b**2*c*f*x
/(2*a**2*b**2*f + 2*b**4*f) - C*a*b**2*d*log(tan(e + f*x)**2 + 1)/(2*a**2*b
**2*f + 2*b**4*f) + C*b**3*c*log(tan(e + f*x)**2 + 1)/(2*a**2*b**2*f + 2*b*
*4*f) - 2*C*b**3*d*f*x/(2*a**2*b**2*f + 2*b**4*f) + 2*C*b**3*d*tan(e + f*x)
/(2*a**2*b**2*f + 2*b**4*f), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.42 (sec) , antiderivative size = 183, normalized size of antiderivative = 1.17

$$\int \frac{(c+d\tan(e+fx))\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx$$
$$=\frac{\frac{2\,Cd\tan(fx+e)}{b}+\frac{2\,(((A-C)a+Bb)c-(Ba-(A-C)b)d)(fx+e)}{a^2+b^2}+\frac{2\,((Ca^2b-Bab^2+Ab^3)c-(Ca^3-Ba^2b+Aab^2)d)\log(b\tan(fx+e)+a)}{a^2b^2+b^4}+\cdots}{2\,f}$$

```
[In] integrate((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
,x, algorithm="maxima")
```

```
[Out] 1/2*(2*C*d*tan(f*x + e)/b + 2*(((A - C)*a + B*b)*c - (B*a - (A - C)*b)*d)*(
f*x + e)/(a^2 + b^2) + 2*((C*a^2*b - B*a*b^2 + A*b^3)*c - (C*a^3 - B*a^2*b
+ A*a*b^2)*d)*log(b*tan(f*x + e) + a)/(a^2*b^2 + b^4) + ((B*a - (A - C)*b)*
c + ((A - C)*a + B*b)*d)*log(tan(f*x + e)^2 + 1)/(a^2 + b^2))/f
```

Giac [A] (verification not implemented)

none

Time = 0.55 (sec) , antiderivative size = 182, normalized size of antiderivative = 1.17

$$\int \frac{(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{a+b\tan(e+fx)}\,dx = \frac{\frac{2\,Cd\tan(fx+e)}{b}+\frac{2\,(Aac-Cac+Bbc-Bad+Abd-Cbd)(fx+e)}{a^2+b^2}+\frac{(Bac-Abc+Cbc+Aad-Cad+Bbd)\log\left(\tan(fx+e)^2+1\right)}{a^2+b^2}+\frac{2\,(Ca^2bc-Bab^2}{}}{2\,f}$$

```
[In] integrate((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
,x, algorithm="giac")
```

```
[Out] 1/2*(2*C*d*tan(f*x + e)/b + 2*(A*a*c - C*a*c + B*b*c - B*a*d + A*b*d - C*b*
d)*(f*x + e)/(a^2 + b^2) + (B*a*c - A*b*c + C*b*c + A*a*d - C*a*d + B*b*d)*
log(tan(f*x + e)^2 + 1)/(a^2 + b^2) + 2*(C*a^2*b*c - B*a*b^2*c + A*b^3*c -
C*a^3*d + B*a^2*b*d - A*a*b^2*d)*log(abs(b*tan(f*x + e) + a))/(a^2*b^2 + b^
4))/f
```

Mupad [B] (verification not implemented)

Time = 9.56 (sec) , antiderivative size = 186, normalized size of antiderivative = 1.19

$$\begin{aligned}\int \frac{(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{a+b\tan(e+fx)}\,dx &= \frac{\ln\left(\tan(e+f\,x)-\mathrm{i}\right)\,\left(A\,d+B\,c-C\,d-A\,c\,1\mathrm{i}+B\,d\,1\mathrm{i}+C\,c\,1\mathrm{i}\right)}{2\,f\,\left(a+b\,1\mathrm{i}\right)}\\ &+\frac{\ln\left(\tan(e+f\,x)+1\mathrm{i}\right)\,\left(B\,d+A\,d\,1\mathrm{i}+B\,c\,1\mathrm{i}-A\,c+C\,c-C\,d\,1\mathrm{i}\right)}{2\,f\,\left(b+a\,1\mathrm{i}\right)}\\ &-\frac{\ln\left(a+b\tan(e+f\,x)\right)\,\left(b^2\,(A\,a\,d+B\,a\,c)-b\,(B\,a^2\,d+C\,a^2\,c)-A\,b^3\,c+C\,a^3\,d\right)}{f\,\left(a^2\,b^2+b^4\right)}\\ &+\frac{C\,d\tan(e+f\,x)}{b\,f}\end{aligned}$$

```
[In] int(((c + d*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a + b*t
an(e + f*x)),x)
```

```
[Out] (log(tan(e + f*x) - 1i)*(A*d - A*c*1i + B*c + B*d*1i + C*c*1i - C*d))/(2*f*
(a + b*1i)) + (log(tan(e + f*x) + 1i)*(A*d*1i - A*c + B*c*1i + B*d + C*c -
C*d*1i))/(2*f*(a*1i + b)) - (log(a + b*tan(e + f*x))*(b^2*(A*a*d + B*a*c) -
 b*(B*a^2*d + C*a^2*c) - A*b^3*c + C*a^3*d))/(f*(b^4 + a^2*b^2)) + (C*d*tan
(e + f*x))/(b*f)
```

3.55 $\int \frac{(c+d\tan(e+fx))(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx$

Optimal result 457
Rubi [A] (verified) 458
Mathematica [C] (verified) 460
Maple [A] (verified) 461
Fricas [B] (verification not implemented) 461
Sympy [C] (verification not implemented) 462
Maxima [A] (verification not implemented) 467
Giac [A] (verification not implemented) 468
Mupad [B] (verification not implemented) 468

Optimal result

Integrand size = 43, antiderivative size = 265

$$\begin{aligned}
&\int \frac{(c+d\tan(e+fx))\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx \\
&= \frac{(a^2(Ac-cC-Bd)-b^2(Ac-cC-Bd)+2ab(Bc+(A-C)d))\,x}{(a^2+b^2)^2} \\
&\quad+\frac{(2ab(Ac-cC-Bd)-a^2(Bc+(A-C)d)+b^2(Bc+(A-C)d))\log(\cos(e+fx))}{(a^2+b^2)^2 f} \\
&\quad+\frac{(a^4Cd+b^4(Bc+Ad)+2ab^3(Ac-cC-Bd)-a^2b^2(Bc+(A-3C)d))\log(a+b\tan(e+fx))}{b^2\,(a^2+b^2)^2 f} \\
&\quad-\frac{(Ab^2-a(bB-aC))\,(bc-ad)}{b^2\,(a^2+b^2)\,f(a+b\tan(e+fx))}
\end{aligned}$$

```
[Out] (a^2*(A*c-B*d-C*c)-b^2*(A*c-B*d-C*c)+2*a*b*(B*c+(A-C)*d))*x/(a^2+b^2)^2+(2*
a*b*(A*c-B*d-C*c)-a^2*(B*c+(A-C)*d)+b^2*(B*c+(A-C)*d))*ln(cos(f*x+e))/(a^2+
b^2)^2/f+(a^4*C*d+b^4*(A*d+B*c)+2*a*b^3*(A*c-B*d-C*c)-a^2*b^2*(B*c+(A-3*C)*
d))*ln(a+b*tan(f*x+e))/b^2/(a^2+b^2)^2/f-(A*b^2-a*(B*b-C*a))*(-a*d+b*c)/b^2
/(a^2+b^2)/f/(a+b*tan(f*x+e))
```

Rubi [A] (verified)

Time = 0.52 (sec) , antiderivative size = 265, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}} = 0.116$, Rules used = {3716, 3707, 3698, 31, 3556}

$$\int \frac{(c+d\tan(e+fx))\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx$$
$$= -\frac{(bc-ad)\,(Ab^2-a(bB-aC))}{b^2f\,(a^2+b^2)\,(a+b\tan(e+fx))} + \frac{\log(\cos(e+fx))\,(-(a^2(d(A-C)+Bc))+2ab(Ac-Bd-cC)+b^2(d(A-C)+Bc))}{f\,(a^2+b^2)^2} + \frac{x(a^2(Ac-Bd-cC)+2ab(d(A-C)+Bc)-b^2(Ac-Bd-cC))}{(a^2+b^2)^2} + \frac{(a^4Cd-a^2b^2(d(A-3C)+Bc)+2ab^3(Ac-Bd-cC)+b^4(Ad+Bc))\log(a+b\tan(e+fx))}{b^2f\,(a^2+b^2)^2}$$

```
[In] Int[((c + d*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a + b*T
an[e + f*x])^2,x]
```

```
[Out] ((a^2*(A*c - c*C - B*d) - b^2*(A*c - c*C - B*d) + 2*a*b*(B*c + (A - C)*d))*
x)/(a^2 + b^2)^2 + ((2*a*b*(A*c - c*C - B*d) - a^2*(B*c + (A - C)*d) + b^2*
(B*c + (A - C)*d))*Log[Cos[e + f*x]])/((a^2 + b^2)^2*f) + ((a^4*C*d + b^4*(
B*c + A*d) + 2*a*b^3*(A*c - c*C - B*d) - a^2*b^2*(B*c + (A - 3*C)*d))*Log[a
 + b*Tan[e + f*x]])/(b^2*(a^2 + b^2)^2*f) - ((A*b^2 - a*(b*B - a*C))*(b*c -
 a*d))/(b^2*(a^2 + b^2)*f*(a + b*Tan[e + f*x]))
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3698

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3707

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[(a*A + b*B - a
```

```
*C)*(x/(a^2 + b^2)), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3716

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.
)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(-(b*c - a*d))*(c^2*C - B*c*d + A*d^2)*((c
+ d*Tan[e + f*x])^(n + 1)/(d^2*f*(n + 1)*(c^2 + d^2))), x] + Dist[1/(d*(c^2
 + d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)*Simp[a*d*(A*c - c*C + B*d) + b*(
c^2*C - B*c*d + A*d^2) + d*(A*b*c + a*B*c - b*c*C - a*A*d + b*B*d + a*C*d)*
Tan[e + f*x] + b*C*(c^2 + d^2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c
, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] && LtQ[n,
 -1]
```

Rubi steps

integral

$$= -\frac{(Ab^2-a(bB-aC))\,(bc-ad)}{b^2\,(a^2+b^2)\,f(a+b\tan(e+fx))} + \frac{\int \frac{a^2Cd+b^2(Bc+Ad)+ab(Ac-cC-Bd)-b(Abc-aBc-bcC-aAd-bBd+aCd)\tan(e+fx)+(a^2+b^2)Cd\tan^2(e+fx)}{a+b\tan(e+fx)}\,dx}{b\,(a^2+b^2)}$$

$$= \frac{(a^2(Ac-cC-Bd)-b^2(Ac-cC-Bd)+2ab(Bc+(A-C)d))\,x}{(a^2+b^2)^2} - \frac{(Ab^2-a(bB-aC))\,(bc-ad)}{b^2\,(a^2+b^2)\,f(a+b\tan(e+fx))} + \frac{(a^4Cd+b^4(Bc+Ad)+2ab^3(Ac-cC-Bd)-a^2b^2(Bc+(A-3C)d))\int \frac{1+\tan^2(e+fx)}{a+b\tan(e+fx)}\,dx}{b\,(a^2+b^2)^2} - \frac{(2ab(Ac-cC-Bd)-a^2(Bc+(A-C)d)+b^2(Bc+(A-C)d))\int \tan(e+fx)\,dx}{(a^2+b^2)^2}$$

$$= \frac{(a^2(Ac-cC-Bd)-b^2(Ac-cC-Bd)+2ab(Bc+(A-C)d))\,x}{(a^2+b^2)^2} + \frac{(2ab(Ac-cC-Bd)-a^2(Bc+(A-C)d)+b^2(Bc+(A-C)d))\log(\cos(e+fx))}{(a^2+b^2)^2\,f} - \frac{(Ab^2-a(bB-aC))\,(bc-ad)}{b^2\,(a^2+b^2)\,f(a+b\tan(e+fx))} + \frac{(a^4Cd+b^4(Bc+Ad)+2ab^3(Ac-cC-Bd)-a^2b^2(Bc+(A-3C)d))\,\mathrm{Subst}\left(\int \frac{1}{a+x}\,dx,x,b\tan(e+f\right.}{b^2\,(a^2+b^2)^2\,f}$$

$$= \frac{(a^2(Ac - cC - Bd) - b^2(Ac - cC - Bd) + 2ab(Bc + (A - C)d))\, x}{(a^2 + b^2)^2} + \frac{(2ab(Ac - cC - Bd) - a^2(Bc + (A - C)d) + b^2(Bc + (A - C)d)) \log(\cos(e + fx))}{(a^2 + b^2)^2 f} + \frac{(a^4Cd + b^4(Bc + Ad) + 2ab^3(Ac - cC - Bd) - a^2b^2(Bc + (A - 3C)d)) \log(a + b\tan(e + fx))}{b^2 (a^2 + b^2)^2 f} - \frac{(Ab^2 - a(bB - aC))\,(bc - ad)}{b^2 (a^2 + b^2) f(a + b\tan(e + fx))}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 2.79 (sec) , antiderivative size = 216, normalized size of antiderivative = 0.82

$$\int \frac{(c + d\tan(e + fx))\,(A + B\tan(e + fx) + C\tan^2(e + fx))}{(a + b\tan(e + fx))^2}\, dx = \frac{\frac{(A+iB-C)(-ic+d)\log(i-\tan(e+fx))}{(a+ib)^2} + \frac{(A-iB-C)(ic+d)\log(i+\tan(e+fx))}{(a-ib)^2} + \frac{2(a^4Cd+b^4(Bc+Ad)+2ab^3(Ac-cC-Bd)-a^2b^2(Bc+(A-3C}{b^2(a^2+b^2)^2}}{2f}$$

```
[In] Integrate[((c + d*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a
 + b*Tan[e + f*x])^2,x]
```

```
[Out] (((A + I*B - C)*((-I)*c + d)*Log[I - Tan[e + f*x]])/(a + I*b)^2 + ((A - I*B
 - C)*(I*c + d)*Log[I + Tan[e + f*x]])/(a - I*b)^2 + (2*(a^4*C*d + b^4*(B*c
 + A*d) + 2*a*b^3*(A*c - c*C - B*d) - a^2*b^2*(B*c + (A - 3*C)*d))*Log[a +
b*Tan[e + f*x]])/(b^2*(a^2 + b^2)^2) - (2*(A*b^2 + a*(-(b*B) + a*C))*(b*c -
 a*d))/(b^2*(a^2 + b^2)*(a + b*Tan[e + f*x])))/(2*f)
```

Maple [A] (verified)

Time = 0.14 (sec) , antiderivative size = 321, normalized size of antiderivative = 1.21

method	result
derivativedivides	$-\frac{-Aa\,b^{2}d+A\,b^{3}c+B\,a^{2}bd-Ba\,b^{2}c-a^{3}Cd+C\,a^{2}bc}{b^{2}\left(a^{2}+b^{2}\right)\left(a+b\tan\left(fx+e\right)\right)}+\frac{\left(-A\,a^{2}b^{2}d+2Aa\,b^{3}c+A\,b^{4}d-B\,a^{2}b^{2}c-2Ba\,b^{3}d+B\,b^{4}c+a^{4}Cd+3C\,a^{2}b^{2}d-2Ca\right.}{\left(a^{2}+b^{2}\right)^{2}b^{2}}$
default	$-\frac{-Aa\,b^{2}d+A\,b^{3}c+B\,a^{2}bd-Ba\,b^{2}c-a^{3}Cd+C\,a^{2}bc}{b^{2}\left(a^{2}+b^{2}\right)\left(a+b\tan\left(fx+e\right)\right)}+\frac{\left(-A\,a^{2}b^{2}d+2Aa\,b^{3}c+A\,b^{4}d-B\,a^{2}b^{2}c-2Ba\,b^{3}d+B\,b^{4}c+a^{4}Cd+3C\,a^{2}b^{2}d-2Ca\right.}{\left(a^{2}+b^{2}\right)^{2}b^{2}}$
norman	$\frac{\frac{a\left(A\,a^{2}c+2Aabd-A\,b^{2}c-B\,a^{2}d+2Babc+B\,b^{2}d-C\,a^{2}c-2Cabd+C\,b^{2}c\right)x}{a^{4}+2a^{2}b^{2}+b^{4}}+\frac{Aa\,b^{2}d-A\,b^{3}c-B\,a^{2}bd+Ba\,b^{2}c+a^{3}Cd-C\,a^{2}bc}{b^{2}f\left(a^{2}+b^{2}\right)}+\frac{b\left(A\,a^{2}\right.}{}}{a+b\tan\left(fx+e\right)}$
parallelrisch	Expression too large to display
risch	Expression too large to display

```
[In] int((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2,x,m
ethod=_RETURNVERBOSE)
```

```
[Out] 1/f*(-(-A*a*b^2*d+A*b^3*c+B*a^2*b*d-B*a*b^2*c-C*a^3*d+C*a^2*b*c)/b^2/(a^2+b
^2)/(a+b*tan(f*x+e))+1/(a^2+b^2)^2*(-A*a^2*b^2*d+2*A*a*b^3*c+A*b^4*d-B*a^2*
b^2*c-2*B*a*b^3*d+B*b^4*c+C*a^4*d+3*C*a^2*b^2*d-2*C*a*b^3*c)/b^2*ln(a+b*tan
(f*x+e))+1/(a^2+b^2)^2*(1/2*(A*a^2*d-2*A*a*b*c-A*b^2*d+B*a^2*c+2*B*a*b*d-B*
b^2*c-C*a^2*d+2*C*a*b*c+C*b^2*d)*ln(1+tan(f*x+e)^2)+(A*a^2*c+2*A*a*b*d-A*b^
2*c-B*a^2*d+2*B*a*b*c+B*b^2*d-C*a^2*c-2*C*a*b*d+C*b^2*c)*arctan(tan(f*x+e))
))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 556 vs. $2(265) = 530$.

Time = 0.39 (sec) , antiderivative size = 556, normalized size of antiderivative = 2.10

$$\int \frac{(c+d\tan(e+fx))\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx$$
$$= \frac{2\,(((A-C)a^3b^2+2\,Ba^2b^3-(A-C)ab^4)c-(Ba^3b^2-2\,(A-C)a^2b^3-Bab^4)d)fx-2\,(Ca^2b^3-Bab^4+}{}$$

```
[In] integrate((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^2,x, algorithm="fricas")
```

```
[Out] 1/2*(2*(((A - C)*a^3*b^2 + 2*B*a^2*b^3 - (A - C)*a*b^4)*c - (B*a^3*b^2 - 2*
(A - C)*a^2*b^3 - B*a*b^4)*d)*f*x - 2*(C*a^2*b^3 - B*a*b^4 + A*b^5)*c + 2*(
C*a^3*b^2 - B*a^2*b^3 + A*a*b^4)*d - ((B*a^3*b^2 - 2*(A - C)*a^2*b^3 - B*a*
```

```
b^4)*c - (C*a^5 - (A - 3*C)*a^3*b^2 - 2*B*a^2*b^3 + A*a*b^4)*d + ((B*a^2*b^
3 - 2*(A - C)*a*b^4 - B*b^5)*c - (C*a^4*b - (A - 3*C)*a^2*b^3 - 2*B*a*b^4 +
 A*b^5)*d)*tan(f*x + e))*log((b^2*tan(f*x + e)^2 + 2*a*b*tan(f*x + e) + a^2
)/(tan(f*x + e)^2 + 1)) - ((C*a^4*b + 2*C*a^2*b^3 + C*b^5)*d*tan(f*x + e) +
 (C*a^5 + 2*C*a^3*b^2 + C*a*b^4)*d)*log(1/(tan(f*x + e)^2 + 1)) + 2*((((A -
 C)*a^2*b^3 + 2*B*a*b^4 - (A - C)*b^5)*c - (B*a^2*b^3 - 2*(A - C)*a*b^4 - B
*b^5)*d)*f*x + (C*a^3*b^2 - B*a^2*b^3 + A*a*b^4)*c - (C*a^4*b - B*a^3*b^2 +
 A*a^2*b^3)*d)*tan(f*x + e))/((a^4*b^3 + 2*a^2*b^5 + b^7)*f*tan(f*x + e) +
(a^5*b^2 + 2*a^3*b^4 + a*b^6)*f)
```

Sympy [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 1.41 (sec) , antiderivative size = 9721, normalized size of antiderivative = 36.68

$$\int \frac{(c+d\tan(e+fx))\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx = \text{Too large to display}$$

```
[In] integrate((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e)
)**2,x)
```

```
[Out] Piecewise((zoo*x*(c + d*tan(e))*(A + B*tan(e) + C*tan(e)**2)/tan(e)**2, Eq(
a, 0) & Eq(b, 0) & Eq(f, 0)), ((A*c*x + A*d*log(tan(e + f*x)**2 + 1)/(2*f)
+ B*c*log(tan(e + f*x)**2 + 1)/(2*f) - B*d*x + B*d*tan(e + f*x)/f - C*c*x +
 C*c*tan(e + f*x)/f - C*d*log(tan(e + f*x)**2 + 1)/(2*f) + C*d*tan(e + f*x)
**2/(2*f))/a**2, Eq(b, 0)), (-A*c*f*x*tan(e + f*x)**2/(4*b**2*f*tan(e + f*x
)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 2*I*A*c*f*x*tan(e + f*x)/(4*b*
*2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + A*c*f*x/(4*b**
2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - A*c*tan(e + f*x
)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 2*I*A*c
/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + I*A*d*f*
x*tan(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b
**2*f) + 2*A*d*f*x*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(
e + f*x) - 4*b**2*f) - I*A*d*f*x/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan
(e + f*x) - 4*b**2*f) + I*A*d*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*
b**2*f*tan(e + f*x) - 4*b**2*f) + I*B*c*f*x*tan(e + f*x)**2/(4*b**2*f*tan(e
 + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 2*B*c*f*x*tan(e + f*x)/(
4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - I*B*c*f*x/
(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + I*B*c*tan
(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) +
 B*d*f*x*tan(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x
) - 4*b**2*f) - 2*I*B*d*f*x*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b*
*2*f*tan(e + f*x) - 4*b**2*f) - B*d*f*x/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**
2*f*tan(e + f*x) - 4*b**2*f) - 3*B*d*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2
 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 2*I*B*d/(4*b**2*f*tan(e + f*x)**2
```

```
- 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + C*c*f*x*tan(e + f*x)**2/(4*b**2*f*t
an(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 2*I*C*c*f*x*tan(e +
f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - C*c*
f*x/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 3*C*c
*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*
f) + 2*I*C*c/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f
) + 3*I*C*d*f*x*tan(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(
e + f*x) - 4*b**2*f) + 6*C*d*f*x*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8
*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 3*I*C*d*f*x/(4*b**2*f*tan(e + f*x)**2
- 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 2*C*d*log(tan(e + f*x)**2 + 1)*tan(
e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f)
 - 4*I*C*d*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2
- 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 2*C*d*log(tan(e + f*x)**2 + 1)/(4*b
**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 5*I*C*d*tan(e
 + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 4
*C*d/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f), Eq(a,
 -I*b)), (-A*c*f*x*tan(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*t
an(e + f*x) - 4*b**2*f) - 2*I*A*c*f*x*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**
2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + A*c*f*x/(4*b**2*f*tan(e + f*x)**2
 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - A*c*tan(e + f*x)/(4*b**2*f*tan(e +
 f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 2*I*A*c/(4*b**2*f*tan(e +
f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - I*A*d*f*x*tan(e + f*x)**2/(
4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 2*A*d*f*x*
tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f
) + I*A*d*f*x/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*
f) - I*A*d*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x)
 - 4*b**2*f) - I*B*c*f*x*tan(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 + 8*I*b*
*2*f*tan(e + f*x) - 4*b**2*f) + 2*B*c*f*x*tan(e + f*x)/(4*b**2*f*tan(e + f*
x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + I*B*c*f*x/(4*b**2*f*tan(e + f
*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - I*B*c*tan(e + f*x)/(4*b**2*f
*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + B*d*f*x*tan(e + f*
x)**2/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 2*I
*B*d*f*x*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) -
 4*b**2*f) - B*d*f*x/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) -
4*b**2*f) - 3*B*d*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e
 + f*x) - 4*b**2*f) - 2*I*B*d/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e
+ f*x) - 4*b**2*f) + C*c*f*x*tan(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 + 8*
I*b**2*f*tan(e + f*x) - 4*b**2*f) + 2*I*C*c*f*x*tan(e + f*x)/(4*b**2*f*tan(
e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - C*c*f*x/(4*b**2*f*tan(e
 + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 3*C*c*tan(e + f*x)/(4*b*
*2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 2*I*C*c/(4*b**
2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 3*I*C*d*f*x*tan
(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f
) + 6*C*d*f*x*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f
*x) - 4*b**2*f) + 3*I*C*d*f*x/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e
```

```
+ f*x) - 4*b**2*f) + 2*C*d*log(tan(e + f*x)**2 + 1)*tan(e + f*x)**2/(4*b**2
*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 4*I*C*d*log(tan(
e + f*x)**2 + 1)*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e
+ f*x) - 4*b**2*f) - 2*C*d*log(tan(e + f*x)**2 + 1)/(4*b**2*f*tan(e + f*x)*
*2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 5*I*C*d*tan(e + f*x)/(4*b**2*f*t
an(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 4*C*d/(4*b**2*f*tan(
e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f), Eq(a, I*b)), (x*(c + d*t
an(e))*(A + B*tan(e) + C*tan(e)**2)/(a + b*tan(e))**2, Eq(f, 0)), (2*A*a**3
*b**2*c*f*x/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4
*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - 2*A*a**3*
b**2*d*log(a/b + tan(e + f*x))/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x)
+ 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e
+ f*x)) + A*a**3*b**2*d*log(tan(e + f*x)**2 + 1)/(2*a**5*b**2*f + 2*a**4*b*
*3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f
 + 2*b**7*f*tan(e + f*x)) + 2*A*a**3*b**2*d/(2*a**5*b**2*f + 2*a**4*b**3*f*
tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*
b**7*f*tan(e + f*x)) + 2*A*a**2*b**3*c*f*x*tan(e + f*x)/(2*a**5*b**2*f + 2*
a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a
*b**6*f + 2*b**7*f*tan(e + f*x)) + 4*A*a**2*b**3*c*log(a/b + tan(e + f*x))/
(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f
*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - 2*A*a**2*b**3*c*log(t
an(e + f*x)**2 + 1)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b*
*4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - 2
*A*a**2*b**3*c/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f
+ 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) + 4*A*a*
*2*b**3*d*f*x/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f +
 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - 2*A*a**
2*b**3*d*log(a/b + tan(e + f*x))*tan(e + f*x)/(2*a**5*b**2*f + 2*a**4*b**3*
f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f +
2*b**7*f*tan(e + f*x)) + A*a**2*b**3*d*log(tan(e + f*x)**2 + 1)*tan(e + f*x
)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5
*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - 2*A*a*b**4*c*f*x/(2
*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*t
an(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) + 4*A*a*b**4*c*log(a/b +
tan(e + f*x))*tan(e + f*x)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*
a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*
x)) - 2*A*a*b**4*c*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*a**5*b**2*f + 2
*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*
a*b**6*f + 2*b**7*f*tan(e + f*x)) + 4*A*a*b**4*d*f*x*tan(e + f*x)/(2*a**5*b
**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e +
f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) + 2*A*a*b**4*d*log(a/b + tan(e +
 f*x))/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2
*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - A*a*b**4*d*log
(tan(e + f*x)**2 + 1)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*
b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) +
```

2*A*a*b**4*d/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - 2*A*b**5*c*f*x*tan(e + f*x)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - 2*A*b**5*c/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) + 2*A*b**5*d*log(a/b + tan(e + f*x))*tan(e + f*x)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - A*b**5*d*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - 2*B*a**4*b*d/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - 2*B*a**3*b**2*c*log(a/b + tan(e + f*x))/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) + B*a**3*b**2*c*log(tan(e + f*x)**2 + 1)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) + 2*B*a**3*b**2*c/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - 2*B*a**3*b**2*d*f*x/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) + 4*B*a**2*b**3*c*f*x/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - 2*B*a**2*b**3*c*log(a/b + tan(e + f*x))*tan(e + f*x)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) + B*a**2*b**3*c*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - 2*B*a**2*b**3*d*f*x*tan(e + f*x)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - 4*B*a**2*b**3*d*log(a/b + tan(e + f*x))/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) + 2*B*a**2*b**3*d*log(tan(e + f*x)**2 + 1)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - 2*B*a**2*b**3*d/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) + 4*B*a*b**4*c*f*x*tan(e + f*x)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) + 2*B*a*b**4*c*log(a/b + tan(e + f*x))/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - B*a*b**4*c*log(tan(e + f*x)**2 + 1)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) + 2*B*a*b**4*c/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) + 2*B*a*b**4*d*f*x/(2

*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*t
an(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - 4*B*a*b**4*d*log(a/b +
tan(e + f*x))*tan(e + f*x)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*
a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*
x)) + 2*B*a*b**4*d*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*a**5*b**2*f + 2
*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*
a*b**6*f + 2*b**7*f*tan(e + f*x)) + 2*B*b**5*c*log(a/b + tan(e + f*x))*tan(
e + f*x)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a*
*2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - B*b**5*c*log
(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f
*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*ta
n(e + f*x)) + 2*B*b**5*d*f*x*tan(e + f*x)/(2*a**5*b**2*f + 2*a**4*b**3*f*ta
n(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b*
*7*f*tan(e + f*x)) + 2*C*a**5*d*log(a/b + tan(e + f*x))/(2*a**5*b**2*f + 2*
a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a
*b**6*f + 2*b**7*f*tan(e + f*x)) + 2*C*a**5*d/(2*a**5*b**2*f + 2*a**4*b**3*
f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f +
2*b**7*f*tan(e + f*x)) - 2*C*a**4*b*c/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e
+ f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f
*tan(e + f*x)) + 2*C*a**4*b*d*log(a/b + tan(e + f*x))*tan(e + f*x)/(2*a**5*
b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e +
 f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - 2*C*a**3*b**2*c*f*x/(2*a**5*b
2*f + 2*a4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e +
f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) + 6*C*a**3*b**2*d*log(a/b + tan(
e + f*x))/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a
2*b5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - C*a**3*b**2
*d*log(tan(e + f*x)**2 + 1)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4
*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f
*x)) + 2*C*a**3*b**2*d/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3
*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x))
- 2*C*a**2*b**3*c*f*x*tan(e + f*x)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f
*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*ta
n(e + f*x)) - 4*C*a**2*b**3*c*log(a/b + tan(e + f*x))/(2*a**5*b**2*f + 2*a*
*4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b
6*f + 2*b7*f*tan(e + f*x)) + 2*C*a**2*b**3*c*log(tan(e + f*x)**2 + 1)/(
2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*
tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - 2*C*a**2*b**3*c/(2*a**
5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e
 + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - 4*C*a**2*b**3*d*f*x/(2*a**5
*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e
+ f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) + 6*C*a**2*b**3*d*log(a/b + ta
n(e + f*x))*tan(e + f*x)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a*
*3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)
) - C*a**2*b**3*d*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*a**5*b**2*f + 2*
a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a

```
*b**6*f + 2*b**7*f*tan(e + f*x)) + 2*C*a*b**4*c*f*x/(2*a**5*b**2*f + 2*a**4
*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**
6*f + 2*b**7*f*tan(e + f*x)) - 4*C*a*b**4*c*log(a/b + tan(e + f*x))*tan(e +
 f*x)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*
b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) + 2*C*a*b**4*c*lo
g(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e +
f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*t
an(e + f*x)) - 4*C*a*b**4*d*f*x*tan(e + f*x)/(2*a**5*b**2*f + 2*a**4*b**3*f
*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2
*b**7*f*tan(e + f*x)) + C*a*b**4*d*log(tan(e + f*x)**2 + 1)/(2*a**5*b**2*f
+ 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) +
 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) + 2*C*b**5*c*f*x*tan(e + f*x)/(2*a**5*
b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e +
 f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) + C*b**5*d*log(tan(e + f*x)**2
+ 1)*tan(e + f*x)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4
*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)), True
))
```

Maxima [A] (verification not implemented)

none

Time = 0.37 (sec) , antiderivative size = 338, normalized size of antiderivative = 1.28

$$\int \frac{(c+d\tan(e+fx))\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx$$

$$= \frac{\frac{2\,(((A-C)a^2+2\,Bab-(A-C)b^2)c-(Ba^2-2\,(A-C)ab-Bb^2)d)(fx+e)}{a^4+2\,a^2b^2+b^4} - \frac{2\,((Ba^2b^2-2\,(A-C)ab^3-Bb^4)c-(Ca^4-(A-3\,C)a^2b^2-2\,Bab^3+Ab^4}{a^4b^2+2\,a^2b^4+b^6}}{}$$

```
[In] integrate((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^2,x, algorithm="maxima")
```

```
[Out] 1/2*(2*(((A - C)*a^2 + 2*B*a*b - (A - C)*b^2)*c - (B*a^2 - 2*(A - C)*a*b -
B*b^2)*d)*(f*x + e)/(a^4 + 2*a^2*b^2 + b^4) - 2*((B*a^2*b^2 - 2*(A - C)*a*b
^3 - B*b^4)*c - (C*a^4 - (A - 3*C)*a^2*b^2 - 2*B*a*b^3 + A*b^4)*d)*log(b*ta
n(f*x + e) + a)/(a^4*b^2 + 2*a^2*b^4 + b^6) + ((B*a^2 - 2*(A - C)*a*b - B*b
^2)*c + ((A - C)*a^2 + 2*B*a*b - (A - C)*b^2)*d)*log(tan(f*x + e)^2 + 1)/(a
^4 + 2*a^2*b^2 + b^4) - 2*((C*a^2*b - B*a*b^2 + A*b^3)*c - (C*a^3 - B*a^2*b
 + A*a*b^2)*d)/(a^3*b^2 + a*b^4 + (a^2*b^3 + b^5)*tan(f*x + e)))/f
```

Giac [A] (verification not implemented)

none

Time = 0.66 (sec) , antiderivative size = 518, normalized size of antiderivative = 1.95

$$\int \frac{(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^2}\,dx$$
$$= \frac{\frac{2\,(Aa^2c-Ca^2c+2\,Babc-Ab^2c+Cb^2c-Ba^2d+2\,Aabd-2\,Cabd+Bb^2d)(fx+e)}{a^4+2\,a^2b^2+b^4}+\frac{(Ba^2c-2\,Aabc+2\,Cabc-Bb^2c+Aa^2d-Ca^2d+2\,Babd-Ab^2d+C}{a^4+2\,a^2b^2+b^4}}{}$$

```
[In] integrate((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^2,x, algorithm="giac")
```

```
[Out] 1/2*(2*(A*a^2*c - C*a^2*c + 2*B*a*b*c - A*b^2*c + C*b^2*c - B*a^2*d + 2*A*a
*b*d - 2*C*a*b*d + B*b^2*d)*(f*x + e)/(a^4 + 2*a^2*b^2 + b^4) + (B*a^2*c -
2*A*a*b*c + 2*C*a*b*c - B*b^2*c + A*a^2*d - C*a^2*d + 2*B*a*b*d - A*b^2*d +
 C*b^2*d)*log(tan(f*x + e)^2 + 1)/(a^4 + 2*a^2*b^2 + b^4) - 2*(B*a^2*b^2*c
- 2*A*a*b^3*c + 2*C*a*b^3*c - B*b^4*c - C*a^4*d + A*a^2*b^2*d - 3*C*a^2*b^2
*d + 2*B*a*b^3*d - A*b^4*d)*log(abs(b*tan(f*x + e) + a))/(a^4*b^2 + 2*a^2*b
^4 + b^6) + 2*(B*a^2*b^2*c*tan(f*x + e) - 2*A*a*b^3*c*tan(f*x + e) + 2*C*a*
b^3*c*tan(f*x + e) - B*b^4*c*tan(f*x + e) - C*a^4*d*tan(f*x + e) + A*a^2*b^
2*d*tan(f*x + e) - 3*C*a^2*b^2*d*tan(f*x + e) + 2*B*a*b^3*d*tan(f*x + e) -
A*b^4*d*tan(f*x + e) - C*a^4*c + 2*B*a^3*b*c - 3*A*a^2*b^2*c + C*a^2*b^2*c
- A*b^4*c - B*a^4*d + 2*A*a^3*b*d - 2*C*a^3*b*d + B*a^2*b^2*d)/((a^4*b + 2*
a^2*b^3 + b^5)*(b*tan(f*x + e) + a)))/f
```

Mupad [B] (verification not implemented)

Time = 21.30 (sec) , antiderivative size = 1875, normalized size of antiderivative = 7.08

$$\int \frac{(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^2}\,dx = \text{Too large to display}$$

```
[In] int(((c + d*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a + b*t
an(e + f*x))^2,x)
```

```
[Out] (log(a + b*tan(e + f*x))*(b^4*(A*d + B*c) - b^3*(2*B*a*d - 2*A*a*c + 2*C*a*
c) - b^2*(A*a^2*d + B*a^2*c - 3*C*a^2*d) + C*a^4*d))/(f*(b^6 + 2*a^2*b^4 +
a^4*b^2)) - (log((A*B*b^4*d^2 - A*B*b^4*c^2 + B*C*a^4*d^2 + B*C*b^4*c^2 - A
^2*b^4*c*d + B^2*b^4*c*d + C^2*a^4*c*d - A^2*a*b^3*c^2 + A^2*a*b^3*d^2 + B^
2*a*b^3*c^2 - B^2*a*b^3*d^2 - C^2*a*b^3*c^2 + C^2*a*b^3*d^2 + A*B*a^2*b^2*c
^2 - A*B*a^2*b^2*d^2 - B*C*a^2*b^2*c^2 + 3*B*C*a^2*b^2*d^2 + A^2*a^2*b^2*c*
d - B^2*a^2*b^2*c*d + 3*C^2*a^2*b^2*c*d - A*C*a^4*c*d + A*C*b^4*c*d + 2*A*C
*a*b^3*c^2 - 2*A*C*a*b^3*d^2 - 4*A*C*a^2*b^2*c*d + 4*A*B*a*b^3*c*d - 4*B*C*
```

a*b^3*c*d)/(b*(a^2 + b^2)^2) + (tan(e + f*x)*(A^2*b^4*c^2 + B^2*b^4*d^2 + C
^2*a^4*d^2 + C^2*b^4*c^2 + C^2*b^4*d^2 + A^2*a^2*b^2*d^2 + B^2*a^2*b^2*c^2
+ 3*C^2*a^2*b^2*d^2 - A*C*a^4*d^2 - 2*A*C*b^4*c^2 - A*C*b^4*d^2 - 4*A*C*a^2
*b^2*d^2 - 2*A*B*b^4*c*d - B*C*a^4*c*d + B*C*b^4*c*d - 2*A*B*a*b^3*c^2 + 2*
A*B*a*b^3*d^2 + 2*B*C*a*b^3*c^2 - 2*B*C*a*b^3*d^2 - 2*A^2*a*b^3*c*d + 2*B^2
*a*b^3*c*d - 2*C^2*a*b^3*c*d + 2*A*B*a^2*b^2*c*d - 4*B*C*a^2*b^2*c*d + 4*A*
C*a*b^3*c*d))/(b*(a^2 + b^2)^2) + ((c + d*1i)*(A + B*1i - C)*(A*b*c - B*b*d
 - 4*C*a*d - C*b*c + (tan(e + f*x)*(3*A*b^4*d + 3*B*b^4*c + 2*C*a^4*d - 5*C
*b^4*d + 4*A*a*b^3*c - 4*B*a*b^3*d - 4*C*a*b^3*c - A*a^2*b^2*d - B*a^2*b^2*
c + C*a^2*b^2*d))/(b*(a^2 + b^2)) + (b*(c + d*1i)*(4*a*b - a^2*tan(e + f*x)
 + 3*b^2*tan(e + f*x))*(A + B*1i - C)*1i)/(a*1i - b)^2)*1i)/(2*(a*1i - b)^2
))*(A*c + A*d*1i + B*c*1i - B*d - C*c - C*d*1i))/(2*f*(2*a*b - a^2*1i + b^2
*1i)) - (log((A*B*b^4*d^2 - A*B*b^4*c^2 + B*C*a^4*d^2 + B*C*b^4*c^2 - A^2*b
^4*c*d + B^2*b^4*c*d + C^2*a^4*c*d - A^2*a*b^3*c^2 + A^2*a*b^3*d^2 + B^2*a*
b^3*c^2 - B^2*a*b^3*d^2 - C^2*a*b^3*c^2 + C^2*a*b^3*d^2 + A*B*a^2*b^2*c^2 -
 A*B*a^2*b^2*d^2 - B*C*a^2*b^2*c^2 + 3*B*C*a^2*b^2*d^2 + A^2*a^2*b^2*c*d -
B^2*a^2*b^2*c*d + 3*C^2*a^2*b^2*c*d - A*C*a^4*c*d + A*C*b^4*c*d + 2*A*C*a*b
^3*c^2 - 2*A*C*a*b^3*d^2 - 4*A*C*a^2*b^2*c*d + 4*A*B*a*b^3*c*d - 4*B*C*a*b^
3*c*d)/(b*(a^2 + b^2)^2) + (tan(e + f*x)*(A^2*b^4*c^2 + B^2*b^4*d^2 + C^2*a
^4*d^2 + C^2*b^4*c^2 + C^2*b^4*d^2 + A^2*a^2*b^2*d^2 + B^2*a^2*b^2*c^2 + 3*
C^2*a^2*b^2*d^2 - A*C*a^4*d^2 - 2*A*C*b^4*c^2 - A*C*b^4*d^2 - 4*A*C*a^2*b^2
*d^2 - 2*A*B*b^4*c*d - B*C*a^4*c*d + B*C*b^4*c*d - 2*A*B*a*b^3*c^2 + 2*A*B*
a*b^3*d^2 + 2*B*C*a*b^3*c^2 - 2*B*C*a*b^3*d^2 - 2*A^2*a*b^3*c*d + 2*B^2*a*b
^3*c*d - 2*C^2*a*b^3*c*d + 2*A*B*a^2*b^2*c*d - 4*B*C*a^2*b^2*c*d + 4*A*C*a*
b^3*c*d))/(b*(a^2 + b^2)^2) + ((c*1i + d)*(B*1i - A + C)*(A*b*c - B*b*d - 4
*C*a*d - C*b*c + (tan(e + f*x)*(3*A*b^4*d + 3*B*b^4*c + 2*C*a^4*d - 5*C*b^4
*d + 4*A*a*b^3*c - 4*B*a*b^3*d - 4*C*a*b^3*c - A*a^2*b^2*d - B*a^2*b^2*c +
C*a^2*b^2*d))/(b*(a^2 + b^2)) + (b*(c*1i + d)*(4*a*b - a^2*tan(e + f*x) + 3
*b^2*tan(e + f*x))*(B*1i - A + C))/(a*1i + b)^2))/(2*(a*1i + b)^2))*(A*c*1i
 + A*d + B*c - B*d*1i - C*c*1i - C*d))/(2*f*(a*b*2i - a^2 + b^2)) - (A*b^3*
c - C*a^3*d - A*a*b^2*d - B*a*b^2*c + B*a^2*b*d + C*a^2*b*c)/(b^2*f*(a^2 +
b^2)*(a + b*tan(e + f*x)))

3.56 $\int \frac{(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^3}\,dx$

Optimal result 470
Rubi [A] (verified) 471
Mathematica [C] (verified) 473
Maple [A] (verified) 474
Fricas [B] (verification not implemented) 474
Sympy [F(-2)] 475
Maxima [A] (verification not implemented) 475
Giac [B] (verification not implemented) 476
Mupad [B] (verification not implemented) 477

Optimal result

Integrand size = 43, antiderivative size = 320

$$\int \frac{(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^3}\,dx$$
$$=\frac{(a^3(Ac-cC-Bd)-3ab^2(Ac-cC-Bd)+3a^2b(Bc+(A-C)d)-b^3(Bc+(A-C)d))\,x}{(a^2+b^2)^3}$$
$$+\frac{(3a^2b(Ac-cC-Bd)-b^3(Ac-cC-Bd)-a^3(Bc+(A-C)d)+3ab^2(Bc+(A-C)d))\log(a\cos(e+}{(a^2+b^2)^3 f}$$
$$-\frac{(Ab^2-a(bB-aC))\,(bc-ad)}{2b^2\,(a^2+b^2)\,f(a+b\tan(e+fx))^2}$$
$$-\frac{a^4Cd+b^4(Bc+Ad)+2ab^3(Ac-cC-Bd)-a^2b^2(Bc+(A-3C)d)}{b^2\,(a^2+b^2)^2\,f(a+b\tan(e+fx))}$$

```
[Out] (a^3*(A*c-B*d-C*c)-3*a*b^2*(A*c-B*d-C*c)+3*a^2*b*(B*c+(A-C)*d)-b^3*(B*c+(A-
C)*d))*x/(a^2+b^2)^3+(3*a^2*b*(A*c-B*d-C*c)-b^3*(A*c-B*d-C*c)-a^3*(B*c+(A-C
)*d)+3*a*b^2*(B*c+(A-C)*d))*ln(a*cos(f*x+e)+b*sin(f*x+e))/(a^2+b^2)^3/f-1/2
*(A*b^2-a*(B*b-C*a))*(-a*d+b*c)/b^2/(a^2+b^2)/f/(a+b*tan(f*x+e))^2+(-a^4*C*
d-b^4*(A*d+B*c)-2*a*b^3*(A*c-B*d-C*c)+a^2*b^2*(B*c+(A-3*C)*d))/b^2/(a^2+b^2
)^2/f/(a+b*tan(f*x+e))
```

Rubi [A] (verified)

Time = 0.77 (sec) , antiderivative size = 320, normalized size of antiderivative = 1.00, number of steps used = 4, number of rules used = 4, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.093, Rules used = {3716, 3709, 3612, 3611}

$$\int \frac{(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^3}\,dx$$
$$= -\frac{(bc-ad)\left(Ab^2-a(bB-aC)\right)}{2b^2f\left(a^2+b^2\right)(a+b\tan(e+fx))^2}$$
$$-\frac{a^4Cd-a^2b^2(d(A-3C)+Bc)+2ab^3(Ac-Bd-cC)+b^4(Ad+Bc)}{b^2f\left(a^2+b^2\right)^2(a+b\tan(e+fx))}$$
$$+\frac{\left(-\left(a^3(d(A-C)+Bc)\right)+3a^2b(Ac-Bd-cC)+3ab^2(d(A-C)+Bc)-b^3(Ac-Bd-cC)\right)\log(a\,\mathrm{co}}{f\left(a^2+b^2\right)^3}$$
$$+\frac{x\left(a^3(Ac-Bd-cC)+3a^2b(d(A-C)+Bc)-3ab^2(Ac-Bd-cC)-b^3(d(A-C)+Bc)\right)}{\left(a^2+b^2\right)^3}$$

```
[In] Int[((c + d*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a + b*T
an[e + f*x])^3,x]
```

```
[Out] ((a^3*(A*c - c*C - B*d) - 3*a*b^2*(A*c - c*C - B*d) + 3*a^2*b*(B*c + (A - C
)*d) - b^3*(B*c + (A - C)*d))*x)/(a^2 + b^2)^3 + ((3*a^2*b*(A*c - c*C - B*d
) - b^3*(A*c - c*C - B*d) - a^3*(B*c + (A - C)*d) + 3*a*b^2*(B*c + (A - C)*
d))*Log[a*Cos[e + f*x] + b*Sin[e + f*x]])/((a^2 + b^2)^3*f) - ((A*b^2 - a*(
b*B - a*C))*(b*c - a*d))/(2*b^2*(a^2 + b^2)*f*(a + b*Tan[e + f*x])^2) - (a^
4*C*d + b^4*(B*c + A*d) + 2*a*b^3*(A*c - c*C - B*d) - a^2*b^2*(B*c + (A - 3
*C)*d))/(b^2*(a^2 + b^2)^2*f*(a + b*Tan[e + f*x]))
```

Rule 3611

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c/(b*f))*Log[RemoveContent[a*Cos[e + f*x] + b*Si
n[e + f*x], x]], x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3612

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[(a*c + b*d)*(x/(a^2 + b^2)), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3709

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 -
 a*b*B + a^2*C)*((a + b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1)*(a^2 + b^2))), x
] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[b*B + a*(A -
C) - (A*b - a*B - b*C)*Tan[e + f*x], x], x], x] /; FreeQ[{a, b, e, f, A, B,
 C}, x] && NeQ[A*b^2 - a*b*B + a^2*C, 0] && LtQ[m, -1] && NeQ[a^2 + b^2, 0]
```

Rule 3716

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.
)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(-(b*c - a*d))*(c^2*C - B*c*d + A*d^2)*((c
+ d*Tan[e + f*x])^(n + 1)/(d^2*f*(n + 1)*(c^2 + d^2))), x] + Dist[1/(d*(c^2
 + d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)*Simp[a*d*(A*c - c*C + B*d) + b*(
c^2*C - B*c*d + A*d^2) + d*(A*b*c + a*B*c - b*c*C - a*A*d + b*B*d + a*C*d)*
Tan[e + f*x] + b*C*(c^2 + d^2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c
, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] && LtQ[n,
 -1]
```

Rubi steps

integral

$$= -\frac{(Ab^2-a(bB-aC))\,(bc-ad)}{2b^2\,(a^2+b^2)\,f(a+b\tan(e+fx))^2} + \frac{\int \frac{a^2Cd+b^2(Bc+Ad)+ab(Ac-cC-Bd)-b(Abc-aBc-bcC-aAd-bBd+aCd)\tan(e+fx)+(a^2+b^2)Cd\tan^2(e+fx)}{(a+b\tan(e+fx))^2}\,dx}{b\,(a^2+b^2)}$$

$$= -\frac{(Ab^2-a(bB-aC))\,(bc-ad)}{2b^2\,(a^2+b^2)\,f(a+b\tan(e+fx))^2} - \frac{a^4Cd+b^4(Bc+Ad)+2ab^3(Ac-cC-Bd)-a^2b^2(Bc+(A-3C)d)}{b^2\,(a^2+b^2)^2\,f(a+b\tan(e+fx))} + \frac{\int \frac{b(a^2(Ac-cC-Bd)-b^2(Ac-cC-Bd)+2ab(Bc+(A-C)d))-b(2ab(Ac-cC-Bd)-a^2(Bc+(A-C)d)+b^2(Bc+(A-C)d))\tan(e+fx)}{a+b\tan(e+fx)}\,dx}{b\,(a^2+b^2)^2}$$

$$= \frac{(a^3(Ac-cC-Bd)-3ab^2(Ac-cC-Bd)+3a^2b(Bc+(A-C)d)-b^3(Bc+(A-C)d))\,x}{(a^2+b^2)^3} - \frac{(Ab^2-a(bB-aC))\,(bc-ad)}{2b^2\,(a^2+b^2)\,f(a+b\tan(e+fx))^2} - \frac{a^4Cd+b^4(Bc+Ad)+2ab^3(Ac-cC-Bd)-a^2b^2(Bc+(A-3C)d)}{b^2\,(a^2+b^2)^2\,f(a+b\tan(e+fx))} + \frac{(3a^2b(Ac-cC-Bd)-b^3(Ac-cC-Bd)-a^3(Bc+(A-C)d)+3ab^2(Bc+(A-C)d))\int \frac{b-a\tan(e+fx)}{a+b\tan(e+fx)}}{(a^2+b^2)^3}$$

$$= \frac{(a^3(Ac - cC - Bd) - 3ab^2(Ac - cC - Bd) + 3a^2b(Bc + (A - C)d) - b^3(Bc + (A - C)d))\, x}{(a^2 + b^2)^3}$$
$$+ \frac{(3a^2b(Ac - cC - Bd) - b^3(Ac - cC - Bd) - a^3(Bc + (A - C)d) + 3ab^2(Bc + (A - C)d)) \log(a \cos(e}{(a^2 + b^2)^3 f}$$
$$- \frac{(Ab^2 - a(bB - aC))\,(bc - ad)}{2b^2\,(a^2 + b^2)\, f(a + b \tan(e + fx))^2}$$
$$- \frac{a^4Cd + b^4(Bc + Ad) + 2ab^3(Ac - cC - Bd) - a^2b^2(Bc + (A - 3C)d)}{b^2\,(a^2 + b^2)^2\, f(a + b \tan(e + fx))}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 6.34 (sec) , antiderivative size = 379, normalized size of antiderivative = 1.18

$$\int \frac{(c + d \tan(e + fx))\,(A + B \tan(e + fx) + C \tan^2(e + fx))}{(a + b \tan(e + fx))^3}\, dx = -\frac{C(c + d \tan(e + fx))}{bf(a + b \tan(e + fx))^2}$$
$$- \frac{-\frac{bcC - bBd - aCd}{2bf(a + b\tan(e + fx))^2} + \frac{\frac{\left(-2b^3(Ac - cC - Bd) + 2ab^2(Bc + (A - C)d)\right)\left(-\frac{\log(i - \tan(e + fx))}{2(ia - b)^3} + \frac{\log(i + \tan(e + fx))}{2(ia + b)^3} + \frac{b\left(3a^2 - b^2\right)\log(a + b\tan(e + fx))}{\left(a^2 + b^2\right)^3} - \frac{}{2(a}\right.}{b}}{}}{}$$

```
[In] Integrate[((c + d*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a
 + b*Tan[e + f*x])^3,x]
```

```
[Out] -((C*(c + d*Tan[e + f*x]))/(b*f*(a + b*Tan[e + f*x])^2)) - (-1/2*(b*c*C - b
*B*d - a*C*d)/(b*f*(a + b*Tan[e + f*x])^2) + (((-2*b^3*(A*c - c*C - B*d) +
2*a*b^2*(B*c + (A - C)*d))*(-1/2*Log[I - Tan[e + f*x]]/(I*a - b)^3 + Log[I
+ Tan[e + f*x]]/(2*(I*a + b)^3) + (b*(3*a^2 - b^2)*Log[a + b*Tan[e + f*x]])
/(a^2 + b^2)^3 - b/(2*(a^2 + b^2)*(a + b*Tan[e + f*x])^2) - (2*a*b)/((a^2 +
 b^2)^2*(a + b*Tan[e + f*x]))))/b - 2*b*(B*c + (A - C)*d)*(((-1/2*I)*Log[I
- Tan[e + f*x]])/(a + I*b)^2 + ((I/2)*Log[I + Tan[e + f*x]])/(a - I*b)^2 +
(2*a*b*Log[a + b*Tan[e + f*x]])/(a^2 + b^2)^2 - b/((a^2 + b^2)*(a + b*Tan[e
 + f*x]))))/(2*b*f))/b
```

Maple [A] (verified)

Time = 0.20 (sec) , antiderivative size = 494, normalized size of antiderivative = 1.54

method	result
derivativedivides	$\frac{-\frac{-Aa\,b^2d+A\,b^3c+B\,a^2bd-Ba\,b^2c-a^3Cd+C\,a^2bc}{2b^2\left(a^2+b^2\right)(a+b\tan(fx+e))^2}-\frac{-A\,a^2b^2d+2Aa\,b^3c+A\,b^4d-B\,a^2b^2c-2Ba\,b^3d+B\,b^4c+a^4Cd+3C\,a^2b^2d-2Ca\,b^3}{\left(a^2+b^2\right)^2b^2(a+b\tan(fx+e))}}{}$
default	$\frac{-\frac{-Aa\,b^2d+A\,b^3c+B\,a^2bd-Ba\,b^2c-a^3Cd+C\,a^2bc}{2b^2\left(a^2+b^2\right)(a+b\tan(fx+e))^2}-\frac{-A\,a^2b^2d+2Aa\,b^3c+A\,b^4d-B\,a^2b^2c-2Ba\,b^3d+B\,b^4c+a^4Cd+3C\,a^2b^2d-2Ca\,b^3}{\left(a^2+b^2\right)^2b^2(a+b\tan(fx+e))}}{}$
norman	$\frac{\frac{\left(A\,a^2b^2d-2Aa\,b^3c-A\,b^4d+B\,a^2b^2c+2Ba\,b^3d-B\,b^4c-a^4Cd-3C\,a^2b^2d+2Ca\,b^3c\right)\tan(fx+e)}{fb\left(a^4+2a^2b^2+b^4\right)}+\frac{\left(A\,a^3c+3A\,a^2bd-3Aa\,b^2c-A\,b^3d-B\right.}{}}{}$
risch	Expression too large to display
parallelrisch	Expression too large to display

```
[In] int((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^3,x,m
ethod=_RETURNVERBOSE)
```

```
[Out] 1/f*(-1/2*(-A*a*b^2*d+A*b^3*c+B*a^2*b*d-B*a*b^2*c-C*a^3*d+C*a^2*b*c)/b^2/(a
^2+b^2)/(a+b*tan(f*x+e))^2-(-A*a^2*b^2*d+2*A*a*b^3*c+A*b^4*d-B*a^2*b^2*c-2*
B*a*b^3*d+B*b^4*c+C*a^4*d+3*C*a^2*b^2*d-2*C*a*b^3*c)/(a^2+b^2)^2/b^2/(a+b*t
an(f*x+e))-(A*a^3*d-3*A*a^2*b*c-3*A*a*b^2*d+A*b^3*c+B*a^3*c+3*B*a^2*b*d-3*B
*a*b^2*c-B*b^3*d-C*a^3*d+3*C*a^2*b*c+3*C*a*b^2*d-C*b^3*c)/(a^2+b^2)^3*ln(a+
b*tan(f*x+e))+1/(a^2+b^2)^3*(1/2*(A*a^3*d-3*A*a^2*b*c-3*A*a*b^2*d+A*b^3*c+B
*a^3*c+3*B*a^2*b*d-3*B*a*b^2*c-B*b^3*d-C*a^3*d+3*C*a^2*b*c+3*C*a*b^2*d-C*b^
3*c)*ln(1+tan(f*x+e)^2)+(A*a^3*c+3*A*a^2*b*d-3*A*a*b^2*c-A*b^3*d-B*a^3*d+3*
B*a^2*b*c+3*B*a*b^2*d-B*b^3*c-C*a^3*c-3*C*a^2*b*d+3*C*a*b^2*c+C*b^3*d)*arct
an(tan(f*x+e))))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 987 vs. 2(316) = 632.

Time = 0.30 (sec) , antiderivative size = 987, normalized size of antiderivative = 3.08

$$\int \frac{(c+d\tan(e+fx))\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx$$
$$=\frac{2\left(((A-C)a^5+3\,Ba^4b-3\,(A-C)a^3b^2-Ba^2b^3)c-(Ba^5-3\,(A-C)a^4b-3\,Ba^3b^2+(A-C)a^2b^3)d\right)f}{}$$

```
[In] integrate((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^3,x, algorithm="fricas")
```

```
[Out] 1/2*(2*(((A - C)*a^5 + 3*B*a^4*b - 3*(A - C)*a^3*b^2 - B*a^2*b^3)*c - (B*a^
5 - 3*(A - C)*a^4*b - 3*B*a^3*b^2 + (A - C)*a^2*b^3)*d)*f*x + (2*(((A - C)*
a^3*b^2 + 3*B*a^2*b^3 - 3*(A - C)*a*b^4 - B*b^5)*c - (B*a^3*b^2 - 3*(A - C)
*a^2*b^3 - 3*B*a*b^4 + (A - C)*b^5)*d)*f*x + (C*a^4*b - 3*B*a^3*b^2 + 5*(A
- C)*a^2*b^3 + 3*B*a*b^4 - A*b^5)*c + (C*a^5 + B*a^4*b - (3*A - 7*C)*a^3*b^
2 - 5*B*a^2*b^3 + 3*A*a*b^4)*d)*tan(f*x + e)^2 - (3*C*a^4*b - 5*B*a^3*b^2 +
 (7*A - 3*C)*a^2*b^3 + B*a*b^4 + A*b^5)*c + (C*a^5 - 3*B*a^4*b + 5*(A - C)*
a^3*b^2 + 3*B*a^2*b^3 - A*a*b^4)*d - (((B*a^3*b^2 - 3*(A - C)*a^2*b^3 - 3*B
*a*b^4 + (A - C)*b^5)*c + ((A - C)*a^3*b^2 + 3*B*a^2*b^3 - 3*(A - C)*a*b^4
- B*b^5)*d)*tan(f*x + e)^2 + (B*a^5 - 3*(A - C)*a^4*b - 3*B*a^3*b^2 + (A -
C)*a^2*b^3)*c + ((A - C)*a^5 + 3*B*a^4*b - 3*(A - C)*a^3*b^2 - B*a^2*b^3)*d
 + 2*((B*a^4*b - 3*(A - C)*a^3*b^2 - 3*B*a^2*b^3 + (A - C)*a*b^4)*c + ((A -
 C)*a^4*b + 3*B*a^3*b^2 - 3*(A - C)*a^2*b^3 - B*a*b^4)*d)*tan(f*x + e))*log
((b^2*tan(f*x + e)^2 + 2*a*b*tan(f*x + e) + a^2)/(tan(f*x + e)^2 + 1)) + 2*
(2*(((A - C)*a^4*b + 3*B*a^3*b^2 - 3*(A - C)*a^2*b^3 - B*a*b^4)*c - (B*a^4*
b - 3*(A - C)*a^3*b^2 - 3*B*a^2*b^3 + (A - C)*a*b^4)*d)*f*x + (C*a^5 - 2*B*
a^4*b + 3*(A - C)*a^3*b^2 + 3*B*a^2*b^3 - (3*A - 2*C)*a*b^4 - B*b^5)*c + (B
*a^5 - (2*A - 3*C)*a^4*b - 3*B*a^3*b^2 + 3*(A - C)*a^2*b^3 + 2*B*a*b^4 - A*
b^5)*d)*tan(f*x + e))/((a^6*b^2 + 3*a^4*b^4 + 3*a^2*b^6 + b^8)*f*tan(f*x +
e)^2 + 2*(a^7*b + 3*a^5*b^3 + 3*a^3*b^5 + a*b^7)*f*tan(f*x + e) + (a^8 + 3*
a^6*b^2 + 3*a^4*b^4 + a^2*b^6)*f)
```

Sympy [F(-2)]

Exception generated.

$$\int \frac{(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^3}\,dx$$
$$= \text{Exception raised: AttributeError}$$

```
[In] integrate((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e)
)**3,x)
```

```
[Out] Exception raised: AttributeError >> 'NoneType' object has no attribute 'pri
mitive'
```

Maxima [A] (verification not implemented)

none

Time = 0.42 (sec) , antiderivative size = 574, normalized size of antiderivative = 1.79

$$\int \frac{(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^3}\,dx$$
$$= \frac{\frac{2\left(\left(\left((A-C)a^3+3\,Ba^2b-3\,(A-C)ab^2-Bb^3\right)c-\left(Ba^3-3\,(A-C)a^2b-3\,Bab^2+(A-C)b^3\right)d\right)(fx+e)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6} - \frac{2\left(\left(Ba^3-3\,(A-C)a^2b-3\,Bab^2+(A-C)b^3\right)c\right.}{}}{}$$

```
[In] integrate((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^3,x, algorithm="maxima")
```

```
[Out] 1/2*(2*(((A - C)*a^3 + 3*B*a^2*b - 3*(A - C)*a*b^2 - B*b^3)*c - (B*a^3 - 3*
(A - C)*a^2*b - 3*B*a*b^2 + (A - C)*b^3)*d)*(f*x + e)/(a^6 + 3*a^4*b^2 + 3*
a^2*b^4 + b^6) - 2*((B*a^3 - 3*(A - C)*a^2*b - 3*B*a*b^2 + (A - C)*b^3)*c +
 ((A - C)*a^3 + 3*B*a^2*b - 3*(A - C)*a*b^2 - B*b^3)*d)*log(b*tan(f*x + e)
+ a)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) + ((B*a^3 - 3*(A - C)*a^2*b - 3*B*
a*b^2 + (A - C)*b^3)*c + ((A - C)*a^3 + 3*B*a^2*b - 3*(A - C)*a*b^2 - B*b^3
)*d)*log(tan(f*x + e)^2 + 1)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) - ((C*a^4*
b - 3*B*a^3*b^2 + (5*A - 3*C)*a^2*b^3 + B*a*b^4 + A*b^5)*c + (C*a^5 + B*a^4
*b - (3*A - 5*C)*a^3*b^2 - 3*B*a^2*b^3 + A*a*b^4)*d - 2*((B*a^2*b^3 - 2*(A
- C)*a*b^4 - B*b^5)*c - (C*a^4*b - (A - 3*C)*a^2*b^3 - 2*B*a*b^4 + A*b^5)*d
)*tan(f*x + e))/(a^6*b^2 + 2*a^4*b^4 + a^2*b^6 + (a^4*b^4 + 2*a^2*b^6 + b^8
)*tan(f*x + e)^2 + 2*(a^5*b^3 + 2*a^3*b^5 + a*b^7)*tan(f*x + e)))/f
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1006 vs. $2(316) = 632$.

Time = 0.81 (sec) , antiderivative size = 1006, normalized size of antiderivative = 3.14

$$\int \frac{(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^3}\,dx$$

$$= \frac{\frac{2\left(Aa^3c-Ca^3c+3\,Ba^2bc-3\,Aab^2c+3\,Cab^2c-Bb^3c-Ba^3d+3\,Aa^2bd-3\,Ca^2bd+3\,Bab^2d-Ab^3d+Cb^3d\right)(fx+e)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}+\frac{(Ba^3c-3\,Aa^2bc+3\,Ca^2bc-3\,B}{}}{}$$

```
[In] integrate((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^3,x, algorithm="giac")
```

```
[Out] 1/2*(2*(A*a^3*c - C*a^3*c + 3*B*a^2*b*c - 3*A*a*b^2*c + 3*C*a*b^2*c - B*b^3
*c - B*a^3*d + 3*A*a^2*b*d - 3*C*a^2*b*d + 3*B*a*b^2*d - A*b^3*d + C*b^3*d)
*(f*x + e)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) + (B*a^3*c - 3*A*a^2*b*c + 3
*C*a^2*b*c - 3*B*a*b^2*c + A*b^3*c - C*b^3*c + A*a^3*d - C*a^3*d + 3*B*a^2*
b*d - 3*A*a*b^2*d + 3*C*a*b^2*d - B*b^3*d)*log(tan(f*x + e)^2 + 1)/(a^6 + 3
*a^4*b^2 + 3*a^2*b^4 + b^6) - 2*(B*a^3*b*c - 3*A*a^2*b^2*c + 3*C*a^2*b^2*c
- 3*B*a*b^3*c + A*b^4*c - C*b^4*c + A*a^3*b*d - C*a^3*b*d + 3*B*a^2*b^2*d -
 3*A*a*b^3*d + 3*C*a*b^3*d - B*b^4*d)*log(abs(b*tan(f*x + e) + a))/(a^6*b +
 3*a^4*b^3 + 3*a^2*b^5 + b^7) + (3*B*a^3*b^4*c*tan(f*x + e)^2 - 9*A*a^2*b^5
*c*tan(f*x + e)^2 + 9*C*a^2*b^5*c*tan(f*x + e)^2 - 9*B*a*b^6*c*tan(f*x + e)
^2 + 3*A*b^7*c*tan(f*x + e)^2 - 3*C*b^7*c*tan(f*x + e)^2 + 3*A*a^3*b^4*d*ta
n(f*x + e)^2 - 3*C*a^3*b^4*d*tan(f*x + e)^2 + 9*B*a^2*b^5*d*tan(f*x + e)^2
- 9*A*a*b^6*d*tan(f*x + e)^2 + 9*C*a*b^6*d*tan(f*x + e)^2 - 3*B*b^7*d*tan(f
*x + e)^2 + 8*B*a^4*b^3*c*tan(f*x + e) - 22*A*a^3*b^4*c*tan(f*x + e) + 22*C
*a^3*b^4*c*tan(f*x + e) - 18*B*a^2*b^5*c*tan(f*x + e) + 2*A*a*b^6*c*tan(f*x
```

```
 + e) - 2*C*a*b^6*c*tan(f*x + e) - 2*B*b^7*c*tan(f*x + e) - 2*C*a^6*b*d*tan
(f*x + e) + 8*A*a^4*b^3*d*tan(f*x + e) - 14*C*a^4*b^3*d*tan(f*x + e) + 22*B
*a^3*b^4*d*tan(f*x + e) - 18*A*a^2*b^5*d*tan(f*x + e) + 12*C*a^2*b^5*d*tan(
f*x + e) - 2*B*a*b^6*d*tan(f*x + e) - 2*A*b^7*d*tan(f*x + e) - C*a^6*b*c +
6*B*a^5*b^2*c - 14*A*a^4*b^3*c + 11*C*a^4*b^3*c - 7*B*a^3*b^4*c - 3*A*a^2*b
^5*c - B*a*b^6*c - A*b^7*c - C*a^7*d - B*a^6*b*d + 6*A*a^5*b^2*d - 9*C*a^5*
b^2*d + 11*B*a^4*b^3*d - 7*A*a^3*b^4*d + 4*C*a^3*b^4*d - A*a*b^6*d)/((a^6*b
^2 + 3*a^4*b^4 + 3*a^2*b^6 + b^8)*(b*tan(f*x + e) + a)^2))/f
```

Mupad [B] (verification not implemented)

Time = 15.53 (sec) , antiderivative size = 502, normalized size of antiderivative = 1.57

$$\int \frac{(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^3}\,dx = -\frac{\frac{A\,b^5\,c+C\,a^5\,d+A\,a\,b^4\,d+B\,a\,b^4\,c+B\,a^4\,b\,d+C\,a^4\,b\,c+5\,A\,a^2\,b^3\,c-3\,A\,a^3\,b^2\,d-3\,B\,a^3\,b^2\,c-3\,B\,a^2\,b^3\,d-3\,C\,a^2\,b^3\,c+5\,C\,a^3\,b^2\,d}{2\,b^2\,\left(a^4+2\,a^2\,b^2+b^4\right)}+\frac{\tan(e+f\,x)\,(}{}}{f\,\left(a^2+2\,a\,b\tan\left(e+f\,x\right)+b^2\tan\left(e+f\right.\right.} - \frac{\ln\left(\tan(e+f\,x)+1\mathrm{i}\right)\,\left(B\,d+A\,d\,1\mathrm{i}+B\,c\,1\mathrm{i}-A\,c+C\,c-C\,d\,1\mathrm{i}\right)}{2\,f\,\left(-a^3\,1\mathrm{i}-3\,a^2\,b+a\,b^2\,3\mathrm{i}+b^3\right)} - \frac{\ln\left(\tan(e+f\,x)-\mathrm{i}\right)\,\left(A\,d+B\,c-C\,d-A\,c\,1\mathrm{i}+B\,d\,1\mathrm{i}+C\,c\,1\mathrm{i}\right)}{2\,f\,\left(-a^3-a^2\,b\,3\mathrm{i}+3\,a\,b^2+b^3\,1\mathrm{i}\right)} - \frac{\ln\left(a+b\tan(e+f\,x)\right)\,\left(\left(A\,d+B\,c-C\,d\right)\,a^3+\left(3\,B\,d-3\,A\,c+3\,C\,c\right)\,a^2\,b+\left(3\,C\,d-3\,B\,c-3\,A\,d\right)\,a\right.}{f\,\left(a^6+3\,a^4\,b^2+3\,a^2\,b^4+b^6\right)}$$

```
[In] int(((c + d*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a + b*t
an(e + f*x))^3,x)
```

```
[Out] - ((A*b^5*c + C*a^5*d + A*a*b^4*d + B*a*b^4*c + B*a^4*b*d + C*a^4*b*c + 5*A
*a^2*b^3*c - 3*A*a^3*b^2*d - 3*B*a^3*b^2*c - 3*B*a^2*b^3*d - 3*C*a^2*b^3*c
+ 5*C*a^3*b^2*d)/(2*b^2*(a^4 + b^4 + 2*a^2*b^2)) + (tan(e + f*x)*(A*b^4*d +
 B*b^4*c + C*a^4*d + 2*A*a*b^3*c - 2*B*a*b^3*d - 2*C*a*b^3*c - A*a^2*b^2*d
- B*a^2*b^2*c + 3*C*a^2*b^2*d))/(b*(a^4 + b^4 + 2*a^2*b^2)))/(f*(a^2 + b^2*
tan(e + f*x)^2 + 2*a*b*tan(e + f*x))) - (log(tan(e + f*x) + 1i)*(A*d*1i - A
*c + B*c*1i + B*d + C*c - C*d*1i))/(2*f*(a*b^2*3i - 3*a^2*b - a^3*1i + b^3)
) - (log(tan(e + f*x) - 1i)*(A*d - A*c*1i + B*c + B*d*1i + C*c*1i - C*d))/(
2*f*(3*a*b^2 - a^2*b*3i - a^3 + b^3*1i)) - (log(a + b*tan(e + f*x))*(a^3*(A
*d + B*c - C*d) - b^3*(B*d - A*c + C*c) + a^2*b*(3*B*d - 3*A*c + 3*C*c) - a
*b^2*(3*A*d + 3*B*c - 3*C*d)))/(f*(a^6 + b^6 + 3*a^2*b^4 + 3*a^4*b^2))
```

3.57 $\int (a+b\tan(e+fx))^3(c+d\tan(e+fx))^2 (A+B\tan(e+$

Optimal result 478
Rubi [A] (verified) 479
Mathematica [C] (verified) 483
Maple [A] (verified) 484
Fricas [A] (verification not implemented) 484
Sympy [B] (verification not implemented) 485
Maxima [A] (verification not implemented) 486
Giac [B] (verification not implemented) 487
Mupad [B] (verification not implemented) 500

Optimal result

Integrand size = 45, antiderivative size = 661

$$\int (a+b\tan(e+fx))^3(c+d\tan(e+fx))^2 \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right) dx$$
$$= -\left(\left(a^3\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-3ab^2\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)\right.\right.$$
$$\left.\left.+3a^2b\left(2c(A-C)d+B\left(c^2-d^2\right)\right)-b^3\left(2c(A-C)d+B\left(c^2-d^2\right)\right)\right)x\right)$$
$$+\frac{(3a^2b(c^2C+2Bcd-Cd^2-A(c^2-d^2))-b^3(c^2C+2Bcd-Cd^2-A(c^2-d^2))-a^3(2c(A-C)d+B(c}{f}$$
$$+\frac{d(3a^2b(Ac-cC-Bd)-b^3(Ac-cC-Bd)+a^3(Bc+(A-C)d)-3ab^2(Bc+(A-C)d))\tan(e+fx}{f}$$
$$+\frac{(a^3B-3ab^2B+3a^2b(A-C)-b^3(A-C))\,(c+d\tan(e+fx))^2}{2f}$$
$$+\frac{(4a^3Cd^3-3a^2bd^2(3cC-16Bd)+3ab^2d(2c^2C-5Bcd+20(A-C)d^2)-b^3(c^3C-2Bc^2d+5c(A-C)}{60d^4f}$$
$$+\frac{b(5b(Ab+aB-bC)d^2+(bc-ad)(bcC-2bBd-aCd))\tan(e+fx)(c+d\tan(e+fx))^3}{20d^3f}$$
$$-\frac{(bcC-2bBd-aCd)(a+b\tan(e+fx))^2(c+d\tan(e+fx))^3}{10d^2f}$$
$$+\frac{C(a+b\tan(e+fx))^3(c+d\tan(e+fx))^3}{6df}$$

```
[Out] -(a^3*(c^2*C+2*B*c*d-C*d^2-A*(c^2-d^2))-3*a*b^2*(c^2*C+2*B*c*d-C*d^2-A*(c^2
-d^2))+3*a^2*b*(2*c*(A-C)*d+B*(c^2-d^2))-b^3*(2*c*(A-C)*d+B*(c^2-d^2)))*x+(
3*a^2*b*(c^2*C+2*B*c*d-C*d^2-A*(c^2-d^2))-b^3*(c^2*C+2*B*c*d-C*d^2-A*(c^2-d
^2))-a^3*(2*c*(A-C)*d+B*(c^2-d^2))+3*a*b^2*(2*c*(A-C)*d+B*(c^2-d^2)))*ln(co
s(f*x+e))/f+d*(3*a^2*b*(A*c-B*d-C*c)-b^3*(A*c-B*d-C*c)+a^3*(B*c+(A-C)*d)-3*
a*b^2*(B*c+(A-C)*d))*tan(f*x+e)/f+1/2*(B*a^3-3*B*a*b^2+3*a^2*b*(A-C)-b^3*(A
-C))*(c+d*tan(f*x+e))^2/f+1/60*(4*a^3*C*d^3-3*a^2*b*d^2*(-16*B*d+3*C*c)+3*a
```

```
*b^2*d*(2*c^2*C-5*B*c*d+20*(A-C)*d^2)-b^3*(c^3*C-2*B*c^2*d+5*c*(A-C)*d^2+20
*B*d^3))*(c+d*tan(f*x+e))^3/d^4/f+1/20*b*(5*b*(A*b+B*a-C*b)*d^2+(-a*d+b*c)*
(-2*B*b*d-C*a*d+C*b*c))*tan(f*x+e)*(c+d*tan(f*x+e))^3/d^3/f-1/10*(-2*B*b*d-
C*a*d+C*b*c)*(a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^3/d^2/f+1/6*C*(a+b*tan(f*x
+e))^3*(c+d*tan(f*x+e))^3/d/f
```

Rubi [A] (verified)

Time = 2.26 (sec) , antiderivative size = 661, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}} = 0.133$, Rules used = {3728, 3718, 3711, 3609, 3606, 3556}

$$\int (a+b\tan(e+fx))^3(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\frac{\log(\cos(e+fx))\left(-(a^3(2cd(A-C)+B(c^2-d^2)))+3a^2b(-A(c^2-d^2)+2Bcd+c^2C-Cd^2)+3ab^2(2c\right.}{f}$$
$$-x\left(a^3\left(-A\left(c^2-d^2\right)+2Bcd+c^2C-Cd^2\right)+3a^2b\left(2cd(A-C)+B\left(c^2-d^2\right)\right)\right.$$
$$\left.-3ab^2\left(-A\left(c^2-d^2\right)+2Bcd+c^2C-Cd^2\right)-b^3\left(2cd(A-C)+B\left(c^2-d^2\right)\right)\right)$$
$$+\frac{(c+d\tan(e+fx))^3\left(4a^3Cd^3-3a^2bd^2(3cC-16Bd)+3ab^2d(20d^2(A-C)-5Bcd+2c^2C)-(b^3(5cd\right.}{60d^4f}$$
$$+\frac{(a^3B+3a^2b(A-C)-3ab^2B-b^3(A-C))\,(c+d\tan(e+fx))^2}{2f}$$
$$+\frac{d\tan(e+fx)\left(a^3(d(A-C)+Bc)+3a^2b(Ac-Bd-cC)-3ab^2(d(A-C)+Bc)-b^3(Ac-Bd-cC\right.}{f}$$
$$+\frac{b\tan(e+fx)(c+d\tan(e+fx))^3\left(5bd^2(aB+Ab-bC)+(bc-ad)(-aCd-2bBd+bcC)\right)}{20d^3f}$$
$$-\frac{(-aCd-2bBd+bcC)(a+b\tan(e+fx))^2(c+d\tan(e+fx))^3}{10d^2f}$$
$$+\frac{C(a+b\tan(e+fx))^3(c+d\tan(e+fx))^3}{6df}$$

```
[In] Int[(a + b*Tan[e + f*x])^3*(c + d*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*T
an[e + f*x]^2),x]
```

```
[Out] -((a^3*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - 3*a*b^2*(c^2*C + 2*B*c*d
 - C*d^2 - A*(c^2 - d^2)) + 3*a^2*b*(2*c*(A - C)*d + B*(c^2 - d^2)) - b^3*(
2*c*(A - C)*d + B*(c^2 - d^2)))*x) + ((3*a^2*b*(c^2*C + 2*B*c*d - C*d^2 - A
*(c^2 - d^2)) - b^3*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - a^3*(2*c*(A
 - C)*d + B*(c^2 - d^2)) + 3*a*b^2*(2*c*(A - C)*d + B*(c^2 - d^2)))*Log[Cos
[e + f*x]])/f + (d*(3*a^2*b*(A*c - c*C - B*d) - b^3*(A*c - c*C - B*d) + a^3
*(B*c + (A - C)*d) - 3*a*b^2*(B*c + (A - C)*d))*Tan[e + f*x])/f + ((a^3*B -
 3*a*b^2*B + 3*a^2*b*(A - C) - b^3*(A - C))*(c + d*Tan[e + f*x])^2)/(2*f) +
 ((4*a^3*C*d^3 - 3*a^2*b*d^2*(3*c*C - 16*B*d) + 3*a*b^2*d*(2*c^2*C - 5*B*c*
```

```
d + 20*(A - C)*d^2) - b^3*(c^3*C - 2*B*c^2*d + 5*c*(A - C)*d^2 + 20*B*d^3))
*(c + d*Tan[e + f*x])^3)/(60*d^4*f) + (b*(5*b*(A*b + a*B - b*C)*d^2 + (b*c
- a*d)*(b*c*C - 2*b*B*d - a*C*d))*Tan[e + f*x]*(c + d*Tan[e + f*x])^3)/(20*
d^3*f) - ((b*c*C - 2*b*B*d - a*C*d)*(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f
*x])^3)/(10*d^2*f) + (C*(a + b*Tan[e + f*x])^3*(c + d*Tan[e + f*x])^3)/(6*d
*f)
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3606

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
 f*x], x], x] + Simp[b*d*(Tan[e + f*x]/f), x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3609

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[d*((a + b*Tan[e + f*x])^m/(f*m)), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\text{integral} = \frac{C(a+b\tan(e+fx))^3(c+d\tan(e+fx))^3}{6df} + \frac{\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^2\left(-3(bcC-a(2A-C)d)+6(Ab+aB-bC)d\tan(e+fx)\right.}{6d}$$

$$= -\frac{(bcC-2bBd-aCd)(a+b\tan(e+fx))^2(c+d\tan(e+fx))^3}{10d^2f} + \frac{C(a+b\tan(e+fx))^3(c+d\tan(e+fx))^3}{6df} + \frac{\int (a+b\tan(e+fx))(c+d\tan(e+fx))^2\left(6(a^2(5A-4C)d^2+b^2c(cC-2Bd)-abd(2cC+3B\right.}{}$$

$$= \frac{b(5b(Ab+aB-bC)d^2+(bc-ad)(bcC-2bBd-aCd))\tan(e+fx)(c+d\tan(e+fx))^3}{20d^3f} - \frac{(bcC-2bBd-aCd)(a+b\tan(e+fx))^2(c+d\tan(e+fx))^3}{10d^2f} + \frac{C(a+b\tan(e+fx))^3(c+d\tan(e+fx))^3}{6df} - \frac{\int (c+d\tan(e+fx))^2\left(-6(4a^3(5A-4C)d^3+3ab^2cd(2cC-5Bd)-3a^2bd^2(3cC+4Bd)-b^3c\right.}{}$$

$$= \frac{(4a^3Cd^3-3a^2bd^2(3cC-16Bd)+3ab^2d(2c^2C-5Bcd+20(A-C)d^2)-b^3(c^3C-2Bc^2d+5c(A}{60d^4f} + \frac{b(5b(Ab+aB-bC)d^2+(bc-ad)(bcC-2bBd-aCd))\tan(e+fx)(c+d\tan(e+fx))^3}{20d^3f} - \frac{(bcC-2bBd-aCd)(a+b\tan(e+fx))^2(c+d\tan(e+fx))^3}{10d^2f} + \frac{C(a+b\tan(e+fx))^3(c+d\tan(e+fx))^3}{6df} - \frac{\int (c+d\tan(e+fx))^2\left(120(3a^2bB-b^3B-a^3(A-C)+3ab^2(A-C))d^3-120(a^3B-3ab^2B\right.}{120d^3}$$

$$= \frac{(a^3B - 3ab^2B + 3a^2b(A-C) - b^3(A-C))\,(c + d\tan(e+fx))^2}{2f}$$
$$+ \frac{(4a^3Cd^3 - 3a^2bd^2(3cC - 16Bd) + 3ab^2d(2c^2C - 5Bcd + 20(A-C)d^2) - b^3(c^3C - 2Bc^2d + 5c(}{60d^4f}$$
$$+ \frac{b(5b(Ab + aB - bC)d^2 + (bc - ad)(bcC - 2bBd - aCd))\tan(e+fx)(c + d\tan(e+fx))^3}{20d^3f}$$
$$- \frac{(bcC - 2bBd - aCd)(a + b\tan(e+fx))^2(c + d\tan(e+fx))^3}{10d^2f}$$
$$+ \frac{C(a + b\tan(e+fx))^3(c + d\tan(e+fx))^3}{6df}$$
$$- \frac{\int (c + d\tan(e+fx))\,(-120d^3(a^3(Ac - cC - Bd) - 3ab^2(Ac - cC - Bd) - 3a^2b(Bc + (A-C)c}{}$$

$$= -\Big(\big(a^3\big(c^2C + 2Bcd - Cd^2 - A\big(c^2 - d^2\big)\big) - 3ab^2\big(c^2C + 2Bcd - Cd^2 - A\big(c^2 - d^2\big)\big)$$
$$+ 3a^2b\big(2c(A-C)d + B\big(c^2 - d^2\big)\big) - b^3\big(2c(A-C)d + B\big(c^2 - d^2\big)\big)\big)\,x\Big)$$
$$+ \frac{d(3a^2b(Ac - cC - Bd) - b^3(Ac - cC - Bd) + a^3(Bc + (A-C)d) - 3ab^2(Bc + (A-C)d))\tan}{f}$$
$$+ \frac{(a^3B - 3ab^2B + 3a^2b(A-C) - b^3(A-C))\,(c + d\tan(e+fx))^2}{2f}$$
$$+ \frac{(4a^3Cd^3 - 3a^2bd^2(3cC - 16Bd) + 3ab^2d(2c^2C - 5Bcd + 20(A-C)d^2) - b^3(c^3C - 2Bc^2d + 5c(}{60d^4f}$$
$$+ \frac{b(5b(Ab + aB - bC)d^2 + (bc - ad)(bcC - 2bBd - aCd))\tan(e+fx)(c + d\tan(e+fx))^3}{20d^3f}$$
$$- \frac{(bcC - 2bBd - aCd)(a + b\tan(e+fx))^2(c + d\tan(e+fx))^3}{10d^2f}$$
$$+ \frac{C(a + b\tan(e+fx))^3(c + d\tan(e+fx))^3}{6df}$$
$$- \big(3a^2b\big(c^2C + 2Bcd - Cd^2 - A\big(c^2 - d^2\big)\big) - b^3\big(c^2C + 2Bcd - Cd^2 - A\big(c^2 - d^2\big)\big)$$
$$- a^3\big(2c(A-C)d + B\big(c^2 - d^2\big)\big) + 3ab^2\big(2c(A-C)d + B\big(c^2 - d^2\big)\big)\big) \int \tan(e+fx)\,dx$$

$$= -\left(\left(a^3\left(c^2C + 2Bcd - Cd^2 - A\left(c^2 - d^2\right)\right) - 3ab^2\left(c^2C + 2Bcd - Cd^2 - A\left(c^2 - d^2\right)\right) + 3a^2b\left(2c(A - C)d + B\left(c^2 - d^2\right)\right) - b^3\left(2c(A - C)d + B\left(c^2 - d^2\right)\right)\right)x\right)$$
$$+ \frac{(3a^2b(c^2C + 2Bcd - Cd^2 - A(c^2 - d^2)) - b^3(c^2C + 2Bcd - Cd^2 - A(c^2 - d^2)) - a^3(2c(A - C}{f}$$
$$+ \frac{d(3a^2b(Ac - cC - Bd) - b^3(Ac - cC - Bd) + a^3(Bc + (A - C)d) - 3ab^2(Bc + (A - C)d))\,\mathrm{ta}}{f}$$
$$+ \frac{(a^3B - 3ab^2B + 3a^2b(A - C) - b^3(A - C))\,(c + d\tan(e + fx))^2}{2f}$$
$$+ \frac{(4a^3Cd^3 - 3a^2bd^2(3cC - 16Bd) + 3ab^2d(2c^2C - 5Bcd + 20(A - C)d^2) - b^3(c^3C - 2Bc^2d + 5}{60d^4f}$$
$$+ \frac{b(5b(Ab + aB - bC)d^2 + (bc - ad)(bcC - 2bBd - aCd))\tan(e + fx)(c + d\tan(e + fx))^3}{20d^3f}$$
$$- \frac{(bcC - 2bBd - aCd)(a + b\tan(e + fx))^2(c + d\tan(e + fx))^3}{10d^2f}$$
$$+ \frac{C(a + b\tan(e + fx))^3(c + d\tan(e + fx))^3}{6df}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 6.74 (sec) , antiderivative size = 573, normalized size of antiderivative = 0.87

$$\int (a + b\tan(e + fx))^3(c + d\tan(e + fx))^2 \left(A + B\tan(e + fx) + C\tan^2(e + fx)\right)\,dx$$
$$= \frac{C(a + b\tan(e + fx))^3(c + d\tan(e + fx))^3}{6df}$$
$$+ \frac{-\frac{3(bcC - 2bBd - aCd)(a + b\tan(e + fx))^2(c + d\tan(e + fx))^3}{5df} + \frac{\frac{3b\left(5b(Ab + aB - bC)d^2 + (bc - ad)(bcC - 2bBd - aCd)\right)\tan(e + fx)(c + d\tan(e + fx))^3}{2df} - \frac{(-}{}}{}}{}$$

```
[In] Integrate[(a + b*Tan[e + f*x])^3*(c + d*Tan[e + f*x])^2*(A + B*Tan[e + f*x]
 + C*Tan[e + f*x]^2),x]
```

```
[Out] (C*(a + b*Tan[e + f*x])^3*(c + d*Tan[e + f*x])^3)/(6*d*f) + ((-3*(b*c*C - 2
*b*B*d - a*C*d)*(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])^3)/(5*d*f) + ((
3*b*(5*b*(A*b + a*B - b*C)*d^2 + (b*c - a*d)*(b*c*C - 2*b*B*d - a*C*d))*Tan
[e + f*x]*(c + d*Tan[e + f*x])^3)/(2*d*f) - (((-24*a*d*(5*b*(A*b + a*B - b*
C)*d^2 + (b*c - a*d)*(b*c*C - 2*b*B*d - a*C*d)) + b*(-120*(a^2*B - b^2*B +
2*a*b*(A - C))*d^3 + 6*c*(5*b*(A*b + a*B - b*C)*d^2 + (b*c - a*d)*(b*c*C -
2*b*B*d - a*C*d))))*(c + d*Tan[e + f*x])^3)/(3*d*f) - (60*(d^2*(3*a^2*b*(A*
c - c*C + B*d) - b^3*(A*c - c*C + B*d) + a^3*(B*c - (A - C)*d) - 3*a*b^2*(B
*c - (A - C)*d))*(I*(c + I*d)^2*Log[I - Tan[e + f*x]] - I*(c - I*d)^2*Log[I
 + Tan[e + f*x]] - 2*d^2*Tan[e + f*x]) + (a^3*B - 3*a*b^2*B + 3*a^2*b*(A -
```

```
C) - b^3*(A - C))*d^2*((I*c - d)^3*Log[I - Tan[e + f*x]] - (I*c + d)^3*Log[
I + Tan[e + f*x]] + 6*c*d^2*Tan[e + f*x] + d^3*Tan[e + f*x]^2)))/f)/(4*d))/
(5*d))/(6*d)
```

Maple [A] (verified)

Time = 0.45 (sec) , antiderivative size = 546, normalized size of antiderivative = 0.83

method	result
parts	$\frac{(2A\,a^3cd+3A\,a^2b\,c^2+B\,a^3c^2)\ln\left(1+\tan(fx+e)^2\right)}{2f}+\frac{(B\,b^3d^2+3Ca\,b^2d^2+2C\,b^3cd)\left(\frac{\tan(fx+e)^5}{5}-\frac{\tan(fx+e)^3}{3}+\tan(fx+\right.}{f}$
norman	$(A\,a^3c^2-A\,a^3d^2-6A\,a^2bcd-3Aa\,b^2c^2+3Aa\,b^2d^2+2A\,b^3cd-2B\,a^3cd-3B\,a^2b\,c^2+$
derivativedivides	Expression too large to display
default	Expression too large to display
parallelrisch	Expression too large to display
risch	Expression too large to display

```
[In] int((a+b*tan(f*x+e))^3*(c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x
,method=_RETURNVERBOSE)
```

```
[Out] 1/2*(2*A*a^3*c*d+3*A*a^2*b*c^2+B*a^3*c^2)/f*ln(1+tan(f*x+e)^2)+(B*b^3*d^2+3
*C*a*b^2*d^2+2*C*b^3*c*d)/f*(1/5*tan(f*x+e)^5-1/3*tan(f*x+e)^3+tan(f*x+e)-a
rctan(tan(f*x+e)))+(A*b^3*d^2+3*B*a*b^2*d^2+2*B*b^3*c*d+3*C*a^2*b*d^2+6*C*a
*b^2*c*d+C*b^3*c^2)/f*(1/4*tan(f*x+e)^4-1/2*tan(f*x+e)^2+1/2*ln(1+tan(f*x+e
)^2))+(A*a^3*d^2+6*A*a^2*b*c*d+3*A*a*b^2*c^2+2*B*a^3*c*d+3*B*a^2*b*c^2+C*a^
3*c^2)/f*(tan(f*x+e)-arctan(tan(f*x+e)))+(3*A*a*b^2*d^2+2*A*b^3*c*d+3*B*a^2
*b*d^2+6*B*a*b^2*c*d+B*b^3*c^2+C*a^3*d^2+6*C*a^2*b*c*d+3*C*a*b^2*c^2)/f*(1/
3*tan(f*x+e)^3-tan(f*x+e)+arctan(tan(f*x+e)))+(3*A*a^2*b*d^2+6*A*a*b^2*c*d+
A*b^3*c^2+B*a^3*d^2+6*B*a^2*b*c*d+3*B*a*b^2*c^2+2*C*a^3*c*d+3*C*a^2*b*c^2)/
f*(1/2*tan(f*x+e)^2-1/2*ln(1+tan(f*x+e)^2))+A*a^3*c^2*x+C*b^3*d^2/f*(1/6*ta
n(f*x+e)^6-1/4*tan(f*x+e)^4+1/2*tan(f*x+e)^2-1/2*ln(1+tan(f*x+e)^2))
```

Fricas [A] (verification not implemented)

none

Time = 0.28 (sec) , antiderivative size = 690, normalized size of antiderivative = 1.04

$$\int (a+b\tan(e+fx))^3(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\frac{10\,Cb^3d^2\tan\left(fx+e\right)^6+12\left(2\,Cb^3cd+\left(3\,Cab^2+Bb^3\right)d^2\right)\tan\left(fx+e\right)^5+15\left(Cb^3c^2+2\left(3\,Cab^2+Bb^3\right)cd\right.}{}$$

```
[In] integrate((a+b*tan(f*x+e))^3*(c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e
)^2),x, algorithm="fricas")
```

```
[Out] 1/60*(10*C*b^3*d^2*tan(f*x + e)^6 + 12*(2*C*b^3*c*d + (3*C*a*b^2 + B*b^3)*d
^2)*tan(f*x + e)^5 + 15*(C*b^3*c^2 + 2*(3*C*a*b^2 + B*b^3)*c*d + (3*C*a^2*b
 + 3*B*a*b^2 + (A - C)*b^3)*d^2)*tan(f*x + e)^4 + 20*((3*C*a*b^2 + B*b^3)*c
^2 + 2*(3*C*a^2*b + 3*B*a*b^2 + (A - C)*b^3)*c*d + (C*a^3 + 3*B*a^2*b + 3*(
A - C)*a*b^2 - B*b^3)*d^2)*tan(f*x + e)^3 + 60*(((A - C)*a^3 - 3*B*a^2*b -
3*(A - C)*a*b^2 + B*b^3)*c^2 - 2*(B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A
- C)*b^3)*c*d - ((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b^3)*d^2)*f*
x + 30*((3*C*a^2*b + 3*B*a*b^2 + (A - C)*b^3)*c^2 + 2*(C*a^3 + 3*B*a^2*b +
3*(A - C)*a*b^2 - B*b^3)*c*d + (B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A -
C)*b^3)*d^2)*tan(f*x + e)^2 - 30*((B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A
 - C)*b^3)*c^2 + 2*((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b^3)*c*d
- (B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*d^2)*log(1/(tan(f*x +
 e)^2 + 1)) + 60*((C*a^3 + 3*B*a^2*b + 3*(A - C)*a*b^2 - B*b^3)*c^2 + 2*(B*
a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*c*d + ((A - C)*a^3 - 3*B*a
^2*b - 3*(A - C)*a*b^2 + B*b^3)*d^2)*tan(f*x + e))/f
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1819 vs. $2(604) = 1208$.

Time = 0.39 (sec) , antiderivative size = 1819, normalized size of antiderivative = 2.75

$$\int (a+b\tan(e+fx))^3(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= \text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))**3*(c+d*tan(f*x+e))**2*(A+B*tan(f*x+e)+C*tan(f*x
+e)**2),x)
```

```
[Out] Piecewise((A*a**3*c**2*x + A*a**3*c*d*log(tan(e + f*x)**2 + 1)/f - A*a**3*d
**2*x + A*a**3*d**2*tan(e + f*x)/f + 3*A*a**2*b*c**2*log(tan(e + f*x)**2 +
1)/(2*f) - 6*A*a**2*b*c*d*x + 6*A*a**2*b*c*d*tan(e + f*x)/f - 3*A*a**2*b*d*
*2*log(tan(e + f*x)**2 + 1)/(2*f) + 3*A*a**2*b*d**2*tan(e + f*x)**2/(2*f) -
 3*A*a*b**2*c**2*x + 3*A*a*b**2*c**2*tan(e + f*x)/f - 3*A*a*b**2*c*d*log(ta
n(e + f*x)**2 + 1)/f + 3*A*a*b**2*c*d*tan(e + f*x)**2/f + 3*A*a*b**2*d**2*x
 + A*a*b**2*d**2*tan(e + f*x)**3/f - 3*A*a*b**2*d**2*tan(e + f*x)/f - A*b**
3*c**2*log(tan(e + f*x)**2 + 1)/(2*f) + A*b**3*c**2*tan(e + f*x)**2/(2*f) +
 2*A*b**3*c*d*x + 2*A*b**3*c*d*tan(e + f*x)**3/(3*f) - 2*A*b**3*c*d*tan(e +
 f*x)/f + A*b**3*d**2*log(tan(e + f*x)**2 + 1)/(2*f) + A*b**3*d**2*tan(e +
f*x)**4/(4*f) - A*b**3*d**2*tan(e + f*x)**2/(2*f) + B*a**3*c**2*log(tan(e +
 f*x)**2 + 1)/(2*f) - 2*B*a**3*c*d*x + 2*B*a**3*c*d*tan(e + f*x)/f - B*a**3
*d**2*log(tan(e + f*x)**2 + 1)/(2*f) + B*a**3*d**2*tan(e + f*x)**2/(2*f) -
3*B*a**2*b*c**2*x + 3*B*a**2*b*c**2*tan(e + f*x)/f - 3*B*a**2*b*c*d*log(tan
```

```
(e + f*x)**2 + 1)/f + 3*B*a**2*b*c*d*tan(e + f*x)**2/f + 3*B*a**2*b*d**2*x
+ B*a**2*b*d**2*tan(e + f*x)**3/f - 3*B*a**2*b*d**2*tan(e + f*x)/f - 3*B*a*
b**2*c**2*log(tan(e + f*x)**2 + 1)/(2*f) + 3*B*a*b**2*c**2*tan(e + f*x)**2/
(2*f) + 6*B*a*b**2*c*d*x + 2*B*a*b**2*c*d*tan(e + f*x)**3/f - 6*B*a*b**2*c*
d*tan(e + f*x)/f + 3*B*a*b**2*d**2*log(tan(e + f*x)**2 + 1)/(2*f) + 3*B*a*b
**2*d**2*tan(e + f*x)**4/(4*f) - 3*B*a*b**2*d**2*tan(e + f*x)**2/(2*f) + B*
b**3*c**2*x + B*b**3*c**2*tan(e + f*x)**3/(3*f) - B*b**3*c**2*tan(e + f*x)/
f + B*b**3*c*d*log(tan(e + f*x)**2 + 1)/f + B*b**3*c*d*tan(e + f*x)**4/(2*f
) - B*b**3*c*d*tan(e + f*x)**2/f - B*b**3*d**2*x + B*b**3*d**2*tan(e + f*x)
**5/(5*f) - B*b**3*d**2*tan(e + f*x)**3/(3*f) + B*b**3*d**2*tan(e + f*x)/f
- C*a**3*c**2*x + C*a**3*c**2*tan(e + f*x)/f - C*a**3*c*d*log(tan(e + f*x)*
*2 + 1)/f + C*a**3*c*d*tan(e + f*x)**2/f + C*a**3*d**2*x + C*a**3*d**2*tan(
e + f*x)**3/(3*f) - C*a**3*d**2*tan(e + f*x)/f - 3*C*a**2*b*c**2*log(tan(e
+ f*x)**2 + 1)/(2*f) + 3*C*a**2*b*c**2*tan(e + f*x)**2/(2*f) + 6*C*a**2*b*c
*d*x + 2*C*a**2*b*c*d*tan(e + f*x)**3/f - 6*C*a**2*b*c*d*tan(e + f*x)/f + 3
*C*a**2*b*d**2*log(tan(e + f*x)**2 + 1)/(2*f) + 3*C*a**2*b*d**2*tan(e + f*x
)**4/(4*f) - 3*C*a**2*b*d**2*tan(e + f*x)**2/(2*f) + 3*C*a*b**2*c**2*x + C*
a*b**2*c**2*tan(e + f*x)**3/f - 3*C*a*b**2*c**2*tan(e + f*x)/f + 3*C*a*b**2
*c*d*log(tan(e + f*x)**2 + 1)/f + 3*C*a*b**2*c*d*tan(e + f*x)**4/(2*f) - 3*
C*a*b**2*c*d*tan(e + f*x)**2/f - 3*C*a*b**2*d**2*x + 3*C*a*b**2*d**2*tan(e
+ f*x)**5/(5*f) - C*a*b**2*d**2*tan(e + f*x)**3/f + 3*C*a*b**2*d**2*tan(e +
 f*x)/f + C*b**3*c**2*log(tan(e + f*x)**2 + 1)/(2*f) + C*b**3*c**2*tan(e +
f*x)**4/(4*f) - C*b**3*c**2*tan(e + f*x)**2/(2*f) - 2*C*b**3*c*d*x + 2*C*b*
*3*c*d*tan(e + f*x)**5/(5*f) - 2*C*b**3*c*d*tan(e + f*x)**3/(3*f) + 2*C*b**
3*c*d*tan(e + f*x)/f - C*b**3*d**2*log(tan(e + f*x)**2 + 1)/(2*f) + C*b**3*
d**2*tan(e + f*x)**6/(6*f) - C*b**3*d**2*tan(e + f*x)**4/(4*f) + C*b**3*d**
2*tan(e + f*x)**2/(2*f), Ne(f, 0)), (x*(a + b*tan(e))**3*(c + d*tan(e))**2*
(A + B*tan(e) + C*tan(e)**2), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.35 (sec) , antiderivative size = 691, normalized size of antiderivative = 1.05

$$\int (a+b\tan(e+fx))^3(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\frac{10\,Cb^3d^2\tan\left(fx+e\right)^6+12\left(2\,Cb^3cd+\left(3\,Cab^2+Bb^3\right)d^2\right)\tan\left(fx+e\right)^5+15\left(Cb^3c^2+2\left(3\,Cab^2+Bb^3\right)cd\right.}{}$$

```
[In] integrate((a+b*tan(f*x+e))^3*(c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e
)^2),x, algorithm="maxima")
```

```
[Out] 1/60*(10*C*b^3*d^2*tan(f*x + e)^6 + 12*(2*C*b^3*c*d + (3*C*a*b^2 + B*b^3)*d
^2)*tan(f*x + e)^5 + 15*(C*b^3*c^2 + 2*(3*C*a*b^2 + B*b^3)*c*d + (3*C*a^2*b
```

```
 + 3*B*a*b^2 + (A - C)*b^3)*d^2)*tan(f*x + e)^4 + 20*((3*C*a*b^2 + B*b^3)*c
^2 + 2*(3*C*a^2*b + 3*B*a*b^2 + (A - C)*b^3)*c*d + (C*a^3 + 3*B*a^2*b + 3*(
A - C)*a*b^2 - B*b^3)*d^2)*tan(f*x + e)^3 + 30*((3*C*a^2*b + 3*B*a*b^2 + (A
 - C)*b^3)*c^2 + 2*(C*a^3 + 3*B*a^2*b + 3*(A - C)*a*b^2 - B*b^3)*c*d + (B*a
^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*d^2)*tan(f*x + e)^2 + 60*((
(A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b^3)*c^2 - 2*(B*a^3 + 3*(A -
C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*c*d - ((A - C)*a^3 - 3*B*a^2*b - 3*(A -
 C)*a*b^2 + B*b^3)*d^2)*(f*x + e) + 30*((B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^
2 - (A - C)*b^3)*c^2 + 2*((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b^3
)*c*d - (B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*d^2)*log(tan(f*
x + e)^2 + 1) + 60*((C*a^3 + 3*B*a^2*b + 3*(A - C)*a*b^2 - B*b^3)*c^2 + 2*(
B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*c*d + ((A - C)*a^3 - 3*B
*a^2*b - 3*(A - C)*a*b^2 + B*b^3)*d^2)*tan(f*x + e))/f
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 21368 vs. $2(650) = 1300$.

Time = 23.58 (sec) , antiderivative size = 21368, normalized size of antiderivative = 32.33

$$\int (a+b\tan(e+fx))^3(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= \text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))^3*(c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e
)^2),x, algorithm="giac")
```

```
[Out] 1/60*(60*A*a^3*c^2*f*x*tan(f*x)^6*tan(e)^6 - 60*C*a^3*c^2*f*x*tan(f*x)^6*ta
n(e)^6 - 180*B*a^2*b*c^2*f*x*tan(f*x)^6*tan(e)^6 - 180*A*a*b^2*c^2*f*x*tan(
f*x)^6*tan(e)^6 + 180*C*a*b^2*c^2*f*x*tan(f*x)^6*tan(e)^6 + 60*B*b^3*c^2*f*
x*tan(f*x)^6*tan(e)^6 - 120*B*a^3*c*d*f*x*tan(f*x)^6*tan(e)^6 - 360*A*a^2*b
*c*d*f*x*tan(f*x)^6*tan(e)^6 + 360*C*a^2*b*c*d*f*x*tan(f*x)^6*tan(e)^6 + 36
0*B*a*b^2*c*d*f*x*tan(f*x)^6*tan(e)^6 + 120*A*b^3*c*d*f*x*tan(f*x)^6*tan(e)
^6 - 120*C*b^3*c*d*f*x*tan(f*x)^6*tan(e)^6 - 60*A*a^3*d^2*f*x*tan(f*x)^6*ta
n(e)^6 + 60*C*a^3*d^2*f*x*tan(f*x)^6*tan(e)^6 + 180*B*a^2*b*d^2*f*x*tan(f*x
)^6*tan(e)^6 + 180*A*a*b^2*d^2*f*x*tan(f*x)^6*tan(e)^6 - 180*C*a*b^2*d^2*f*
x*tan(f*x)^6*tan(e)^6 - 60*B*b^3*d^2*f*x*tan(f*x)^6*tan(e)^6 - 30*B*a^3*c^2
*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 +
 tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^6*tan(e)^6 - 90*A*a^2*b*c^2*log(4*(ta
n(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^
2 + tan(e)^2 + 1))*tan(f*x)^6*tan(e)^6 + 90*C*a^2*b*c^2*log(4*(tan(f*x)^2*t
an(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)
^2 + 1))*tan(f*x)^6*tan(e)^6 + 90*B*a*b^2*c^2*log(4*(tan(f*x)^2*tan(e)^2 -
2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*t
an(f*x)^6*tan(e)^6 + 30*A*b^3*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*t
an(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^6*ta
```

n(e)^6 - 30*C*b^3*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(
tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^6*tan(e)^6 - 60*
A*a^3*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*t
an(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^6*tan(e)^6 + 60*C*a^3*c*d*lo
g(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + ta
n(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^6*tan(e)^6 + 180*B*a^2*b*c*d*log(4*(tan(
f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2
+ tan(e)^2 + 1))*tan(f*x)^6*tan(e)^6 + 180*A*a*b^2*c*d*log(4*(tan(f*x)^2*ta
n(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^
2 + 1))*tan(f*x)^6*tan(e)^6 - 180*C*a*b^2*c*d*log(4*(tan(f*x)^2*tan(e)^2 -
2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*t
an(f*x)^6*tan(e)^6 - 60*B*b^3*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*t
an(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^6*ta
n(e)^6 + 30*B*a^3*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(
tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^6*tan(e)^6 + 90*
A*a^2*b*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2
*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^6*tan(e)^6 - 90*C*a^2*b*d^
2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2
+ tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^6*tan(e)^6 - 90*B*a*b^2*d^2*log(4*(t
an(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)
^2 + tan(e)^2 + 1))*tan(f*x)^6*tan(e)^6 - 30*A*b^3*d^2*log(4*(tan(f*x)^2*ta
n(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^
2 + 1))*tan(f*x)^6*tan(e)^6 + 30*C*b^3*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*t
an(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(
f*x)^6*tan(e)^6 - 360*A*a^3*c^2*f*x*tan(f*x)^5*tan(e)^5 + 360*C*a^3*c^2*f*x
*tan(f*x)^5*tan(e)^5 + 1080*B*a^2*b*c^2*f*x*tan(f*x)^5*tan(e)^5 + 1080*A*a*
b^2*c^2*f*x*tan(f*x)^5*tan(e)^5 - 1080*C*a*b^2*c^2*f*x*tan(f*x)^5*tan(e)^5
- 360*B*b^3*c^2*f*x*tan(f*x)^5*tan(e)^5 + 720*B*a^3*c*d*f*x*tan(f*x)^5*tan(
e)^5 + 2160*A*a^2*b*c*d*f*x*tan(f*x)^5*tan(e)^5 - 2160*C*a^2*b*c*d*f*x*tan(
f*x)^5*tan(e)^5 - 2160*B*a*b^2*c*d*f*x*tan(f*x)^5*tan(e)^5 - 720*A*b^3*c*d*
f*x*tan(f*x)^5*tan(e)^5 + 720*C*b^3*c*d*f*x*tan(f*x)^5*tan(e)^5 + 360*A*a^3
*d^2*f*x*tan(f*x)^5*tan(e)^5 - 360*C*a^3*d^2*f*x*tan(f*x)^5*tan(e)^5 - 1080
*B*a^2*b*d^2*f*x*tan(f*x)^5*tan(e)^5 - 1080*A*a*b^2*d^2*f*x*tan(f*x)^5*tan(
e)^5 + 1080*C*a*b^2*d^2*f*x*tan(f*x)^5*tan(e)^5 + 360*B*b^3*d^2*f*x*tan(f*x
)^5*tan(e)^5 + 90*C*a^2*b*c^2*tan(f*x)^6*tan(e)^6 + 90*B*a*b^2*c^2*tan(f*x)
^6*tan(e)^6 + 30*A*b^3*c^2*tan(f*x)^6*tan(e)^6 - 45*C*b^3*c^2*tan(f*x)^6*ta
n(e)^6 + 60*C*a^3*c*d*tan(f*x)^6*tan(e)^6 + 180*B*a^2*b*c*d*tan(f*x)^6*tan(
e)^6 + 180*A*a*b^2*c*d*tan(f*x)^6*tan(e)^6 - 270*C*a*b^2*c*d*tan(f*x)^6*tan
(e)^6 - 90*B*b^3*c*d*tan(f*x)^6*tan(e)^6 + 30*B*a^3*d^2*tan(f*x)^6*tan(e)^6
+ 90*A*a^2*b*d^2*tan(f*x)^6*tan(e)^6 - 135*C*a^2*b*d^2*tan(f*x)^6*tan(e)^6
- 135*B*a*b^2*d^2*tan(f*x)^6*tan(e)^6 - 45*A*b^3*d^2*tan(f*x)^6*tan(e)^6 +
55*C*b^3*d^2*tan(f*x)^6*tan(e)^6 + 180*B*a^3*c^2*log(4*(tan(f*x)^2*tan(e)^
2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1
))*tan(f*x)^5*tan(e)^5 + 540*A*a^2*b*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan
(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*

x)^5*tan(e)^5 - 540*C*a^2*b*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan
(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5*tan(
e)^5 - 540*B*a*b^2*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/
(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5 - 18
0*A*b^3*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2
*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5 + 180*C*b^3*c^2
log(4(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 +
 tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5 + 360*A*a^3*c*d*log(4*(tan
(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2
 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5 - 360*C*a^3*c*d*log(4*(tan(f*x)^2*tan
(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2
 + 1))*tan(f*x)^5*tan(e)^5 - 1080*B*a^2*b*c*d*log(4*(tan(f*x)^2*tan(e)^2 -
2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*t
an(f*x)^5*tan(e)^5 - 1080*A*a*b^2*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*
x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^
5*tan(e)^5 + 1080*C*a*b^2*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e
) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)
^5 + 360*B*b^3*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan
(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5 - 180*B*
a^3*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan
(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5 - 540*A*a^2*b*d^2*l
og(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + t
an(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5 + 540*C*a^2*b*d^2*log(4*(tan
(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2
 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5 + 540*B*a*b^2*d^2*log(4*(tan(f*x)^2*t
an(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)
^2 + 1))*tan(f*x)^5*tan(e)^5 + 180*A*b^3*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2
*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*ta
n(f*x)^5*tan(e)^5 - 180*C*b^3*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*t
an(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5*ta
n(e)^5 - 60*C*a^3*c^2*tan(f*x)^6*tan(e)^5 - 180*B*a^2*b*c^2*tan(f*x)^6*tan(
e)^5 - 180*A*a*b^2*c^2*tan(f*x)^6*tan(e)^5 + 180*C*a*b^2*c^2*tan(f*x)^6*tan
(e)^5 + 60*B*b^3*c^2*tan(f*x)^6*tan(e)^5 - 120*B*a^3*c*d*tan(f*x)^6*tan(e)^
5 - 360*A*a^2*b*c*d*tan(f*x)^6*tan(e)^5 + 360*C*a^2*b*c*d*tan(f*x)^6*tan(e)
^5 + 360*B*a*b^2*c*d*tan(f*x)^6*tan(e)^5 + 120*A*b^3*c*d*tan(f*x)^6*tan(e)^
5 - 120*C*b^3*c*d*tan(f*x)^6*tan(e)^5 - 60*A*a^3*d^2*tan(f*x)^6*tan(e)^5 +
60*C*a^3*d^2*tan(f*x)^6*tan(e)^5 + 180*B*a^2*b*d^2*tan(f*x)^6*tan(e)^5 + 18
0*A*a*b^2*d^2*tan(f*x)^6*tan(e)^5 - 180*C*a*b^2*d^2*tan(f*x)^6*tan(e)^5 - 6
0*B*b^3*d^2*tan(f*x)^6*tan(e)^5 - 60*C*a^3*c^2*tan(f*x)^5*tan(e)^6 - 180*B*
a^2*b*c^2*tan(f*x)^5*tan(e)^6 - 180*A*a*b^2*c^2*tan(f*x)^5*tan(e)^6 + 180*C
*a*b^2*c^2*tan(f*x)^5*tan(e)^6 + 60*B*b^3*c^2*tan(f*x)^5*tan(e)^6 - 120*B*a
^3*c*d*tan(f*x)^5*tan(e)^6 - 360*A*a^2*b*c*d*tan(f*x)^5*tan(e)^6 + 360*C*a^
2*b*c*d*tan(f*x)^5*tan(e)^6 + 360*B*a*b^2*c*d*tan(f*x)^5*tan(e)^6 + 120*A*b
^3*c*d*tan(f*x)^5*tan(e)^6 - 120*C*b^3*c*d*tan(f*x)^5*tan(e)^6 - 60*A*a^3*d
^2*tan(f*x)^5*tan(e)^6 + 60*C*a^3*d^2*tan(f*x)^5*tan(e)^6 + 180*B*a^2*b*d^2

*tan(f*x)^5*tan(e)^6 + 180*A*a*b^2*d^2*tan(f*x)^5*tan(e)^6 - 180*C*a*b^2*d^2*tan(f*x)^5*tan(e)^6 - 60*B*b^3*d^2*tan(f*x)^5*tan(e)^6 + 900*A*a^3*c^2*f*x*tan(f*x)^4*tan(e)^4 - 900*C*a^3*c^2*f*x*tan(f*x)^4*tan(e)^4 - 2700*B*a^2*b*c^2*f*x*tan(f*x)^4*tan(e)^4 - 2700*A*a*b^2*c^2*f*x*tan(f*x)^4*tan(e)^4 + 2700*C*a*b^2*c^2*f*x*tan(f*x)^4*tan(e)^4 + 900*B*b^3*c^2*f*x*tan(f*x)^4*tan(e)^4 - 1800*B*a^3*c*d*f*x*tan(f*x)^4*tan(e)^4 - 5400*A*a^2*b*c*d*f*x*tan(f*x)^4*tan(e)^4 + 5400*C*a^2*b*c*d*f*x*tan(f*x)^4*tan(e)^4 + 5400*B*a*b^2*c*d*f*x*tan(f*x)^4*tan(e)^4 + 1800*A*b^3*c*d*f*x*tan(f*x)^4*tan(e)^4 - 1800*C*b^3*c*d*f*x*tan(f*x)^4*tan(e)^4 - 900*A*a^3*d^2*f*x*tan(f*x)^4*tan(e)^4 + 900*C*a^3*d^2*f*x*tan(f*x)^4*tan(e)^4 + 2700*B*a^2*b*d^2*f*x*tan(f*x)^4*tan(e)^4 + 2700*A*a*b^2*d^2*f*x*tan(f*x)^4*tan(e)^4 - 2700*C*a*b^2*d^2*f*x*tan(f*x)^4*tan(e)^4 - 900*B*b^3*d^2*f*x*tan(f*x)^4*tan(e)^4 + 90*C*a^2*b*c^2*tan(f*x)^6*tan(e)^4 + 90*B*a*b^2*c^2*tan(f*x)^6*tan(e)^4 + 30*A*b^3*c^2*tan(f*x)^6*tan(e)^4 - 30*C*b^3*c^2*tan(f*x)^6*tan(e)^4 + 60*C*a^3*c*d*tan(f*x)^6*tan(e)^4 + 180*B*a^2*b*c*d*tan(f*x)^6*tan(e)^4 + 180*A*a*b^2*c*d*tan(f*x)^6*tan(e)^4 - 180*C*a*b^2*c*d*tan(f*x)^6*tan(e)^4 - 60*B*b^3*c*d*tan(f*x)^6*tan(e)^4 + 30*B*a^3*d^2*tan(f*x)^6*tan(e)^4 + 90*A*a^2*b*d^2*tan(f*x)^6*tan(e)^4 - 90*C*a^2*b*d^2*tan(f*x)^6*tan(e)^4 - 90*B*a*b^2*d^2*tan(f*x)^6*tan(e)^4 - 30*A*b^3*d^2*tan(f*x)^6*tan(e)^4 + 30*C*b^3*d^2*tan(f*x)^6*tan(e)^4 - 360*C*a^2*b*c^2*tan(f*x)^5*tan(e)^5 - 360*B*a*b^2*c^2*tan(f*x)^5*tan(e)^5 - 120*A*b^3*c^2*tan(f*x)^5*tan(e)^5 + 210*C*b^3*c^2*tan(f*x)^5*tan(e)^5 - 240*C*a^3*c*d*tan(f*x)^5*tan(e)^5 - 720*B*a^2*b*c*d*tan(f*x)^5*tan(e)^5 - 720*A*a*b^2*c*d*tan(f*x)^5*tan(e)^5 + 1260*C*a*b^2*c*d*tan(f*x)^5*tan(e)^5 + 420*B*b^3*c*d*tan(f*x)^5*tan(e)^5 - 120*B*a^3*d^2*tan(f*x)^5*tan(e)^5 - 360*A*a^2*b*d^2*tan(f*x)^5*tan(e)^5 + 630*C*a^2*b*d^2*tan(f*x)^5*tan(e)^5 + 630*B*a*b^2*d^2*tan(f*x)^5*tan(e)^5 + 210*A*b^3*d^2*tan(f*x)^5*tan(e)^5 - 270*C*b^3*d^2*tan(f*x)^5*tan(e)^5 + 90*C*a^2*b*c^2*tan(f*x)^4*tan(e)^6 + 90*B*a*b^2*c^2*tan(f*x)^4*tan(e)^6 + 30*A*b^3*c^2*tan(f*x)^4*tan(e)^6 - 30*C*b^3*c^2*tan(f*x)^4*tan(e)^6 + 60*C*a^3*c*d*tan(f*x)^4*tan(e)^6 + 180*B*a^2*b*c*d*tan(f*x)^4*tan(e)^6 + 180*A*a*b^2*c*d*tan(f*x)^4*tan(e)^6 - 180*C*a*b^2*c*d*tan(f*x)^4*tan(e)^6 - 60*B*b^3*c*d*tan(f*x)^4*tan(e)^6 + 30*B*a^3*d^2*tan(f*x)^4*tan(e)^6 + 90*A*a^2*b*d^2*tan(f*x)^4*tan(e)^6 - 90*C*a^2*b*d^2*tan(f*x)^4*tan(e)^6 - 90*B*a*b^2*d^2*tan(f*x)^4*tan(e)^6 - 30*A*b^3*d^2*tan(f*x)^4*tan(e)^6 + 30*C*b^3*d^2*tan(f*x)^4*tan(e)^6 - 60*C*a*b^2*c^2*tan(f*x)^6*tan(e)^3 - 20*B*b^3*c^2*tan(f*x)^6*tan(e)^3 - 120*C*a^2*b*c*d*tan(f*x)^6*tan(e)^3 - 120*B*a*b^2*c*d*tan(f*x)^6*tan(e)^3 - 40*A*b^3*c*d*tan(f*x)^6*tan(e)^3 + 40*C*b^3*c*d*tan(f*x)^6*tan(e)^3 - 20*C*a^3*d^2*tan(f*x)^6*tan(e)^3 - 60*B*a^2*b*d^2*tan(f*x)^6*tan(e)^3 - 60*A*a*b^2*d^2*tan(f*x)^6*tan(e)^3 + 60*C*a*b^2*d^2*tan(f*x)^6*tan(e)^3 + 20*B*b^3*d^2*tan(f*x)^6*tan(e)^3 - 450*B*a^3*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 - 1350*A*a^2*b*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 1350*C*a^2*b*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 1350*B*a*b^2*c^2*log(4*(

tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x
)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 450*A*b^3*c^2*log(4*(tan(f*x)^2*
tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e
)^2 + 1))*tan(f*x)^4*tan(e)^4 - 450*C*b^3*c^2*log(4*(tan(f*x)^2*tan(e)^2 -
2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*t
an(f*x)^4*tan(e)^4 - 900*A*a^3*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*
tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*t
an(e)^4 + 900*C*a^3*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)
/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 2
700*B*a^2*b*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*
x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 2700*A*a*
b^2*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan
(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 - 2700*C*a*b^2*c*d*
log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 +
tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 - 900*B*b^3*c*d*log(4*(tan(
f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2
+ tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 450*B*a^3*d^2*log(4*(tan(f*x)^2*tan(
e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2
+ 1))*tan(f*x)^4*tan(e)^4 + 1350*A*a^2*b*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2
*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*ta
n(f*x)^4*tan(e)^4 - 1350*C*a^2*b*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x
)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4
*tan(e)^4 - 1350*B*a*b^2*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e)
 + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^
4 - 450*A*b^3*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(
f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 450*C*b
^3*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(
e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 300*C*a^3*c^2*tan(
f*x)^5*tan(e)^4 + 900*B*a^2*b*c^2*tan(f*x)^5*tan(e)^4 + 900*A*a*b^2*c^2*tan
(f*x)^5*tan(e)^4 - 1080*C*a*b^2*c^2*tan(f*x)^5*tan(e)^4 - 360*B*b^3*c^2*tan
(f*x)^5*tan(e)^4 + 600*B*a^3*c*d*tan(f*x)^5*tan(e)^4 + 1800*A*a^2*b*c*d*tan
(f*x)^5*tan(e)^4 - 2160*C*a^2*b*c*d*tan(f*x)^5*tan(e)^4 - 2160*B*a*b^2*c*d*
tan(f*x)^5*tan(e)^4 - 720*A*b^3*c*d*tan(f*x)^5*tan(e)^4 + 720*C*b^3*c*d*tan
(f*x)^5*tan(e)^4 + 300*A*a^3*d^2*tan(f*x)^5*tan(e)^4 - 360*C*a^3*d^2*tan(f*
x)^5*tan(e)^4 - 1080*B*a^2*b*d^2*tan(f*x)^5*tan(e)^4 - 1080*A*a*b^2*d^2*tan
(f*x)^5*tan(e)^4 + 1080*C*a*b^2*d^2*tan(f*x)^5*tan(e)^4 + 360*B*b^3*d^2*tan
(f*x)^5*tan(e)^4 + 300*C*a^3*c^2*tan(f*x)^4*tan(e)^5 + 900*B*a^2*b*c^2*tan(
f*x)^4*tan(e)^5 + 900*A*a*b^2*c^2*tan(f*x)^4*tan(e)^5 - 1080*C*a*b^2*c^2*ta
n(f*x)^4*tan(e)^5 - 360*B*b^3*c^2*tan(f*x)^4*tan(e)^5 + 600*B*a^3*c*d*tan(f
*x)^4*tan(e)^5 + 1800*A*a^2*b*c*d*tan(f*x)^4*tan(e)^5 - 2160*C*a^2*b*c*d*ta
n(f*x)^4*tan(e)^5 - 2160*B*a*b^2*c*d*tan(f*x)^4*tan(e)^5 - 720*A*b^3*c*d*ta
n(f*x)^4*tan(e)^5 + 720*C*b^3*c*d*tan(f*x)^4*tan(e)^5 + 300*A*a^3*d^2*tan(f
*x)^4*tan(e)^5 - 360*C*a^3*d^2*tan(f*x)^4*tan(e)^5 - 1080*B*a^2*b*d^2*tan(f
*x)^4*tan(e)^5 - 1080*A*a*b^2*d^2*tan(f*x)^4*tan(e)^5 + 1080*C*a*b^2*d^2*ta
n(f*x)^4*tan(e)^5 + 360*B*b^3*d^2*tan(f*x)^4*tan(e)^5 - 60*C*a*b^2*c^2*tan(

f*x)^3*tan(e)^6 - 20*B*b^3*c^2*tan(f*x)^3*tan(e)^6 - 120*C*a^2*b*c*d*tan(f*
x)^3*tan(e)^6 - 120*B*a*b^2*c*d*tan(f*x)^3*tan(e)^6 - 40*A*b^3*c*d*tan(f*x)
^3*tan(e)^6 + 40*C*b^3*c*d*tan(f*x)^3*tan(e)^6 - 20*C*a^3*d^2*tan(f*x)^3*ta
n(e)^6 - 60*B*a^2*b*d^2*tan(f*x)^3*tan(e)^6 - 60*A*a*b^2*d^2*tan(f*x)^3*tan
(e)^6 + 60*C*a*b^2*d^2*tan(f*x)^3*tan(e)^6 + 20*B*b^3*d^2*tan(f*x)^3*tan(e)
^6 + 15*C*b^3*c^2*tan(f*x)^6*tan(e)^2 + 90*C*a*b^2*c*d*tan(f*x)^6*tan(e)^2
+ 30*B*b^3*c*d*tan(f*x)^6*tan(e)^2 + 45*C*a^2*b*d^2*tan(f*x)^6*tan(e)^2 + 4
5*B*a*b^2*d^2*tan(f*x)^6*tan(e)^2 + 15*A*b^3*d^2*tan(f*x)^6*tan(e)^2 - 15*C
*b^3*d^2*tan(f*x)^6*tan(e)^2 - 1200*A*a^3*c^2*f*x*tan(f*x)^3*tan(e)^3 + 120
0*C*a^3*c^2*f*x*tan(f*x)^3*tan(e)^3 + 3600*B*a^2*b*c^2*f*x*tan(f*x)^3*tan(e
)^3 + 3600*A*a*b^2*c^2*f*x*tan(f*x)^3*tan(e)^3 - 3600*C*a*b^2*c^2*f*x*tan(f
*x)^3*tan(e)^3 - 1200*B*b^3*c^2*f*x*tan(f*x)^3*tan(e)^3 + 2400*B*a^3*c*d*f*
x*tan(f*x)^3*tan(e)^3 + 7200*A*a^2*b*c*d*f*x*tan(f*x)^3*tan(e)^3 - 7200*C*a
^2*b*c*d*f*x*tan(f*x)^3*tan(e)^3 - 7200*B*a*b^2*c*d*f*x*tan(f*x)^3*tan(e)^3
 - 2400*A*b^3*c*d*f*x*tan(f*x)^3*tan(e)^3 + 2400*C*b^3*c*d*f*x*tan(f*x)^3*t
an(e)^3 + 1200*A*a^3*d^2*f*x*tan(f*x)^3*tan(e)^3 - 1200*C*a^3*d^2*f*x*tan(f
*x)^3*tan(e)^3 - 3600*B*a^2*b*d^2*f*x*tan(f*x)^3*tan(e)^3 - 3600*A*a*b^2*d^
2*f*x*tan(f*x)^3*tan(e)^3 + 3600*C*a*b^2*d^2*f*x*tan(f*x)^3*tan(e)^3 + 1200
*B*b^3*d^2*f*x*tan(f*x)^3*tan(e)^3 - 360*C*a^2*b*c^2*tan(f*x)^5*tan(e)^3 -
360*B*a*b^2*c^2*tan(f*x)^5*tan(e)^3 - 120*A*b^3*c^2*tan(f*x)^5*tan(e)^3 + 1
80*C*b^3*c^2*tan(f*x)^5*tan(e)^3 - 240*C*a^3*c*d*tan(f*x)^5*tan(e)^3 - 720*
B*a^2*b*c*d*tan(f*x)^5*tan(e)^3 - 720*A*a*b^2*c*d*tan(f*x)^5*tan(e)^3 + 108
0*C*a*b^2*c*d*tan(f*x)^5*tan(e)^3 + 360*B*b^3*c*d*tan(f*x)^5*tan(e)^3 - 120
*B*a^3*d^2*tan(f*x)^5*tan(e)^3 - 360*A*a^2*b*d^2*tan(f*x)^5*tan(e)^3 + 540*
C*a^2*b*d^2*tan(f*x)^5*tan(e)^3 + 540*B*a*b^2*d^2*tan(f*x)^5*tan(e)^3 + 180
*A*b^3*d^2*tan(f*x)^5*tan(e)^3 - 180*C*b^3*d^2*tan(f*x)^5*tan(e)^3 + 630*C*
a^2*b*c^2*tan(f*x)^4*tan(e)^4 + 630*B*a*b^2*c^2*tan(f*x)^4*tan(e)^4 + 210*A
*b^3*c^2*tan(f*x)^4*tan(e)^4 - 345*C*b^3*c^2*tan(f*x)^4*tan(e)^4 + 420*C*a^
3*c*d*tan(f*x)^4*tan(e)^4 + 1260*B*a^2*b*c*d*tan(f*x)^4*tan(e)^4 + 1260*A*a
*b^2*c*d*tan(f*x)^4*tan(e)^4 - 2070*C*a*b^2*c*d*tan(f*x)^4*tan(e)^4 - 690*B
*b^3*c*d*tan(f*x)^4*tan(e)^4 + 210*B*a^3*d^2*tan(f*x)^4*tan(e)^4 + 630*A*a^
2*b*d^2*tan(f*x)^4*tan(e)^4 - 1035*C*a^2*b*d^2*tan(f*x)^4*tan(e)^4 - 1035*B
*a*b^2*d^2*tan(f*x)^4*tan(e)^4 - 345*A*b^3*d^2*tan(f*x)^4*tan(e)^4 + 495*C*
b^3*d^2*tan(f*x)^4*tan(e)^4 - 360*C*a^2*b*c^2*tan(f*x)^3*tan(e)^5 - 360*B*a
*b^2*c^2*tan(f*x)^3*tan(e)^5 - 120*A*b^3*c^2*tan(f*x)^3*tan(e)^5 + 180*C*b^
3*c^2*tan(f*x)^3*tan(e)^5 - 240*C*a^3*c*d*tan(f*x)^3*tan(e)^5 - 720*B*a^2*b
*c*d*tan(f*x)^3*tan(e)^5 - 720*A*a*b^2*c*d*tan(f*x)^3*tan(e)^5 + 1080*C*a*b
^2*c*d*tan(f*x)^3*tan(e)^5 + 360*B*b^3*c*d*tan(f*x)^3*tan(e)^5 - 120*B*a^3*
d^2*tan(f*x)^3*tan(e)^5 - 360*A*a^2*b*d^2*tan(f*x)^3*tan(e)^5 + 540*C*a^2*b
*d^2*tan(f*x)^3*tan(e)^5 + 540*B*a*b^2*d^2*tan(f*x)^3*tan(e)^5 + 180*A*b^3*
d^2*tan(f*x)^3*tan(e)^5 - 180*C*b^3*d^2*tan(f*x)^3*tan(e)^5 + 15*C*b^3*c^2*
tan(f*x)^2*tan(e)^6 + 90*C*a*b^2*c*d*tan(f*x)^2*tan(e)^6 + 30*B*b^3*c*d*tan
(f*x)^2*tan(e)^6 + 45*C*a^2*b*d^2*tan(f*x)^2*tan(e)^6 + 45*B*a*b^2*d^2*tan(
f*x)^2*tan(e)^6 + 15*A*b^3*d^2*tan(f*x)^2*tan(e)^6 - 15*C*b^3*d^2*tan(f*x)^
2*tan(e)^6 - 24*C*b^3*c*d*tan(f*x)^6*tan(e) - 36*C*a*b^2*d^2*tan(f*x)^6*tan

(e) - 12*B*b^3*d^2*tan(f*x)^6*tan(e) + 180*C*a*b^2*c^2*tan(f*x)^5*tan(e)^2
+ 60*B*b^3*c^2*tan(f*x)^5*tan(e)^2 + 360*C*a^2*b*c*d*tan(f*x)^5*tan(e)^2 +
360*B*a*b^2*c*d*tan(f*x)^5*tan(e)^2 + 120*A*b^3*c*d*tan(f*x)^5*tan(e)^2 - 2
40*C*b^3*c*d*tan(f*x)^5*tan(e)^2 + 60*C*a^3*d^2*tan(f*x)^5*tan(e)^2 + 180*B
*a^2*b*d^2*tan(f*x)^5*tan(e)^2 + 180*A*a*b^2*d^2*tan(f*x)^5*tan(e)^2 - 360*
C*a*b^2*d^2*tan(f*x)^5*tan(e)^2 - 120*B*b^3*d^2*tan(f*x)^5*tan(e)^2 + 600*B
*a^3*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*ta
n(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 1800*A*a^2*b*c^2
log(4(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 +
tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 1800*C*a^2*b*c^2*log(4*(
tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x
)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 1800*B*a*b^2*c^2*log(4*(tan(f*x)
^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + ta
n(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 600*A*b^3*c^2*log(4*(tan(f*x)^2*tan(e)^2
- 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)
)*tan(f*x)^3*tan(e)^3 + 600*C*b^3*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*
x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^
3*tan(e)^3 + 1200*A*a^3*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e)
+ 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3
- 1200*C*a^3*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(
f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 3600*B*
a^2*b*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*t
an(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 3600*A*a*b^2*c*
d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2
+ tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 3600*C*a*b^2*c*d*log(4*
(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*
x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 1200*B*b^3*c*d*log(4*(tan(f*x)^
2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan
(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 600*B*a^3*d^2*log(4*(tan(f*x)^2*tan(e)^2
- 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))
*tan(f*x)^3*tan(e)^3 - 1800*A*a^2*b*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(
f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x
)^3*tan(e)^3 + 1800*C*a^2*b*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan
(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(
e)^3 + 1800*B*a*b^2*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)
/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 6
00*A*b^3*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^
2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 600*C*b^3*d^
2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2
+ tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 600*C*a^3*c^2*tan(f*x)^
4*tan(e)^3 - 1800*B*a^2*b*c^2*tan(f*x)^4*tan(e)^3 - 1800*A*a*b^2*c^2*tan(f*
x)^4*tan(e)^3 + 2340*C*a*b^2*c^2*tan(f*x)^4*tan(e)^3 + 780*B*b^3*c^2*tan(f*
x)^4*tan(e)^3 - 1200*B*a^3*c*d*tan(f*x)^4*tan(e)^3 - 3600*A*a^2*b*c*d*tan(f
*x)^4*tan(e)^3 + 4680*C*a^2*b*c*d*tan(f*x)^4*tan(e)^3 + 4680*B*a*b^2*c*d*ta
n(f*x)^4*tan(e)^3 + 1560*A*b^3*c*d*tan(f*x)^4*tan(e)^3 - 1800*C*b^3*c*d*tan

(f*x)^4*tan(e)^3 - 600*A*a^3*d^2*tan(f*x)^4*tan(e)^3 + 780*C*a^3*d^2*tan(f*x)^4*tan(e)^3 + 2340*B*a^2*b*d^2*tan(f*x)^4*tan(e)^3 + 2340*A*a*b^2*d^2*tan(f*x)^4*tan(e)^3 - 2700*C*a*b^2*d^2*tan(f*x)^4*tan(e)^3 - 900*B*b^3*d^2*tan(f*x)^4*tan(e)^3 - 600*C*a^3*c^2*tan(f*x)^3*tan(e)^4 - 1800*B*a^2*b*c^2*tan(f*x)^3*tan(e)^4 - 1800*A*a*b^2*c^2*tan(f*x)^3*tan(e)^4 + 2340*C*a*b^2*c^2*tan(f*x)^3*tan(e)^4 + 780*B*b^3*c^2*tan(f*x)^3*tan(e)^4 - 1200*B*a^3*c*d*tan(f*x)^3*tan(e)^4 - 3600*A*a^2*b*c*d*tan(f*x)^3*tan(e)^4 + 4680*C*a^2*b*c*d*tan(f*x)^3*tan(e)^4 + 4680*B*a*b^2*c*d*tan(f*x)^3*tan(e)^4 + 1560*A*b^3*c*d*tan(f*x)^3*tan(e)^4 - 1800*C*b^3*c*d*tan(f*x)^3*tan(e)^4 - 600*A*a^3*d^2*tan(f*x)^3*tan(e)^4 + 780*C*a^3*d^2*tan(f*x)^3*tan(e)^4 + 2340*B*a^2*b*d^2*tan(f*x)^3*tan(e)^4 + 2340*A*a*b^2*d^2*tan(f*x)^3*tan(e)^4 - 2700*C*a*b^2*d^2*tan(f*x)^3*tan(e)^4 - 900*B*b^3*d^2*tan(f*x)^3*tan(e)^4 + 180*C*a*b^2*c^2*tan(f*x)^2*tan(e)^5 + 60*B*b^3*c^2*tan(f*x)^2*tan(e)^5 + 360*C*a^2*b*c*d*tan(f*x)^2*tan(e)^5 + 360*B*a*b^2*c*d*tan(f*x)^2*tan(e)^5 + 120*A*b^3*c*d*tan(f*x)^2*tan(e)^5 - 240*C*b^3*c*d*tan(f*x)^2*tan(e)^5 + 60*C*a^3*d^2*tan(f*x)^2*tan(e)^5 + 180*B*a^2*b*d^2*tan(f*x)^2*tan(e)^5 + 180*A*a*b^2*d^2*tan(f*x)^2*tan(e)^5 - 360*C*a*b^2*d^2*tan(f*x)^2*tan(e)^5 - 120*B*b^3*d^2*tan(f*x)^2*tan(e)^5 - 24*C*b^3*c*d*tan(f*x)*tan(e)^6 - 36*C*a*b^2*d^2*tan(f*x)*tan(e)^6 - 12*B*b^3*d^2*tan(f*x)*tan(e)^6 + 10*C*b^3*d^2*tan(f*x)^6 - 30*C*b^3*c^2*tan(f*x)^5*tan(e) - 180*C*a*b^2*c*d*tan(f*x)^5*tan(e) - 60*B*b^3*c*d*tan(f*x)^5*tan(e) - 90*C*a^2*b*d^2*tan(f*x)^5*tan(e) - 90*B*a*b^2*d^2*tan(f*x)^5*tan(e) - 30*A*b^3*d^2*tan(f*x)^5*tan(e) + 90*C*b^3*d^2*tan(f*x)^5*tan(e) + 900*A*a^3*c^2*f*x*tan(f*x)^2*tan(e)^2 - 900*C*a^3*c^2*f*x*tan(f*x)^2*tan(e)^2 - 2700*B*a^2*b*c^2*f*x*tan(f*x)^2*tan(e)^2 - 2700*A*a*b^2*c^2*f*x*tan(f*x)^2*tan(e)^2 + 2700*C*a*b^2*c^2*f*x*tan(f*x)^2*tan(e)^2 + 900*B*b^3*c^2*f*x*tan(f*x)^2*tan(e)^2 - 1800*B*a^3*c*d*f*x*tan(f*x)^2*tan(e)^2 - 5400*A*a^2*b*c*d*f*x*tan(f*x)^2*tan(e)^2 + 5400*C*a^2*b*c*d*f*x*tan(f*x)^2*tan(e)^2 + 5400*B*a*b^2*c*d*f*x*tan(f*x)^2*tan(e)^2 + 1800*A*b^3*c*d*f*x*tan(f*x)^2*tan(e)^2 - 1800*C*b^3*c*d*f*x*tan(f*x)^2*tan(e)^2 - 900*A*a^3*d^2*f*x*tan(f*x)^2*tan(e)^2 + 900*C*a^3*d^2*f*x*tan(f*x)^2*tan(e)^2 + 2700*B*a^2*b*d^2*f*x*tan(f*x)^2*tan(e)^2 + 2700*A*a*b^2*d^2*f*x*tan(f*x)^2*tan(e)^2 - 2700*C*a*b^2*d^2*f*x*tan(f*x)^2*tan(e)^2 - 900*B*b^3*d^2*f*x*tan(f*x)^2*tan(e)^2 + 540*C*a^2*b*c^2*tan(f*x)^4*tan(e)^2 + 540*B*a*b^2*c^2*tan(f*x)^4*tan(e)^2 + 180*A*b^3*c^2*tan(f*x)^4*tan(e)^2 - 300*C*b^3*c^2*tan(f*x)^4*tan(e)^2 + 360*C*a^3*c*d*tan(f*x)^4*tan(e)^2 + 1080*B*a^2*b*c*d*tan(f*x)^4*tan(e)^2 + 1080*A*a*b^2*c*d*tan(f*x)^4*tan(e)^2 - 1800*C*a*b^2*c*d*tan(f*x)^4*tan(e)^2 - 600*B*b^3*c*d*tan(f*x)^4*tan(e)^2 + 180*B*a^3*d^2*tan(f*x)^4*tan(e)^2 + 540*A*a^2*b*d^2*tan(f*x)^4*tan(e)^2 - 900*C*a^2*b*d^2*tan(f*x)^4*tan(e)^2 - 900*B*a*b^2*d^2*tan(f*x)^4*tan(e)^2 - 300*A*b^3*d^2*tan(f*x)^4*tan(e)^2 + 450*C*b^3*d^2*tan(f*x)^4*tan(e)^2 - 720*C*a^2*b*c^2*tan(f*x)^3*tan(e)^3 - 720*B*a*b^2*c^2*tan(f*x)^3*tan(e)^3 - 240*A*b^3*c^2*tan(f*x)^3*tan(e)^3 + 360*C*b^3*c^2*tan(f*x)^3*tan(e)^3 - 480*C*a^3*c*d*tan(f*x)^3*tan(e)^3 - 1440*B*a^2*b*c*d*tan(f*x)^3*tan(e)^3 - 1440*A*a*b^2*c*d*tan(f*x)^3*tan(e)^3 + 2160*C*a*b^2*c*d*tan(f*x)^3*tan(e)^3 + 720*B*b^3*c*d*tan(f*x)^3*tan(e)^3 - 240*B*a^3*d^2*tan(f*x)^3*tan(e)^3 - 720*A*a^2*b*d^2*tan(f*x)^3*tan(e)^3

+ 1080*C*a^2*b*d^2*tan(f*x)^3*tan(e)^3 + 1080*B*a*b^2*d^2*tan(f*x)^3*tan(e)
^3 + 360*A*b^3*d^2*tan(f*x)^3*tan(e)^3 - 360*C*b^3*d^2*tan(f*x)^3*tan(e)^3
+ 540*C*a^2*b*c^2*tan(f*x)^2*tan(e)^4 + 540*B*a*b^2*c^2*tan(f*x)^2*tan(e)^4
+ 180*A*b^3*c^2*tan(f*x)^2*tan(e)^4 - 300*C*b^3*c^2*tan(f*x)^2*tan(e)^4 +
360*C*a^3*c*d*tan(f*x)^2*tan(e)^4 + 1080*B*a^2*b*c*d*tan(f*x)^2*tan(e)^4 +
1080*A*a*b^2*c*d*tan(f*x)^2*tan(e)^4 - 1800*C*a*b^2*c*d*tan(f*x)^2*tan(e)^4
- 600*B*b^3*c*d*tan(f*x)^2*tan(e)^4 + 180*B*a^3*d^2*tan(f*x)^2*tan(e)^4 +
540*A*a^2*b*d^2*tan(f*x)^2*tan(e)^4 - 900*C*a^2*b*d^2*tan(f*x)^2*tan(e)^4 -
900*B*a*b^2*d^2*tan(f*x)^2*tan(e)^4 - 300*A*b^3*d^2*tan(f*x)^2*tan(e)^4 +
450*C*b^3*d^2*tan(f*x)^2*tan(e)^4 - 30*C*b^3*c^2*tan(f*x)*tan(e)^5 - 180*C*
a*b^2*c*d*tan(f*x)*tan(e)^5 - 60*B*b^3*c*d*tan(f*x)*tan(e)^5 - 90*C*a^2*b*d
^2*tan(f*x)*tan(e)^5 - 90*B*a*b^2*d^2*tan(f*x)*tan(e)^5 - 30*A*b^3*d^2*tan(
f*x)*tan(e)^5 + 90*C*b^3*d^2*tan(f*x)*tan(e)^5 + 10*C*b^3*d^2*tan(e)^6 + 24
*C*b^3*c*d*tan(f*x)^5 + 36*C*a*b^2*d^2*tan(f*x)^5 + 12*B*b^3*d^2*tan(f*x)^5
- 180*C*a*b^2*c^2*tan(f*x)^4*tan(e) - 60*B*b^3*c^2*tan(f*x)^4*tan(e) - 360
*C*a^2*b*c*d*tan(f*x)^4*tan(e) - 360*B*a*b^2*c*d*tan(f*x)^4*tan(e) - 120*A*
b^3*c*d*tan(f*x)^4*tan(e) + 240*C*b^3*c*d*tan(f*x)^4*tan(e) - 60*C*a^3*d^2*
tan(f*x)^4*tan(e) - 180*B*a^2*b*d^2*tan(f*x)^4*tan(e) - 180*A*a*b^2*d^2*tan
(f*x)^4*tan(e) + 360*C*a*b^2*d^2*tan(f*x)^4*tan(e) + 120*B*b^3*d^2*tan(f*x)
^4*tan(e) - 450*B*a^3*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) +
1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 -
1350*A*a^2*b*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(
f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 1350*C*
a^2*b*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*t
an(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 1350*B*a*b^2*c^
2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2
+ tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 450*A*b^3*c^2*log(4*(ta
n(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^
2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 450*C*b^3*c^2*log(4*(tan(f*x)^2*ta
n(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^
2 + 1))*tan(f*x)^2*tan(e)^2 - 900*A*a^3*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*
tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan
(f*x)^2*tan(e)^2 + 900*C*a^3*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*ta
n(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan
(e)^2 + 2700*B*a^2*b*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1
)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 +
2700*A*a*b^2*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f
*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 2700*C*a
*b^2*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*ta
n(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 900*B*b^3*c*d*lo
g(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + ta
n(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 450*B*a^3*d^2*log(4*(tan(f*
x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 +
tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 1350*A*a^2*b*d^2*log(4*(tan(f*x)^2*tan
(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2

+ 1))*tan(f*x)^2*tan(e)^2 - 1350*C*a^2*b*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 1350*B*a*b^2*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 450*A*b^3*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 450*C*b^3*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 600*C*a^3*c^2*tan(f*x)^3*tan(e)^2 + 1800*B*a^2*b*c^2*tan(f*x)^3*tan(e)^2 + 1800*A*a*b^2*c^2*tan(f*x)^3*tan(e)^2 - 2340*C*a*b^2*c^2*tan(f*x)^3*tan(e)^2 - 780*B*b^3*c^2*tan(f*x)^3*tan(e)^2 + 1200*B*a^3*c*d*tan(f*x)^3*tan(e)^2 + 3600*A*a^2*b*c*d*tan(f*x)^3*tan(e)^2 - 4680*C*a^2*b*c*d*tan(f*x)^3*tan(e)^2 - 4680*B*a*b^2*c*d*tan(f*x)^3*tan(e)^2 - 1560*A*b^3*c*d*tan(f*x)^3*tan(e)^2 + 1800*C*b^3*c*d*tan(f*x)^3*tan(e)^2 + 600*A*a^3*d^2*tan(f*x)^3*tan(e)^2 - 780*C*a^3*d^2*tan(f*x)^3*tan(e)^2 - 2340*B*a^2*b*d^2*tan(f*x)^3*tan(e)^2 - 2340*A*a*b^2*d^2*tan(f*x)^3*tan(e)^2 + 2700*C*a*b^2*d^2*tan(f*x)^3*tan(e)^2 + 900*B*b^3*d^2*tan(f*x)^3*tan(e)^2 + 600*C*a^3*c^2*tan(f*x)^2*tan(e)^3 + 1800*B*a^2*b*c^2*tan(f*x)^2*tan(e)^3 + 1800*A*a*b^2*c^2*tan(f*x)^2*tan(e)^3 - 2340*C*a*b^2*c^2*tan(f*x)^2*tan(e)^3 - 780*B*b^3*c^2*tan(f*x)^2*tan(e)^3 + 1200*B*a^3*c*d*tan(f*x)^2*tan(e)^3 + 3600*A*a^2*b*c*d*tan(f*x)^2*tan(e)^3 - 4680*C*a^2*b*c*d*tan(f*x)^2*tan(e)^3 - 4680*B*a*b^2*c*d*tan(f*x)^2*tan(e)^3 - 1560*A*b^3*c*d*tan(f*x)^2*tan(e)^3 + 1800*C*b^3*c*d*tan(f*x)^2*tan(e)^3 + 600*A*a^3*d^2*tan(f*x)^2*tan(e)^3 - 780*C*a^3*d^2*tan(f*x)^2*tan(e)^3 - 2340*B*a^2*b*d^2*tan(f*x)^2*tan(e)^3 - 2340*A*a*b^2*d^2*tan(f*x)^2*tan(e)^3 + 2700*C*a*b^2*d^2*tan(f*x)^2*tan(e)^3 + 900*B*b^3*d^2*tan(f*x)^2*tan(e)^3 - 180*C*a*b^2*c^2*tan(f*x)*tan(e)^4 - 60*B*b^3*c^2*tan(f*x)*tan(e)^4 - 360*C*a^2*b*c*d*tan(f*x)*tan(e)^4 - 360*B*a*b^2*c*d*tan(f*x)*tan(e)^4 - 120*A*b^3*c*d*tan(f*x)*tan(e)^4 + 240*C*b^3*c*d*tan(f*x)*tan(e)^4 - 60*C*a^3*d^2*tan(f*x)*tan(e)^4 - 180*B*a^2*b*d^2*tan(f*x)*tan(e)^4 - 180*A*a*b^2*d^2*tan(f*x)*tan(e)^4 + 360*C*a*b^2*d^2*tan(f*x)*tan(e)^4 + 120*B*b^3*d^2*tan(f*x)*tan(e)^4 + 24*C*b^3*c*d*tan(e)^5 + 36*C*a*b^2*d^2*tan(e)^5 + 12*B*b^3*d^2*tan(e)^5 + 15*C*b^3*c^2*tan(f*x)^4 + 90*C*a*b^2*c*d*tan(f*x)^4 + 30*B*b^3*c*d*tan(f*x)^4 + 45*C*a^2*b*d^2*tan(f*x)^4 + 45*B*a*b^2*d^2*tan(f*x)^4 + 15*A*b^3*d^2*tan(f*x)^4 - 15*C*b^3*d^2*tan(f*x)^4 - 360*A*a^3*c^2*f*x*tan(f*x)*tan(e) + 360*C*a^3*c^2*f*x*tan(f*x)*tan(e) + 1080*B*a^2*b*c^2*f*x*tan(f*x)*tan(e) + 1080*A*a*b^2*c^2*f*x*tan(f*x)*tan(e) - 1080*C*a*b^2*c^2*f*x*tan(f*x)*tan(e) - 360*B*b^3*c^2*f*x*tan(f*x)*tan(e) + 720*B*a^3*c*d*f*x*tan(f*x)*tan(e) + 2160*A*a^2*b*c*d*f*x*tan(f*x)*tan(e) - 2160*C*a^2*b*c*d*f*x*tan(f*x)*tan(e) - 2160*B*a*b^2*c*d*f*x*tan(f*x)*tan(e) - 720*A*b^3*c*d*f*x*tan(f*x)*tan(e) + 720*C*b^3*c*d*f*x*tan(f*x)*tan(e) + 360*A*a^3*d^2*f*x*tan(f*x)*tan(e) - 360*C*a^3*d^2*f*x*tan(f*x)*tan(e) - 1080*B*a^2*b*d^2*f*x*tan(f*x)*tan(e) - 1080*A*a*b^2*d^2*f*x*tan(f*x)*tan(e) + 1080*C*a*b^2*d^2*f*x*tan(f*x)*tan(e) + 360*B*b^3*d^2*f*x*tan(f*x)*tan(e) - 360*C*a^2*b*c^2*tan(f*x)^3*tan(e) - 360*B*a*b^2*c^2*tan(f*x)^3*tan(e) - 120*A*b^3*c^2*tan(f*x)^3*tan(e) + 180*C*b^3*c^2*tan(f*x)^3*tan(e) - 240*C*a^3*c*d*tan(f*x)^3*tan(e) - 720*B

```
*a^2*b*c*d*tan(f*x)^3*tan(e) - 720*A*a*b^2*c*d*tan(f*x)^3*tan(e) + 1080*C*a
*b^2*c*d*tan(f*x)^3*tan(e) + 360*B*b^3*c*d*tan(f*x)^3*tan(e) - 120*B*a^3*d^
2*tan(f*x)^3*tan(e) - 360*A*a^2*b*d^2*tan(f*x)^3*tan(e) + 540*C*a^2*b*d^2*t
an(f*x)^3*tan(e) + 540*B*a*b^2*d^2*tan(f*x)^3*tan(e) + 180*A*b^3*d^2*tan(f*
x)^3*tan(e) - 180*C*b^3*d^2*tan(f*x)^3*tan(e) + 630*C*a^2*b*c^2*tan(f*x)^2*
tan(e)^2 + 630*B*a*b^2*c^2*tan(f*x)^2*tan(e)^2 + 210*A*b^3*c^2*tan(f*x)^2*t
an(e)^2 - 345*C*b^3*c^2*tan(f*x)^2*tan(e)^2 + 420*C*a^3*c*d*tan(f*x)^2*tan(
e)^2 + 1260*B*a^2*b*c*d*tan(f*x)^2*tan(e)^2 + 1260*A*a*b^2*c*d*tan(f*x)^2*t
an(e)^2 - 2070*C*a*b^2*c*d*tan(f*x)^2*tan(e)^2 - 690*B*b^3*c*d*tan(f*x)^2*t
an(e)^2 + 210*B*a^3*d^2*tan(f*x)^2*tan(e)^2 + 630*A*a^2*b*d^2*tan(f*x)^2*ta
n(e)^2 - 1035*C*a^2*b*d^2*tan(f*x)^2*tan(e)^2 - 1035*B*a*b^2*d^2*tan(f*x)^2
*tan(e)^2 - 345*A*b^3*d^2*tan(f*x)^2*tan(e)^2 + 495*C*b^3*d^2*tan(f*x)^2*ta
n(e)^2 - 360*C*a^2*b*c^2*tan(f*x)*tan(e)^3 - 360*B*a*b^2*c^2*tan(f*x)*tan(e
)^3 - 120*A*b^3*c^2*tan(f*x)*tan(e)^3 + 180*C*b^3*c^2*tan(f*x)*tan(e)^3 - 2
40*C*a^3*c*d*tan(f*x)*tan(e)^3 - 720*B*a^2*b*c*d*tan(f*x)*tan(e)^3 - 720*A*
a*b^2*c*d*tan(f*x)*tan(e)^3 + 1080*C*a*b^2*c*d*tan(f*x)*tan(e)^3 + 360*B*b^
3*c*d*tan(f*x)*tan(e)^3 - 120*B*a^3*d^2*tan(f*x)*tan(e)^3 - 360*A*a^2*b*d^2
*tan(f*x)*tan(e)^3 + 540*C*a^2*b*d^2*tan(f*x)*tan(e)^3 + 540*B*a*b^2*d^2*ta
n(f*x)*tan(e)^3 + 180*A*b^3*d^2*tan(f*x)*tan(e)^3 - 180*C*b^3*d^2*tan(f*x)*
tan(e)^3 + 15*C*b^3*c^2*tan(e)^4 + 90*C*a*b^2*c*d*tan(e)^4 + 30*B*b^3*c*d*t
an(e)^4 + 45*C*a^2*b*d^2*tan(e)^4 + 45*B*a*b^2*d^2*tan(e)^4 + 15*A*b^3*d^2*
tan(e)^4 - 15*C*b^3*d^2*tan(e)^4 + 60*C*a*b^2*c^2*tan(f*x)^3 + 20*B*b^3*c^2
*tan(f*x)^3 + 120*C*a^2*b*c*d*tan(f*x)^3 + 120*B*a*b^2*c*d*tan(f*x)^3 + 40*
A*b^3*c*d*tan(f*x)^3 - 40*C*b^3*c*d*tan(f*x)^3 + 20*C*a^3*d^2*tan(f*x)^3 +
60*B*a^2*b*d^2*tan(f*x)^3 + 60*A*a*b^2*d^2*tan(f*x)^3 - 60*C*a*b^2*d^2*tan(
f*x)^3 - 20*B*b^3*d^2*tan(f*x)^3 + 180*B*a^3*c^2*log(4*(tan(f*x)^2*tan(e)^2
 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)
)*tan(f*x)*tan(e) + 540*A*a^2*b*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)
*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*ta
n(e) - 540*C*a^2*b*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/
(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) - 540*B*
a*b^2*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*t
an(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) - 180*A*b^3*c^2*log(4
*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f
*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 180*C*b^3*c^2*log(4*(tan(f*x)^2*ta
n(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^
2 + 1))*tan(f*x)*tan(e) + 360*A*a^3*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(
f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x
)*tan(e) - 360*C*a^3*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1
)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) - 1080
*B*a^2*b*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^
2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) - 1080*A*a*b^2*c*d
*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 +
 tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 1080*C*a*b^2*c*d*log(4*(tan(
f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2
```

+ tan(e)^2 + 1))*tan(f*x)*tan(e) + 360*B*b^3*c*d*log(4*(tan(f*x)^2*tan(e)^2
 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)
)*tan(f*x)*tan(e) - 180*B*a^3*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*t
an(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(
e) - 540*A*a^2*b*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(t
an(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 540*C*a^
2*b*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan
(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 540*B*a*b^2*d^2*log(4
*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f
*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 180*A*b^3*d^2*log(4*(tan(f*x)^2*ta
n(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^
2 + 1))*tan(f*x)*tan(e) - 180*C*b^3*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(
f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x
)*tan(e) - 300*C*a^3*c^2*tan(f*x)^2*tan(e) - 900*B*a^2*b*c^2*tan(f*x)^2*tan
(e) - 900*A*a*b^2*c^2*tan(f*x)^2*tan(e) + 1080*C*a*b^2*c^2*tan(f*x)^2*tan(e
) + 360*B*b^3*c^2*tan(f*x)^2*tan(e) - 600*B*a^3*c*d*tan(f*x)^2*tan(e) - 180
0*A*a^2*b*c*d*tan(f*x)^2*tan(e) + 2160*C*a^2*b*c*d*tan(f*x)^2*tan(e) + 2160
*B*a*b^2*c*d*tan(f*x)^2*tan(e) + 720*A*b^3*c*d*tan(f*x)^2*tan(e) - 720*C*b^
3*c*d*tan(f*x)^2*tan(e) - 300*A*a^3*d^2*tan(f*x)^2*tan(e) + 360*C*a^3*d^2*t
an(f*x)^2*tan(e) + 1080*B*a^2*b*d^2*tan(f*x)^2*tan(e) + 1080*A*a*b^2*d^2*ta
n(f*x)^2*tan(e) - 1080*C*a*b^2*d^2*tan(f*x)^2*tan(e) - 360*B*b^3*d^2*tan(f*
x)^2*tan(e) - 300*C*a^3*c^2*tan(f*x)*tan(e)^2 - 900*B*a^2*b*c^2*tan(f*x)*ta
n(e)^2 - 900*A*a*b^2*c^2*tan(f*x)*tan(e)^2 + 1080*C*a*b^2*c^2*tan(f*x)*tan(
e)^2 + 360*B*b^3*c^2*tan(f*x)*tan(e)^2 - 600*B*a^3*c*d*tan(f*x)*tan(e)^2 -
1800*A*a^2*b*c*d*tan(f*x)*tan(e)^2 + 2160*C*a^2*b*c*d*tan(f*x)*tan(e)^2 + 2
160*B*a*b^2*c*d*tan(f*x)*tan(e)^2 + 720*A*b^3*c*d*tan(f*x)*tan(e)^2 - 720*C
*b^3*c*d*tan(f*x)*tan(e)^2 - 300*A*a^3*d^2*tan(f*x)*tan(e)^2 + 360*C*a^3*d^
2*tan(f*x)*tan(e)^2 + 1080*B*a^2*b*d^2*tan(f*x)*tan(e)^2 + 1080*A*a*b^2*d^2
*tan(f*x)*tan(e)^2 - 1080*C*a*b^2*d^2*tan(f*x)*tan(e)^2 - 360*B*b^3*d^2*tan
(f*x)*tan(e)^2 + 60*C*a*b^2*c^2*tan(e)^3 + 20*B*b^3*c^2*tan(e)^3 + 120*C*a^
2*b*c*d*tan(e)^3 + 120*B*a*b^2*c*d*tan(e)^3 + 40*A*b^3*c*d*tan(e)^3 - 40*C*
b^3*c*d*tan(e)^3 + 20*C*a^3*d^2*tan(e)^3 + 60*B*a^2*b*d^2*tan(e)^3 + 60*A*a
*b^2*d^2*tan(e)^3 - 60*C*a*b^2*d^2*tan(e)^3 - 20*B*b^3*d^2*tan(e)^3 + 60*A*
a^3*c^2*f*x - 60*C*a^3*c^2*f*x - 180*B*a^2*b*c^2*f*x - 180*A*a*b^2*c^2*f*x
+ 180*C*a*b^2*c^2*f*x + 60*B*b^3*c^2*f*x - 120*B*a^3*c*d*f*x - 360*A*a^2*b*
c*d*f*x + 360*C*a^2*b*c*d*f*x + 360*B*a*b^2*c*d*f*x + 120*A*b^3*c*d*f*x - 1
20*C*b^3*c*d*f*x - 60*A*a^3*d^2*f*x + 60*C*a^3*d^2*f*x + 180*B*a^2*b*d^2*f*
x + 180*A*a*b^2*d^2*f*x - 180*C*a*b^2*d^2*f*x - 60*B*b^3*d^2*f*x + 90*C*a^2
*b*c^2*tan(f*x)^2 + 90*B*a*b^2*c^2*tan(f*x)^2 + 30*A*b^3*c^2*tan(f*x)^2 - 3
0*C*b^3*c^2*tan(f*x)^2 + 60*C*a^3*c*d*tan(f*x)^2 + 180*B*a^2*b*c*d*tan(f*x)
^2 + 180*A*a*b^2*c*d*tan(f*x)^2 - 180*C*a*b^2*c*d*tan(f*x)^2 - 60*B*b^3*c*d
*tan(f*x)^2 + 30*B*a^3*d^2*tan(f*x)^2 + 90*A*a^2*b*d^2*tan(f*x)^2 - 90*C*a^
2*b*d^2*tan(f*x)^2 - 90*B*a*b^2*d^2*tan(f*x)^2 - 30*A*b^3*d^2*tan(f*x)^2 +
30*C*b^3*d^2*tan(f*x)^2 - 360*C*a^2*b*c^2*tan(f*x)*tan(e) - 360*B*a*b^2*c^2
*tan(f*x)*tan(e) - 120*A*b^3*c^2*tan(f*x)*tan(e) + 210*C*b^3*c^2*tan(f*x)*t

an(e) - 240*C*a^3*c*d*tan(f*x)*tan(e) - 720*B*a^2*b*c*d*tan(f*x)*tan(e) - 7
20*A*a*b^2*c*d*tan(f*x)*tan(e) + 1260*C*a*b^2*c*d*tan(f*x)*tan(e) + 420*B*b
^3*c*d*tan(f*x)*tan(e) - 120*B*a^3*d^2*tan(f*x)*tan(e) - 360*A*a^2*b*d^2*ta
n(f*x)*tan(e) + 630*C*a^2*b*d^2*tan(f*x)*tan(e) + 630*B*a*b^2*d^2*tan(f*x)*
tan(e) + 210*A*b^3*d^2*tan(f*x)*tan(e) - 270*C*b^3*d^2*tan(f*x)*tan(e) + 90
*C*a^2*b*c^2*tan(e)^2 + 90*B*a*b^2*c^2*tan(e)^2 + 30*A*b^3*c^2*tan(e)^2 - 3
0*C*b^3*c^2*tan(e)^2 + 60*C*a^3*c*d*tan(e)^2 + 180*B*a^2*b*c*d*tan(e)^2 + 1
80*A*a*b^2*c*d*tan(e)^2 - 180*C*a*b^2*c*d*tan(e)^2 - 60*B*b^3*c*d*tan(e)^2
+ 30*B*a^3*d^2*tan(e)^2 + 90*A*a^2*b*d^2*tan(e)^2 - 90*C*a^2*b*d^2*tan(e)^2
 - 90*B*a*b^2*d^2*tan(e)^2 - 30*A*b^3*d^2*tan(e)^2 + 30*C*b^3*d^2*tan(e)^2
- 30*B*a^3*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x
)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 90*A*a^2*b*c^2*log(4*(tan(f*x)
^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + ta
n(e)^2 + 1)) + 90*C*a^2*b*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e
) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) + 90*B*a*b^2*c^2*
log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 +
tan(f*x)^2 + tan(e)^2 + 1)) + 30*A*b^3*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*t
an(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 30
*C*b^3*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*
tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 60*A*a^3*c*d*log(4*(tan(f*x)^2*tan
(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2
 + 1)) + 60*C*a^3*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(
tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) + 180*B*a^2*b*c*d*log(4*(
tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x
)^2 + tan(e)^2 + 1)) + 180*A*a*b^2*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f
*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 180*C*
a*b^2*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*t
an(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 60*B*b^3*c*d*log(4*(tan(f*x)^2*tan(
e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2
+ 1)) + 30*B*a^3*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(t
an(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) + 90*A*a^2*b*d^2*log(4*(ta
n(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^
2 + tan(e)^2 + 1)) - 90*C*a^2*b*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)
*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 90*B*a*b^
2*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e
)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 30*A*b^3*d^2*log(4*(tan(f*x)^2*tan(e)^2
 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)
) + 30*C*b^3*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f
*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) + 60*C*a^3*c^2*tan(f*x) + 180*
B*a^2*b*c^2*tan(f*x) + 180*A*a*b^2*c^2*tan(f*x) - 180*C*a*b^2*c^2*tan(f*x)
- 60*B*b^3*c^2*tan(f*x) + 120*B*a^3*c*d*tan(f*x) + 360*A*a^2*b*c*d*tan(f*x)
 - 360*C*a^2*b*c*d*tan(f*x) - 360*B*a*b^2*c*d*tan(f*x) - 120*A*b^3*c*d*tan(
f*x) + 120*C*b^3*c*d*tan(f*x) + 60*A*a^3*d^2*tan(f*x) - 60*C*a^3*d^2*tan(f*
x) - 180*B*a^2*b*d^2*tan(f*x) - 180*A*a*b^2*d^2*tan(f*x) + 180*C*a*b^2*d^2*
tan(f*x) + 60*B*b^3*d^2*tan(f*x) + 60*C*a^3*c^2*tan(e) + 180*B*a^2*b*c^2*ta

```
n(e) + 180*A*a*b^2*c^2*tan(e) - 180*C*a*b^2*c^2*tan(e) - 60*B*b^3*c^2*tan(e
) + 120*B*a^3*c*d*tan(e) + 360*A*a^2*b*c*d*tan(e) - 360*C*a^2*b*c*d*tan(e)
- 360*B*a*b^2*c*d*tan(e) - 120*A*b^3*c*d*tan(e) + 120*C*b^3*c*d*tan(e) + 60
*A*a^3*d^2*tan(e) - 60*C*a^3*d^2*tan(e) - 180*B*a^2*b*d^2*tan(e) - 180*A*a*
b^2*d^2*tan(e) + 180*C*a*b^2*d^2*tan(e) + 60*B*b^3*d^2*tan(e) + 90*C*a^2*b*
c^2 + 90*B*a*b^2*c^2 + 30*A*b^3*c^2 - 45*C*b^3*c^2 + 60*C*a^3*c*d + 180*B*a
^2*b*c*d + 180*A*a*b^2*c*d - 270*C*a*b^2*c*d - 90*B*b^3*c*d + 30*B*a^3*d^2
+ 90*A*a^2*b*d^2 - 135*C*a^2*b*d^2 - 135*B*a*b^2*d^2 - 45*A*b^3*d^2 + 55*C*
b^3*d^2)/(f*tan(f*x)^6*tan(e)^6 - 6*f*tan(f*x)^5*tan(e)^5 + 15*f*tan(f*x)^4
*tan(e)^4 - 20*f*tan(f*x)^3*tan(e)^3 + 15*f*tan(f*x)^2*tan(e)^2 - 6*f*tan(f
*x)*tan(e) + f)
```

Mupad [B] (verification not implemented)

Time = 8.86 (sec) , antiderivative size = 891, normalized size of antiderivative = 1.35

$$\int (a+b\tan(e+fx))^3(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= x\left(A\,a^3\,c^2-A\,a^3\,d^2+B\,b^3\,c^2-C\,a^3\,c^2-B\,b^3\,d^2+C\,a^3\,d^2+2\,A\,b^3\,c\,d-2\,B\,a^3\,c\,d-2\,C\,b^3\,c\,d-3\,A\,a\,b^2\,c^2+3\,A\,a\,b^2\,d^2-3\,B\,a^2\,b\,c^2+3\,B\,a^2\,b\,d^2+3\,C\,a\,b^2\,c^2-3\,C\,a\,b^2\,d^2-6\,A\,a^2\,b\,c\,d+6\,B\,a\,b^2\,c\,d+6\,C\,a^2\,b\,c\,d\right)$$
$$-\frac{\tan(e+f\,x)\,\left(B\,b^3\,c^2-A\,a^3\,d^2-b^2\,d\,(B\,b\,d+3\,C\,a\,d+2\,C\,b\,c)-C\,a^3\,c^2+C\,a^3\,d^2+2\,A\,b^3\,c\,d-2\,B\,a^3\right.}{}$$
$$-\frac{\ln\left(\tan(e+f\,x)^2+1\right)\,\left(\frac{A\,b^3\,c^2}{2}-\frac{B\,a^3\,c^2}{2}-\frac{A\,b^3\,d^2}{2}+\frac{B\,a^3\,d^2}{2}-\frac{C\,b^3\,c^2}{2}+\frac{C\,b^3\,d^2}{2}-A\,a^3\,c\,d-B\,b^3\,c\,d+C\,a^3\,c\,d\right.}{}$$
$$+\frac{\tan(e+f\,x)^4\,\left(\frac{A\,b^3\,d^2}{4}+\frac{C\,b^3\,c^2}{4}-\frac{C\,b^3\,d^2}{4}+\frac{B\,b^3\,c\,d}{2}+\frac{3\,B\,a\,b^2\,d^2}{4}+\frac{3\,C\,a^2\,b\,d^2}{4}+\frac{3\,C\,a\,b^2\,c\,d}{2}\right)}{f}$$
$$+\frac{\tan(e+f\,x)^3\,\left(\frac{B\,b^3\,c^2}{3}-\frac{b^2\,d\,(B\,b\,d+3\,C\,a\,d+2\,C\,b\,c)}{3}+\frac{C\,a^3\,d^2}{3}+\frac{2\,A\,b^3\,c\,d}{3}+A\,a\,b^2\,d^2+B\,a^2\,b\,d^2+C\,a\,b^2\,c^2+2\,B\,a\right.}{f}$$
$$+\frac{\tan(e+f\,x)^2\,\left(\frac{A\,b^3\,c^2}{2}-\frac{A\,b^3\,d^2}{2}+\frac{B\,a^3\,d^2}{2}-\frac{C\,b^3\,c^2}{2}+\frac{C\,b^3\,d^2}{2}-B\,b^3\,c\,d+C\,a^3\,c\,d+\frac{3\,A\,a^2\,b\,d^2}{2}+\frac{3\,B\,a\,b^2\,c^2}{2}-\frac{3\,B\,a}{}\right.}{f}$$
$$+\frac{b^2\,d\tan(e+f\,x)^5\,(B\,b\,d+3\,C\,a\,d+2\,C\,b\,c)}{5\,f}+\frac{C\,b^3\,d^2\tan(e+f\,x)^6}{6\,f}$$

```
[In] int((a + b*tan(e + f*x))^3*(c + d*tan(e + f*x))^2*(A + B*tan(e + f*x) + C*t
an(e + f*x)^2),x)
```

```
[Out] x*(A*a^3*c^2 - A*a^3*d^2 + B*b^3*c^2 - C*a^3*c^2 - B*b^3*d^2 + C*a^3*d^2 +
2*A*b^3*c*d - 2*B*a^3*c*d - 2*C*b^3*c*d - 3*A*a*b^2*c^2 + 3*A*a*b^2*d^2 - 3
*B*a^2*b*c^2 + 3*B*a^2*b*d^2 + 3*C*a*b^2*c^2 - 3*C*a*b^2*d^2 - 6*A*a^2*b*c*
d + 6*B*a*b^2*c*d + 6*C*a^2*b*c*d) - (tan(e + f*x)*(B*b^3*c^2 - A*a^3*d^2 -
```

b^2*d*(B*b*d + 3*C*a*d + 2*C*b*c) - C*a^3*c^2 + C*a^3*d^2 + 2*A*b^3*c*d - 2*B*a^3*c*d - 3*A*a*b^2*c^2 + 3*A*a*b^2*d^2 - 3*B*a^2*b*c^2 + 3*B*a^2*b*d^2 + 3*C*a*b^2*c^2 - 6*A*a^2*b*c*d + 6*B*a*b^2*c*d + 6*C*a^2*b*c*d))/f - (log(tan(e + f*x)^2 + 1)*((A*b^3*c^2)/2 - (B*a^3*c^2)/2 - (A*b^3*d^2)/2 + (B*a^3*d^2)/2 - (C*b^3*c^2)/2 + (C*b^3*d^2)/2 - A*a^3*c*d - B*b^3*c*d + C*a^3*c*d - (3*A*a^2*b*c^2)/2 + (3*A*a^2*b*d^2)/2 + (3*B*a*b^2*c^2)/2 - (3*B*a*b^2*d^2)/2 + (3*C*a^2*b*c^2)/2 - (3*C*a^2*b*d^2)/2 + 3*A*a*b^2*c*d + 3*B*a^2*b*c*d - 3*C*a*b^2*c*d))/f + (tan(e + f*x)^4*((A*b^3*d^2)/4 + (C*b^3*c^2)/4 - (C*b^3*d^2)/4 + (B*b^3*c*d)/2 + (3*B*a*b^2*d^2)/4 + (3*C*a^2*b*d^2)/4 + (3*C*a*b^2*c*d)/2))/f + (tan(e + f*x)^3*((B*b^3*c^2)/3 - (b^2*d*(B*b*d + 3*C*a*d + 2*C*b*c))/3 + (C*a^3*d^2)/3 + (2*A*b^3*c*d)/3 + A*a*b^2*d^2 + B*a^2*b*d^2 + C*a*b^2*c^2 + 2*B*a*b^2*c*d + 2*C*a^2*b*c*d))/f + (tan(e + f*x)^2*((A*b^3*c^2)/2 - (A*b^3*d^2)/2 + (B*a^3*d^2)/2 - (C*b^3*c^2)/2 + (C*b^3*d^2)/2 - B*b^3*c*d + C*a^3*c*d + (3*A*a^2*b*d^2)/2 + (3*B*a*b^2*c^2)/2 - (3*B*a*b^2*d^2)/2 + (3*C*a^2*b*c^2)/2 - (3*C*a^2*b*d^2)/2 + 3*A*a*b^2*c*d + 3*B*a^2*b*c*d - 3*C*a*b^2*c*d))/f + (b^2*d*tan(e + f*x)^5*(B*b*d + 3*C*a*d + 2*C*b*c))/(5*f) + (C*b^3*d^2*tan(e + f*x)^6)/(6*f)

3.58 $\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^2\left(A+B\tan(e+\right.$

Optimal result 502
Rubi [A] (verified) 503
Mathematica [C] (verified) 506
Maple [A] (verified) 507
Fricas [A] (verification not implemented) 508
Sympy [B] (verification not implemented) 508
Maxima [A] (verification not implemented) 509
Giac [B] (verification not implemented) 510
Mupad [B] (verification not implemented) 517

Optimal result

Integrand size = 45, antiderivative size = 443

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= -\left(\left(a^2\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-b^2\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)\right.\right.$$
$$\left.\left.+2ab\left(2c(A-C)d+B\left(c^2-d^2\right)\right)\right)x\right)$$
$$+\frac{(2ab(c^2C+2Bcd-Cd^2-A(c^2-d^2))-a^2(2c(A-C)d+B(c^2-d^2))+b^2(2c(A-C)d+B(c^2-d^2)))}{f}$$
$$+\frac{d(2ab(Ac-cC-Bd)+a^2(Bc+(A-C)d)-b^2(Bc+(A-C)d))\tan(e+fx)}{f}$$
$$+\frac{(a^2B-b^2B+2ab(A-C))\,(c+d\tan(e+fx))^2}{2f}$$
$$+\frac{(8a^2Cd^2-10abd(cC-4Bd)+b^2(2c^2C-5Bcd+20(A-C)d^2))\,(c+d\tan(e+fx))^3}{60d^3f}$$
$$-\frac{b(2bcC-5bBd-2aCd)\tan(e+fx)(c+d\tan(e+fx))^3}{20d^2f}$$
$$+\frac{C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^3}{5df}$$

```
[Out] -(a^2*(c^2*C+2*B*c*d-C*d^2-A*(c^2-d^2))-b^2*(c^2*C+2*B*c*d-C*d^2-A*(c^2-d^2
))+2*a*b*(2*c*(A-C)*d+B*(c^2-d^2)))*x+(2*a*b*(c^2*C+2*B*c*d-C*d^2-A*(c^2-d^
2))-a^2*(2*c*(A-C)*d+B*(c^2-d^2))+b^2*(2*c*(A-C)*d+B*(c^2-d^2)))*ln(cos(f*x
+e))/f+d*(2*a*b*(A*c-B*d-C*c)+a^2*(B*c+(A-C)*d)-b^2*(B*c+(A-C)*d))*tan(f*x+
e)/f+1/2*(B*a^2-B*b^2+2*a*b*(A-C))*(c+d*tan(f*x+e))^2/f+1/60*(8*a^2*C*d^2-1
0*a*b*d*(-4*B*d+C*c)+b^2*(2*c^2*C-5*B*c*d+20*(A-C)*d^2))*(c+d*tan(f*x+e))^3
/d^3/f-1/20*b*(-5*B*b*d-2*C*a*d+2*C*b*c)*tan(f*x+e)*(c+d*tan(f*x+e))^3/d^2/
f+1/5*C*(a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^3/d/f
```

Rubi [A] (verified)

Time = 1.35 (sec) , antiderivative size = 443, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}} = 0.133$, Rules used = {3728, 3718, 3711, 3609, 3606, 3556}

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$

$$= \frac{\log(\cos(e+fx))\left(-(a^2(2cd(A-C)+B(c^2-d^2)))+2ab(-A(c^2-d^2)+2Bcd+c^2C-Cd^2)+b^2(2cd(A\right.}{f}$$

$$-x\left(a^2\left(-A\left(c^2-d^2\right)+2Bcd+c^2C-Cd^2\right)+2ab\left(2cd(A-C)+B\left(c^2-d^2\right)\right)\right.$$
$$\left.-b^2\left(-A\left(c^2-d^2\right)+2Bcd+c^2C-Cd^2\right)\right)$$
$$+\frac{(c+d\tan(e+fx))^3\left(8a^2Cd^2-10abd(cC-4Bd)+b^2(20d^2(A-C)-5Bcd+2c^2C)\right)}{60d^3f}$$
$$+\frac{(a^2B+2ab(A-C)-b^2B)\,(c+d\tan(e+fx))^2}{2f}$$
$$+\frac{d\tan(e+fx)\left(a^2(d(A-C)+Bc)+2ab(Ac-Bd-cC)-b^2(d(A-C)+Bc)\right)}{f}$$
$$-\frac{b\tan(e+fx)(-2aCd-5bBd+2bcC)(c+d\tan(e+fx))^3}{20d^2f}$$
$$+\frac{C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^3}{5df}$$

```
[In] Int[(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*T
an[e + f*x]^2),x]
```

```
[Out] -((a^2*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - b^2*(c^2*C + 2*B*c*d - C
*d^2 - A*(c^2 - d^2)) + 2*a*b*(2*c*(A - C)*d + B*(c^2 - d^2)))*x) + ((2*a*b
*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - a^2*(2*c*(A - C)*d + B*(c^2 -
d^2)) + b^2*(2*c*(A - C)*d + B*(c^2 - d^2)))*Log[Cos[e + f*x]])/f + (d*(2*a
*b*(A*c - c*C - B*d) + a^2*(B*c + (A - C)*d) - b^2*(B*c + (A - C)*d))*Tan[e
 + f*x])/f + ((a^2*B - b^2*B + 2*a*b*(A - C))*(c + d*Tan[e + f*x])^2)/(2*f)
 + ((8*a^2*C*d^2 - 10*a*b*d*(c*C - 4*B*d) + b^2*(2*c^2*C - 5*B*c*d + 20*(A
- C)*d^2))*(c + d*Tan[e + f*x])^3)/(60*d^3*f) - (b*(2*b*c*C - 5*b*B*d - 2*a
*C*d)*Tan[e + f*x]*(c + d*Tan[e + f*x])^3)/(20*d^2*f) + (C*(a + b*Tan[e + f
*x])^2*(c + d*Tan[e + f*x])^3)/(5*d*f)
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3606

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
```

```
 f*x], x], x] + Simp[b*d*(Tan[e + f*x]/f), x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3609

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[d*((a + b*Tan[e + f*x])^m/(f*m)), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\text{integral} = \frac{C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^3}{5df}$$
$$+\frac{\int (a+b\tan(e+fx))(c+d\tan(e+fx))^2\left(-2bcC+a(5A-3C)d+5(Ab+aB-bC)d\tan(e+fx)-\right.}{5d}$$
$$=-\frac{b(2bcC-5bBd-2aCd)\tan(e+fx)(c+d\tan(e+fx))^3}{20d^2f}$$
$$+\frac{C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^3}{5df}$$
$$-\frac{\int (c+d\tan(e+fx))^2\left(10abcCd-4a^2(5A-3C)d^2-b^2c(2cC-5Bd)-20(a^2B-b^2B+2ab(\right.}{}$$
$$=\frac{\left(8a^2Cd^2-10abd(cC-4Bd)+b^2(2c^2C-5Bcd+20(A-C)d^2)\right)(c+d\tan(e+fx))^3}{60d^3f}$$
$$-\frac{b(2bcC-5bBd-2aCd)\tan(e+fx)(c+d\tan(e+fx))^3}{20d^2f}$$
$$+\frac{C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^3}{5df}$$
$$-\frac{\int (c+d\tan(e+fx))^2\left(20\left(2abB-a^2(A-C)+b^2(A-C)\right)d^2-20(a^2B-b^2B+2ab(A-C))\right.}{20d^2}$$
$$=\frac{\left(a^2B-b^2B+2ab(A-C)\right)(c+d\tan(e+fx))^2}{2f}$$
$$+\frac{\left(8a^2Cd^2-10abd(cC-4Bd)+b^2(2c^2C-5Bcd+20(A-C)d^2)\right)(c+d\tan(e+fx))^3}{60d^3f}$$
$$-\frac{b(2bcC-5bBd-2aCd)\tan(e+fx)(c+d\tan(e+fx))^3}{20d^2f}$$
$$+\frac{C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^3}{5df}$$
$$-\frac{\int (c+d\tan(e+fx))\left(-20d^2(a^2(Ac-cC-Bd)-b^2(Ac-cC-Bd)-2ab(Bc+(A-C)d))\right.}{20d^2}$$

$$= -\left(\left(a^2\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-b^2\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)+2ab\left(2c(A-C)d+B\left(c^2-d^2\right)\right)\right)x\right) + \frac{d(2ab(Ac-cC-Bd)+a^2(Bc+(A-C)d)-b^2(Bc+(A-C)d))\tan(e+fx)}{f} + \frac{(a^2B-b^2B+2ab(A-C))(c+d\tan(e+fx))^2}{2f} + \frac{(8a^2Cd^2-10abd(cC-4Bd)+b^2(2c^2C-5Bcd+20(A-C)d^2))(c+d\tan(e+fx))^3}{60d^3f} - \frac{b(2bcC-5bBd-2aCd)\tan(e+fx)(c+d\tan(e+fx))^3}{20d^2f} + \frac{C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^3}{5df} - \left(2ab\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-a^2\left(2c(A-C)d+B\left(c^2-d^2\right)\right)+b^2\left(2c(A-C)d+B\left(c^2-d^2\right)\right)\right)\int\tan(e+fx)\,dx$$

$$= -\left(\left(a^2\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-b^2\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)+2ab\left(2c(A-C)d+B\left(c^2-d^2\right)\right)\right)x\right) + \frac{(2ab(c^2C+2Bcd-Cd^2-A(c^2-d^2))-a^2(2c(A-C)d+B(c^2-d^2))+b^2(2c(A-C)d+B(c^2}{f} + \frac{d(2ab(Ac-cC-Bd)+a^2(Bc+(A-C)d)-b^2(Bc+(A-C)d))\tan(e+fx)}{f} + \frac{(a^2B-b^2B+2ab(A-C))(c+d\tan(e+fx))^2}{2f} + \frac{(8a^2Cd^2-10abd(cC-4Bd)+b^2(2c^2C-5Bcd+20(A-C)d^2))(c+d\tan(e+fx))^3}{60d^3f} - \frac{b(2bcC-5bBd-2aCd)\tan(e+fx)(c+d\tan(e+fx))^3}{20d^2f} + \frac{C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^3}{5df}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 6.55 (sec) , antiderivative size = 383, normalized size of antiderivative = 0.86

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$

$$= \frac{C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^3}{5df} + \frac{\frac{b(-2bcC+5bBd+2aCd)\tan(e+fx)(c+d\tan(e+fx))^3}{4df} - \frac{\left(-8a^2Cd^2+10abd(cC-4Bd)-b^2\left(2c^2C-5Bcd+20(A-C)d^2\right)\right)(c+d\tan(e+fx))^3}{3df} - \frac{10\left(d\left(2ab(\right.\right.}{}}{}$$

[In] Integrate[(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2),x]

[Out] (C*(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])^3)/(5*d*f) + ((b*(-2*b*c*C + 5*b*B*d + 2*a*C*d)*Tan[e + f*x]*(c + d*Tan[e + f*x])^3)/(4*d*f) - (((-8*a^2*C*d^2 + 10*a*b*d*(c*C - 4*B*d) - b^2*(2*c^2*C - 5*B*c*d + 20*(A - C)*d^2))*(c + d*Tan[e + f*x])^3)/(3*d*f) - (10*(d*(2*a*b*(A*c - c*C + B*d) + a^2*(B*c - (A - C)*d) - b^2*(B*c - (A - C)*d))*(I*(c + I*d)^2*Log[I - Tan[e + f*x]] - I*(c - I*d)^2*Log[I + Tan[e + f*x]] - 2*d^2*Tan[e + f*x]) + (a^2*B - b^2*B + 2*a*b*(A - C))*d*((I*c - d)^3*Log[I - Tan[e + f*x]] - (I*c + d)^3*Log[I + Tan[e + f*x]] + 6*c*d^2*Tan[e + f*x] + d^3*Tan[e + f*x]^2)))/f)/(4*d))/(5*d)

Maple [A] (verified)

Time = 0.26 (sec) , antiderivative size = 392, normalized size of antiderivative = 0.88

method	result
parts	$\frac{(2Aa^2cd+2Aabc^2+Ba^2c^2)\ln\left(1+\tan(fx+e)^2\right)}{2f}+\frac{(Bb^2d^2+2Cabd^2+2Cb^2cd)\left(\frac{\tan(fx+e)^4}{4}-\frac{\tan(fx+e)^2}{2}+\frac{\ln(1+\tan(}{2}\right.}{f}$
norman	$(Aa^2c^2-Aa^2d^2-4Aabcd-Ab^2c^2+Ab^2d^2-2Ba^2cd-2Babc^2+2Babd^2+2Bb^2c$
derivativedivides	$\frac{\frac{Ca^2d^2\tan(fx+e)^3}{3}+\frac{Cb^2c^2\tan(fx+e)^3}{3}-\frac{Cb^2d^2\tan(fx+e)^3}{3}+\frac{Ba^2d^2\tan(fx+e)^2}{2}+\frac{Bb^2c^2\tan(fx+e)^2}{2}-\frac{Bb^2d^2\tan(fx+e)^2}{2}-t}{}$
default	$\frac{\frac{Ca^2d^2\tan(fx+e)^3}{3}+\frac{Cb^2c^2\tan(fx+e)^3}{3}-\frac{Cb^2d^2\tan(fx+e)^3}{3}+\frac{Ba^2d^2\tan(fx+e)^2}{2}+\frac{Bb^2c^2\tan(fx+e)^2}{2}-\frac{Bb^2d^2\tan(fx+e)^2}{2}-t}{}$
parallelrisch	$\frac{20Ca^2d^2\tan(fx+e)^3+20Cb^2c^2\tan(fx+e)^3-20Cb^2d^2\tan(fx+e)^3+30Ba^2d^2\tan(fx+e)^2+30Bb^2c^2\tan(fx+e)^2-30}{}$
risch	Expression too large to display

[In] int((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x,method=_RETURNVERBOSE)

[Out] 1/2*(2*A*a^2*c*d+2*A*a*b*c^2+B*a^2*c^2)/f*ln(1+tan(f*x+e)^2)+(B*b^2*d^2+2*C*a*b*d^2+2*C*b^2*c*d)/f*(1/4*tan(f*x+e)^4-1/2*tan(f*x+e)^2+1/2*ln(1+tan(f*x+e)^2))+(A*b^2*d^2+2*B*a*b*d^2+2*B*b^2*c*d+C*a^2*d^2+4*C*a*b*c*d+C*b^2*c^2)/f*(1/3*tan(f*x+e)^3-tan(f*x+e)+arctan(tan(f*x+e)))+(A*a^2*d^2+4*A*a*b*c*d+A*b^2*c^2+2*B*a^2*c*d+2*B*a*b*c^2+C*a^2*c^2)/f*(tan(f*x+e)-arctan(tan(f*x+e)))+(2*A*a*b*d^2+2*A*b^2*c*d+B*a^2*d^2+4*B*a*b*c*d+B*b^2*c^2+2*C*a^2*c*d+2*C*a*b*c^2)/f*(1/2*tan(f*x+e)^2-1/2*ln(1+tan(f*x+e)^2))+A*a^2*c^2*x+C*b^2*d^2/f*(1/5*tan(f*x+e)^5-1/3*tan(f*x+e)^3+tan(f*x+e)-arctan(tan(f*x+e)))

Fricas [A] (verification not implemented)

none

Time = 0.26 (sec) , antiderivative size = 462, normalized size of antiderivative = 1.04

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\frac{12\,Cb^2d^2\tan\left(fx+e\right)^5+15\left(2\,Cb^2cd+(2\,Cab+Bb^2)d^2\right)\tan\left(fx+e\right)^4+20\left(Cb^2c^2+2\left(2\,Cab+Bb^2\right)cd+\right.}{}$$

```
[In] integrate((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e
)^2),x, algorithm="fricas")
```

```
[Out] 1/60*(12*C*b^2*d^2*tan(f*x + e)^5 + 15*(2*C*b^2*c*d + (2*C*a*b + B*b^2)*d^2
)*tan(f*x + e)^4 + 20*(C*b^2*c^2 + 2*(2*C*a*b + B*b^2)*c*d + (C*a^2 + 2*B*a
*b + (A - C)*b^2)*d^2)*tan(f*x + e)^3 + 60*(((A - C)*a^2 - 2*B*a*b - (A - C
)*b^2)*c^2 - 2*(B*a^2 + 2*(A - C)*a*b - B*b^2)*c*d - ((A - C)*a^2 - 2*B*a*b
 - (A - C)*b^2)*d^2)*f*x + 30*((2*C*a*b + B*b^2)*c^2 + 2*(C*a^2 + 2*B*a*b +
 (A - C)*b^2)*c*d + (B*a^2 + 2*(A - C)*a*b - B*b^2)*d^2)*tan(f*x + e)^2 - 3
0*((B*a^2 + 2*(A - C)*a*b - B*b^2)*c^2 + 2*((A - C)*a^2 - 2*B*a*b - (A - C)
*b^2)*c*d - (B*a^2 + 2*(A - C)*a*b - B*b^2)*d^2)*log(1/(tan(f*x + e)^2 + 1)
) + 60*((C*a^2 + 2*B*a*b + (A - C)*b^2)*c^2 + 2*(B*a^2 + 2*(A - C)*a*b - B*
b^2)*c*d + ((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*d^2)*tan(f*x + e))/f
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1134 vs. 2(396) = 792.

Time = 0.30 (sec) , antiderivative size = 1134, normalized size of antiderivative = 2.56

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))**2*(c+d*tan(f*x+e))**2*(A+B*tan(f*x+e)+C*tan(f*x
+e)**2),x)
```

```
[Out] Piecewise((A*a**2*c**2*x + A*a**2*c*d*log(tan(e + f*x)**2 + 1)/f - A*a**2*d
**2*x + A*a**2*d**2*tan(e + f*x)/f + A*a*b*c**2*log(tan(e + f*x)**2 + 1)/f
- 4*A*a*b*c*d*x + 4*A*a*b*c*d*tan(e + f*x)/f - A*a*b*d**2*log(tan(e + f*x)*
*2 + 1)/f + A*a*b*d**2*tan(e + f*x)**2/f - A*b**2*c**2*x + A*b**2*c**2*tan(
e + f*x)/f - A*b**2*c*d*log(tan(e + f*x)**2 + 1)/f + A*b**2*c*d*tan(e + f*x
)**2/f + A*b**2*d**2*x + A*b**2*d**2*tan(e + f*x)**3/(3*f) - A*b**2*d**2*ta
n(e + f*x)/f + B*a**2*c**2*log(tan(e + f*x)**2 + 1)/(2*f) - 2*B*a**2*c*d*x
+ 2*B*a**2*c*d*tan(e + f*x)/f - B*a**2*d**2*log(tan(e + f*x)**2 + 1)/(2*f)
```

```
+ B*a**2*d**2*tan(e + f*x)**2/(2*f) - 2*B*a*b*c**2*x + 2*B*a*b*c**2*tan(e +
 f*x)/f - 2*B*a*b*c*d*log(tan(e + f*x)**2 + 1)/f + 2*B*a*b*c*d*tan(e + f*x)
**2/f + 2*B*a*b*d**2*x + 2*B*a*b*d**2*tan(e + f*x)**3/(3*f) - 2*B*a*b*d**2*
tan(e + f*x)/f - B*b**2*c**2*log(tan(e + f*x)**2 + 1)/(2*f) + B*b**2*c**2*t
an(e + f*x)**2/(2*f) + 2*B*b**2*c*d*x + 2*B*b**2*c*d*tan(e + f*x)**3/(3*f)
- 2*B*b**2*c*d*tan(e + f*x)/f + B*b**2*d**2*log(tan(e + f*x)**2 + 1)/(2*f)
+ B*b**2*d**2*tan(e + f*x)**4/(4*f) - B*b**2*d**2*tan(e + f*x)**2/(2*f) - C
*a**2*c**2*x + C*a**2*c**2*tan(e + f*x)/f - C*a**2*c*d*log(tan(e + f*x)**2
+ 1)/f + C*a**2*c*d*tan(e + f*x)**2/f + C*a**2*d**2*x + C*a**2*d**2*tan(e +
 f*x)**3/(3*f) - C*a**2*d**2*tan(e + f*x)/f - C*a*b*c**2*log(tan(e + f*x)**
2 + 1)/f + C*a*b*c**2*tan(e + f*x)**2/f + 4*C*a*b*c*d*x + 4*C*a*b*c*d*tan(e
 + f*x)**3/(3*f) - 4*C*a*b*c*d*tan(e + f*x)/f + C*a*b*d**2*log(tan(e + f*x)
**2 + 1)/f + C*a*b*d**2*tan(e + f*x)**4/(2*f) - C*a*b*d**2*tan(e + f*x)**2/
f + C*b**2*c**2*x + C*b**2*c**2*tan(e + f*x)**3/(3*f) - C*b**2*c**2*tan(e +
 f*x)/f + C*b**2*c*d*log(tan(e + f*x)**2 + 1)/f + C*b**2*c*d*tan(e + f*x)**
4/(2*f) - C*b**2*c*d*tan(e + f*x)**2/f - C*b**2*d**2*x + C*b**2*d**2*tan(e
+ f*x)**5/(5*f) - C*b**2*d**2*tan(e + f*x)**3/(3*f) + C*b**2*d**2*tan(e + f
*x)/f, Ne(f, 0)), (x*(a + b*tan(e))**2*(c + d*tan(e))**2*(A + B*tan(e) + C*
tan(e)**2), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.39 (sec) , antiderivative size = 463, normalized size of antiderivative = 1.05

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^2 \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= \frac{12\,Cb^2d^2\tan{(fx+e)}^5+15\,(2\,Cb^2cd+(2\,Cab+Bb^2)d^2)\tan{(fx+e)}^4+20\,(Cb^2c^2+2\,(2\,Cab+Bb^2)cd}{}$$

```
[In] integrate((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e
)^2),x, algorithm="maxima")
```

```
[Out] 1/60*(12*C*b^2*d^2*tan(f*x + e)^5 + 15*(2*C*b^2*c*d + (2*C*a*b + B*b^2)*d^2
)*tan(f*x + e)^4 + 20*(C*b^2*c^2 + 2*(2*C*a*b + B*b^2)*c*d + (C*a^2 + 2*B*a
*b + (A - C)*b^2)*d^2)*tan(f*x + e)^3 + 30*((2*C*a*b + B*b^2)*c^2 + 2*(C*a^
2 + 2*B*a*b + (A - C)*b^2)*c*d + (B*a^2 + 2*(A - C)*a*b - B*b^2)*d^2)*tan(f
*x + e)^2 + 60*(((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*c^2 - 2*(B*a^2 + 2*(A
 - C)*a*b - B*b^2)*c*d - ((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*d^2)*(f*x +
e) + 30*((B*a^2 + 2*(A - C)*a*b - B*b^2)*c^2 + 2*((A - C)*a^2 - 2*B*a*b - (
A - C)*b^2)*c*d - (B*a^2 + 2*(A - C)*a*b - B*b^2)*d^2)*log(tan(f*x + e)^2 +
 1) + 60*((C*a^2 + 2*B*a*b + (A - C)*b^2)*c^2 + 2*(B*a^2 + 2*(A - C)*a*b -
B*b^2)*c*d + ((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*d^2)*tan(f*x + e))/f
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 11957 vs. 2(436) = 872.

Time = 10.52 (sec) , antiderivative size = 11957, normalized size of antiderivative = 26.99

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= \text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e
)^2),x, algorithm="giac")
```

```
[Out] 1/60*(60*A*a^2*c^2*f*x*tan(f*x)^5*tan(e)^5 - 60*C*a^2*c^2*f*x*tan(f*x)^5*ta
n(e)^5 - 120*B*a*b*c^2*f*x*tan(f*x)^5*tan(e)^5 - 60*A*b^2*c^2*f*x*tan(f*x)^
5*tan(e)^5 + 60*C*b^2*c^2*f*x*tan(f*x)^5*tan(e)^5 - 120*B*a^2*c*d*f*x*tan(f
*x)^5*tan(e)^5 - 240*A*a*b*c*d*f*x*tan(f*x)^5*tan(e)^5 + 240*C*a*b*c*d*f*x*
tan(f*x)^5*tan(e)^5 + 120*B*b^2*c*d*f*x*tan(f*x)^5*tan(e)^5 - 60*A*a^2*d^2*
f*x*tan(f*x)^5*tan(e)^5 + 60*C*a^2*d^2*f*x*tan(f*x)^5*tan(e)^5 + 120*B*a*b*
d^2*f*x*tan(f*x)^5*tan(e)^5 + 60*A*b^2*d^2*f*x*tan(f*x)^5*tan(e)^5 - 60*C*b
^2*d^2*f*x*tan(f*x)^5*tan(e)^5 - 30*B*a^2*c^2*log(4*(tan(f*x)^2*tan(e)^2 -
2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*t
an(f*x)^5*tan(e)^5 - 60*A*a*b*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*t
an(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5*ta
n(e)^5 + 60*C*a*b*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(
tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5 + 30*
B*b^2*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*t
an(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5 - 60*A*a^2*c*d*lo
g(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + ta
n(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5 + 60*C*a^2*c*d*log(4*(tan(f*x
)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + t
an(e)^2 + 1))*tan(f*x)^5*tan(e)^5 + 120*B*a*b*c*d*log(4*(tan(f*x)^2*tan(e)^
2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1
))*tan(f*x)^5*tan(e)^5 + 60*A*b^2*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*
x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^
5*tan(e)^5 - 60*C*b^2*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) +
1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5 +
 30*B*a^2*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)
^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5 + 60*A*a*b*d^
2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2
+ tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5 - 60*C*a*b*d^2*log(4*(tan
(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2
 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5 - 30*B*b^2*d^2*log(4*(tan(f*x)^2*tan(
e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2
+ 1))*tan(f*x)^5*tan(e)^5 - 300*A*a^2*c^2*f*x*tan(f*x)^4*tan(e)^4 + 300*C*a
^2*c^2*f*x*tan(f*x)^4*tan(e)^4 + 600*B*a*b*c^2*f*x*tan(f*x)^4*tan(e)^4 + 30
```

0*A*b^2*c^2*f*x*tan(f*x)^4*tan(e)^4 - 300*C*b^2*c^2*f*x*tan(f*x)^4*tan(e)^4
 + 600*B*a^2*c*d*f*x*tan(f*x)^4*tan(e)^4 + 1200*A*a*b*c*d*f*x*tan(f*x)^4*ta
n(e)^4 - 1200*C*a*b*c*d*f*x*tan(f*x)^4*tan(e)^4 - 600*B*b^2*c*d*f*x*tan(f*x
)^4*tan(e)^4 + 300*A*a^2*d^2*f*x*tan(f*x)^4*tan(e)^4 - 300*C*a^2*d^2*f*x*ta
n(f*x)^4*tan(e)^4 - 600*B*a*b*d^2*f*x*tan(f*x)^4*tan(e)^4 - 300*A*b^2*d^2*f
*x*tan(f*x)^4*tan(e)^4 + 300*C*b^2*d^2*f*x*tan(f*x)^4*tan(e)^4 + 60*C*a*b*c
^2*tan(f*x)^5*tan(e)^5 + 30*B*b^2*c^2*tan(f*x)^5*tan(e)^5 + 60*C*a^2*c*d*ta
n(f*x)^5*tan(e)^5 + 120*B*a*b*c*d*tan(f*x)^5*tan(e)^5 + 60*A*b^2*c*d*tan(f*
x)^5*tan(e)^5 - 90*C*b^2*c*d*tan(f*x)^5*tan(e)^5 + 30*B*a^2*d^2*tan(f*x)^5*
tan(e)^5 + 60*A*a*b*d^2*tan(f*x)^5*tan(e)^5 - 90*C*a*b*d^2*tan(f*x)^5*tan(e
)^5 - 45*B*b^2*d^2*tan(f*x)^5*tan(e)^5 + 150*B*a^2*c^2*log(4*(tan(f*x)^2*ta
n(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^
2 + 1))*tan(f*x)^4*tan(e)^4 + 300*A*a*b*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*
tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan
(f*x)^4*tan(e)^4 - 300*C*a*b*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*ta
n(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan
(e)^4 - 150*B*b^2*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(
tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 300
*A*a^2*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*
tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 - 300*C*a^2*c*d*
log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 +
tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 - 600*B*a*b*c*d*log(4*(tan(
f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2
+ tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 - 300*A*b^2*c*d*log(4*(tan(f*x)^2*tan(
e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2
+ 1))*tan(f*x)^4*tan(e)^4 + 300*C*b^2*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*ta
n(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f
*x)^4*tan(e)^4 - 150*B*a^2*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(
e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e
)^4 - 300*A*a*b*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(ta
n(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 300*C
*a*b*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*ta
n(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 150*B*b^2*d^2*lo
g(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + ta
n(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 - 60*C*a^2*c^2*tan(f*x)^5*tan
(e)^4 - 120*B*a*b*c^2*tan(f*x)^5*tan(e)^4 - 60*A*b^2*c^2*tan(f*x)^5*tan(e)^
4 + 60*C*b^2*c^2*tan(f*x)^5*tan(e)^4 - 120*B*a^2*c*d*tan(f*x)^5*tan(e)^4 -
240*A*a*b*c*d*tan(f*x)^5*tan(e)^4 + 240*C*a*b*c*d*tan(f*x)^5*tan(e)^4 + 120
*B*b^2*c*d*tan(f*x)^5*tan(e)^4 - 60*A*a^2*d^2*tan(f*x)^5*tan(e)^4 + 60*C*a^
2*d^2*tan(f*x)^5*tan(e)^4 + 120*B*a*b*d^2*tan(f*x)^5*tan(e)^4 + 60*A*b^2*d^
2*tan(f*x)^5*tan(e)^4 - 60*C*b^2*d^2*tan(f*x)^5*tan(e)^4 - 60*C*a^2*c^2*tan
(f*x)^4*tan(e)^5 - 120*B*a*b*c^2*tan(f*x)^4*tan(e)^5 - 60*A*b^2*c^2*tan(f*x
)^4*tan(e)^5 + 60*C*b^2*c^2*tan(f*x)^4*tan(e)^5 - 120*B*a^2*c*d*tan(f*x)^4*
tan(e)^5 - 240*A*a*b*c*d*tan(f*x)^4*tan(e)^5 + 240*C*a*b*c*d*tan(f*x)^4*tan
(e)^5 + 120*B*b^2*c*d*tan(f*x)^4*tan(e)^5 - 60*A*a^2*d^2*tan(f*x)^4*tan(e)^

5 + 60*C*a^2*d^2*tan(f*x)^4*tan(e)^5 + 120*B*a*b*d^2*tan(f*x)^4*tan(e)^5 +
60*A*b^2*d^2*tan(f*x)^4*tan(e)^5 - 60*C*b^2*d^2*tan(f*x)^4*tan(e)^5 + 600*A
*a^2*c^2*f*x*tan(f*x)^3*tan(e)^3 - 600*C*a^2*c^2*f*x*tan(f*x)^3*tan(e)^3 -
1200*B*a*b*c^2*f*x*tan(f*x)^3*tan(e)^3 - 600*A*b^2*c^2*f*x*tan(f*x)^3*tan(e
)^3 + 600*C*b^2*c^2*f*x*tan(f*x)^3*tan(e)^3 - 1200*B*a^2*c*d*f*x*tan(f*x)^3
*tan(e)^3 - 2400*A*a*b*c*d*f*x*tan(f*x)^3*tan(e)^3 + 2400*C*a*b*c*d*f*x*tan
(f*x)^3*tan(e)^3 + 1200*B*b^2*c*d*f*x*tan(f*x)^3*tan(e)^3 - 600*A*a^2*d^2*f
*x*tan(f*x)^3*tan(e)^3 + 600*C*a^2*d^2*f*x*tan(f*x)^3*tan(e)^3 + 1200*B*a*b
*d^2*f*x*tan(f*x)^3*tan(e)^3 + 600*A*b^2*d^2*f*x*tan(f*x)^3*tan(e)^3 - 600*
C*b^2*d^2*f*x*tan(f*x)^3*tan(e)^3 + 60*C*a*b*c^2*tan(f*x)^5*tan(e)^3 + 30*B
*b^2*c^2*tan(f*x)^5*tan(e)^3 + 60*C*a^2*c*d*tan(f*x)^5*tan(e)^3 + 120*B*a*b
*c*d*tan(f*x)^5*tan(e)^3 + 60*A*b^2*c*d*tan(f*x)^5*tan(e)^3 - 60*C*b^2*c*d*
tan(f*x)^5*tan(e)^3 + 30*B*a^2*d^2*tan(f*x)^5*tan(e)^3 + 60*A*a*b*d^2*tan(f
*x)^5*tan(e)^3 - 60*C*a*b*d^2*tan(f*x)^5*tan(e)^3 - 30*B*b^2*d^2*tan(f*x)^5
*tan(e)^3 - 180*C*a*b*c^2*tan(f*x)^4*tan(e)^4 - 90*B*b^2*c^2*tan(f*x)^4*tan
(e)^4 - 180*C*a^2*c*d*tan(f*x)^4*tan(e)^4 - 360*B*a*b*c*d*tan(f*x)^4*tan(e)
^4 - 180*A*b^2*c*d*tan(f*x)^4*tan(e)^4 + 330*C*b^2*c*d*tan(f*x)^4*tan(e)^4
- 90*B*a^2*d^2*tan(f*x)^4*tan(e)^4 - 180*A*a*b*d^2*tan(f*x)^4*tan(e)^4 + 33
0*C*a*b*d^2*tan(f*x)^4*tan(e)^4 + 165*B*b^2*d^2*tan(f*x)^4*tan(e)^4 + 60*C*
a*b*c^2*tan(f*x)^3*tan(e)^5 + 30*B*b^2*c^2*tan(f*x)^3*tan(e)^5 + 60*C*a^2*c
*d*tan(f*x)^3*tan(e)^5 + 120*B*a*b*c*d*tan(f*x)^3*tan(e)^5 + 60*A*b^2*c*d*t
an(f*x)^3*tan(e)^5 - 60*C*b^2*c*d*tan(f*x)^3*tan(e)^5 + 30*B*a^2*d^2*tan(f*
x)^3*tan(e)^5 + 60*A*a*b*d^2*tan(f*x)^3*tan(e)^5 - 60*C*a*b*d^2*tan(f*x)^3*
tan(e)^5 - 30*B*b^2*d^2*tan(f*x)^3*tan(e)^5 - 20*C*b^2*c^2*tan(f*x)^5*tan(e
)^2 - 80*C*a*b*c*d*tan(f*x)^5*tan(e)^2 - 40*B*b^2*c*d*tan(f*x)^5*tan(e)^2 -
 20*C*a^2*d^2*tan(f*x)^5*tan(e)^2 - 40*B*a*b*d^2*tan(f*x)^5*tan(e)^2 - 20*A
*b^2*d^2*tan(f*x)^5*tan(e)^2 + 20*C*b^2*d^2*tan(f*x)^5*tan(e)^2 - 300*B*a^2
*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)
^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 600*A*a*b*c^2*log(4*
(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*
x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 600*C*a*b*c^2*log(4*(tan(f*x)^2
*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(
e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 300*B*b^2*c^2*log(4*(tan(f*x)^2*tan(e)^2 -
 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*
tan(f*x)^3*tan(e)^3 - 600*A*a^2*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)
*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*
tan(e)^3 + 600*C*a^2*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1
)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 +
1200*B*a*b*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x
)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 600*A*b^2*
c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^
2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 600*C*b^2*c*d*log(4*(
tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x
)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 300*B*a^2*d^2*log(4*(tan(f*x)^2*
tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e

)^2 + 1))*tan(f*x)^3*tan(e)^3 + 600*A*a*b*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 600*C*a*b*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 300*B*b^2*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 240*C*a^2*c^2*tan(f*x)^4*tan(e)^3 + 480*B*a*b*c^2*tan(f*x)^4*tan(e)^3 + 240*A*b^2*c^2*tan(f*x)^4*tan(e)^3 - 300*C*b^2*c^2*tan(f*x)^4*tan(e)^3 + 480*B*a^2*c*d*tan(f*x)^4*tan(e)^3 + 960*A*a*b*c*d*tan(f*x)^4*tan(e)^3 - 1200*C*a*b*c*d*tan(f*x)^4*tan(e)^3 - 600*B*b^2*c*d*tan(f*x)^4*tan(e)^3 + 240*A*a^2*d^2*tan(f*x)^4*tan(e)^3 - 300*C*a^2*d^2*tan(f*x)^4*tan(e)^3 - 600*B*a*b*d^2*tan(f*x)^4*tan(e)^3 - 300*A*b^2*d^2*tan(f*x)^4*tan(e)^3 + 300*C*b^2*d^2*tan(f*x)^4*tan(e)^3 + 240*C*a^2*c^2*tan(f*x)^3*tan(e)^4 + 480*B*a*b*c^2*tan(f*x)^3*tan(e)^4 + 240*A*b^2*c^2*tan(f*x)^3*tan(e)^4 - 300*C*b^2*c^2*tan(f*x)^3*tan(e)^4 + 480*B*a^2*c*d*tan(f*x)^3*tan(e)^4 + 960*A*a*b*c*d*tan(f*x)^3*tan(e)^4 - 1200*C*a*b*c*d*tan(f*x)^3*tan(e)^4 - 600*B*b^2*c*d*tan(f*x)^3*tan(e)^4 + 240*A*a^2*d^2*tan(f*x)^3*tan(e)^4 - 300*C*a^2*d^2*tan(f*x)^3*tan(e)^4 - 600*B*a*b*d^2*tan(f*x)^3*tan(e)^4 - 300*A*b^2*d^2*tan(f*x)^3*tan(e)^4 + 300*C*b^2*d^2*tan(f*x)^3*tan(e)^4 - 20*C*b^2*c^2*tan(f*x)^2*tan(e)^5 - 80*C*a*b*c*d*tan(f*x)^2*tan(e)^5 - 40*B*b^2*c*d*tan(f*x)^2*tan(e)^5 - 20*C*a^2*d^2*tan(f*x)^2*tan(e)^5 - 40*B*a*b*d^2*tan(f*x)^2*tan(e)^5 - 20*A*b^2*d^2*tan(f*x)^2*tan(e)^5 + 20*C*b^2*d^2*tan(f*x)^2*tan(e)^5 + 30*C*b^2*c*d*tan(f*x)^5*tan(e) + 30*C*a*b*d^2*tan(f*x)^5*tan(e) + 15*B*b^2*d^2*tan(f*x)^5*tan(e) - 600*A*a^2*c^2*f*x*tan(f*x)^2*tan(e)^2 + 600*C*a^2*c^2*f*x*tan(f*x)^2*tan(e)^2 + 1200*B*a*b*c^2*f*x*tan(f*x)^2*tan(e)^2 + 600*A*b^2*c^2*f*x*tan(f*x)^2*tan(e)^2 - 600*C*b^2*c^2*f*x*tan(f*x)^2*tan(e)^2 + 1200*B*a^2*c*d*f*x*tan(f*x)^2*tan(e)^2 + 2400*A*a*b*c*d*f*x*tan(f*x)^2*tan(e)^2 - 2400*C*a*b*c*d*f*x*tan(f*x)^2*tan(e)^2 - 1200*B*b^2*c*d*f*x*tan(f*x)^2*tan(e)^2 + 600*A*a^2*d^2*f*x*tan(f*x)^2*tan(e)^2 - 600*C*a^2*d^2*f*x*tan(f*x)^2*tan(e)^2 - 1200*B*a*b*d^2*f*x*tan(f*x)^2*tan(e)^2 - 600*A*b^2*d^2*f*x*tan(f*x)^2*tan(e)^2 + 600*C*b^2*d^2*f*x*tan(f*x)^2*tan(e)^2 - 180*C*a*b*c^2*tan(f*x)^4*tan(e)^2 - 90*B*b^2*c^2*tan(f*x)^4*tan(e)^2 - 180*C*a^2*c*d*tan(f*x)^4*tan(e)^2 - 360*B*a*b*c*d*tan(f*x)^4*tan(e)^2 - 180*A*b^2*c*d*tan(f*x)^4*tan(e)^2 + 300*C*b^2*c*d*tan(f*x)^4*tan(e)^2 - 90*B*a^2*d^2*tan(f*x)^4*tan(e)^2 - 180*A*a*b*d^2*tan(f*x)^4*tan(e)^2 + 300*C*a*b*d^2*tan(f*x)^4*tan(e)^2 + 150*B*b^2*d^2*tan(f*x)^4*tan(e)^2 + 240*C*a*b*c^2*tan(f*x)^3*tan(e)^3 + 120*B*b^2*c^2*tan(f*x)^3*tan(e)^3 + 240*C*a^2*c*d*tan(f*x)^3*tan(e)^3 + 480*B*a*b*c*d*tan(f*x)^3*tan(e)^3 + 240*A*b^2*c*d*tan(f*x)^3*tan(e)^3 - 360*C*b^2*c*d*tan(f*x)^3*tan(e)^3 + 120*B*a^2*d^2*tan(f*x)^3*tan(e)^3 + 240*A*a*b*d^2*tan(f*x)^3*tan(e)^3 - 360*C*a*b*d^2*tan(f*x)^3*tan(e)^3 - 180*B*b^2*d^2*tan(f*x)^3*tan(e)^3 - 180*C*a*b*c^2*tan(f*x)^2*tan(e)^4 - 90*B*b^2*c^2*tan(f*x)^2*tan(e)^4 - 180*C*a^2*c*d*tan(f*x)^2*tan(e)^4 - 360*B*a*b*c*d*tan(f*x)^2*tan(e)^4 - 180*A*b^2*c*d*tan(f*x)^2*tan(e)^4 + 300*C*b^2*c*d*tan(f*x)^2*tan(e)^4 - 90*B*a^2*d^2*tan(f*x)^2*tan(e)^4 - 180*A*a*b*d^2*tan(f*x)^2*tan(e)^4 + 300*C*a*b*d^2*tan(f*x)^2*tan(e)^4 + 150*B*b^2*d^2*tan(f*x)^2*tan(e)^4 + 30

*C*b^2*c*d*tan(f*x)*tan(e)^5 + 30*C*a*b*d^2*tan(f*x)*tan(e)^5 + 15*B*b^2*d^
2*tan(f*x)*tan(e)^5 - 12*C*b^2*d^2*tan(f*x)^5 + 40*C*b^2*c^2*tan(f*x)^4*tan
(e) + 160*C*a*b*c*d*tan(f*x)^4*tan(e) + 80*B*b^2*c*d*tan(f*x)^4*tan(e) + 40
*C*a^2*d^2*tan(f*x)^4*tan(e) + 80*B*a*b*d^2*tan(f*x)^4*tan(e) + 40*A*b^2*d^
2*tan(f*x)^4*tan(e) - 100*C*b^2*d^2*tan(f*x)^4*tan(e) + 300*B*a^2*c^2*log(4
*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f
*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 600*A*a*b*c^2*log(4*(tan(f*x)^
2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan
(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 600*C*a*b*c^2*log(4*(tan(f*x)^2*tan(e)^2
- 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))
*tan(f*x)^2*tan(e)^2 - 300*B*b^2*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x
)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2
*tan(e)^2 + 600*A*a^2*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) +
1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 -
600*C*a^2*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x
)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 1200*B*a*b
*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)
^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 600*A*b^2*c*d*log(4*
(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*
x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 600*C*b^2*c*d*log(4*(tan(f*x)^2
*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(
e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 300*B*a^2*d^2*log(4*(tan(f*x)^2*tan(e)^2 -
2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*
tan(f*x)^2*tan(e)^2 - 600*A*a*b*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)
*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*
tan(e)^2 + 600*C*a*b*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1
)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 +
300*B*b^2*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)
^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 360*C*a^2*c
^2*tan(f*x)^3*tan(e)^2 - 720*B*a*b*c^2*tan(f*x)^3*tan(e)^2 - 360*A*b^2*c^2*
tan(f*x)^3*tan(e)^2 + 480*C*b^2*c^2*tan(f*x)^3*tan(e)^2 - 720*B*a^2*c*d*tan
(f*x)^3*tan(e)^2 - 1440*A*a*b*c*d*tan(f*x)^3*tan(e)^2 + 1920*C*a*b*c*d*tan(
f*x)^3*tan(e)^2 + 960*B*b^2*c*d*tan(f*x)^3*tan(e)^2 - 360*A*a^2*d^2*tan(f*x
)^3*tan(e)^2 + 480*C*a^2*d^2*tan(f*x)^3*tan(e)^2 + 960*B*a*b*d^2*tan(f*x)^3
*tan(e)^2 + 480*A*b^2*d^2*tan(f*x)^3*tan(e)^2 - 600*C*b^2*d^2*tan(f*x)^3*ta
n(e)^2 - 360*C*a^2*c^2*tan(f*x)^2*tan(e)^3 - 720*B*a*b*c^2*tan(f*x)^2*tan(e
)^3 - 360*A*b^2*c^2*tan(f*x)^2*tan(e)^3 + 480*C*b^2*c^2*tan(f*x)^2*tan(e)^3
- 720*B*a^2*c*d*tan(f*x)^2*tan(e)^3 - 1440*A*a*b*c*d*tan(f*x)^2*tan(e)^3 +
1920*C*a*b*c*d*tan(f*x)^2*tan(e)^3 + 960*B*b^2*c*d*tan(f*x)^2*tan(e)^3 - 3
60*A*a^2*d^2*tan(f*x)^2*tan(e)^3 + 480*C*a^2*d^2*tan(f*x)^2*tan(e)^3 + 960*
B*a*b*d^2*tan(f*x)^2*tan(e)^3 + 480*A*b^2*d^2*tan(f*x)^2*tan(e)^3 - 600*C*b
^2*d^2*tan(f*x)^2*tan(e)^3 + 40*C*b^2*c^2*tan(f*x)*tan(e)^4 + 160*C*a*b*c*d
*tan(f*x)*tan(e)^4 + 80*B*b^2*c*d*tan(f*x)*tan(e)^4 + 40*C*a^2*d^2*tan(f*x)
*tan(e)^4 + 80*B*a*b*d^2*tan(f*x)*tan(e)^4 + 40*A*b^2*d^2*tan(f*x)*tan(e)^4
- 100*C*b^2*d^2*tan(f*x)*tan(e)^4 - 12*C*b^2*d^2*tan(e)^5 - 30*C*b^2*c*d*t

an(f*x)^4 - 30*C*a*b*d^2*tan(f*x)^4 - 15*B*b^2*d^2*tan(f*x)^4 + 300*A*a^2*c ^2*f*x*tan(f*x)*tan(e) - 300*C*a^2*c^2*f*x*tan(f*x)*tan(e) - 600*B*a*b*c^2* f*x*tan(f*x)*tan(e) - 300*A*b^2*c^2*f*x*tan(f*x)*tan(e) + 300*C*b^2*c^2*f*x *tan(f*x)*tan(e) - 600*B*a^2*c*d*f*x*tan(f*x)*tan(e) - 1200*A*a*b*c*d*f*x*t an(f*x)*tan(e) + 1200*C*a*b*c*d*f*x*tan(f*x)*tan(e) + 600*B*b^2*c*d*f*x*tan (f*x)*tan(e) - 300*A*a^2*d^2*f*x*tan(f*x)*tan(e) + 300*C*a^2*d^2*f*x*tan(f* x)*tan(e) + 600*B*a*b*d^2*f*x*tan(f*x)*tan(e) + 300*A*b^2*d^2*f*x*tan(f*x)* tan(e) - 300*C*b^2*d^2*f*x*tan(f*x)*tan(e) + 180*C*a*b*c^2*tan(f*x)^3*tan(e) + 90*B*b^2*c^2*tan(f*x)^3*tan(e) + 180*C*a^2*c*d*tan(f*x)^3*tan(e) + 360* B*a*b*c*d*tan(f*x)^3*tan(e) + 180*A*b^2*c*d*tan(f*x)^3*tan(e) - 300*C*b^2*c *d*tan(f*x)^3*tan(e) + 90*B*a^2*d^2*tan(f*x)^3*tan(e) + 180*A*a*b*d^2*tan(f *x)^3*tan(e) - 300*C*a*b*d^2*tan(f*x)^3*tan(e) - 150*B*b^2*d^2*tan(f*x)^3*t an(e) - 240*C*a*b*c^2*tan(f*x)^2*tan(e)^2 - 120*B*b^2*c^2*tan(f*x)^2*tan(e) ^2 - 240*C*a^2*c*d*tan(f*x)^2*tan(e)^2 - 480*B*a*b*c*d*tan(f*x)^2*tan(e)^2 - 240*A*b^2*c*d*tan(f*x)^2*tan(e)^2 + 360*C*b^2*c*d*tan(f*x)^2*tan(e)^2 - 1 20*B*a^2*d^2*tan(f*x)^2*tan(e)^2 - 240*A*a*b*d^2*tan(f*x)^2*tan(e)^2 + 360* C*a*b*d^2*tan(f*x)^2*tan(e)^2 + 180*B*b^2*d^2*tan(f*x)^2*tan(e)^2 + 180*C*a *b*c^2*tan(f*x)*tan(e)^3 + 90*B*b^2*c^2*tan(f*x)*tan(e)^3 + 180*C*a^2*c*d*t an(f*x)*tan(e)^3 + 360*B*a*b*c*d*tan(f*x)*tan(e)^3 + 180*A*b^2*c*d*tan(f*x) *tan(e)^3 - 300*C*b^2*c*d*tan(f*x)*tan(e)^3 + 90*B*a^2*d^2*tan(f*x)*tan(e)^ 3 + 180*A*a*b*d^2*tan(f*x)*tan(e)^3 - 300*C*a*b*d^2*tan(f*x)*tan(e)^3 - 150 *B*b^2*d^2*tan(f*x)*tan(e)^3 - 30*C*b^2*c*d*tan(e)^4 - 30*C*a*b*d^2*tan(e)^ 4 - 15*B*b^2*d^2*tan(e)^4 - 20*C*b^2*c^2*tan(f*x)^3 - 80*C*a*b*c*d*tan(f*x) ^3 - 40*B*b^2*c*d*tan(f*x)^3 - 20*C*a^2*d^2*tan(f*x)^3 - 40*B*a*b*d^2*tan(f *x)^3 - 20*A*b^2*d^2*tan(f*x)^3 + 20*C*b^2*d^2*tan(f*x)^3 - 150*B*a^2*c^2*l og(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + t an(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) - 300*A*a*b*c^2*log(4*(tan(f*x)^ 2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan (e)^2 + 1))*tan(f*x)*tan(e) + 300*C*a*b*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2* tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan (f*x)*tan(e) + 150*B*b^2*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) - 300*A*a^2*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x) ^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 300*C*a^2*c*d*l og(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + t an(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 600*B*a*b*c*d*log(4*(tan(f*x)^ 2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan (e)^2 + 1))*tan(f*x)*tan(e) + 300*A*b^2*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2* tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan (f*x)*tan(e) - 300*C*b^2*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 150*B*a^2*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x) ^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 300*A*a*b*d^2*l og(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + t an(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) - 300*C*a*b*d^2*log(4*(tan(f*x)^

2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan
(e)^2 + 1))*tan(f*x)*tan(e) - 150*B*b^2*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*
tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan
(f*x)*tan(e) + 240*C*a^2*c^2*tan(f*x)^2*tan(e) + 480*B*a*b*c^2*tan(f*x)^2*t
an(e) + 240*A*b^2*c^2*tan(f*x)^2*tan(e) - 300*C*b^2*c^2*tan(f*x)^2*tan(e) +
480*B*a^2*c*d*tan(f*x)^2*tan(e) + 960*A*a*b*c*d*tan(f*x)^2*tan(e) - 1200*C
*a*b*c*d*tan(f*x)^2*tan(e) - 600*B*b^2*c*d*tan(f*x)^2*tan(e) + 240*A*a^2*d^
2*tan(f*x)^2*tan(e) - 300*C*a^2*d^2*tan(f*x)^2*tan(e) - 600*B*a*b*d^2*tan(f
*x)^2*tan(e) - 300*A*b^2*d^2*tan(f*x)^2*tan(e) + 300*C*b^2*d^2*tan(f*x)^2*t
an(e) + 240*C*a^2*c^2*tan(f*x)*tan(e)^2 + 480*B*a*b*c^2*tan(f*x)*tan(e)^2 +
240*A*b^2*c^2*tan(f*x)*tan(e)^2 - 300*C*b^2*c^2*tan(f*x)*tan(e)^2 + 480*B*
a^2*c*d*tan(f*x)*tan(e)^2 + 960*A*a*b*c*d*tan(f*x)*tan(e)^2 - 1200*C*a*b*c*
d*tan(f*x)*tan(e)^2 - 600*B*b^2*c*d*tan(f*x)*tan(e)^2 + 240*A*a^2*d^2*tan(f
*x)*tan(e)^2 - 300*C*a^2*d^2*tan(f*x)*tan(e)^2 - 600*B*a*b*d^2*tan(f*x)*tan
(e)^2 - 300*A*b^2*d^2*tan(f*x)*tan(e)^2 + 300*C*b^2*d^2*tan(f*x)*tan(e)^2 -
20*C*b^2*c^2*tan(e)^3 - 80*C*a*b*c*d*tan(e)^3 - 40*B*b^2*c*d*tan(e)^3 - 20
*C*a^2*d^2*tan(e)^3 - 40*B*a*b*d^2*tan(e)^3 - 20*A*b^2*d^2*tan(e)^3 + 20*C*
b^2*d^2*tan(e)^3 - 60*A*a^2*c^2*f*x + 60*C*a^2*c^2*f*x + 120*B*a*b*c^2*f*x
+ 60*A*b^2*c^2*f*x - 60*C*b^2*c^2*f*x + 120*B*a^2*c*d*f*x + 240*A*a*b*c*d*f
*x - 240*C*a*b*c*d*f*x - 120*B*b^2*c*d*f*x + 60*A*a^2*d^2*f*x - 60*C*a^2*d^
2*f*x - 120*B*a*b*d^2*f*x - 60*A*b^2*d^2*f*x + 60*C*b^2*d^2*f*x - 60*C*a*b*
c^2*tan(f*x)^2 - 30*B*b^2*c^2*tan(f*x)^2 - 60*C*a^2*c*d*tan(f*x)^2 - 120*B*
a*b*c*d*tan(f*x)^2 - 60*A*b^2*c*d*tan(f*x)^2 + 60*C*b^2*c*d*tan(f*x)^2 - 30
*B*a^2*d^2*tan(f*x)^2 - 60*A*a*b*d^2*tan(f*x)^2 + 60*C*a*b*d^2*tan(f*x)^2 +
30*B*b^2*d^2*tan(f*x)^2 + 180*C*a*b*c^2*tan(f*x)*tan(e) + 90*B*b^2*c^2*tan
(f*x)*tan(e) + 180*C*a^2*c*d*tan(f*x)*tan(e) + 360*B*a*b*c*d*tan(f*x)*tan(e
) + 180*A*b^2*c*d*tan(f*x)*tan(e) - 330*C*b^2*c*d*tan(f*x)*tan(e) + 90*B*a^
2*d^2*tan(f*x)*tan(e) + 180*A*a*b*d^2*tan(f*x)*tan(e) - 330*C*a*b*d^2*tan(f
*x)*tan(e) - 165*B*b^2*d^2*tan(f*x)*tan(e) - 60*C*a*b*c^2*tan(e)^2 - 30*B*b
^2*c^2*tan(e)^2 - 60*C*a^2*c*d*tan(e)^2 - 120*B*a*b*c*d*tan(e)^2 - 60*A*b^2
*c*d*tan(e)^2 + 60*C*b^2*c*d*tan(e)^2 - 30*B*a^2*d^2*tan(e)^2 - 60*A*a*b*d^
2*tan(e)^2 + 60*C*a*b*d^2*tan(e)^2 + 30*B*b^2*d^2*tan(e)^2 + 30*B*a^2*c^2*l
og(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + t
an(f*x)^2 + tan(e)^2 + 1)) + 60*A*a*b*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*ta
n(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 60*
C*a*b*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*t
an(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 30*B*b^2*c^2*log(4*(tan(f*x)^2*tan(
e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2
+ 1)) + 60*A*a^2*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(t
an(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 60*C*a^2*c*d*log(4*(tan(
f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2
+ tan(e)^2 + 1)) - 120*B*a*b*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*ta
n(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 60*A*b^2*c*d
log(4(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 +
tan(f*x)^2 + tan(e)^2 + 1)) + 60*C*b^2*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*

```
tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 3
0*B*a^2*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2
*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 60*A*a*b*d^2*log(4*(tan(f*x)^2*ta
n(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^
2 + 1)) + 60*C*a*b*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/
(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) + 30*B*b^2*d^2*log(4*(ta
n(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^
2 + tan(e)^2 + 1)) - 60*C*a^2*c^2*tan(f*x) - 120*B*a*b*c^2*tan(f*x) - 60*A*
b^2*c^2*tan(f*x) + 60*C*b^2*c^2*tan(f*x) - 120*B*a^2*c*d*tan(f*x) - 240*A*a
*b*c*d*tan(f*x) + 240*C*a*b*c*d*tan(f*x) + 120*B*b^2*c*d*tan(f*x) - 60*A*a^
2*d^2*tan(f*x) + 60*C*a^2*d^2*tan(f*x) + 120*B*a*b*d^2*tan(f*x) + 60*A*b^2*
d^2*tan(f*x) - 60*C*b^2*d^2*tan(f*x) - 60*C*a^2*c^2*tan(e) - 120*B*a*b*c^2*
tan(e) - 60*A*b^2*c^2*tan(e) + 60*C*b^2*c^2*tan(e) - 120*B*a^2*c*d*tan(e) -
 240*A*a*b*c*d*tan(e) + 240*C*a*b*c*d*tan(e) + 120*B*b^2*c*d*tan(e) - 60*A*
a^2*d^2*tan(e) + 60*C*a^2*d^2*tan(e) + 120*B*a*b*d^2*tan(e) + 60*A*b^2*d^2*
tan(e) - 60*C*b^2*d^2*tan(e) - 60*C*a*b*c^2 - 30*B*b^2*c^2 - 60*C*a^2*c*d -
 120*B*a*b*c*d - 60*A*b^2*c*d + 90*C*b^2*c*d - 30*B*a^2*d^2 - 60*A*a*b*d^2
+ 90*C*a*b*d^2 + 45*B*b^2*d^2)/(f*tan(f*x)^5*tan(e)^5 - 5*f*tan(f*x)^4*tan(
e)^4 + 10*f*tan(f*x)^3*tan(e)^3 - 10*f*tan(f*x)^2*tan(e)^2 + 5*f*tan(f*x)*t
an(e) - f)
```

Mupad [B] (verification not implemented)

Time = 8.36 (sec) , antiderivative size = 561, normalized size of antiderivative = 1.27

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= x\left(A\,a^2\,c^2-A\,a^2\,d^2-A\,b^2\,c^2+A\,b^2\,d^2-C\,a^2\,c^2+C\,a^2\,d^2+C\,b^2\,c^2-C\,b^2\,d^2-2\,B\,a\,b\,c^2 + 2\,B\,a\,b\,d^2-2\,B\,a^2\,c\,d+2\,B\,b^2\,c\,d-4\,A\,a\,b\,c\,d+4\,C\,a\,b\,c\,d\right)$$
$$-\frac{\ln\left(\tan(e+f\,x)^2+1\right)\,\left(\frac{B\,a^2\,d^2}{2}-\frac{B\,a^2\,c^2}{2}+\frac{B\,b^2\,c^2}{2}-\frac{B\,b^2\,d^2}{2}-A\,a\,b\,c^2+A\,a\,b\,d^2-A\,a^2\,c\,d+C\,a\,b\,c^2+A\,b\right.}{f}$$
$$+\frac{\tan(e+f\,x)^2\,\left(\frac{B\,a^2\,d^2}{2}+\frac{B\,b^2\,c^2}{2}-\frac{b\,d\,(B\,b\,d+2\,C\,a\,d+2\,C\,b\,c)}{2}+A\,a\,b\,d^2+C\,a\,b\,c^2+A\,b^2\,c\,d+C\,a^2\,c\,d+2\,B\,a\,b\right.}{f}$$
$$+\frac{\tan(e+f\,x)^3\,\left(\frac{A\,b^2\,d^2}{3}+\frac{C\,a^2\,d^2}{3}+\frac{C\,b^2\,c^2}{3}-\frac{C\,b^2\,d^2}{3}+\frac{2\,B\,a\,b\,d^2}{3}+\frac{2\,B\,b^2\,c\,d}{3}+\frac{4\,C\,a\,b\,c\,d}{3}\right)}{f}$$
$$+\frac{\tan(e+f\,x)\,\left(A\,a^2\,d^2+A\,b^2\,c^2-A\,b^2\,d^2+C\,a^2\,c^2-C\,a^2\,d^2-C\,b^2\,c^2+C\,b^2\,d^2+2\,B\,a\,b\,c^2-2\,B\,a\,b\,d^2\right.}{f}$$
$$+\frac{b\,d\tan(e+f\,x)^4\,(B\,b\,d+2\,C\,a\,d+2\,C\,b\,c)}{4\,f}+\frac{C\,b^2\,d^2\tan(e+f\,x)^5}{5\,f}$$

```
[In] int((a + b*tan(e + f*x))^2*(c + d*tan(e + f*x))^2*(A + B*tan(e + f*x) + C*t
an(e + f*x)^2),x)
```

```
[Out] x*(A*a^2*c^2 - A*a^2*d^2 - A*b^2*c^2 + A*b^2*d^2 - C*a^2*c^2 + C*a^2*d^2 +
C*b^2*c^2 - C*b^2*d^2 - 2*B*a*b*c^2 + 2*B*a*b*d^2 - 2*B*a^2*c*d + 2*B*b^2*c
*d - 4*A*a*b*c*d + 4*C*a*b*c*d) - (log(tan(e + f*x)^2 + 1)*((B*a^2*d^2)/2 -
 (B*a^2*c^2)/2 + (B*b^2*c^2)/2 - (B*b^2*d^2)/2 - A*a*b*c^2 + A*a*b*d^2 - A*
a^2*c*d + C*a*b*c^2 + A*b^2*c*d - C*a*b*d^2 + C*a^2*c*d - C*b^2*c*d + 2*B*a
*b*c*d))/f + (tan(e + f*x)^2*((B*a^2*d^2)/2 + (B*b^2*c^2)/2 - (b*d*(B*b*d +
 2*C*a*d + 2*C*b*c))/2 + A*a*b*d^2 + C*a*b*c^2 + A*b^2*c*d + C*a^2*c*d + 2*
B*a*b*c*d))/f + (tan(e + f*x)^3*((A*b^2*d^2)/3 + (C*a^2*d^2)/3 + (C*b^2*c^2
)/3 - (C*b^2*d^2)/3 + (2*B*a*b*d^2)/3 + (2*B*b^2*c*d)/3 + (4*C*a*b*c*d)/3))
/f + (tan(e + f*x)*(A*a^2*d^2 + A*b^2*c^2 - A*b^2*d^2 + C*a^2*c^2 - C*a^2*d
^2 - C*b^2*c^2 + C*b^2*d^2 + 2*B*a*b*c^2 - 2*B*a*b*d^2 + 2*B*a^2*c*d - 2*B*
b^2*c*d + 4*A*a*b*c*d - 4*C*a*b*c*d))/f + (b*d*tan(e + f*x)^4*(B*b*d + 2*C*
a*d + 2*C*b*c))/(4*f) + (C*b^2*d^2*tan(e + f*x)^5)/(5*f)
```

3.59 $\int (a+b\tan(e+fx))(c+d\tan(e+fx))^2 (A+B\tan(e+$

Optimal result 519
Rubi [A] (verified) 520
Mathematica [C] (verified) 522
Maple [A] (verified) 523
Fricas [A] (verification not implemented) 523
Sympy [B] (verification not implemented) 524
Maxima [A] (verification not implemented) 524
Giac [B] (verification not implemented) 525
Mupad [B] (verification not implemented) 529

Optimal result

Integrand size = 43, antiderivative size = 266

$$\begin{aligned}
&\int (a+b\tan(e+fx))(c+d\tan(e+fx))^2 \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx \\
&= -\left(\left(a\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)+b\left(2c(A-C)d+B\left(c^2-d^2\right)\right)\right)x\right) \\
&\quad -\frac{(a(Bc^2-2cCd-Bd^2)-b(c^2C+2Bcd-Cd^2)+A(2acd+b(c^2-d^2)))\log(\cos(e+fx))}{f} \\
&\quad +\frac{d(Abc+aBc-bcC+aAd-bBd-aCd)\tan(e+fx)}{f} \\
&\quad +\frac{(Ab+aB-bC)(c+d\tan(e+fx))^2}{2f} \\
&\quad -\frac{(bcC-4bBd-4aCd)(c+d\tan(e+fx))^3}{12d^2f}+\frac{bC\tan(e+fx)(c+d\tan(e+fx))^3}{4df}
\end{aligned}$$

```
[Out] -(a*(c^2*C+2*B*c*d-C*d^2-A*(c^2-d^2))+b*(2*c*(A-C)*d+B*(c^2-d^2)))*x-(a*(B*
c^2-B*d^2-2*C*c*d)-b*(2*B*c*d+C*c^2-C*d^2)+A*(2*a*c*d+b*(c^2-d^2)))*ln(cos(
f*x+e))/f+d*(A*a*d+A*b*c+B*a*c-B*b*d-C*a*d-C*b*c)*tan(f*x+e)/f+1/2*(A*b+B*a
-C*b)*(c+d*tan(f*x+e))^2/f-1/12*(-4*B*b*d-4*C*a*d+C*b*c)*(c+d*tan(f*x+e))^3
/d^2/f+1/4*b*C*tan(f*x+e)*(c+d*tan(f*x+e))^3/d/f
```

Rubi [A] (verified)

Time = 0.51 (sec) , antiderivative size = 264, normalized size of antiderivative = 0.99, number of steps used = 5, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.116, Rules used = {3718, 3711, 3609, 3606, 3556}

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = -\frac{\log(\cos(e+fx))\left(2aAcd+aB(c^2-d^2)-2acCd+Ab(c^2-d^2)-b(2Bcd+c^2C-Cd^2)\right)}{f} - x\left(a\left(-A\left(c^2-d^2\right)+2Bcd+c^2C-Cd^2\right)+b\left(2cd(A-C)+B\left(c^2-d^2\right)\right)\right) + \frac{(aB+Ab-bC)(c+d\tan(e+fx))^2}{2f} + \frac{d\tan(e+fx)(aAd+aBc-aCd+Abc-bBd-bcC)}{f} - \frac{(-4aCd-4bBd+bcC)(c+d\tan(e+fx))^3}{12d^2f} + \frac{bC\tan(e+fx)(c+d\tan(e+fx))^3}{4df}$$

```
[In] Int[(a + b*Tan[e + f*x])*(c + d*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan
[e + f*x]^2),x]
```

```
[Out] -((a*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2)) + b*(2*c*(A - C)*d + B*(c^2
- d^2)))*x) - ((2*a*A*c*d - 2*a*c*C*d + A*b*(c^2 - d^2) + a*B*(c^2 - d^2) -
 b*(c^2*C + 2*B*c*d - C*d^2))*Log[Cos[e + f*x]])/f + (d*(A*b*c + a*B*c - b*
c*C + a*A*d - b*B*d - a*C*d)*Tan[e + f*x])/f + ((A*b + a*B - b*C)*(c + d*Ta
n[e + f*x])^2)/(2*f) - ((b*c*C - 4*b*B*d - 4*a*C*d)*(c + d*Tan[e + f*x])^3)
/(12*d^2*f) + (b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^3)/(4*d*f)
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3606

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
 f*x], x], x] + Simp[b*d*(Tan[e + f*x]/f), x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3609

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[d*((a + b*Tan[e + f*x])^m/(f*m)), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
```

```
0] && GtQ[m, 0]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rubi steps

$$\text{integral} = \frac{bC\tan(e+fx)(c+d\tan(e+fx))^3}{4df} - \frac{\int(c+d\tan(e+fx))^2\left(bcC-4aAd-4(Ab+aB-bC)d\tan(e+fx)+(bcC-4bBd-4aCd)\tan^2(e+\right.}{4d}$$

$$= -\frac{(bcC-4bBd-4aCd)(c+d\tan(e+fx))^3}{12d^2f} + \frac{bC\tan(e+fx)(c+d\tan(e+fx))^3}{4df} - \frac{\int(c+d\tan(e+fx))^2(4(bB-a(A-C))d-4(Ab+aB-bC)d\tan(e+fx))\,dx}{4d}$$

$$= \frac{(Ab+aB-bC)(c+d\tan(e+fx))^2}{2f} - \frac{(bcC-4bBd-4aCd)(c+d\tan(e+fx))^3}{12d^2f} + \frac{bC\tan(e+fx)(c+d\tan(e+fx))^3}{4df} - \frac{\int(c+d\tan(e+fx))(4d(bBc+b(A-C)d-a(Ac-cC-Bd))-4d(Abc+aBc-bcC+aA}{4d}$$

$$
\begin{aligned}
&= -\left(\left(a\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)+b\left(2c(A-C)d+B\left(c^2-d^2\right)\right)\right)x\right)\\
&\quad+\frac{d(Abc+aBc-bcC+aAd-bBd-aCd)\tan(e+fx)}{f}\\
&\quad+\frac{(Ab+aB-bC)(c+d\tan(e+fx))^2}{2f}\\
&\quad-\frac{(bcC-4bBd-4aCd)(c+d\tan(e+fx))^3}{12d^2f}\\
&\quad+\frac{bC\tan(e+fx)(c+d\tan(e+fx))^3}{4df}-\left(-2aAcd+2acCd-Ab\left(c^2-d^2\right)\right.\\
&\qquad\left.-aB\left(c^2-d^2\right)+b\left(c^2C+2Bcd-Cd^2\right)\right)\int\tan(e+fx)\,dx\\
&= -\left(\left(a\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)+b\left(2c(A-C)d+B\left(c^2-d^2\right)\right)\right)x\right)\\
&\quad-\frac{(2aAcd-2acCd+Ab(c^2-d^2)+aB(c^2-d^2)-b(c^2C+2Bcd-Cd^2))\log(\cos(e+fx))}{f}\\
&\quad+\frac{d(Abc+aBc-bcC+aAd-bBd-aCd)\tan(e+fx)}{f}\\
&\quad+\frac{(Ab+aB-bC)(c+d\tan(e+fx))^2}{2f}\\
&\quad-\frac{(bcC-4bBd-4aCd)(c+d\tan(e+fx))^3}{12d^2f}+\frac{bC\tan(e+fx)(c+d\tan(e+fx))^3}{4df}
\end{aligned}
$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 3.01 (sec) , antiderivative size = 241, normalized size of antiderivative = 0.91

$$
\int(a+b\tan(e+fx))(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx
$$

$$
=\frac{\frac{(-bcC+4bBd+4aCd)(c+d\tan(e+fx))^3}{d}+3bC\tan(e+fx)(c+d\tan(e+fx))^3+6(Abc+aBc-bcC-aAd+bBd}{}
$$

```
[In] Integrate[(a + b*Tan[e + f*x])*(c + d*Tan[e + f*x])^2*(A + B*Tan[e + f*x] +
 C*Tan[e + f*x]^2),x]
```

```
[Out] (((-(b*c*C) + 4*b*B*d + 4*a*C*d)*(c + d*Tan[e + f*x])^3)/d + 3*b*C*Tan[e +
f*x]*(c + d*Tan[e + f*x])^3 + 6*(A*b*c + a*B*c - b*c*C - a*A*d + b*B*d + a*
C*d)*(I*((c + I*d)^2*Log[I - Tan[e + f*x]] - (c - I*d)^2*Log[I + Tan[e + f*
x]]) - 2*d^2*Tan[e + f*x]) + 6*(A*b + a*B - b*C)*((I*c - d)^3*Log[I - Tan[e
 + f*x]] - (I*c + d)^3*Log[I + Tan[e + f*x]] + 6*c*d^2*Tan[e + f*x] + d^3*T
an[e + f*x]^2))/(12*d*f)
```

Maple [A] (verified)

Time = 0.15 (sec) , antiderivative size = 246, normalized size of antiderivative = 0.92

method	result
parts	$\frac{(2Aacd+Ab\,c^2+Ba\,c^2)\ln\left(1+\tan(fx+e)^2\right)}{2f}+\frac{(Bb\,d^2+Ca\,d^2+2Cbcd)\left(\frac{\tan(fx+e)^3}{3}-\tan(fx+e)+\arctan(\tan(fx+e))\right)}{f}$
norman	$(Aa\,c^2-Aa\,d^2-2Abcd-2Bacd-Bb\,c^2+Bb\,d^2-Ca\,c^2+Ca\,d^2+2Cbcd)\,x+\frac{(Aa\,d}{}$
derivativedivides	$\frac{Cb\,d^2\tan(fx+e)^4}{4}+\frac{Bb\,d^2\tan(fx+e)^3}{3}+\frac{Ca\,d^2\tan(fx+e)^3}{3}+\frac{2Cbcd\tan(fx+e)^3}{3}+\frac{Ab\,d^2\tan(fx+e)^2}{2}+\frac{Ba\,d^2\tan(fx+e)^2}{2}+Bbcd\,\mathrm{ta}$
default	$\frac{Cb\,d^2\tan(fx+e)^4}{4}+\frac{Bb\,d^2\tan(fx+e)^3}{3}+\frac{Ca\,d^2\tan(fx+e)^3}{3}+\frac{2Cbcd\tan(fx+e)^3}{3}+\frac{Ab\,d^2\tan(fx+e)^2}{2}+\frac{Ba\,d^2\tan(fx+e)^2}{2}+Bbcd\,\mathrm{ta}$
parallelrisch	$\frac{12A\ln\left(1+\tan(fx+e)^2\right)acd-12B\ln\left(1+\tan(fx+e)^2\right)bcd-12C\ln\left(1+\tan(fx+e)^2\right)acd+3Cb\,d^2\tan(fx+e)^4+4Bb\,d^2\,\mathrm{ta}}{}$
risch	Expression too large to display

```
[In] int((a+b*tan(f*x+e))*(c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x,m
ethod=_RETURNVERBOSE)
```

```
[Out] 1/2*(2*A*a*c*d+A*b*c^2+B*a*c^2)/f*ln(1+tan(f*x+e)^2)+(B*b*d^2+C*a*d^2+2*C*b
*c*d)/f*(1/3*tan(f*x+e)^3-tan(f*x+e)+arctan(tan(f*x+e)))+(A*b*d^2+B*a*d^2+2
*B*b*c*d+2*C*a*c*d+C*b*c^2)/f*(1/2*tan(f*x+e)^2-1/2*ln(1+tan(f*x+e)^2))+(A*
a*d^2+2*A*b*c*d+2*B*a*c*d+B*b*c^2+C*a*c^2)/f*(tan(f*x+e)-arctan(tan(f*x+e))
)+A*a*c^2*x+C*b*d^2/f*(1/4*tan(f*x+e)^4-1/2*tan(f*x+e)^2+1/2*ln(1+tan(f*x+e
)^2))
```

Fricas [A] (verification not implemented)

none

Time = 0.26 (sec) , antiderivative size = 259, normalized size of antiderivative = 0.97

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\frac{3\,Cbd^2\tan{(fx+e)}^4+4\,(2\,Cbcd+(Ca+Bb)d^2)\tan{(fx+e)}^3+12\,(((A-C)a-Bb)c^2-2\,(Ba+(A-}{}$$

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x, algorithm="fricas")
```

```
[Out] 1/12*(3*C*b*d^2*tan(f*x + e)^4 + 4*(2*C*b*c*d + (C*a + B*b)*d^2)*tan(f*x +
e)^3 + 12*(((A - C)*a - B*b)*c^2 - 2*(B*a + (A - C)*b)*c*d - ((A - C)*a - B
*b)*d^2)*f*x + 6*(C*b*c^2 + 2*(C*a + B*b)*c*d + (B*a + (A - C)*b)*d^2)*tan(
```

```
f*x + e)^2 - 6*((B*a + (A - C)*b)*c^2 + 2*((A - C)*a - B*b)*c*d - (B*a + (A
 - C)*b)*d^2)*log(1/(tan(f*x + e)^2 + 1)) + 12*((C*a + B*b)*c^2 + 2*(B*a +
(A - C)*b)*c*d + ((A - C)*a - B*b)*d^2)*tan(f*x + e))/f
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 617 vs. 2(246) = 492.

Time = 0.20 (sec) , antiderivative size = 617, normalized size of antiderivative = 2.32

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\begin{cases} Aac^2x+\frac{Aacd\log\left(\tan^2(e+fx)+1\right)}{f}-Aad^2x+\frac{Aad^2\tan(e+fx)}{f}+\frac{Abc^2\log\left(\tan^2(e+fx)+1\right)}{2f}-2Abcdx+\frac{2Abcd\tan(e+fx)}{f} \\ x(a+b\tan(e))(c+d\tan(e))^2\left(A+B\tan(e)+C\tan^2(e)\right) \end{cases}$$

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))**2*(A+B*tan(f*x+e)+C*tan(f*x+e)
**2),x)
```

```
[Out] Piecewise((A*a*c**2*x + A*a*c*d*log(tan(e + f*x)**2 + 1)/f - A*a*d**2*x + A
*a*d**2*tan(e + f*x)/f + A*b*c**2*log(tan(e + f*x)**2 + 1)/(2*f) - 2*A*b*c*
d*x + 2*A*b*c*d*tan(e + f*x)/f - A*b*d**2*log(tan(e + f*x)**2 + 1)/(2*f) +
A*b*d**2*tan(e + f*x)**2/(2*f) + B*a*c**2*log(tan(e + f*x)**2 + 1)/(2*f) -
2*B*a*c*d*x + 2*B*a*c*d*tan(e + f*x)/f - B*a*d**2*log(tan(e + f*x)**2 + 1)/
(2*f) + B*a*d**2*tan(e + f*x)**2/(2*f) - B*b*c**2*x + B*b*c**2*tan(e + f*x)
/f - B*b*c*d*log(tan(e + f*x)**2 + 1)/f + B*b*c*d*tan(e + f*x)**2/f + B*b*d
**2*x + B*b*d**2*tan(e + f*x)**3/(3*f) - B*b*d**2*tan(e + f*x)/f - C*a*c**2
*x + C*a*c**2*tan(e + f*x)/f - C*a*c*d*log(tan(e + f*x)**2 + 1)/f + C*a*c*d
*tan(e + f*x)**2/f + C*a*d**2*x + C*a*d**2*tan(e + f*x)**3/(3*f) - C*a*d**2
*tan(e + f*x)/f - C*b*c**2*log(tan(e + f*x)**2 + 1)/(2*f) + C*b*c**2*tan(e
+ f*x)**2/(2*f) + 2*C*b*c*d*x + 2*C*b*c*d*tan(e + f*x)**3/(3*f) - 2*C*b*c*d
*tan(e + f*x)/f + C*b*d**2*log(tan(e + f*x)**2 + 1)/(2*f) + C*b*d**2*tan(e
+ f*x)**4/(4*f) - C*b*d**2*tan(e + f*x)**2/(2*f), Ne(f, 0)), (x*(a + b*tan(
e))*(c + d*tan(e))**2*(A + B*tan(e) + C*tan(e)**2), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.32 (sec) , antiderivative size = 260, normalized size of antiderivative = 0.98

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\frac{3\,Cbd^2\tan(fx+e)^4+4\,(2\,Cbcd+(Ca+Bb)d^2)\tan(fx+e)^3+6\,(Cbc^2+2\,(Ca+Bb)cd+(Ba+(A-}{}$$

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x, algorithm="maxima")
```

```
[Out] 1/12*(3*C*b*d^2*tan(f*x + e)^4 + 4*(2*C*b*c*d + (C*a + B*b)*d^2)*tan(f*x +
e)^3 + 6*(C*b*c^2 + 2*(C*a + B*b)*c*d + (B*a + (A - C)*b)*d^2)*tan(f*x + e)
^2 + 12*(((A - C)*a - B*b)*c^2 - 2*(B*a + (A - C)*b)*c*d - ((A - C)*a - B*b
)*d^2)*(f*x + e) + 6*((B*a + (A - C)*b)*c^2 + 2*((A - C)*a - B*b)*c*d - (B*
a + (A - C)*b)*d^2)*log(tan(f*x + e)^2 + 1) + 12*((C*a + B*b)*c^2 + 2*(B*a
+ (A - C)*b)*c*d + ((A - C)*a - B*b)*d^2)*tan(f*x + e))/f
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 5631 vs. $2(260) = 520$.

Time = 4.09 (sec) , antiderivative size = 5631, normalized size of antiderivative = 21.17

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= \text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x, algorithm="giac")
```

```
[Out] 1/12*(12*A*a*c^2*f*x*tan(f*x)^4*tan(e)^4 - 12*C*a*c^2*f*x*tan(f*x)^4*tan(e)
^4 - 12*B*b*c^2*f*x*tan(f*x)^4*tan(e)^4 - 24*B*a*c*d*f*x*tan(f*x)^4*tan(e)^
4 - 24*A*b*c*d*f*x*tan(f*x)^4*tan(e)^4 + 24*C*b*c*d*f*x*tan(f*x)^4*tan(e)^4
 - 12*A*a*d^2*f*x*tan(f*x)^4*tan(e)^4 + 12*C*a*d^2*f*x*tan(f*x)^4*tan(e)^4
+ 12*B*b*d^2*f*x*tan(f*x)^4*tan(e)^4 - 6*B*a*c^2*log(4*(tan(f*x)^2*tan(e)^2
 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)
)*tan(f*x)^4*tan(e)^4 - 6*A*b*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*t
an(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*ta
n(e)^4 + 6*C*b*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan
(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 - 12*A*a
*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)
^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 12*C*a*c*d*log(4*(ta
n(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^
2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 12*B*b*c*d*log(4*(tan(f*x)^2*tan(e
)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 +
 1))*tan(f*x)^4*tan(e)^4 + 6*B*a*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x
)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4
*tan(e)^4 + 6*A*b*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(
tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 - 6*C
*b*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(
e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 - 48*A*a*c^2*f*x*tan
(f*x)^3*tan(e)^3 + 48*C*a*c^2*f*x*tan(f*x)^3*tan(e)^3 + 48*B*b*c^2*f*x*tan(
f*x)^3*tan(e)^3 + 96*B*a*c*d*f*x*tan(f*x)^3*tan(e)^3 + 96*A*b*c*d*f*x*tan(f
```

*x)^3*tan(e)^3 - 96*C*b*c*d*f*x*tan(f*x)^3*tan(e)^3 + 48*A*a*d^2*f*x*tan(f*
x)^3*tan(e)^3 - 48*C*a*d^2*f*x*tan(f*x)^3*tan(e)^3 - 48*B*b*d^2*f*x*tan(f*x
)^3*tan(e)^3 + 6*C*b*c^2*tan(f*x)^4*tan(e)^4 + 12*C*a*c*d*tan(f*x)^4*tan(e)
^4 + 12*B*b*c*d*tan(f*x)^4*tan(e)^4 + 6*B*a*d^2*tan(f*x)^4*tan(e)^4 + 6*A*b
*d^2*tan(f*x)^4*tan(e)^4 - 9*C*b*d^2*tan(f*x)^4*tan(e)^4 + 24*B*a*c^2*log(4
*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f
*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 24*A*b*c^2*log(4*(tan(f*x)^2*t
an(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)
^2 + 1))*tan(f*x)^3*tan(e)^3 - 24*C*b*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*ta
n(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f
*x)^3*tan(e)^3 + 48*A*a*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e)
+ 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3
- 48*C*a*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)
^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 48*B*b*c*d*
log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 +
tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 24*B*a*d^2*log(4*(tan(f*x
)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + t
an(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 24*A*b*d^2*log(4*(tan(f*x)^2*tan(e)^2 -
2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*
tan(f*x)^3*tan(e)^3 + 24*C*b*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*ta
n(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan
(e)^3 - 12*C*a*c^2*tan(f*x)^4*tan(e)^3 - 12*B*b*c^2*tan(f*x)^4*tan(e)^3 - 2
4*B*a*c*d*tan(f*x)^4*tan(e)^3 - 24*A*b*c*d*tan(f*x)^4*tan(e)^3 + 24*C*b*c*d
*tan(f*x)^4*tan(e)^3 - 12*A*a*d^2*tan(f*x)^4*tan(e)^3 + 12*C*a*d^2*tan(f*x)
^4*tan(e)^3 + 12*B*b*d^2*tan(f*x)^4*tan(e)^3 - 12*C*a*c^2*tan(f*x)^3*tan(e)
^4 - 12*B*b*c^2*tan(f*x)^3*tan(e)^4 - 24*B*a*c*d*tan(f*x)^3*tan(e)^4 - 24*A
*b*c*d*tan(f*x)^3*tan(e)^4 + 24*C*b*c*d*tan(f*x)^3*tan(e)^4 - 12*A*a*d^2*ta
n(f*x)^3*tan(e)^4 + 12*C*a*d^2*tan(f*x)^3*tan(e)^4 + 12*B*b*d^2*tan(f*x)^3*
tan(e)^4 + 72*A*a*c^2*f*x*tan(f*x)^2*tan(e)^2 - 72*C*a*c^2*f*x*tan(f*x)^2*t
an(e)^2 - 72*B*b*c^2*f*x*tan(f*x)^2*tan(e)^2 - 144*B*a*c*d*f*x*tan(f*x)^2*t
an(e)^2 - 144*A*b*c*d*f*x*tan(f*x)^2*tan(e)^2 + 144*C*b*c*d*f*x*tan(f*x)^2*
tan(e)^2 - 72*A*a*d^2*f*x*tan(f*x)^2*tan(e)^2 + 72*C*a*d^2*f*x*tan(f*x)^2*t
an(e)^2 + 72*B*b*d^2*f*x*tan(f*x)^2*tan(e)^2 + 6*C*b*c^2*tan(f*x)^4*tan(e)^
2 + 12*C*a*c*d*tan(f*x)^4*tan(e)^2 + 12*B*b*c*d*tan(f*x)^4*tan(e)^2 + 6*B*a
*d^2*tan(f*x)^4*tan(e)^2 + 6*A*b*d^2*tan(f*x)^4*tan(e)^2 - 6*C*b*d^2*tan(f*
x)^4*tan(e)^2 - 12*C*b*c^2*tan(f*x)^3*tan(e)^3 - 24*C*a*c*d*tan(f*x)^3*tan(
e)^3 - 24*B*b*c*d*tan(f*x)^3*tan(e)^3 - 12*B*a*d^2*tan(f*x)^3*tan(e)^3 - 12
*A*b*d^2*tan(f*x)^3*tan(e)^3 + 24*C*b*d^2*tan(f*x)^3*tan(e)^3 + 6*C*b*c^2*t
an(f*x)^2*tan(e)^4 + 12*C*a*c*d*tan(f*x)^2*tan(e)^4 + 12*B*b*c*d*tan(f*x)^2
*tan(e)^4 + 6*B*a*d^2*tan(f*x)^2*tan(e)^4 + 6*A*b*d^2*tan(f*x)^2*tan(e)^4 -
6*C*b*d^2*tan(f*x)^2*tan(e)^4 - 8*C*b*c*d*tan(f*x)^4*tan(e) - 4*C*a*d^2*ta
n(f*x)^4*tan(e) - 4*B*b*d^2*tan(f*x)^4*tan(e) - 36*B*a*c^2*log(4*(tan(f*x)^
2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan
(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 36*A*b*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2
*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*ta

n(f*x)^2*tan(e)^2 + 36*C*b*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 72*A*a*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 72*C*a*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 72*B*b*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 36*B*a*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 36*A*b*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 36*C*b*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 36*C*a*c^2*tan(f*x)^3*tan(e)^2 + 36*B*b*c^2*tan(f*x)^3*tan(e)^2 + 72*B*a*c*d*tan(f*x)^3*tan(e)^2 + 72*A*b*c*d*tan(f*x)^3*tan(e)^2 - 96*C*b*c*d*tan(f*x)^3*tan(e)^2 + 36*A*a*d^2*tan(f*x)^3*tan(e)^2 - 48*C*a*d^2*tan(f*x)^3*tan(e)^2 - 48*B*b*d^2*tan(f*x)^3*tan(e)^2 + 36*C*a*c^2*tan(f*x)^2*tan(e)^3 + 36*B*b*c^2*tan(f*x)^2*tan(e)^3 + 72*B*a*c*d*tan(f*x)^2*tan(e)^3 + 72*A*b*c*d*tan(f*x)^2*tan(e)^3 - 96*C*b*c*d*tan(f*x)^2*tan(e)^3 + 36*A*a*d^2*tan(f*x)^2*tan(e)^3 - 48*C*a*d^2*tan(f*x)^2*tan(e)^3 - 48*B*b*d^2*tan(f*x)^2*tan(e)^3 - 8*C*b*c*d*tan(f*x)*tan(e)^4 - 4*C*a*d^2*tan(f*x)*tan(e)^4 - 4*B*b*d^2*tan(f*x)*tan(e)^4 + 3*C*b*d^2*tan(f*x)^4 - 48*A*a*c^2*f*x*tan(f*x)*tan(e) + 48*C*a*c^2*f*x*tan(f*x)*tan(e) + 48*B*b*c^2*f*x*tan(f*x)*tan(e) + 96*B*a*c*d*f*x*tan(f*x)*tan(e) + 96*A*b*c*d*f*x*tan(f*x)*tan(e) - 96*C*b*c*d*f*x*tan(f*x)*tan(e) + 48*A*a*d^2*f*x*tan(f*x)*tan(e) - 48*C*a*d^2*f*x*tan(f*x)*tan(e) - 48*B*b*d^2*f*x*tan(f*x)*tan(e) - 12*C*b*c^2*tan(f*x)^3*tan(e) - 24*C*a*c*d*tan(f*x)^3*tan(e) - 24*B*b*c*d*tan(f*x)^3*tan(e) - 12*B*a*d^2*tan(f*x)^3*tan(e) - 12*A*b*d^2*tan(f*x)^3*tan(e) + 24*C*b*d^2*tan(f*x)^3*tan(e) + 12*C*b*c^2*tan(f*x)^2*tan(e)^2 + 24*C*a*c*d*tan(f*x)^2*tan(e)^2 + 24*B*b*c*d*tan(f*x)^2*tan(e)^2 + 12*B*a*d^2*tan(f*x)^2*tan(e)^2 + 12*A*b*d^2*tan(f*x)^2*tan(e)^2 - 12*C*b*d^2*tan(f*x)^2*tan(e)^2 - 12*C*b*c^2*tan(f*x)*tan(e)^3 - 24*C*a*c*d*tan(f*x)*tan(e)^3 - 24*B*b*c*d*tan(f*x)*tan(e)^3 - 12*B*a*d^2*tan(f*x)*tan(e)^3 - 12*A*b*d^2*tan(f*x)*tan(e)^3 + 24*C*b*d^2*tan(f*x)*tan(e)^3 + 3*C*b*d^2*tan(e)^4 + 8*C*b*c*d*tan(f*x)^3 + 4*C*a*d^2*tan(f*x)^3 + 4*B*b*d^2*tan(f*x)^3 + 24*B*a*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 24*A*b*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) - 24*C*b*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 48*A*a*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) - 48*C*a*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) - 48*B*b*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) - 24*B*a*d^2*lo g(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) - 24*A*b*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 24*C*b*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) - 36*C*a*c^2*tan(f*x)^2*tan(e) - 36*B*b*c^2*tan(f*x)^2*tan(e) - 72*B*a*c*d*tan(f*x)^2*tan(e) - 72*A*b*c*d*tan(f*x)^2*tan(e) + 96*C*b*c*d*tan(f*x)^2*tan(e) - 36*A*a*d^2*tan(f*x)^2*tan(e) + 48*C*a*d^2*tan(f*x)^2*tan(e) + 48*B*b*d^2*tan(f*x)^2*tan(e) - 36*C*a*c^2*tan(f*x)*tan(e)^2 - 36*B*b*c^2*tan(f*x)*tan(e)^2 - 72*B*a*c*d*tan(f*x)*tan(e)^2 - 72*A*b*c*d*tan(f*x)*tan(e)^2 + 96*C*b*c*d*tan(f*x)*tan(e)^2 - 36*A*a*d^2*tan(f*x)*tan(e)^2 + 48*C*a*d^2*tan(f*x)*tan(e)^2 + 48*B*b*d^2*tan(f*x)*tan(e)^2 + 8*C*b*c*d*tan(e)^3 + 4*C*a*d^2*tan(e)^3 + 4*B*b*d^2*tan(e)^3 + 12*A*a*c^2*f*x - 12*C*a*c^2*f*x - 12*B*b*c^2*f*x - 24*B*a*c*d*f*x - 24*A*b*c*d*f*x + 24*C*b*c*d*f*x - 12*A*a*d^2*f*x + 12*C*a*d^2*f*x + 12*B*b*d^2*f*x + 6*C*b*c^2*tan(f*x)^2 + 12*C*a*c*d*tan(f*x)^2 + 12*B*b*c*d*tan(f*x)^2 + 6*B*a*d^2*tan(f*x)^2 + 6*A*b*d^2*tan(f*x)^2 - 6*C*b*d^2*tan(f*x)^2 - 12*C*b*c^2*tan(f*x)*tan(e) - 24*C*a*c*d*tan(f*x)*tan(e) - 24*B*b*c*d*tan(f*x)*tan(e) - 12*B*a*d^2*tan(f*x)*tan(e) - 12*A*b*d^2*tan(f*x)*tan(e) + 24*C*b*d^2*tan(f*x)*tan(e) + 6*C*b*c^2*tan(e)^2 + 12*C*a*c*d*tan(e)^2 + 12*B*b*c*d*tan(e)^2 + 6*B*a*d^2*tan(e)^2 + 6*A*b*d^2*tan(e)^2 - 6*C*b*d^2*tan(e)^2 - 6*B*a*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 6*A*b*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) + 6*C*b*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 12*A*a*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) + 12*C*a*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) + 12*B*b*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) + 6*B*a*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) + 6*A*b*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 6*C*b*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) + 12*C*a*c^2*tan(f*x) + 12*B*b*c^2*tan(f*x) + 24*B*a*c*d*tan(f*x) + 24*A*b*c*d*tan(f*x) - 24*C*b*c*d*tan(f*x) + 12*A*a*d^2*tan(f*x) - 12*C*a*d^2*tan(f*x) - 12*B*b*d^2*tan(f*x) + 12*C*a*c^2*tan(e) + 12*B*b*c^2*tan(e) + 24*B*a*c*d*tan(e) + 24*A*b*c*d*tan(e) - 24*C*b*c*d*tan(e) + 12*A*a*d^2*tan(e) - 12*C*a*d^2*tan(e) - 12*B*b*d^2*tan(e) + 6*C*b*c^2 + 12*C*a*c*d + 12*B*b*c*d + 6*B*a*d^2 + 6*A*b*d^2 - 9*C*b*d^2)/(f*tan(f*x)^4*tan(e)^4 - 4*f*tan(f*x)^3*tan(e)^3 + 6*f*tan(f*x)^2*tan(e)^2 - 4*f*tan(f*x)*tan(e) + f)

Mupad [B] (verification not implemented)

Time = 8.42 (sec) , antiderivative size = 300, normalized size of antiderivative = 1.13

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))^2 \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\frac{\tan(e+f\,x)^2\,\left(\frac{A\,b\,d^2}{2}+\frac{B\,a\,d^2}{2}+\frac{C\,b\,c^2}{2}-\frac{C\,b\,d^2}{2}+B\,b\,c\,d+C\,a\,c\,d\right)}{f}$$
$$-x\left(A\,a\,d^2-A\,a\,c^2+B\,b\,c^2+C\,a\,c^2-B\,b\,d^2-C\,a\,d^2+2\,A\,b\,c\,d+2\,B\,a\,c\,d-2\,C\,b\,c\,d\right)$$
$$-\frac{\ln\left(\tan(e+f\,x)^2+1\right)\,\left(\frac{A\,b\,d^2}{2}-\frac{B\,a\,c^2}{2}-\frac{A\,b\,c^2}{2}+\frac{B\,a\,d^2}{2}+\frac{C\,b\,c^2}{2}-\frac{C\,b\,d^2}{2}-A\,a\,c\,d+B\,b\,c\,d+C\,a\,c\,d\right)}{f}$$
$$+\frac{\tan(e+f\,x)\,\left(A\,a\,d^2+B\,b\,c^2+C\,a\,c^2-B\,b\,d^2-C\,a\,d^2+2\,A\,b\,c\,d+2\,B\,a\,c\,d-2\,C\,b\,c\,d\right)}{f}$$
$$+\frac{\tan(e+f\,x)^3\,\left(\frac{B\,b\,d^2}{3}+\frac{C\,a\,d^2}{3}+\frac{2\,C\,b\,c\,d}{3}\right)}{f}+\frac{C\,b\,d^2\,\tan(e+f\,x)^4}{4\,f}$$

```
[In] int((a + b*tan(e + f*x))*(c + d*tan(e + f*x))^2*(A + B*tan(e + f*x) + C*tan
(e + f*x)^2),x)
```

```
[Out] (tan(e + f*x)^2*((A*b*d^2)/2 + (B*a*d^2)/2 + (C*b*c^2)/2 - (C*b*d^2)/2 + B*
b*c*d + C*a*c*d))/f - x*(A*a*d^2 - A*a*c^2 + B*b*c^2 + C*a*c^2 - B*b*d^2 -
C*a*d^2 + 2*A*b*c*d + 2*B*a*c*d - 2*C*b*c*d) - (log(tan(e + f*x)^2 + 1)*((A
*b*d^2)/2 - (B*a*c^2)/2 - (A*b*c^2)/2 + (B*a*d^2)/2 + (C*b*c^2)/2 - (C*b*d^
2)/2 - A*a*c*d + B*b*c*d + C*a*c*d))/f + (tan(e + f*x)*(A*a*d^2 + B*b*c^2 +
 C*a*c^2 - B*b*d^2 - C*a*d^2 + 2*A*b*c*d + 2*B*a*c*d - 2*C*b*c*d))/f + (tan
(e + f*x)^3*((B*b*d^2)/3 + (C*a*d^2)/3 + (2*C*b*c*d)/3))/f + (C*b*d^2*tan(e
 + f*x)^4)/(4*f)
```

3.60 $\int (c+d\tan(e+fx))^2 \left(A+B\tan(e+fx)+C\tan^2(e+$

Optimal result 530
Rubi [A] (verified) 530
Mathematica [C] (verified) 532
Maple [A] (verified) 532
Fricas [A] (verification not implemented) 533
Sympy [B] (verification not implemented) 533
Maxima [A] (verification not implemented) 534
Giac [B] (verification not implemented) 534
Mupad [B] (verification not implemented) 536

Optimal result

Integrand size = 33, antiderivative size = 131

$$\int (c+d\tan(e+fx))^2 \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= -\left(\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)x\right)-\frac{(2c(A-C)d+B(c^2-d^2))\log(\cos(e+fx))}{f}$$
$$+\frac{d(Bc+(A-C)d)\tan(e+fx)}{f}+\frac{B(c+d\tan(e+fx))^2}{2f}+\frac{C(c+d\tan(e+fx))^3}{3df}$$

```
[Out] -(c^2*C+2*B*c*d-C*d^2-A*(c^2-d^2))*x-(2*c*(A-C)*d+B*(c^2-d^2))*ln(cos(f*x+e
))/f+d*(B*c+(A-C)*d)*tan(f*x+e)/f+1/2*B*(c+d*tan(f*x+e))^2/f+1/3*C*(c+d*tan
(f*x+e))^3/d/f
```

Rubi [A] (verified)

Time = 0.19 (sec) , antiderivative size = 131, normalized size of antiderivative = 1.00, number of steps used = 4, number of rules used = 4, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.121, Rules used = {3711, 3609, 3606, 3556}

$$\int (c+d\tan(e+fx))^2 \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= -\frac{(2cd(A-C)+B(c^2-d^2))\log(\cos(e+fx))}{f}-x\left(-A\left(c^2-d^2\right)+2Bcd+c^2C-Cd^2\right)$$
$$+\frac{d\tan(e+fx)(d(A-C)+Bc)}{f}+\frac{B(c+d\tan(e+fx))^2}{2f}+\frac{C(c+d\tan(e+fx))^3}{3df}$$

```
[In] Int[(c + d*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] -((c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2))*x) - ((2*c*(A - C)*d + B*(c^2 -
 d^2))*Log[Cos[e + f*x]])/f + (d*(B*c + (A - C)*d)*Tan[e + f*x])/f + (B*(c
+ d*Tan[e + f*x])^2)/(2*f) + (C*(c + d*Tan[e + f*x])^3)/(3*d*f)
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3606

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
 f*x], x], x] + Simp[b*d*(Tan[e + f*x]/f), x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3609

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[d*((a + b*Tan[e + f*x])^m/(f*m)), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rubi steps

$$\begin{aligned}\text{integral} &= \frac{C(c+d\tan(e+fx))^3}{3df}+\int(A-C+B\tan(e+fx))(c+d\tan(e+fx))^2\,dx\\&=\frac{B(c+d\tan(e+fx))^2}{2f}+\frac{C(c+d\tan(e+fx))^3}{3df}\\&\quad+\int(c+d\tan(e+fx))(Ac-cC-Bd+(Bc+(A-C)d)\tan(e+fx))\,dx\\&=-\left(\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)x\right)+\frac{d(Bc+(A-C)d)\tan(e+fx)}{f}\\&\quad+\frac{B(c+d\tan(e+fx))^2}{2f}+\frac{C(c+d\tan(e+fx))^3}{3df}\\&\quad+\left(2c(A-C)d+B\left(c^2-d^2\right)\right)\int\tan(e+fx)\,dx\end{aligned}$$

$$= -\left(\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)x\right)-\frac{\left(2c(A-C)d+B\left(c^2-d^2\right)\right)\log(\cos(e+fx))}{f}$$
$$+\frac{d(Bc+(A-C)d)\tan(e+fx)}{f}+\frac{B(c+d\tan(e+fx))^2}{2f}+\frac{C(c+d\tan(e+fx))^3}{3df}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 1.13 (sec) , antiderivative size = 176, normalized size of antiderivative = 1.34

$$\int (c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= \frac{2C(c+d\tan(e+fx))^3+3(Bc+(-A+C)d)\left(i\left((c+id)^2\log(i-\tan(e+fx))-(c-id)^2\log(i+\tan(e+\right.\right.}{}$$

```
[In] Integrate[(c + d*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] (2*C*(c + d*Tan[e + f*x])^3 + 3*(B*c + (-A + C)*d)*(I*((c + I*d)^2*Log[I -
Tan[e + f*x]] - (c - I*d)^2*Log[I + Tan[e + f*x]]) - 2*d^2*Tan[e + f*x]) +
3*B*((I*c - d)^3*Log[I - Tan[e + f*x]] - (I*c + d)^3*Log[I + Tan[e + f*x]]
+ 6*c*d^2*Tan[e + f*x] + d^3*Tan[e + f*x]^2))/(6*d*f)
```

Maple [A] (verified)

Time = 0.08 (sec) , antiderivative size = 141, normalized size of antiderivative = 1.08

method	result
norman	$\left(A\,c^2-A\,d^2-2Bcd-c^2C+C\,d^2\right)x+\frac{\left(A\,d^2+2Bcd+c^2C-C\,d^2\right)\tan(fx+e)}{f}+\frac{C\,d^2\tan(fx+e)^3}{3f}+\frac{d(}{}$
parts	$A\,c^2x+\frac{\left(2Acd+B\,c^2\right)\ln\left(1+\tan(fx+e)^2\right)}{2f}+\frac{\left(B\,d^2+2Ccd\right)\left(\frac{\tan(fx+e)^2}{2}-\frac{\ln\left(1+\tan(fx+e)^2\right)}{2}\right)}{f}+\frac{\left(A\,d^2+2Bcd+c^2C\right.}{}$
derivativedivides	$\frac{\frac{C\,d^2\tan(fx+e)^3}{3}+\frac{B\,d^2\tan(fx+e)^2}{2}+Ccd\tan(fx+e)^2+\tan(fx+e)A\,d^2+2\tan(fx+e)Bcd+\tan(fx+e)c^2C-\tan(fx+e)C\,d^2}{f}$
default	$\frac{\frac{C\,d^2\tan(fx+e)^3}{3}+\frac{B\,d^2\tan(fx+e)^2}{2}+Ccd\tan(fx+e)^2+\tan(fx+e)A\,d^2+2\tan(fx+e)Bcd+\tan(fx+e)c^2C-\tan(fx+e)C\,d^2}{f}$
parallelrisch	$\frac{2C\,d^2\tan(fx+e)^3+6A\,c^2fx-6A\,d^2fx-12Bcdfx+3B\,d^2\tan(fx+e)^2-6C\,c^2fx+6C\,d^2fx+6Ccd\tan(fx+e)^2+6A\ln\left(1+\right.}{}$
risch	$-\frac{4iCcde}{f}+\frac{4iAcde}{f}+2iAcdx-\frac{2iB\,d^2e}{f}+A\,c^2x-A\,d^2x-2Bcdx-C\,c^2x+C\,d^2x+\frac{2i(-6iC}{}$

```
[In] int((c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x,method=_RETURNVERB
OSE)
```

```
[Out] (A*c^2-A*d^2-2*B*c*d-C*c^2+C*d^2)*x+(A*d^2+2*B*c*d+C*c^2-C*d^2)/f*tan(f*x+e
)+1/3*C*d^2/f*tan(f*x+e)^3+1/2*d*(B*d+2*C*c)/f*tan(f*x+e)^2+1/2*(2*A*c*d+B*
c^2-B*d^2-2*C*c*d)/f*ln(1+tan(f*x+e)^2)
```

Fricas [A] (verification not implemented)

none

Time = 0.27 (sec) , antiderivative size = 134, normalized size of antiderivative = 1.02

$$\int (c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\frac{2\,Cd^2\tan\left(fx+e\right)^3+6\left((A-C)c^2-2\,Bcd-(A-C)d^2\right)fx+3\left(2\,Ccd+Bd^2\right)\tan\left(fx+e\right)^2-3\left(Bc^2+\right.}{6\,f}$$

```
[In] integrate((c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorithm="
fricas")
```

```
[Out] 1/6*(2*C*d^2*tan(f*x + e)^3 + 6*((A - C)*c^2 - 2*B*c*d - (A - C)*d^2)*f*x +
 3*(2*C*c*d + B*d^2)*tan(f*x + e)^2 - 3*(B*c^2 + 2*(A - C)*c*d - B*d^2)*log
(1/(tan(f*x + e)^2 + 1)) + 6*(C*c^2 + 2*B*c*d + (A - C)*d^2)*tan(f*x + e))/
f
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 241 vs. 2(107) = 214.

Time = 0.13 (sec) , antiderivative size = 241, normalized size of antiderivative = 1.84

$$\int (c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\begin{cases} Ac^2x+\frac{Acd\log\left(\tan^2\left(e+fx\right)+1\right)}{f}-Ad^2x+\frac{Ad^2\tan\left(e+fx\right)}{f}+\frac{Bc^2\log\left(\tan^2\left(e+fx\right)+1\right)}{2f}-2Bcdx+\frac{2Bcd\tan\left(e+fx\right)}{f}-\frac{Bd^2}{} \\ x(c+d\tan\left(e\right))^2\left(A+B\tan\left(e\right)+C\tan^2\left(e\right)\right) \end{cases}$$

```
[In] integrate((c+d*tan(f*x+e))**2*(A+B*tan(f*x+e)+C*tan(f*x+e)**2),x)
```

```
[Out] Piecewise((A*c**2*x + A*c*d*log(tan(e + f*x)**2 + 1)/f - A*d**2*x + A*d**2*
tan(e + f*x)/f + B*c**2*log(tan(e + f*x)**2 + 1)/(2*f) - 2*B*c*d*x + 2*B*c*
d*tan(e + f*x)/f - B*d**2*log(tan(e + f*x)**2 + 1)/(2*f) + B*d**2*tan(e + f
*x)**2/(2*f) - C*c**2*x + C*c**2*tan(e + f*x)/f - C*c*d*log(tan(e + f*x)**2
 + 1)/f + C*c*d*tan(e + f*x)**2/f + C*d**2*x + C*d**2*tan(e + f*x)**3/(3*f)
 - C*d**2*tan(e + f*x)/f, Ne(f, 0)), (x*(c + d*tan(e))**2*(A + B*tan(e) + C
*tan(e)**2), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.29 (sec) , antiderivative size = 135, normalized size of antiderivative = 1.03

$$\int (c+d\tan(e+fx))^2 \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \frac{2\,Cd^2\tan\left(fx+e\right)^3+3\left(2\,Ccd+Bd^2\right)\tan\left(fx+e\right)^2+6\left((A-C)c^2-2\,Bcd-(A-C)d^2\right)(fx+e)+3\,(B}{6\,f}$$

```
[In] integrate((c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorithm="
maxima")
```

```
[Out] 1/6*(2*C*d^2*tan(f*x + e)^3 + 3*(2*C*c*d + B*d^2)*tan(f*x + e)^2 + 6*((A -
C)*c^2 - 2*B*c*d - (A - C)*d^2)*(f*x + e) + 3*(B*c^2 + 2*(A - C)*c*d - B*d^
2)*log(tan(f*x + e)^2 + 1) + 6*(C*c^2 + 2*B*c*d + (A - C)*d^2)*tan(f*x + e)
)/f
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1825 vs. $2(127) = 254$.

Time = 1.36 (sec) , antiderivative size = 1825, normalized size of antiderivative = 13.93

$$\int (c+d\tan(e+fx))^2 \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \text{Too large to display}$$

```
[In] integrate((c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorithm="
giac")
```

```
[Out] 1/6*(6*A*c^2*f*x*tan(f*x)^3*tan(e)^3 - 6*C*c^2*f*x*tan(f*x)^3*tan(e)^3 - 12
*B*c*d*f*x*tan(f*x)^3*tan(e)^3 - 6*A*d^2*f*x*tan(f*x)^3*tan(e)^3 + 6*C*d^2*
f*x*tan(f*x)^3*tan(e)^3 - 3*B*c^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*t
an(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*ta
n(e)^3 - 6*A*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f
*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 6*C*c*d*
log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 +
tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 3*B*d^2*log(4*(tan(f*x)^2
*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(
e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 18*A*c^2*f*x*tan(f*x)^2*tan(e)^2 + 18*C*c^
2*f*x*tan(f*x)^2*tan(e)^2 + 36*B*c*d*f*x*tan(f*x)^2*tan(e)^2 + 18*A*d^2*f*x
*tan(f*x)^2*tan(e)^2 - 18*C*d^2*f*x*tan(f*x)^2*tan(e)^2 + 6*C*c*d*tan(f*x)^
3*tan(e)^3 + 3*B*d^2*tan(f*x)^3*tan(e)^3 + 9*B*c^2*log(4*(tan(f*x)^2*tan(e)
^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 +
1))*tan(f*x)^2*tan(e)^2 + 18*A*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*
```

```
tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*t
an(e)^2 - 18*C*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan
(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 9*B*d^
2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2
 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 6*C*c^2*tan(f*x)^3*tan(
e)^2 - 12*B*c*d*tan(f*x)^3*tan(e)^2 - 6*A*d^2*tan(f*x)^3*tan(e)^2 + 6*C*d^2
*tan(f*x)^3*tan(e)^2 - 6*C*c^2*tan(f*x)^2*tan(e)^3 - 12*B*c*d*tan(f*x)^2*ta
n(e)^3 - 6*A*d^2*tan(f*x)^2*tan(e)^3 + 6*C*d^2*tan(f*x)^2*tan(e)^3 + 18*A*c
^2*f*x*tan(f*x)*tan(e) - 18*C*c^2*f*x*tan(f*x)*tan(e) - 36*B*c*d*f*x*tan(f*
x)*tan(e) - 18*A*d^2*f*x*tan(f*x)*tan(e) + 18*C*d^2*f*x*tan(f*x)*tan(e) + 6
*C*c*d*tan(f*x)^3*tan(e) + 3*B*d^2*tan(f*x)^3*tan(e) - 6*C*c*d*tan(f*x)^2*t
an(e)^2 - 3*B*d^2*tan(f*x)^2*tan(e)^2 + 6*C*c*d*tan(f*x)*tan(e)^3 + 3*B*d^2
*tan(f*x)*tan(e)^3 - 2*C*d^2*tan(f*x)^3 - 9*B*c^2*log(4*(tan(f*x)^2*tan(e)^
2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1
))*tan(f*x)*tan(e) - 18*A*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e
) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) +
 18*C*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*t
an(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 9*B*d^2*log(4*(tan(
f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2
 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 12*C*c^2*tan(f*x)^2*tan(e) + 24*B*c*d*ta
n(f*x)^2*tan(e) + 12*A*d^2*tan(f*x)^2*tan(e) - 18*C*d^2*tan(f*x)^2*tan(e) +
 12*C*c^2*tan(f*x)*tan(e)^2 + 24*B*c*d*tan(f*x)*tan(e)^2 + 12*A*d^2*tan(f*x
)*tan(e)^2 - 18*C*d^2*tan(f*x)*tan(e)^2 - 2*C*d^2*tan(e)^3 - 6*A*c^2*f*x +
6*C*c^2*f*x + 12*B*c*d*f*x + 6*A*d^2*f*x - 6*C*d^2*f*x - 6*C*c*d*tan(f*x)^2
 - 3*B*d^2*tan(f*x)^2 + 6*C*c*d*tan(f*x)*tan(e) + 3*B*d^2*tan(f*x)*tan(e) -
 6*C*c*d*tan(e)^2 - 3*B*d^2*tan(e)^2 + 3*B*c^2*log(4*(tan(f*x)^2*tan(e)^2 -
 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))
 + 6*A*c*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*t
an(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 6*C*c*d*log(4*(tan(f*x)^2*tan(e)^2
 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))
 - 3*B*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*
tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 6*C*c^2*tan(f*x) - 12*B*c*d*tan(f*
x) - 6*A*d^2*tan(f*x) + 6*C*d^2*tan(f*x) - 6*C*c^2*tan(e) - 12*B*c*d*tan(e)
 - 6*A*d^2*tan(e) + 6*C*d^2*tan(e) - 6*C*c*d - 3*B*d^2)/(f*tan(f*x)^3*tan(e
)^3 - 3*f*tan(f*x)^2*tan(e)^2 + 3*f*tan(f*x)*tan(e) - f)
```

Mupad [B] (verification not implemented)

Time = 8.38 (sec) , antiderivative size = 141, normalized size of antiderivative = 1.08

$$\int (c+d\tan(e+fx))^2 \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\frac{\tan(e+f\,x)^2\left(\frac{B\,d^2}{2}+C\,c\,d\right)}{f}-x\left(A\,d^2-A\,c^2+C\,c^2-C\,d^2+2\,B\,c\,d\right)$$
$$+\frac{\tan(e+f\,x)\,\left(A\,d^2+C\,c^2-C\,d^2+2\,B\,c\,d\right)}{f}$$
$$+\frac{\ln\left(\tan(e+f\,x)^2+1\right)\,\left(\frac{B\,c^2}{2}-\frac{B\,d^2}{2}+A\,c\,d-C\,c\,d\right)}{f}+\frac{C\,d^2\tan(e+f\,x)^3}{3\,f}$$

```
[In] int((c + d*tan(e + f*x))^2*(A + B*tan(e + f*x) + C*tan(e + f*x)^2),x)
```

```
[Out] (tan(e + f*x)^2*((B*d^2)/2 + C*c*d))/f - x*(A*d^2 - A*c^2 + C*c^2 - C*d^2 +
 2*B*c*d) + (tan(e + f*x)*(A*d^2 + C*c^2 - C*d^2 + 2*B*c*d))/f + (log(tan(e
 + f*x)^2 + 1)*((B*c^2)/2 - (B*d^2)/2 + A*c*d - C*c*d))/f + (C*d^2*tan(e +
f*x)^3)/(3*f)
```

3.61 $\int \frac{(c+d\tan(e+fx))^2(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx$

Optimal result 537
Rubi [A] (verified) 538
Mathematica [C] (verified) 540
Maple [A] (verified) 541
Fricas [A] (verification not implemented) 541
Sympy [C] (verification not implemented) 542
Maxima [A] (verification not implemented) 544
Giac [A] (verification not implemented) 545
Mupad [B] (verification not implemented) 546

Optimal result

Integrand size = 45, antiderivative size = 254

$$
\begin{aligned}
&\int \frac{(c+d\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx \\
&= -\frac{(a(c^2C+2Bcd-Cd^2-A(c^2-d^2))-b(2c(A-C)d+B(c^2-d^2)))\,x}{a^2+b^2} \\
&\quad -\frac{(a(Bc^2-2cCd-Bd^2)+b(c^2C+2Bcd-Cd^2)+A(2acd-b(c^2-d^2)))\log(\cos(e+fx))}{(a^2+b^2)\,f} \\
&\quad +\frac{(Ab^2-a(bB-aC))\,(bc-ad)^2\log(a+b\tan(e+fx))}{b^3\,(a^2+b^2)\,f} \\
&\quad +\frac{d(bcC+bBd-aCd)\tan(e+fx)}{b^2f}+\frac{C(c+d\tan(e+fx))^2}{2bf}
\end{aligned}
$$

```
[Out] -(a*(c^2*C+2*B*c*d-C*d^2-A*(c^2-d^2))-b*(2*c*(A-C)*d+B*(c^2-d^2)))*x/(a^2+b
^2)-(a*(B*c^2-B*d^2-2*C*c*d)+b*(2*B*c*d+C*c^2-C*d^2)+A*(2*a*c*d-b*(c^2-d^2)
))*ln(cos(f*x+e))/(a^2+b^2)/f+(A*b^2-a*(B*b-C*a))*(-a*d+b*c)^2*ln(a+b*tan(f
*x+e))/b^3/(a^2+b^2)/f+d*(B*b*d-C*a*d+C*b*c)*tan(f*x+e)/b^2/f+1/2*C*(c+d*ta
n(f*x+e))^2/b/f
```

Rubi [A] (verified)

Time = 0.90 (sec) , antiderivative size = 252, normalized size of antiderivative = 0.99, number of steps used = 6, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.133, Rules used = {3728, 3718, 3707, 3698, 31, 3556}

$$\int \frac{(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{a+b\tan(e+fx)}\,dx = -\frac{\log(\cos(e+fx))\left(2aAcd+aB(c^2-d^2)-2acCd-Ab(c^2-d^2)+b(2Bcd+c^2C-Cd^2)\right)}{f\left(a^2+b^2\right)} - \frac{x(a(-A(c^2-d^2)+2Bcd+c^2C-Cd^2)-b(2cd(A-C)+B(c^2-d^2)))}{a^2+b^2} + \frac{(bc-ad)^2\left(Ab^2-a(bB-aC)\right)\log(a+b\tan(e+fx))}{b^3f\left(a^2+b^2\right)} + \frac{d\tan(e+fx)(-aCd+bBd+bcC)}{b^2f} + \frac{C(c+d\tan(e+fx))^2}{2bf}$$

```
[In] Int[((c + d*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a + b
*Tan[e + f*x]),x]
```

```
[Out] -(((a*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - b*(2*c*(A - C)*d + B*(c^2
 - d^2)))*x)/(a^2 + b^2)) - ((2*a*A*c*d - 2*a*c*C*d - A*b*(c^2 - d^2) + a*B
*(c^2 - d^2) + b*(c^2*C + 2*B*c*d - C*d^2))*Log[Cos[e + f*x]])/((a^2 + b^2)
*f) + ((A*b^2 - a*(b*B - a*C))*(b*c - a*d)^2*Log[a + b*Tan[e + f*x]])/(b^3*
(a^2 + b^2)*f) + (d*(b*c*C + b*B*d - a*C*d)*Tan[e + f*x])/(b^2*f) + (C*(c +
 d*Tan[e + f*x])^2)/(2*b*f)
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3698

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3707

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[(a*A + b*B - a
```

```
*C)*(x/(a^2 + b^2)), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\text{integral} = \frac{C(c+d\tan(e+fx))^2}{2bf} + \frac{\int \frac{(c+d\tan(e+fx))\left(2(Abc-aCd)+2b(Bc+(A-C)d)\tan(e+fx)+2(bcC+bBd-aCd)\tan^2(e+fx)\right)}{a+b\tan(e+fx)}\,dx}{2b}$$

$$= \frac{d(bcC+bBd-aCd)\tan(e+fx)}{b^2f} + \frac{C(c+d\tan(e+fx))^2}{2bf} - \frac{\int \frac{-2\left(Ab^2c^2-ad(2bcC+bBd-aCd)\right)-2b^2\left(2c(A-C)d+B\left(c^2-d^2\right)\right)\tan(e+fx)-2\left(a^2Cd^2-abd(2cC+Bd)+b^2\left(c^2C+2Bcd+(A-C)\right.\right.}{a+b\tan(e+fx)}}{2b^2}$$

$$= -\frac{(a(c^2C + 2Bcd - Cd^2 - A(c^2 - d^2)) - b(2c(A - C)d + B(c^2 - d^2)))\,x}{a^2 + b^2} + \frac{d(bcC + bBd - aCd)\tan(e + fx)}{b^2 f} + \frac{C(c + d\tan(e + fx))^2}{2bf} + \frac{((Ab^2 - a(bB - aC))\,(bc - ad)^2)\int \frac{1+\tan^2(e+fx)}{a+b\tan(e+fx)}\,dx}{b^2\,(a^2 + b^2)} + \frac{(2aAcd - 2acCd - Ab(c^2 - d^2) + aB(c^2 - d^2) + b(c^2C + 2Bcd - Cd^2))\int \tan(e + fx)\,dx}{a^2 + b^2}$$

$$= -\frac{(a(c^2C + 2Bcd - Cd^2 - A(c^2 - d^2)) - b(2c(A - C)d + B(c^2 - d^2)))\,x}{a^2 + b^2} - \frac{(2aAcd - 2acCd - Ab(c^2 - d^2) + aB(c^2 - d^2) + b(c^2C + 2Bcd - Cd^2))\log(\cos(e + fx))}{(a^2 + b^2)\,f} + \frac{d(bcC + bBd - aCd)\tan(e + fx)}{b^2 f} + \frac{C(c + d\tan(e + fx))^2}{2bf} + \frac{((Ab^2 - a(bB - aC))\,(bc - ad)^2)\,\mathrm{Subst}\left(\int \frac{1}{a+x}\,dx, x, b\tan(e + fx)\right)}{b^3\,(a^2 + b^2)\,f}$$

$$= -\frac{(a(c^2C + 2Bcd - Cd^2 - A(c^2 - d^2)) - b(2c(A - C)d + B(c^2 - d^2)))\,x}{a^2 + b^2} - \frac{(2aAcd - 2acCd - Ab(c^2 - d^2) + aB(c^2 - d^2) + b(c^2C + 2Bcd - Cd^2))\log(\cos(e + fx))}{(a^2 + b^2)\,f} + \frac{(Ab^2 - a(bB - aC))\,(bc - ad)^2\log(a + b\tan(e + fx))}{b^3\,(a^2 + b^2)\,f} + \frac{d(bcC + bBd - aCd)\tan(e + fx)}{b^2 f} + \frac{C(c + d\tan(e + fx))^2}{2bf}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 3.21 (sec) , antiderivative size = 190, normalized size of antiderivative = 0.75

$$\int \frac{(c + d\tan(e + fx))^2\,(A + B\tan(e + fx) + C\tan^2(e + fx))}{a + b\tan(e + fx)}\,dx = \frac{\frac{b(-iA+B+iC)(c+id)^2\log(i-\tan(e+fx))}{a+ib} + \frac{b(iA+B-iC)(c-id)^2\log(i+\tan(e+fx))}{a-ib} + \frac{2(Ab^2+a(-bB+aC))(bc-ad)^2\log(a+b\tan(e+fx))}{b^2(a^2+b^2)} +}{2bf}$$

```
[In] Integrate[((c + d*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(a + b*Tan[e + f*x]),x]
[Out] ((b*((-I)*A + B + I*C)*(c + I*d)^2*Log[I - Tan[e + f*x]])/(a + I*b) + (b*(I
*A + B - I*C)*(c - I*d)^2*Log[I + Tan[e + f*x]])/(a - I*b) + (2*(A*b^2 + a*
(-(b*B) + a*C))*(b*c - a*d)^2*Log[a + b*Tan[e + f*x]])/(b^2*(a^2 + b^2)) +
(2*d*(b*c*C + b*B*d - a*C*d)*Tan[e + f*x])/b + C*(c + d*Tan[e + f*x])^2)/(2
*b*f)
```

Maple [A] (verified)

Time = 0.18 (sec) , antiderivative size = 317, normalized size of antiderivative = 1.25

method	result
derivativedivides	$\frac{\frac{d\left(\frac{\tan(fx+e)^2 Cbd}{2}+\tan(fx+e)bdB-\tan(fx+e)Cad+2\tan(fx+e)Cbc\right)}{b^2}+\frac{\left(A a^2 d^2 b^2-2Aa b^3 cd+A b^4 c^2-B a^3 d^2 b+2B a^2 cd b^2-Ba\right.}{b^3\left(a^2+\right.}}{}$
default	$\frac{\frac{d\left(\frac{\tan(fx+e)^2 Cbd}{2}+\tan(fx+e)bdB-\tan(fx+e)Cad+2\tan(fx+e)Cbc\right)}{b^2}+\frac{\left(A a^2 d^2 b^2-2Aa b^3 cd+A b^4 c^2-B a^3 d^2 b+2B a^2 cd b^2-Ba\right.}{b^3\left(a^2+\right.}}{}$
norman	$\frac{(Aa c^2-Aa d^2+2Abcd-2Bacd+Bb c^2-Bb d^2-Ca c^2+Ca d^2-2Cbcd)x}{a^2+b^2}+\frac{d(bdB-Cad+2Cbc)\tan(fx+e)}{b^2 f}+\frac{C d^2 \tan}{}$
parallelrisch	$\frac{2A\ln(a+b\tan(fx+e))a^2b^2d^2+B\ln\left(1+\tan(fx+e)^2\right)a b^3c^2-B\ln\left(1+\tan(fx+e)^2\right)a b^3d^2+2B\ln\left(1+\tan(fx+e)^2\right)b^4cd}{}$
risch	Expression too large to display

```
[In] int((c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e)),x,m
ethod=_RETURNVERBOSE)
```

```
[Out] 1/f*(d/b^2*(1/2*tan(f*x+e)^2*C*b*d+tan(f*x+e)*b*d*B-tan(f*x+e)*C*a*d+2*tan(
f*x+e)*C*b*c)+1/b^3*(A*a^2*b^2*d^2-2*A*a*b^3*c*d+A*b^4*c^2-B*a^3*b*d^2+2*B*
a^2*b^2*c*d-B*a*b^3*c^2+C*a^4*d^2-2*C*a^3*b*c*d+C*a^2*b^2*c^2)/(a^2+b^2)*ln
(a+b*tan(f*x+e))+1/(a^2+b^2)*(1/2*(2*A*a*c*d-A*b*c^2+A*b*d^2+B*a*c^2-B*a*d^
2+2*B*b*c*d-2*C*a*c*d+C*b*c^2-C*b*d^2)*ln(1+tan(f*x+e)^2)+(A*a*c^2-A*a*d^2+
2*A*b*c*d-2*B*a*c*d+B*b*c^2-B*b*d^2-C*a*c^2+C*a*d^2-2*C*b*c*d)*arctan(tan(f
*x+e))))
```

Fricas [A] (verification not implemented)

none

Time = 0.49 (sec) , antiderivative size = 397, normalized size of antiderivative = 1.56

$$\int \frac{(c+d\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx$$
$$=\frac{(Ca^2b^2+Cb^4)d^2\tan{(fx+e)}^2+2\left(\left(\left((A-C)ab^3+Bb^4\right)c^2-2\left(Bab^3-(A-C)b^4\right)cd-\left(\left(A-C\right)ab^3+B\right.\right.\right.}{}$$

```
[In] integrate((c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e
)),x, algorithm="fricas")
```

```
[Out] 1/2*((C*a^2*b^2 + C*b^4)*d^2*tan(f*x + e)^2 + 2*(((A - C)*a*b^3 + B*b^4)*c^
2 - 2*(B*a*b^3 - (A - C)*b^4)*c*d - ((A - C)*a*b^3 + B*b^4)*d^2)*f*x + ((C*
a^2*b^2 - B*a*b^3 + A*b^4)*c^2 - 2*(C*a^3*b - B*a^2*b^2 + A*a*b^3)*c*d + (C
```

```
*a^4 - B*a^3*b + A*a^2*b^2)*d^2)*log((b^2*tan(f*x + e)^2 + 2*a*b*tan(f*x +
e) + a^2)/(tan(f*x + e)^2 + 1)) - ((C*a^2*b^2 + C*b^4)*c^2 - 2*(C*a^3*b - B
*a^2*b^2 + C*a*b^3 - B*b^4)*c*d + (C*a^4 - B*a^3*b + A*a^2*b^2 - B*a*b^3 +
(A - C)*b^4)*d^2)*log(1/(tan(f*x + e)^2 + 1)) + 2*(2*(C*a^2*b^2 + C*b^4)*c*
d - (C*a^3*b - B*a^2*b^2 + C*a*b^3 - B*b^4)*d^2)*tan(f*x + e))/((a^2*b^3 +
b^5)*f)
```

Sympy [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 2.17 (sec) , antiderivative size = 4444, normalized size of antiderivative = 17.50

$$\int \frac{(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{a+b\tan(e+fx)}\,dx = \text{Too large to display}$$

```
[In] integrate((c+d*tan(f*x+e))**2*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x
+e)),x)
```

```
[Out] Piecewise((zoo*x*(c + d*tan(e))**2*(A + B*tan(e) + C*tan(e)**2)/tan(e), Eq(
a, 0) & Eq(b, 0) & Eq(f, 0)), ((A*c**2*x + A*c*d*log(tan(e + f*x)**2 + 1)/f
 - A*d**2*x + A*d**2*tan(e + f*x)/f + B*c**2*log(tan(e + f*x)**2 + 1)/(2*f)
 - 2*B*c*d*x + 2*B*c*d*tan(e + f*x)/f - B*d**2*log(tan(e + f*x)**2 + 1)/(2*
f) + B*d**2*tan(e + f*x)**2/(2*f) - C*c**2*x + C*c**2*tan(e + f*x)/f - C*c*
d*log(tan(e + f*x)**2 + 1)/f + C*c*d*tan(e + f*x)**2/f + C*d**2*x + C*d**2*
tan(e + f*x)**3/(3*f) - C*d**2*tan(e + f*x)/f)/a, Eq(b, 0)), (I*A*c**2*f*x*
tan(e + f*x)/(2*b*f*tan(e + f*x) - 2*I*b*f) + A*c**2*f*x/(2*b*f*tan(e + f*x
) - 2*I*b*f) + I*A*c**2/(2*b*f*tan(e + f*x) - 2*I*b*f) + 2*A*c*d*f*x*tan(e
+ f*x)/(2*b*f*tan(e + f*x) - 2*I*b*f) - 2*I*A*c*d*f*x/(2*b*f*tan(e + f*x) -
 2*I*b*f) - 2*A*c*d/(2*b*f*tan(e + f*x) - 2*I*b*f) + I*A*d**2*f*x*tan(e + f
*x)/(2*b*f*tan(e + f*x) - 2*I*b*f) + A*d**2*f*x/(2*b*f*tan(e + f*x) - 2*I*b
*f) + A*d**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*b*f*tan(e + f*x) - 2*
I*b*f) - I*A*d**2*log(tan(e + f*x)**2 + 1)/(2*b*f*tan(e + f*x) - 2*I*b*f) -
 I*A*d**2/(2*b*f*tan(e + f*x) - 2*I*b*f) + B*c**2*f*x*tan(e + f*x)/(2*b*f*t
an(e + f*x) - 2*I*b*f) - I*B*c**2*f*x/(2*b*f*tan(e + f*x) - 2*I*b*f) - B*c*
*2/(2*b*f*tan(e + f*x) - 2*I*b*f) + 2*I*B*c*d*f*x*tan(e + f*x)/(2*b*f*tan(e
 + f*x) - 2*I*b*f) + 2*B*c*d*f*x/(2*b*f*tan(e + f*x) - 2*I*b*f) + 2*B*c*d*l
og(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*b*f*tan(e + f*x) - 2*I*b*f) - 2*I*B
*c*d*log(tan(e + f*x)**2 + 1)/(2*b*f*tan(e + f*x) - 2*I*b*f) - 2*I*B*c*d/(2
*b*f*tan(e + f*x) - 2*I*b*f) - 3*B*d**2*f*x*tan(e + f*x)/(2*b*f*tan(e + f*x
) - 2*I*b*f) + 3*I*B*d**2*f*x/(2*b*f*tan(e + f*x) - 2*I*b*f) + I*B*d**2*log
(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*b*f*tan(e + f*x) - 2*I*b*f) + B*d**2*
log(tan(e + f*x)**2 + 1)/(2*b*f*tan(e + f*x) - 2*I*b*f) + 2*B*d**2*tan(e +
f*x)**2/(2*b*f*tan(e + f*x) - 2*I*b*f) + 3*B*d**2/(2*b*f*tan(e + f*x) - 2*I
*b*f) + I*C*c**2*f*x*tan(e + f*x)/(2*b*f*tan(e + f*x) - 2*I*b*f) + C*c**2*f
*x/(2*b*f*tan(e + f*x) - 2*I*b*f) + C*c**2*log(tan(e + f*x)**2 + 1)*tan(e +
```

f*x)/(2*b*f*tan(e + f*x) - 2*I*b*f) - I*C*c**2*log(tan(e + f*x)**2 + 1)/(2
*b*f*tan(e + f*x) - 2*I*b*f) - I*C*c**2/(2*b*f*tan(e + f*x) - 2*I*b*f) - 6*
C*c*d*f*x*tan(e + f*x)/(2*b*f*tan(e + f*x) - 2*I*b*f) + 6*I*C*c*d*f*x/(2*b*
f*tan(e + f*x) - 2*I*b*f) + 2*I*C*c*d*log(tan(e + f*x)**2 + 1)*tan(e + f*x)
/(2*b*f*tan(e + f*x) - 2*I*b*f) + 2*C*c*d*log(tan(e + f*x)**2 + 1)/(2*b*f*t
an(e + f*x) - 2*I*b*f) + 4*C*c*d*tan(e + f*x)**2/(2*b*f*tan(e + f*x) - 2*I*
b*f) + 6*C*c*d/(2*b*f*tan(e + f*x) - 2*I*b*f) - 3*I*C*d**2*f*x*tan(e + f*x)
/(2*b*f*tan(e + f*x) - 2*I*b*f) - 3*C*d**2*f*x/(2*b*f*tan(e + f*x) - 2*I*b*
f) - 2*C*d**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*b*f*tan(e + f*x) - 2
*I*b*f) + 2*I*C*d**2*log(tan(e + f*x)**2 + 1)/(2*b*f*tan(e + f*x) - 2*I*b*f
) + C*d**2*tan(e + f*x)**3/(2*b*f*tan(e + f*x) - 2*I*b*f) + I*C*d**2*tan(e
+ f*x)**2/(2*b*f*tan(e + f*x) - 2*I*b*f) + 3*I*C*d**2/(2*b*f*tan(e + f*x) -
2*I*b*f), Eq(a, -I*b)), (-I*A*c**2*f*x*tan(e + f*x)/(2*b*f*tan(e + f*x) +
2*I*b*f) + A*c**2*f*x/(2*b*f*tan(e + f*x) + 2*I*b*f) - I*A*c**2/(2*b*f*tan(
e + f*x) + 2*I*b*f) + 2*A*c*d*f*x*tan(e + f*x)/(2*b*f*tan(e + f*x) + 2*I*b*
f) + 2*I*A*c*d*f*x/(2*b*f*tan(e + f*x) + 2*I*b*f) - 2*A*c*d/(2*b*f*tan(e +
f*x) + 2*I*b*f) - I*A*d**2*f*x*tan(e + f*x)/(2*b*f*tan(e + f*x) + 2*I*b*f)
+ A*d**2*f*x/(2*b*f*tan(e + f*x) + 2*I*b*f) + A*d**2*log(tan(e + f*x)**2 +
1)*tan(e + f*x)/(2*b*f*tan(e + f*x) + 2*I*b*f) + I*A*d**2*log(tan(e + f*x)*
*2 + 1)/(2*b*f*tan(e + f*x) + 2*I*b*f) + I*A*d**2/(2*b*f*tan(e + f*x) + 2*I
*b*f) + B*c**2*f*x*tan(e + f*x)/(2*b*f*tan(e + f*x) + 2*I*b*f) + I*B*c**2*f
*x/(2*b*f*tan(e + f*x) + 2*I*b*f) - B*c**2/(2*b*f*tan(e + f*x) + 2*I*b*f) -
2*I*B*c*d*f*x*tan(e + f*x)/(2*b*f*tan(e + f*x) + 2*I*b*f) + 2*B*c*d*f*x/(2
*b*f*tan(e + f*x) + 2*I*b*f) + 2*B*c*d*log(tan(e + f*x)**2 + 1)*tan(e + f*x
)/(2*b*f*tan(e + f*x) + 2*I*b*f) + 2*I*B*c*d*log(tan(e + f*x)**2 + 1)/(2*b*
f*tan(e + f*x) + 2*I*b*f) + 2*I*B*c*d/(2*b*f*tan(e + f*x) + 2*I*b*f) - 3*B*
d**2*f*x*tan(e + f*x)/(2*b*f*tan(e + f*x) + 2*I*b*f) - 3*I*B*d**2*f*x/(2*b*
f*tan(e + f*x) + 2*I*b*f) - I*B*d**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/
(2*b*f*tan(e + f*x) + 2*I*b*f) + B*d**2*log(tan(e + f*x)**2 + 1)/(2*b*f*tan
(e + f*x) + 2*I*b*f) + 2*B*d**2*tan(e + f*x)**2/(2*b*f*tan(e + f*x) + 2*I*b
*f) + 3*B*d**2/(2*b*f*tan(e + f*x) + 2*I*b*f) - I*C*c**2*f*x*tan(e + f*x)/(
2*b*f*tan(e + f*x) + 2*I*b*f) + C*c**2*f*x/(2*b*f*tan(e + f*x) + 2*I*b*f) +
C*c**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*b*f*tan(e + f*x) + 2*I*b*f
) + I*C*c**2*log(tan(e + f*x)**2 + 1)/(2*b*f*tan(e + f*x) + 2*I*b*f) + I*C*
c**2/(2*b*f*tan(e + f*x) + 2*I*b*f) - 6*C*c*d*f*x*tan(e + f*x)/(2*b*f*tan(e
+ f*x) + 2*I*b*f) - 6*I*C*c*d*f*x/(2*b*f*tan(e + f*x) + 2*I*b*f) - 2*I*C*c
*d*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*b*f*tan(e + f*x) + 2*I*b*f) + 2
*C*c*d*log(tan(e + f*x)**2 + 1)/(2*b*f*tan(e + f*x) + 2*I*b*f) + 4*C*c*d*ta
n(e + f*x)**2/(2*b*f*tan(e + f*x) + 2*I*b*f) + 6*C*c*d/(2*b*f*tan(e + f*x)
+ 2*I*b*f) + 3*I*C*d**2*f*x*tan(e + f*x)/(2*b*f*tan(e + f*x) + 2*I*b*f) - 3
*C*d**2*f*x/(2*b*f*tan(e + f*x) + 2*I*b*f) - 2*C*d**2*log(tan(e + f*x)**2 +
1)*tan(e + f*x)/(2*b*f*tan(e + f*x) + 2*I*b*f) - 2*I*C*d**2*log(tan(e + f*
x)**2 + 1)/(2*b*f*tan(e + f*x) + 2*I*b*f) + C*d**2*tan(e + f*x)**3/(2*b*f*t
an(e + f*x) + 2*I*b*f) - I*C*d**2*tan(e + f*x)**2/(2*b*f*tan(e + f*x) + 2*I
*b*f) - 3*I*C*d**2/(2*b*f*tan(e + f*x) + 2*I*b*f), Eq(a, I*b)), (x*(c + d*t

```
an(e))**2*(A + B*tan(e) + C*tan(e)**2)/(a + b*tan(e)), Eq(f, 0)), (2*A*a**2
*b**2*d**2*log(a/b + tan(e + f*x))/(2*a**2*b**3*f + 2*b**5*f) + 2*A*a*b**3*
c**2*f*x/(2*a**2*b**3*f + 2*b**5*f) - 4*A*a*b**3*c*d*log(a/b + tan(e + f*x)
)/(2*a**2*b**3*f + 2*b**5*f) + 2*A*a*b**3*c*d*log(tan(e + f*x)**2 + 1)/(2*a
**2*b**3*f + 2*b**5*f) - 2*A*a*b**3*d**2*f*x/(2*a**2*b**3*f + 2*b**5*f) + 2
*A*b**4*c**2*log(a/b + tan(e + f*x))/(2*a**2*b**3*f + 2*b**5*f) - A*b**4*c*
*2*log(tan(e + f*x)**2 + 1)/(2*a**2*b**3*f + 2*b**5*f) + 4*A*b**4*c*d*f*x/(
2*a**2*b**3*f + 2*b**5*f) + A*b**4*d**2*log(tan(e + f*x)**2 + 1)/(2*a**2*b*
*3*f + 2*b**5*f) - 2*B*a**3*b*d**2*log(a/b + tan(e + f*x))/(2*a**2*b**3*f +
 2*b**5*f) + 4*B*a**2*b**2*c*d*log(a/b + tan(e + f*x))/(2*a**2*b**3*f + 2*b
**5*f) + 2*B*a**2*b**2*d**2*tan(e + f*x)/(2*a**2*b**3*f + 2*b**5*f) - 2*B*a
*b**3*c**2*log(a/b + tan(e + f*x))/(2*a**2*b**3*f + 2*b**5*f) + B*a*b**3*c*
*2*log(tan(e + f*x)**2 + 1)/(2*a**2*b**3*f + 2*b**5*f) - 4*B*a*b**3*c*d*f*x
/(2*a**2*b**3*f + 2*b**5*f) - B*a*b**3*d**2*log(tan(e + f*x)**2 + 1)/(2*a**
2*b**3*f + 2*b**5*f) + 2*B*b**4*c**2*f*x/(2*a**2*b**3*f + 2*b**5*f) + 2*B*b
**4*c*d*log(tan(e + f*x)**2 + 1)/(2*a**2*b**3*f + 2*b**5*f) - 2*B*b**4*d**2
*f*x/(2*a**2*b**3*f + 2*b**5*f) + 2*B*b**4*d**2*tan(e + f*x)/(2*a**2*b**3*f
 + 2*b**5*f) + 2*C*a**4*d**2*log(a/b + tan(e + f*x))/(2*a**2*b**3*f + 2*b**
5*f) - 4*C*a**3*b*c*d*log(a/b + tan(e + f*x))/(2*a**2*b**3*f + 2*b**5*f) -
2*C*a**3*b*d**2*tan(e + f*x)/(2*a**2*b**3*f + 2*b**5*f) + 2*C*a**2*b**2*c**
2*log(a/b + tan(e + f*x))/(2*a**2*b**3*f + 2*b**5*f) + 4*C*a**2*b**2*c*d*ta
n(e + f*x)/(2*a**2*b**3*f + 2*b**5*f) + C*a**2*b**2*d**2*tan(e + f*x)**2/(2
*a**2*b**3*f + 2*b**5*f) - 2*C*a*b**3*c**2*f*x/(2*a**2*b**3*f + 2*b**5*f) -
 2*C*a*b**3*c*d*log(tan(e + f*x)**2 + 1)/(2*a**2*b**3*f + 2*b**5*f) + 2*C*a
*b**3*d**2*f*x/(2*a**2*b**3*f + 2*b**5*f) - 2*C*a*b**3*d**2*tan(e + f*x)/(2
*a**2*b**3*f + 2*b**5*f) + C*b**4*c**2*log(tan(e + f*x)**2 + 1)/(2*a**2*b**
3*f + 2*b**5*f) - 4*C*b**4*c*d*f*x/(2*a**2*b**3*f + 2*b**5*f) + 4*C*b**4*c*
d*tan(e + f*x)/(2*a**2*b**3*f + 2*b**5*f) - C*b**4*d**2*log(tan(e + f*x)**2
 + 1)/(2*a**2*b**3*f + 2*b**5*f) + C*b**4*d**2*tan(e + f*x)**2/(2*a**2*b**3
*f + 2*b**5*f), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.31 (sec) , antiderivative size = 290, normalized size of antiderivative = 1.14

$$\int \frac{(c+d\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx$$

$$= \frac{\frac{2\,\left(((A-C)a+Bb)c^2-2\,(Ba-(A-C)b)cd-((A-C)a+Bb)d^2\right)(fx+e)}{a^2+b^2}+\frac{2\,\left((Ca^2b^2-Bab^3+Ab^4)\,c^2-2\,(Ca^3b-Ba^2b^2+Aab^3)\,cd+(Ca^4-Ba^3b+Aa}{a^2b^3+b^5}}{}$$

```
[In] integrate((c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e
)),x, algorithm="maxima")
```

```
[Out] 1/2*(2*(((A - C)*a + B*b)*c^2 - 2*(B*a - (A - C)*b)*c*d - ((A - C)*a + B*b)
*d^2)*(f*x + e)/(a^2 + b^2) + 2*((C*a^2*b^2 - B*a*b^3 + A*b^4)*c^2 - 2*(C*a
^3*b - B*a^2*b^2 + A*a*b^3)*c*d + (C*a^4 - B*a^3*b + A*a^2*b^2)*d^2)*log(b*
tan(f*x + e) + a)/(a^2*b^3 + b^5) + ((B*a - (A - C)*b)*c^2 + 2*((A - C)*a +
 B*b)*c*d - (B*a - (A - C)*b)*d^2)*log(tan(f*x + e)^2 + 1)/(a^2 + b^2) + (C
*b*d^2*tan(f*x + e)^2 + 2*(2*C*b*c*d - (C*a - B*b)*d^2)*tan(f*x + e))/b^2)/
f
```

Giac [A] (verification not implemented)

none

Time = 0.70 (sec) , antiderivative size = 331, normalized size of antiderivative = 1.30

$$\int \frac{(c+d\tan(e+fx))^2 \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{a+b\tan(e+fx)}\,dx$$

$$= \frac{\frac{2\left(Aac^2-Cac^2+Bbc^2-2\,Bacd+2\,Abcd-2\,Cbcd-Aad^2+Cad^2-Bbd^2\right)(fx+e)}{a^2+b^2}+\frac{\left(Bac^2-Abc^2+Cbc^2+2\,Aacd-2\,Cacd+2\,Bbcd-Bad^2+Abd^2-\right.}{a^2+b^2}}{}$$

```
[In] integrate((c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e
)),x, algorithm="giac")
```

```
[Out] 1/2*(2*(A*a*c^2 - C*a*c^2 + B*b*c^2 - 2*B*a*c*d + 2*A*b*c*d - 2*C*b*c*d - A
*a*d^2 + C*a*d^2 - B*b*d^2)*(f*x + e)/(a^2 + b^2) + (B*a*c^2 - A*b*c^2 + C*
b*c^2 + 2*A*a*c*d - 2*C*a*c*d + 2*B*b*c*d - B*a*d^2 + A*b*d^2 - C*b*d^2)*lo
g(tan(f*x + e)^2 + 1)/(a^2 + b^2) + 2*(C*a^2*b^2*c^2 - B*a*b^3*c^2 + A*b^4*
c^2 - 2*C*a^3*b*c*d + 2*B*a^2*b^2*c*d - 2*A*a*b^3*c*d + C*a^4*d^2 - B*a^3*b
*d^2 + A*a^2*b^2*d^2)*log(abs(b*tan(f*x + e) + a))/(a^2*b^3 + b^5) + (C*b*d
^2*tan(f*x + e)^2 + 4*C*b*c*d*tan(f*x + e) - 2*C*a*d^2*tan(f*x + e) + 2*B*b
*d^2*tan(f*x + e))/b^2)/f
```

Mupad [B] (verification not implemented)

Time = 10.71 (sec) , antiderivative size = 325, normalized size of antiderivative = 1.28

$$\int \frac{(c+d\tan(e+fx))^2 \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{a+b\tan(e+fx)}\,dx$$
$$= \frac{\tan(e+f\,x)\,\left(\frac{B\,d^2+2\,C\,c\,d}{b}-\frac{C\,a\,d^2}{b^2}\right)}{f}$$
$$+\frac{\ln\left(a+b\tan(e+f\,x)\right)\,\left(b^2\left(C\,a^2\,c^2+2\,B\,a^2\,c\,d+A\,a^2\,d^2\right)-b\left(B\,a^3\,d^2+2\,C\,c\,a^3\,d\right)-b^3\left(B\,a\,c^2+2\,A\,a\,d\right.\right.}{f\,\left(a^2\,b^3+b^5\right)}$$
$$+\frac{\ln\left(\tan(e+f\,x)+1\mathrm{i}\right)\,\left(A\,d^2-A\,c^2+B\,c^2\,1\mathrm{i}-B\,d^2\,1\mathrm{i}+C\,c^2-C\,d^2+A\,c\,d\,2\mathrm{i}+2\,B\,c\,d-C\,c\,d\,2\mathrm{i}\right)}{2\,f\,\left(b+a\,1\mathrm{i}\right)}$$
$$+\frac{\ln\left(\tan(e+f\,x)-\mathrm{i}\right)\,\left(B\,c^2-B\,d^2+2\,A\,c\,d-2\,C\,c\,d-A\,c^2\,1\mathrm{i}+A\,d^2\,1\mathrm{i}+C\,c^2\,1\mathrm{i}-C\,d^2\,1\mathrm{i}+B\,c\,d\,2\mathrm{i}\right)}{2\,f\,\left(a+b\,1\mathrm{i}\right)}$$
$$+\frac{C\,d^2\,{\tan(e+f\,x)}^2}{2\,b\,f}$$

```
[In] int(((c + d*tan(e + f*x))^2*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a + b
*tan(e + f*x)),x)
```

```
[Out] (tan(e + f*x)*((B*d^2 + 2*C*c*d)/b - (C*a*d^2)/b^2))/f + (log(a + b*tan(e +
 f*x))*(b^2*(A*a^2*d^2 + C*a^2*c^2 + 2*B*a^2*c*d) - b*(B*a^3*d^2 + 2*C*a^3*
c*d) - b^3*(B*a*c^2 + 2*A*a*c*d) + A*b^4*c^2 + C*a^4*d^2))/(f*(b^5 + a^2*b^
3)) + (log(tan(e + f*x) + 1i)*(A*d^2 - A*c^2 + B*c^2*1i - B*d^2*1i + C*c^2
- C*d^2 + A*c*d*2i + 2*B*c*d - C*c*d*2i))/(2*f*(a*1i + b)) + (log(tan(e + f
*x) - 1i)*(A*d^2*1i - A*c^2*1i + B*c^2 - B*d^2 + C*c^2*1i - C*d^2*1i + 2*A*
c*d + B*c*d*2i - 2*C*c*d))/(2*f*(a + b*1i)) + (C*d^2*tan(e + f*x)^2)/(2*b*f
)
```

3.62 $\int \frac{(c+d\tan(e+fx))^2(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx$

Optimal result . . . 547
Rubi [A] (verified) . . . 548
Mathematica [C] (verified) . . . 551
Maple [A] (verified) . . . 551
Fricas [B] (verification not implemented) . . . 552
Sympy [C] (verification not implemented) . . . 553
Maxima [A] (verification not implemented) . . . 561
Giac [B] (verification not implemented) . . . 562
Mupad [B] (verification not implemented) . . . 562

Optimal result

Integrand size = 45, antiderivative size = 415

$$\int \frac{(c+d\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx =$$
$$-\frac{(a^2(c^2C+2Bcd-Cd^2-A(c^2-d^2))-b^2(c^2C+2Bcd-Cd^2-A(c^2-d^2))-2ab(2c(A-C)d+B(c}{(a^2+b^2)^2}$$
$$-\frac{(2ab(c^2C+2Bcd-Cd^2-A(c^2-d^2))+a^2(2c(A-C)d+B(c^2-d^2))-b^2(2c(A-C)d+B(c^2-d^2)}{(a^2+b^2)^2\,f}$$
$$-\frac{(bc-ad)\,(a^3bBd-2a^4Cd-b^4(Bc+2Ad)-ab^3(2Ac-2cC-3Bd)+a^2b^2(Bc-4Cd))\log(a+b\tan}{b^3\,(a^2+b^2)^2\,f}$$
$$+\frac{(Ab^2-abB+2a^2C+b^2C)\,d^2\tan(e+fx)}{b^2\,(a^2+b^2)\,f}-\frac{(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^2}{b\,(a^2+b^2)\,f(a+b\tan(e+fx))}$$

```
[Out] -(a^2*(c^2*C+2*B*c*d-C*d^2-A*(c^2-d^2))-b^2*(c^2*C+2*B*c*d-C*d^2-A*(c^2-d^2
))-2*a*b*(2*c*(A-C)*d+B*(c^2-d^2)))*x/(a^2+b^2)^2-(2*a*b*(c^2*C+2*B*c*d-C*d
^2-A*(c^2-d^2))+a^2*(2*c*(A-C)*d+B*(c^2-d^2))-b^2*(2*c*(A-C)*d+B*(c^2-d^2))
)*ln(cos(f*x+e))/(a^2+b^2)^2/f-(-a*d+b*c)*(a^3*b*B*d-2*a^4*C*d-b^4*(2*A*d+B
*c)-a*b^3*(2*A*c-3*B*d-2*C*c)+a^2*b^2*(B*c-4*C*d))*ln(a+b*tan(f*x+e))/b^3/(
a^2+b^2)^2/f+(A*b^2-B*a*b+2*C*a^2+C*b^2)*d^2*tan(f*x+e)/b^2/(a^2+b^2)/f-(A*
b^2-a*(B*b-C*a))*(c+d*tan(f*x+e))^2/b/(a^2+b^2)/f/(a+b*tan(f*x+e))
```

Rubi [A] (verified)

Time = 1.17 (sec) , antiderivative size = 415, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.133, Rules used = {3726, 3718, 3707, 3698, 31, 3556}

$$\int \frac{(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^2}\,dx =$$
$$-\frac{\log(\cos(e+fx))\left(a^2(2cd(A-C)+B(c^2-d^2))+2ab(-A(c^2-d^2)+2Bcd+c^2C-Cd^2)-b^2(2cd(A-\right.}{f\left(a^2+b^2\right)^2}$$
$$-\frac{x(a^2(-A(c^2-d^2)+2Bcd+c^2C-Cd^2)-2ab(2cd(A-C)+B(c^2-d^2))-b^2(-A(c^2-d^2)+2Bcd+}{\left(a^2+b^2\right)^2}$$
$$-\frac{(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^2}{bf\,(a^2+b^2)\,(a+b\tan(e+fx))}+\frac{d^2\tan(e+fx)\,(2a^2C-abB+Ab^2+b^2C)}{b^2f\,(a^2+b^2)}$$
$$-\frac{(bc-ad)\left(-2a^4Cd+a^3bBd+a^2b^2(Bc-4Cd)-ab^3(2Ac-3Bd-2cC)-b^4(2Ad+Bc)\right)\log(a+b\tan}{b^3f\left(a^2+b^2\right)^2}$$

```
[In] Int[((c + d*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a + b
*Tan[e + f*x])^2,x]
```

```
[Out] -(((a^2*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - b^2*(c^2*C + 2*B*c*d -
C*d^2 - A*(c^2 - d^2)) - 2*a*b*(2*c*(A - C)*d + B*(c^2 - d^2)))*x)/(a^2 + b
^2)^2) - ((2*a*b*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2)) + a^2*(2*c*(A -
C)*d + B*(c^2 - d^2)) - b^2*(2*c*(A - C)*d + B*(c^2 - d^2)))*Log[Cos[e + f*
x]])/((a^2 + b^2)^2*f) - ((b*c - a*d)*(a^3*b*B*d - 2*a^4*C*d - b^4*(B*c + 2
*A*d) - a*b^3*(2*A*c - 2*c*C - 3*B*d) + a^2*b^2*(B*c - 4*C*d))*Log[a + b*Ta
n[e + f*x]])/(b^3*(a^2 + b^2)^2*f) + ((A*b^2 - a*b*B + 2*a^2*C + b^2*C)*d^2
*Tan[e + f*x])/(b^2*(a^2 + b^2)*f) - ((A*b^2 - a*(b*B - a*C))*(c + d*Tan[e
+ f*x])^2)/(b*(a^2 + b^2)*f*(a + b*Tan[e + f*x]))
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3698

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3707

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[(a*A + b*B - a
*C)*(x/(a^2 + b^2)), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3726

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*d^2 + c*(c*C - B*d))*(a + b*Tan[e +
 f*x])^m*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rubi steps

$$\text{integral} = -\frac{(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^2}{b\,(a^2+b^2)\,f(a+b\tan(e+fx))}$$
$$+\frac{\int \frac{(c+d\tan(e+fx))\left((bB-aC)(bc-2ad)+Ab(ac+2bd)-b((A-C)(bc-ad)-B(ac+bd))\tan(e+fx)+\left(Ab^2-abB+2a^2C+b^2C\right)d\tan^2(e+fx)\right)}{a+b\tan(e+fx)}\,d}{b\,(a^2+b^2)}$$

$$=\frac{(Ab^2-abB+2a^2C+b^2C)\,d^2\tan(e+fx)}{b^2\,(a^2+b^2)\,f}-\frac{(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^2}{b\,(a^2+b^2)\,f(a+b\tan(e+fx))}$$
$$-\frac{\int \frac{a\left(Ab^2-abB+2a^2C+b^2C\right)d^2-bc((bB-aC)(bc-2ad)+Ab(ac+2bd))-b^2\left(2aAcd-2acCd-Ab\left(c^2-d^2\right)+aB\left(c^2-d^2\right)+b\left(c^2C+2Bc\right.\right.}{a+b\tan(e+fx)}}{b^2\,(a^2+b^2)}$$

$$=$$

$$-\frac{(a^2(c^2C+2Bcd-Cd^2-A(c^2-d^2))-b^2(c^2C+2Bcd-Cd^2-A(c^2-d^2))-2ab(2c(A-C)d}{\left(a^2+b^2\right)^2}$$

$$+\frac{\left(Ab^2-abB+2a^2C+b^2C\right)d^2\tan(e+fx)}{b^2\left(a^2+b^2\right)f}$$

$$-\frac{\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^2}{b\left(a^2+b^2\right)f(a+b\tan(e+fx))}$$

$$-\frac{\left((bc-ad)\left(a^3bBd-2a^4Cd-b^4(Bc+2Ad)-ab^3(2Ac-2cC-3Bd)+a^2b^2(Bc-4Cd)\right)\right)\int\frac{1+}{a+}}{b^2\left(a^2+b^2\right)^2}$$

$$+\frac{(2ab(c^2C+2Bcd-Cd^2-A(c^2-d^2))+a^2(2c(A-C)d+B(c^2-d^2))-b^2(2c(A-C)d+B(c^2}{\left(a^2+b^2\right)^2}$$

$$=$$

$$-\frac{(a^2(c^2C+2Bcd-Cd^2-A(c^2-d^2))-b^2(c^2C+2Bcd-Cd^2-A(c^2-d^2))-2ab(2c(A-C)d}{\left(a^2+b^2\right)^2}$$

$$-\frac{(2ab(c^2C+2Bcd-Cd^2-A(c^2-d^2))+a^2(2c(A-C)d+B(c^2-d^2))-b^2(2c(A-C)d+B(c^2}{\left(a^2+b^2\right)^2f}$$

$$+\frac{\left(Ab^2-abB+2a^2C+b^2C\right)d^2\tan(e+fx)}{b^2\left(a^2+b^2\right)f}$$

$$-\frac{\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^2}{b\left(a^2+b^2\right)f(a+b\tan(e+fx))}$$

$$-\frac{\left((bc-ad)\left(a^3bBd-2a^4Cd-b^4(Bc+2Ad)-ab^3(2Ac-2cC-3Bd)+a^2b^2(Bc-4Cd)\right)\right)\text{Subs}}{b^3\left(a^2+b^2\right)^2f}$$

$$=$$

$$-\frac{(a^2(c^2C+2Bcd-Cd^2-A(c^2-d^2))-b^2(c^2C+2Bcd-Cd^2-A(c^2-d^2))-2ab(2c(A-C)d}{\left(a^2+b^2\right)^2}$$

$$-\frac{(2ab(c^2C+2Bcd-Cd^2-A(c^2-d^2))+a^2(2c(A-C)d+B(c^2-d^2))-b^2(2c(A-C)d+B(c^2}{\left(a^2+b^2\right)^2f}$$

$$-\frac{(bc-ad)\left(a^3bBd-2a^4Cd-b^4(Bc+2Ad)-ab^3(2Ac-2cC-3Bd)+a^2b^2(Bc-4Cd)\right)\log(a+}{b^3\left(a^2+b^2\right)^2f}$$

$$+\frac{\left(Ab^2-abB+2a^2C+b^2C\right)d^2\tan(e+fx)}{b^2\left(a^2+b^2\right)f}$$

$$-\frac{\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^2}{b\left(a^2+b^2\right)f(a+b\tan(e+fx))}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 6.41 (sec) , antiderivative size = 277, normalized size of antiderivative = 0.67

$$\int \frac{(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^2}\,dx$$
$$=\frac{\frac{(-iA+B+iC)(c+id)^2\log(i-\tan(e+fx))}{(a+ib)^2}+\frac{(iA+B-iC)(c-id)^2\log(i+\tan(e+fx))}{(a-ib)^2}+\frac{2(bc-ad)\left(-a^3bBd+2a^4Cd+b^4(Bc+2Ad)+ab^3(2Ac-2\right.}{b^3(a^2+b}}{2f}$$

```
[In] Integrate[((c + d*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(a + b*Tan[e + f*x])^2,x]
```

```
[Out] ((((-I)*A + B + I*C)*(c + I*d)^2*Log[I - Tan[e + f*x]])/(a + I*b)^2 + ((I*A
 + B - I*C)*(c - I*d)^2*Log[I + Tan[e + f*x]])/(a - I*b)^2 + (2*(b*c - a*d)
*(-(a^3*b*B*d) + 2*a^4*C*d + b^4*(B*c + 2*A*d) + a*b^3*(2*A*c - 2*c*C - 3*B
*d) + a^2*b^2*(-(B*c) + 4*C*d))*Log[a + b*Tan[e + f*x]])/(b^3*(a^2 + b^2)^2
) - (2*(A*b^2 - a*b*B + 2*a^2*C + b^2*C)*(b*c - a*d)^2)/(b^3*(a^2 + b^2)*(a
 + b*Tan[e + f*x])) + (2*C*(c + d*Tan[e + f*x])^2)/(b*(a + b*Tan[e + f*x]))
)/(2*f)
```

Maple [A] (verified)

Time = 0.31 (sec) , antiderivative size = 552, normalized size of antiderivative = 1.33

method	result
derivativedivides	$\frac{\frac{\tan(fx+e)C\,d^2}{b^2}-\frac{A\,a^2d^2b^2-2Aa\,b^3cd+A\,b^4c^2-B\,a^3d^2b+2B\,a^2cd\,b^2-Ba\,b^3c^2+a^4C\,d^2-2C\,a^3cdb+C\,a^2c^2b^2}{b^3\left(a^2+b^2\right)(a+b\tan(fx+e))}+\frac{\left(-2A\,a^2b^3cd+2Aa\right.}{}}{}$
default	$\frac{\frac{\tan(fx+e)C\,d^2}{b^2}-\frac{A\,a^2d^2b^2-2Aa\,b^3cd+A\,b^4c^2-B\,a^3d^2b+2B\,a^2cd\,b^2-Ba\,b^3c^2+a^4C\,d^2-2C\,a^3cdb+C\,a^2c^2b^2}{b^3\left(a^2+b^2\right)(a+b\tan(fx+e))}+\frac{\left(-2A\,a^2b^3cd+2Aa\right.}{}}{}$
norman	$\frac{\frac{a\left(A\,a^2c^2-A\,a^2d^2+4Aabcd-A\,b^2c^2+A\,b^2d^2-2B\,a^2cd+2Bab\,c^2-2Bab\,d^2+2B\,b^2cd-C\,a^2c^2+a^2C\,d^2-4Cabcd+C\,b^2c^2-C\,b^2d^2\right)x}{a^4+2a^2b^2+b^4}+}{}$
parallelrisch	Expression too large to display
risch	Expression too large to display

```
[In] int((c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2,x
,method=_RETURNVERBOSE)
```

```
[Out] 1/f*(tan(f*x+e)*C*d^2/b^2-1/b^3*(A*a^2*b^2*d^2-2*A*a*b^3*c*d+A*b^4*c^2-B*a^
3*b*d^2+2*B*a^2*b^2*c*d-B*a*b^3*c^2+C*a^4*d^2-2*C*a^3*b*c*d+C*a^2*b^2*c^2)/
(a^2+b^2)/(a+b*tan(f*x+e))+(-2*A*a^2*b^3*c*d+2*A*a*b^4*c^2-2*A*a*b^4*d^2+2*
```

```
A*b^5*c*d+B*a^4*b*d^2-B*a^2*b^3*c^2+3*B*a^2*b^3*d^2-4*B*a*b^4*c*d+B*b^5*c^2
-2*C*a^5*d^2+2*C*a^4*b*c*d-4*C*a^3*b^2*d^2+6*C*a^2*b^3*c*d-2*C*a*b^4*c^2)/b
^3/(a^2+b^2)^2*ln(a+b*tan(f*x+e))+1/(a^2+b^2)^2*(1/2*(2*A*a^2*c*d-2*A*a*b*c
^2+2*A*a*b*d^2-2*A*b^2*c*d+B*a^2*c^2-B*a^2*d^2+4*B*a*b*c*d-B*b^2*c^2+B*b^2*
d^2-2*C*a^2*c*d+2*C*a*b*c^2-2*C*a*b*d^2+2*C*b^2*c*d)*ln(1+tan(f*x+e)^2)+(A*
a^2*c^2-A*a^2*d^2+4*A*a*b*c*d-A*b^2*c^2+A*b^2*d^2-2*B*a^2*c*d+2*B*a*b*c^2-2
*B*a*b*d^2+2*B*b^2*c*d-C*a^2*c^2+C*a^2*d^2-4*C*a*b*c*d+C*b^2*c^2-C*b^2*d^2)
*arctan(tan(f*x+e))))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 964 vs. $2(413) = 826$.

Time = 0.59 (sec) , antiderivative size = 964, normalized size of antiderivative = 2.32

$$\int \frac{(c+d\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx$$
$$= \frac{2\,(Ca^4b^2+2\,Ca^2b^4+Cb^6)d^2\tan{(fx+e)}^2-2\,(Ca^2b^4-Bab^5+Ab^6)c^2+4\,(Ca^3b^3-Ba^2b^4+Aab^5)cd-2}{}$$

```
[In] integrate((c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e
))^2,x, algorithm="fricas")
```

```
[Out] 1/2*(2*(C*a^4*b^2 + 2*C*a^2*b^4 + C*b^6)*d^2*tan(f*x + e)^2 - 2*(C*a^2*b^4
- B*a*b^5 + A*b^6)*c^2 + 4*(C*a^3*b^3 - B*a^2*b^4 + A*a*b^5)*c*d - 2*(C*a^4
*b^2 - B*a^3*b^3 + A*a^2*b^4)*d^2 + 2*(((A - C)*a^3*b^3 + 2*B*a^2*b^4 - (A
- C)*a*b^5)*c^2 - 2*(B*a^3*b^3 - 2*(A - C)*a^2*b^4 - B*a*b^5)*c*d - ((A - C
)*a^3*b^3 + 2*B*a^2*b^4 - (A - C)*a*b^5)*d^2)*f*x - ((B*a^3*b^3 - 2*(A - C)
*a^2*b^4 - B*a*b^5)*c^2 - 2*(C*a^5*b - (A - 3*C)*a^3*b^3 - 2*B*a^2*b^4 + A*
a*b^5)*c*d + (2*C*a^6 - B*a^5*b + 4*C*a^4*b^2 - 3*B*a^3*b^3 + 2*A*a^2*b^4)*
d^2 + ((B*a^2*b^4 - 2*(A - C)*a*b^5 - B*b^6)*c^2 - 2*(C*a^4*b^2 - (A - 3*C)
*a^2*b^4 - 2*B*a*b^5 + A*b^6)*c*d + (2*C*a^5*b - B*a^4*b^2 + 4*C*a^3*b^3 -
3*B*a^2*b^4 + 2*A*a*b^5)*d^2)*tan(f*x + e))*log((b^2*tan(f*x + e)^2 + 2*a*b
*tan(f*x + e) + a^2)/(tan(f*x + e)^2 + 1)) - (2*(C*a^5*b + 2*C*a^3*b^3 + C*
a*b^5)*c*d - (2*C*a^6 - B*a^5*b + 4*C*a^4*b^2 - 2*B*a^3*b^3 + 2*C*a^2*b^4 -
 B*a*b^5)*d^2 + (2*(C*a^4*b^2 + 2*C*a^2*b^4 + C*b^6)*c*d - (2*C*a^5*b - B*a
^4*b^2 + 4*C*a^3*b^3 - 2*B*a^2*b^4 + 2*C*a*b^5 - B*b^6)*d^2)*tan(f*x + e))*
log(1/(tan(f*x + e)^2 + 1)) + 2*((C*a^3*b^3 - B*a^2*b^4 + A*a*b^5)*c^2 - 2*
(C*a^4*b^2 - B*a^3*b^3 + A*a^2*b^4)*c*d + (2*C*a^5*b - B*a^4*b^2 + (A + 2*C
)*a^3*b^3 + C*a*b^5)*d^2 + (((A - C)*a^2*b^4 + 2*B*a*b^5 - (A - C)*b^6)*c^2
 - 2*(B*a^2*b^4 - 2*(A - C)*a*b^5 - B*b^6)*c*d - ((A - C)*a^2*b^4 + 2*B*a*b
^5 - (A - C)*b^6)*d^2)*f*x)*tan(f*x + e))/((a^4*b^4 + 2*a^2*b^6 + b^8)*f*ta
n(f*x + e) + (a^5*b^3 + 2*a^3*b^5 + a*b^7)*f)
```

Sympy [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 2.97 (sec) , antiderivative size = 16225, normalized size of antiderivative = 39.10

$$\int \frac{(c+d\tan(e+fx))^2 (A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx = \text{Too large to display}$$

```
[In] integrate((c+d*tan(f*x+e))**2*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x
+e))**2,x)
```

```
[Out] Piecewise((zoo*x*(c + d*tan(e))**2*(A + B*tan(e) + C*tan(e)**2)/tan(e)**2,
Eq(a, 0) & Eq(b, 0) & Eq(f, 0)), ((A*c**2*x + A*c*d*log(tan(e + f*x)**2 + 1
)/f - A*d**2*x + A*d**2*tan(e + f*x)/f + B*c**2*log(tan(e + f*x)**2 + 1)/(2
*f) - 2*B*c*d*x + 2*B*c*d*tan(e + f*x)/f - B*d**2*log(tan(e + f*x)**2 + 1)/
(2*f) + B*d**2*tan(e + f*x)**2/(2*f) - C*c**2*x + C*c**2*tan(e + f*x)/f - C
*c*d*log(tan(e + f*x)**2 + 1)/f + C*c*d*tan(e + f*x)**2/f + C*d**2*x + C*d*
*2*tan(e + f*x)**3/(3*f) - C*d**2*tan(e + f*x)/f)/a**2, Eq(b, 0)), (-A*c**2
*f*x*tan(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) -
4*b**2*f) + 2*I*A*c**2*f*x*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**
2*f*tan(e + f*x) - 4*b**2*f) + A*c**2*f*x/(4*b**2*f*tan(e + f*x)**2 - 8*I*b
**2*f*tan(e + f*x) - 4*b**2*f) - A*c**2*tan(e + f*x)/(4*b**2*f*tan(e + f*x)
**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 2*I*A*c**2/(4*b**2*f*tan(e + f*
x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 2*I*A*c*d*f*x*tan(e + f*x)**2
/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 4*A*c*d*
f*x*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b*
*2*f) - 2*I*A*c*d*f*x/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) -
 4*b**2*f) + 2*I*A*c*d*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*
tan(e + f*x) - 4*b**2*f) + A*d**2*f*x*tan(e + f*x)**2/(4*b**2*f*tan(e + f*x
)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 2*I*A*d**2*f*x*tan(e + f*x)/(4
*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - A*d**2*f*x/
(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 3*A*d**2*
tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f
) + 2*I*A*d**2/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2
*f) + I*B*c**2*f*x*tan(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*t
an(e + f*x) - 4*b**2*f) + 2*B*c**2*f*x*tan(e + f*x)/(4*b**2*f*tan(e + f*x)*
*2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - I*B*c**2*f*x/(4*b**2*f*tan(e + f
*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + I*B*c**2*tan(e + f*x)/(4*b**
2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 2*B*c*d*f*x*tan
(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f
) - 4*I*B*c*d*f*x*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e
 + f*x) - 4*b**2*f) - 2*B*c*d*f*x/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*ta
n(e + f*x) - 4*b**2*f) - 6*B*c*d*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8
*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 4*I*B*c*d/(4*b**2*f*tan(e + f*x)**2 -
8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 3*I*B*d**2*f*x*tan(e + f*x)**2/(4*b**
```

2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 6*B*d**2*f*x*ta
n(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f)
- 3*I*B*d**2*f*x/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b*
*2*f) + 2*B*d**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)**2/(4*b**2*f*tan(e +
f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 4*I*B*d**2*log(tan(e + f*x
)**2 + 1)*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x)
- 4*b**2*f) - 2*B*d**2*log(tan(e + f*x)**2 + 1)/(4*b**2*f*tan(e + f*x)**2 -
8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 5*I*B*d**2*tan(e + f*x)/(4*b**2*f*ta
n(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 4*B*d**2/(4*b**2*f*ta
n(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + C*c**2*f*x*tan(e + f*
x)**2/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 2*I
*C*c**2*f*x*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x
) - 4*b**2*f) - C*c**2*f*x/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f
*x) - 4*b**2*f) - 3*C*c**2*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**
2*f*tan(e + f*x) - 4*b**2*f) + 2*I*C*c**2/(4*b**2*f*tan(e + f*x)**2 - 8*I*b
2*f*tan(e + f*x) - 4*b2*f) + 6*I*C*c*d*f*x*tan(e + f*x)**2/(4*b**2*f*ta
n(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 12*C*c*d*f*x*tan(e +
f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 6*I*
C*c*d*f*x/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) +
4*C*c*d*log(tan(e + f*x)**2 + 1)*tan(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2
- 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 8*I*C*c*d*log(tan(e + f*x)**2 + 1)
*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*
f) - 4*C*c*d*log(tan(e + f*x)**2 + 1)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*
f*tan(e + f*x) - 4*b**2*f) - 10*I*C*c*d*tan(e + f*x)/(4*b**2*f*tan(e + f*x)
2 - 8*I*b2*f*tan(e + f*x) - 4*b**2*f) - 8*C*c*d/(4*b**2*f*tan(e + f*x)*
*2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 9*C*d**2*f*x*tan(e + f*x)**2/(4*
b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 18*I*C*d**2*
f*x*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b*
*2*f) + 9*C*d**2*f*x/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) -
4*b**2*f) + 4*I*C*d**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)**2/(4*b**2*f*t
an(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 8*C*d**2*log(tan(e +
f*x)**2 + 1)*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f
*x) - 4*b**2*f) - 4*I*C*d**2*log(tan(e + f*x)**2 + 1)/(4*b**2*f*tan(e + f*x
)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 4*C*d**2*tan(e + f*x)**3/(4*b*
*2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 19*C*d**2*tan(
e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) -
14*I*C*d**2/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f)
, Eq(a, -I*b)), (-A*c**2*f*x*tan(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 + 8*
I*b**2*f*tan(e + f*x) - 4*b**2*f) - 2*I*A*c**2*f*x*tan(e + f*x)/(4*b**2*f*t
an(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + A*c**2*f*x/(4*b**2*f
*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - A*c**2*tan(e + f*x
)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 2*I*A*c
2/(4*b2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 2*I*A
*c*d*f*x*tan(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x
) - 4*b**2*f) + 4*A*c*d*f*x*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b*

```
*2*f*tan(e + f*x) - 4*b**2*f) + 2*I*A*c*d*f*x/(4*b**2*f*tan(e + f*x)**2 + 8
*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 2*I*A*c*d*tan(e + f*x)/(4*b**2*f*tan(e
 + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + A*d**2*f*x*tan(e + f*x)*
*2/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 2*I*A*
d**2*f*x*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) -
 4*b**2*f) - A*d**2*f*x/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x)
 - 4*b**2*f) - 3*A*d**2*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f
*tan(e + f*x) - 4*b**2*f) - 2*I*A*d**2/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2
*f*tan(e + f*x) - 4*b**2*f) - I*B*c**2*f*x*tan(e + f*x)**2/(4*b**2*f*tan(e
+ f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 2*B*c**2*f*x*tan(e + f*x)
/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + I*B*c**2
*f*x/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - I*B*
c**2*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b
**2*f) + 2*B*c*d*f*x*tan(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f
*tan(e + f*x) - 4*b**2*f) + 4*I*B*c*d*f*x*tan(e + f*x)/(4*b**2*f*tan(e + f*
x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 2*B*c*d*f*x/(4*b**2*f*tan(e +
 f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 6*B*c*d*tan(e + f*x)/(4*b*
*2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 4*I*B*c*d/(4*b
**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 3*I*B*d**2*f*
x*tan(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b
**2*f) + 6*B*d**2*f*x*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*t
an(e + f*x) - 4*b**2*f) + 3*I*B*d**2*f*x/(4*b**2*f*tan(e + f*x)**2 + 8*I*b*
*2*f*tan(e + f*x) - 4*b**2*f) + 2*B*d**2*log(tan(e + f*x)**2 + 1)*tan(e + f
*x)**2/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 4*
I*B*d**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 +
8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 2*B*d**2*log(tan(e + f*x)**2 + 1)/(4*
b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 5*I*B*d**2*t
an(e + f*x)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f)
 - 4*B*d**2/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f)
 + C*c**2*f*x*tan(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e
+ f*x) - 4*b**2*f) + 2*I*C*c**2*f*x*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2
+ 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - C*c**2*f*x/(4*b**2*f*tan(e + f*x)**
2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 3*C*c**2*tan(e + f*x)/(4*b**2*f*t
an(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 2*I*C*c**2/(4*b**2*f
*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 6*I*C*c*d*f*x*tan(
e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f)
 + 12*C*c*d*f*x*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e +
 f*x) - 4*b**2*f) + 6*I*C*c*d*f*x/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*ta
n(e + f*x) - 4*b**2*f) + 4*C*c*d*log(tan(e + f*x)**2 + 1)*tan(e + f*x)**2/(
4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 8*I*C*c*d*
log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*
f*tan(e + f*x) - 4*b**2*f) - 4*C*c*d*log(tan(e + f*x)**2 + 1)/(4*b**2*f*tan
(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 10*I*C*c*d*tan(e + f*x
)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 8*C*c*d
/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 9*C*d**2
```

*f*x*tan(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) -
4*b**2*f) - 18*I*C*d**2*f*x*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b*
*2*f*tan(e + f*x) - 4*b**2*f) + 9*C*d**2*f*x/(4*b**2*f*tan(e + f*x)**2 + 8*
I*b**2*f*tan(e + f*x) - 4*b**2*f) - 4*I*C*d**2*log(tan(e + f*x)**2 + 1)*tan
(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f
) + 8*C*d**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**
2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 4*I*C*d**2*log(tan(e + f*x)**2 +
1)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 4*C*d*
*2*tan(e + f*x)**3/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*
b**2*f) + 19*C*d**2*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan
(e + f*x) - 4*b**2*f) + 14*I*C*d**2/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*
tan(e + f*x) - 4*b**2*f), Eq(a, I*b)), (x*(c + d*tan(e))**2*(A + B*tan(e) +
C*tan(e)**2)/(a + b*tan(e))**2, Eq(f, 0)), (-2*A*a**4*b**2*d**2/(2*a**5*b*
*3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f
*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) + 2*A*a**3*b**3*c**2*f*x/(2*a**5*
b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e +
f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) - 4*A*a**3*b**3*c*d*log(a/b + t
an(e + f*x))/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f +
4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) + 2*A*a**3
*b**3*c*d*log(tan(e + f*x)**2 + 1)/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f
*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*ta
n(e + f*x)) + 4*A*a**3*b**3*c*d/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x)
+ 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e
+ f*x)) - 2*A*a**3*b**3*d**2*f*x/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*
x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan
(e + f*x)) + 2*A*a**2*b**4*c**2*f*x*tan(e + f*x)/(2*a**5*b**3*f + 2*a**4*b*
*4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f
+ 2*b**8*f*tan(e + f*x)) + 4*A*a**2*b**4*c**2*log(a/b + tan(e + f*x))/(2*a
5*b3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan
(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) - 2*A*a**2*b**4*c**2*log(ta
n(e + f*x)**2 + 1)/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**
5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) - 2*
A*a**2*b**4*c**2/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*
f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) + 8*A*
a**2*b**4*c*d*f*x/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5
*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) - 4*A
*a**2*b**4*c*d*log(a/b + tan(e + f*x))*tan(e + f*x)/(2*a**5*b**3*f + 2*a**4
*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**
7*f + 2*b**8*f*tan(e + f*x)) + 2*A*a**2*b**4*c*d*log(tan(e + f*x)**2 + 1)*t
an(e + f*x)/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4
*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) - 2*A*a**2*
b**4*d**2*f*x*tan(e + f*x)/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*
a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*
x)) - 4*A*a**2*b**4*d**2*log(a/b + tan(e + f*x))/(2*a**5*b**3*f + 2*a**4*b*
*4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f

+ 2*b**8*f*tan(e + f*x)) + 2*A*a**2*b**4*d**2*log(tan(e + f*x)**2 + 1)/(2*
a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*ta
n(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) - 2*A*a**2*b**4*d**2/(2*a*
*5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(
e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) - 2*A*a*b**5*c**2*f*x/(2*a**
5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e
 + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) + 4*A*a*b**5*c**2*log(a/b + t
an(e + f*x))*tan(e + f*x)/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a
3*b5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x
)) - 2*A*a*b**5*c**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*a**5*b**3*f +
 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) +
2*a*b**7*f + 2*b**8*f*tan(e + f*x)) + 8*A*a*b**5*c*d*f*x*tan(e + f*x)/(2*a*
*5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(
e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) + 4*A*a*b**5*c*d*log(a/b + t
an(e + f*x))/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f +
4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) - 2*A*a*b*
*5*c*d*log(tan(e + f*x)**2 + 1)/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x)
 + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e
 + f*x)) + 4*A*a*b**5*c*d/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a
3*b5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x
)) + 2*A*a*b**5*d**2*f*x/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a*
*3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)
) - 4*A*a*b**5*d**2*log(a/b + tan(e + f*x))*tan(e + f*x)/(2*a**5*b**3*f + 2
*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*
a*b**7*f + 2*b**8*f*tan(e + f*x)) + 2*A*a*b**5*d**2*log(tan(e + f*x)**2 + 1
)*tan(e + f*x)/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f
+ 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) - 2*A*b*
*6*c**2*f*x*tan(e + f*x)/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a*
*3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)
) - 2*A*b**6*c**2/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5
*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) + 4*A
*b**6*c*d*log(a/b + tan(e + f*x))*tan(e + f*x)/(2*a**5*b**3*f + 2*a**4*b**4
*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f +
 2*b**8*f*tan(e + f*x)) - 2*A*b**6*c*d*log(tan(e + f*x)**2 + 1)*tan(e + f*x
)/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6
*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) + 2*A*b**6*d**2*f*x*t
an(e + f*x)/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4
*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) + 2*B*a**5*
b*d**2*log(a/b + tan(e + f*x))/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x)
+ 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e
+ f*x)) + 2*B*a**5*b*d**2/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a
3*b5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x
)) - 4*B*a**4*b**2*c*d/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3
*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x))
+ 2*B*a**4*b**2*d**2*log(a/b + tan(e + f*x))*tan(e + f*x)/(2*a**5*b**3*f +

2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2
*a*b**7*f + 2*b**8*f*tan(e + f*x)) - 2*B*a**3*b**3*c**2*log(a/b + tan(e + f
*x))/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b
6*f*tan(e + f*x) + 2*a*b7*f + 2*b**8*f*tan(e + f*x)) + B*a**3*b**3*c**2
*log(tan(e + f*x)**2 + 1)/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a
3*b5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x
)) + 2*B*a**3*b**3*c**2/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**
3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x))
 - 4*B*a**3*b**3*c*d*f*x/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a*
*3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)
) + 6*B*a**3*b**3*d**2*log(a/b + tan(e + f*x))/(2*a**5*b**3*f + 2*a**4*b**4
*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f +
 2*b**8*f*tan(e + f*x)) - B*a**3*b**3*d**2*log(tan(e + f*x)**2 + 1)/(2*a**5
*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e
+ f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) + 2*B*a**3*b**3*d**2/(2*a**5*b
3*f + 2*a4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e +
f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) + 4*B*a**2*b**4*c**2*f*x/(2*a**5
*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e
+ f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) - 2*B*a**2*b**4*c**2*log(a/b +
 tan(e + f*x))*tan(e + f*x)/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4
*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f
*x)) + B*a**2*b**4*c**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*a**5*b**3*
f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x)
 + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) - 4*B*a**2*b**4*c*d*f*x*tan(e + f*x)
/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*
f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) - 8*B*a**2*b**4*c*d*lo
g(a/b + tan(e + f*x))/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*
b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) +
 4*B*a**2*b**4*c*d*log(tan(e + f*x)**2 + 1)/(2*a**5*b**3*f + 2*a**4*b**4*f*
tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*
b**8*f*tan(e + f*x)) - 4*B*a**2*b**4*c*d/(2*a**5*b**3*f + 2*a**4*b**4*f*tan
(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**
8*f*tan(e + f*x)) - 4*B*a**2*b**4*d**2*f*x/(2*a**5*b**3*f + 2*a**4*b**4*f*t
an(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b
8*f*tan(e + f*x)) + 6*B*a2*b**4*d**2*log(a/b + tan(e + f*x))*tan(e + f*
x)/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**
6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) - B*a**2*b**4*d**2*l
og(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e +
 f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*
tan(e + f*x)) + 4*B*a*b**5*c**2*f*x*tan(e + f*x)/(2*a**5*b**3*f + 2*a**4*b*
*4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f
 + 2*b**8*f*tan(e + f*x)) + 2*B*a*b**5*c**2*log(a/b + tan(e + f*x))/(2*a**5
*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e
+ f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) - B*a*b**5*c**2*log(tan(e + f*
x)**2 + 1)/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*

a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) + 2*B*a*b**5
*c**2/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*
b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) + 4*B*a*b**5*c*d*
f*x/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b*
*6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) - 8*B*a*b**5*c*d*lo
g(a/b + tan(e + f*x))*tan(e + f*x)/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f
*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*ta
n(e + f*x)) + 4*B*a*b**5*c*d*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*a**5*
b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e +
 f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) - 4*B*a*b**5*d**2*f*x*tan(e + f
*x)/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b*
*6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) + B*a*b**5*d**2*log
(tan(e + f*x)**2 + 1)/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*
b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) +
 2*B*b**6*c**2*log(a/b + tan(e + f*x))*tan(e + f*x)/(2*a**5*b**3*f + 2*a**4
*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**
7*f + 2*b**8*f*tan(e + f*x)) - B*b**6*c**2*log(tan(e + f*x)**2 + 1)*tan(e +
 f*x)/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*
b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) + 4*B*b**6*c*d*f*
x*tan(e + f*x)/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f
+ 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) + B*b**6
*d**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*a**5*b**3*f + 2*a**4*b**4*f*
tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*
b**8*f*tan(e + f*x)) - 4*C*a**6*d**2*log(a/b + tan(e + f*x))/(2*a**5*b**3*f
 + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x)
+ 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) - 4*C*a**6*d**2/(2*a**5*b**3*f + 2*a*
*4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b
7*f + 2*b8*f*tan(e + f*x)) + 4*C*a**5*b*c*d*log(a/b + tan(e + f*x))/(2*
a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*ta
n(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) + 4*C*a**5*b*c*d/(2*a**5*b
3*f + 2*a4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e +
f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) - 4*C*a**5*b*d**2*log(a/b + tan(
e + f*x))*tan(e + f*x)/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3
*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x))
- 2*C*a**4*b**2*c**2/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b
5*f + 4*a2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) +
4*C*a**4*b**2*c*d*log(a/b + tan(e + f*x))*tan(e + f*x)/(2*a**5*b**3*f + 2*a
4*b4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*
b**7*f + 2*b**8*f*tan(e + f*x)) - 8*C*a**4*b**2*d**2*log(a/b + tan(e + f*x)
)/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6
*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) + 2*C*a**4*b**2*d**2*
tan(e + f*x)**2/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f
 + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) - 6*C*a
4*b2*d**2/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f +
 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) - 2*C*a**

3*b**3*c**2*f*x/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f
 + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) + 12*C*
a**3*b**3*c*d*log(a/b + tan(e + f*x))/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e
+ f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f
*tan(e + f*x)) - 2*C*a**3*b**3*c*d*log(tan(e + f*x)**2 + 1)/(2*a**5*b**3*f
+ 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) +
 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) + 4*C*a**3*b**3*c*d/(2*a**5*b**3*f + 2
*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*
a*b**7*f + 2*b**8*f*tan(e + f*x)) + 2*C*a**3*b**3*d**2*f*x/(2*a**5*b**3*f +
 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) +
2*a*b**7*f + 2*b**8*f*tan(e + f*x)) - 8*C*a**3*b**3*d**2*log(a/b + tan(e +
f*x))*tan(e + f*x)/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**
5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) - 2*
C*a**2*b**4*c**2*f*x*tan(e + f*x)/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*
x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan
(e + f*x)) - 4*C*a**2*b**4*c**2*log(a/b + tan(e + f*x))/(2*a**5*b**3*f + 2*
a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a
*b**7*f + 2*b**8*f*tan(e + f*x)) + 2*C*a**2*b**4*c**2*log(tan(e + f*x)**2 +
 1)/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b*
*6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) - 2*C*a**2*b**4*c**
2/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6
*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) - 8*C*a**2*b**4*c*d*f
*x/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**
6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) + 12*C*a**2*b**4*c*d
*log(a/b + tan(e + f*x))*tan(e + f*x)/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e
+ f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f
*tan(e + f*x)) - 2*C*a**2*b**4*c*d*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2
*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*t
an(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) + 2*C*a**2*b**4*d**2*f*x*
tan(e + f*x)/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f +
4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) - 2*C*a**2
*b**4*d**2*log(tan(e + f*x)**2 + 1)/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e +
f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*t
an(e + f*x)) + 4*C*a**2*b**4*d**2*tan(e + f*x)**2/(2*a**5*b**3*f + 2*a**4*b
4*f*tan(e + f*x) + 4*a3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*
f + 2*b**8*f*tan(e + f*x)) - 2*C*a**2*b**4*d**2/(2*a**5*b**3*f + 2*a**4*b**
4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f
+ 2*b**8*f*tan(e + f*x)) + 2*C*a*b**5*c**2*f*x/(2*a**5*b**3*f + 2*a**4*b**4
*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f +
 2*b**8*f*tan(e + f*x)) - 4*C*a*b**5*c**2*log(a/b + tan(e + f*x))*tan(e + f
*x)/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b*
*6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) + 2*C*a*b**5*c**2*l
og(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e +
 f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*
tan(e + f*x)) - 8*C*a*b**5*c*d*f*x*tan(e + f*x)/(2*a**5*b**3*f + 2*a**4*b**

```
4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f
+ 2*b**8*f*tan(e + f*x)) + 2*C*a*b**5*c*d*log(tan(e + f*x)**2 + 1)/(2*a**5*
b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e +
 f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) - 2*C*a*b**5*d**2*f*x/(2*a**5*b
**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e +
f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) - 2*C*a*b**5*d**2*log(tan(e + f*
x)**2 + 1)*tan(e + f*x)/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**
3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x))
 + 2*C*b**6*c**2*f*x*tan(e + f*x)/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*
x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan
(e + f*x)) + 2*C*b**6*c*d*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*a**5*b**
3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*tan(e + f*
x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) - 2*C*b**6*d**2*f*x*tan(e + f*x)/(
2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**2*b**6*f*
tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)) + 2*C*b**6*d**2*tan(e +
f*x)**2/(2*a**5*b**3*f + 2*a**4*b**4*f*tan(e + f*x) + 4*a**3*b**5*f + 4*a**
2*b**6*f*tan(e + f*x) + 2*a*b**7*f + 2*b**8*f*tan(e + f*x)), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.31 (sec) , antiderivative size = 496, normalized size of antiderivative = 1.20

$$\int \frac{(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^2}\,dx$$
$$= \frac{\frac{2\,Cd^2\tan(fx+e)}{b^2}+\frac{2\left(\left(\left((A-C)a^2+2\,Bab-(A-C)b^2\right)c^2-2\left(Ba^2-2\,(A-C)ab-Bb^2\right)cd-\left((A-C)a^2+2\,Bab-(A-C)b^2\right)d^2\right)(fx+e)}{a^4+2\,a^2b^2+b^4}-\frac{2\left(\left(Ba^2\right.\right.}{}}{}$$

```
[In] integrate((c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e
))^2,x, algorithm="maxima")
```

```
[Out] 1/2*(2*C*d^2*tan(f*x + e)/b^2 + 2*(((A - C)*a^2 + 2*B*a*b - (A - C)*b^2)*c^
2 - 2*(B*a^2 - 2*(A - C)*a*b - B*b^2)*c*d - ((A - C)*a^2 + 2*B*a*b - (A - C
)*b^2)*d^2)*(f*x + e)/(a^4 + 2*a^2*b^2 + b^4) - 2*((B*a^2*b^3 - 2*(A - C)*a
*b^4 - B*b^5)*c^2 - 2*(C*a^4*b - (A - 3*C)*a^2*b^3 - 2*B*a*b^4 + A*b^5)*c*d
 + (2*C*a^5 - B*a^4*b + 4*C*a^3*b^2 - 3*B*a^2*b^3 + 2*A*a*b^4)*d^2)*log(b*t
an(f*x + e) + a)/(a^4*b^3 + 2*a^2*b^5 + b^7) + ((B*a^2 - 2*(A - C)*a*b - B*
b^2)*c^2 + 2*((A - C)*a^2 + 2*B*a*b - (A - C)*b^2)*c*d - (B*a^2 - 2*(A - C)
*a*b - B*b^2)*d^2)*log(tan(f*x + e)^2 + 1)/(a^4 + 2*a^2*b^2 + b^4) - 2*((C*
a^2*b^2 - B*a*b^3 + A*b^4)*c^2 - 2*(C*a^3*b - B*a^2*b^2 + A*a*b^3)*c*d + (C
*a^4 - B*a^3*b + A*a^2*b^2)*d^2)/(a^3*b^3 + a*b^5 + (a^2*b^4 + b^6)*tan(f*x
 + e)))/f
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 893 vs. $2(413) = 826$.

Time = 0.84 (sec) , antiderivative size = 893, normalized size of antiderivative = 2.15

$$\int \frac{(c+d\tan(e+fx))^2 (A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx$$

$$= \frac{\frac{2\,Cd^2\tan(fx+e)}{b^2} + \frac{2\,(Aa^2c^2-Ca^2c^2+2\,Babc^2-Ab^2c^2+Cb^2c^2-2\,Ba^2cd+4\,Aabcd-4\,Cabcd+2\,Bb^2cd-Aa^2d^2+Ca^2d^2-2\,Babd^2+Ab^2d^2-Cb^2d^2}{a^4+2\,a^2b^2+b^4}}{}$$

```
[In] integrate((c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e
))^2,x, algorithm="giac")
```

```
[Out] 1/2*(2*C*d^2*tan(f*x + e)/b^2 + 2*(A*a^2*c^2 - C*a^2*c^2 + 2*B*a*b*c^2 - A*
b^2*c^2 + C*b^2*c^2 - 2*B*a^2*c*d + 4*A*a*b*c*d - 4*C*a*b*c*d + 2*B*b^2*c*d
 - A*a^2*d^2 + C*a^2*d^2 - 2*B*a*b*d^2 + A*b^2*d^2 - C*b^2*d^2)*(f*x + e)/(
a^4 + 2*a^2*b^2 + b^4) + (B*a^2*c^2 - 2*A*a*b*c^2 + 2*C*a*b*c^2 - B*b^2*c^2
 + 2*A*a^2*c*d - 2*C*a^2*c*d + 4*B*a*b*c*d - 2*A*b^2*c*d + 2*C*b^2*c*d - B*
a^2*d^2 + 2*A*a*b*d^2 - 2*C*a*b*d^2 + B*b^2*d^2)*log(tan(f*x + e)^2 + 1)/(a
^4 + 2*a^2*b^2 + b^4) - 2*(B*a^2*b^3*c^2 - 2*A*a*b^4*c^2 + 2*C*a*b^4*c^2 -
B*b^5*c^2 - 2*C*a^4*b*c*d + 2*A*a^2*b^3*c*d - 6*C*a^2*b^3*c*d + 4*B*a*b^4*c
*d - 2*A*b^5*c*d + 2*C*a^5*d^2 - B*a^4*b*d^2 + 4*C*a^3*b^2*d^2 - 3*B*a^2*b^
3*d^2 + 2*A*a*b^4*d^2)*log(abs(b*tan(f*x + e) + a))/(a^4*b^3 + 2*a^2*b^5 +
b^7) + 2*(B*a^2*b^4*c^2*tan(f*x + e) - 2*A*a*b^5*c^2*tan(f*x + e) + 2*C*a*b
^5*c^2*tan(f*x + e) - B*b^6*c^2*tan(f*x + e) - 2*C*a^4*b^2*c*d*tan(f*x + e)
 + 2*A*a^2*b^4*c*d*tan(f*x + e) - 6*C*a^2*b^4*c*d*tan(f*x + e) + 4*B*a*b^5*
c*d*tan(f*x + e) - 2*A*b^6*c*d*tan(f*x + e) + 2*C*a^5*b*d^2*tan(f*x + e) -
B*a^4*b^2*d^2*tan(f*x + e) + 4*C*a^3*b^3*d^2*tan(f*x + e) - 3*B*a^2*b^4*d^2
*tan(f*x + e) + 2*A*a*b^5*d^2*tan(f*x + e) - C*a^4*b^2*c^2 + 2*B*a^3*b^3*c^
2 - 3*A*a^2*b^4*c^2 + C*a^2*b^4*c^2 - A*b^6*c^2 - 2*B*a^4*b^2*c*d + 4*A*a^3
*b^3*c*d - 4*C*a^3*b^3*c*d + 2*B*a^2*b^4*c*d + C*a^6*d^2 - A*a^4*b^2*d^2 +
3*C*a^4*b^2*d^2 - 2*B*a^3*b^3*d^2 + A*a^2*b^4*d^2)/((a^4*b^3 + 2*a^2*b^5 +
b^7)*(b*tan(f*x + e) + a)))/f
```

Mupad [B] (verification not implemented)

Time = 32.73 (sec) , antiderivative size = 3958, normalized size of antiderivative = 9.54

$$\int \frac{(c+d\tan(e+fx))^2 (A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx = \text{Too large to display}$$

```
[In] int(((c + d*tan(e + f*x))^2*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a + b
*tan(e + f*x))^2,x)
```

[Out] (log((2*C^2*a^5*d^4 + 4*C^2*a^3*b^2*d^4 - 2*C^2*a^5*c^2*d^2 - A*B*b^5*c^4 -
 2*A*C*a^5*d^4 + B*C*b^5*c^4 - A^2*a*b^4*c^4 - A^2*a*b^4*d^4 + B^2*a*b^4*c^
4 + B^2*a*b^4*d^4 - C^2*a*b^4*c^4 + 2*A^2*b^5*c*d^3 - 2*A^2*b^5*c^3*d + C^2
*a*b^4*d^4 + 2*B^2*b^5*c^3*d - 4*C^2*a^3*b^2*c^2*d^2 + A*B*a^2*b^3*c^4 + 3*
A*B*a^2*b^3*d^4 - 4*A*C*a^3*b^2*d^4 - B*C*a^2*b^3*c^4 + 5*A*B*b^5*c^2*d^2 +
 2*A*C*a^5*c^2*d^2 - 3*B*C*a^2*b^3*d^4 - B*C*b^5*c^2*d^2 + 2*B^2*a^4*b*c*d^
3 - 2*C^2*a^4*b*c*d^3 + 2*C^2*a^4*b*c^3*d + 6*A^2*a*b^4*c^2*d^2 - 2*A^2*a^2
*b^3*c*d^3 + 2*A^2*a^2*b^3*c^3*d - 6*B^2*a*b^4*c^2*d^2 + 6*B^2*a^2*b^3*c*d^
3 - 2*B^2*a^2*b^3*c^3*d + 4*C^2*a*b^4*c^2*d^2 - 6*C^2*a^2*b^3*c*d^3 + 6*C^2
*a^2*b^3*c^3*d + A*B*a^4*b*d^4 + 2*A*C*a*b^4*c^4 - B*C*a^4*b*d^4 - 2*A*C*b^
5*c*d^3 + 2*A*C*b^5*c^3*d - 4*B*C*a^5*c*d^3 - 8*A*B*a*b^4*c*d^3 + 8*A*B*a*b
^4*c^3*d + 2*A*C*a^4*b*c*d^3 - 2*A*C*a^4*b*c^3*d + 4*B*C*a*b^4*c*d^3 - 8*B*
C*a*b^4*c^3*d - A*B*a^4*b*c^2*d^2 - 10*A*C*a*b^4*c^2*d^2 + 8*A*C*a^2*b^3*c*
d^3 - 8*A*C*a^2*b^3*c^3*d - 8*B*C*a^3*b^2*c*d^3 + 5*B*C*a^4*b*c^2*d^2 - 8*A
*B*a^2*b^3*c^2*d^2 + 4*A*C*a^3*b^2*c^2*d^2 + 16*B*C*a^2*b^3*c^2*d^2)/(b^2*(
a^2 + b^2)^2) + ((c*1i + d)^2*((tan(e + f*x)*(3*B*b^5*c^2 - 5*B*b^5*d^2 - 4
*C*a^5*d^2 + 6*A*b^5*c*d - 10*C*b^5*c*d + 4*A*a*b^4*c^2 - 4*A*a*b^4*d^2 + 2
*B*a^4*b*d^2 - 4*C*a*b^4*c^2 + 8*C*a*b^4*d^2 - B*a^2*b^3*c^2 + B*a^2*b^3*d^
2 - 8*B*a*b^4*c*d + 4*C*a^4*b*c*d - 2*A*a^2*b^3*c*d + 2*C*a^2*b^3*c*d))/(b^
2*(a^2 + b^2)) - (A*b^2*d^2 - A*b^2*c^2 - 8*C*a^2*d^2 + C*b^2*c^2 - C*b^2*d
^2 + 4*B*a*b*d^2 + 2*B*b^2*c*d + 8*C*a*b*c*d)/b + (b*(c*1i + d)^2*(4*a*b -
a^2*tan(e + f*x) + 3*b^2*tan(e + f*x))*(A*1i + B - C*1i))/(a*1i + b)^2)*(A*
1i + B - C*1i))/(2*(a*1i + b)^2) + (tan(e + f*x)*(A^2*b^5*c^4 + A^2*b^5*d^4
 + B^2*b^5*d^4 + C^2*b^5*c^4 + C^2*b^5*d^4 + B^2*a^2*b^3*c^4 + 3*B^2*a^2*b^
3*d^4 - 2*A^2*b^5*c^2*d^2 + 3*B^2*b^5*c^2*d^2 + 2*C^2*b^5*c^2*d^2 - 2*A*C*b
^5*c^4 - 2*A*C*b^5*d^4 - 2*B*C*a^5*d^4 + B^2*a^4*b*d^4 - 4*C^2*a^5*c*d^3 +
4*A^2*a^2*b^3*c^2*d^2 - 4*B^2*a^2*b^3*c^2*d^2 + 12*C^2*a^2*b^3*c^2*d^2 - 4*
B*C*a^3*b^2*d^4 + 2*B*C*a^5*c^2*d^2 + 4*A^2*a*b^4*c*d^3 - 4*A^2*a*b^4*c^3*d
 - 4*B^2*a*b^4*c*d^3 + 4*B^2*a*b^4*c^3*d - 4*C^2*a*b^4*c^3*d - B^2*a^4*b*c^
2*d^2 - 8*C^2*a^3*b^2*c*d^3 + 4*C^2*a^4*b*c^2*d^2 - 2*A*B*a*b^4*c^4 - 2*A*B
*a*b^4*d^4 + 2*B*C*a*b^4*c^4 + 2*A*B*b^5*c*d^3 - 4*A*B*b^5*c^3*d + 4*A*C*a^
5*c*d^3 + 2*B*C*b^5*c^3*d - 2*A*B*a^4*b*c*d^3 - 4*A*C*a*b^4*c*d^3 + 8*A*C*a
*b^4*c^3*d + 4*B*C*a^4*b*c*d^3 - 2*B*C*a^4*b*c^3*d + 12*A*B*a*b^4*c^2*d^2 -
 8*A*B*a^2*b^3*c*d^3 + 4*A*B*a^2*b^3*c^3*d + 8*A*C*a^3*b^2*c*d^3 - 4*A*C*a^
4*b*c^2*d^2 - 10*B*C*a*b^4*c^2*d^2 + 12*B*C*a^2*b^3*c*d^3 - 8*B*C*a^2*b^3*c
^3*d - 16*A*C*a^2*b^3*c^2*d^2 + 4*B*C*a^3*b^2*c^2*d^2))/(b^2*(a^2 + b^2)^2)
)*(A*d^2*1i - A*c^2*1i - B*c^2 + B*d^2 + C*c^2*1i - C*d^2*1i - 2*A*c*d + B*
c*d*2i + 2*C*c*d))/(2*f*(a*b*2i - a^2 + b^2)) - (log(a + b*tan(e + f*x))*(b
^3*(B*a^2*c^2 - 3*B*a^2*d^2 + 2*A*a^2*c*d - 6*C*a^2*c*d) - b^5*(B*c^2 + 2*A
*c*d) - b*(B*a^4*d^2 + 2*C*a^4*c*d) + b^4*(2*A*a*d^2 - 2*A*a*c^2 + 2*C*a*c^
2 + 4*B*a*c*d) + 2*C*a^5*d^2 + 4*C*a^3*b^2*d^2))/(f*(b^7 + 2*a^2*b^5 + a^4*
b^3)) + (log((2*C^2*a^5*d^4 + 4*C^2*a^3*b^2*d^4 - 2*C^2*a^5*c^2*d^2 - A*B*b
^5*c^4 - 2*A*C*a^5*d^4 + B*C*b^5*c^4 - A^2*a*b^4*c^4 - A^2*a*b^4*d^4 + B^2*
a*b^4*c^4 + B^2*a*b^4*d^4 - C^2*a*b^4*c^4 + 2*A^2*b^5*c*d^3 - 2*A^2*b^5*c^3
*d + C^2*a*b^4*d^4 + 2*B^2*b^5*c^3*d - 4*C^2*a^3*b^2*c^2*d^2 + A*B*a^2*b^3*

c^4 + 3*A*B*a^2*b^3*d^4 - 4*A*C*a^3*b^2*d^4 - B*C*a^2*b^3*c^4 + 5*A*B*b^5*c
^2*d^2 + 2*A*C*a^5*c^2*d^2 - 3*B*C*a^2*b^3*d^4 - B*C*b^5*c^2*d^2 + 2*B^2*a^
4*b*c*d^3 - 2*C^2*a^4*b*c*d^3 + 2*C^2*a^4*b*c^3*d + 6*A^2*a*b^4*c^2*d^2 - 2
*A^2*a^2*b^3*c*d^3 + 2*A^2*a^2*b^3*c^3*d - 6*B^2*a*b^4*c^2*d^2 + 6*B^2*a^2*
b^3*c*d^3 - 2*B^2*a^2*b^3*c^3*d + 4*C^2*a*b^4*c^2*d^2 - 6*C^2*a^2*b^3*c*d^3
 + 6*C^2*a^2*b^3*c^3*d + A*B*a^4*b*d^4 + 2*A*C*a*b^4*c^4 - B*C*a^4*b*d^4 -
2*A*C*b^5*c*d^3 + 2*A*C*b^5*c^3*d - 4*B*C*a^5*c*d^3 - 8*A*B*a*b^4*c*d^3 + 8
*A*B*a*b^4*c^3*d + 2*A*C*a^4*b*c*d^3 - 2*A*C*a^4*b*c^3*d + 4*B*C*a*b^4*c*d^
3 - 8*B*C*a*b^4*c^3*d - A*B*a^4*b*c^2*d^2 - 10*A*C*a*b^4*c^2*d^2 + 8*A*C*a^
2*b^3*c*d^3 - 8*A*C*a^2*b^3*c^3*d - 8*B*C*a^3*b^2*c*d^3 + 5*B*C*a^4*b*c^2*d
^2 - 8*A*B*a^2*b^3*c^2*d^2 + 4*A*C*a^3*b^2*c^2*d^2 + 16*B*C*a^2*b^3*c^2*d^2
)/(b^2*(a^2 + b^2)^2) + ((c*1i - d)^2*((A*b^2*d^2 - A*b^2*c^2 - 8*C*a^2*d^2
 + C*b^2*c^2 - C*b^2*d^2 + 4*B*a*b*d^2 + 2*B*b^2*c*d + 8*C*a*b*c*d)/b - (ta
n(e + f*x)*(3*B*b^5*c^2 - 5*B*b^5*d^2 - 4*C*a^5*d^2 + 6*A*b^5*c*d - 10*C*b^
5*c*d + 4*A*a*b^4*c^2 - 4*A*a*b^4*d^2 + 2*B*a^4*b*d^2 - 4*C*a*b^4*c^2 + 8*C
*a*b^4*d^2 - B*a^2*b^3*c^2 + B*a^2*b^3*d^2 - 8*B*a*b^4*c*d + 4*C*a^4*b*c*d
 - 2*A*a^2*b^3*c*d + 2*C*a^2*b^3*c*d))/(b^2*(a^2 + b^2)) + (b*(c*1i - d)^2*(
4*a*b - a^2*tan(e + f*x) + 3*b^2*tan(e + f*x))*(A + B*1i - C)*1i)/(a*1i - b
)^2)*(A + B*1i - C)*1i)/(2*(a*1i - b)^2) + (tan(e + f*x)*(A^2*b^5*c^4 + A^2
*b^5*d^4 + B^2*b^5*d^4 + C^2*b^5*c^4 + C^2*b^5*d^4 + B^2*a^2*b^3*c^4 + 3*B^
2*a^2*b^3*d^4 - 2*A^2*b^5*c^2*d^2 + 3*B^2*b^5*c^2*d^2 + 2*C^2*b^5*c^2*d^2 -
 2*A*C*b^5*c^4 - 2*A*C*b^5*d^4 - 2*B*C*a^5*d^4 + B^2*a^4*b*d^4 - 4*C^2*a^5*
c*d^3 + 4*A^2*a^2*b^3*c^2*d^2 - 4*B^2*a^2*b^3*c^2*d^2 + 12*C^2*a^2*b^3*c^2*
d^2 - 4*B*C*a^3*b^2*d^4 + 2*B*C*a^5*c^2*d^2 + 4*A^2*a*b^4*c*d^3 - 4*A^2*a*b
^4*c^3*d - 4*B^2*a*b^4*c*d^3 + 4*B^2*a*b^4*c^3*d - 4*C^2*a*b^4*c^3*d - B^2*
a^4*b*c^2*d^2 - 8*C^2*a^3*b^2*c*d^3 + 4*C^2*a^4*b*c^2*d^2 - 2*A*B*a*b^4*c^4
 - 2*A*B*a*b^4*d^4 + 2*B*C*a*b^4*c^4 + 2*A*B*b^5*c*d^3 - 4*A*B*b^5*c^3*d +
4*A*C*a^5*c*d^3 + 2*B*C*b^5*c^3*d - 2*A*B*a^4*b*c*d^3 - 4*A*C*a*b^4*c*d^3 +
 8*A*C*a*b^4*c^3*d + 4*B*C*a^4*b*c*d^3 - 2*B*C*a^4*b*c^3*d + 12*A*B*a*b^4*c
^2*d^2 - 8*A*B*a^2*b^3*c*d^3 + 4*A*B*a^2*b^3*c^3*d + 8*A*C*a^3*b^2*c*d^3 -
4*A*C*a^4*b*c^2*d^2 - 10*B*C*a*b^4*c^2*d^2 + 12*B*C*a^2*b^3*c*d^3 - 8*B*C*a
^2*b^3*c^3*d - 16*A*C*a^2*b^3*c^2*d^2 + 4*B*C*a^3*b^2*c^2*d^2))/(b^2*(a^2 +
 b^2)^2))*(A*d^2 - A*c^2 - B*c^2*1i + B*d^2*1i + C*c^2 - C*d^2 - A*c*d*2i +
 2*B*c*d + C*c*d*2i))/(2*f*(2*a*b - a^2*1i + b^2*1i)) + (C*d^2*tan(e + f*x)
)/(b^2*f) - (A*b^4*c^2 + C*a^4*d^2 - B*a*b^3*c^2 - B*a^3*b*d^2 + A*a^2*b^2*
d^2 + C*a^2*b^2*c^2 - 2*A*a*b^3*c*d - 2*C*a^3*b*c*d + 2*B*a^2*b^2*c*d)/(b*f
*(a*b^2 + b^3*tan(e + f*x))*(a^2 + b^2))

3.63 $\int \frac{(c+d\tan(e+fx))^2(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx$

Optimal result . . . 565
Rubi [A] (verified) . . . 566
Mathematica [C] (verified) . . . 569
Maple [A] (verified) . . . 570
Fricas [B] (verification not implemented) . . . 571
Sympy [F(-2)] . . . 572
Maxima [A] (verification not implemented) . . . 572
Giac [B] (verification not implemented) . . . 573
Mupad [B] (verification not implemented) . . . 574

Optimal result

Integrand size = 45, antiderivative size = 597

$$\int \frac{(c+d\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx =$$
$$-\frac{(a^3(c^2C+2Bcd-Cd^2-A(c^2-d^2))-3ab^2(c^2C+2Bcd-Cd^2-A(c^2-d^2))-3a^2b(2c(A-C)d+}{(a^2+b^2)^3}$$
$$-\frac{(3a^2b(c^2C+2Bcd-Cd^2-A(c^2-d^2))-b^3(c^2C+2Bcd-Cd^2-A(c^2-d^2))+a^3(2c(A-C)d+B(}{(a^2+b^2)^3\,f}$$
$$+\frac{(a^6Cd^2+3a^4b^2Cd^2-3a^2b^4(c^2C+2Bcd-2Cd^2-A(c^2-d^2))+b^6(c(cC+2Bd)-A(c^2-d^2))-a^3b}{b^3\,(a^2+b^2)^3\,f}$$
$$-\frac{(bc-ad)\,(a^4Cd+b^4(Bc+Ad)+2ab^3(Ac-cC-Bd)-a^2b^2(Bc+(A-3C)d))}{b^3\,(a^2+b^2)^2\,f(a+b\tan(e+fx))}$$
$$-\frac{(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^2}{2b\,(a^2+b^2)\,f(a+b\tan(e+fx))^2}$$

```
[Out] -(a^3*(c^2*C+2*B*c*d-C*d^2-A*(c^2-d^2))-3*a*b^2*(c^2*C+2*B*c*d-C*d^2-A*(c^2
-d^2))-3*a^2*b*(2*c*(A-C)*d+B*(c^2-d^2))+b^3*(2*c*(A-C)*d+B*(c^2-d^2)))*x/(
a^2+b^2)^3-(3*a^2*b*(c^2*C+2*B*c*d-C*d^2-A*(c^2-d^2))-b^3*(c^2*C+2*B*c*d-C*
d^2-A*(c^2-d^2))+a^3*(2*c*(A-C)*d+B*(c^2-d^2))-3*a*b^2*(2*c*(A-C)*d+B*(c^2-
d^2)))*ln(cos(f*x+e))/(a^2+b^2)^3/f+(a^6*C*d^2+3*a^4*b^2*C*d^2-3*a^2*b^4*(c
^2*C+2*B*c*d-2*C*d^2-A*(c^2-d^2))+b^6*(c*(2*B*d+C*c)-A*(c^2-d^2))-a^3*b^3*(
2*c*(A-C)*d+B*(c^2-d^2))+3*a*b^5*(2*c*(A-C)*d+B*(c^2-d^2)))*ln(a+b*tan(f*x+
e))/b^3/(a^2+b^2)^3/f-(-a*d+b*c)*(a^4*C*d+b^4*(A*d+B*c)+2*a*b^3*(A*c-B*d-C*
c)-a^2*b^2*(B*c+(A-3*C)*d))/b^3/(a^2+b^2)^2/f/(a+b*tan(f*x+e))-1/2*(A*b^2-a
*(B*b-C*a))*(c+d*tan(f*x+e))^2/b/(a^2+b^2)/f/(a+b*tan(f*x+e))^2
```

Rubi [A] (verified)

Time = 1.45 (sec) , antiderivative size = 597, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}} = 0.133$, Rules used = {3726, 3716, 3707, 3698, 31, 3556}

$$\int \frac{(c+d\tan(e+fx))^2(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx$$
$$= -\frac{(Ab^2-a(bB-aC))(c+d\tan(e+fx))^2}{2bf(a^2+b^2)(a+b\tan(e+fx))^2}$$
$$-\frac{(bc-ad)(a^4Cd-a^2b^2(d(A-3C)+Bc)+2ab^3(Ac-Bd-cC)+b^4(Ad+Bc))}{b^3f(a^2+b^2)^2(a+b\tan(e+fx))}$$
$$-\frac{\log(\cos(e+fx))(a^3(2cd(A-C)+B(c^2-d^2))+3a^2b(-A(c^2-d^2)+2Bcd+c^2C-Cd^2)-3ab^2(2cd(\ldots}{f(a^2+b^2)^3}$$
$$-\frac{x(a^3(-A(c^2-d^2)+2Bcd+c^2C-Cd^2)-3a^2b(2cd(A-C)+B(c^2-d^2))-3ab^2(-A(c^2-d^2)+2Bc\ldots}{(a^2+b^2)^3}$$
$$+\frac{(a^6Cd^2+3a^4b^2Cd^2-a^3b^3(2cd(A-C)+B(c^2-d^2))-3a^2b^4(-A(c^2-d^2)+2Bcd+c^2C-2Cd^2)+3\ldots}{b^3f(a^2+b^2)^3}$$

```
[In] Int[((c + d*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a + b
*Tan[e + f*x])^3,x]
```

```
[Out] -(((a^3*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - 3*a*b^2*(c^2*C + 2*B*c*
d - C*d^2 - A*(c^2 - d^2)) - 3*a^2*b*(2*c*(A - C)*d + B*(c^2 - d^2)) + b^3*
(2*c*(A - C)*d + B*(c^2 - d^2)))*x)/(a^2 + b^2)^3) - ((3*a^2*b*(c^2*C + 2*B
*c*d - C*d^2 - A*(c^2 - d^2)) - b^3*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2
)) + a^3*(2*c*(A - C)*d + B*(c^2 - d^2)) - 3*a*b^2*(2*c*(A - C)*d + B*(c^2
- d^2)))*Log[Cos[e + f*x]])/((a^2 + b^2)^3*f) + ((a^6*C*d^2 + 3*a^4*b^2*C*d
^2 - 3*a^2*b^4*(c^2*C + 2*B*c*d - 2*C*d^2 - A*(c^2 - d^2)) + b^6*(c*(c*C +
2*B*d) - A*(c^2 - d^2)) - a^3*b^3*(2*c*(A - C)*d + B*(c^2 - d^2)) + 3*a*b^5
*(2*c*(A - C)*d + B*(c^2 - d^2)))*Log[a + b*Tan[e + f*x]])/(b^3*(a^2 + b^2)
^3*f) - ((b*c - a*d)*(a^4*C*d + b^4*(B*c + A*d) + 2*a*b^3*(A*c - c*C - B*d)
 - a^2*b^2*(B*c + (A - 3*C)*d)))/(b^3*(a^2 + b^2)^2*f*(a + b*Tan[e + f*x]))
 - ((A*b^2 - a*(b*B - a*C))*(c + d*Tan[e + f*x])^2)/(2*b*(a^2 + b^2)*f*(a +
 b*Tan[e + f*x])^2)
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3698

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3707

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[(a*A + b*B - a
*C)*(x/(a^2 + b^2)), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3716

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.
)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(-(b*c - a*d))*(c^2*C - B*c*d + A*d^2)*((c
+ d*Tan[e + f*x])^(n + 1)/(d^2*f*(n + 1)*(c^2 + d^2))), x] + Dist[1/(d*(c^2
 + d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)*Simp[a*d*(A*c - c*C + B*d) + b*(
c^2*C - B*c*d + A*d^2) + d*(A*b*c + a*B*c - b*c*C - a*A*d + b*B*d + a*C*d)*
Tan[e + f*x] + b*C*(c^2 + d^2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c
, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] && LtQ[n,
 -1]
```

Rule 3726

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*d^2 + c*(c*C - B*d))*(a + b*Tan[e +
 f*x])^m*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rubi steps

$$\text{integral} = -\frac{(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^2}{2b\,(a^2+b^2)\,f(a+b\tan(e+fx))^2} + \frac{\int \frac{(c+d\tan(e+fx))(2((bB-aC)(bc-ad)+Ab(ac+bd))-2b((A-C)(bc-ad)-B(ac+bd))\tan(e+fx)+2(a^2+b^2)Cd\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx}{2b\,(a^2+b^2)}$$

$$= -\frac{(bc-ad)\,(a^4Cd+b^4(Bc+Ad)+2ab^3(Ac-cC-Bd)-a^2b^2(Bc+(A-3C)d))}{b^3\,(a^2+b^2)^2\,f(a+b\tan(e+fx))} - \frac{(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^2}{2b\,(a^2+b^2)\,f(a+b\tan(e+fx))^2} + \frac{\int \frac{2(a^4Cd^2-a^2b^2(c^2C+2Bcd-3Cd^2-A(c^2-d^2))+b^4(c(cC+2Bd)-A(c^2-d^2))+2ab^3(2c(A-C)d+B(c^2-d^2)))+2b^2(2ab(c^2C+2B}{a+b\tan(e+}}{2b^2\,(a^2+}$$

$$= -\frac{(a^3(c^2C+2Bcd-Cd^2-A(c^2-d^2))-3ab^2(c^2C+2Bcd-Cd^2-A(c^2-d^2))-3a^2b(2c(A-C}{(a^2+b^2)^3} - \frac{(bc-ad)\,(a^4Cd+b^4(Bc+Ad)+2ab^3(Ac-cC-Bd)-a^2b^2(Bc+(A-3C)d))}{b^3\,(a^2+b^2)^2\,f(a+b\tan(e+fx))} - \frac{(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^2}{2b\,(a^2+b^2)\,f(a+b\tan(e+fx))^2} + \frac{(3a^2b(c^2C+2Bcd-Cd^2-A(c^2-d^2))-b^3(c^2C+2Bcd-Cd^2-A(c^2-d^2))+a^3(2c(A-C)}{(a^2+b^2)^3} + \frac{(a^6Cd^2+3a^4b^2Cd^2-3a^2b^4(c^2C+2Bcd-2Cd^2-A(c^2-d^2))+b^6(c(cC+2Bd)-A(c^2-d^2))}{b^2\,(a^2+b^2}$$

$$= -\frac{(a^3(c^2C+2Bcd-Cd^2-A(c^2-d^2))-3ab^2(c^2C+2Bcd-Cd^2-A(c^2-d^2))-3a^2b(2c(A-C}{(a^2+b^2)^3} - \frac{(3a^2b(c^2C+2Bcd-Cd^2-A(c^2-d^2))-b^3(c^2C+2Bcd-Cd^2-A(c^2-d^2))+a^3(2c(A-C)}{(a^2+b^2)^3\,f} - \frac{(bc-ad)\,(a^4Cd+b^4(Bc+Ad)+2ab^3(Ac-cC-Bd)-a^2b^2(Bc+(A-3C)d))}{b^3\,(a^2+b^2)^2\,f(a+b\tan(e+fx))} - \frac{(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^2}{2b\,(a^2+b^2)\,f(a+b\tan(e+fx))^2} + \frac{(a^6Cd^2+3a^4b^2Cd^2-3a^2b^4(c^2C+2Bcd-2Cd^2-A(c^2-d^2))+b^6(c(cC+2Bd)-A(c^2-d^2))}{b^3\,(}$$

$$= -\frac{(a^3(c^2C + 2Bcd - Cd^2 - A(c^2 - d^2)) - 3ab^2(c^2C + 2Bcd - Cd^2 - A(c^2 - d^2)) - 3a^2b(2c(A -}{(a^2+b^2)^3}$$
$$-\frac{(3a^2b(c^2C + 2Bcd - Cd^2 - A(c^2 - d^2)) - b^3(c^2C + 2Bcd - Cd^2 - A(c^2 - d^2)) + a^3(2c(A - C}{(a^2+b^2)^3 f}$$
$$+\frac{(a^6Cd^2 + 3a^4b^2Cd^2 - 3a^2b^4(c^2C + 2Bcd - 2Cd^2 - A(c^2 - d^2)) + b^6(c(cC + 2Bd) - A(c^2 - d^2}{b^3(a^2 +}$$
$$-\frac{(bc-ad)(a^4Cd + b^4(Bc + Ad) + 2ab^3(Ac - cC - Bd) - a^2b^2(Bc + (A - 3C)d))}{b^3(a^2+b^2)^2 f(a + b\tan(e + fx))}$$
$$-\frac{(Ab^2 - a(bB - aC))(c + d\tan(e + fx))^2}{2b(a^2+b^2) f(a + b\tan(e + fx))^2}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 7.04 (sec) , antiderivative size = 1041, normalized size of antiderivative = 1.74

$$\int \frac{(c + d\tan(e + fx))^2 (A + B\tan(e + fx) + C\tan^2(e + fx))}{(a + b\tan(e + fx))^3} dx =$$
$$-\frac{(3a^2Abc^2 - Ab^3c^2 - a^3Bc^2 + 3ab^2Bc^2 - 3a^2bc^2C + b^3c^2C - 2a^3Acd + 6aAb^2cd - 6a^2bBcd + 2b^3Bcd +}{}$$
$$+\frac{(-3a^2Abc^2 + Ab^3c^2 + a^3Bc^2 - 3ab^2Bc^2 + 3a^2bc^2C - b^3c^2C + 2a^3Acd - 6aAb^2cd + 6a^2bBcd - 2b^3Bcd}{}$$
$$+\frac{(a^6Cd^2 + 3a^4b^2Cd^2 - 3a^2b^4(c^2C + 2Bcd - 2Cd^2 - A(c^2 - d^2)) + b^6(c(cC + 2Bd) - A(c^2 - d^2)) - a^3b}{b^3(a^2+b^2)^3 f}$$
$$-\frac{(Ab^2 - a(bB - aC))(bc - ad)^2}{2b^3(a^2+b^2) f(a + b\tan(e + fx))^2}$$
$$+\frac{(bc-ad)(a^3bBd - 2a^4Cd - b^4(Bc + 2Ad) - ab^3(2Ac - 2cC - 3Bd) + a^2b^2(Bc - 4Cd))}{b^3(a^2+b^2)^2 f(a + b\tan(e + fx))}$$

```
[In] Integrate[((c + d*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(a + b*Tan[e + f*x])^3,x]
```

```
[Out] -1/2*((3*a^2*A*b*c^2 - A*b^3*c^2 - a^3*B*c^2 + 3*a*b^2*B*c^2 - 3*a^2*b*c^2*
C + b^3*c^2*C - 2*a^3*A*c*d + 6*a*A*b^2*c*d - 6*a^2*b*B*c*d + 2*b^3*B*c*d +
 2*a^3*c*C*d - 6*a*b^2*c*C*d - 3*a^2*A*b*d^2 + A*b^3*d^2 + a^3*B*d^2 - 3*a*
b^2*B*d^2 + 3*a^2*b*C*d^2 - b^3*C*d^2 + I*(a^3*A*c^2 - 3*a*A*b^2*c^2 + 3*a^
2*b*B*c^2 - b^3*B*c^2 - a^3*c^2*C + 3*a*b^2*c^2*C + 6*a^2*A*b*c*d - 2*A*b^3
*c*d - 2*a^3*B*c*d + 6*a*b^2*B*c*d - 6*a^2*b*c*C*d + 2*b^3*c*C*d - a^3*A*d^
2 + 3*a*A*b^2*d^2 - 3*a^2*b*B*d^2 + b^3*B*d^2 + a^3*C*d^2 - 3*a*b^2*C*d^2))
*Log[I - Tan[e + f*x]])/((a^2 + b^2)^3*f) + ((-3*a^2*A*b*c^2 + A*b^3*c^2 +
a^3*B*c^2 - 3*a*b^2*B*c^2 + 3*a^2*b*c^2*C - b^3*c^2*C + 2*a^3*A*c*d - 6*a*A
```

*b^2*c*d + 6*a^2*b*B*c*d - 2*b^3*B*c*d - 2*a^3*c*C*d + 6*a*b^2*c*C*d + 3*a^2*A*b*d^2 - A*b^3*d^2 - a^3*B*d^2 + 3*a*b^2*B*d^2 - 3*a^2*b*C*d^2 + b^3*C*d^2 + I*(a^3*A*c^2 - 3*a*A*b^2*c^2 + 3*a^2*b*B*c^2 - b^3*B*c^2 - a^3*c^2*C + 3*a*b^2*c^2*C + 6*a^2*A*b*c*d - 2*A*b^3*c*d - 2*a^3*B*c*d + 6*a*b^2*B*c*d - 6*a^2*b*c*C*d + 2*b^3*c*C*d - a^3*A*d^2 + 3*a*A*b^2*d^2 - 3*a^2*b*B*d^2 + b^3*B*d^2 + a^3*C*d^2 - 3*a*b^2*C*d^2))*Log[I + Tan[e + f*x]])/(2*(a^2 + b^2)^3*f) + ((a^6*C*d^2 + 3*a^4*b^2*C*d^2 - 3*a^2*b^4*(c^2*C + 2*B*c*d - 2*C*d^2 - A*(c^2 - d^2)) + b^6*(c*(c*C + 2*B*d) - A*(c^2 - d^2)) - a^3*b^3*(2*c*(A - C)*d + B*(c^2 - d^2)) + 3*a*b^5*(2*c*(A - C)*d + B*(c^2 - d^2)))*Log[a + b*Tan[e + f*x]])/(b^3*(a^2 + b^2)^3*f) - ((A*b^2 - a*(b*B - a*C))*(b*c - a*d)^2)/(2*b^3*(a^2 + b^2)*f*(a + b*Tan[e + f*x])^2) + ((b*c - a*d)*(a^3*b*B*d - 2*a^4*C*d - b^4*(B*c + 2*A*d) - a*b^3*(2*A*c - 2*c*C - 3*B*d) + a^2*b^2*(B*c - 4*C*d)))/(b^3*(a^2 + b^2)^2*f*(a + b*Tan[e + f*x]))

Maple [A] (verified)

Time = 0.44 (sec) , antiderivative size = 865, normalized size of antiderivative = 1.45

method	result
derivativedivides	$\frac{-\frac{A a^2 d^2 b^2-2 A a b^3 c d+A b^4 c^2-B a^3 d^2 b+2 B a^2 c d b^2-B a b^3 c^2+a^4 C d^2-2 C a^3 c d b+C a^2 c^2 b^2}{2 b^3 \left(a^2+b^2\right)(a+b \tan (f x+e))^2}-\frac{-2 A a^2 b^3 c d+2 A a b^4 c^2-2 A a b^4 d^2+2 A}{}}{}$
default	$\frac{-\frac{A a^2 d^2 b^2-2 A a b^3 c d+A b^4 c^2-B a^3 d^2 b+2 B a^2 c d b^2-B a b^3 c^2+a^4 C d^2-2 C a^3 c d b+C a^2 c^2 b^2}{2 b^3 \left(a^2+b^2\right)(a+b \tan (f x+e))^2}-\frac{-2 A a^2 b^3 c d+2 A a b^4 c^2-2 A a b^4 d^2+2 A}{}}{}$
norman	Expression too large to display
risch	Expression too large to display
parallelrisch	Expression too large to display

[In] int((c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^3,x,method=_RETURNVERBOSE)

[Out] 1/f*(-1/2*(A*a^2*b^2*d^2-2*A*a*b^3*c*d+A*b^4*c^2-B*a^3*b*d^2+2*B*a^2*b^2*c*d-B*a*b^3*c^2+C*a^4*d^2-2*C*a^3*b*c*d+C*a^2*b^2*c^2)/b^3/(a^2+b^2)/(a+b*tan(f*x+e))^2-(-2*A*a^2*b^3*c*d+2*A*a*b^4*c^2-2*A*a*b^4*d^2+2*A*b^5*c*d+B*a^4*b*d^2-B*a^2*b^3*c^2+3*B*a^2*b^3*d^2-4*B*a*b^4*c*d+B*b^5*c^2-2*C*a^5*d^2+2*C*a^4*b*c*d-4*C*a^3*b^2*d^2+6*C*a^2*b^3*c*d-2*C*a*b^4*c^2)/b^3/(a^2+b^2)^2/(a+b*tan(f*x+e))+1/(a^2+b^2)^3*(-2*A*a^3*b^3*c*d+3*A*a^2*b^4*c^2-3*A*a^2*b^4*d^2+6*A*a*b^5*c*d-A*b^6*c^2+A*b^6*d^2-B*a^3*b^3*c^2+B*a^3*b^3*d^2-6*B*a^2*b^4*c*d+3*B*a*b^5*c^2-3*B*a*b^5*d^2+2*B*b^6*c*d+C*a^6*d^2+3*C*a^4*b^2*d^2+2*C*a^3*b^3*c*d-3*C*a^2*b^4*c^2+6*C*a^2*b^4*d^2-6*C*a*b^5*c*d+C*b^6*c^2)/b^3*ln(a+b*tan(f*x+e))+1/(a^2+b^2)^3*(1/2*(2*A*a^3*c*d-3*A*a^2*b*c^2+3*A*a^2*b*d^2-6*A*a*b^2*c*d+A*b^3*c^2-A*b^3*d^2+B*a^3*c^2-B*a^3*d^2+6*B*a^2*b*c*d-3*B*a*b^2*c^2+3*B*a*b^2*d^2-2*B*b^3*c*d-2*C*a^3*c*d+3*C*a^2*b*c^2-3*C*a^2*b*d^2+6*C*a*b^2*c*d-C*b^3*c^2+C*b^3*d^2)*ln(1+tan(f*x+e)^2)+(A*a^3*c^2-A*a^3*d

```
^2+6*A*a^2*b*c*d-3*A*a*b^2*c^2+3*A*a*b^2*d^2-2*A*b^3*c*d-2*B*a^3*c*d+3*B*a^
2*b*c^2-3*B*a^2*b*d^2+6*B*a*b^2*c*d-B*b^3*c^2+B*b^3*d^2-C*a^3*c^2+C*a^3*d^2
-6*C*a^2*b*c*d+3*C*a*b^2*c^2-3*C*a*b^2*d^2+2*C*b^3*c*d)*arctan(tan(f*x+e)))
)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1699 vs. $2(595) = 1190$.

Time = 0.74 (sec) , antiderivative size = 1699, normalized size of antiderivative = 2.85

$$\int \frac{(c+d\tan(e+fx))^2 \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^3}\,dx = \text{Too large to display}$$

```
[In] integrate((c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e
))^3,x, algorithm="fricas")
```

```
[Out] -1/2*((3*C*a^4*b^4 - 5*B*a^3*b^5 + (7*A - 3*C)*a^2*b^6 + B*a*b^7 + A*b^8)*c
^2 - 2*(C*a^5*b^3 - 3*B*a^4*b^4 + 5*(A - C)*a^3*b^5 + 3*B*a^2*b^6 - A*a*b^7
)*c*d - (C*a^6*b^2 + B*a^5*b^3 - (3*A - 7*C)*a^4*b^4 - 5*B*a^3*b^5 + 3*A*a^
2*b^6)*d^2 - 2*(((A - C)*a^5*b^3 + 3*B*a^4*b^4 - 3*(A - C)*a^3*b^5 - B*a^2*
b^6)*c^2 - 2*(B*a^5*b^3 - 3*(A - C)*a^4*b^4 - 3*B*a^3*b^5 + (A - C)*a^2*b^6
)*c*d - ((A - C)*a^5*b^3 + 3*B*a^4*b^4 - 3*(A - C)*a^3*b^5 - B*a^2*b^6)*d^2
)*f*x - ((C*a^4*b^4 - 3*B*a^3*b^5 + 5*(A - C)*a^2*b^6 + 3*B*a*b^7 - A*b^8)*
c^2 + 2*(C*a^5*b^3 + B*a^4*b^4 - (3*A - 7*C)*a^3*b^5 - 5*B*a^2*b^6 + 3*A*a*
b^7)*c*d - (3*C*a^6*b^2 - B*a^5*b^3 - (A - 9*C)*a^4*b^4 - 7*B*a^3*b^5 + 5*A
*a^2*b^6)*d^2 + 2*(((A - C)*a^3*b^5 + 3*B*a^2*b^6 - 3*(A - C)*a*b^7 - B*b^8
)*c^2 - 2*(B*a^3*b^5 - 3*(A - C)*a^2*b^6 - 3*B*a*b^7 + (A - C)*b^8)*c*d - (
(A - C)*a^3*b^5 + 3*B*a^2*b^6 - 3*(A - C)*a*b^7 - B*b^8)*d^2)*f*x)*tan(f*x
+ e)^2 + ((B*a^5*b^3 - 3*(A - C)*a^4*b^4 - 3*B*a^3*b^5 + (A - C)*a^2*b^6)*c
^2 + 2*((A - C)*a^5*b^3 + 3*B*a^4*b^4 - 3*(A - C)*a^3*b^5 - B*a^2*b^6)*c*d
- (C*a^8 + 3*C*a^6*b^2 + B*a^5*b^3 - 3*(A - 2*C)*a^4*b^4 - 3*B*a^3*b^5 + A*
a^2*b^6)*d^2 + ((B*a^3*b^5 - 3*(A - C)*a^2*b^6 - 3*B*a*b^7 + (A - C)*b^8)*c
^2 + 2*((A - C)*a^3*b^5 + 3*B*a^2*b^6 - 3*(A - C)*a*b^7 - B*b^8)*c*d - (C*a
^6*b^2 + 3*C*a^4*b^4 + B*a^3*b^5 - 3*(A - 2*C)*a^2*b^6 - 3*B*a*b^7 + A*b^8)
*d^2)*tan(f*x + e)^2 + 2*((B*a^4*b^4 - 3*(A - C)*a^3*b^5 - 3*B*a^2*b^6 + (A
 - C)*a*b^7)*c^2 + 2*((A - C)*a^4*b^4 + 3*B*a^3*b^5 - 3*(A - C)*a^2*b^6 - B
*a*b^7)*c*d - (C*a^7*b + 3*C*a^5*b^3 + B*a^4*b^4 - 3*(A - 2*C)*a^3*b^5 - 3*
B*a^2*b^6 + A*a*b^7)*d^2)*tan(f*x + e))*log((b^2*tan(f*x + e)^2 + 2*a*b*tan
(f*x + e) + a^2)/(tan(f*x + e)^2 + 1)) + ((C*a^6*b^2 + 3*C*a^4*b^4 + 3*C*a^
2*b^6 + C*b^8)*d^2*tan(f*x + e)^2 + 2*(C*a^7*b + 3*C*a^5*b^3 + 3*C*a^3*b^5
+ C*a*b^7)*d^2*tan(f*x + e) + (C*a^8 + 3*C*a^6*b^2 + 3*C*a^4*b^4 + C*a^2*b^
6)*d^2)*log(1/(tan(f*x + e)^2 + 1)) - 2*((C*a^5*b^3 - 2*B*a^4*b^4 + 3*(A -
C)*a^3*b^5 + 3*B*a^2*b^6 - (3*A - 2*C)*a*b^7 - B*b^8)*c^2 + 2*(B*a^5*b^3 -
(2*A - 3*C)*a^4*b^4 - 3*B*a^3*b^5 + 3*(A - C)*a^2*b^6 + 2*B*a*b^7 - A*b^8)*
c*d - (C*a^7*b - (A - 3*C)*a^5*b^3 - 3*B*a^4*b^4 + (3*A - 4*C)*a^3*b^5 + 3*
```

```
B*a^2*b^6 - 2*A*a*b^7)*d^2 + 2*(((A - C)*a^4*b^4 + 3*B*a^3*b^5 - 3*(A - C)*
a^2*b^6 - B*a*b^7)*c^2 - 2*(B*a^4*b^4 - 3*(A - C)*a^3*b^5 - 3*B*a^2*b^6 + (
A - C)*a*b^7)*c*d - ((A - C)*a^4*b^4 + 3*B*a^3*b^5 - 3*(A - C)*a^2*b^6 - B*
a*b^7)*d^2)*f*x)*tan(f*x + e))/((a^6*b^5 + 3*a^4*b^7 + 3*a^2*b^9 + b^11)*f*
tan(f*x + e)^2 + 2*(a^7*b^4 + 3*a^5*b^6 + 3*a^3*b^8 + a*b^10)*f*tan(f*x + e
) + (a^8*b^3 + 3*a^6*b^5 + 3*a^4*b^7 + a^2*b^9)*f)
```

Sympy [F(-2)]

Exception generated.

$$\int \frac{(c+d\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx = \text{Exception raised: AttributeError}$$

```
[In] integrate((c+d*tan(f*x+e))**2*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x
+e))**3,x)
```

```
[Out] Exception raised: AttributeError >> 'NoneType' object has no attribute 'pri
mitive'
```

Maxima [A] (verification not implemented)

none

Time = 0.33 (sec) , antiderivative size = 839, normalized size of antiderivative = 1.41

$$\int \frac{(c+d\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx = \frac{\frac{2\,\left(\left((A-C)a^3+3\,Ba^2b-3\,(A-C)ab^2-Bb^3\right)c^2-2\,\left(Ba^3-3\,(A-C)a^2b-3\,Bab^2+(A-C)b^3\right)cd-\left((A-C)a^3+3\,Ba^2b-3\,(A-C)ab^2-Bb^3\right)d^2\right)(fx+e)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}}{}$$

```
[In] integrate((c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e
))^3,x, algorithm="maxima")
```

```
[Out] 1/2*(2*(((A - C)*a^3 + 3*B*a^2*b - 3*(A - C)*a*b^2 - B*b^3)*c^2 - 2*(B*a^3
- 3*(A - C)*a^2*b - 3*B*a*b^2 + (A - C)*b^3)*c*d - ((A - C)*a^3 + 3*B*a^2*b
 - 3*(A - C)*a*b^2 - B*b^3)*d^2)*(f*x + e)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b
^6) - 2*((B*a^3*b^3 - 3*(A - C)*a^2*b^4 - 3*B*a*b^5 + (A - C)*b^6)*c^2 + 2*
((A - C)*a^3*b^3 + 3*B*a^2*b^4 - 3*(A - C)*a*b^5 - B*b^6)*c*d - (C*a^6 + 3*
C*a^4*b^2 + B*a^3*b^3 - 3*(A - 2*C)*a^2*b^4 - 3*B*a*b^5 + A*b^6)*d^2)*log(b
*tan(f*x + e) + a)/(a^6*b^3 + 3*a^4*b^5 + 3*a^2*b^7 + b^9) + ((B*a^3 - 3*(A
 - C)*a^2*b - 3*B*a*b^2 + (A - C)*b^3)*c^2 + 2*((A - C)*a^3 + 3*B*a^2*b - 3
*(A - C)*a*b^2 - B*b^3)*c*d - (B*a^3 - 3*(A - C)*a^2*b - 3*B*a*b^2 + (A - C
```

```
)*b^3)*d^2)*log(tan(f*x + e)^2 + 1)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) - (
(C*a^4*b^2 - 3*B*a^3*b^3 + (5*A - 3*C)*a^2*b^4 + B*a*b^5 + A*b^6)*c^2 + 2*(
C*a^5*b + B*a^4*b^2 - (3*A - 5*C)*a^3*b^3 - 3*B*a^2*b^4 + A*a*b^5)*c*d - (3
*C*a^6 - B*a^5*b - (A - 7*C)*a^4*b^2 - 5*B*a^3*b^3 + 3*A*a^2*b^4)*d^2 - 2*(
(B*a^2*b^4 - 2*(A - C)*a*b^5 - B*b^6)*c^2 - 2*(C*a^4*b^2 - (A - 3*C)*a^2*b^
4 - 2*B*a*b^5 + A*b^6)*c*d + (2*C*a^5*b - B*a^4*b^2 + 4*C*a^3*b^3 - 3*B*a^2
*b^4 + 2*A*a*b^5)*d^2)*tan(f*x + e))/(a^6*b^3 + 2*a^4*b^5 + a^2*b^7 + (a^4*
b^5 + 2*a^2*b^7 + b^9)*tan(f*x + e)^2 + 2*(a^5*b^4 + 2*a^3*b^6 + a*b^8)*tan
(f*x + e)))/f
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1668 vs. $2(595) = 1190$.

Time = 1.01 (sec) , antiderivative size = 1668, normalized size of antiderivative = 2.79

$$\int \frac{(c+d\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx = \text{Too large to display}$$

```
[In] integrate((c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e
))^3,x, algorithm="giac")
```

```
[Out] 1/2*(2*(A*a^3*c^2 - C*a^3*c^2 + 3*B*a^2*b*c^2 - 3*A*a*b^2*c^2 + 3*C*a*b^2*c
^2 - B*b^3*c^2 - 2*B*a^3*c*d + 6*A*a^2*b*c*d - 6*C*a^2*b*c*d + 6*B*a*b^2*c*
d - 2*A*b^3*c*d + 2*C*b^3*c*d - A*a^3*d^2 + C*a^3*d^2 - 3*B*a^2*b*d^2 + 3*A
*a*b^2*d^2 - 3*C*a*b^2*d^2 + B*b^3*d^2)*(f*x + e)/(a^6 + 3*a^4*b^2 + 3*a^2*
b^4 + b^6) + (B*a^3*c^2 - 3*A*a^2*b*c^2 + 3*C*a^2*b*c^2 - 3*B*a*b^2*c^2 + A
*b^3*c^2 - C*b^3*c^2 + 2*A*a^3*c*d - 2*C*a^3*c*d + 6*B*a^2*b*c*d - 6*A*a*b^
2*c*d + 6*C*a*b^2*c*d - 2*B*b^3*c*d - B*a^3*d^2 + 3*A*a^2*b*d^2 - 3*C*a^2*b
*d^2 + 3*B*a*b^2*d^2 - A*b^3*d^2 + C*b^3*d^2)*log(tan(f*x + e)^2 + 1)/(a^6
+ 3*a^4*b^2 + 3*a^2*b^4 + b^6) - 2*(B*a^3*b^3*c^2 - 3*A*a^2*b^4*c^2 + 3*C*a
^2*b^4*c^2 - 3*B*a*b^5*c^2 + A*b^6*c^2 - C*b^6*c^2 + 2*A*a^3*b^3*c*d - 2*C*
a^3*b^3*c*d + 6*B*a^2*b^4*c*d - 6*A*a*b^5*c*d + 6*C*a*b^5*c*d - 2*B*b^6*c*d
 - C*a^6*d^2 - 3*C*a^4*b^2*d^2 - B*a^3*b^3*d^2 + 3*A*a^2*b^4*d^2 - 6*C*a^2*
b^4*d^2 + 3*B*a*b^5*d^2 - A*b^6*d^2)*log(abs(b*tan(f*x + e) + a))/(a^6*b^3
+ 3*a^4*b^5 + 3*a^2*b^7 + b^9) + (3*B*a^3*b^4*c^2*tan(f*x + e)^2 - 9*A*a^2*
b^5*c^2*tan(f*x + e)^2 + 9*C*a^2*b^5*c^2*tan(f*x + e)^2 - 9*B*a*b^6*c^2*tan
(f*x + e)^2 + 3*A*b^7*c^2*tan(f*x + e)^2 - 3*C*b^7*c^2*tan(f*x + e)^2 + 6*A
*a^3*b^4*c*d*tan(f*x + e)^2 - 6*C*a^3*b^4*c*d*tan(f*x + e)^2 + 18*B*a^2*b^5
*c*d*tan(f*x + e)^2 - 18*A*a*b^6*c*d*tan(f*x + e)^2 + 18*C*a*b^6*c*d*tan(f*
x + e)^2 - 6*B*b^7*c*d*tan(f*x + e)^2 - 3*C*a^6*b*d^2*tan(f*x + e)^2 - 9*C*
a^4*b^3*d^2*tan(f*x + e)^2 - 3*B*a^3*b^4*d^2*tan(f*x + e)^2 + 9*A*a^2*b^5*d
^2*tan(f*x + e)^2 - 18*C*a^2*b^5*d^2*tan(f*x + e)^2 + 9*B*a*b^6*d^2*tan(f*x
 + e)^2 - 3*A*b^7*d^2*tan(f*x + e)^2 + 8*B*a^4*b^3*c^2*tan(f*x + e) - 22*A*
a^3*b^4*c^2*tan(f*x + e) + 22*C*a^3*b^4*c^2*tan(f*x + e) - 18*B*a^2*b^5*c^2
*tan(f*x + e) + 2*A*a*b^6*c^2*tan(f*x + e) - 2*C*a*b^6*c^2*tan(f*x + e) - 2
```

```
*B*b^7*c^2*tan(f*x + e) - 4*C*a^6*b*c*d*tan(f*x + e) + 16*A*a^4*b^3*c*d*tan
(f*x + e) - 28*C*a^4*b^3*c*d*tan(f*x + e) + 44*B*a^3*b^4*c*d*tan(f*x + e) -
 36*A*a^2*b^5*c*d*tan(f*x + e) + 24*C*a^2*b^5*c*d*tan(f*x + e) - 4*B*a*b^6*
c*d*tan(f*x + e) - 4*A*b^7*c*d*tan(f*x + e) - 2*C*a^7*d^2*tan(f*x + e) - 2*
B*a^6*b*d^2*tan(f*x + e) - 6*C*a^5*b^2*d^2*tan(f*x + e) - 14*B*a^4*b^3*d^2*
tan(f*x + e) + 22*A*a^3*b^4*d^2*tan(f*x + e) - 28*C*a^3*b^4*d^2*tan(f*x + e
) + 12*B*a^2*b^5*d^2*tan(f*x + e) - 2*A*a*b^6*d^2*tan(f*x + e) - C*a^6*b*c^
2 + 6*B*a^5*b^2*c^2 - 14*A*a^4*b^3*c^2 + 11*C*a^4*b^3*c^2 - 7*B*a^3*b^4*c^2
 - 3*A*a^2*b^5*c^2 - B*a*b^6*c^2 - A*b^7*c^2 - 2*C*a^7*c*d - 2*B*a^6*b*c*d
+ 12*A*a^5*b^2*c*d - 18*C*a^5*b^2*c*d + 22*B*a^4*b^3*c*d - 14*A*a^3*b^4*c*d
 + 8*C*a^3*b^4*c*d - 2*A*a*b^6*c*d - B*a^7*d^2 - A*a^6*b*d^2 + C*a^6*b*d^2
- 9*B*a^5*b^2*d^2 + 11*A*a^4*b^3*d^2 - 11*C*a^4*b^3*d^2 + 4*B*a^3*b^4*d^2)/
((a^6*b^2 + 3*a^4*b^4 + 3*a^2*b^6 + b^8)*(b*tan(f*x + e) + a)^2))/f
```

Mupad [B] (verification not implemented)

Time = 27.62 (sec) , antiderivative size = 807, normalized size of antiderivative = 1.35

$$\int \frac{(c+d\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx =$$

$$-\frac{\ln\left(a+b\tan(e+f\,x)\right)\,\left(\frac{a^2\,\left(b^4\,\left(3\,A\,d^2-3\,A\,c^2+3\,C\,c^2-6\,C\,d^2+6\,B\,c\,d\right)+3\,C\,b^4\,d^2\right)-b^6\,\left(A\,d^2-A\,c^2+C\,c^2+2\,B\,c\,d\right)+C\,b^6\,d^2-a\,b^5\,\left(3\,B\right.}{a^6\,b^3+3\,a^4\,b^5+3\,a^2\,b^7+b^9}\right.}{f}$$

$$-\frac{\frac{A\,b^6\,c^2-3\,C\,a^6\,d^2+B\,a\,b^5\,c^2+B\,a^5\,b\,d^2+5\,A\,a^2\,b^4\,c^2-3\,A\,a^2\,b^4\,d^2+A\,a^4\,b^2\,d^2-3\,B\,a^3\,b^3\,c^2+5\,B\,a^3\,b^3\,d^2-3\,C\,a^2\,b^4\,c^2+C\,a^4\,b^2\,c^2-7\,C\,a^4\,b^2\,d^2+}{2\,b^3\,\left(a^4+2\,a^2\,b^2+b^4\right)}}{}$$

$$-\frac{\ln\left(\tan(e+f\,x)-\mathrm{i}\right)\,\left(B\,c^2-B\,d^2+2\,A\,c\,d-2\,C\,c\,d-A\,c^2\,1\mathrm{i}+A\,d^2\,1\mathrm{i}+C\,c^2\,1\mathrm{i}-C\,d^2\,1\mathrm{i}+B\,c\,d\,2\mathrm{i}\right)}{2\,f\,\left(-a^3-a^2\,b\,3\mathrm{i}+3\,a\,b^2+b^3\,1\mathrm{i}\right)}$$
$$-\frac{\ln\left(\tan(e+f\,x)+1\mathrm{i}\right)\,\left(A\,d^2-A\,c^2+B\,c^2\,1\mathrm{i}-B\,d^2\,1\mathrm{i}+C\,c^2-C\,d^2+A\,c\,d\,2\mathrm{i}+2\,B\,c\,d-C\,c\,d\,2\mathrm{i}\right)}{2\,f\,\left(-a^3\,1\mathrm{i}-3\,a^2\,b+a\,b^2\,3\mathrm{i}+b^3\right)}$$

```
[In] int(((c + d*tan(e + f*x))^2*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a + b
*tan(e + f*x))^3,x)
```

```
[Out] - (log(a + b*tan(e + f*x))*((a^2*(b^4*(3*A*d^2 - 3*A*c^2 + 3*C*c^2 - 6*C*d^
2 + 6*B*c*d) + 3*C*b^4*d^2) - b^6*(A*d^2 - A*c^2 + C*c^2 + 2*B*c*d) + C*b^6
*d^2 - a*b^5*(3*B*c^2 - 3*B*d^2 + 6*A*c*d - 6*C*c*d) + a^3*b^3*(B*c^2 - B*d
^2 + 2*A*c*d - 2*C*c*d))/(b^9 + 3*a^2*b^7 + 3*a^4*b^5 + a^6*b^3) - (C*d^2)/
b^3))/f - ((A*b^6*c^2 - 3*C*a^6*d^2 + B*a*b^5*c^2 + B*a^5*b*d^2 + 5*A*a^2*b
^4*c^2 - 3*A*a^2*b^4*d^2 + A*a^4*b^2*d^2 - 3*B*a^3*b^3*c^2 + 5*B*a^3*b^3*d^
2 - 3*C*a^2*b^4*c^2 + C*a^4*b^2*c^2 - 7*C*a^4*b^2*d^2 + 2*A*a*b^5*c*d + 2*C
*a^5*b*c*d - 6*A*a^3*b^3*c*d - 6*B*a^2*b^4*c*d + 2*B*a^4*b^2*c*d + 10*C*a^3
*b^3*c*d)/(2*b^3*(a^4 + b^4 + 2*a^2*b^2)) + (tan(e + f*x)*(B*b^5*c^2 - 2*C*
a^5*d^2 + 2*A*b^5*c*d + 2*A*a*b^4*c^2 - 2*A*a*b^4*d^2 + B*a^4*b*d^2 - 2*C*a
*b^4*c^2 - B*a^2*b^3*c^2 + 3*B*a^2*b^3*d^2 - 4*C*a^3*b^2*d^2 - 4*B*a*b^4*c*
```

```
d + 2*C*a^4*b*c*d - 2*A*a^2*b^3*c*d + 6*C*a^2*b^3*c*d))/(b^2*(a^4 + b^4 + 2
*a^2*b^2)))/(f*(a^2 + b^2*tan(e + f*x)^2 + 2*a*b*tan(e + f*x))) - (log(tan(
e + f*x) - 1i)*(A*d^2*1i - A*c^2*1i + B*c^2 - B*d^2 + C*c^2*1i - C*d^2*1i +
 2*A*c*d + B*c*d*2i - 2*C*c*d))/(2*f*(3*a*b^2 - a^2*b*3i - a^3 + b^3*1i)) -
 (log(tan(e + f*x) + 1i)*(A*d^2 - A*c^2 + B*c^2*1i - B*d^2*1i + C*c^2 - C*d
^2 + A*c*d*2i + 2*B*c*d - C*c*d*2i))/(2*f*(a*b^2*3i - 3*a^2*b - a^3*1i + b^
3))
```

3.64 $\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^3\left(A+B\tan(e+\right.$

Optimal result 576
Rubi [A] (verified) 577
Mathematica [C] (verified) 581
Maple [A] (verified) 582
Fricas [A] (verification not implemented) 582
Sympy [B] (verification not implemented) 583
Maxima [A] (verification not implemented) 584
Giac [B] (verification not implemented) 585
Mupad [B] (verification not implemented) 598

Optimal result

Integrand size = 45, antiderivative size = 603

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\left(a^2\left(Ac^3-c^3C-3Bc^2d-3Acd^2+3cCd^2+Bd^3\right)\right.$$
$$+b^2\left(c^3C+3Bc^2d-3cCd^2-Bd^3-A\left(c^3-3cd^2\right)\right)$$
$$\left.-2ab\left((A-C)d\left(3c^2-d^2\right)+B\left(c^3-3cd^2\right)\right)\right)x$$
$$+\frac{(2ab(c^3C+3Bc^2d-3cCd^2-Bd^3-A(c^3-3cd^2))-a^2((A-C)d(3c^2-d^2)+B(c^3-3cd^2))+b^2((A-}{f}$$
$$-\frac{d(2ab(c^2C+2Bcd-Cd^2-A(c^2-d^2))-a^2(2c(A-C)d+B(c^2-d^2))+b^2(2c(A-C)d+B(c^2-d^2)}{f}$$
$$+\frac{(2ab(Ac-cC-Bd)+a^2(Bc+(A-C)d)-b^2(Bc+(A-C)d))\,(c+d\tan(e+fx))^2}{2f}$$
$$+\frac{(a^2B-b^2B+2ab(A-C))\,(c+d\tan(e+fx))^3}{3f}$$
$$+\frac{(5a^2Cd^2-6abd(cC-5Bd)+b^2(c^2C-3Bcd+15(A-C)d^2))\,(c+d\tan(e+fx))^4}{60d^3f}$$
$$-\frac{b(bcC-3bBd-aCd)\tan(e+fx)(c+d\tan(e+fx))^4}{15d^2f}$$
$$+\frac{C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^4}{6df}$$

```
[Out] (a^2*(A*c^3-3*A*c*d^2-3*B*c^2*d+B*d^3-C*c^3+3*C*c*d^2)+b^2*(c^3*C+3*B*c^2*d
-3*C*c*d^2-B*d^3-A*(c^3-3*c*d^2))-2*a*b*((A-C)*d*(3*c^2-d^2)+B*(c^3-3*c*d^2
)))*x+(2*a*b*(c^3*C+3*B*c^2*d-3*C*c*d^2-B*d^3-A*(c^3-3*c*d^2))-a^2*((A-C)*d
*(3*c^2-d^2)+B*(c^3-3*c*d^2))+b^2*((A-C)*d*(3*c^2-d^2)+B*(c^3-3*c*d^2)))*ln
(cos(f*x+e))/f-d*(2*a*b*(c^2*C+2*B*c*d-C*d^2-A*(c^2-d^2))-a^2*(2*c*(A-C)*d+
B*(c^2-d^2))+b^2*(2*c*(A-C)*d+B*(c^2-d^2)))*tan(f*x+e)/f+1/2*(2*a*b*(A*c-B*
```

```
d-C*c)+a^2*(B*c+(A-C)*d)-b^2*(B*c+(A-C)*d))*(c+d*tan(f*x+e))^2/f+1/3*(B*a^2
-B*b^2+2*a*b*(A-C))*(c+d*tan(f*x+e))^3/f+1/60*(5*a^2*C*d^2-6*a*b*d*(-5*B*d+
C*c)+b^2*(c^2*C-3*B*c*d+15*(A-C)*d^2))*(c+d*tan(f*x+e))^4/d^3/f-1/15*b*(-3*
B*b*d-C*a*d+C*b*c)*tan(f*x+e)*(c+d*tan(f*x+e))^4/d^2/f+1/6*C*(a+b*tan(f*x+e
))^2*(c+d*tan(f*x+e))^4/d/f
```

Rubi [A] (verified)

Time = 1.66 (sec) , antiderivative size = 603, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}} = 0.133$, Rules used = {3728, 3718, 3711, 3609, 3606, 3556}

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx =$$

$$-\frac{d\tan(e+fx)\left(-(a^2(2cd(A-C)+B(c^2-d^2)))+2ab(-A(c^2-d^2)+2Bcd+c^2C-Cd^2)+b^2(2cd(A\right.}{f}$$

$$+\frac{(c+d\tan(e+fx))^4\left(5a^2Cd^2-6abd(cC-5Bd)+b^2(15d^2(A-C)-3Bcd+c^2C)\right)}{60d^3f}$$

$$+\frac{\log(\cos(e+fx))\left(-(a^2(d(A-C)\left(3c^2-d^2\right)+B(c^3-3cd^2)))+2ab(-A(c^3-3cd^2)+3Bc^2d-Bd^3+\right.}{f}$$

$$+x\left(a^2\left(Ac^3-3Acd^2-3Bc^2d+Bd^3-c^3C+3cCd^2\right)\right.$$
$$-2ab\left(d(A-C)\left(3c^2-d^2\right)+B\left(c^3-3cd^2\right)\right)$$
$$\left.+b^2\left(-A\left(c^3-3cd^2\right)+3Bc^2d-Bd^3+c^3C-3cCd^2\right)\right)$$

$$+\frac{\left(a^2B+2ab(A-C)-b^2B\right)(c+d\tan(e+fx))^3}{3f}$$

$$+\frac{(c+d\tan(e+fx))^2\left(a^2(d(A-C)+Bc)+2ab(Ac-Bd-cC)-b^2(d(A-C)+Bc)\right)}{2f}$$

$$-\frac{b\tan(e+fx)(-aCd-3bBd+bcC)(c+d\tan(e+fx))^4}{15d^2f}$$

$$+\frac{C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^4}{6df}$$

```
[In] Int[(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*T
an[e + f*x]^2),x]
```

```
[Out] (a^2*(A*c^3 - c^3*C - 3*B*c^2*d - 3*A*c*d^2 + 3*c*C*d^2 + B*d^3) + b^2*(c^3
*C + 3*B*c^2*d - 3*c*C*d^2 - B*d^3 - A*(c^3 - 3*c*d^2)) - 2*a*b*((A - C)*d*
(3*c^2 - d^2) + B*(c^3 - 3*c*d^2)))*x + ((2*a*b*(c^3*C + 3*B*c^2*d - 3*c*C*
d^2 - B*d^3 - A*(c^3 - 3*c*d^2)) - a^2*((A - C)*d*(3*c^2 - d^2) + B*(c^3 -
3*c*d^2)) + b^2*((A - C)*d*(3*c^2 - d^2) + B*(c^3 - 3*c*d^2)))*Log[Cos[e +
f*x]])/f - (d*(2*a*b*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - a^2*(2*c*(
A - C)*d + B*(c^2 - d^2)) + b^2*(2*c*(A - C)*d + B*(c^2 - d^2)))*Tan[e + f*
x])/f + ((2*a*b*(A*c - c*C - B*d) + a^2*(B*c + (A - C)*d) - b^2*(B*c + (A -
```

```
C)*d))*(c + d*Tan[e + f*x])^2)/(2*f) + ((a^2*B - b^2*B + 2*a*b*(A - C))*(c
 + d*Tan[e + f*x])^3)/(3*f) + ((5*a^2*C*d^2 - 6*a*b*d*(c*C - 5*B*d) + b^2*(
c^2*C - 3*B*c*d + 15*(A - C)*d^2))*(c + d*Tan[e + f*x])^4)/(60*d^3*f) - (b*
(b*c*C - 3*b*B*d - a*C*d)*Tan[e + f*x]*(c + d*Tan[e + f*x])^4)/(15*d^2*f) +
 (C*(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])^4)/(6*d*f)
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3606

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
 f*x], x], x] + Simp[b*d*(Tan[e + f*x]/f), x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3609

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[d*((a + b*Tan[e + f*x])^m/(f*m)), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\text{integral} = \frac{C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^4}{6df} + \frac{\int (a+b\tan(e+fx))(c+d\tan(e+fx))^3\left(-2(bcC-3aAd+2aCd)+6(Ab+aB-bC)d\tan(e+fx)\right.}{6d}$$

$$= -\frac{b(bcC-3bBd-aCd)\tan(e+fx)(c+d\tan(e+fx))^4}{15d^2f} + \frac{C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^4}{6df} - \frac{\int (c+d\tan(e+fx))^3\left(2(6abcCd-5a^2(3A-2C)d^2-b^2c(cC-3Bd))-30(a^2B-b^2B+2ab\right.}{}$$

$$= \frac{\left(5a^2Cd^2-6abd(cC-5Bd)+b^2(c^2C-3Bcd+15(A-C)d^2)\right)(c+d\tan(e+fx))^4}{60d^3f} - \frac{b(bcC-3bBd-aCd)\tan(e+fx)(c+d\tan(e+fx))^4}{15d^2f} + \frac{C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^4}{6df} - \frac{\int (c+d\tan(e+fx))^3\left(30(2abB-a^2(A-C)+b^2(A-C))d^2-30(a^2B-b^2B+2ab(A-C))\right.}{30d^2}$$

$$= \frac{\left(a^2B-b^2B+2ab(A-C)\right)(c+d\tan(e+fx))^3}{3f} + \frac{\left(5a^2Cd^2-6abd(cC-5Bd)+b^2(c^2C-3Bcd+15(A-C)d^2)\right)(c+d\tan(e+fx))^4}{60d^3f} - \frac{b(bcC-3bBd-aCd)\tan(e+fx)(c+d\tan(e+fx))^4}{15d^2f} + \frac{C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^4}{6df} - \frac{\int (c+d\tan(e+fx))^2\left(-30d^2(a^2(Ac-cC-Bd)-b^2(Ac-cC-Bd)-2ab(Bc+(A-C)d)\right)}{30d^2}$$

$$= \frac{(2ab(Ac-cC-Bd)+a^2(Bc+(A-C)d)-b^2(Bc+(A-C)d))\,(c+d\tan(e+fx))^2}{2f}$$
$$+\frac{(a^2B-b^2B+2ab(A-C))\,(c+d\tan(e+fx))^3}{3f}$$
$$+\frac{(5a^2Cd^2-6abd(cC-5Bd)+b^2(c^2C-3Bcd+15(A-C)d^2))\,(c+d\tan(e+fx))^4}{60d^3f}$$
$$-\frac{b(bcC-3bBd-aCd)\tan(e+fx)(c+d\tan(e+fx))^4}{15d^2f}$$
$$+\frac{C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^4}{6df}$$
$$-\frac{\int(c+d\tan(e+fx))\,(30d^2(a^2(c^2C+2Bcd-Cd^2-A(c^2-d^2))-b^2(c^2C+2Bcd-Cd^2-A(c^2}{}$$

$$= \big(a^2\big(Ac^3-c^3C-3Bc^2d-3Acd^2+3cCd^2+Bd^3\big)$$
$$+b^2\big(c^3C+3Bc^2d-3cCd^2-Bd^3-A\big(c^3-3cd^2\big)\big)$$
$$-2ab\big((A-C)d\big(3c^2-d^2\big)+B\big(c^3-3cd^2\big)\big)\big)\,x$$
$$-\frac{d(2ab(c^2C+2Bcd-Cd^2-A(c^2-d^2))-a^2(2c(A-C)d+B(c^2-d^2))+b^2(2c(A-C)d+B(}{f}$$
$$+\frac{(2ab(Ac-cC-Bd)+a^2(Bc+(A-C)d)-b^2(Bc+(A-C)d))\,(c+d\tan(e+fx))^2}{2f}$$
$$+\frac{(a^2B-b^2B+2ab(A-C))\,(c+d\tan(e+fx))^3}{3f}$$
$$+\frac{(5a^2Cd^2-6abd(cC-5Bd)+b^2(c^2C-3Bcd+15(A-C)d^2))\,(c+d\tan(e+fx))^4}{60d^3f}$$
$$-\frac{b(bcC-3bBd-aCd)\tan(e+fx)(c+d\tan(e+fx))^4}{15d^2f}$$
$$+\frac{C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^4}{6df}$$
$$-\frac{(30d^3(a^2(c^2C+2Bcd-Cd^2-A(c^2-d^2))-b^2(c^2C+2Bcd-Cd^2-A(c^2-d^2))+2ab(2c(A-}{}$$

$$= \big(a^2\big(Ac^3 - c^3C - 3Bc^2d - 3Acd^2 + 3cCd^2 + Bd^3\big) + b^2\big(c^3C + 3Bc^2d - 3cCd^2 - Bd^3 - A\big(c^3 - 3cd^2\big)\big) - 2ab\big((A-C)d\big(3c^2 - d^2\big) + B\big(c^3 - 3cd^2\big)\big)\big)\,x$$
$$+ \frac{(2ab(c^3C + 3Bc^2d - 3cCd^2 - Bd^3 - A(c^3 - 3cd^2)) - a^2((A-C)d(3c^2 - d^2) + B(c^3 - 3cd^2))}{f}$$
$$- \frac{d(2ab(c^2C + 2Bcd - Cd^2 - A(c^2 - d^2)) - a^2(2c(A-C)d + B(c^2 - d^2)) + b^2(2c(A-C)d + B}{f}$$
$$+ \frac{(2ab(Ac - cC - Bd) + a^2(Bc + (A-C)d) - b^2(Bc + (A-C)d))\,(c + d\tan(e+fx))^2}{2f}$$
$$+ \frac{(a^2B - b^2B + 2ab(A-C))\,(c + d\tan(e+fx))^3}{3f}$$
$$+ \frac{(5a^2Cd^2 - 6abd(cC - 5Bd) + b^2(c^2C - 3Bcd + 15(A-C)d^2))\,(c + d\tan(e+fx))^4}{60d^3f}$$
$$- \frac{b(bcC - 3bBd - aCd)\tan(e+fx)(c + d\tan(e+fx))^4}{15d^2f}$$
$$+ \frac{C(a + b\tan(e+fx))^2(c + d\tan(e+fx))^4}{6df}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 6.66 (sec) , antiderivative size = 419, normalized size of antiderivative = 0.69

$$\int (a + b\tan(e+fx))^2(c + d\tan(e+fx))^3\left(A + B\tan(e+fx) + C\tan^2(e+fx)\right)\,dx$$
$$= \frac{C(a + b\tan(e+fx))^2(c + d\tan(e+fx))^4}{6df}$$
$$+ \frac{-\frac{2b(bcC-3bBd-aCd)\tan(e+fx)(c+d\tan(e+fx))^4}{5df} - \frac{-\frac{\left(5a^2Cd^2-6abd(cC-5Bd)+b^2\left(c^2C-3Bcd+15(A-C)d^2\right)\right)(c+d\tan(e+fx))^4}{2df} + \frac{5\left(3d\left(2ab(A\right.\right.}{}}{}}{}$$

```
[In] Integrate[(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])^3*(A + B*Tan[e + f*x]
 + C*Tan[e + f*x]^2),x]
```

```
[Out] (C*(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])^4)/(6*d*f) + ((-2*b*(b*c*C -
 3*b*B*d - a*C*d)*Tan[e + f*x]*(c + d*Tan[e + f*x])^4)/(5*d*f) - (-1/2*((5*
a^2*C*d^2 - 6*a*b*d*(c*C - 5*B*d) + b^2*(c^2*C - 3*B*c*d + 15*(A - C)*d^2))
*(c + d*Tan[e + f*x])^4)/(d*f) + (5*(3*d*(2*a*b*(A*c - c*C + B*d) + a^2*(B*
c - (A - C)*d) - b^2*(B*c - (A - C)*d))*((I*c - d)^3*Log[I - Tan[e + f*x]]
- (I*c + d)^3*Log[I + Tan[e + f*x]] + 6*c*d^2*Tan[e + f*x] + d^3*Tan[e + f*
x]^2) + (a^2*B - b^2*B + 2*a*b*(A - C))*d*((3*I)*(c + I*d)^4*Log[I - Tan[e
+ f*x]] - (3*I)*(c - I*d)^4*Log[I + Tan[e + f*x]] - 6*d^2*(6*c^2 - d^2)*Tan
[e + f*x] - 12*c*d^3*Tan[e + f*x]^2 - 2*d^4*Tan[e + f*x]^3)))/f)/(5*d))/(6*
d)
```

Maple [A] (verified)

Time = 0.44 (sec) , antiderivative size = 546, normalized size of antiderivative = 0.91

method	result
parts	$\frac{(3A\,a^2c^2d+2Aab\,c^3+B\,a^2c^3)\ln\left(1+\tan(fx+e)^2\right)}{2f}+\frac{(B\,b^2d^3+2Cab\,d^3+3C\,b^2c\,d^2)\left(\frac{\tan(fx+e)^5}{5}-\frac{\tan(fx+e)^3}{3}+\tan(fx+\right.}{f}$
norman	$(A\,a^2c^3-3A\,a^2c\,d^2-6Aab\,c^2d+2Aab\,d^3-A\,b^2c^3+3A\,b^2c\,d^2-3B\,a^2c^2d+B\,a^2d^3-2$
derivativedivides	Expression too large to display
default	Expression too large to display
parallelrisch	Expression too large to display
risch	Expression too large to display

```
[In] int((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x
,method=_RETURNVERBOSE)
```

```
[Out] 1/2*(3*A*a^2*c^2*d+2*A*a*b*c^3+B*a^2*c^3)/f*ln(1+tan(f*x+e)^2)+(B*b^2*d^3+2
*C*a*b*d^3+3*C*b^2*c*d^2)/f*(1/5*tan(f*x+e)^5-1/3*tan(f*x+e)^3+tan(f*x+e)-a
rctan(tan(f*x+e)))+(A*b^2*d^3+2*B*a*b*d^3+3*B*b^2*c*d^2+C*a^2*d^3+6*C*a*b*c
*d^2+3*C*b^2*c^2*d)/f*(1/4*tan(f*x+e)^4-1/2*tan(f*x+e)^2+1/2*ln(1+tan(f*x+e
)^2))+(3*A*a^2*c*d^2+6*A*a*b*c^2*d+A*b^2*c^3+3*B*a^2*c^2*d+2*B*a*b*c^3+C*a^
2*c^3)/f*(tan(f*x+e)-arctan(tan(f*x+e)))+(2*A*a*b*d^3+3*A*b^2*c*d^2+B*a^2*d
^3+6*B*a*b*c*d^2+3*B*b^2*c^2*d+3*C*a^2*c*d^2+6*C*a*b*c^2*d+C*b^2*c^3)/f*(1/
3*tan(f*x+e)^3-tan(f*x+e)+arctan(tan(f*x+e)))+(A*a^2*d^3+6*A*a*b*c*d^2+3*A*
b^2*c^2*d+3*B*a^2*c*d^2+6*B*a*b*c^2*d+B*b^2*c^3+3*C*a^2*c^2*d+2*C*a*b*c^3)/
f*(1/2*tan(f*x+e)^2-1/2*ln(1+tan(f*x+e)^2))+A*a^2*c^3*x+C*b^2*d^3/f*(1/6*ta
n(f*x+e)^6-1/4*tan(f*x+e)^4+1/2*tan(f*x+e)^2-1/2*ln(1+tan(f*x+e)^2))
```

Fricas [A] (verification not implemented)

none

Time = 0.29 (sec) , antiderivative size = 679, normalized size of antiderivative = 1.13

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\frac{10\,Cb^2d^3\tan\left(fx+e\right)^6+12\left(3\,Cb^2cd^2+(2\,Cab+Bb^2)d^3\right)\tan\left(fx+e\right)^5+15\left(3\,Cb^2c^2d+3\,(2\,Cab+Bb^2)c\right.}{}$$

```
[In] integrate((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e
)^2),x, algorithm="fricas")
```

```
[Out] 1/60*(10*C*b^2*d^3*tan(f*x + e)^6 + 12*(3*C*b^2*c*d^2 + (2*C*a*b + B*b^2)*d
^3)*tan(f*x + e)^5 + 15*(3*C*b^2*c^2*d + 3*(2*C*a*b + B*b^2)*c*d^2 + (C*a^2
```

```
 + 2*B*a*b + (A - C)*b^2)*d^3)*tan(f*x + e)^4 + 20*(C*b^2*c^3 + 3*(2*C*a*b
+ B*b^2)*c^2*d + 3*(C*a^2 + 2*B*a*b + (A - C)*b^2)*c*d^2 + (B*a^2 + 2*(A -
C)*a*b - B*b^2)*d^3)*tan(f*x + e)^3 + 60*(((A - C)*a^2 - 2*B*a*b - (A - C)*
b^2)*c^3 - 3*(B*a^2 + 2*(A - C)*a*b - B*b^2)*c^2*d - 3*((A - C)*a^2 - 2*B*a
*b - (A - C)*b^2)*c*d^2 + (B*a^2 + 2*(A - C)*a*b - B*b^2)*d^3)*f*x + 30*((2
*C*a*b + B*b^2)*c^3 + 3*(C*a^2 + 2*B*a*b + (A - C)*b^2)*c^2*d + 3*(B*a^2 +
2*(A - C)*a*b - B*b^2)*c*d^2 + ((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*d^3)*t
an(f*x + e)^2 - 30*((B*a^2 + 2*(A - C)*a*b - B*b^2)*c^3 + 3*((A - C)*a^2 -
2*B*a*b - (A - C)*b^2)*c^2*d - 3*(B*a^2 + 2*(A - C)*a*b - B*b^2)*c*d^2 - ((
A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*d^3)*log(1/(tan(f*x + e)^2 + 1)) + 60*(
(C*a^2 + 2*B*a*b + (A - C)*b^2)*c^3 + 3*(B*a^2 + 2*(A - C)*a*b - B*b^2)*c^2
*d + 3*((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*c*d^2 - (B*a^2 + 2*(A - C)*a*b
 - B*b^2)*d^3)*tan(f*x + e))/f
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1819 vs. $2(547) = 1094$.

Time = 0.39 (sec) , antiderivative size = 1819, normalized size of antiderivative = 3.02

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= \text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))**2*(c+d*tan(f*x+e))**3*(A+B*tan(f*x+e)+C*tan(f*x
+e)**2),x)
```

```
[Out] Piecewise((A*a**2*c**3*x + 3*A*a**2*c**2*d*log(tan(e + f*x)**2 + 1)/(2*f) -
 3*A*a**2*c*d**2*x + 3*A*a**2*c*d**2*tan(e + f*x)/f - A*a**2*d**3*log(tan(e
 + f*x)**2 + 1)/(2*f) + A*a**2*d**3*tan(e + f*x)**2/(2*f) + A*a*b*c**3*log(
tan(e + f*x)**2 + 1)/f - 6*A*a*b*c**2*d*x + 6*A*a*b*c**2*d*tan(e + f*x)/f -
 3*A*a*b*c*d**2*log(tan(e + f*x)**2 + 1)/f + 3*A*a*b*c*d**2*tan(e + f*x)**2
/f + 2*A*a*b*d**3*x + 2*A*a*b*d**3*tan(e + f*x)**3/(3*f) - 2*A*a*b*d**3*tan
(e + f*x)/f - A*b**2*c**3*x + A*b**2*c**3*tan(e + f*x)/f - 3*A*b**2*c**2*d*
log(tan(e + f*x)**2 + 1)/(2*f) + 3*A*b**2*c**2*d*tan(e + f*x)**2/(2*f) + 3*
A*b**2*c*d**2*x + A*b**2*c*d**2*tan(e + f*x)**3/f - 3*A*b**2*c*d**2*tan(e +
 f*x)/f + A*b**2*d**3*log(tan(e + f*x)**2 + 1)/(2*f) + A*b**2*d**3*tan(e +
f*x)**4/(4*f) - A*b**2*d**3*tan(e + f*x)**2/(2*f) + B*a**2*c**3*log(tan(e +
 f*x)**2 + 1)/(2*f) - 3*B*a**2*c**2*d*x + 3*B*a**2*c**2*d*tan(e + f*x)/f -
3*B*a**2*c*d**2*log(tan(e + f*x)**2 + 1)/(2*f) + 3*B*a**2*c*d**2*tan(e + f*
x)**2/(2*f) + B*a**2*d**3*x + B*a**2*d**3*tan(e + f*x)**3/(3*f) - B*a**2*d*
*3*tan(e + f*x)/f - 2*B*a*b*c**3*x + 2*B*a*b*c**3*tan(e + f*x)/f - 3*B*a*b*
c**2*d*log(tan(e + f*x)**2 + 1)/f + 3*B*a*b*c**2*d*tan(e + f*x)**2/f + 6*B*
a*b*c*d**2*x + 2*B*a*b*c*d**2*tan(e + f*x)**3/f - 6*B*a*b*c*d**2*tan(e + f*
x)/f + B*a*b*d**3*log(tan(e + f*x)**2 + 1)/f + B*a*b*d**3*tan(e + f*x)**4/(
2*f) - B*a*b*d**3*tan(e + f*x)**2/f - B*b**2*c**3*log(tan(e + f*x)**2 + 1)/
```

```
(2*f) + B*b**2*c**3*tan(e + f*x)**2/(2*f) + 3*B*b**2*c**2*d*x + B*b**2*c**2
*d*tan(e + f*x)**3/f - 3*B*b**2*c**2*d*tan(e + f*x)/f + 3*B*b**2*c*d**2*log
(tan(e + f*x)**2 + 1)/(2*f) + 3*B*b**2*c*d**2*tan(e + f*x)**4/(4*f) - 3*B*b
**2*c*d**2*tan(e + f*x)**2/(2*f) - B*b**2*d**3*x + B*b**2*d**3*tan(e + f*x)
**5/(5*f) - B*b**2*d**3*tan(e + f*x)**3/(3*f) + B*b**2*d**3*tan(e + f*x)/f
- C*a**2*c**3*x + C*a**2*c**3*tan(e + f*x)/f - 3*C*a**2*c**2*d*log(tan(e +
f*x)**2 + 1)/(2*f) + 3*C*a**2*c**2*d*tan(e + f*x)**2/(2*f) + 3*C*a**2*c*d**
2*x + C*a**2*c*d**2*tan(e + f*x)**3/f - 3*C*a**2*c*d**2*tan(e + f*x)/f + C*
a**2*d**3*log(tan(e + f*x)**2 + 1)/(2*f) + C*a**2*d**3*tan(e + f*x)**4/(4*f
) - C*a**2*d**3*tan(e + f*x)**2/(2*f) - C*a*b*c**3*log(tan(e + f*x)**2 + 1)
/f + C*a*b*c**3*tan(e + f*x)**2/f + 6*C*a*b*c**2*d*x + 2*C*a*b*c**2*d*tan(e
 + f*x)**3/f - 6*C*a*b*c**2*d*tan(e + f*x)/f + 3*C*a*b*c*d**2*log(tan(e + f
*x)**2 + 1)/f + 3*C*a*b*c*d**2*tan(e + f*x)**4/(2*f) - 3*C*a*b*c*d**2*tan(e
 + f*x)**2/f - 2*C*a*b*d**3*x + 2*C*a*b*d**3*tan(e + f*x)**5/(5*f) - 2*C*a*
b*d**3*tan(e + f*x)**3/(3*f) + 2*C*a*b*d**3*tan(e + f*x)/f + C*b**2*c**3*x
+ C*b**2*c**3*tan(e + f*x)**3/(3*f) - C*b**2*c**3*tan(e + f*x)/f + 3*C*b**2
*c**2*d*log(tan(e + f*x)**2 + 1)/(2*f) + 3*C*b**2*c**2*d*tan(e + f*x)**4/(4
*f) - 3*C*b**2*c**2*d*tan(e + f*x)**2/(2*f) - 3*C*b**2*c*d**2*x + 3*C*b**2*
c*d**2*tan(e + f*x)**5/(5*f) - C*b**2*c*d**2*tan(e + f*x)**3/f + 3*C*b**2*c
*d**2*tan(e + f*x)/f - C*b**2*d**3*log(tan(e + f*x)**2 + 1)/(2*f) + C*b**2*
d**3*tan(e + f*x)**6/(6*f) - C*b**2*d**3*tan(e + f*x)**4/(4*f) + C*b**2*d**
3*tan(e + f*x)**2/(2*f), Ne(f, 0)), (x*(a + b*tan(e))**2*(c + d*tan(e))**3*
(A + B*tan(e) + C*tan(e)**2), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.31 (sec) , antiderivative size = 680, normalized size of antiderivative = 1.13

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^3 \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= \frac{10\,Cb^2d^3\tan\left(fx+e\right)^6+12\left(3\,Cb^2cd^2+\left(2\,Cab+Bb^2\right)d^3\right)\tan\left(fx+e\right)^5+15\left(3\,Cb^2c^2d+3\left(2\,Cab+Bb^2\right)c\right.}{}$$

```
[In] integrate((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e
)^2),x, algorithm="maxima")
```

```
[Out] 1/60*(10*C*b^2*d^3*tan(f*x + e)^6 + 12*(3*C*b^2*c*d^2 + (2*C*a*b + B*b^2)*d
^3)*tan(f*x + e)^5 + 15*(3*C*b^2*c^2*d + 3*(2*C*a*b + B*b^2)*c*d^2 + (C*a^2
 + 2*B*a*b + (A - C)*b^2)*d^3)*tan(f*x + e)^4 + 20*(C*b^2*c^3 + 3*(2*C*a*b
+ B*b^2)*c^2*d + 3*(C*a^2 + 2*B*a*b + (A - C)*b^2)*c*d^2 + (B*a^2 + 2*(A -
C)*a*b - B*b^2)*d^3)*tan(f*x + e)^3 + 30*((2*C*a*b + B*b^2)*c^3 + 3*(C*a^2
+ 2*B*a*b + (A - C)*b^2)*c^2*d + 3*(B*a^2 + 2*(A - C)*a*b - B*b^2)*c*d^2 +
((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*d^3)*tan(f*x + e)^2 + 60*(((A - C)*a^
```

```
2 - 2*B*a*b - (A - C)*b^2)*c^3 - 3*(B*a^2 + 2*(A - C)*a*b - B*b^2)*c^2*d -
3*((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*c*d^2 + (B*a^2 + 2*(A - C)*a*b - B*
b^2)*d^3)*(f*x + e) + 30*((B*a^2 + 2*(A - C)*a*b - B*b^2)*c^3 + 3*((A - C)*
a^2 - 2*B*a*b - (A - C)*b^2)*c^2*d - 3*(B*a^2 + 2*(A - C)*a*b - B*b^2)*c*d^
2 - ((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*d^3)*log(tan(f*x + e)^2 + 1) + 60
*((C*a^2 + 2*B*a*b + (A - C)*b^2)*c^3 + 3*(B*a^2 + 2*(A - C)*a*b - B*b^2)*c
^2*d + 3*((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*c*d^2 - (B*a^2 + 2*(A - C)*a
*b - B*b^2)*d^3)*tan(f*x + e))/f
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 21368 vs. $2(593) = 1186$.

Time = 23.09 (sec) , antiderivative size = 21368, normalized size of antiderivative = 35.44

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= \text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e
)^2),x, algorithm="giac")
```

```
[Out] 1/60*(60*A*a^2*c^3*f*x*tan(f*x)^6*tan(e)^6 - 60*C*a^2*c^3*f*x*tan(f*x)^6*ta
n(e)^6 - 120*B*a*b*c^3*f*x*tan(f*x)^6*tan(e)^6 - 60*A*b^2*c^3*f*x*tan(f*x)^
6*tan(e)^6 + 60*C*b^2*c^3*f*x*tan(f*x)^6*tan(e)^6 - 180*B*a^2*c^2*d*f*x*tan
(f*x)^6*tan(e)^6 - 360*A*a*b*c^2*d*f*x*tan(f*x)^6*tan(e)^6 + 360*C*a*b*c^2*
d*f*x*tan(f*x)^6*tan(e)^6 + 180*B*b^2*c^2*d*f*x*tan(f*x)^6*tan(e)^6 - 180*A
*a^2*c*d^2*f*x*tan(f*x)^6*tan(e)^6 + 180*C*a^2*c*d^2*f*x*tan(f*x)^6*tan(e)^
6 + 360*B*a*b*c*d^2*f*x*tan(f*x)^6*tan(e)^6 + 180*A*b^2*c*d^2*f*x*tan(f*x)^
6*tan(e)^6 - 180*C*b^2*c*d^2*f*x*tan(f*x)^6*tan(e)^6 + 60*B*a^2*d^3*f*x*tan
(f*x)^6*tan(e)^6 + 120*A*a*b*d^3*f*x*tan(f*x)^6*tan(e)^6 - 120*C*a*b*d^3*f*
x*tan(f*x)^6*tan(e)^6 - 60*B*b^2*d^3*f*x*tan(f*x)^6*tan(e)^6 - 30*B*a^2*c^3
*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 +
 tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^6*tan(e)^6 - 60*A*a*b*c^3*log(4*(tan(
f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2
+ tan(e)^2 + 1))*tan(f*x)^6*tan(e)^6 + 60*C*a*b*c^3*log(4*(tan(f*x)^2*tan(e
)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 +
 1))*tan(f*x)^6*tan(e)^6 + 30*B*b^2*c^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(
f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x
)^6*tan(e)^6 - 90*A*a^2*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e
) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^6*tan(e)
^6 + 90*C*a^2*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(ta
n(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^6*tan(e)^6 + 180*B
*a*b*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*
tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^6*tan(e)^6 + 90*A*b^2*c^2*d
*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 +
```

tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^6*tan(e)^6 - 90*C*b^2*c^2*d*log(4*(ta
n(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^
2 + tan(e)^2 + 1))*tan(f*x)^6*tan(e)^6 + 90*B*a^2*c*d^2*log(4*(tan(f*x)^2*t
an(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)
^2 + 1))*tan(f*x)^6*tan(e)^6 + 180*A*a*b*c*d^2*log(4*(tan(f*x)^2*tan(e)^2 -
2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*
tan(f*x)^6*tan(e)^6 - 180*C*a*b*c*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*
x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^
6*tan(e)^6 - 90*B*b^2*c*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e)
+ 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^6*tan(e)^6
+ 30*A*a^2*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*
x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^6*tan(e)^6 - 30*C*a^2*
d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^
2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^6*tan(e)^6 - 60*B*a*b*d^3*log(4*(t
an(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)
^2 + tan(e)^2 + 1))*tan(f*x)^6*tan(e)^6 - 30*A*b^2*d^3*log(4*(tan(f*x)^2*ta
n(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^
2 + 1))*tan(f*x)^6*tan(e)^6 + 30*C*b^2*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*t
an(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(
f*x)^6*tan(e)^6 - 360*A*a^2*c^3*f*x*tan(f*x)^5*tan(e)^5 + 360*C*a^2*c^3*f*x
*tan(f*x)^5*tan(e)^5 + 720*B*a*b*c^3*f*x*tan(f*x)^5*tan(e)^5 + 360*A*b^2*c^
3*f*x*tan(f*x)^5*tan(e)^5 - 360*C*b^2*c^3*f*x*tan(f*x)^5*tan(e)^5 + 1080*B*
a^2*c^2*d*f*x*tan(f*x)^5*tan(e)^5 + 2160*A*a*b*c^2*d*f*x*tan(f*x)^5*tan(e)^
5 - 2160*C*a*b*c^2*d*f*x*tan(f*x)^5*tan(e)^5 - 1080*B*b^2*c^2*d*f*x*tan(f*x
)^5*tan(e)^5 + 1080*A*a^2*c*d^2*f*x*tan(f*x)^5*tan(e)^5 - 1080*C*a^2*c*d^2*
f*x*tan(f*x)^5*tan(e)^5 - 2160*B*a*b*c*d^2*f*x*tan(f*x)^5*tan(e)^5 - 1080*A
*b^2*c*d^2*f*x*tan(f*x)^5*tan(e)^5 + 1080*C*b^2*c*d^2*f*x*tan(f*x)^5*tan(e)
^5 - 360*B*a^2*d^3*f*x*tan(f*x)^5*tan(e)^5 - 720*A*a*b*d^3*f*x*tan(f*x)^5*t
an(e)^5 + 720*C*a*b*d^3*f*x*tan(f*x)^5*tan(e)^5 + 360*B*b^2*d^3*f*x*tan(f*x
)^5*tan(e)^5 + 60*C*a*b*c^3*tan(f*x)^6*tan(e)^6 + 30*B*b^2*c^3*tan(f*x)^6*t
an(e)^6 + 90*C*a^2*c^2*d*tan(f*x)^6*tan(e)^6 + 180*B*a*b*c^2*d*tan(f*x)^6*t
an(e)^6 + 90*A*b^2*c^2*d*tan(f*x)^6*tan(e)^6 - 135*C*b^2*c^2*d*tan(f*x)^6*t
an(e)^6 + 90*B*a^2*c*d^2*tan(f*x)^6*tan(e)^6 + 180*A*a*b*c*d^2*tan(f*x)^6*t
an(e)^6 - 270*C*a*b*c*d^2*tan(f*x)^6*tan(e)^6 - 135*B*b^2*c*d^2*tan(f*x)^6*
tan(e)^6 + 30*A*a^2*d^3*tan(f*x)^6*tan(e)^6 - 45*C*a^2*d^3*tan(f*x)^6*tan(e
)^6 - 90*B*a*b*d^3*tan(f*x)^6*tan(e)^6 - 45*A*b^2*d^3*tan(f*x)^6*tan(e)^6 +
55*C*b^2*d^3*tan(f*x)^6*tan(e)^6 + 180*B*a^2*c^3*log(4*(tan(f*x)^2*tan(e)^
2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1
))*tan(f*x)^5*tan(e)^5 + 360*A*a*b*c^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f
*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)
^5*tan(e)^5 - 360*C*a*b*c^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e)
+ 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5
- 180*B*b^2*c^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f
*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5 + 540*A*a^
2*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan

(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5 - 540*C*a^2*c^2*d*l
og(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + t
an(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5 - 1080*B*a*b*c^2*d*log(4*(ta
n(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^
2 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5 - 540*A*b^2*c^2*d*log(4*(tan(f*x)^2*
tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e
)^2 + 1))*tan(f*x)^5*tan(e)^5 + 540*C*b^2*c^2*d*log(4*(tan(f*x)^2*tan(e)^2
- 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))
*tan(f*x)^5*tan(e)^5 - 540*B*a^2*c*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f
*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)
^5*tan(e)^5 - 1080*A*a*b*c*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(
e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5*tan(e
)^5 + 1080*C*a*b*c*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/
(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5 + 54
0*B*b^2*c*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)
^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5 - 180*A*a^2*d
^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2
 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5 + 180*C*a^2*d^3*log(4*(t
an(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)
^2 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5 + 360*B*a*b*d^3*log(4*(tan(f*x)^2*t
an(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)
^2 + 1))*tan(f*x)^5*tan(e)^5 + 180*A*b^2*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2
*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*ta
n(f*x)^5*tan(e)^5 - 180*C*b^2*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*t
an(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5*ta
n(e)^5 - 60*C*a^2*c^3*tan(f*x)^6*tan(e)^5 - 120*B*a*b*c^3*tan(f*x)^6*tan(e)
^5 - 60*A*b^2*c^3*tan(f*x)^6*tan(e)^5 + 60*C*b^2*c^3*tan(f*x)^6*tan(e)^5 -
180*B*a^2*c^2*d*tan(f*x)^6*tan(e)^5 - 360*A*a*b*c^2*d*tan(f*x)^6*tan(e)^5 +
 360*C*a*b*c^2*d*tan(f*x)^6*tan(e)^5 + 180*B*b^2*c^2*d*tan(f*x)^6*tan(e)^5
- 180*A*a^2*c*d^2*tan(f*x)^6*tan(e)^5 + 180*C*a^2*c*d^2*tan(f*x)^6*tan(e)^5
 + 360*B*a*b*c*d^2*tan(f*x)^6*tan(e)^5 + 180*A*b^2*c*d^2*tan(f*x)^6*tan(e)^
5 - 180*C*b^2*c*d^2*tan(f*x)^6*tan(e)^5 + 60*B*a^2*d^3*tan(f*x)^6*tan(e)^5
+ 120*A*a*b*d^3*tan(f*x)^6*tan(e)^5 - 120*C*a*b*d^3*tan(f*x)^6*tan(e)^5 - 6
0*B*b^2*d^3*tan(f*x)^6*tan(e)^5 - 60*C*a^2*c^3*tan(f*x)^5*tan(e)^6 - 120*B*
a*b*c^3*tan(f*x)^5*tan(e)^6 - 60*A*b^2*c^3*tan(f*x)^5*tan(e)^6 + 60*C*b^2*c
^3*tan(f*x)^5*tan(e)^6 - 180*B*a^2*c^2*d*tan(f*x)^5*tan(e)^6 - 360*A*a*b*c^
2*d*tan(f*x)^5*tan(e)^6 + 360*C*a*b*c^2*d*tan(f*x)^5*tan(e)^6 + 180*B*b^2*c
^2*d*tan(f*x)^5*tan(e)^6 - 180*A*a^2*c*d^2*tan(f*x)^5*tan(e)^6 + 180*C*a^2*
c*d^2*tan(f*x)^5*tan(e)^6 + 360*B*a*b*c*d^2*tan(f*x)^5*tan(e)^6 + 180*A*b^2
*c*d^2*tan(f*x)^5*tan(e)^6 - 180*C*b^2*c*d^2*tan(f*x)^5*tan(e)^6 + 60*B*a^2
*d^3*tan(f*x)^5*tan(e)^6 + 120*A*a*b*d^3*tan(f*x)^5*tan(e)^6 - 120*C*a*b*d^
3*tan(f*x)^5*tan(e)^6 - 60*B*b^2*d^3*tan(f*x)^5*tan(e)^6 + 900*A*a^2*c^3*f*
x*tan(f*x)^4*tan(e)^4 - 900*C*a^2*c^3*f*x*tan(f*x)^4*tan(e)^4 - 1800*B*a*b*
c^3*f*x*tan(f*x)^4*tan(e)^4 - 900*A*b^2*c^3*f*x*tan(f*x)^4*tan(e)^4 + 900*C
*b^2*c^3*f*x*tan(f*x)^4*tan(e)^4 - 2700*B*a^2*c^2*d*f*x*tan(f*x)^4*tan(e)^4

- 5400*A*a*b*c^2*d*f*x*tan(f*x)^4*tan(e)^4 + 5400*C*a*b*c^2*d*f*x*tan(f*x)
^4*tan(e)^4 + 2700*B*b^2*c^2*d*f*x*tan(f*x)^4*tan(e)^4 - 2700*A*a^2*c*d^2*f
*x*tan(f*x)^4*tan(e)^4 + 2700*C*a^2*c*d^2*f*x*tan(f*x)^4*tan(e)^4 + 5400*B*
a*b*c*d^2*f*x*tan(f*x)^4*tan(e)^4 + 2700*A*b^2*c*d^2*f*x*tan(f*x)^4*tan(e)^
4 - 2700*C*b^2*c*d^2*f*x*tan(f*x)^4*tan(e)^4 + 900*B*a^2*d^3*f*x*tan(f*x)^4
*tan(e)^4 + 1800*A*a*b*d^3*f*x*tan(f*x)^4*tan(e)^4 - 1800*C*a*b*d^3*f*x*tan
(f*x)^4*tan(e)^4 - 900*B*b^2*d^3*f*x*tan(f*x)^4*tan(e)^4 + 60*C*a*b*c^3*tan
(f*x)^6*tan(e)^4 + 30*B*b^2*c^3*tan(f*x)^6*tan(e)^4 + 90*C*a^2*c^2*d*tan(f*
x)^6*tan(e)^4 + 180*B*a*b*c^2*d*tan(f*x)^6*tan(e)^4 + 90*A*b^2*c^2*d*tan(f*
x)^6*tan(e)^4 - 90*C*b^2*c^2*d*tan(f*x)^6*tan(e)^4 + 90*B*a^2*c*d^2*tan(f*x
)^6*tan(e)^4 + 180*A*a*b*c*d^2*tan(f*x)^6*tan(e)^4 - 180*C*a*b*c*d^2*tan(f*
x)^6*tan(e)^4 - 90*B*b^2*c*d^2*tan(f*x)^6*tan(e)^4 + 30*A*a^2*d^3*tan(f*x)^
6*tan(e)^4 - 30*C*a^2*d^3*tan(f*x)^6*tan(e)^4 - 60*B*a*b*d^3*tan(f*x)^6*tan
(e)^4 - 30*A*b^2*d^3*tan(f*x)^6*tan(e)^4 + 30*C*b^2*d^3*tan(f*x)^6*tan(e)^4
- 240*C*a*b*c^3*tan(f*x)^5*tan(e)^5 - 120*B*b^2*c^3*tan(f*x)^5*tan(e)^5 -
360*C*a^2*c^2*d*tan(f*x)^5*tan(e)^5 - 720*B*a*b*c^2*d*tan(f*x)^5*tan(e)^5 -
360*A*b^2*c^2*d*tan(f*x)^5*tan(e)^5 + 630*C*b^2*c^2*d*tan(f*x)^5*tan(e)^5
- 360*B*a^2*c*d^2*tan(f*x)^5*tan(e)^5 - 720*A*a*b*c*d^2*tan(f*x)^5*tan(e)^5
+ 1260*C*a*b*c*d^2*tan(f*x)^5*tan(e)^5 + 630*B*b^2*c*d^2*tan(f*x)^5*tan(e)
^5 - 120*A*a^2*d^3*tan(f*x)^5*tan(e)^5 + 210*C*a^2*d^3*tan(f*x)^5*tan(e)^5
+ 420*B*a*b*d^3*tan(f*x)^5*tan(e)^5 + 210*A*b^2*d^3*tan(f*x)^5*tan(e)^5 - 2
70*C*b^2*d^3*tan(f*x)^5*tan(e)^5 + 60*C*a*b*c^3*tan(f*x)^4*tan(e)^6 + 30*B*
b^2*c^3*tan(f*x)^4*tan(e)^6 + 90*C*a^2*c^2*d*tan(f*x)^4*tan(e)^6 + 180*B*a*
b*c^2*d*tan(f*x)^4*tan(e)^6 + 90*A*b^2*c^2*d*tan(f*x)^4*tan(e)^6 - 90*C*b^2
*c^2*d*tan(f*x)^4*tan(e)^6 + 90*B*a^2*c*d^2*tan(f*x)^4*tan(e)^6 + 180*A*a*b
*c*d^2*tan(f*x)^4*tan(e)^6 - 180*C*a*b*c*d^2*tan(f*x)^4*tan(e)^6 - 90*B*b^2
*c*d^2*tan(f*x)^4*tan(e)^6 + 30*A*a^2*d^3*tan(f*x)^4*tan(e)^6 - 30*C*a^2*d^
3*tan(f*x)^4*tan(e)^6 - 60*B*a*b*d^3*tan(f*x)^4*tan(e)^6 - 30*A*b^2*d^3*tan
(f*x)^4*tan(e)^6 + 30*C*b^2*d^3*tan(f*x)^4*tan(e)^6 - 20*C*b^2*c^3*tan(f*x)
^6*tan(e)^3 - 120*C*a*b*c^2*d*tan(f*x)^6*tan(e)^3 - 60*B*b^2*c^2*d*tan(f*x)
^6*tan(e)^3 - 60*C*a^2*c*d^2*tan(f*x)^6*tan(e)^3 - 120*B*a*b*c*d^2*tan(f*x)
^6*tan(e)^3 - 60*A*b^2*c*d^2*tan(f*x)^6*tan(e)^3 + 60*C*b^2*c*d^2*tan(f*x)^
6*tan(e)^3 - 20*B*a^2*d^3*tan(f*x)^6*tan(e)^3 - 40*A*a*b*d^3*tan(f*x)^6*tan
(e)^3 + 40*C*a*b*d^3*tan(f*x)^6*tan(e)^3 + 20*B*b^2*d^3*tan(f*x)^6*tan(e)^3
- 450*B*a^2*c^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f
*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 - 900*A*a*
b*c^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e
)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 900*C*a*b*c^3*log(4
*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f
*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 450*B*b^2*c^3*log(4*(tan(f*x)^
2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan
(e)^2 + 1))*tan(f*x)^4*tan(e)^4 - 1350*A*a^2*c^2*d*log(4*(tan(f*x)^2*tan(e)
^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 +
1))*tan(f*x)^4*tan(e)^4 + 1350*C*a^2*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*t
an(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(

f*x)^4*tan(e)^4 + 2700*B*a*b*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*
tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*t
an(e)^4 + 1350*A*b^2*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) +
 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4
- 1350*C*b^2*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan
(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 1350*B
*a^2*c*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*
tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 2700*A*a*b*c*d
^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2
 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 - 2700*C*a*b*c*d^2*log(4
*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f
*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 - 1350*B*b^2*c*d^2*log(4*(tan(f*
x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 +
tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 450*A*a^2*d^3*log(4*(tan(f*x)^2*tan(e)
^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 +
1))*tan(f*x)^4*tan(e)^4 - 450*C*a^2*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(
f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x
)^4*tan(e)^4 - 900*B*a*b*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e)
 + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^
4 - 450*A*b^2*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(
f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 450*C*b
^2*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(
e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 300*C*a^2*c^3*tan(
f*x)^5*tan(e)^4 + 600*B*a*b*c^3*tan(f*x)^5*tan(e)^4 + 300*A*b^2*c^3*tan(f*x
)^5*tan(e)^4 - 360*C*b^2*c^3*tan(f*x)^5*tan(e)^4 + 900*B*a^2*c^2*d*tan(f*x)
^5*tan(e)^4 + 1800*A*a*b*c^2*d*tan(f*x)^5*tan(e)^4 - 2160*C*a*b*c^2*d*tan(f
*x)^5*tan(e)^4 - 1080*B*b^2*c^2*d*tan(f*x)^5*tan(e)^4 + 900*A*a^2*c*d^2*tan
(f*x)^5*tan(e)^4 - 1080*C*a^2*c*d^2*tan(f*x)^5*tan(e)^4 - 2160*B*a*b*c*d^2*
tan(f*x)^5*tan(e)^4 - 1080*A*b^2*c*d^2*tan(f*x)^5*tan(e)^4 + 1080*C*b^2*c*d
^2*tan(f*x)^5*tan(e)^4 - 360*B*a^2*d^3*tan(f*x)^5*tan(e)^4 - 720*A*a*b*d^3*
tan(f*x)^5*tan(e)^4 + 720*C*a*b*d^3*tan(f*x)^5*tan(e)^4 + 360*B*b^2*d^3*tan
(f*x)^5*tan(e)^4 + 300*C*a^2*c^3*tan(f*x)^4*tan(e)^5 + 600*B*a*b*c^3*tan(f*
x)^4*tan(e)^5 + 300*A*b^2*c^3*tan(f*x)^4*tan(e)^5 - 360*C*b^2*c^3*tan(f*x)^
4*tan(e)^5 + 900*B*a^2*c^2*d*tan(f*x)^4*tan(e)^5 + 1800*A*a*b*c^2*d*tan(f*x
)^4*tan(e)^5 - 2160*C*a*b*c^2*d*tan(f*x)^4*tan(e)^5 - 1080*B*b^2*c^2*d*tan(
f*x)^4*tan(e)^5 + 900*A*a^2*c*d^2*tan(f*x)^4*tan(e)^5 - 1080*C*a^2*c*d^2*ta
n(f*x)^4*tan(e)^5 - 2160*B*a*b*c*d^2*tan(f*x)^4*tan(e)^5 - 1080*A*b^2*c*d^2
*tan(f*x)^4*tan(e)^5 + 1080*C*b^2*c*d^2*tan(f*x)^4*tan(e)^5 - 360*B*a^2*d^3
*tan(f*x)^4*tan(e)^5 - 720*A*a*b*d^3*tan(f*x)^4*tan(e)^5 + 720*C*a*b*d^3*ta
n(f*x)^4*tan(e)^5 + 360*B*b^2*d^3*tan(f*x)^4*tan(e)^5 - 20*C*b^2*c^3*tan(f*
x)^3*tan(e)^6 - 120*C*a*b*c^2*d*tan(f*x)^3*tan(e)^6 - 60*B*b^2*c^2*d*tan(f*
x)^3*tan(e)^6 - 60*C*a^2*c*d^2*tan(f*x)^3*tan(e)^6 - 120*B*a*b*c*d^2*tan(f*
x)^3*tan(e)^6 - 60*A*b^2*c*d^2*tan(f*x)^3*tan(e)^6 + 60*C*b^2*c*d^2*tan(f*x
)^3*tan(e)^6 - 20*B*a^2*d^3*tan(f*x)^3*tan(e)^6 - 40*A*a*b*d^3*tan(f*x)^3*t
an(e)^6 + 40*C*a*b*d^3*tan(f*x)^3*tan(e)^6 + 20*B*b^2*d^3*tan(f*x)^3*tan(e)

^6 + 45*C*b^2*c^2*d*tan(f*x)^6*tan(e)^2 + 90*C*a*b*c*d^2*tan(f*x)^6*tan(e)^
2 + 45*B*b^2*c*d^2*tan(f*x)^6*tan(e)^2 + 15*C*a^2*d^3*tan(f*x)^6*tan(e)^2 +
 30*B*a*b*d^3*tan(f*x)^6*tan(e)^2 + 15*A*b^2*d^3*tan(f*x)^6*tan(e)^2 - 15*C
*b^2*d^3*tan(f*x)^6*tan(e)^2 - 1200*A*a^2*c^3*f*x*tan(f*x)^3*tan(e)^3 + 120
0*C*a^2*c^3*f*x*tan(f*x)^3*tan(e)^3 + 2400*B*a*b*c^3*f*x*tan(f*x)^3*tan(e)^
3 + 1200*A*b^2*c^3*f*x*tan(f*x)^3*tan(e)^3 - 1200*C*b^2*c^3*f*x*tan(f*x)^3*
tan(e)^3 + 3600*B*a^2*c^2*d*f*x*tan(f*x)^3*tan(e)^3 + 7200*A*a*b*c^2*d*f*x*
tan(f*x)^3*tan(e)^3 - 7200*C*a*b*c^2*d*f*x*tan(f*x)^3*tan(e)^3 - 3600*B*b^2
*c^2*d*f*x*tan(f*x)^3*tan(e)^3 + 3600*A*a^2*c*d^2*f*x*tan(f*x)^3*tan(e)^3 -
 3600*C*a^2*c*d^2*f*x*tan(f*x)^3*tan(e)^3 - 7200*B*a*b*c*d^2*f*x*tan(f*x)^3
*tan(e)^3 - 3600*A*b^2*c*d^2*f*x*tan(f*x)^3*tan(e)^3 + 3600*C*b^2*c*d^2*f*x
*tan(f*x)^3*tan(e)^3 - 1200*B*a^2*d^3*f*x*tan(f*x)^3*tan(e)^3 - 2400*A*a*b*
d^3*f*x*tan(f*x)^3*tan(e)^3 + 2400*C*a*b*d^3*f*x*tan(f*x)^3*tan(e)^3 + 1200
*B*b^2*d^3*f*x*tan(f*x)^3*tan(e)^3 - 240*C*a*b*c^3*tan(f*x)^5*tan(e)^3 - 12
0*B*b^2*c^3*tan(f*x)^5*tan(e)^3 - 360*C*a^2*c^2*d*tan(f*x)^5*tan(e)^3 - 720
*B*a*b*c^2*d*tan(f*x)^5*tan(e)^3 - 360*A*b^2*c^2*d*tan(f*x)^5*tan(e)^3 + 54
0*C*b^2*c^2*d*tan(f*x)^5*tan(e)^3 - 360*B*a^2*c*d^2*tan(f*x)^5*tan(e)^3 - 7
20*A*a*b*c*d^2*tan(f*x)^5*tan(e)^3 + 1080*C*a*b*c*d^2*tan(f*x)^5*tan(e)^3 +
 540*B*b^2*c*d^2*tan(f*x)^5*tan(e)^3 - 120*A*a^2*d^3*tan(f*x)^5*tan(e)^3 +
180*C*a^2*d^3*tan(f*x)^5*tan(e)^3 + 360*B*a*b*d^3*tan(f*x)^5*tan(e)^3 + 180
*A*b^2*d^3*tan(f*x)^5*tan(e)^3 - 180*C*b^2*d^3*tan(f*x)^5*tan(e)^3 + 420*C*
a*b*c^3*tan(f*x)^4*tan(e)^4 + 210*B*b^2*c^3*tan(f*x)^4*tan(e)^4 + 630*C*a^2
*c^2*d*tan(f*x)^4*tan(e)^4 + 1260*B*a*b*c^2*d*tan(f*x)^4*tan(e)^4 + 630*A*b
^2*c^2*d*tan(f*x)^4*tan(e)^4 - 1035*C*b^2*c^2*d*tan(f*x)^4*tan(e)^4 + 630*B
*a^2*c*d^2*tan(f*x)^4*tan(e)^4 + 1260*A*a*b*c*d^2*tan(f*x)^4*tan(e)^4 - 207
0*C*a*b*c*d^2*tan(f*x)^4*tan(e)^4 - 1035*B*b^2*c*d^2*tan(f*x)^4*tan(e)^4 +
210*A*a^2*d^3*tan(f*x)^4*tan(e)^4 - 345*C*a^2*d^3*tan(f*x)^4*tan(e)^4 - 690
*B*a*b*d^3*tan(f*x)^4*tan(e)^4 - 345*A*b^2*d^3*tan(f*x)^4*tan(e)^4 + 495*C*
b^2*d^3*tan(f*x)^4*tan(e)^4 - 240*C*a*b*c^3*tan(f*x)^3*tan(e)^5 - 120*B*b^2
*c^3*tan(f*x)^3*tan(e)^5 - 360*C*a^2*c^2*d*tan(f*x)^3*tan(e)^5 - 720*B*a*b*
c^2*d*tan(f*x)^3*tan(e)^5 - 360*A*b^2*c^2*d*tan(f*x)^3*tan(e)^5 + 540*C*b^2
*c^2*d*tan(f*x)^3*tan(e)^5 - 360*B*a^2*c*d^2*tan(f*x)^3*tan(e)^5 - 720*A*a*
b*c*d^2*tan(f*x)^3*tan(e)^5 + 1080*C*a*b*c*d^2*tan(f*x)^3*tan(e)^5 + 540*B*
b^2*c*d^2*tan(f*x)^3*tan(e)^5 - 120*A*a^2*d^3*tan(f*x)^3*tan(e)^5 + 180*C*a
^2*d^3*tan(f*x)^3*tan(e)^5 + 360*B*a*b*d^3*tan(f*x)^3*tan(e)^5 + 180*A*b^2*
d^3*tan(f*x)^3*tan(e)^5 - 180*C*b^2*d^3*tan(f*x)^3*tan(e)^5 + 45*C*b^2*c^2*
d*tan(f*x)^2*tan(e)^6 + 90*C*a*b*c*d^2*tan(f*x)^2*tan(e)^6 + 45*B*b^2*c*d^2
*tan(f*x)^2*tan(e)^6 + 15*C*a^2*d^3*tan(f*x)^2*tan(e)^6 + 30*B*a*b*d^3*tan(
f*x)^2*tan(e)^6 + 15*A*b^2*d^3*tan(f*x)^2*tan(e)^6 - 15*C*b^2*d^3*tan(f*x)^
2*tan(e)^6 - 36*C*b^2*c*d^2*tan(f*x)^6*tan(e) - 24*C*a*b*d^3*tan(f*x)^6*tan
(e) - 12*B*b^2*d^3*tan(f*x)^6*tan(e) + 60*C*b^2*c^3*tan(f*x)^5*tan(e)^2 + 3
60*C*a*b*c^2*d*tan(f*x)^5*tan(e)^2 + 180*B*b^2*c^2*d*tan(f*x)^5*tan(e)^2 +
180*C*a^2*c*d^2*tan(f*x)^5*tan(e)^2 + 360*B*a*b*c*d^2*tan(f*x)^5*tan(e)^2 +
 180*A*b^2*c*d^2*tan(f*x)^5*tan(e)^2 - 360*C*b^2*c*d^2*tan(f*x)^5*tan(e)^2
+ 60*B*a^2*d^3*tan(f*x)^5*tan(e)^2 + 120*A*a*b*d^3*tan(f*x)^5*tan(e)^2 - 24

0*C*a*b*d^3*tan(f*x)^5*tan(e)^2 - 120*B*b^2*d^3*tan(f*x)^5*tan(e)^2 + 600*B *a^2*c^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*ta n(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 1200*A*a*b*c^3*l og(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + t an(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 1200*C*a*b*c^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 600*B*b^2*c^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 1800*A*a^2*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2 *tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*ta n(f*x)^3*tan(e)^3 - 1800*C*a^2*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3 *tan(e)^3 - 3600*B*a*b*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^ 3 - 1800*A*b^2*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(t an(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 1800 *C*b^2*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^ 2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 1800*B*a^2*c *d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e) ^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 3600*A*a*b*c*d^2*log (4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan (f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 3600*C*a*b*c*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 1800*B*b^2*c*d^2*log(4*(tan(f*x)^2*t an(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e) ^2 + 1))*tan(f*x)^3*tan(e)^3 - 600*A*a^2*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2 *tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*ta n(f*x)^3*tan(e)^3 + 600*C*a^2*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*t an(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*ta n(e)^3 + 1200*B*a*b*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1) /(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 6 00*A*b^2*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^ 2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 600*C*b^2*d^ 3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 600*C*a^2*c^3*tan(f*x)^ 4*tan(e)^3 - 1200*B*a*b*c^3*tan(f*x)^4*tan(e)^3 - 600*A*b^2*c^3*tan(f*x)^4* tan(e)^3 + 780*C*b^2*c^3*tan(f*x)^4*tan(e)^3 - 1800*B*a^2*c^2*d*tan(f*x)^4* tan(e)^3 - 3600*A*a*b*c^2*d*tan(f*x)^4*tan(e)^3 + 4680*C*a*b*c^2*d*tan(f*x) ^4*tan(e)^3 + 2340*B*b^2*c^2*d*tan(f*x)^4*tan(e)^3 - 1800*A*a^2*c*d^2*tan(f *x)^4*tan(e)^3 + 2340*C*a^2*c*d^2*tan(f*x)^4*tan(e)^3 + 4680*B*a*b*c*d^2*ta n(f*x)^4*tan(e)^3 + 2340*A*b^2*c*d^2*tan(f*x)^4*tan(e)^3 - 2700*C*b^2*c*d^2 *tan(f*x)^4*tan(e)^3 + 780*B*a^2*d^3*tan(f*x)^4*tan(e)^3 + 1560*A*a*b*d^3*t an(f*x)^4*tan(e)^3 - 1800*C*a*b*d^3*tan(f*x)^4*tan(e)^3 - 900*B*b^2*d^3*tan (f*x)^4*tan(e)^3 - 600*C*a^2*c^3*tan(f*x)^3*tan(e)^4 - 1200*B*a*b*c^3*tan(f *x)^3*tan(e)^4 - 600*A*b^2*c^3*tan(f*x)^3*tan(e)^4 + 780*C*b^2*c^3*tan(f*x)

^3*tan(e)^4 - 1800*B*a^2*c^2*d*tan(f*x)^3*tan(e)^4 - 3600*A*a*b*c^2*d*tan(f*x)^3*tan(e)^4 + 4680*C*a*b*c^2*d*tan(f*x)^3*tan(e)^4 + 2340*B*b^2*c^2*d*tan(f*x)^3*tan(e)^4 - 1800*A*a^2*c*d^2*tan(f*x)^3*tan(e)^4 + 2340*C*a^2*c*d^2*tan(f*x)^3*tan(e)^4 + 4680*B*a*b*c*d^2*tan(f*x)^3*tan(e)^4 + 2340*A*b^2*c*d^2*tan(f*x)^3*tan(e)^4 - 2700*C*b^2*c*d^2*tan(f*x)^3*tan(e)^4 + 780*B*a^2*d^3*tan(f*x)^3*tan(e)^4 + 1560*A*a*b*d^3*tan(f*x)^3*tan(e)^4 - 1800*C*a*b*d^3*tan(f*x)^3*tan(e)^4 - 900*B*b^2*d^3*tan(f*x)^3*tan(e)^4 + 60*C*b^2*c^3*tan(f*x)^2*tan(e)^5 + 360*C*a*b*c^2*d*tan(f*x)^2*tan(e)^5 + 180*B*b^2*c^2*d*tan(f*x)^2*tan(e)^5 + 180*C*a^2*c*d^2*tan(f*x)^2*tan(e)^5 + 360*B*a*b*c*d^2*tan(f*x)^2*tan(e)^5 + 180*A*b^2*c*d^2*tan(f*x)^2*tan(e)^5 - 360*C*b^2*c*d^2*tan(f*x)^2*tan(e)^5 + 60*B*a^2*d^3*tan(f*x)^2*tan(e)^5 + 120*A*a*b*d^3*tan(f*x)^2*tan(e)^5 - 240*C*a*b*d^3*tan(f*x)^2*tan(e)^5 - 120*B*b^2*d^3*tan(f*x)^2*tan(e)^5 - 36*C*b^2*c*d^2*tan(f*x)*tan(e)^6 - 24*C*a*b*d^3*tan(f*x)*tan(e)^6 - 12*B*b^2*d^3*tan(f*x)*tan(e)^6 + 10*C*b^2*d^3*tan(f*x)^6 - 90*C*b^2*c^2*d*tan(f*x)^5*tan(e) - 180*C*a*b*c*d^2*tan(f*x)^5*tan(e) - 90*B*b^2*c*d^2*tan(f*x)^5*tan(e) - 30*C*a^2*d^3*tan(f*x)^5*tan(e) - 60*B*a*b*d^3*tan(f*x)^5*tan(e) - 30*A*b^2*d^3*tan(f*x)^5*tan(e) + 90*C*b^2*d^3*tan(f*x)^5*tan(e) + 900*A*a^2*c^3*f*x*tan(f*x)^2*tan(e)^2 - 900*C*a^2*c^3*f*x*tan(f*x)^2*tan(e)^2 - 1800*B*a*b*c^3*f*x*tan(f*x)^2*tan(e)^2 - 900*A*b^2*c^3*f*x*tan(f*x)^2*tan(e)^2 + 900*C*b^2*c^3*f*x*tan(f*x)^2*tan(e)^2 - 2700*B*a^2*c^2*d*f*x*tan(f*x)^2*tan(e)^2 - 5400*A*a*b*c^2*d*f*x*tan(f*x)^2*tan(e)^2 + 5400*C*a*b*c^2*d*f*x*tan(f*x)^2*tan(e)^2 + 2700*B*b^2*c^2*d*f*x*tan(f*x)^2*tan(e)^2 - 2700*A*a^2*c*d^2*f*x*tan(f*x)^2*tan(e)^2 + 2700*C*a^2*c*d^2*f*x*tan(f*x)^2*tan(e)^2 + 5400*B*a*b*c*d^2*f*x*tan(f*x)^2*tan(e)^2 + 2700*A*b^2*c*d^2*f*x*tan(f*x)^2*tan(e)^2 - 2700*C*b^2*c*d^2*f*x*tan(f*x)^2*tan(e)^2 + 900*B*a^2*d^3*f*x*tan(f*x)^2*tan(e)^2 + 1800*A*a*b*d^3*f*x*tan(f*x)^2*tan(e)^2 - 1800*C*a*b*d^3*f*x*tan(f*x)^2*tan(e)^2 - 900*B*b^2*d^3*f*x*tan(f*x)^2*tan(e)^2 + 360*C*a*b*c^3*tan(f*x)^4*tan(e)^2 + 180*B*b^2*c^3*tan(f*x)^4*tan(e)^2 + 540*C*a^2*c^2*d*tan(f*x)^4*tan(e)^2 + 1080*B*a*b*c^2*d*tan(f*x)^4*tan(e)^2 + 540*A*b^2*c^2*d*tan(f*x)^4*tan(e)^2 - 900*C*b^2*c^2*d*tan(f*x)^4*tan(e)^2 + 540*B*a^2*c*d^2*tan(f*x)^4*tan(e)^2 + 1080*A*a*b*c*d^2*tan(f*x)^4*tan(e)^2 - 1800*C*a*b*c*d^2*tan(f*x)^4*tan(e)^2 - 900*B*b^2*c*d^2*tan(f*x)^4*tan(e)^2 + 180*A*a^2*d^3*tan(f*x)^4*tan(e)^2 - 300*C*a^2*d^3*tan(f*x)^4*tan(e)^2 - 600*B*a*b*d^3*tan(f*x)^4*tan(e)^2 - 300*A*b^2*d^3*tan(f*x)^4*tan(e)^2 + 450*C*b^2*d^3*tan(f*x)^4*tan(e)^2 - 480*C*a*b*c^3*tan(f*x)^3*tan(e)^3 - 240*B*b^2*c^3*tan(f*x)^3*tan(e)^3 - 720*C*a^2*c^2*d*tan(f*x)^3*tan(e)^3 - 1440*B*a*b*c^2*d*tan(f*x)^3*tan(e)^3 - 720*A*b^2*c^2*d*tan(f*x)^3*tan(e)^3 + 1080*C*b^2*c^2*d*tan(f*x)^3*tan(e)^3 - 720*B*a^2*c*d^2*tan(f*x)^3*tan(e)^3 - 1440*A*a*b*c*d^2*tan(f*x)^3*tan(e)^3 + 2160*C*a*b*c*d^2*tan(f*x)^3*tan(e)^3 + 1080*B*b^2*c*d^2*tan(f*x)^3*tan(e)^3 - 240*A*a^2*d^3*tan(f*x)^3*tan(e)^3 + 360*C*a^2*d^3*tan(f*x)^3*tan(e)^3 + 720*B*a*b*d^3*tan(f*x)^3*tan(e)^3 + 360*A*b^2*d^3*tan(f*x)^3*tan(e)^3 - 360*C*b^2*d^3*tan(f*x)^3*tan(e)^3 + 360*C*a*b*c^3*tan(f*x)^2*tan(e)^4 + 180*B*b^2*c^3*tan(f*x)^2*tan(e)^4 + 540*C*a^2*c^2*d*tan(f*x)^2*tan(e)^4 + 1080*B*a*b*c^2*d*tan(f*x)^2*tan(e)^4 + 540*A*b^2*c^2*d*tan(f*x)^2*tan(e)^4 - 900*C*b^2*c^2*d*tan(f*x)^2*tan(e)^4

+ 540*B*a^2*c*d^2*tan(f*x)^2*tan(e)^4 + 1080*A*a*b*c*d^2*tan(f*x)^2*tan(e)^
4 - 1800*C*a*b*c*d^2*tan(f*x)^2*tan(e)^4 - 900*B*b^2*c*d^2*tan(f*x)^2*tan(e
)^4 + 180*A*a^2*d^3*tan(f*x)^2*tan(e)^4 - 300*C*a^2*d^3*tan(f*x)^2*tan(e)^4
- 600*B*a*b*d^3*tan(f*x)^2*tan(e)^4 - 300*A*b^2*d^3*tan(f*x)^2*tan(e)^4 +
450*C*b^2*d^3*tan(f*x)^2*tan(e)^4 - 90*C*b^2*c^2*d*tan(f*x)*tan(e)^5 - 180*
C*a*b*c*d^2*tan(f*x)*tan(e)^5 - 90*B*b^2*c*d^2*tan(f*x)*tan(e)^5 - 30*C*a^2
*d^3*tan(f*x)*tan(e)^5 - 60*B*a*b*d^3*tan(f*x)*tan(e)^5 - 30*A*b^2*d^3*tan(
f*x)*tan(e)^5 + 90*C*b^2*d^3*tan(f*x)*tan(e)^5 + 10*C*b^2*d^3*tan(e)^6 + 36
*C*b^2*c*d^2*tan(f*x)^5 + 24*C*a*b*d^3*tan(f*x)^5 + 12*B*b^2*d^3*tan(f*x)^5
- 60*C*b^2*c^3*tan(f*x)^4*tan(e) - 360*C*a*b*c^2*d*tan(f*x)^4*tan(e) - 180
*B*b^2*c^2*d*tan(f*x)^4*tan(e) - 180*C*a^2*c*d^2*tan(f*x)^4*tan(e) - 360*B*
a*b*c*d^2*tan(f*x)^4*tan(e) - 180*A*b^2*c*d^2*tan(f*x)^4*tan(e) + 360*C*b^2
*c*d^2*tan(f*x)^4*tan(e) - 60*B*a^2*d^3*tan(f*x)^4*tan(e) - 120*A*a*b*d^3*t
an(f*x)^4*tan(e) + 240*C*a*b*d^3*tan(f*x)^4*tan(e) + 120*B*b^2*d^3*tan(f*x)
^4*tan(e) - 450*B*a^2*c^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) +
1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 -
900*A*a*b*c^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x
)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 900*C*a*b*
c^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^
2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 450*B*b^2*c^3*log(4*(
tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x
)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 1350*A*a^2*c^2*d*log(4*(tan(f*x)
^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + ta
n(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 1350*C*a^2*c^2*d*log(4*(tan(f*x)^2*tan(e
)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 +
1))*tan(f*x)^2*tan(e)^2 + 2700*B*a*b*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*
tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan
(f*x)^2*tan(e)^2 + 1350*A*b^2*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)
*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*
tan(e)^2 - 1350*C*b^2*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e)
+ 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2
+ 1350*B*a^2*c*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(ta
n(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 2700*
A*a*b*c*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2
*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 2700*C*a*b*c*
d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^
2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 1350*B*b^2*c*d^2*log(
4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(
f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 450*A*a^2*d^3*log(4*(tan(f*x)
^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + ta
n(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 450*C*a^2*d^3*log(4*(tan(f*x)^2*tan(e)^2
- 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)
)*tan(f*x)^2*tan(e)^2 - 900*B*a*b*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*
x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^
2*tan(e)^2 - 450*A*b^2*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) +

1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2
+ 450*C*b^2*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*
x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 600*C*a^2
*c^3*tan(f*x)^3*tan(e)^2 + 1200*B*a*b*c^3*tan(f*x)^3*tan(e)^2 + 600*A*b^2*c
^3*tan(f*x)^3*tan(e)^2 - 780*C*b^2*c^3*tan(f*x)^3*tan(e)^2 + 1800*B*a^2*c^2
*d*tan(f*x)^3*tan(e)^2 + 3600*A*a*b*c^2*d*tan(f*x)^3*tan(e)^2 - 4680*C*a*b*
c^2*d*tan(f*x)^3*tan(e)^2 - 2340*B*b^2*c^2*d*tan(f*x)^3*tan(e)^2 + 1800*A*a
^2*c*d^2*tan(f*x)^3*tan(e)^2 - 2340*C*a^2*c*d^2*tan(f*x)^3*tan(e)^2 - 4680*
B*a*b*c*d^2*tan(f*x)^3*tan(e)^2 - 2340*A*b^2*c*d^2*tan(f*x)^3*tan(e)^2 + 27
00*C*b^2*c*d^2*tan(f*x)^3*tan(e)^2 - 780*B*a^2*d^3*tan(f*x)^3*tan(e)^2 - 15
60*A*a*b*d^3*tan(f*x)^3*tan(e)^2 + 1800*C*a*b*d^3*tan(f*x)^3*tan(e)^2 + 900
*B*b^2*d^3*tan(f*x)^3*tan(e)^2 + 600*C*a^2*c^3*tan(f*x)^2*tan(e)^3 + 1200*B
*a*b*c^3*tan(f*x)^2*tan(e)^3 + 600*A*b^2*c^3*tan(f*x)^2*tan(e)^3 - 780*C*b^
2*c^3*tan(f*x)^2*tan(e)^3 + 1800*B*a^2*c^2*d*tan(f*x)^2*tan(e)^3 + 3600*A*a
*b*c^2*d*tan(f*x)^2*tan(e)^3 - 4680*C*a*b*c^2*d*tan(f*x)^2*tan(e)^3 - 2340*
B*b^2*c^2*d*tan(f*x)^2*tan(e)^3 + 1800*A*a^2*c*d^2*tan(f*x)^2*tan(e)^3 - 23
40*C*a^2*c*d^2*tan(f*x)^2*tan(e)^3 - 4680*B*a*b*c*d^2*tan(f*x)^2*tan(e)^3 -
2340*A*b^2*c*d^2*tan(f*x)^2*tan(e)^3 + 2700*C*b^2*c*d^2*tan(f*x)^2*tan(e)^
3 - 780*B*a^2*d^3*tan(f*x)^2*tan(e)^3 - 1560*A*a*b*d^3*tan(f*x)^2*tan(e)^3
+ 1800*C*a*b*d^3*tan(f*x)^2*tan(e)^3 + 900*B*b^2*d^3*tan(f*x)^2*tan(e)^3 -
60*C*b^2*c^3*tan(f*x)*tan(e)^4 - 360*C*a*b*c^2*d*tan(f*x)*tan(e)^4 - 180*B*
b^2*c^2*d*tan(f*x)*tan(e)^4 - 180*C*a^2*c*d^2*tan(f*x)*tan(e)^4 - 360*B*a*b
*c*d^2*tan(f*x)*tan(e)^4 - 180*A*b^2*c*d^2*tan(f*x)*tan(e)^4 + 360*C*b^2*c*
d^2*tan(f*x)*tan(e)^4 - 60*B*a^2*d^3*tan(f*x)*tan(e)^4 - 120*A*a*b*d^3*tan(
f*x)*tan(e)^4 + 240*C*a*b*d^3*tan(f*x)*tan(e)^4 + 120*B*b^2*d^3*tan(f*x)*ta
n(e)^4 + 36*C*b^2*c*d^2*tan(e)^5 + 24*C*a*b*d^3*tan(e)^5 + 12*B*b^2*d^3*tan
(e)^5 + 45*C*b^2*c^2*d*tan(f*x)^4 + 90*C*a*b*c*d^2*tan(f*x)^4 + 45*B*b^2*c*
d^2*tan(f*x)^4 + 15*C*a^2*d^3*tan(f*x)^4 + 30*B*a*b*d^3*tan(f*x)^4 + 15*A*b
^2*d^3*tan(f*x)^4 - 15*C*b^2*d^3*tan(f*x)^4 - 360*A*a^2*c^3*f*x*tan(f*x)*ta
n(e) + 360*C*a^2*c^3*f*x*tan(f*x)*tan(e) + 720*B*a*b*c^3*f*x*tan(f*x)*tan(e
) + 360*A*b^2*c^3*f*x*tan(f*x)*tan(e) - 360*C*b^2*c^3*f*x*tan(f*x)*tan(e) +
1080*B*a^2*c^2*d*f*x*tan(f*x)*tan(e) + 2160*A*a*b*c^2*d*f*x*tan(f*x)*tan(e
) - 2160*C*a*b*c^2*d*f*x*tan(f*x)*tan(e) - 1080*B*b^2*c^2*d*f*x*tan(f*x)*ta
n(e) + 1080*A*a^2*c*d^2*f*x*tan(f*x)*tan(e) - 1080*C*a^2*c*d^2*f*x*tan(f*x)
*tan(e) - 2160*B*a*b*c*d^2*f*x*tan(f*x)*tan(e) - 1080*A*b^2*c*d^2*f*x*tan(f
*x)*tan(e) + 1080*C*b^2*c*d^2*f*x*tan(f*x)*tan(e) - 360*B*a^2*d^3*f*x*tan(f
*x)*tan(e) - 720*A*a*b*d^3*f*x*tan(f*x)*tan(e) + 720*C*a*b*d^3*f*x*tan(f*x)
*tan(e) + 360*B*b^2*d^3*f*x*tan(f*x)*tan(e) - 240*C*a*b*c^3*tan(f*x)^3*tan(
e) - 120*B*b^2*c^3*tan(f*x)^3*tan(e) - 360*C*a^2*c^2*d*tan(f*x)^3*tan(e) -
720*B*a*b*c^2*d*tan(f*x)^3*tan(e) - 360*A*b^2*c^2*d*tan(f*x)^3*tan(e) + 540
*C*b^2*c^2*d*tan(f*x)^3*tan(e) - 360*B*a^2*c*d^2*tan(f*x)^3*tan(e) - 720*A*
a*b*c*d^2*tan(f*x)^3*tan(e) + 1080*C*a*b*c*d^2*tan(f*x)^3*tan(e) + 540*B*b^
2*c*d^2*tan(f*x)^3*tan(e) - 120*A*a^2*d^3*tan(f*x)^3*tan(e) + 180*C*a^2*d^3
*tan(f*x)^3*tan(e) + 360*B*a*b*d^3*tan(f*x)^3*tan(e) + 180*A*b^2*d^3*tan(f*
x)^3*tan(e) - 180*C*b^2*d^3*tan(f*x)^3*tan(e) + 420*C*a*b*c^3*tan(f*x)^2*ta

n(e)^2 + 210*B*b^2*c^3*tan(f*x)^2*tan(e)^2 + 630*C*a^2*c^2*d*tan(f*x)^2*tan
(e)^2 + 1260*B*a*b*c^2*d*tan(f*x)^2*tan(e)^2 + 630*A*b^2*c^2*d*tan(f*x)^2*t
an(e)^2 - 1035*C*b^2*c^2*d*tan(f*x)^2*tan(e)^2 + 630*B*a^2*c*d^2*tan(f*x)^2
*tan(e)^2 + 1260*A*a*b*c*d^2*tan(f*x)^2*tan(e)^2 - 2070*C*a*b*c*d^2*tan(f*x
)^2*tan(e)^2 - 1035*B*b^2*c*d^2*tan(f*x)^2*tan(e)^2 + 210*A*a^2*d^3*tan(f*x
)^2*tan(e)^2 - 345*C*a^2*d^3*tan(f*x)^2*tan(e)^2 - 690*B*a*b*d^3*tan(f*x)^2
*tan(e)^2 - 345*A*b^2*d^3*tan(f*x)^2*tan(e)^2 + 495*C*b^2*d^3*tan(f*x)^2*ta
n(e)^2 - 240*C*a*b*c^3*tan(f*x)*tan(e)^3 - 120*B*b^2*c^3*tan(f*x)*tan(e)^3
 - 360*C*a^2*c^2*d*tan(f*x)*tan(e)^3 - 720*B*a*b*c^2*d*tan(f*x)*tan(e)^3 - 3
60*A*b^2*c^2*d*tan(f*x)*tan(e)^3 + 540*C*b^2*c^2*d*tan(f*x)*tan(e)^3 - 360*
B*a^2*c*d^2*tan(f*x)*tan(e)^3 - 720*A*a*b*c*d^2*tan(f*x)*tan(e)^3 + 1080*C*
a*b*c*d^2*tan(f*x)*tan(e)^3 + 540*B*b^2*c*d^2*tan(f*x)*tan(e)^3 - 120*A*a^2
*d^3*tan(f*x)*tan(e)^3 + 180*C*a^2*d^3*tan(f*x)*tan(e)^3 + 360*B*a*b*d^3*ta
n(f*x)*tan(e)^3 + 180*A*b^2*d^3*tan(f*x)*tan(e)^3 - 180*C*b^2*d^3*tan(f*x)*
tan(e)^3 + 45*C*b^2*c^2*d*tan(e)^4 + 90*C*a*b*c*d^2*tan(e)^4 + 45*B*b^2*c*d
^2*tan(e)^4 + 15*C*a^2*d^3*tan(e)^4 + 30*B*a*b*d^3*tan(e)^4 + 15*A*b^2*d^3*
tan(e)^4 - 15*C*b^2*d^3*tan(e)^4 + 20*C*b^2*c^3*tan(f*x)^3 + 120*C*a*b*c^2*
d*tan(f*x)^3 + 60*B*b^2*c^2*d*tan(f*x)^3 + 60*C*a^2*c*d^2*tan(f*x)^3 + 120*
B*a*b*c*d^2*tan(f*x)^3 + 60*A*b^2*c*d^2*tan(f*x)^3 - 60*C*b^2*c*d^2*tan(f*x
)^3 + 20*B*a^2*d^3*tan(f*x)^3 + 40*A*a*b*d^3*tan(f*x)^3 - 40*C*a*b*d^3*tan(
f*x)^3 - 20*B*b^2*d^3*tan(f*x)^3 + 180*B*a^2*c^3*log(4*(tan(f*x)^2*tan(e)^2
 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)
)*tan(f*x)*tan(e) + 360*A*a*b*c^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*t
an(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(
e) - 360*C*a*b*c^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan
(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) - 180*B*b^2*
c^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^
2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 540*A*a^2*c^2*d*log(4*(ta
n(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^
2 + tan(e)^2 + 1))*tan(f*x)*tan(e) - 540*C*a^2*c^2*d*log(4*(tan(f*x)^2*tan(
e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2
 + 1))*tan(f*x)*tan(e) - 1080*B*a*b*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan
(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*
x)*tan(e) - 540*A*b^2*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e)
 + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 5
40*C*b^2*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x
)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) - 540*B*a^2*c*d^
2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2
 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) - 1080*A*a*b*c*d^2*log(4*(tan
(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2
 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 1080*C*a*b*c*d^2*log(4*(tan(f*x)^2*tan(
e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2
 + 1))*tan(f*x)*tan(e) + 540*B*b^2*c*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(
f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x
)*tan(e) - 180*A*a^2*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1

)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 180*
C*a^2*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*t
an(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 360*B*a*b*d^3*log(4
*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f
*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 180*A*b^2*d^3*log(4*(tan(f*x)^2*ta
n(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^
2 + 1))*tan(f*x)*tan(e) - 180*C*b^2*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(
f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x
)*tan(e) - 300*C*a^2*c^3*tan(f*x)^2*tan(e) - 600*B*a*b*c^3*tan(f*x)^2*tan(e
) - 300*A*b^2*c^3*tan(f*x)^2*tan(e) + 360*C*b^2*c^3*tan(f*x)^2*tan(e) - 900
*B*a^2*c^2*d*tan(f*x)^2*tan(e) - 1800*A*a*b*c^2*d*tan(f*x)^2*tan(e) + 2160*
C*a*b*c^2*d*tan(f*x)^2*tan(e) + 1080*B*b^2*c^2*d*tan(f*x)^2*tan(e) - 900*A*
a^2*c*d^2*tan(f*x)^2*tan(e) + 1080*C*a^2*c*d^2*tan(f*x)^2*tan(e) + 2160*B*a
*b*c*d^2*tan(f*x)^2*tan(e) + 1080*A*b^2*c*d^2*tan(f*x)^2*tan(e) - 1080*C*b^
2*c*d^2*tan(f*x)^2*tan(e) + 360*B*a^2*d^3*tan(f*x)^2*tan(e) + 720*A*a*b*d^3
*tan(f*x)^2*tan(e) - 720*C*a*b*d^3*tan(f*x)^2*tan(e) - 360*B*b^2*d^3*tan(f*
x)^2*tan(e) - 300*C*a^2*c^3*tan(f*x)*tan(e)^2 - 600*B*a*b*c^3*tan(f*x)*tan(
e)^2 - 300*A*b^2*c^3*tan(f*x)*tan(e)^2 + 360*C*b^2*c^3*tan(f*x)*tan(e)^2 -
900*B*a^2*c^2*d*tan(f*x)*tan(e)^2 - 1800*A*a*b*c^2*d*tan(f*x)*tan(e)^2 + 21
60*C*a*b*c^2*d*tan(f*x)*tan(e)^2 + 1080*B*b^2*c^2*d*tan(f*x)*tan(e)^2 - 900
*A*a^2*c*d^2*tan(f*x)*tan(e)^2 + 1080*C*a^2*c*d^2*tan(f*x)*tan(e)^2 + 2160*
B*a*b*c*d^2*tan(f*x)*tan(e)^2 + 1080*A*b^2*c*d^2*tan(f*x)*tan(e)^2 - 1080*C
*b^2*c*d^2*tan(f*x)*tan(e)^2 + 360*B*a^2*d^3*tan(f*x)*tan(e)^2 + 720*A*a*b*
d^3*tan(f*x)*tan(e)^2 - 720*C*a*b*d^3*tan(f*x)*tan(e)^2 - 360*B*b^2*d^3*tan
(f*x)*tan(e)^2 + 20*C*b^2*c^3*tan(e)^3 + 120*C*a*b*c^2*d*tan(e)^3 + 60*B*b^
2*c^2*d*tan(e)^3 + 60*C*a^2*c*d^2*tan(e)^3 + 120*B*a*b*c*d^2*tan(e)^3 + 60*
A*b^2*c*d^2*tan(e)^3 - 60*C*b^2*c*d^2*tan(e)^3 + 20*B*a^2*d^3*tan(e)^3 + 40
*A*a*b*d^3*tan(e)^3 - 40*C*a*b*d^3*tan(e)^3 - 20*B*b^2*d^3*tan(e)^3 + 60*A*
a^2*c^3*f*x - 60*C*a^2*c^3*f*x - 120*B*a*b*c^3*f*x - 60*A*b^2*c^3*f*x + 60*
C*b^2*c^3*f*x - 180*B*a^2*c^2*d*f*x - 360*A*a*b*c^2*d*f*x + 360*C*a*b*c^2*d
*f*x + 180*B*b^2*c^2*d*f*x - 180*A*a^2*c*d^2*f*x + 180*C*a^2*c*d^2*f*x + 36
0*B*a*b*c*d^2*f*x + 180*A*b^2*c*d^2*f*x - 180*C*b^2*c*d^2*f*x + 60*B*a^2*d^
3*f*x + 120*A*a*b*d^3*f*x - 120*C*a*b*d^3*f*x - 60*B*b^2*d^3*f*x + 60*C*a*b
*c^3*tan(f*x)^2 + 30*B*b^2*c^3*tan(f*x)^2 + 90*C*a^2*c^2*d*tan(f*x)^2 + 180
*B*a*b*c^2*d*tan(f*x)^2 + 90*A*b^2*c^2*d*tan(f*x)^2 - 90*C*b^2*c^2*d*tan(f*
x)^2 + 90*B*a^2*c*d^2*tan(f*x)^2 + 180*A*a*b*c*d^2*tan(f*x)^2 - 180*C*a*b*c
*d^2*tan(f*x)^2 - 90*B*b^2*c*d^2*tan(f*x)^2 + 30*A*a^2*d^3*tan(f*x)^2 - 30*
C*a^2*d^3*tan(f*x)^2 - 60*B*a*b*d^3*tan(f*x)^2 - 30*A*b^2*d^3*tan(f*x)^2 +
30*C*b^2*d^3*tan(f*x)^2 - 240*C*a*b*c^3*tan(f*x)*tan(e) - 120*B*b^2*c^3*tan
(f*x)*tan(e) - 360*C*a^2*c^2*d*tan(f*x)*tan(e) - 720*B*a*b*c^2*d*tan(f*x)*t
an(e) - 360*A*b^2*c^2*d*tan(f*x)*tan(e) + 630*C*b^2*c^2*d*tan(f*x)*tan(e) -
360*B*a^2*c*d^2*tan(f*x)*tan(e) - 720*A*a*b*c*d^2*tan(f*x)*tan(e) + 1260*C
*a*b*c*d^2*tan(f*x)*tan(e) + 630*B*b^2*c*d^2*tan(f*x)*tan(e) - 120*A*a^2*d^
3*tan(f*x)*tan(e) + 210*C*a^2*d^3*tan(f*x)*tan(e) + 420*B*a*b*d^3*tan(f*x)*
tan(e) + 210*A*b^2*d^3*tan(f*x)*tan(e) - 270*C*b^2*d^3*tan(f*x)*tan(e) + 60

*C*a*b*c^3*tan(e)^2 + 30*B*b^2*c^3*tan(e)^2 + 90*C*a^2*c^2*d*tan(e)^2 + 180
*B*a*b*c^2*d*tan(e)^2 + 90*A*b^2*c^2*d*tan(e)^2 - 90*C*b^2*c^2*d*tan(e)^2 +
90*B*a^2*c*d^2*tan(e)^2 + 180*A*a*b*c*d^2*tan(e)^2 - 180*C*a*b*c*d^2*tan(e
)^2 - 90*B*b^2*c*d^2*tan(e)^2 + 30*A*a^2*d^3*tan(e)^2 - 30*C*a^2*d^3*tan(e)
^2 - 60*B*a*b*d^3*tan(e)^2 - 30*A*b^2*d^3*tan(e)^2 + 30*C*b^2*d^3*tan(e)^2
- 30*B*a^2*c^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x
)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 60*A*a*b*c^3*log(4*(tan(f*x)^2
*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(
e)^2 + 1)) + 60*C*a*b*c^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) +
1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) + 30*B*b^2*c^3*log(4*
(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*
x)^2 + tan(e)^2 + 1)) - 90*A*a^2*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f
*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) + 90*C*a
^2*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*ta
n(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) + 180*B*a*b*c^2*d*log(4*(tan(f*x)^2*ta
n(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^
2 + 1)) + 90*A*b^2*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1
)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 90*C*b^2*c^2*d*log(4
*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f
*x)^2 + tan(e)^2 + 1)) + 90*B*a^2*c*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(
f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) + 180*A
*a*b*c*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*
tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 180*C*a*b*c*d^2*log(4*(tan(f*x)^2*
tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e
)^2 + 1)) - 90*B*b^2*c*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) +
1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) + 30*A*a^2*d^3*log(4
*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f
*x)^2 + tan(e)^2 + 1)) - 30*C*a^2*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*
x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 60*B*a*
b*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e
)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 30*A*b^2*d^3*log(4*(tan(f*x)^2*tan(e)^2
- 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)
) + 30*C*b^2*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f
*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) + 60*C*a^2*c^3*tan(f*x) + 120*
B*a*b*c^3*tan(f*x) + 60*A*b^2*c^3*tan(f*x) - 60*C*b^2*c^3*tan(f*x) + 180*B*
a^2*c^2*d*tan(f*x) + 360*A*a*b*c^2*d*tan(f*x) - 360*C*a*b*c^2*d*tan(f*x) -
180*B*b^2*c^2*d*tan(f*x) + 180*A*a^2*c*d^2*tan(f*x) - 180*C*a^2*c*d^2*tan(f
*x) - 360*B*a*b*c*d^2*tan(f*x) - 180*A*b^2*c*d^2*tan(f*x) + 180*C*b^2*c*d^2
*tan(f*x) - 60*B*a^2*d^3*tan(f*x) - 120*A*a*b*d^3*tan(f*x) + 120*C*a*b*d^3*
tan(f*x) + 60*B*b^2*d^3*tan(f*x) + 60*C*a^2*c^3*tan(e) + 120*B*a*b*c^3*tan(
e) + 60*A*b^2*c^3*tan(e) - 60*C*b^2*c^3*tan(e) + 180*B*a^2*c^2*d*tan(e) + 3
60*A*a*b*c^2*d*tan(e) - 360*C*a*b*c^2*d*tan(e) - 180*B*b^2*c^2*d*tan(e) + 1
80*A*a^2*c*d^2*tan(e) - 180*C*a^2*c*d^2*tan(e) - 360*B*a*b*c*d^2*tan(e) - 1
80*A*b^2*c*d^2*tan(e) + 180*C*b^2*c*d^2*tan(e) - 60*B*a^2*d^3*tan(e) - 120*
A*a*b*d^3*tan(e) + 120*C*a*b*d^3*tan(e) + 60*B*b^2*d^3*tan(e) + 60*C*a*b*c^

```
3 + 30*B*b^2*c^3 + 90*C*a^2*c^2*d + 180*B*a*b*c^2*d + 90*A*b^2*c^2*d - 135*
C*b^2*c^2*d + 90*B*a^2*c*d^2 + 180*A*a*b*c*d^2 - 270*C*a*b*c*d^2 - 135*B*b^
2*c*d^2 + 30*A*a^2*d^3 - 45*C*a^2*d^3 - 90*B*a*b*d^3 - 45*A*b^2*d^3 + 55*C*
b^2*d^3)/(f*tan(f*x)^6*tan(e)^6 - 6*f*tan(f*x)^5*tan(e)^5 + 15*f*tan(f*x)^4
*tan(e)^4 - 20*f*tan(f*x)^3*tan(e)^3 + 15*f*tan(f*x)^2*tan(e)^2 - 6*f*tan(f
*x)*tan(e) + f)
```

Mupad [B] (verification not implemented)

Time = 8.63 (sec) , antiderivative size = 891, normalized size of antiderivative = 1.48

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= x\left(A\,a^2\,c^3-A\,b^2\,c^3+B\,a^2\,d^3-C\,a^2\,c^3-B\,b^2\,d^3+C\,b^2\,c^3+2\,A\,a\,b\,d^3-2\,B\,a\,b\,c^3 -2\,C\,a\,b\,d^3-3\,A\,a^2\,c\,d^2+3\,A\,b^2\,c\,d^2-3\,B\,a^2\,c^2\,d+3\,B\,b^2\,c^2\,d+3\,C\,a^2\,c\,d^2-3\,C\,b^2\,c\,d^2 -6\,A\,a\,b\,c^2\,d+6\,B\,a\,b\,c\,d^2+6\,C\,a\,b\,c^2\,d\right)$$
$$-\frac{\tan(e+f\,x)\,\left(B\,a^2\,d^3-A\,b^2\,c^3-b\,d^2\,(B\,b\,d+2\,C\,a\,d+3\,C\,b\,c)-C\,a^2\,c^3+C\,b^2\,c^3+2\,A\,a\,b\,d^3-2\,B\,a\,b}{}$$
$$-\frac{\ln\left(\tan(e+f\,x)^2+1\right)\,\left(\frac{A\,a^2\,d^3}{2}-\frac{B\,a^2\,c^3}{2}-\frac{A\,b^2\,d^3}{2}+\frac{B\,b^2\,c^3}{2}-\frac{C\,a^2\,d^3}{2}+\frac{C\,b^2\,d^3}{2}-A\,a\,b\,c^3-B\,a\,b\,d^3+C\,a\,b\,c^3}{}$$
$$+\frac{\tan(e+f\,x)^4\,\left(\frac{A\,b^2\,d^3}{4}+\frac{C\,a^2\,d^3}{4}-\frac{C\,b^2\,d^3}{4}+\frac{B\,a\,b\,d^3}{2}+\frac{3\,B\,b^2\,c\,d^2}{4}+\frac{3\,C\,b^2\,c^2\,d}{4}+\frac{3\,C\,a\,b\,c\,d^2}{2}\right)}{f}$$
$$+\frac{\tan(e+f\,x)^3\,\left(\frac{B\,a^2\,d^3}{3}-\frac{b\,d^2\,(B\,b\,d+2\,C\,a\,d+3\,C\,b\,c)}{3}+\frac{C\,b^2\,c^3}{3}+\frac{2\,A\,a\,b\,d^3}{3}+A\,b^2\,c\,d^2+B\,b^2\,c^2\,d+C\,a^2\,c\,d^2+2\,B\,a}{f}$$
$$+\frac{\tan(e+f\,x)^2\,\left(\frac{A\,a^2\,d^3}{2}-\frac{A\,b^2\,d^3}{2}+\frac{B\,b^2\,c^3}{2}-\frac{C\,a^2\,d^3}{2}+\frac{C\,b^2\,d^3}{2}-B\,a\,b\,d^3+C\,a\,b\,c^3+\frac{3\,A\,b^2\,c^2\,d}{2}+\frac{3\,B\,a^2\,c\,d^2}{2}-\frac{3\,B\,b}{}}{f}$$
$$+\frac{b\,d^2\tan(e+f\,x)^5\,(B\,b\,d+2\,C\,a\,d+3\,C\,b\,c)}{5\,f}+\frac{C\,b^2\,d^3\tan(e+f\,x)^6}{6\,f}$$

```
[In] int((a + b*tan(e + f*x))^2*(c + d*tan(e + f*x))^3*(A + B*tan(e + f*x) + C*t
an(e + f*x)^2),x)
```

```
[Out] x*(A*a^2*c^3 - A*b^2*c^3 + B*a^2*d^3 - C*a^2*c^3 - B*b^2*d^3 + C*b^2*c^3 +
2*A*a*b*d^3 - 2*B*a*b*c^3 - 2*C*a*b*d^3 - 3*A*a^2*c*d^2 + 3*A*b^2*c*d^2 - 3
*B*a^2*c^2*d + 3*B*b^2*c^2*d + 3*C*a^2*c*d^2 - 3*C*b^2*c*d^2 - 6*A*a*b*c^2*
d + 6*B*a*b*c*d^2 + 6*C*a*b*c^2*d) - (tan(e + f*x)*(B*a^2*d^3 - A*b^2*c^3 -
 b*d^2*(B*b*d + 2*C*a*d + 3*C*b*c) - C*a^2*c^3 + C*b^2*c^3 + 2*A*a*b*d^3 -
2*B*a*b*c^3 - 3*A*a^2*c*d^2 + 3*A*b^2*c*d^2 - 3*B*a^2*c^2*d + 3*B*b^2*c^2*d
 + 3*C*a^2*c*d^2 - 6*A*a*b*c^2*d + 6*B*a*b*c*d^2 + 6*C*a*b*c^2*d))/f - (log
(tan(e + f*x)^2 + 1)*((A*a^2*d^3)/2 - (B*a^2*c^3)/2 - (A*b^2*d^3)/2 + (B*b^
2*c^3)/2 - (C*a^2*d^3)/2 + (C*b^2*d^3)/2 - A*a*b*c^3 - B*a*b*d^3 + C*a*b*c^
```

3 - (3*A*a^2*c^2*d)/2 + (3*A*b^2*c^2*d)/2 + (3*B*a^2*c*d^2)/2 - (3*B*b^2*c*d^2)/2 + (3*C*a^2*c^2*d)/2 - (3*C*b^2*c^2*d)/2 + 3*A*a*b*c*d^2 + 3*B*a*b*c^2*d - 3*C*a*b*c*d^2))/f + (tan(e + f*x)^4*((A*b^2*d^3)/4 + (C*a^2*d^3)/4 - (C*b^2*d^3)/4 + (B*a*b*d^3)/2 + (3*B*b^2*c*d^2)/4 + (3*C*b^2*c^2*d)/4 + (3*C*a*b*c*d^2)/2))/f + (tan(e + f*x)^3*((B*a^2*d^3)/3 - (b*d^2*(B*b*d + 2*C*a*d + 3*C*b*c))/3 + (C*b^2*c^3)/3 + (2*A*a*b*d^3)/3 + A*b^2*c*d^2 + B*b^2*c^2*d + C*a^2*c*d^2 + 2*B*a*b*c*d^2 + 2*C*a*b*c^2*d))/f + (tan(e + f*x)^2*((A*a^2*d^3)/2 - (A*b^2*d^3)/2 + (B*b^2*c^3)/2 - (C*a^2*d^3)/2 + (C*b^2*d^3)/2 - B*a*b*d^3 + C*a*b*c^3 + (3*A*b^2*c^2*d)/2 + (3*B*a^2*c*d^2)/2 - (3*B*b^2*c*d^2)/2 + (3*C*a^2*c^2*d)/2 - (3*C*b^2*c^2*d)/2 + 3*A*a*b*c*d^2 + 3*B*a*b*c^2*d - 3*C*a*b*c*d^2))/f + (b*d^2*tan(e + f*x)^5*(B*b*d + 2*C*a*d + 3*C*b*c))/(5*f) + (C*b^2*d^3*tan(e + f*x)^6)/(6*f)

3.65 $\int (a+b\tan(e+fx))(c+d\tan(e+fx))^3 \left(A+B\tan(e+f\right.$

Optimal result 600
Rubi [A] (verified) 601
Mathematica [C] (verified) 604
Maple [A] (verified) 604
Fricas [A] (verification not implemented) 605
Sympy [B] (verification not implemented) 605
Maxima [A] (verification not implemented) 606
Giac [B] (verification not implemented) 607
Mupad [B] (verification not implemented) 614

Optimal result

Integrand size = 43, antiderivative size = 389

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))^3 \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= \left(a\left(Ac^3-c^3C-3Bc^2d-3Acd^2+3cCd^2+Bd^3\right)-b\left((A-C)d\left(3c^2-d^2\right)+B\left(c^3-3cd^2\right)\right)\right)x$$
$$-\frac{(A(bc^3+3ac^2d-3bcd^2-ad^3)-b(c^3C+3Bc^2d-3cCd^2-Bd^3)+a(Bc^3-3c^2Cd-3Bcd^2+Cd^3))\,\mathrm{lo}}{f}$$
$$+\frac{d(a(Bc^2-2cCd-Bd^2)-b(c^2C+2Bcd-Cd^2)+A(2acd+b(c^2-d^2)))\tan(e+fx)}{f}$$
$$+\frac{(Abc+aBc-bcC+aAd-bBd-aCd)(c+d\tan(e+fx))^2}{2f}$$
$$+\frac{(Ab+aB-bC)(c+d\tan(e+fx))^3}{3f}$$
$$-\frac{(bcC-5bBd-5aCd)(c+d\tan(e+fx))^4}{20d^2f}+\frac{bC\tan(e+fx)(c+d\tan(e+fx))^4}{5df}$$

```
[Out] (a*(A*c^3-3*A*c*d^2-3*B*c^2*d+B*d^3-C*c^3+3*C*c*d^2)-b*((A-C)*d*(3*c^2-d^2)
+B*(c^3-3*c*d^2)))*x-(A*(3*a*c^2*d-a*d^3+b*c^3-3*b*c*d^2)-b*(3*B*c^2*d-B*d^
3+C*c^3-3*C*c*d^2)+a*(B*c^3-3*B*c*d^2-3*C*c^2*d+C*d^3))*ln(cos(f*x+e))/f+d*
(a*(B*c^2-B*d^2-2*C*c*d)-b*(2*B*c*d+C*c^2-C*d^2)+A*(2*a*c*d+b*(c^2-d^2)))*t
an(f*x+e)/f+1/2*(A*a*d+A*b*c+B*a*c-B*b*d-C*a*d-C*b*c)*(c+d*tan(f*x+e))^2/f+
1/3*(A*b+B*a-C*b)*(c+d*tan(f*x+e))^3/f-1/20*(-5*B*b*d-5*C*a*d+C*b*c)*(c+d*t
an(f*x+e))^4/d^2/f+1/5*b*C*tan(f*x+e)*(c+d*tan(f*x+e))^4/d/f
```

Rubi [A] (verified)

Time = 0.79 (sec) , antiderivative size = 387, normalized size of antiderivative = 0.99, number of steps used = 6, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.116, Rules used = {3718, 3711, 3609, 3606, 3556}

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\frac{d\tan(e+fx)\left(2aAcd+aB(c^2-d^2)-2acCd+Ab(c^2-d^2)-b(2Bcd+c^2C-Cd^2)\right)}{f}$$
$$-\frac{\log(\cos(e+fx))\left(A(3ac^2d-ad^3+bc^3-3bcd^2)+a(Bc^3-3Bcd^2-3c^2Cd+Cd^3)-b(3Bc^2d-Bd^3+\right.}{f}$$
$$-x\left(-a\left(Ac^3-3Acd^2-3Bc^2d+Bd^3-c^3C+3cCd^2\right)+bd(A-C)\left(3c^2-d^2\right)+bB\left(c^3-3cd^2\right)\right)$$
$$+\frac{(aB+Ab-bC)(c+d\tan(e+fx))^3}{3f}$$
$$+\frac{(c+d\tan(e+fx))^2(aAd+aBc-aCd+Abc-bBd-bcC)}{2f}$$
$$-\frac{(-5aCd-5bBd+bcC)(c+d\tan(e+fx))^4}{20d^2f}+\frac{bC\tan(e+fx)(c+d\tan(e+fx))^4}{5df}$$

```
[In] Int[(a + b*Tan[e + f*x])*(c + d*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan
[e + f*x]^2),x]
```

```
[Out] -((b*(A - C)*d*(3*c^2 - d^2) + b*B*(c^3 - 3*c*d^2) - a*(A*c^3 - c^3*C - 3*B
*c^2*d - 3*A*c*d^2 + 3*c*C*d^2 + B*d^3))*x) - ((A*(b*c^3 + 3*a*c^2*d - 3*b*
c*d^2 - a*d^3) - b*(c^3*C + 3*B*c^2*d - 3*c*C*d^2 - B*d^3) + a*(B*c^3 - 3*c
^2*C*d - 3*B*c*d^2 + C*d^3))*Log[Cos[e + f*x]])/f + (d*(2*a*A*c*d - 2*a*c*C
*d + A*b*(c^2 - d^2) + a*B*(c^2 - d^2) - b*(c^2*C + 2*B*c*d - C*d^2))*Tan[e
 + f*x])/f + ((A*b*c + a*B*c - b*c*C + a*A*d - b*B*d - a*C*d)*(c + d*Tan[e
+ f*x])^2)/(2*f) + ((A*b + a*B - b*C)*(c + d*Tan[e + f*x])^3)/(3*f) - ((b*c
*C - 5*b*B*d - 5*a*C*d)*(c + d*Tan[e + f*x])^4)/(20*d^2*f) + (b*C*Tan[e + f
*x]*(c + d*Tan[e + f*x])^4)/(5*d*f)
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3606

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
 f*x], x], x] + Simp[b*d*(Tan[e + f*x]/f), x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3609

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[d*((a + b*Tan[e + f*x])^m/(f*m)), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rubi steps

$$\text{integral} = \frac{bC\tan(e+fx)(c+d\tan(e+fx))^4}{5df} - \frac{\int (c+d\tan(e+fx))^3\left(bcC-5aAd-5(Ab+aB-bC)d\tan(e+fx)+(bcC-5bBd-5aCd)\tan^2(e+\right.}{5d}$$

$$= -\frac{(bcC-5bBd-5aCd)(c+d\tan(e+fx))^4}{20d^2f} + \frac{bC\tan(e+fx)(c+d\tan(e+fx))^4}{5df} - \frac{\int (c+d\tan(e+fx))^3(5(bB-a(A-C))d-5(Ab+aB-bC)d\tan(e+fx))\,dx}{5d}$$

$$= \frac{(Ab+aB-bC)(c+d\tan(e+fx))^3}{3f} - \frac{(bcC-5bBd-5aCd)(c+d\tan(e+fx))^4}{20d^2f} + \frac{bC\tan(e+fx)(c+d\tan(e+fx))^4}{5df} - \frac{\int (c+d\tan(e+fx))^2(5d(bBc+b(A-C)d-a(Ac-cC-Bd))-5d(Abc+aBc-bcC+aAd}{5d}$$

$$= \frac{(Abc + aBc - bcC + aAd - bBd - aCd)(c + d\tan(e + fx))^2}{2f}$$
$$+ \frac{(Ab + aB - bC)(c + d\tan(e + fx))^3}{3f}$$
$$- \frac{(bcC - 5bBd - 5aCd)(c + d\tan(e + fx))^4}{20d^2 f} + \frac{bC\tan(e + fx)(c + d\tan(e + fx))^4}{5df}$$
$$- \frac{\int (c + d\tan(e + fx))\left(5d(a(c^2C + 2Bcd - Cd^2 - A(c^2 - d^2)) + b(2c(A - C)d + B(c^2 - d^2))) \cdots\right.}{5d \cdots}$$

$$= -\left(\left(b(A - C)d\left(3c^2 - d^2\right) + bB\left(c^3 - 3cd^2\right) - a\left(Ac^3 - c^3C - 3Bc^2d - 3Acd^2 + 3cCd^2 + Bd^3\right)\right)x\right)$$
$$+ \frac{d(2aAcd - 2acCd + Ab(c^2 - d^2) + aB(c^2 - d^2) - b(c^2C + 2Bcd - Cd^2))\tan(e + fx)}{f}$$
$$+ \frac{(Abc + aBc - bcC + aAd - bBd - aCd)(c + d\tan(e + fx))^2}{2f}$$
$$+ \frac{(Ab + aB - bC)(c + d\tan(e + fx))^3}{3f}$$
$$- \frac{(bcC - 5bBd - 5aCd)(c + d\tan(e + fx))^4}{20d^2 f}$$
$$+ \frac{bC\tan(e + fx)(c + d\tan(e + fx))^4}{5df}$$
$$- \left(-A\left(bc^3 + 3ac^2d - 3bcd^2 - ad^3\right) + b\left(c^3C + 3Bc^2d - 3cCd^2 - Bd^3\right) - a\left(Bc^3 - 3c^2Cd - 3Bcd^2 + Cd^3\right)\right)\int \tan(e + fx)\,dx$$

$$= -\left(\left(b(A - C)d\left(3c^2 - d^2\right) + bB\left(c^3 - 3cd^2\right) - a\left(Ac^3 - c^3C - 3Bc^2d - 3Acd^2 + 3cCd^2 + Bd^3\right)\right)x\right)$$
$$- \frac{(A(bc^3 + 3ac^2d - 3bcd^2 - ad^3) - b(c^3C + 3Bc^2d - 3cCd^2 - Bd^3) + a(Bc^3 - 3c^2Cd - 3Bcd^2 + \cdots}{f}$$
$$+ \frac{d(2aAcd - 2acCd + Ab(c^2 - d^2) + aB(c^2 - d^2) - b(c^2C + 2Bcd - Cd^2))\tan(e + fx)}{f}$$
$$+ \frac{(Abc + aBc - bcC + aAd - bBd - aCd)(c + d\tan(e + fx))^2}{2f}$$
$$+ \frac{(Ab + aB - bC)(c + d\tan(e + fx))^3}{3f}$$
$$- \frac{(bcC - 5bBd - 5aCd)(c + d\tan(e + fx))^4}{20d^2 f} + \frac{bC\tan(e + fx)(c + d\tan(e + fx))^4}{5df}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 6.37 (sec) , antiderivative size = 297, normalized size of antiderivative = 0.76

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\frac{bC\tan(e+fx)(c+d\tan(e+fx))^4}{5df}$$
$$-\frac{\frac{(bcC-5bBd-5aCd)(c+d\tan(e+fx))^4}{4df}+\frac{5(3(Abc+aBc-bcC-aAd+bBd+aCd)((ic-d)^3\log(i-\tan(e+fx))-(ic+d)^3\log(i+\tan(e+fx))+6cd}{}}{}$$

```
[In] Integrate[(a + b*Tan[e + f*x])*(c + d*Tan[e + f*x])^3*(A + B*Tan[e + f*x] +
 C*Tan[e + f*x]^2),x]
```

```
[Out] (b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^4)/(5*d*f) - (((b*c*C - 5*b*B*d - 5*
a*C*d)*(c + d*Tan[e + f*x])^4)/(4*d*f) + (5*(3*(A*b*c + a*B*c - b*c*C - a*A
*d + b*B*d + a*C*d)*((I*c - d)^3*Log[I - Tan[e + f*x]] - (I*c + d)^3*Log[I
+ Tan[e + f*x]] + 6*c*d^2*Tan[e + f*x] + d^3*Tan[e + f*x]^2) + (A*b + a*B -
 b*C)*((3*I)*(c + I*d)^4*Log[I - Tan[e + f*x]] - (3*I)*(c - I*d)^4*Log[I +
Tan[e + f*x]] - 6*d^2*(6*c^2 - d^2)*Tan[e + f*x] - 12*c*d^3*Tan[e + f*x]^2
- 2*d^4*Tan[e + f*x]^3)))/(6*f))/(5*d)
```

Maple [A] (verified)

Time = 0.24 (sec) , antiderivative size = 347, normalized size of antiderivative = 0.89

method	result
parts	$\frac{(3Aac^2d+Abc^3+Bac^3)\ln\left(1+\tan(fx+e)^2\right)}{2f}+\frac{(Bbd^3+Cad^3+3Cbcd^2)\left(\frac{\tan(fx+e)^4}{4}-\frac{\tan(fx+e)^2}{2}+\frac{\ln\left(1+\tan(fx+e)^2\right)}{2}\right.}{f}$
norman	$(Aac^3-3Aacd^2-3Abc^2d+Abd^3-3Bac^2d+Bad^3-Bbc^3+3Bbcd^2-Cac^3+3C$
derivativedivides	$\frac{Cbc^3\tan(fx+e)^2}{2}-\frac{Cbd^3\tan(fx+e)^3}{3}+\frac{Aad^3\tan(fx+e)^2}{2}-\frac{Bbd^3\tan(fx+e)^2}{2}-\frac{Cad^3\tan(fx+e)^2}{2}+\frac{Cbd^3\tan(fx+e)^5}{5}+\frac{Bbd^3\tan(}{4}$
default	$\frac{Cbc^3\tan(fx+e)^2}{2}-\frac{Cbd^3\tan(fx+e)^3}{3}+\frac{Aad^3\tan(fx+e)^2}{2}-\frac{Bbd^3\tan(fx+e)^2}{2}-\frac{Cad^3\tan(fx+e)^2}{2}+\frac{Cbd^3\tan(fx+e)^5}{5}+\frac{Bbd^3\tan(}{4}$
parallelrisch	$30Cbc^3\tan(fx+e)^2-20Cbd^3\tan(fx+e)^3+30Aad^3\tan(fx+e)^2-30Bbd^3\tan(fx+e)^2-30Cad^3\tan(fx+e)^2+12Cbd^3\mathrm{ta}$
risch	Expression too large to display

```
[In] int((a+b*tan(f*x+e))*(c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x,m
ethod=_RETURNVERBOSE)
```

```
[Out] 1/2*(3*A*a*c^2*d+A*b*c^3+B*a*c^3)/f*ln(1+tan(f*x+e)^2)+(B*b*d^3+C*a*d^3+3*C
*b*c*d^2)/f*(1/4*tan(f*x+e)^4-1/2*tan(f*x+e)^2+1/2*ln(1+tan(f*x+e)^2))+(A*b
*d^3+B*a*d^3+3*B*b*c*d^2+3*C*a*c*d^2+3*C*b*c^2*d)/f*(1/3*tan(f*x+e)^3-tan(f
*x+e)+arctan(tan(f*x+e)))+(3*A*a*c*d^2+3*A*b*c^2*d+3*B*a*c^2*d+B*b*c^3+C*a*
c^3)/f*(tan(f*x+e)-arctan(tan(f*x+e)))+(A*a*d^3+3*A*b*c*d^2+3*B*a*c*d^2+3*B
*b*c^2*d+3*C*a*c^2*d+C*b*c^3)/f*(1/2*tan(f*x+e)^2-1/2*ln(1+tan(f*x+e)^2))+A
*a*c^3*x+C*b*d^3/f*(1/5*tan(f*x+e)^5-1/3*tan(f*x+e)^3+tan(f*x+e)-arctan(tan
(f*x+e)))
```

Fricas [A] (verification not implemented)

none

Time = 0.28 (sec) , antiderivative size = 386, normalized size of antiderivative = 0.99

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\frac{12\,Cbd^3\tan\left(fx+e\right)^5+15\left(3\,Cbcd^2+(Ca+Bb)d^3\right)\tan\left(fx+e\right)^4+20\left(3\,Cbc^2d+3\left(Ca+Bb\right)cd^2+(B\cdots\right.}{}$$

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x, algorithm="fricas")
```

```
[Out] 1/60*(12*C*b*d^3*tan(f*x + e)^5 + 15*(3*C*b*c*d^2 + (C*a + B*b)*d^3)*tan(f*
x + e)^4 + 20*(3*C*b*c^2*d + 3*(C*a + B*b)*c*d^2 + (B*a + (A - C)*b)*d^3)*t
an(f*x + e)^3 + 60*(((A - C)*a - B*b)*c^3 - 3*(B*a + (A - C)*b)*c^2*d - 3*(
(A - C)*a - B*b)*c*d^2 + (B*a + (A - C)*b)*d^3)*f*x + 30*(C*b*c^3 + 3*(C*a
+ B*b)*c^2*d + 3*(B*a + (A - C)*b)*c*d^2 + ((A - C)*a - B*b)*d^3)*tan(f*x +
 e)^2 - 30*((B*a + (A - C)*b)*c^3 + 3*((A - C)*a - B*b)*c^2*d - 3*(B*a + (A
 - C)*b)*c*d^2 - ((A - C)*a - B*b)*d^3)*log(1/(tan(f*x + e)^2 + 1)) + 60*((
C*a + B*b)*c^3 + 3*(B*a + (A - C)*b)*c^2*d + 3*((A - C)*a - B*b)*c*d^2 - (B
*a + (A - C)*b)*d^3)*tan(f*x + e))/f
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1001 vs. 2(379) = 758.

Time = 0.28 (sec) , antiderivative size = 1001, normalized size of antiderivative = 2.57

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))**3*(A+B*tan(f*x+e)+C*tan(f*x+e)
**2),x)
```

```
[Out] Piecewise((A*a*c**3*x + 3*A*a*c**2*d*log(tan(e + f*x)**2 + 1)/(2*f) - 3*A*a
*c*d**2*x + 3*A*a*c*d**2*tan(e + f*x)/f - A*a*d**3*log(tan(e + f*x)**2 + 1)
/(2*f) + A*a*d**3*tan(e + f*x)**2/(2*f) + A*b*c**3*log(tan(e + f*x)**2 + 1)
/(2*f) - 3*A*b*c**2*d*x + 3*A*b*c**2*d*tan(e + f*x)/f - 3*A*b*c*d**2*log(ta
n(e + f*x)**2 + 1)/(2*f) + 3*A*b*c*d**2*tan(e + f*x)**2/(2*f) + A*b*d**3*x
+ A*b*d**3*tan(e + f*x)**3/(3*f) - A*b*d**3*tan(e + f*x)/f + B*a*c**3*log(t
an(e + f*x)**2 + 1)/(2*f) - 3*B*a*c**2*d*x + 3*B*a*c**2*d*tan(e + f*x)/f -
3*B*a*c*d**2*log(tan(e + f*x)**2 + 1)/(2*f) + 3*B*a*c*d**2*tan(e + f*x)**2/
(2*f) + B*a*d**3*x + B*a*d**3*tan(e + f*x)**3/(3*f) - B*a*d**3*tan(e + f*x)
/f - B*b*c**3*x + B*b*c**3*tan(e + f*x)/f - 3*B*b*c**2*d*log(tan(e + f*x)**
2 + 1)/(2*f) + 3*B*b*c**2*d*tan(e + f*x)**2/(2*f) + 3*B*b*c*d**2*x + B*b*c*
d**2*tan(e + f*x)**3/f - 3*B*b*c*d**2*tan(e + f*x)/f + B*b*d**3*log(tan(e +
 f*x)**2 + 1)/(2*f) + B*b*d**3*tan(e + f*x)**4/(4*f) - B*b*d**3*tan(e + f*x
)**2/(2*f) - C*a*c**3*x + C*a*c**3*tan(e + f*x)/f - 3*C*a*c**2*d*log(tan(e
+ f*x)**2 + 1)/(2*f) + 3*C*a*c**2*d*tan(e + f*x)**2/(2*f) + 3*C*a*c*d**2*x
+ C*a*c*d**2*tan(e + f*x)**3/f - 3*C*a*c*d**2*tan(e + f*x)/f + C*a*d**3*log
(tan(e + f*x)**2 + 1)/(2*f) + C*a*d**3*tan(e + f*x)**4/(4*f) - C*a*d**3*tan
(e + f*x)**2/(2*f) - C*b*c**3*log(tan(e + f*x)**2 + 1)/(2*f) + C*b*c**3*tan
(e + f*x)**2/(2*f) + 3*C*b*c**2*d*x + C*b*c**2*d*tan(e + f*x)**3/f - 3*C*b*
c**2*d*tan(e + f*x)/f + 3*C*b*c*d**2*log(tan(e + f*x)**2 + 1)/(2*f) + 3*C*b
*c*d**2*tan(e + f*x)**4/(4*f) - 3*C*b*c*d**2*tan(e + f*x)**2/(2*f) - C*b*d*
*3*x + C*b*d**3*tan(e + f*x)**5/(5*f) - C*b*d**3*tan(e + f*x)**3/(3*f) + C*
b*d**3*tan(e + f*x)/f, Ne(f, 0)), (x*(a + b*tan(e))*(c + d*tan(e))**3*(A +
B*tan(e) + C*tan(e)**2), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.32 (sec) , antiderivative size = 387, normalized size of antiderivative = 0.99

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= \frac{12\,Cbd^3\tan\left(fx+e\right)^5+15\left(3\,Cbcd^2+(Ca+Bb)d^3\right)\tan\left(fx+e\right)^4+20\left(3\,Cbc^2d+3\,(Ca+Bb)cd^2+(Ba\right.}{}$$

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x, algorithm="maxima")
```

```
[Out] 1/60*(12*C*b*d^3*tan(f*x + e)^5 + 15*(3*C*b*c*d^2 + (C*a + B*b)*d^3)*tan(f*
x + e)^4 + 20*(3*C*b*c^2*d + 3*(C*a + B*b)*c*d^2 + (B*a + (A - C)*b)*d^3)*t
an(f*x + e)^3 + 30*(C*b*c^3 + 3*(C*a + B*b)*c^2*d + 3*(B*a + (A - C)*b)*c*d
^2 + ((A - C)*a - B*b)*d^3)*tan(f*x + e)^2 + 60*(((A - C)*a - B*b)*c^3 - 3*
(B*a + (A - C)*b)*c^2*d - 3*((A - C)*a - B*b)*c*d^2 + (B*a + (A - C)*b)*d^3
)*(f*x + e) + 30*((B*a + (A - C)*b)*c^3 + 3*((A - C)*a - B*b)*c^2*d - 3*(B*
```

```
a + (A - C)*b)*c*d^2 - ((A - C)*a - B*b)*d^3)*log(tan(f*x + e)^2 + 1) + 60*
((C*a + B*b)*c^3 + 3*(B*a + (A - C)*b)*c^2*d + 3*((A - C)*a - B*b)*c*d^2 -
(B*a + (A - C)*b)*d^3)*tan(f*x + e))/f
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 10353 vs. $2(381) = 762$.

Time = 9.71 (sec) , antiderivative size = 10353, normalized size of antiderivative = 26.61

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x, algorithm="giac")
```

```
[Out] 1/60*(60*A*a*c^3*f*x*tan(f*x)^5*tan(e)^5 - 60*C*a*c^3*f*x*tan(f*x)^5*tan(e)
^5 - 60*B*b*c^3*f*x*tan(f*x)^5*tan(e)^5 - 180*B*a*c^2*d*f*x*tan(f*x)^5*tan(
e)^5 - 180*A*b*c^2*d*f*x*tan(f*x)^5*tan(e)^5 + 180*C*b*c^2*d*f*x*tan(f*x)^5
*tan(e)^5 - 180*A*a*c*d^2*f*x*tan(f*x)^5*tan(e)^5 + 180*C*a*c*d^2*f*x*tan(f
*x)^5*tan(e)^5 + 180*B*b*c*d^2*f*x*tan(f*x)^5*tan(e)^5 + 60*B*a*d^3*f*x*tan
(f*x)^5*tan(e)^5 + 60*A*b*d^3*f*x*tan(f*x)^5*tan(e)^5 - 60*C*b*d^3*f*x*tan(
f*x)^5*tan(e)^5 - 30*B*a*c^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e)
 + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^
5 - 30*A*b*c^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x
)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5 + 30*C*b*c^3
*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 +
 tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5 - 90*A*a*c^2*d*log(4*(tan(
f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2
+ tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5 + 90*C*a*c^2*d*log(4*(tan(f*x)^2*tan(e
)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 +
 1))*tan(f*x)^5*tan(e)^5 + 90*B*b*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(
f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x
)^5*tan(e)^5 + 90*B*a*c*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e)
+ 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5
 + 90*A*b*c*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*
x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5 - 90*C*b*c*
d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^
2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5 + 30*A*a*d^3*log(4*(tan
(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2
 + tan(e)^2 + 1))*tan(f*x)^5*tan(e)^5 - 30*C*a*d^3*log(4*(tan(f*x)^2*tan(e)
^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 +
1))*tan(f*x)^5*tan(e)^5 - 30*B*b*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x
)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^5
*tan(e)^5 - 300*A*a*c^3*f*x*tan(f*x)^4*tan(e)^4 + 300*C*a*c^3*f*x*tan(f*x)^
```

4*tan(e)^4 + 300*B*b*c^3*f*x*tan(f*x)^4*tan(e)^4 + 900*B*a*c^2*d*f*x*tan(f*x)^4*tan(e)^4 + 900*A*b*c^2*d*f*x*tan(f*x)^4*tan(e)^4 - 900*C*b*c^2*d*f*x*tan(f*x)^4*tan(e)^4 + 900*A*a*c*d^2*f*x*tan(f*x)^4*tan(e)^4 - 900*C*a*c*d^2*f*x*tan(f*x)^4*tan(e)^4 - 900*B*b*c*d^2*f*x*tan(f*x)^4*tan(e)^4 - 300*B*a*d^3*f*x*tan(f*x)^4*tan(e)^4 - 300*A*b*d^3*f*x*tan(f*x)^4*tan(e)^4 + 300*C*b*d^3*f*x*tan(f*x)^4*tan(e)^4 + 30*C*b*c^3*tan(f*x)^5*tan(e)^5 + 90*C*a*c^2*d*tan(f*x)^5*tan(e)^5 + 90*B*b*c^2*d*tan(f*x)^5*tan(e)^5 + 90*B*a*c*d^2*tan(f*x)^5*tan(e)^5 + 90*A*b*c*d^2*tan(f*x)^5*tan(e)^5 - 135*C*b*c*d^2*tan(f*x)^5*tan(e)^5 + 30*A*a*d^3*tan(f*x)^5*tan(e)^5 - 45*C*a*d^3*tan(f*x)^5*tan(e)^5 - 45*B*b*d^3*tan(f*x)^5*tan(e)^5 + 150*B*a*c^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 150*A*b*c^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 - 150*C*b*c^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 450*A*a*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 - 450*C*a*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 - 450*B*b*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 - 450*B*a*c*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 - 450*A*b*c*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 450*C*b*c*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 - 150*A*a*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 150*C*a*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 150*B*b*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 - 60*C*a*c^3*tan(f*x)^5*tan(e)^4 - 60*B*b*c^3*tan(f*x)^5*tan(e)^4 - 180*B*a*c^2*d*tan(f*x)^5*tan(e)^4 - 180*A*b*c^2*d*tan(f*x)^5*tan(e)^4 + 180*C*b*c^2*d*tan(f*x)^5*tan(e)^4 - 180*A*a*c*d^2*tan(f*x)^5*tan(e)^4 + 180*C*a*c*d^2*tan(f*x)^5*tan(e)^4 + 180*B*b*c*d^2*tan(f*x)^5*tan(e)^4 + 60*B*a*d^3*tan(f*x)^5*tan(e)^4 + 60*A*b*d^3*tan(f*x)^5*tan(e)^4 - 60*C*b*d^3*tan(f*x)^5*tan(e)^4 - 60*C*a*c^3*tan(f*x)^4*tan(e)^5 - 60*B*b*c^3*tan(f*x)^4*tan(e)^5 - 180*B*a*c^2*d*tan(f*x)^4*tan(e)^5 - 180*A*b*c^2*d*tan(f*x)^4*tan(e)^5 + 180*C*b*c^2*d*tan(f*x)^4*tan(e)^5 - 180*A*a*c*d^2*tan(f*x)^4*tan(e)^5 + 180*C*a*c*d^2*tan(f*x)^4*tan(e)^5 + 180*B*b*c*d^2*tan(f*x)^4*tan(e)^5 + 60*B*a*d^3*tan(f*x)^4*tan(e)^5 + 60*A*b*d^3*tan(f*x)^4*tan(e)^5 - 60*C*b*d^3*tan(f*x)^4*tan(e)^5 + 600*A*a*c^3*f*x*tan(f*x)^3*tan(e)^3 - 600*C*a*c^3*f*x*tan(f*x)^3*tan(e)^3 - 600*B*b*c^3*f*x*tan(f*x)^3*tan(e)^3 - 1800*B*a*c^2*d*f*x*tan(f*x)^3*tan(e)^3 - 1800*A*b*c^2*d*f*x*tan(f*x)^3*tan(e)^3 + 1800*C*b*c^2*d*f*x*tan(f*x)^3*tan(e)^3 - 1800*A*

a*c*d^2*f*x*tan(f*x)^3*tan(e)^3 + 1800*C*a*c*d^2*f*x*tan(f*x)^3*tan(e)^3 +
1800*B*b*c*d^2*f*x*tan(f*x)^3*tan(e)^3 + 600*B*a*d^3*f*x*tan(f*x)^3*tan(e)^
3 + 600*A*b*d^3*f*x*tan(f*x)^3*tan(e)^3 - 600*C*b*d^3*f*x*tan(f*x)^3*tan(e)
^3 + 30*C*b*c^3*tan(f*x)^5*tan(e)^3 + 90*C*a*c^2*d*tan(f*x)^5*tan(e)^3 + 90
*B*b*c^2*d*tan(f*x)^5*tan(e)^3 + 90*B*a*c*d^2*tan(f*x)^5*tan(e)^3 + 90*A*b*
c*d^2*tan(f*x)^5*tan(e)^3 - 90*C*b*c*d^2*tan(f*x)^5*tan(e)^3 + 30*A*a*d^3*t
an(f*x)^5*tan(e)^3 - 30*C*a*d^3*tan(f*x)^5*tan(e)^3 - 30*B*b*d^3*tan(f*x)^5
*tan(e)^3 - 90*C*b*c^3*tan(f*x)^4*tan(e)^4 - 270*C*a*c^2*d*tan(f*x)^4*tan(e
)^4 - 270*B*b*c^2*d*tan(f*x)^4*tan(e)^4 - 270*B*a*c*d^2*tan(f*x)^4*tan(e)^4
 - 270*A*b*c*d^2*tan(f*x)^4*tan(e)^4 + 495*C*b*c*d^2*tan(f*x)^4*tan(e)^4 -
90*A*a*d^3*tan(f*x)^4*tan(e)^4 + 165*C*a*d^3*tan(f*x)^4*tan(e)^4 + 165*B*b*
d^3*tan(f*x)^4*tan(e)^4 + 30*C*b*c^3*tan(f*x)^3*tan(e)^5 + 90*C*a*c^2*d*tan
(f*x)^3*tan(e)^5 + 90*B*b*c^2*d*tan(f*x)^3*tan(e)^5 + 90*B*a*c*d^2*tan(f*x)
^3*tan(e)^5 + 90*A*b*c*d^2*tan(f*x)^3*tan(e)^5 - 90*C*b*c*d^2*tan(f*x)^3*ta
n(e)^5 + 30*A*a*d^3*tan(f*x)^3*tan(e)^5 - 30*C*a*d^3*tan(f*x)^3*tan(e)^5 -
30*B*b*d^3*tan(f*x)^3*tan(e)^5 - 60*C*b*c^2*d*tan(f*x)^5*tan(e)^2 - 60*C*a*
c*d^2*tan(f*x)^5*tan(e)^2 - 60*B*b*c*d^2*tan(f*x)^5*tan(e)^2 - 20*B*a*d^3*t
an(f*x)^5*tan(e)^2 - 20*A*b*d^3*tan(f*x)^5*tan(e)^2 + 20*C*b*d^3*tan(f*x)^5
*tan(e)^2 - 300*B*a*c^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)
/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 3
00*A*b*c^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*
tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 300*C*b*c^3*lo
g(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + ta
n(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 900*A*a*c^2*d*log(4*(tan(f*
x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 +
tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 900*C*a*c^2*d*log(4*(tan(f*x)^2*tan(e)
^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 +
1))*tan(f*x)^3*tan(e)^3 + 900*B*b*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(
f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x
)^3*tan(e)^3 + 900*B*a*c*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e)
 + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^
3 + 900*A*b*c*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(
f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 900*C*b
*c*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(
e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 300*A*a*d^3*log(4*
(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*
x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 300*C*a*d^3*log(4*(tan(f*x)^2*t
an(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)
^2 + 1))*tan(f*x)^3*tan(e)^3 - 300*B*b*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*t
an(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(
f*x)^3*tan(e)^3 + 240*C*a*c^3*tan(f*x)^4*tan(e)^3 + 240*B*b*c^3*tan(f*x)^4*
tan(e)^3 + 720*B*a*c^2*d*tan(f*x)^4*tan(e)^3 + 720*A*b*c^2*d*tan(f*x)^4*tan
(e)^3 - 900*C*b*c^2*d*tan(f*x)^4*tan(e)^3 + 720*A*a*c*d^2*tan(f*x)^4*tan(e)
^3 - 900*C*a*c*d^2*tan(f*x)^4*tan(e)^3 - 900*B*b*c*d^2*tan(f*x)^4*tan(e)^3
- 300*B*a*d^3*tan(f*x)^4*tan(e)^3 - 300*A*b*d^3*tan(f*x)^4*tan(e)^3 + 300*C

*b*d^3*tan(f*x)^4*tan(e)^3 + 240*C*a*c^3*tan(f*x)^3*tan(e)^4 + 240*B*b*c^3*
tan(f*x)^3*tan(e)^4 + 720*B*a*c^2*d*tan(f*x)^3*tan(e)^4 + 720*A*b*c^2*d*tan
(f*x)^3*tan(e)^4 - 900*C*b*c^2*d*tan(f*x)^3*tan(e)^4 + 720*A*a*c*d^2*tan(f*
x)^3*tan(e)^4 - 900*C*a*c*d^2*tan(f*x)^3*tan(e)^4 - 900*B*b*c*d^2*tan(f*x)^
3*tan(e)^4 - 300*B*a*d^3*tan(f*x)^3*tan(e)^4 - 300*A*b*d^3*tan(f*x)^3*tan(e
)^4 + 300*C*b*d^3*tan(f*x)^3*tan(e)^4 - 60*C*b*c^2*d*tan(f*x)^2*tan(e)^5 -
60*C*a*c*d^2*tan(f*x)^2*tan(e)^5 - 60*B*b*c*d^2*tan(f*x)^2*tan(e)^5 - 20*B*
a*d^3*tan(f*x)^2*tan(e)^5 - 20*A*b*d^3*tan(f*x)^2*tan(e)^5 + 20*C*b*d^3*tan
(f*x)^2*tan(e)^5 + 45*C*b*c*d^2*tan(f*x)^5*tan(e) + 15*C*a*d^3*tan(f*x)^5*t
an(e) + 15*B*b*d^3*tan(f*x)^5*tan(e) - 600*A*a*c^3*f*x*tan(f*x)^2*tan(e)^2
+ 600*C*a*c^3*f*x*tan(f*x)^2*tan(e)^2 + 600*B*b*c^3*f*x*tan(f*x)^2*tan(e)^2
 + 1800*B*a*c^2*d*f*x*tan(f*x)^2*tan(e)^2 + 1800*A*b*c^2*d*f*x*tan(f*x)^2*t
an(e)^2 - 1800*C*b*c^2*d*f*x*tan(f*x)^2*tan(e)^2 + 1800*A*a*c*d^2*f*x*tan(f
*x)^2*tan(e)^2 - 1800*C*a*c*d^2*f*x*tan(f*x)^2*tan(e)^2 - 1800*B*b*c*d^2*f*
x*tan(f*x)^2*tan(e)^2 - 600*B*a*d^3*f*x*tan(f*x)^2*tan(e)^2 - 600*A*b*d^3*f
*x*tan(f*x)^2*tan(e)^2 + 600*C*b*d^3*f*x*tan(f*x)^2*tan(e)^2 - 90*C*b*c^3*t
an(f*x)^4*tan(e)^2 - 270*C*a*c^2*d*tan(f*x)^4*tan(e)^2 - 270*B*b*c^2*d*tan(
f*x)^4*tan(e)^2 - 270*B*a*c*d^2*tan(f*x)^4*tan(e)^2 - 270*A*b*c*d^2*tan(f*x
)^4*tan(e)^2 + 450*C*b*c*d^2*tan(f*x)^4*tan(e)^2 - 90*A*a*d^3*tan(f*x)^4*ta
n(e)^2 + 150*C*a*d^3*tan(f*x)^4*tan(e)^2 + 150*B*b*d^3*tan(f*x)^4*tan(e)^2
+ 120*C*b*c^3*tan(f*x)^3*tan(e)^3 + 360*C*a*c^2*d*tan(f*x)^3*tan(e)^3 + 360
*B*b*c^2*d*tan(f*x)^3*tan(e)^3 + 360*B*a*c*d^2*tan(f*x)^3*tan(e)^3 + 360*A*
b*c*d^2*tan(f*x)^3*tan(e)^3 - 540*C*b*c*d^2*tan(f*x)^3*tan(e)^3 + 120*A*a*d
^3*tan(f*x)^3*tan(e)^3 - 180*C*a*d^3*tan(f*x)^3*tan(e)^3 - 180*B*b*d^3*tan(
f*x)^3*tan(e)^3 - 90*C*b*c^3*tan(f*x)^2*tan(e)^4 - 270*C*a*c^2*d*tan(f*x)^2
*tan(e)^4 - 270*B*b*c^2*d*tan(f*x)^2*tan(e)^4 - 270*B*a*c*d^2*tan(f*x)^2*ta
n(e)^4 - 270*A*b*c*d^2*tan(f*x)^2*tan(e)^4 + 450*C*b*c*d^2*tan(f*x)^2*tan(e
)^4 - 90*A*a*d^3*tan(f*x)^2*tan(e)^4 + 150*C*a*d^3*tan(f*x)^2*tan(e)^4 + 15
0*B*b*d^3*tan(f*x)^2*tan(e)^4 + 45*C*b*c*d^2*tan(f*x)*tan(e)^5 + 15*C*a*d^3
*tan(f*x)*tan(e)^5 + 15*B*b*d^3*tan(f*x)*tan(e)^5 - 12*C*b*d^3*tan(f*x)^5 +
 120*C*b*c^2*d*tan(f*x)^4*tan(e) + 120*C*a*c*d^2*tan(f*x)^4*tan(e) + 120*B*
b*c*d^2*tan(f*x)^4*tan(e) + 40*B*a*d^3*tan(f*x)^4*tan(e) + 40*A*b*d^3*tan(f
*x)^4*tan(e) - 100*C*b*d^3*tan(f*x)^4*tan(e) + 300*B*a*c^3*log(4*(tan(f*x)^
2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan
(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 300*A*b*c^3*log(4*(tan(f*x)^2*tan(e)^2 -
2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*t
an(f*x)^2*tan(e)^2 - 300*C*b*c^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*ta
n(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan
(e)^2 + 900*A*a*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(
tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 900
*C*a*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*
tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 900*B*b*c^2*d*
log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 +
tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 900*B*a*c*d^2*log(4*(tan(
f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2

+ tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 900*A*b*c*d^2*log(4*(tan(f*x)^2*tan(
e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2
+ 1))*tan(f*x)^2*tan(e)^2 + 900*C*b*c*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*ta
n(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f
*x)^2*tan(e)^2 - 300*A*a*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e)
 + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^
2 + 300*C*a*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*
x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 300*B*b*d
^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2
 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 360*C*a*c^3*tan(f*x)^3
*tan(e)^2 - 360*B*b*c^3*tan(f*x)^3*tan(e)^2 - 1080*B*a*c^2*d*tan(f*x)^3*tan
(e)^2 - 1080*A*b*c^2*d*tan(f*x)^3*tan(e)^2 + 1440*C*b*c^2*d*tan(f*x)^3*tan(
e)^2 - 1080*A*a*c*d^2*tan(f*x)^3*tan(e)^2 + 1440*C*a*c*d^2*tan(f*x)^3*tan(e
)^2 + 1440*B*b*c*d^2*tan(f*x)^3*tan(e)^2 + 480*B*a*d^3*tan(f*x)^3*tan(e)^2
+ 480*A*b*d^3*tan(f*x)^3*tan(e)^2 - 600*C*b*d^3*tan(f*x)^3*tan(e)^2 - 360*C
*a*c^3*tan(f*x)^2*tan(e)^3 - 360*B*b*c^3*tan(f*x)^2*tan(e)^3 - 1080*B*a*c^2
*d*tan(f*x)^2*tan(e)^3 - 1080*A*b*c^2*d*tan(f*x)^2*tan(e)^3 + 1440*C*b*c^2*
d*tan(f*x)^2*tan(e)^3 - 1080*A*a*c*d^2*tan(f*x)^2*tan(e)^3 + 1440*C*a*c*d^2
*tan(f*x)^2*tan(e)^3 + 1440*B*b*c*d^2*tan(f*x)^2*tan(e)^3 + 480*B*a*d^3*tan
(f*x)^2*tan(e)^3 + 480*A*b*d^3*tan(f*x)^2*tan(e)^3 - 600*C*b*d^3*tan(f*x)^2
*tan(e)^3 + 120*C*b*c^2*d*tan(f*x)*tan(e)^4 + 120*C*a*c*d^2*tan(f*x)*tan(e)
^4 + 120*B*b*c*d^2*tan(f*x)*tan(e)^4 + 40*B*a*d^3*tan(f*x)*tan(e)^4 + 40*A*
b*d^3*tan(f*x)*tan(e)^4 - 100*C*b*d^3*tan(f*x)*tan(e)^4 - 12*C*b*d^3*tan(e)
^5 - 45*C*b*c*d^2*tan(f*x)^4 - 15*C*a*d^3*tan(f*x)^4 - 15*B*b*d^3*tan(f*x)^
4 + 300*A*a*c^3*f*x*tan(f*x)*tan(e) - 300*C*a*c^3*f*x*tan(f*x)*tan(e) - 300
*B*b*c^3*f*x*tan(f*x)*tan(e) - 900*B*a*c^2*d*f*x*tan(f*x)*tan(e) - 900*A*b*
c^2*d*f*x*tan(f*x)*tan(e) + 900*C*b*c^2*d*f*x*tan(f*x)*tan(e) - 900*A*a*c*d
^2*f*x*tan(f*x)*tan(e) + 900*C*a*c*d^2*f*x*tan(f*x)*tan(e) + 900*B*b*c*d^2*
f*x*tan(f*x)*tan(e) + 300*B*a*d^3*f*x*tan(f*x)*tan(e) + 300*A*b*d^3*f*x*tan
(f*x)*tan(e) - 300*C*b*d^3*f*x*tan(f*x)*tan(e) + 90*C*b*c^3*tan(f*x)^3*tan(
e) + 270*C*a*c^2*d*tan(f*x)^3*tan(e) + 270*B*b*c^2*d*tan(f*x)^3*tan(e) + 27
0*B*a*c*d^2*tan(f*x)^3*tan(e) + 270*A*b*c*d^2*tan(f*x)^3*tan(e) - 450*C*b*c
*d^2*tan(f*x)^3*tan(e) + 90*A*a*d^3*tan(f*x)^3*tan(e) - 150*C*a*d^3*tan(f*x
)^3*tan(e) - 150*B*b*d^3*tan(f*x)^3*tan(e) - 120*C*b*c^3*tan(f*x)^2*tan(e)^
2 - 360*C*a*c^2*d*tan(f*x)^2*tan(e)^2 - 360*B*b*c^2*d*tan(f*x)^2*tan(e)^2 -
 360*B*a*c*d^2*tan(f*x)^2*tan(e)^2 - 360*A*b*c*d^2*tan(f*x)^2*tan(e)^2 + 54
0*C*b*c*d^2*tan(f*x)^2*tan(e)^2 - 120*A*a*d^3*tan(f*x)^2*tan(e)^2 + 180*C*a
*d^3*tan(f*x)^2*tan(e)^2 + 180*B*b*d^3*tan(f*x)^2*tan(e)^2 + 90*C*b*c^3*tan
(f*x)*tan(e)^3 + 270*C*a*c^2*d*tan(f*x)*tan(e)^3 + 270*B*b*c^2*d*tan(f*x)*t
an(e)^3 + 270*B*a*c*d^2*tan(f*x)*tan(e)^3 + 270*A*b*c*d^2*tan(f*x)*tan(e)^3
 - 450*C*b*c*d^2*tan(f*x)*tan(e)^3 + 90*A*a*d^3*tan(f*x)*tan(e)^3 - 150*C*a
*d^3*tan(f*x)*tan(e)^3 - 150*B*b*d^3*tan(f*x)*tan(e)^3 - 45*C*b*c*d^2*tan(e
)^4 - 15*C*a*d^3*tan(e)^4 - 15*B*b*d^3*tan(e)^4 - 60*C*b*c^2*d*tan(f*x)^3 -
 60*C*a*c*d^2*tan(f*x)^3 - 60*B*b*c*d^2*tan(f*x)^3 - 20*B*a*d^3*tan(f*x)^3
- 20*A*b*d^3*tan(f*x)^3 + 20*C*b*d^3*tan(f*x)^3 - 150*B*a*c^3*log(4*(tan(f*

x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) - 150*A*b*c^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 150*C*b*c^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) - 450*A*a*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 450*C*a*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 450*B*b*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 450*B*a*c*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 450*A*b*c*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) - 450*C*b*c*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 150*A*a*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) - 150*C*a*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) - 150*B*b*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 240*C*a*c^3*tan(f*x)^2*tan(e) + 240*B*b*c^3*tan(f*x)^2*tan(e) + 720*B*a*c^2*d*tan(f*x)^2*tan(e) + 720*A*b*c^2*d*tan(f*x)^2*tan(e) - 900*C*b*c^2*d*tan(f*x)^2*tan(e) + 720*A*a*c*d^2*tan(f*x)^2*tan(e) - 900*C*a*c*d^2*tan(f*x)^2*tan(e) - 900*B*b*c*d^2*tan(f*x)^2*tan(e) - 300*B*a*d^3*tan(f*x)^2*tan(e) - 300*A*b*d^3*tan(f*x)^2*tan(e) + 300*C*b*d^3*tan(f*x)^2*tan(e) + 240*C*a*c^3*tan(f*x)*tan(e)^2 + 240*B*b*c^3*tan(f*x)*tan(e)^2 + 720*B*a*c^2*d*tan(f*x)*tan(e)^2 + 720*A*b*c^2*d*tan(f*x)*tan(e)^2 - 900*C*b*c^2*d*tan(f*x)*tan(e)^2 + 720*A*a*c*d^2*tan(f*x)*tan(e)^2 - 900*C*a*c*d^2*tan(f*x)*tan(e)^2 - 900*B*b*c*d^2*tan(f*x)*tan(e)^2 - 300*B*a*d^3*tan(f*x)*tan(e)^2 - 300*A*b*d^3*tan(f*x)*tan(e)^2 + 300*C*b*d^3*tan(f*x)*tan(e)^2 - 60*C*b*c^2*d*tan(e)^3 - 60*C*a*c*d^2*tan(e)^3 - 60*B*b*c*d^2*tan(e)^3 - 20*B*a*d^3*tan(e)^3 - 20*A*b*d^3*tan(e)^3 + 20*C*b*d^3*tan(e)^3 - 60*A*a*c^3*f*x + 60*C*a*c^3*f*x + 60*B*b*c^3*f*x + 180*B*a*c^2*d*f*x + 180*A*b*c^2*d*f*x - 180*C*b*c^2*d*f*x + 180*A*a*c*d^2*f*x - 180*C*a*c*d^2*f*x - 180*B*b*c*d^2*f*x - 60*B*a*d^3*f*x - 60*A*b*d^3*f*x + 60*C*b*d^3*f*x - 30*C*b*c^3*tan(f*x)^2 - 90*C*a*c^2*d*tan(f*x)^2 - 90*B*b*c^2*d*tan(f*x)^2 - 90*B*a*c*d^2*tan(f*x)^2 - 90*A*b*c*d^2*tan(f*x)^2 + 90*C*b*c*d^2*tan(f*x)^2 - 30*A*a*d^3*tan(f*x)^2 + 30*C*a*d^3*tan(f*x)^2 + 30*B*b*d^3*tan(f*x)^2 + 90*C*b*c^3*tan(f*x)*tan(e) + 270*C*a*c^2*d*tan(f*x)*tan(e) + 270*B*b*c^2*d*tan(f*x)*tan(e) + 270*B*a*c*d^2*tan(f*x)*tan(e) + 270*A*b*c*d^2*tan(f*x)*tan(e) - 495*C*b*c*d^2*tan(f*x)*tan(e) + 90*A*a*d^3*tan(f*x)*tan(e) - 165*C*a*d^3*tan(f*x)*tan(e) - 165*B*b*d^3*tan(f*x)*tan(e) - 30*C*b*c^3*tan(e)^2 - 90*C*a*c^2*d*tan(e)^2 - 90*B*b*c^2*d*tan(e)^2 - 90*B*a*c*d^2*tan(e)^2 - 90*A*b*c*d^2*tan(e)^2 + 90*C*b*c*d^2*tan(e)^2 - 30*A*a*d^3*tan(e)^2 + 30*C*a*d^3*tan(e)^2 + 30*B*b*d^3*tan (e)^2 + 30*B*a*c^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) + 30*A*b*c^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 30*C*b*c^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) + 90*A*a*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 90*C*a*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 90*B*b*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 90*B*a*c*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 90*A*b*c*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) + 90*C*b*c*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 30*A*a*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) + 30*C*a*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) + 30*B*b*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 60*C*a*c^3*tan(f*x) - 60*B*b*c^3*tan(f*x) - 180*B*a*c^2*d*tan(f*x) - 180*A*b*c^2*d*tan(f*x) + 180*C*b*c^2*d*tan(f*x) - 180*A*a*c*d^2*tan(f*x) + 180*C*a*c*d^2*tan(f*x) + 180*B*b*c*d^2*tan(f*x) + 60*B*a*d^3*tan(f*x) + 60*A*b*d^3*tan(f*x) - 60*C*b*d^3*tan(f*x) - 60*C*a*c^3*tan(e) - 60*B*b*c^3*tan(e) - 180*B*a*c^2*d*tan(e) - 180*A*b*c^2*d*tan(e) + 180*C*b*c^2*d*tan(e) - 180*A*a*c*d^2*tan(e) + 180*C*a*c*d^2*tan(e) + 180*B*b*c*d^2*tan(e) + 60*B*a*d^3*tan(e) + 60*A*b*d^3*tan(e) - 60*C*b*d^3*tan(e) - 30*C*b*c^3 - 90*C*a*c^2*d - 90*B*b*c^2*d - 90*B*a*c*d^2 - 90*A*b*c*d^2 + 135*C*b*c*d^2 - 30*A*a*d^3 + 45*C*a*d^3 + 45*B*b*d^3)/(f*tan(f*x)^5*tan(e)^5 - 5*f*tan(f*x)^4*tan(e)^4 + 10*f*tan(f*x)^3*tan(e)^3 - 10*f*tan(f*x)^2*tan(e)^2 + 5*f*tan(f*x)*tan(e) - f)

Mupad [B] (verification not implemented)

Time = 8.38 (sec) , antiderivative size = 478, normalized size of antiderivative = 1.23

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$

$$= x\left(A\,a\,c^3+A\,b\,d^3+B\,a\,d^3-B\,b\,c^3-C\,a\,c^3-C\,b\,d^3-3\,A\,a\,c\,d^2-3\,A\,b\,c^2\,d-3\,B\,a\,c^2\,d+3\,B\,b\,c\,d^2+3\,C\,a\,c\,d^2+3\,C\,b\,c^2\,d\right)+\frac{\tan(e+f\,x)^4\left(\frac{B\,b\,d^3}{4}+\frac{C\,a\,d^3}{4}+\frac{3\,C\,b\,c\,d^2}{4}\right)}{f}$$

$$+\frac{\tan(e+f\,x)^3\left(\frac{A\,b\,d^3}{3}+\frac{B\,a\,d^3}{3}-\frac{C\,b\,d^3}{3}+B\,b\,c\,d^2+C\,a\,c\,d^2+C\,b\,c^2\,d\right)}{f}$$

$$+\frac{\tan(e+f\,x)^2\left(\frac{A\,a\,d^3}{2}-\frac{B\,b\,d^3}{2}-\frac{C\,a\,d^3}{2}+\frac{C\,b\,c^3}{2}+\frac{3\,A\,b\,c\,d^2}{2}+\frac{3\,B\,a\,c\,d^2}{2}+\frac{3\,B\,b\,c^2\,d}{2}+\frac{3\,C\,a\,c^2\,d}{2}-\frac{3\,C\,b\,c\,d^2}{2}\right)}{f}$$

$$-\frac{\ln\left(\tan(e+f\,x)^2+1\right)\left(\frac{A\,a\,d^3}{2}-\frac{A\,b\,c^3}{2}-\frac{B\,a\,c^3}{2}-\frac{B\,b\,d^3}{2}-\frac{C\,a\,d^3}{2}+\frac{C\,b\,c^3}{2}-\frac{3\,A\,a\,c^2\,d}{2}+\frac{3\,A\,b\,c\,d^2}{2}+\frac{3\,B\,a\,c\,d^2}{2}+\frac{3\,B\ldots}{}\right.}{f}$$

$$+\frac{\tan(e+f\,x)\,\left(B\,b\,c^3-B\,a\,d^3-A\,b\,d^3+C\,a\,c^3+C\,b\,d^3+3\,A\,a\,c\,d^2+3\,A\,b\,c^2\,d+3\,B\,a\,c^2\,d-3\,B\,b\,c\,d^2-\ldots\right.}{f}$$

$$+\frac{C\,b\,d^3\tan(e+f\,x)^5}{5\,f}$$

```
[In] int((a + b*tan(e + f*x))*(c + d*tan(e + f*x))^3*(A + B*tan(e + f*x) + C*tan
(e + f*x)^2),x)
```

```
[Out] x*(A*a*c^3 + A*b*d^3 + B*a*d^3 - B*b*c^3 - C*a*c^3 - C*b*d^3 - 3*A*a*c*d^2
- 3*A*b*c^2*d - 3*B*a*c^2*d + 3*B*b*c*d^2 + 3*C*a*c*d^2 + 3*C*b*c^2*d) + (t
an(e + f*x)^4*((B*b*d^3)/4 + (C*a*d^3)/4 + (3*C*b*c*d^2)/4))/f + (tan(e + f
*x)^3*((A*b*d^3)/3 + (B*a*d^3)/3 - (C*b*d^3)/3 + B*b*c*d^2 + C*a*c*d^2 + C*
b*c^2*d))/f + (tan(e + f*x)^2*((A*a*d^3)/2 - (B*b*d^3)/2 - (C*a*d^3)/2 + (C
*b*c^3)/2 + (3*A*b*c*d^2)/2 + (3*B*a*c*d^2)/2 + (3*B*b*c^2*d)/2 + (3*C*a*c^
2*d)/2 - (3*C*b*c*d^2)/2))/f - (log(tan(e + f*x)^2 + 1)*((A*a*d^3)/2 - (A*b
*c^3)/2 - (B*a*c^3)/2 - (B*b*d^3)/2 - (C*a*d^3)/2 + (C*b*c^3)/2 - (3*A*a*c^
2*d)/2 + (3*A*b*c*d^2)/2 + (3*B*a*c*d^2)/2 + (3*B*b*c^2*d)/2 + (3*C*a*c^2*d
)/2 - (3*C*b*c*d^2)/2))/f + (tan(e + f*x)*(B*b*c^3 - B*a*d^3 - A*b*d^3 + C*
a*c^3 + C*b*d^3 + 3*A*a*c*d^2 + 3*A*b*c^2*d + 3*B*a*c^2*d - 3*B*b*c*d^2 - 3
*C*a*c*d^2 - 3*C*b*c^2*d))/f + (C*b*d^3*tan(e + f*x)^5)/(5*f)
```

3.66 $\int (c+d\tan(e+fx))^3 \left(A+B\tan(e+fx)+C\tan^2(e+\right.$

Optimal result 615
Rubi [A] (verified) 615
Mathematica [C] (verified) 618
Maple [A] (verified) 618
Fricas [A] (verification not implemented) 619
Sympy [B] (verification not implemented) 619
Maxima [A] (verification not implemented) 620
Giac [B] (verification not implemented) 620
Mupad [B] (verification not implemented) 623

Optimal result

Integrand size = 33, antiderivative size = 191

$$
\begin{aligned}
&\int (c+d\tan(e+fx))^3 \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx \\
&= -\left(\left(c^3C+3Bc^2d-3cCd^2-Bd^3-A\left(c^3-3cd^2\right)\right)x\right) \\
&\quad -\frac{((A-C)d(3c^2-d^2)+B(c^3-3cd^2))\log(\cos(e+fx))}{f} \\
&\quad +\frac{d(2c(A-C)d+B(c^2-d^2))\tan(e+fx)}{f}+\frac{(Bc+(A-C)d)(c+d\tan(e+fx))^2}{2f} \\
&\quad +\frac{B(c+d\tan(e+fx))^3}{3f}+\frac{C(c+d\tan(e+fx))^4}{4df}
\end{aligned}
$$

```
[Out] -(c^3*C+3*B*c^2*d-3*C*c*d^2-B*d^3-A*(c^3-3*c*d^2))*x-((A-C)*d*(3*c^2-d^2)+B
*(c^3-3*c*d^2))*ln(cos(f*x+e))/f+d*(2*c*(A-C)*d+B*(c^2-d^2))*tan(f*x+e)/f+1
/2*(B*c+(A-C)*d)*(c+d*tan(f*x+e))^2/f+1/3*B*(c+d*tan(f*x+e))^3/f+1/4*C*(c+d
*tan(f*x+e))^4/d/f
```

Rubi [A] (verified)

Time = 0.28 (sec) , antiderivative size = 191, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 4, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.121, Rules used

$= \{3711, 3609, 3606, 3556\}$

$$\int (c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\frac{d\tan(e+fx)\left(2cd(A-C)+B(c^2-d^2)\right)}{f}$$
$$-\frac{\left(d(A-C)\left(3c^2-d^2\right)+B(c^3-3cd^2)\right)\log(\cos(e+fx))}{f}$$
$$-x\left(-A\left(c^3-3cd^2\right)+3Bc^2d-Bd^3+c^3C-3cCd^2\right)$$
$$+\frac{(d(A-C)+Bc)(c+d\tan(e+fx))^2}{2f}$$
$$+\frac{B(c+d\tan(e+fx))^3}{3f}+\frac{C(c+d\tan(e+fx))^4}{4df}$$

```
[In] Int[(c + d*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] -((c^3*C + 3*B*c^2*d - 3*c*C*d^2 - B*d^3 - A*(c^3 - 3*c*d^2))*x) - (((A - C
)*d*(3*c^2 - d^2) + B*(c^3 - 3*c*d^2))*Log[Cos[e + f*x]])/f + (d*(2*c*(A -
C)*d + B*(c^2 - d^2))*Tan[e + f*x])/f + ((B*c + (A - C)*d)*(c + d*Tan[e + f
*x])^2)/(2*f) + (B*(c + d*Tan[e + f*x])^3)/(3*f) + (C*(c + d*Tan[e + f*x])^
4)/(4*d*f)
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3606

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
 f*x], x], x] + Simp[b*d*(Tan[e + f*x]/f), x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3609

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[d*((a + b*Tan[e + f*x])^m/(f*m)), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
```

```
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \frac{C(c+d\tan(e+fx))^4}{4df} + \int (A-C+B\tan(e+fx))(c+d\tan(e+fx))^3\,dx \\
&= \frac{B(c+d\tan(e+fx))^3}{3f} + \frac{C(c+d\tan(e+fx))^4}{4df} \\
&\quad + \int (c+d\tan(e+fx))^2(Ac-cC-Bd+(Bc+(A-C)d)\tan(e+fx))\,dx \\
&= \frac{(Bc+(A-C)d)(c+d\tan(e+fx))^2}{2f} + \frac{B(c+d\tan(e+fx))^3}{3f} \\
&\quad + \frac{C(c+d\tan(e+fx))^4}{4df} + \int (c+d\tan(e+fx))\left(-c^2C-2Bcd+Cd^2\right. \\
&\qquad\qquad \left.+A\left(c^2-d^2\right)+\left(2c(A-C)d+B\left(c^2-d^2\right)\right)\tan(e+fx)\right)\,dx \\
&= -\left(\left(c^3C+3Bc^2d-3cCd^2-Bd^3-A\left(c^3-3cd^2\right)\right)x\right) \\
&\quad + \frac{d(2c(A-C)d+B(c^2-d^2))\tan(e+fx)}{f} \\
&\quad + \frac{(Bc+(A-C)d)(c+d\tan(e+fx))^2}{2f} + \frac{B(c+d\tan(e+fx))^3}{3f} \\
&\quad + \frac{C(c+d\tan(e+fx))^4}{4df} + \left((A-C)d\left(3c^2-d^2\right)+B\left(c^3-3cd^2\right)\right)\int \tan(e+fx)\,dx \\
&= -\left(\left(c^3C+3Bc^2d-3cCd^2-Bd^3-A\left(c^3-3cd^2\right)\right)x\right) \\
&\quad - \frac{((A-C)d(3c^2-d^2)+B(c^3-3cd^2))\log(\cos(e+fx))}{f} \\
&\quad + \frac{d(2c(A-C)d+B(c^2-d^2))\tan(e+fx)}{f} \\
&\quad + \frac{(Bc+(A-C)d)(c+d\tan(e+fx))^2}{2f} \\
&\quad + \frac{B(c+d\tan(e+fx))^3}{3f} + \frac{C(c+d\tan(e+fx))^4}{4df}
\end{aligned}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 2.61 (sec) , antiderivative size = 212, normalized size of antiderivative = 1.11

$$\int (c+d\tan(e+fx))^3 \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right) dx$$
$$= \frac{3C(c+d\tan(e+fx))^4-6(Bc+(-A+C)d)\left((ic-d)^3\log(i-\tan(e+fx))-(ic+d)^3\log(i+\tan(e+\ldots\right.}{}$$

```
[In] Integrate[(c + d*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] (3*C*(c + d*Tan[e + f*x])^4 - 6*(B*c + (-A + C)*d)*((I*c - d)^3*Log[I - Tan
[e + f*x]] - (I*c + d)^3*Log[I + Tan[e + f*x]] + 6*c*d^2*Tan[e + f*x] + d^3
*Tan[e + f*x]^2) + 2*B*((-3*I)*(c + I*d)^4*Log[I - Tan[e + f*x]] + (3*I)*(c
 - I*d)^4*Log[I + Tan[e + f*x]] - 6*d^2*(-6*c^2 + d^2)*Tan[e + f*x] + 12*c*
d^3*Tan[e + f*x]^2 + 2*d^4*Tan[e + f*x]^3))/(12*d*f)
```

Maple [A] (verified)

Time = 0.11 (sec) , antiderivative size = 210, normalized size of antiderivative = 1.10

method	result
parts	$A c^3 x+\frac{(3A c^2 d+B c^3)\ln\left(1+\tan(fx+e)^2\right)}{2f}+\frac{(B d^3+3Cc d^2)\left(\frac{\tan(fx+e)^3}{3}-\tan(fx+e)+\arctan(\tan(fx+e))\right)}{f}+\ldots$
norman	$(A c^3-3Ac d^2-3B c^2 d+B d^3-c^3 C+3Cc d^2)x+\frac{(3Ac d^2+3B c^2 d-B d^3+c^3 C-3Cc d^2)\tan(fx+\ldots}{f}$
derivativedivides	$\frac{C d^3\tan(fx+e)^4}{4}+\frac{B d^3\tan(fx+e)^3}{3}+Cc d^2\tan(fx+e)^3+\frac{A d^3\tan(fx+e)^2}{2}+\frac{3Bc d^2\tan(fx+e)^2}{2}+\frac{3C c^2 d\tan(fx+e)^2}{2}-\frac{C d^3\tan(\ldots}{2}$
default	$\frac{C d^3\tan(fx+e)^4}{4}+\frac{B d^3\tan(fx+e)^3}{3}+Cc d^2\tan(fx+e)^3+\frac{A d^3\tan(fx+e)^2}{2}+\frac{3Bc d^2\tan(fx+e)^2}{2}+\frac{3C c^2 d\tan(fx+e)^2}{2}-\frac{C d^3\tan(\ldots}{2}$
parallelrisch	$\frac{3C d^3\tan(fx+e)^4+4B d^3\tan(fx+e)^3+6A d^3\tan(fx+e)^2-6C d^3\tan(fx+e)^2-12\tan(fx+e)B d^3+12\tan(fx+e)c^3C+1\ldots}{}$
risch	$A c^3 x+B d^3 x-C c^3 x+\frac{2i(9Ac d^2-12Cc d^2+9B c^2 d-4B d^3+3c^3C-36Cc d^2 e^{4i(fx+e)}+27Ac d^2 e^{2i(fx+e)}+27\ldots}{}$

```
[In] int((c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x,method=_RETURNVERB
OSE)
```

```
[Out] A*c^3*x+1/2*(3*A*c^2*d+B*c^3)/f*ln(1+tan(f*x+e)^2)+(B*d^3+3*C*c*d^2)/f*(1/3
*tan(f*x+e)^3-tan(f*x+e)+arctan(tan(f*x+e)))+(A*d^3+3*B*c*d^2+3*C*c^2*d)/f*
(1/2*tan(f*x+e)^2-1/2*ln(1+tan(f*x+e)^2))+(3*A*c*d^2+3*B*c^2*d+C*c^3)/f*(ta
n(f*x+e)-arctan(tan(f*x+e)))+C*d^3/f*(1/4*tan(f*x+e)^4-1/2*tan(f*x+e)^2+1/2
*ln(1+tan(f*x+e)^2))
```

Fricas [A] (verification not implemented)

none

Time = 0.26 (sec) , antiderivative size = 201, normalized size of antiderivative = 1.05

$$\int (c+d\tan(e+fx))^3 \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= \frac{3\,Cd^3\tan\left(fx+e\right)^4+4\left(3\,Ccd^2+Bd^3\right)\tan\left(fx+e\right)^3+12\left((A-C)c^3-3\,Bc^2d-3\,(A-C)cd^2+Bd^3\right)f}{}$$

```
[In] integrate((c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorithm="
fricas")
```

```
[Out] 1/12*(3*C*d^3*tan(f*x + e)^4 + 4*(3*C*c*d^2 + B*d^3)*tan(f*x + e)^3 + 12*((
A - C)*c^3 - 3*B*c^2*d - 3*(A - C)*c*d^2 + B*d^3)*f*x + 6*(3*C*c^2*d + 3*B*
c*d^2 + (A - C)*d^3)*tan(f*x + e)^2 - 6*(B*c^3 + 3*(A - C)*c^2*d - 3*B*c*d^
2 - (A - C)*d^3)*log(1/(tan(f*x + e)^2 + 1)) + 12*(C*c^3 + 3*B*c^2*d + 3*(A
 - C)*c*d^2 - B*d^3)*tan(f*x + e))/f
```

Sympy [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 410 vs. $2(163) = 326$.

Time = 0.17 (sec) , antiderivative size = 410, normalized size of antiderivative = 2.15

$$\int (c+d\tan(e+fx))^3 \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= \begin{cases} Ac^3x+\frac{3Ac^2d\log\left(\tan^2\left(e+fx\right)+1\right)}{2f}-3Acd^2x+\frac{3Acd^2\tan\left(e+fx\right)}{f}-\frac{Ad^3\log\left(\tan^2\left(e+fx\right)+1\right)}{2f}+\frac{Ad^3\tan^2\left(e+fx\right)}{2f}+\frac{Bc^3\log\left(\mathrm{t}\right.}{} \\ x(c+d\tan\left(e\right))^3\left(A+B\tan\left(e\right)+C\tan^2\left(e\right)\right) \end{cases}$$

```
[In] integrate((c+d*tan(f*x+e))**3*(A+B*tan(f*x+e)+C*tan(f*x+e)**2),x)
```

```
[Out] Piecewise((A*c**3*x + 3*A*c**2*d*log(tan(e + f*x)**2 + 1)/(2*f) - 3*A*c*d**
2*x + 3*A*c*d**2*tan(e + f*x)/f - A*d**3*log(tan(e + f*x)**2 + 1)/(2*f) + A
*d**3*tan(e + f*x)**2/(2*f) + B*c**3*log(tan(e + f*x)**2 + 1)/(2*f) - 3*B*c
**2*d*x + 3*B*c**2*d*tan(e + f*x)/f - 3*B*c*d**2*log(tan(e + f*x)**2 + 1)/(
2*f) + 3*B*c*d**2*tan(e + f*x)**2/(2*f) + B*d**3*x + B*d**3*tan(e + f*x)**3
/(3*f) - B*d**3*tan(e + f*x)/f - C*c**3*x + C*c**3*tan(e + f*x)/f - 3*C*c**
2*d*log(tan(e + f*x)**2 + 1)/(2*f) + 3*C*c**2*d*tan(e + f*x)**2/(2*f) + 3*C
*c*d**2*x + C*c*d**2*tan(e + f*x)**3/f - 3*C*c*d**2*tan(e + f*x)/f + C*d**3
*log(tan(e + f*x)**2 + 1)/(2*f) + C*d**3*tan(e + f*x)**4/(4*f) - C*d**3*tan
(e + f*x)**2/(2*f), Ne(f, 0)), (x*(c + d*tan(e))**3*(A + B*tan(e) + C*tan(e
)**2), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.32 (sec) , antiderivative size = 202, normalized size of antiderivative = 1.06

$$\int (c+d\tan(e+fx))^3 \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\frac{3\,Cd^3\tan\left(fx+e\right)^4+4\,(3\,Ccd^2+Bd^3)\tan\left(fx+e\right)^3+6\,(3\,Cc^2d+3\,Bcd^2+(A-C)d^3)\tan\left(fx+e\right)^2+1}{}$$

```
[In] integrate((c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorithm="
maxima")
```

```
[Out] 1/12*(3*C*d^3*tan(f*x + e)^4 + 4*(3*C*c*d^2 + B*d^3)*tan(f*x + e)^3 + 6*(3*
C*c^2*d + 3*B*c*d^2 + (A - C)*d^3)*tan(f*x + e)^2 + 12*((A - C)*c^3 - 3*B*c
^2*d - 3*(A - C)*c*d^2 + B*d^3)*(f*x + e) + 6*(B*c^3 + 3*(A - C)*c^2*d - 3*
B*c*d^2 - (A - C)*d^3)*log(tan(f*x + e)^2 + 1) + 12*(C*c^3 + 3*B*c^2*d + 3*
(A - C)*c*d^2 - B*d^3)*tan(f*x + e))/f
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 3720 vs. $2(185) = 370$.

Time = 3.19 (sec) , antiderivative size = 3720, normalized size of antiderivative = 19.48

$$\int (c+d\tan(e+fx))^3 \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\text{Too large to display}$$

```
[In] integrate((c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorithm="
giac")
```

```
[Out] 1/12*(12*A*c^3*f*x*tan(f*x)^4*tan(e)^4 - 12*C*c^3*f*x*tan(f*x)^4*tan(e)^4 -
 36*B*c^2*d*f*x*tan(f*x)^4*tan(e)^4 - 36*A*c*d^2*f*x*tan(f*x)^4*tan(e)^4 +
36*C*c*d^2*f*x*tan(f*x)^4*tan(e)^4 + 12*B*d^3*f*x*tan(f*x)^4*tan(e)^4 - 6*B
*c^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)
^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 - 18*A*c^2*d*log(4*(ta
n(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^
2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 18*C*c^2*d*log(4*(tan(f*x)^2*tan(e
)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 +
 1))*tan(f*x)^4*tan(e)^4 + 18*B*c*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*
x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^
4*tan(e)^4 + 6*A*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(t
an(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 - 6*C*
d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^
2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 - 48*A*c^3*f*x*tan(f*x)
```

^3*tan(e)^3 + 48*C*c^3*f*x*tan(f*x)^3*tan(e)^3 + 144*B*c^2*d*f*x*tan(f*x)^3
*tan(e)^3 + 144*A*c*d^2*f*x*tan(f*x)^3*tan(e)^3 - 144*C*c*d^2*f*x*tan(f*x)^
3*tan(e)^3 - 48*B*d^3*f*x*tan(f*x)^3*tan(e)^3 + 18*C*c^2*d*tan(f*x)^4*tan(e
)^4 + 18*B*c*d^2*tan(f*x)^4*tan(e)^4 + 6*A*d^3*tan(f*x)^4*tan(e)^4 - 9*C*d^
3*tan(f*x)^4*tan(e)^4 + 24*B*c^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*ta
n(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan
(e)^3 + 72*A*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan
(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 72*C*c
^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)
^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 72*B*c*d^2*log(4*(ta
n(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^
2 + tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 24*A*d^3*log(4*(tan(f*x)^2*tan(e)^
2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1
))*tan(f*x)^3*tan(e)^3 + 24*C*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*t
an(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^3*ta
n(e)^3 - 12*C*c^3*tan(f*x)^4*tan(e)^3 - 36*B*c^2*d*tan(f*x)^4*tan(e)^3 - 36
*A*c*d^2*tan(f*x)^4*tan(e)^3 + 36*C*c*d^2*tan(f*x)^4*tan(e)^3 + 12*B*d^3*ta
n(f*x)^4*tan(e)^3 - 12*C*c^3*tan(f*x)^3*tan(e)^4 - 36*B*c^2*d*tan(f*x)^3*ta
n(e)^4 - 36*A*c*d^2*tan(f*x)^3*tan(e)^4 + 36*C*c*d^2*tan(f*x)^3*tan(e)^4 +
12*B*d^3*tan(f*x)^3*tan(e)^4 + 72*A*c^3*f*x*tan(f*x)^2*tan(e)^2 - 72*C*c^3*
f*x*tan(f*x)^2*tan(e)^2 - 216*B*c^2*d*f*x*tan(f*x)^2*tan(e)^2 - 216*A*c*d^2
*f*x*tan(f*x)^2*tan(e)^2 + 216*C*c*d^2*f*x*tan(f*x)^2*tan(e)^2 + 72*B*d^3*f
*x*tan(f*x)^2*tan(e)^2 + 18*C*c^2*d*tan(f*x)^4*tan(e)^2 + 18*B*c*d^2*tan(f*
x)^4*tan(e)^2 + 6*A*d^3*tan(f*x)^4*tan(e)^2 - 6*C*d^3*tan(f*x)^4*tan(e)^2 -
36*C*c^2*d*tan(f*x)^3*tan(e)^3 - 36*B*c*d^2*tan(f*x)^3*tan(e)^3 - 12*A*d^3
*tan(f*x)^3*tan(e)^3 + 24*C*d^3*tan(f*x)^3*tan(e)^3 + 18*C*c^2*d*tan(f*x)^2
*tan(e)^4 + 18*B*c*d^2*tan(f*x)^2*tan(e)^4 + 6*A*d^3*tan(f*x)^2*tan(e)^4 -
6*C*d^3*tan(f*x)^2*tan(e)^4 - 12*C*c*d^2*tan(f*x)^4*tan(e) - 4*B*d^3*tan(f*
x)^4*tan(e) - 36*B*c^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/
(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 10
8*A*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*t
an(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 108*C*c^2*d*log
(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan
(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 108*B*c*d^2*log(4*(tan(f*x)^
2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan
(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 36*A*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*t
an(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(
f*x)^2*tan(e)^2 - 36*C*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) +
1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2
+ 36*C*c^3*tan(f*x)^3*tan(e)^2 + 108*B*c^2*d*tan(f*x)^3*tan(e)^2 + 108*A*c*
d^2*tan(f*x)^3*tan(e)^2 - 144*C*c*d^2*tan(f*x)^3*tan(e)^2 - 48*B*d^3*tan(f*
x)^3*tan(e)^2 + 36*C*c^3*tan(f*x)^2*tan(e)^3 + 108*B*c^2*d*tan(f*x)^2*tan(e
)^3 + 108*A*c*d^2*tan(f*x)^2*tan(e)^3 - 144*C*c*d^2*tan(f*x)^2*tan(e)^3 - 4
8*B*d^3*tan(f*x)^2*tan(e)^3 - 12*C*c*d^2*tan(f*x)*tan(e)^4 - 4*B*d^3*tan(f*
x)*tan(e)^4 + 3*C*d^3*tan(f*x)^4 - 48*A*c^3*f*x*tan(f*x)*tan(e) + 48*C*c^3*

f*x*tan(f*x)*tan(e) + 144*B*c^2*d*f*x*tan(f*x)*tan(e) + 144*A*c*d^2*f*x*tan(f*x)*tan(e) - 144*C*c*d^2*f*x*tan(f*x)*tan(e) - 48*B*d^3*f*x*tan(f*x)*tan(e) - 36*C*c^2*d*tan(f*x)^3*tan(e) - 36*B*c*d^2*tan(f*x)^3*tan(e) - 12*A*d^3*tan(f*x)^3*tan(e) + 24*C*d^3*tan(f*x)^3*tan(e) + 36*C*c^2*d*tan(f*x)^2*tan(e)^2 + 36*B*c*d^2*tan(f*x)^2*tan(e)^2 + 12*A*d^3*tan(f*x)^2*tan(e)^2 - 12*C*d^3*tan(f*x)^2*tan(e)^2 - 36*C*c^2*d*tan(f*x)*tan(e)^3 - 36*B*c*d^2*tan(f*x)*tan(e)^3 - 12*A*d^3*tan(f*x)*tan(e)^3 + 24*C*d^3*tan(f*x)*tan(e)^3 + 3*C*d^3*tan(e)^4 + 12*C*c*d^2*tan(f*x)^3 + 4*B*d^3*tan(f*x)^3 + 24*B*c^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 72*A*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) - 72*C*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) - 72*B*c*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) - 24*A*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) + 24*C*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1))*tan(f*x)*tan(e) - 36*C*c^3*tan(f*x)^2*tan(e) - 108*B*c^2*d*tan(f*x)^2*tan(e) - 108*A*c*d^2*tan(f*x)^2*tan(e) + 144*C*c*d^2*tan(f*x)^2*tan(e) + 48*B*d^3*tan(f*x)^2*tan(e) - 36*C*c^3*tan(f*x)*tan(e)^2 - 108*B*c^2*d*tan(f*x)*tan(e)^2 - 108*A*c*d^2*tan(f*x)*tan(e)^2 + 144*C*c*d^2*tan(f*x)*tan(e)^2 + 48*B*d^3*tan(f*x)*tan(e)^2 + 12*C*c*d^2*tan(e)^3 + 4*B*d^3*tan(e)^3 + 12*A*c^3*f*x - 12*C*c^3*f*x - 36*B*c^2*d*f*x - 36*A*c*d^2*f*x + 36*C*c*d^2*f*x + 12*B*d^3*f*x + 18*C*c^2*d*tan(f*x)^2 + 18*B*c*d^2*tan(f*x)^2 + 6*A*d^3*tan(f*x)^2 - 6*C*d^3*tan(f*x)^2 - 36*C*c^2*d*tan(f*x)*tan(e) - 36*B*c*d^2*tan(f*x)*tan(e) - 12*A*d^3*tan(f*x)*tan(e) + 24*C*d^3*tan(f*x)*tan(e) + 18*C*c^2*d*tan(e)^2 + 18*B*c*d^2*tan(e)^2 + 6*A*d^3*tan(e)^2 - 6*C*d^3*tan(e)^2 - 6*B*c^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 18*A*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) + 18*C*c^2*d*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) + 18*B*c*d^2*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) + 6*A*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) - 6*C*d^3*log(4*(tan(f*x)^2*tan(e)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(f*x)^2*tan(e)^2 + tan(f*x)^2 + tan(e)^2 + 1)) + 12*C*c^3*tan(f*x) + 36*B*c^2*d*tan(f*x) + 36*A*c*d^2*tan(f*x) - 36*C*c*d^2*tan(f*x) - 12*B*d^3*tan(f*x) + 12*C*c^3*tan(e) + 36*B*c^2*d*tan(e) + 36*A*c*d^2*tan(e) - 36*C*c*d^2*tan(e) - 12*B*d^3*tan(e) + 18*C*c^2*d + 18*B*c*d^2 + 6*A*d^3 - 9*C*d^3)/(f*tan(f*x)^4*tan(e)^4 - 4*f*tan(f*x)^3*tan(e)^3 + 6*f*tan(f*x)^2*tan(e)^2 - 4*f*tan(f*x)*tan(e) + f)

Mupad [B] (verification not implemented)

Time = 8.17 (sec) , antiderivative size = 221, normalized size of antiderivative = 1.16

$$\int (c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=x\left(A\,c^3+B\,d^3-C\,c^3-3\,A\,c\,d^2-3\,B\,c^2\,d+3\,C\,c\,d^2\right)+\frac{\tan(e+f\,x)\,\left(C\,c^3-B\,d^3+3\,A\,c\,d^2+3\,B\,c^2\,d-3\,C\,c\,d^2\right)}{f}+\frac{\tan(e+f\,x)^3\,\left(\frac{B\,d^3}{3}+C\,c\,d^2\right)}{f}-\frac{\ln\left(\tan(e+f\,x)^2+1\right)\,\left(\frac{A\,d^3}{2}-\frac{B\,c^3}{2}-\frac{C\,d^3}{2}-\frac{3\,A\,c^2\,d}{2}+\frac{3\,B\,c\,d^2}{2}+\frac{3\,C\,c^2\,d}{2}\right)}{f}+\frac{\tan(e+f\,x)^2\,\left(\frac{A\,d^3}{2}-\frac{C\,d^3}{2}+\frac{3\,B\,c\,d^2}{2}+\frac{3\,C\,c^2\,d}{2}\right)}{f}+\frac{C\,d^3\,\tan(e+f\,x)^4}{4\,f}$$

```
[In] int((c + d*tan(e + f*x))^3*(A + B*tan(e + f*x) + C*tan(e + f*x)^2),x)
```

```
[Out] x*(A*c^3 + B*d^3 - C*c^3 - 3*A*c*d^2 - 3*B*c^2*d + 3*C*c*d^2) + (tan(e + f*
x)*(C*c^3 - B*d^3 + 3*A*c*d^2 + 3*B*c^2*d - 3*C*c*d^2))/f + (tan(e + f*x)^3
*((B*d^3)/3 + C*c*d^2))/f - (log(tan(e + f*x)^2 + 1)*((A*d^3)/2 - (B*c^3)/2
 - (C*d^3)/2 - (3*A*c^2*d)/2 + (3*B*c*d^2)/2 + (3*C*c^2*d)/2))/f + (tan(e +
 f*x)^2*((A*d^3)/2 - (C*d^3)/2 + (3*B*c*d^2)/2 + (3*C*c^2*d)/2))/f + (C*d^3
*tan(e + f*x)^4)/(4*f)
```

3.67 $\int \frac{(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{a+b\tan(e+fx)}\,dx$

Optimal result 624
Rubi [A] (verified) 625
Mathematica [C] (verified) 628
Maple [A] (verified) 628
Fricas [A] (verification not implemented) 629
Sympy [C] (verification not implemented) 629
Maxima [A] (verification not implemented) 633
Giac [A] (verification not implemented) 634
Mupad [B] (verification not implemented) 635

Optimal result

Integrand size = 45, antiderivative size = 363

$$
\begin{aligned}
&\int \frac{(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{a+b\tan(e+fx)}\,dx = \\
&-\frac{(a(c^3C+3Bc^2d-3cCd^2-Bd^3-A(c^3-3cd^2))-b((A-C)d(3c^2-d^2)+B(c^3-3cd^2)))\,x}{a^2+b^2} \\
&-\frac{(b(c^3C+3Bc^2d-3cCd^2-Bd^3)+a(Bc^3-3c^2Cd-3Bcd^2+Cd^3)+A(ad(3c^2-d^2)-b(c^3-3cd^2)))}{(a^2+b^2)\,f} \\
&+\frac{(Ab^2-a(bB-aC))\,(bc-ad)^3\log(a+b\tan(e+fx))}{b^4\,(a^2+b^2)\,f} \\
&+\frac{d(b^2d(Bc+(A-C)d)+(bc-ad)(bcC+bBd-aCd))\tan(e+fx)}{b^3f} \\
&+\frac{(bcC+bBd-aCd)(c+d\tan(e+fx))^2}{2b^2f}+\frac{C(c+d\tan(e+fx))^3}{3bf}
\end{aligned}
$$

```
[Out] -(a*(c^3*C+3*B*c^2*d-3*C*c*d^2-B*d^3-A*(c^3-3*c*d^2))-b*((A-C)*d*(3*c^2-d^2
)+B*(c^3-3*c*d^2)))*x/(a^2+b^2)-(b*(3*B*c^2*d-B*d^3+C*c^3-3*C*c*d^2)+a*(B*c
^3-3*B*c*d^2-3*C*c^2*d+C*d^3)+A*(a*d*(3*c^2-d^2)-b*(c^3-3*c*d^2)))*ln(cos(f
*x+e))/(a^2+b^2)/f+(A*b^2-a*(B*b-C*a))*(-a*d+b*c)^3*ln(a+b*tan(f*x+e))/b^4/
(a^2+b^2)/f+d*(b^2*d*(B*c+(A-C)*d)+(-a*d+b*c)*(B*b*d-C*a*d+C*b*c))*tan(f*x+
e)/b^3/f+1/2*(B*b*d-C*a*d+C*b*c)*(c+d*tan(f*x+e))^2/b^2/f+1/3*C*(c+d*tan(f*
x+e))^3/b/f
```

Rubi [A] (verified)

Time = 1.66 (sec) , antiderivative size = 363, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.133, Rules used = {3728, 3718, 3707, 3698, 31, 3556}

$$\int \frac{(c+d\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx =$$

$$-\frac{\log(\cos(e+fx))\,(A(ad(3c^2-d^2)-b(c^3-3cd^2))+a(Bc^3-3Bcd^2-3c^2Cd+Cd^3)+b(3Bc^2d-Bd^3}{f\,(a^2+b^2)}$$

$$-\frac{x(a(-A(c^3-3cd^2)+3Bc^2d-Bd^3+c^3C-3cCd^2)-b(d(A-C)\,(3c^2-d^2)+B(c^3-3cd^2)))}{a^2+b^2}$$

$$+\frac{(bc-ad)^3\,(Ab^2-a(bB-aC))\log(a+b\tan(e+fx))}{b^4f\,(a^2+b^2)}$$

$$+\frac{d\tan(e+fx)\,((bc-ad)(-aCd+bBd+bcC)+b^2d(d(A-C)+Bc))}{b^3f}$$

$$+\frac{(-aCd+bBd+bcC)(c+d\tan(e+fx))^2}{2b^2f}+\frac{C(c+d\tan(e+fx))^3}{3bf}$$

```
[In] Int[((c + d*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a + b
*Tan[e + f*x]),x]
```

```
[Out] -(((a*(c^3*C + 3*B*c^2*d - 3*c*C*d^2 - B*d^3 - A*(c^3 - 3*c*d^2)) - b*((A -
 C)*d*(3*c^2 - d^2) + B*(c^3 - 3*c*d^2)))*x)/(a^2 + b^2)) - ((b*(c^3*C + 3*
B*c^2*d - 3*c*C*d^2 - B*d^3) + a*(B*c^3 - 3*c^2*C*d - 3*B*c*d^2 + C*d^3) +
A*(a*d*(3*c^2 - d^2) - b*(c^3 - 3*c*d^2)))*Log[Cos[e + f*x]])/((a^2 + b^2)*
f) + ((A*b^2 - a*(b*B - a*C))*(b*c - a*d)^3*Log[a + b*Tan[e + f*x]])/(b^4*(
a^2 + b^2)*f) + (d*(b^2*d*(B*c + (A - C)*d) + (b*c - a*d)*(b*c*C + b*B*d -
a*C*d))*Tan[e + f*x])/(b^3*f) + ((b*c*C + b*B*d - a*C*d)*(c + d*Tan[e + f*x
])^2)/(2*b^2*f) + (C*(c + d*Tan[e + f*x])^3)/(3*b*f)
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3698

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3707

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[(a*A + b*B - a
*C)*(x/(a^2 + b^2)), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\text{integral} = \frac{C(c+d\tan(e+fx))^3}{3bf} + \frac{\int \frac{(c+d\tan(e+fx))^2(3(Abc-aCd)+3b(Bc+(A-C)d)\tan(e+fx)+3(bcC+bBd-aCd)\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx}{3b}$$

$$= \frac{(bcC+bBd-aCd)(c+d\tan(e+fx))^2}{2b^2f} + \frac{C(c+d\tan(e+fx))^3}{3bf} + \frac{\int \frac{(c+d\tan(e+fx))(6(Ab^2c^2+ad(aCd-b(2cC+Bd)))+6b^2(2c(A-C)d+B(c^2-d^2))\tan(e+fx)+6(b^2d(Bc+(A-C)d)+(bc-ad)(bcC}{a+b\tan(e+fx)}}{6b^2}$$

$$= \frac{d(b^2d(Bc+(A-C)d)+(bc-ad)(bcC+bBd-aCd))\tan(e+fx)}{b^3f} + \frac{(bcC+bBd-aCd)(c+d\tan(e+fx))^2}{2b^2f} + \frac{C(c+d\tan(e+fx))^3}{3bf} - \frac{\int \frac{-6(Ab^2(bc^3-ad^3)-ad(a^2Cd^2-abd(3cC+Bd)+b^2(3c^2C+3Bcd-Cd^2)))-6b^3((A-C)d(3c^2-d^2)+B(c^3-3cd^2))\tan(e+fx)+6}{a+b\tan(e+fx)}}{6b^3}$$

$$= -\frac{(a(c^3C+3Bc^2d-3cCd^2-Bd^3-A(c^3-3cd^2))-b((A-C)d(3c^2-d^2)+B(c^3-3cd^2)))x}{a^2+b^2} + \frac{d(b^2d(Bc+(A-C)d)+(bc-ad)(bcC+bBd-aCd))\tan(e+fx)}{b^3f} + \frac{(bcC+bBd-aCd)(c+d\tan(e+fx))^2}{2b^2f} + \frac{C(c+d\tan(e+fx))^3}{3bf} + \frac{((Ab^2-a(bB-aC))(bc-ad)^3)\int \frac{1+\tan^2(e+fx)}{a+b\tan(e+fx)}\,dx}{b^3(a^2+b^2)} + \frac{(b(c^3C+3Bc^2d-3cCd^2-Bd^3)+a(Bc^3-3c^2Cd-3Bcd^2+Cd^3)+A(ad(3c^2-d^2)-b(c^3-}{a^2+b^2}$$

$$= -\frac{(a(c^3C+3Bc^2d-3cCd^2-Bd^3-A(c^3-3cd^2))-b((A-C)d(3c^2-d^2)+B(c^3-3cd^2)))x}{a^2+b^2} - \frac{(b(c^3C+3Bc^2d-3cCd^2-Bd^3)+a(Bc^3-3c^2Cd-3Bcd^2+Cd^3)+A(ad(3c^2-d^2)-b(c^3-}{(a^2+b^2)f} + \frac{d(b^2d(Bc+(A-C)d)+(bc-ad)(bcC+bBd-aCd))\tan(e+fx)}{b^3f} + \frac{(bcC+bBd-aCd)(c+d\tan(e+fx))^2}{2b^2f} + \frac{C(c+d\tan(e+fx))^3}{3bf} + \frac{((Ab^2-a(bB-aC))(bc-ad)^3)\,\mathrm{Subst}\left(\int \frac{1}{a+x}\,dx, x, b\tan(e+fx)\right)}{b^4(a^2+b^2)f}$$

$$= -\frac{(a(c^3C+3Bc^2d-3cCd^2-Bd^3-A(c^3-3cd^2))-b((A-C)d(3c^2-d^2)+B(c^3-3cd^2)))x}{a^2+b^2} - \frac{(b(c^3C+3Bc^2d-3cCd^2-Bd^3)+a(Bc^3-3c^2Cd-3Bcd^2+Cd^3)+A(ad(3c^2-d^2)-b(c^3-}{(a^2+b^2)f} + \frac{(Ab^2-a(bB-aC))(bc-ad)^3\log(a+b\tan(e+fx))}{b^4(a^2+b^2)f} + \frac{d(b^2d(Bc+(A-C)d)+(bc-ad)(bcC+bBd-aCd))\tan(e+fx)}{b^3f} + \frac{(bcC+bBd-aCd)(c+d\tan(e+fx))^2}{2b^2f} + \frac{C(c+d\tan(e+fx))^3}{3bf}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 4.89 (sec) , antiderivative size = 255, normalized size of antiderivative = 0.70

$$\int \frac{(c+d\tan(e+fx))^3 \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{a+b\tan(e+fx)}\,dx$$
$$= \frac{\frac{3b^2(-iA+B+iC)(c+id)^3\log(i-\tan(e+fx))}{a+ib} - \frac{3b^2(A-iB-C)(ic+d)^3\log(i+\tan(e+fx))}{a-ib} + \frac{6(Ab^2+a(-bB+aC))(bc-ad)^3\log(a+b\tan(e+fx)}{b^2(a^2+b^2)}}{}$$

```
[In] Integrate[((c + d*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(a + b*Tan[e + f*x]),x]
```

```
[Out] ((3*b^2*((-I)*A + B + I*C)*(c + I*d)^3*Log[I - Tan[e + f*x]])/(a + I*b) - (
3*b^2*(A - I*B - C)*(I*c + d)^3*Log[I + Tan[e + f*x]])/(a - I*b) + (6*(A*b^
2 + a*(-(b*B) + a*C))*(b*c - a*d)^3*Log[a + b*Tan[e + f*x]])/(b^2*(a^2 + b^
2)) + 6*b*d^2*(B*c + (A - C)*d)*Tan[e + f*x] + (6*d*(b*c - a*d)*(b*c*C + b*
B*d - a*C*d)*Tan[e + f*x])/b + 3*(b*c*C + b*B*d - a*C*d)*(c + d*Tan[e + f*x
])^2 + 2*b*C*(c + d*Tan[e + f*x])^3)/(6*b^2*f)
```

Maple [A] (verified)

Time = 0.25 (sec) , antiderivative size = 501, normalized size of antiderivative = 1.38

method	result
norman	$\frac{(A a c^3-3A a c d^2+3A b c^2 d-A b d^3-3B a c^2 d+B a d^3+B b c^3-3B b c d^2-C a c^3+3C a c d^2-3C b c^2 d+C b d^3)x}{a^2+b^2}+\frac{(A b^2 d^2-B}{}$
derivativedivides	$\frac{d\left(\frac{C b^2 d^2\tan(fx+e)^3}{3}+\frac{B b^2 d^2\tan(fx+e)^2}{2}-\frac{C a b d^2\tan(fx+e)^2}{2}+\frac{3C b^2 c d\tan(fx+e)^2}{2}+\tan(fx+e)A b^2 d^2-\tan(fx+e)B a b d^2+3\tan(\right.}{b^3}$
default	$\frac{d\left(\frac{C b^2 d^2\tan(fx+e)^3}{3}+\frac{B b^2 d^2\tan(fx+e)^2}{2}-\frac{C a b d^2\tan(fx+e)^2}{2}+\frac{3C b^2 c d\tan(fx+e)^2}{2}+\tan(fx+e)A b^2 d^2-\tan(fx+e)B a b d^2+3\tan(\right.}{b^3}$
parallelrisch	Expression too large to display
risch	Expression too large to display

```
[In] int((c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e)),x,m
ethod=_RETURNVERBOSE)
```

```
[Out] (A*a*c^3-3*A*a*c*d^2+3*A*b*c^2*d-A*b*d^3-3*B*a*c^2*d+B*a*d^3+B*b*c^3-3*B*b*
c*d^2-C*a*c^3+3*C*a*c*d^2-3*C*b*c^2*d+C*b*d^3)/(a^2+b^2)*x+(A*b^2*d^2-B*a*b
*d^2+3*B*b^2*c*d+C*a^2*d^2-3*C*a*b*c*d+3*C*b^2*c^2-C*b^2*d^2)*d/f/b^3*tan(f
*x+e)+1/3*C*d^3/b/f*tan(f*x+e)^3+1/2*d^2*(B*b*d-C*a*d+3*C*b*c)/b^2/f*tan(f*
x+e)^2+1/2*(3*A*a*c^2*d-A*a*d^3-A*b*c^3+3*A*b*c*d^2+B*a*c^3-3*B*a*c*d^2+3*B
```

```
*b*c^2*d-B*b*d^3-3*C*a*c^2*d+C*a*d^3+C*b*c^3-3*C*b*c*d^2)/f/(a^2+b^2)*ln(1+
tan(f*x+e)^2)-(A*a^3*b^2*d^3-3*A*a^2*b^3*c*d^2+3*A*a*b^4*c^2*d-A*b^5*c^3-B*
a^4*b*d^3+3*B*a^3*b^2*c*d^2-3*B*a^2*b^3*c^2*d+B*a*b^4*c^3+C*a^5*d^3-3*C*a^4
*b*c*d^2+3*C*a^3*b^2*c^2*d-C*a^2*b^3*c^3)/(a^2+b^2)/b^4/f*ln(a+b*tan(f*x+e)
)
```

Fricas [A] (verification not implemented)

none

Time = 0.79 (sec) , antiderivative size = 623, normalized size of antiderivative = 1.72

$$\int \frac{(c+d\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx$$
$$= \frac{2\,(Ca^2b^3+Cb^5)d^3\tan\left(fx+e\right)^3+6\,(((A-C)ab^4+Bb^5)c^3-3\,(Bab^4-(A-C)b^5)c^2d-3\,((A-C)ab^4}{}$$

```
[In] integrate((c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e
)),x, algorithm="fricas")
```

```
[Out] 1/6*(2*(C*a^2*b^3 + C*b^5)*d^3*tan(f*x + e)^3 + 6*(((A - C)*a*b^4 + B*b^5)*
c^3 - 3*(B*a*b^4 - (A - C)*b^5)*c^2*d - 3*((A - C)*a*b^4 + B*b^5)*c*d^2 + (
B*a*b^4 - (A - C)*b^5)*d^3)*f*x + 3*(3*(C*a^2*b^3 + C*b^5)*c*d^2 - (C*a^3*b
^2 - B*a^2*b^3 + C*a*b^4 - B*b^5)*d^3)*tan(f*x + e)^2 + 3*((C*a^2*b^3 - B*a
*b^4 + A*b^5)*c^3 - 3*(C*a^3*b^2 - B*a^2*b^3 + A*a*b^4)*c^2*d + 3*(C*a^4*b
- B*a^3*b^2 + A*a^2*b^3)*c*d^2 - (C*a^5 - B*a^4*b + A*a^3*b^2)*d^3)*log((b^
2*tan(f*x + e)^2 + 2*a*b*tan(f*x + e) + a^2)/(tan(f*x + e)^2 + 1)) - 3*((C*
a^2*b^3 + C*b^5)*c^3 - 3*(C*a^3*b^2 - B*a^2*b^3 + C*a*b^4 - B*b^5)*c^2*d +
3*(C*a^4*b - B*a^3*b^2 + A*a^2*b^3 - B*a*b^4 + (A - C)*b^5)*c*d^2 - (C*a^5
- B*a^4*b + A*a^3*b^2 + (A - C)*a*b^4 + B*b^5)*d^3)*log(1/(tan(f*x + e)^2 +
 1)) + 6*(3*(C*a^2*b^3 + C*b^5)*c^2*d - 3*(C*a^3*b^2 - B*a^2*b^3 + C*a*b^4
- B*b^5)*c*d^2 + (C*a^4*b - B*a^3*b^2 + A*a^2*b^3 - B*a*b^4 + (A - C)*b^5)*
d^3)*tan(f*x + e))/((a^2*b^4 + b^6)*f)
```

Sympy [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 20.20 (sec) , antiderivative size = 7096, normalized size of antiderivative = 19.55

$$\int \frac{(c+d\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx = \text{Too large to display}$$

```
[In] integrate((c+d*tan(f*x+e))**3*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x
+e)),x)
```

[Out] Piecewise((zoo*x*(c + d*tan(e))**3*(A + B*tan(e) + C*tan(e)**2)/tan(e), Eq(
a, 0) & Eq(b, 0) & Eq(f, 0)), ((A*c**3*x + 3*A*c**2*d*log(tan(e + f*x)**2 +
 1)/(2*f) - 3*A*c*d**2*x + 3*A*c*d**2*tan(e + f*x)/f - A*d**3*log(tan(e + f
*x)**2 + 1)/(2*f) + A*d**3*tan(e + f*x)**2/(2*f) + B*c**3*log(tan(e + f*x)*
*2 + 1)/(2*f) - 3*B*c**2*d*x + 3*B*c**2*d*tan(e + f*x)/f - 3*B*c*d**2*log(t
an(e + f*x)**2 + 1)/(2*f) + 3*B*c*d**2*tan(e + f*x)**2/(2*f) + B*d**3*x + B
*d**3*tan(e + f*x)**3/(3*f) - B*d**3*tan(e + f*x)/f - C*c**3*x + C*c**3*tan
(e + f*x)/f - 3*C*c**2*d*log(tan(e + f*x)**2 + 1)/(2*f) + 3*C*c**2*d*tan(e
+ f*x)**2/(2*f) + 3*C*c*d**2*x + C*c*d**2*tan(e + f*x)**3/f - 3*C*c*d**2*ta
n(e + f*x)/f + C*d**3*log(tan(e + f*x)**2 + 1)/(2*f) + C*d**3*tan(e + f*x)*
*4/(4*f) - C*d**3*tan(e + f*x)**2/(2*f))/a, Eq(b, 0)), (3*I*A*c**3*f*x*tan(
e + f*x)/(6*b*f*tan(e + f*x) - 6*I*b*f) + 3*A*c**3*f*x/(6*b*f*tan(e + f*x)
- 6*I*b*f) + 3*I*A*c**3/(6*b*f*tan(e + f*x) - 6*I*b*f) + 9*A*c**2*d*f*x*tan
(e + f*x)/(6*b*f*tan(e + f*x) - 6*I*b*f) - 9*I*A*c**2*d*f*x/(6*b*f*tan(e +
f*x) - 6*I*b*f) - 9*A*c**2*d/(6*b*f*tan(e + f*x) - 6*I*b*f) + 9*I*A*c*d**2*
f*x*tan(e + f*x)/(6*b*f*tan(e + f*x) - 6*I*b*f) + 9*A*c*d**2*f*x/(6*b*f*tan
(e + f*x) - 6*I*b*f) + 9*A*c*d**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(6*
b*f*tan(e + f*x) - 6*I*b*f) - 9*I*A*c*d**2*log(tan(e + f*x)**2 + 1)/(6*b*f*
tan(e + f*x) - 6*I*b*f) - 9*I*A*c*d**2/(6*b*f*tan(e + f*x) - 6*I*b*f) - 9*A
*d**3*f*x*tan(e + f*x)/(6*b*f*tan(e + f*x) - 6*I*b*f) + 9*I*A*d**3*f*x/(6*b
*f*tan(e + f*x) - 6*I*b*f) + 3*I*A*d**3*log(tan(e + f*x)**2 + 1)*tan(e + f*
x)/(6*b*f*tan(e + f*x) - 6*I*b*f) + 3*A*d**3*log(tan(e + f*x)**2 + 1)/(6*b*
f*tan(e + f*x) - 6*I*b*f) + 6*A*d**3*tan(e + f*x)**2/(6*b*f*tan(e + f*x) -
6*I*b*f) + 9*A*d**3/(6*b*f*tan(e + f*x) - 6*I*b*f) + 3*B*c**3*f*x*tan(e + f
*x)/(6*b*f*tan(e + f*x) - 6*I*b*f) - 3*I*B*c**3*f*x/(6*b*f*tan(e + f*x) - 6
*I*b*f) - 3*B*c**3/(6*b*f*tan(e + f*x) - 6*I*b*f) + 9*I*B*c**2*d*f*x*tan(e
+ f*x)/(6*b*f*tan(e + f*x) - 6*I*b*f) + 9*B*c**2*d*f*x/(6*b*f*tan(e + f*x)
- 6*I*b*f) + 9*B*c**2*d*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(6*b*f*tan(e
+ f*x) - 6*I*b*f) - 9*I*B*c**2*d*log(tan(e + f*x)**2 + 1)/(6*b*f*tan(e + f*
x) - 6*I*b*f) - 9*I*B*c**2*d/(6*b*f*tan(e + f*x) - 6*I*b*f) - 27*B*c*d**2*f
*x*tan(e + f*x)/(6*b*f*tan(e + f*x) - 6*I*b*f) + 27*I*B*c*d**2*f*x/(6*b*f*t
an(e + f*x) - 6*I*b*f) + 9*I*B*c*d**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)
/(6*b*f*tan(e + f*x) - 6*I*b*f) + 9*B*c*d**2*log(tan(e + f*x)**2 + 1)/(6*b*
f*tan(e + f*x) - 6*I*b*f) + 18*B*c*d**2*tan(e + f*x)**2/(6*b*f*tan(e + f*x)
 - 6*I*b*f) + 27*B*c*d**2/(6*b*f*tan(e + f*x) - 6*I*b*f) - 9*I*B*d**3*f*x*t
an(e + f*x)/(6*b*f*tan(e + f*x) - 6*I*b*f) - 9*B*d**3*f*x/(6*b*f*tan(e + f*
x) - 6*I*b*f) - 6*B*d**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(6*b*f*tan(e
 + f*x) - 6*I*b*f) + 6*I*B*d**3*log(tan(e + f*x)**2 + 1)/(6*b*f*tan(e + f*x
) - 6*I*b*f) + 3*B*d**3*tan(e + f*x)**3/(6*b*f*tan(e + f*x) - 6*I*b*f) + 3*
I*B*d**3*tan(e + f*x)**2/(6*b*f*tan(e + f*x) - 6*I*b*f) + 9*I*B*d**3/(6*b*f
*tan(e + f*x) - 6*I*b*f) + 3*I*C*c**3*f*x*tan(e + f*x)/(6*b*f*tan(e + f*x)
- 6*I*b*f) + 3*C*c**3*f*x/(6*b*f*tan(e + f*x) - 6*I*b*f) + 3*C*c**3*log(tan
(e + f*x)**2 + 1)*tan(e + f*x)/(6*b*f*tan(e + f*x) - 6*I*b*f) - 3*I*C*c**3*
log(tan(e + f*x)**2 + 1)/(6*b*f*tan(e + f*x) - 6*I*b*f) - 3*I*C*c**3/(6*b*f
*tan(e + f*x) - 6*I*b*f) - 27*C*c**2*d*f*x*tan(e + f*x)/(6*b*f*tan(e + f*x)

- 6*I*b*f) + 27*I*C*c**2*d*f*x/(6*b*f*tan(e + f*x) - 6*I*b*f) + 9*I*C*c**2
*d*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(6*b*f*tan(e + f*x) - 6*I*b*f) + 9
*C*c**2*d*log(tan(e + f*x)**2 + 1)/(6*b*f*tan(e + f*x) - 6*I*b*f) + 18*C*c*
*2*d*tan(e + f*x)**2/(6*b*f*tan(e + f*x) - 6*I*b*f) + 27*C*c**2*d/(6*b*f*ta
n(e + f*x) - 6*I*b*f) - 27*I*C*c*d**2*f*x*tan(e + f*x)/(6*b*f*tan(e + f*x)
- 6*I*b*f) - 27*C*c*d**2*f*x/(6*b*f*tan(e + f*x) - 6*I*b*f) - 18*C*c*d**2*l
og(tan(e + f*x)**2 + 1)*tan(e + f*x)/(6*b*f*tan(e + f*x) - 6*I*b*f) + 18*I*
C*c*d**2*log(tan(e + f*x)**2 + 1)/(6*b*f*tan(e + f*x) - 6*I*b*f) + 9*C*c*d*
*2*tan(e + f*x)**3/(6*b*f*tan(e + f*x) - 6*I*b*f) + 9*I*C*c*d**2*tan(e + f*
x)**2/(6*b*f*tan(e + f*x) - 6*I*b*f) + 27*I*C*c*d**2/(6*b*f*tan(e + f*x) -
6*I*b*f) + 15*C*d**3*f*x*tan(e + f*x)/(6*b*f*tan(e + f*x) - 6*I*b*f) - 15*I
*C*d**3*f*x/(6*b*f*tan(e + f*x) - 6*I*b*f) - 6*I*C*d**3*log(tan(e + f*x)**2
 + 1)*tan(e + f*x)/(6*b*f*tan(e + f*x) - 6*I*b*f) - 6*C*d**3*log(tan(e + f*
x)**2 + 1)/(6*b*f*tan(e + f*x) - 6*I*b*f) + 2*C*d**3*tan(e + f*x)**4/(6*b*f
*tan(e + f*x) - 6*I*b*f) + I*C*d**3*tan(e + f*x)**3/(6*b*f*tan(e + f*x) - 6
*I*b*f) - 9*C*d**3*tan(e + f*x)**2/(6*b*f*tan(e + f*x) - 6*I*b*f) - 15*C*d*
*3/(6*b*f*tan(e + f*x) - 6*I*b*f), Eq(a, -I*b)), (-3*I*A*c**3*f*x*tan(e + f
*x)/(6*b*f*tan(e + f*x) + 6*I*b*f) + 3*A*c**3*f*x/(6*b*f*tan(e + f*x) + 6*I
*b*f) - 3*I*A*c**3/(6*b*f*tan(e + f*x) + 6*I*b*f) + 9*A*c**2*d*f*x*tan(e +
f*x)/(6*b*f*tan(e + f*x) + 6*I*b*f) + 9*I*A*c**2*d*f*x/(6*b*f*tan(e + f*x)
+ 6*I*b*f) - 9*A*c**2*d/(6*b*f*tan(e + f*x) + 6*I*b*f) - 9*I*A*c*d**2*f*x*t
an(e + f*x)/(6*b*f*tan(e + f*x) + 6*I*b*f) + 9*A*c*d**2*f*x/(6*b*f*tan(e +
f*x) + 6*I*b*f) + 9*A*c*d**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(6*b*f*t
an(e + f*x) + 6*I*b*f) + 9*I*A*c*d**2*log(tan(e + f*x)**2 + 1)/(6*b*f*tan(e
 + f*x) + 6*I*b*f) + 9*I*A*c*d**2/(6*b*f*tan(e + f*x) + 6*I*b*f) - 9*A*d**3
*f*x*tan(e + f*x)/(6*b*f*tan(e + f*x) + 6*I*b*f) - 9*I*A*d**3*f*x/(6*b*f*ta
n(e + f*x) + 6*I*b*f) - 3*I*A*d**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(6
*b*f*tan(e + f*x) + 6*I*b*f) + 3*A*d**3*log(tan(e + f*x)**2 + 1)/(6*b*f*tan
(e + f*x) + 6*I*b*f) + 6*A*d**3*tan(e + f*x)**2/(6*b*f*tan(e + f*x) + 6*I*b
*f) + 9*A*d**3/(6*b*f*tan(e + f*x) + 6*I*b*f) + 3*B*c**3*f*x*tan(e + f*x)/(
6*b*f*tan(e + f*x) + 6*I*b*f) + 3*I*B*c**3*f*x/(6*b*f*tan(e + f*x) + 6*I*b*
f) - 3*B*c**3/(6*b*f*tan(e + f*x) + 6*I*b*f) - 9*I*B*c**2*d*f*x*tan(e + f*x
)/(6*b*f*tan(e + f*x) + 6*I*b*f) + 9*B*c**2*d*f*x/(6*b*f*tan(e + f*x) + 6*I
*b*f) + 9*B*c**2*d*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(6*b*f*tan(e + f*x
) + 6*I*b*f) + 9*I*B*c**2*d*log(tan(e + f*x)**2 + 1)/(6*b*f*tan(e + f*x) +
6*I*b*f) + 9*I*B*c**2*d/(6*b*f*tan(e + f*x) + 6*I*b*f) - 27*B*c*d**2*f*x*ta
n(e + f*x)/(6*b*f*tan(e + f*x) + 6*I*b*f) - 27*I*B*c*d**2*f*x/(6*b*f*tan(e
+ f*x) + 6*I*b*f) - 9*I*B*c*d**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(6*b
*f*tan(e + f*x) + 6*I*b*f) + 9*B*c*d**2*log(tan(e + f*x)**2 + 1)/(6*b*f*tan
(e + f*x) + 6*I*b*f) + 18*B*c*d**2*tan(e + f*x)**2/(6*b*f*tan(e + f*x) + 6*
I*b*f) + 27*B*c*d**2/(6*b*f*tan(e + f*x) + 6*I*b*f) + 9*I*B*d**3*f*x*tan(e
+ f*x)/(6*b*f*tan(e + f*x) + 6*I*b*f) - 9*B*d**3*f*x/(6*b*f*tan(e + f*x) +
6*I*b*f) - 6*B*d**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(6*b*f*tan(e + f*
x) + 6*I*b*f) - 6*I*B*d**3*log(tan(e + f*x)**2 + 1)/(6*b*f*tan(e + f*x) + 6
*I*b*f) + 3*B*d**3*tan(e + f*x)**3/(6*b*f*tan(e + f*x) + 6*I*b*f) - 3*I*B*d

```
**3*tan(e + f*x)**2/(6*b*f*tan(e + f*x) + 6*I*b*f) - 9*I*B*d**3/(6*b*f*tan(
e + f*x) + 6*I*b*f) - 3*I*C*c**3*f*x*tan(e + f*x)/(6*b*f*tan(e + f*x) + 6*I
*b*f) + 3*C*c**3*f*x/(6*b*f*tan(e + f*x) + 6*I*b*f) + 3*C*c**3*log(tan(e +
f*x)**2 + 1)*tan(e + f*x)/(6*b*f*tan(e + f*x) + 6*I*b*f) + 3*I*C*c**3*log(t
an(e + f*x)**2 + 1)/(6*b*f*tan(e + f*x) + 6*I*b*f) + 3*I*C*c**3/(6*b*f*tan(
e + f*x) + 6*I*b*f) - 27*C*c**2*d*f*x*tan(e + f*x)/(6*b*f*tan(e + f*x) + 6*
I*b*f) - 27*I*C*c**2*d*f*x/(6*b*f*tan(e + f*x) + 6*I*b*f) - 9*I*C*c**2*d*lo
g(tan(e + f*x)**2 + 1)*tan(e + f*x)/(6*b*f*tan(e + f*x) + 6*I*b*f) + 9*C*c*
*2*d*log(tan(e + f*x)**2 + 1)/(6*b*f*tan(e + f*x) + 6*I*b*f) + 18*C*c**2*d*
tan(e + f*x)**2/(6*b*f*tan(e + f*x) + 6*I*b*f) + 27*C*c**2*d/(6*b*f*tan(e +
 f*x) + 6*I*b*f) + 27*I*C*c*d**2*f*x*tan(e + f*x)/(6*b*f*tan(e + f*x) + 6*I
*b*f) - 27*C*c*d**2*f*x/(6*b*f*tan(e + f*x) + 6*I*b*f) - 18*C*c*d**2*log(ta
n(e + f*x)**2 + 1)*tan(e + f*x)/(6*b*f*tan(e + f*x) + 6*I*b*f) - 18*I*C*c*d
**2*log(tan(e + f*x)**2 + 1)/(6*b*f*tan(e + f*x) + 6*I*b*f) + 9*C*c*d**2*ta
n(e + f*x)**3/(6*b*f*tan(e + f*x) + 6*I*b*f) - 9*I*C*c*d**2*tan(e + f*x)**2
/(6*b*f*tan(e + f*x) + 6*I*b*f) - 27*I*C*c*d**2/(6*b*f*tan(e + f*x) + 6*I*b
*f) + 15*C*d**3*f*x*tan(e + f*x)/(6*b*f*tan(e + f*x) + 6*I*b*f) + 15*I*C*d*
*3*f*x/(6*b*f*tan(e + f*x) + 6*I*b*f) + 6*I*C*d**3*log(tan(e + f*x)**2 + 1)
*tan(e + f*x)/(6*b*f*tan(e + f*x) + 6*I*b*f) - 6*C*d**3*log(tan(e + f*x)**2
 + 1)/(6*b*f*tan(e + f*x) + 6*I*b*f) + 2*C*d**3*tan(e + f*x)**4/(6*b*f*tan(
e + f*x) + 6*I*b*f) - I*C*d**3*tan(e + f*x)**3/(6*b*f*tan(e + f*x) + 6*I*b*
f) - 9*C*d**3*tan(e + f*x)**2/(6*b*f*tan(e + f*x) + 6*I*b*f) - 15*C*d**3/(6
*b*f*tan(e + f*x) + 6*I*b*f), Eq(a, I*b)), (x*(c + d*tan(e))**3*(A + B*tan(
e) + C*tan(e)**2)/(a + b*tan(e)), Eq(f, 0)), (-6*A*a**3*b**2*d**3*log(a/b +
 tan(e + f*x))/(6*a**2*b**4*f + 6*b**6*f) + 18*A*a**2*b**3*c*d**2*log(a/b +
 tan(e + f*x))/(6*a**2*b**4*f + 6*b**6*f) + 6*A*a**2*b**3*d**3*tan(e + f*x)
/(6*a**2*b**4*f + 6*b**6*f) + 6*A*a*b**4*c**3*f*x/(6*a**2*b**4*f + 6*b**6*f
) - 18*A*a*b**4*c**2*d*log(a/b + tan(e + f*x))/(6*a**2*b**4*f + 6*b**6*f) +
 9*A*a*b**4*c**2*d*log(tan(e + f*x)**2 + 1)/(6*a**2*b**4*f + 6*b**6*f) - 18
*A*a*b**4*c*d**2*f*x/(6*a**2*b**4*f + 6*b**6*f) - 3*A*a*b**4*d**3*log(tan(e
 + f*x)**2 + 1)/(6*a**2*b**4*f + 6*b**6*f) + 6*A*b**5*c**3*log(a/b + tan(e
+ f*x))/(6*a**2*b**4*f + 6*b**6*f) - 3*A*b**5*c**3*log(tan(e + f*x)**2 + 1)
/(6*a**2*b**4*f + 6*b**6*f) + 18*A*b**5*c**2*d*f*x/(6*a**2*b**4*f + 6*b**6*
f) + 9*A*b**5*c*d**2*log(tan(e + f*x)**2 + 1)/(6*a**2*b**4*f + 6*b**6*f) -
6*A*b**5*d**3*f*x/(6*a**2*b**4*f + 6*b**6*f) + 6*A*b**5*d**3*tan(e + f*x)/(
6*a**2*b**4*f + 6*b**6*f) + 6*B*a**4*b*d**3*log(a/b + tan(e + f*x))/(6*a**2
*b**4*f + 6*b**6*f) - 18*B*a**3*b**2*c*d**2*log(a/b + tan(e + f*x))/(6*a**2
*b**4*f + 6*b**6*f) - 6*B*a**3*b**2*d**3*tan(e + f*x)/(6*a**2*b**4*f + 6*b*
*6*f) + 18*B*a**2*b**3*c**2*d*log(a/b + tan(e + f*x))/(6*a**2*b**4*f + 6*b*
*6*f) + 18*B*a**2*b**3*c*d**2*tan(e + f*x)/(6*a**2*b**4*f + 6*b**6*f) + 3*B
*a**2*b**3*d**3*tan(e + f*x)**2/(6*a**2*b**4*f + 6*b**6*f) - 6*B*a*b**4*c**
3*log(a/b + tan(e + f*x))/(6*a**2*b**4*f + 6*b**6*f) + 3*B*a*b**4*c**3*log(
tan(e + f*x)**2 + 1)/(6*a**2*b**4*f + 6*b**6*f) - 18*B*a*b**4*c**2*d*f*x/(6
*a**2*b**4*f + 6*b**6*f) - 9*B*a*b**4*c*d**2*log(tan(e + f*x)**2 + 1)/(6*a*
*2*b**4*f + 6*b**6*f) + 6*B*a*b**4*d**3*f*x/(6*a**2*b**4*f + 6*b**6*f) - 6*
```

```
B*a*b**4*d**3*tan(e + f*x)/(6*a**2*b**4*f + 6*b**6*f) + 6*B*b**5*c**3*f*x/(
6*a**2*b**4*f + 6*b**6*f) + 9*B*b**5*c**2*d*log(tan(e + f*x)**2 + 1)/(6*a**
2*b**4*f + 6*b**6*f) - 18*B*b**5*c*d**2*f*x/(6*a**2*b**4*f + 6*b**6*f) + 18
*B*b**5*c*d**2*tan(e + f*x)/(6*a**2*b**4*f + 6*b**6*f) - 3*B*b**5*d**3*log(
tan(e + f*x)**2 + 1)/(6*a**2*b**4*f + 6*b**6*f) + 3*B*b**5*d**3*tan(e + f*x
)**2/(6*a**2*b**4*f + 6*b**6*f) - 6*C*a**5*d**3*log(a/b + tan(e + f*x))/(6*
a**2*b**4*f + 6*b**6*f) + 18*C*a**4*b*c*d**2*log(a/b + tan(e + f*x))/(6*a**
2*b**4*f + 6*b**6*f) + 6*C*a**4*b*d**3*tan(e + f*x)/(6*a**2*b**4*f + 6*b**6
*f) - 18*C*a**3*b**2*c**2*d*log(a/b + tan(e + f*x))/(6*a**2*b**4*f + 6*b**6
*f) - 18*C*a**3*b**2*c*d**2*tan(e + f*x)/(6*a**2*b**4*f + 6*b**6*f) - 3*C*a
**3*b**2*d**3*tan(e + f*x)**2/(6*a**2*b**4*f + 6*b**6*f) + 6*C*a**2*b**3*c*
*3*log(a/b + tan(e + f*x))/(6*a**2*b**4*f + 6*b**6*f) + 18*C*a**2*b**3*c**2
*d*tan(e + f*x)/(6*a**2*b**4*f + 6*b**6*f) + 9*C*a**2*b**3*c*d**2*tan(e + f
*x)**2/(6*a**2*b**4*f + 6*b**6*f) + 2*C*a**2*b**3*d**3*tan(e + f*x)**3/(6*a
**2*b**4*f + 6*b**6*f) - 6*C*a*b**4*c**3*f*x/(6*a**2*b**4*f + 6*b**6*f) - 9
*C*a*b**4*c**2*d*log(tan(e + f*x)**2 + 1)/(6*a**2*b**4*f + 6*b**6*f) + 18*C
*a*b**4*c*d**2*f*x/(6*a**2*b**4*f + 6*b**6*f) - 18*C*a*b**4*c*d**2*tan(e +
f*x)/(6*a**2*b**4*f + 6*b**6*f) + 3*C*a*b**4*d**3*log(tan(e + f*x)**2 + 1)/
(6*a**2*b**4*f + 6*b**6*f) - 3*C*a*b**4*d**3*tan(e + f*x)**2/(6*a**2*b**4*f
 + 6*b**6*f) + 3*C*b**5*c**3*log(tan(e + f*x)**2 + 1)/(6*a**2*b**4*f + 6*b*
*6*f) - 18*C*b**5*c**2*d*f*x/(6*a**2*b**4*f + 6*b**6*f) + 18*C*b**5*c**2*d*
tan(e + f*x)/(6*a**2*b**4*f + 6*b**6*f) - 9*C*b**5*c*d**2*log(tan(e + f*x)*
*2 + 1)/(6*a**2*b**4*f + 6*b**6*f) + 9*C*b**5*c*d**2*tan(e + f*x)**2/(6*a**
2*b**4*f + 6*b**6*f) + 6*C*b**5*d**3*f*x/(6*a**2*b**4*f + 6*b**6*f) + 2*C*b
**5*d**3*tan(e + f*x)**3/(6*a**2*b**4*f + 6*b**6*f) - 6*C*b**5*d**3*tan(e +
 f*x)/(6*a**2*b**4*f + 6*b**6*f), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.35 (sec) , antiderivative size = 436, normalized size of antiderivative = 1.20

$$\int \frac{(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{a+b\tan(e+fx)}\,dx$$

$$= \frac{\frac{6\left(\left(\left(A-C\right)a+Bb\right)c^3-3\left(Ba-\left(A-C\right)b\right)c^2d-3\left(\left(A-C\right)a+Bb\right)cd^2+\left(Ba-\left(A-C\right)b\right)d^3\right)\left(fx+e\right)}{a^2+b^2}+\frac{6\left(\left(Ca^2b^3-Bab^4+Ab^5\right)c^3-3\left(Ca^3b^2-Ba^2b^3+A\right.\right.}{}}{}$$

```
[In] integrate((c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e
)),x, algorithm="maxima")
```

```
[Out] 1/6*(6*(((A - C)*a + B*b)*c^3 - 3*(B*a - (A - C)*b)*c^2*d - 3*((A - C)*a +
B*b)*c*d^2 + (B*a - (A - C)*b)*d^3)*(f*x + e)/(a^2 + b^2) + 6*((C*a^2*b^3 -
 B*a*b^4 + A*b^5)*c^3 - 3*(C*a^3*b^2 - B*a^2*b^3 + A*a*b^4)*c^2*d + 3*(C*a^
4*b - B*a^3*b^2 + A*a^2*b^3)*c*d^2 - (C*a^5 - B*a^4*b + A*a^3*b^2)*d^3)*log
```

```
(b*tan(f*x + e) + a)/(a^2*b^4 + b^6) + 3*((B*a - (A - C)*b)*c^3 + 3*((A - C
)*a + B*b)*c^2*d - 3*(B*a - (A - C)*b)*c*d^2 - ((A - C)*a + B*b)*d^3)*log(t
an(f*x + e)^2 + 1)/(a^2 + b^2) + (2*C*b^2*d^3*tan(f*x + e)^3 + 3*(3*C*b^2*c
*d^2 - (C*a*b - B*b^2)*d^3)*tan(f*x + e)^2 + 6*(3*C*b^2*c^2*d - 3*(C*a*b -
B*b^2)*c*d^2 + (C*a^2 - B*a*b + (A - C)*b^2)*d^3)*tan(f*x + e))/b^3)/f
```

Giac [A] (verification not implemented)

none

Time = 0.97 (sec) , antiderivative size = 559, normalized size of antiderivative = 1.54

$$\int \frac{(c+d\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx$$
$$= \frac{\frac{6\,(Aac^3-Cac^3+Bbc^3-3\,Bac^2d+3\,Abc^2d-3\,Cbc^2d-3\,Aacd^2+3\,Cacd^2-3\,Bbcd^2+Bad^3-Abd^3+Cbd^3)(fx+e)}{a^2+b^2}+\frac{3\,(Bac^3-Abc^3+Cbc^3+3\,Aac^2d}{}}{}$$

```
[In] integrate((c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e
)),x, algorithm="giac")
```

```
[Out] 1/6*(6*(A*a*c^3 - C*a*c^3 + B*b*c^3 - 3*B*a*c^2*d + 3*A*b*c^2*d - 3*C*b*c^2
*d - 3*A*a*c*d^2 + 3*C*a*c*d^2 - 3*B*b*c*d^2 + B*a*d^3 - A*b*d^3 + C*b*d^3)
*(f*x + e)/(a^2 + b^2) + 3*(B*a*c^3 - A*b*c^3 + C*b*c^3 + 3*A*a*c^2*d - 3*C
*a*c^2*d + 3*B*b*c^2*d - 3*B*a*c*d^2 + 3*A*b*c*d^2 - 3*C*b*c*d^2 - A*a*d^3
+ C*a*d^3 - B*b*d^3)*log(tan(f*x + e)^2 + 1)/(a^2 + b^2) + 6*(C*a^2*b^3*c^3
 - B*a*b^4*c^3 + A*b^5*c^3 - 3*C*a^3*b^2*c^2*d + 3*B*a^2*b^3*c^2*d - 3*A*a*
b^4*c^2*d + 3*C*a^4*b*c*d^2 - 3*B*a^3*b^2*c*d^2 + 3*A*a^2*b^3*c*d^2 - C*a^5
*d^3 + B*a^4*b*d^3 - A*a^3*b^2*d^3)*log(abs(b*tan(f*x + e) + a))/(a^2*b^4 +
 b^6) + (2*C*b^2*d^3*tan(f*x + e)^3 + 9*C*b^2*c*d^2*tan(f*x + e)^2 - 3*C*a*
b*d^3*tan(f*x + e)^2 + 3*B*b^2*d^3*tan(f*x + e)^2 + 18*C*b^2*c^2*d*tan(f*x
+ e) - 18*C*a*b*c*d^2*tan(f*x + e) + 18*B*b^2*c*d^2*tan(f*x + e) + 6*C*a^2*
d^3*tan(f*x + e) - 6*B*a*b*d^3*tan(f*x + e) + 6*A*b^2*d^3*tan(f*x + e) - 6*
C*b^2*d^3*tan(f*x + e))/b^3)/f
```

Mupad [B] (verification not implemented)

Time = 12.19 (sec) , antiderivative size = 508, normalized size of antiderivative = 1.40

$$\int \frac{(c+d\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx$$
$$=\frac{\tan(e+f\,x)^2\left(\frac{B\,d^3+3\,C\,c\,d^2}{2\,b}-\frac{C\,a\,d^3}{2\,b^2}\right)}{f}$$
$$-\frac{\tan(e+f\,x)\left(\frac{a\left(\frac{B\,d^3+3\,C\,c\,d^2}{b}-\frac{C\,a\,d^3}{b^2}\right)}{b}-\frac{3\,C\,c^2\,d+3\,B\,c\,d^2+A\,d^3}{b}+\frac{C\,d^3}{b}\right)}{f}$$
$$-\frac{\ln\left(a+b\tan(e+f\,x)\right)\,\left(b^4\left(B\,a\,c^3+3\,A\,a\,d\,c^2\right)-b^3\left(C\,a^2\,c^3+3\,B\,a^2\,c^2\,d+3\,A\,a^2\,c\,d^2\right)+b^2\left(3\,C\,a^3\,c^2\,d\right.\right.}{f\,\left(a^2\,b^4+b^6\right)}$$
$$-\frac{\ln\left(\tan(e+f\,x)+1\mathrm{i}\right)\,\left(A\,c^3+A\,d^3\,1\mathrm{i}-B\,c^3\,1\mathrm{i}+B\,d^3-C\,c^3-C\,d^3\,1\mathrm{i}-3\,A\,c\,d^2-A\,c^2\,d\,3\mathrm{i}+B\,c\,d^2\,3\mathrm{i}\right.}{2\,f\,\left(b+a\,1\mathrm{i}\right)}$$
$$-\frac{\ln\left(\tan(e+f\,x)-\mathrm{i}\right)\,\left(A\,d^3-B\,c^3-C\,d^3-3\,A\,c^2\,d+3\,B\,c\,d^2+3\,C\,c^2\,d+A\,c^3\,1\mathrm{i}+B\,d^3\,1\mathrm{i}-C\,c^3\,1\mathrm{i}-\right.}{2\,f\,\left(a+b\,1\mathrm{i}\right)}$$
$$+\frac{C\,d^3\tan(e+f\,x)^3}{3\,b\,f}$$

```
[In] int(((c + d*tan(e + f*x))^3*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a + b
*tan(e + f*x)),x)
```

```
[Out] (tan(e + f*x)^2*((B*d^3 + 3*C*c*d^2)/(2*b) - (C*a*d^3)/(2*b^2)))/f - (tan(e
 + f*x)*((a*((B*d^3 + 3*C*c*d^2)/b - (C*a*d^3)/b^2))/b - (A*d^3 + 3*B*c*d^2
 + 3*C*c^2*d)/b + (C*d^3)/b))/f - (log(a + b*tan(e + f*x))*(b^4*(B*a*c^3 +
3*A*a*c^2*d) - b^3*(C*a^2*c^3 + 3*A*a^2*c*d^2 + 3*B*a^2*c^2*d) + b^2*(A*a^3
*d^3 + 3*B*a^3*c*d^2 + 3*C*a^3*c^2*d) - b*(B*a^4*d^3 + 3*C*a^4*c*d^2) - A*b
^5*c^3 + C*a^5*d^3))/(f*(b^6 + a^2*b^4)) - (log(tan(e + f*x) + 1i)*(A*c^3 +
 A*d^3*1i - B*c^3*1i + B*d^3 - C*c^3 - C*d^3*1i - 3*A*c*d^2 - A*c^2*d*3i +
B*c*d^2*3i - 3*B*c^2*d + 3*C*c*d^2 + C*c^2*d*3i))/(2*f*(a*1i + b)) - (log(t
an(e + f*x) - 1i)*(A*c^3*1i + A*d^3 - B*c^3 + B*d^3*1i - C*c^3*1i - C*d^3 -
 A*c*d^2*3i - 3*A*c^2*d + 3*B*c*d^2 - B*c^2*d*3i + C*c*d^2*3i + 3*C*c^2*d))
/(2*f*(a + b*1i)) + (C*d^3*tan(e + f*x)^3)/(3*b*f)
```

3.68 $\int \frac{(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^2}\,dx$

Optimal result 636
Rubi [A] (verified) 637
Mathematica [C] (verified) 641
Maple [A] (verified) 642
Fricas [B] (verification not implemented) 642
Sympy [C] (verification not implemented) 643
Maxima [A] (verification not implemented) 656
Giac [B] (verification not implemented) 657
Mupad [B] (verification not implemented) 658

Optimal result

Integrand size = 45, antiderivative size = 574

$$\int \frac{(c+d\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx =$$
$$-\frac{(b^2(Ac^3-c^3C-3Bc^2d-3Acd^2+3cCd^2+Bd^3)+a^2(c^3C+3Bc^2d-3cCd^2-Bd^3-A(c^3-3cd^2))-}{(a^2+b^2)^2}$$
$$+\frac{(2ab(Ac^3-c^3C-3Bc^2d-3Acd^2+3cCd^2+Bd^3)-a^2((A-C)d(3c^2-d^2)+B(c^3-3cd^2))+b^2((A-}{(a^2+b^2)^2 f}$$
$$-\frac{(bc-ad)^2\,(2a^3bBd-3a^4Cd-b^4(Bc+3Ad)-2ab^3(Ac-cC-2Bd)+a^2b^2(Bc-(A+5C)d))\log(a+}{b^4\,(a^2+b^2)^2 f}$$
$$-\frac{d^2(3a^3Cd-Ab^2(bc-ad)-b^3(2cC+Bd)-a^2b(3cC+2Bd)+ab^2(Bc+2Cd))\tan(e+fx)}{b^3\,(a^2+b^2)\,f}$$
$$+\frac{(2Ab^2-2abB+3a^2C+b^2C)\,d(c+d\tan(e+fx))^2}{2b^2\,(a^2+b^2)\,f}$$
$$-\frac{(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^3}{b\,(a^2+b^2)\,f(a+b\tan(e+fx))}$$

```
[Out] -(b^2*(A*c^3-3*A*c*d^2-3*B*c^2*d+B*d^3-C*c^3+3*C*c*d^2)+a^2*(c^3*C+3*B*c^2*
d-3*C*c*d^2-B*d^3-A*(c^3-3*c*d^2))-2*a*b*((A-C)*d*(3*c^2-d^2)+B*(c^3-3*c*d^
2)))*x/(a^2+b^2)^2+(2*a*b*(A*c^3-3*A*c*d^2-3*B*c^2*d+B*d^3-C*c^3+3*C*c*d^2)
-a^2*((A-C)*d*(3*c^2-d^2)+B*(c^3-3*c*d^2))+b^2*((A-C)*d*(3*c^2-d^2)+B*(c^3-
3*c*d^2)))*ln(cos(f*x+e))/(a^2+b^2)^2/f-(-a*d+b*c)^2*(2*a^3*b*B*d-3*a^4*C*d
-b^4*(3*A*d+B*c)-2*a*b^3*(A*c-2*B*d-C*c)+a^2*b^2*(B*c-(A+5*C)*d))*ln(a+b*ta
n(f*x+e))/b^4/(a^2+b^2)^2/f-d^2*(3*a^3*C*d-A*b^2*(-a*d+b*c)-b^3*(B*d+2*C*c)
-a^2*b*(2*B*d+3*C*c)+a*b^2*(B*c+2*C*d))*tan(f*x+e)/b^3/(a^2+b^2)/f+1/2*(2*A
*b^2-2*B*a*b+3*C*a^2+C*b^2)*d*(c+d*tan(f*x+e))^2/b^2/(a^2+b^2)/f-(A*b^2-a*(
B*b-C*a))*(c+d*tan(f*x+e))^3/b/(a^2+b^2)/f/(a+b*tan(f*x+e))
```

Rubi [A] (verified)

Time = 2.39 (sec) , antiderivative size = 574, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 7, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.156, Rules used = {3726, 3728, 3718, 3707, 3698, 31, 3556}

$$\int \frac{(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^2}\,dx$$

$$= \frac{\log(\cos(e+fx))\left(-(a^2(d(A-C)\left(3c^2-d^2\right)+B(c^3-3cd^2)))+2ab(Ac^3-3Acd^2-3Bc^2d+Bd^3-c^3C\right.}{f\left(a^2+b^2\right)^2}$$

$$-\frac{x(a^2(-A(c^3-3cd^2)+3Bc^2d-Bd^3+c^3C-3cCd^2)-2ab(d(A-C)\left(3c^2-d^2\right)+B(c^3-3cd^2))+b^2}{\left(a^2+b^2\right)^2}$$

$$-\frac{\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^3}{bf\left(a^2+b^2\right)(a+b\tan(e+fx))}$$

$$+\frac{d(3a^2C-2abB+2Ab^2+b^2C)\,(c+d\tan(e+fx))^2}{2b^2f\left(a^2+b^2\right)}$$

$$-\frac{d^2\tan(e+fx)\left(3a^3Cd-a^2b(2Bd+3cC)-Ab^2(bc-ad)+ab^2(Bc+2Cd)-b^3(Bd+2cC)\right)}{b^3f\left(a^2+b^2\right)}$$

$$-\frac{(bc-ad)^2\left(-3a^4Cd+2a^3bBd+a^2b^2(Bc-d(A+5C))-2ab^3(Ac-2Bd-cC)-b^4(3Ad+Bc)\right)\log(}{b^4f\left(a^2+b^2\right)^2}$$

```
[In] Int[((c + d*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a + b
*Tan[e + f*x])^2,x]
```

```
[Out] -(((b^2*(A*c^3 - c^3*C - 3*B*c^2*d - 3*A*c*d^2 + 3*c*C*d^2 + B*d^3) + a^2*(
c^3*C + 3*B*c^2*d - 3*c*C*d^2 - B*d^3 - A*(c^3 - 3*c*d^2)) - 2*a*b*((A - C)
*d*(3*c^2 - d^2) + B*(c^3 - 3*c*d^2)))*x)/(a^2 + b^2)^2) + ((2*a*b*(A*c^3 -
 c^3*C - 3*B*c^2*d - 3*A*c*d^2 + 3*c*C*d^2 + B*d^3) - a^2*((A - C)*d*(3*c^2
 - d^2) + B*(c^3 - 3*c*d^2)) + b^2*((A - C)*d*(3*c^2 - d^2) + B*(c^3 - 3*c*
d^2)))*Log[Cos[e + f*x]])/((a^2 + b^2)^2*f) - ((b*c - a*d)^2*(2*a^3*b*B*d -
 3*a^4*C*d - b^4*(B*c + 3*A*d) - 2*a*b^3*(A*c - c*C - 2*B*d) + a^2*b^2*(B*c
 - (A + 5*C)*d))*Log[a + b*Tan[e + f*x]])/(b^4*(a^2 + b^2)^2*f) - (d^2*(3*a
^3*C*d - A*b^2*(b*c - a*d) - b^3*(2*c*C + B*d) - a^2*b*(3*c*C + 2*B*d) + a*
b^2*(B*c + 2*C*d))*Tan[e + f*x])/(b^3*(a^2 + b^2)*f) + ((2*A*b^2 - 2*a*b*B
+ 3*a^2*C + b^2*C)*d*(c + d*Tan[e + f*x])^2)/(2*b^2*(a^2 + b^2)*f) - ((A*b^
2 - a*(b*B - a*C))*(c + d*Tan[e + f*x])^3)/(b*(a^2 + b^2)*f*(a + b*Tan[e +
f*x]))
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3698

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3707

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[(a*A + b*B - a
*C)*(x/(a^2 + b^2)), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3726

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*d^2 + c*(c*C - B*d))*(a + b*Tan[e +
 f*x])^m*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
```

```
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\text{integral} = -\frac{(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^3}{b\,(a^2+b^2)\,f(a+b\tan(e+fx))} + \frac{\int \frac{(c+d\tan(e+fx))^2((bB-aC)(bc-3ad)+Ab(ac+3bd)-b((A-C)(bc-ad)-B(ac+bd))\tan(e+fx)+(2Ab^2-2abB+3a^2C+b^2C)\,d\tan^2(e+fx)}{a+b\tan(e+fx)}}{b\,(a^2+b^2)}$$

$$= \frac{(2Ab^2-2abB+3a^2C+b^2C)\,d(c+d\tan(e+fx))^2}{2b^2\,(a^2+b^2)\,f} - \frac{(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^3}{b\,(a^2+b^2)\,f(a+b\tan(e+fx))} + \frac{\int \frac{(c+d\tan(e+fx))(-2(a(2Ab^2-2abB+3a^2C+b^2C)d^2-bc((bB-aC)(bc-3ad)+Ab(ac+3bd)))+2b^2(2aAcd-2acCd-Ab(c^2-d^2)-}{a+b}}{2b^2}$$

$$= -\frac{d^2(3a^3Cd-Ab^2(bc-ad)-b^3(2cC+Bd)-a^2b(3cC+2Bd)+ab^2(Bc+2Cd))\tan(e+fx)}{b^3\,(a^2+b^2)\,f} + \frac{(2Ab^2-2abB+3a^2C+b^2C)\,d(c+d\tan(e+fx))^2}{2b^2\,(a^2+b^2)\,f} - \frac{(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^3}{b\,(a^2+b^2)\,f(a+b\tan(e+fx))} - \frac{\int \frac{-2(3a^4Cd^3+b^4c^2(Bc+3Ad)-2a^3bd^2(3cC+Bd)+a^2b^2d(3c^2C+3Bcd+(A+2C)d^2)+ab^3(Ac^3-c^3C-3Bc^2d-3Acd^2-3cCd^2-B}{}}{}$$

$$= -\frac{(b^2(Ac^3 - c^3C - 3Bc^2d - 3Acd^2 + 3cCd^2 + Bd^3) + a^2(c^3C + 3Bc^2d - 3cCd^2 - Bd^3 - A(c^3 - 3}{(a^2+b^2)^2}$$
$$- \frac{d^2(3a^3Cd - Ab^2(bc - ad) - b^3(2cC + Bd) - a^2b(3cC + 2Bd) + ab^2(Bc + 2Cd))\tan(e + fx)}{b^3(a^2+b^2)f}$$
$$+ \frac{(2Ab^2 - 2abB + 3a^2C + b^2C)d(c + d\tan(e + fx))^2}{2b^2(a^2+b^2)f}$$
$$- \frac{(Ab^2 - a(bB - aC))(c + d\tan(e + fx))^3}{b(a^2+b^2)f(a + b\tan(e + fx))}$$
$$- \frac{((bc - ad)^2(2a^3bBd - 3a^4Cd - b^4(Bc + 3Ad) - 2ab^3(Ac - cC - 2Bd) + a^2b^2(Bc - (A + 5C)d)}{b^3(a^2+b^2)^2}$$
$$+ \frac{(2b(a^2 + b^2)d(3a^2Cd^2 - 2abd(3cC + Bd) + b^2(3c^2C + 3Bcd + (A - C)d^2)) - 2b(3a^4Cd^3 + b^4c^2}{}$$

$$= -\frac{(b^2(Ac^3 - c^3C - 3Bc^2d - 3Acd^2 + 3cCd^2 + Bd^3) + a^2(c^3C + 3Bc^2d - 3cCd^2 - Bd^3 - A(c^3 - 3}{(a^2+b^2)^2}$$
$$+ \frac{(2ab(Ac^3 - c^3C - 3Bc^2d - 3Acd^2 + 3cCd^2 + Bd^3) - a^2((A - C)d(3c^2 - d^2) + B(c^3 - 3cd^2)) +}{(a^2+b^2)^2 f}$$
$$- \frac{d^2(3a^3Cd - Ab^2(bc - ad) - b^3(2cC + Bd) - a^2b(3cC + 2Bd) + ab^2(Bc + 2Cd))\tan(e + fx)}{b^3(a^2+b^2)f}$$
$$+ \frac{(2Ab^2 - 2abB + 3a^2C + b^2C)d(c + d\tan(e + fx))^2}{2b^2(a^2+b^2)f}$$
$$- \frac{(Ab^2 - a(bB - aC))(c + d\tan(e + fx))^3}{b(a^2+b^2)f(a + b\tan(e + fx))}$$
$$- \frac{((bc - ad)^2(2a^3bBd - 3a^4Cd - b^4(Bc + 3Ad) - 2ab^3(Ac - cC - 2Bd) + a^2b^2(Bc - (A + 5C)d)}{b^4(a^2+b^2)^2 f}$$

$$= -\frac{(b^2(Ac^3 - c^3C - 3Bc^2d - 3Acd^2 + 3cCd^2 + Bd^3) + a^2(c^3C + 3Bc^2d - 3cCd^2 - Bd^3 - A(c^3 - 3}{(a^2+b^2)^2}$$
$$+ \frac{(2ab(Ac^3 - c^3C - 3Bc^2d - 3Acd^2 + 3cCd^2 + Bd^3) - a^2((A - C)d(3c^2 - d^2) + B(c^3 - 3cd^2)) +}{(a^2+b^2)^2 f}$$
$$- \frac{(bc - ad)^2(2a^3bBd - 3a^4Cd - b^4(Bc + 3Ad) - 2ab^3(Ac - cC - 2Bd) + a^2b^2(Bc - (A + 5C)d))}{b^4(a^2+b^2)^2 f}$$
$$- \frac{d^2(3a^3Cd - Ab^2(bc - ad) - b^3(2cC + Bd) - a^2b(3cC + 2Bd) + ab^2(Bc + 2Cd))\tan(e + fx)}{b^3(a^2+b^2)f}$$
$$+ \frac{(2Ab^2 - 2abB + 3a^2C + b^2C)d(c + d\tan(e + fx))^2}{2b^2(a^2+b^2)f}$$
$$- \frac{(Ab^2 - a(bB - aC))(c + d\tan(e + fx))^3}{b(a^2+b^2)f(a + b\tan(e + fx))}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 7.59 (sec) , antiderivative size = 1024, normalized size of antiderivative = 1.78

$$\int \frac{(c+d\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx=\frac{C(c+d\tan(e+fx))^3}{2bf(a+b\tan(e+fx))}$$
$$+\frac{\frac{(3bcC+2bBd-3aCd)(c+d\tan(e+fx))^2}{bf(a+b\tan(e+fx))}+\frac{2\left(-\frac{b^2\left(2aAbc^3-a^2Bc^3+b^2Bc^3-2abc^3C-3a^2Ac^2d+3Ab^2c^2d-6abBc^2d+3a^2c^2Cd-3b^2c^2Cd-6aAbcd^2+3\right.}{}\right.}{}}{}$$

```
[In] Integrate[((c + d*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(a + b*Tan[e + f*x])^2,x]
```

```
[Out] (C*(c + d*Tan[e + f*x])^3)/(2*b*f*(a + b*Tan[e + f*x])) + (((3*b*c*C + 2*b*
B*d - 3*a*C*d)*(c + d*Tan[e + f*x])^2)/(b*f*(a + b*Tan[e + f*x])) + (2*(-1/
2*(b^2*(2*a*A*b*c^3 - a^2*B*c^3 + b^2*B*c^3 - 2*a*b*c^3*C - 3*a^2*A*c^2*d +
 3*A*b^2*c^2*d - 6*a*b*B*c^2*d + 3*a^2*c^2*C*d - 3*b^2*c^2*C*d - 6*a*A*b*c*
d^2 + 3*a^2*B*c*d^2 - 3*b^2*B*c*d^2 + 6*a*b*c*C*d^2 + a^2*A*d^3 - A*b^2*d^3
 + 2*a*b*B*d^3 - a^2*C*d^3 + b^2*C*d^3 + I*(a^2*A*c^3 - A*b^2*c^3 + 2*a*b*B
*c^3 - a^2*c^3*C + b^2*c^3*C + 6*a*A*b*c^2*d - 3*a^2*B*c^2*d + 3*b^2*B*c^2*
d - 6*a*b*c^2*C*d - 3*a^2*A*c*d^2 + 3*A*b^2*c*d^2 - 6*a*b*B*c*d^2 + 3*a^2*c
*C*d^2 - 3*b^2*c*C*d^2 - 2*a*A*b*d^3 + a^2*B*d^3 - b^2*B*d^3 + 2*a*b*C*d^3)
)*Log[I - Tan[e + f*x]])/((a^2 + b^2)^2*f) + (b^2*(-2*a*A*b*c^3 + a^2*B*c^3
 - b^2*B*c^3 + 2*a*b*c^3*C + 3*a^2*A*c^2*d - 3*A*b^2*c^2*d + 6*a*b*B*c^2*d
- 3*a^2*c^2*C*d + 3*b^2*c^2*C*d + 6*a*A*b*c*d^2 - 3*a^2*B*c*d^2 + 3*b^2*B*c
*d^2 - 6*a*b*c*C*d^2 - a^2*A*d^3 + A*b^2*d^3 - 2*a*b*B*d^3 + a^2*C*d^3 - b^
2*C*d^3 + I*(a^2*A*c^3 - A*b^2*c^3 + 2*a*b*B*c^3 - a^2*c^3*C + b^2*c^3*C +
6*a*A*b*c^2*d - 3*a^2*B*c^2*d + 3*b^2*B*c^2*d - 6*a*b*c^2*C*d - 3*a^2*A*c*d
^2 + 3*A*b^2*c*d^2 - 6*a*b*B*c*d^2 + 3*a^2*c*C*d^2 - 3*b^2*c*C*d^2 - 2*a*A*
b*d^3 + a^2*B*d^3 - b^2*B*d^3 + 2*a*b*C*d^3))*Log[I + Tan[e + f*x]])/(2*(a^
2 + b^2)^2*f) - ((b*c - a*d)^2*(2*a^3*b*B*d - 3*a^4*C*d - b^4*(B*c + 3*A*d)
 - 2*a*b^3*(A*c - c*C - 2*B*d) + a^2*b^2*(B*c - (A + 5*C)*d))*Log[a + b*Tan
[e + f*x]])/(b^2*(a^2 + b^2)^2*f) + ((b*c - a*d)^2*(3*a^3*C*d - A*b^2*(b*c
- a*d) - b^3*(2*c*C + B*d) - a^2*b*(3*c*C + 2*B*d) + a*b^2*(B*c + 2*C*d)))/
(b^2*(a^2 + b^2)*f*(a + b*Tan[e + f*x]))))/b)/(2*b)
```

Maple [A] (verified)

Time = 0.36 (sec) , antiderivative size = 829, normalized size of antiderivative = 1.44

method	result
derivativedivides	$\frac{\frac{d^{2}\left(\frac{\tan(fx+e)^{2}Cbd}{2}+\tan(fx+e)bdB-2\tan(fx+e)Cad+3\tan(fx+e)Cbc\right)}{b^{3}}-\frac{-Aa^{3}d^{3}b^{2}+3Ab^{3}cd^{2}a^{2}-3Ab^{4}c^{2}da+Ab^{5}c^{3}+Ba^{4}d^{3}b-}{b^{4}}}{}$
default	$\frac{\frac{d^{2}\left(\frac{\tan(fx+e)^{2}Cbd}{2}+\tan(fx+e)bdB-2\tan(fx+e)Cad+3\tan(fx+e)Cbc\right)}{b^{3}}-\frac{-Aa^{3}d^{3}b^{2}+3Ab^{3}cd^{2}a^{2}-3Ab^{4}c^{2}da+Ab^{5}c^{3}+Ba^{4}d^{3}b-}{b^{4}}}{}$
norman	Expression too large to display
parallelrisch	Expression too large to display
risch	Expression too large to display

```
[In] int((c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2,x
,method=_RETURNVERBOSE)
```

```
[Out] 1/f*(d^2/b^3*(1/2*tan(f*x+e)^2*C*b*d+tan(f*x+e)*b*d*B-2*tan(f*x+e)*C*a*d+3*
tan(f*x+e)*C*b*c)-(-A*a^3*b^2*d^3+3*A*a^2*b^3*c*d^2-3*A*a*b^4*c^2*d+A*b^5*c
^3+B*a^4*b*d^3-3*B*a^3*b^2*c*d^2+3*B*a^2*b^3*c^2*d-B*a*b^4*c^3-C*a^5*d^3+3*
C*a^4*b*c*d^2-3*C*a^3*b^2*c^2*d+C*a^2*b^3*c^3)/b^4/(a^2+b^2)/(a+b*tan(f*x+e
))+1/b^4*(A*a^4*b^2*d^3-3*A*a^2*b^4*c^2*d+3*A*a^2*b^4*d^3+2*A*a*b^5*c^3-6*A
*a*b^5*c*d^2+3*A*b^6*c^2*d-2*B*a^5*b*d^3+3*B*a^4*b^2*c*d^2-4*B*a^3*b^3*d^3-
B*a^2*b^4*c^3+9*B*a^2*b^4*c*d^2-6*B*a*b^5*c^2*d+B*b^6*c^3+3*C*a^6*d^3-6*C*a
^5*b*c*d^2+3*C*a^4*b^2*c^2*d+5*C*a^4*b^2*d^3-12*C*a^3*b^3*c*d^2+9*C*a^2*b^4
*c^2*d-2*C*a*b^5*c^3)/(a^2+b^2)^2*ln(a+b*tan(f*x+e))+1/(a^2+b^2)^2*(1/2*(3*
A*a^2*c^2*d-A*a^2*d^3-2*A*a*b*c^3+6*A*a*b*c*d^2-3*A*b^2*c^2*d+A*b^2*d^3+B*a
^2*c^3-3*B*a^2*c*d^2+6*B*a*b*c^2*d-2*B*a*b*d^3-B*b^2*c^3+3*B*b^2*c*d^2-3*C*
a^2*c^2*d+C*a^2*d^3+2*C*a*b*c^3-6*C*a*b*c*d^2+3*C*b^2*c^2*d-C*b^2*d^3)*ln(1
+tan(f*x+e)^2)+(A*a^2*c^3-3*A*a^2*c*d^2+6*A*a*b*c^2*d-2*A*a*b*d^3-A*b^2*c^3
+3*A*b^2*c*d^2-3*B*a^2*c^2*d+B*a^2*d^3+2*B*a*b*c^3-6*B*a*b*c*d^2+3*B*b^2*c^
2*d-B*b^2*d^3-C*a^2*c^3+3*C*a^2*c*d^2-6*C*a*b*c^2*d+2*C*a*b*d^3+C*b^2*c^3-3
*C*b^2*c*d^2)*arctan(tan(f*x+e))))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1512 vs. 2(571) = 1142.

Time = 1.16 (sec) , antiderivative size = 1512, normalized size of antiderivative = 2.63

$$\int \frac{(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^2}\,dx = \text{Too large to display}$$

```
[In] integrate((c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e
))^2,x, algorithm="fricas")
```

```
[Out] 1/2*((C*a^4*b^3 + 2*C*a^2*b^5 + C*b^7)*d^3*tan(f*x + e)^3 - 2*(C*a^2*b^5 -
B*a*b^6 + A*b^7)*c^3 + 6*(C*a^3*b^4 - B*a^2*b^5 + A*a*b^6)*c^2*d - 6*(C*a^4
*b^3 - B*a^3*b^4 + A*a^2*b^5)*c*d^2 + (3*C*a^5*b^2 - 2*B*a^4*b^3 + 2*(A + C
)*a^3*b^4 + C*a*b^6)*d^3 + 2*(((A - C)*a^3*b^4 + 2*B*a^2*b^5 - (A - C)*a*b^
6)*c^3 - 3*(B*a^3*b^4 - 2*(A - C)*a^2*b^5 - B*a*b^6)*c^2*d - 3*((A - C)*a^3
*b^4 + 2*B*a^2*b^5 - (A - C)*a*b^6)*c*d^2 + (B*a^3*b^4 - 2*(A - C)*a^2*b^5
- B*a*b^6)*d^3)*f*x + (6*(C*a^4*b^3 + 2*C*a^2*b^5 + C*b^7)*c*d^2 - (3*C*a^5
*b^2 - 2*B*a^4*b^3 + 6*C*a^3*b^4 - 4*B*a^2*b^5 + 3*C*a*b^6 - 2*B*b^7)*d^3)*
tan(f*x + e)^2 - ((B*a^3*b^4 - 2*(A - C)*a^2*b^5 - B*a*b^6)*c^3 - 3*(C*a^5*
b^2 - (A - 3*C)*a^3*b^4 - 2*B*a^2*b^5 + A*a*b^6)*c^2*d + 3*(2*C*a^6*b - B*a
^5*b^2 + 4*C*a^4*b^3 - 3*B*a^3*b^4 + 2*A*a^2*b^5)*c*d^2 - (3*C*a^7 - 2*B*a^
6*b + (A + 5*C)*a^5*b^2 - 4*B*a^4*b^3 + 3*A*a^3*b^4)*d^3 + ((B*a^2*b^5 - 2*
(A - C)*a*b^6 - B*b^7)*c^3 - 3*(C*a^4*b^3 - (A - 3*C)*a^2*b^5 - 2*B*a*b^6 +
 A*b^7)*c^2*d + 3*(2*C*a^5*b^2 - B*a^4*b^3 + 4*C*a^3*b^4 - 3*B*a^2*b^5 + 2*
A*a*b^6)*c*d^2 - (3*C*a^6*b - 2*B*a^5*b^2 + (A + 5*C)*a^4*b^3 - 4*B*a^3*b^4
 + 3*A*a^2*b^5)*d^3)*tan(f*x + e))*log((b^2*tan(f*x + e)^2 + 2*a*b*tan(f*x
+ e) + a^2)/(tan(f*x + e)^2 + 1)) - (3*(C*a^5*b^2 + 2*C*a^3*b^4 + C*a*b^6)*
c^2*d - 3*(2*C*a^6*b - B*a^5*b^2 + 4*C*a^4*b^3 - 2*B*a^3*b^4 + 2*C*a^2*b^5
- B*a*b^6)*c*d^2 + (3*C*a^7 - 2*B*a^6*b + (A + 5*C)*a^5*b^2 - 4*B*a^4*b^3 +
 (2*A + C)*a^3*b^4 - 2*B*a^2*b^5 + (A - C)*a*b^6)*d^3 + (3*(C*a^4*b^3 + 2*C
*a^2*b^5 + C*b^7)*c^2*d - 3*(2*C*a^5*b^2 - B*a^4*b^3 + 4*C*a^3*b^4 - 2*B*a^
2*b^5 + 2*C*a*b^6 - B*b^7)*c*d^2 + (3*C*a^6*b - 2*B*a^5*b^2 + (A + 5*C)*a^4
*b^3 - 4*B*a^3*b^4 + (2*A + C)*a^2*b^5 - 2*B*a*b^6 + (A - C)*b^7)*d^3)*tan(
f*x + e))*log(1/(tan(f*x + e)^2 + 1)) + (2*(C*a^3*b^4 - B*a^2*b^5 + A*a*b^6
)*c^3 - 6*(C*a^4*b^3 - B*a^3*b^4 + A*a^2*b^5)*c^2*d + 6*(2*C*a^5*b^2 - B*a^
4*b^3 + (A + 2*C)*a^3*b^4 + C*a*b^6)*c*d^2 - (6*C*a^6*b - 4*B*a^5*b^2 + (2*
A + 7*C)*a^4*b^3 - 4*B*a^3*b^4 + 2*C*a^2*b^5 - 2*B*a*b^6 - C*b^7)*d^3 + 2*(
((A - C)*a^2*b^5 + 2*B*a*b^6 - (A - C)*b^7)*c^3 - 3*(B*a^2*b^5 - 2*(A - C)*
a*b^6 - B*b^7)*c^2*d - 3*((A - C)*a^2*b^5 + 2*B*a*b^6 - (A - C)*b^7)*c*d^2
+ (B*a^2*b^5 - 2*(A - C)*a*b^6 - B*b^7)*d^3)*f*x)*tan(f*x + e))/((a^4*b^5 +
 2*a^2*b^7 + b^9)*f*tan(f*x + e) + (a^5*b^4 + 2*a^3*b^6 + a*b^8)*f)
```

Sympy [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 26.69 (sec) , antiderivative size = 24300, normalized size of antiderivative = 42.33

$$\int \frac{(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^2}\,dx = \text{Too large to display}$$

```
[In] integrate((c+d*tan(f*x+e))**3*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x
+e))**2,x)
```

```
[Out] Piecewise((zoo*x*(c + d*tan(e))**3*(A + B*tan(e) + C*tan(e)**2)/tan(e)**2,
Eq(a, 0) & Eq(b, 0) & Eq(f, 0)), ((A*c**3*x + 3*A*c**2*d*log(tan(e + f*x)**
```

2 + 1)/(2*f) - 3*A*c*d**2*x + 3*A*c*d**2*tan(e + f*x)/f - A*d**3*log(tan(e + f*x)**2 + 1)/(2*f) + A*d**3*tan(e + f*x)**2/(2*f) + B*c**3*log(tan(e + f*x)**2 + 1)/(2*f) - 3*B*c**2*d*x + 3*B*c**2*d*tan(e + f*x)/f - 3*B*c*d**2*log(tan(e + f*x)**2 + 1)/(2*f) + 3*B*c*d**2*tan(e + f*x)**2/(2*f) + B*d**3*x + B*d**3*tan(e + f*x)**3/(3*f) - B*d**3*tan(e + f*x)/f - C*c**3*x + C*c**3*tan(e + f*x)/f - 3*C*c**2*d*log(tan(e + f*x)**2 + 1)/(2*f) + 3*C*c**2*d*tan(e + f*x)**2/(2*f) + 3*C*c*d**2*x + C*c*d**2*tan(e + f*x)**3/f - 3*C*c*d**2*tan(e + f*x)/f + C*d**3*log(tan(e + f*x)**2 + 1)/(2*f) + C*d**3*tan(e + f*x)**4/(4*f) - C*d**3*tan(e + f*x)**2/(2*f))/a**2, Eq(b, 0)), (-A*c**3*f*x*tan(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 2*I*A*c**3*f*x*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + A*c**3*f*x/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - A*c**3*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 2*I*A*c**3/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 3*I*A*c**2*d*f*x*tan(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 6*A*c**2*d*f*x*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 3*I*A*c**2*d*f*x/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 3*I*A*c**2*d*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 3*A*c*d**2*f*x*tan(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 6*I*A*c*d**2*f*x*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 3*A*c*d**2*f*x/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 9*A*c*d**2*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 6*I*A*c*d**2/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 3*I*A*d**3*f*x*tan(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 6*A*d**3*f*x*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 3*I*A*d**3*f*x/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 2*A*d**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 4*I*A*d**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 2*A*d**3*log(tan(e + f*x)**2 + 1)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 5*I*A*d**3*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 4*A*d**3/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + I*B*c**3*f*x*tan(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 2*B*c**3*f*x*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - I*B*c**3*f*x/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + I*B*c**3*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 3*B*c**2*d*f*x*tan(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 6*I*B*c**2*d*f*x*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 3*B*c**2*d*f*x/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 9*B*c**2*d*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*

b**2*f) + 6*I*B*c**2*d/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x)
- 4*b**2*f) + 9*I*B*c*d**2*f*x*tan(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 -
8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 18*B*c*d**2*f*x*tan(e + f*x)/(4*b**2*
f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 9*I*B*c*d**2*f*x/
(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 6*B*c*d**
2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 - 8*I*
b**2*f*tan(e + f*x) - 4*b**2*f) - 12*I*B*c*d**2*log(tan(e + f*x)**2 + 1)*ta
n(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f)
- 6*B*c*d**2*log(tan(e + f*x)**2 + 1)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*
f*tan(e + f*x) - 4*b**2*f) - 15*I*B*c*d**2*tan(e + f*x)/(4*b**2*f*tan(e + f
*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 12*B*c*d**2/(4*b**2*f*tan(e
+ f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 9*B*d**3*f*x*tan(e + f*x)
2/(4*b2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 18*I*
B*d**3*f*x*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x)
 - 4*b**2*f) + 9*B*d**3*f*x/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e +
f*x) - 4*b**2*f) + 4*I*B*d**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)**2/(4*b
2*f*tan(e + f*x)2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 8*B*d**3*log(
tan(e + f*x)**2 + 1)*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*ta
n(e + f*x) - 4*b**2*f) - 4*I*B*d**3*log(tan(e + f*x)**2 + 1)/(4*b**2*f*tan(
e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 4*B*d**3*tan(e + f*x)**
3/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 19*B*d*
*3*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**
2*f) - 14*I*B*d**3/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*
b**2*f) + C*c**3*f*x*tan(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f
*tan(e + f*x) - 4*b**2*f) - 2*I*C*c**3*f*x*tan(e + f*x)/(4*b**2*f*tan(e + f
*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - C*c**3*f*x/(4*b**2*f*tan(e +
 f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 3*C*c**3*tan(e + f*x)/(4*b
2*f*tan(e + f*x)2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 2*I*C*c**3/(4
*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 9*I*C*c**2*
d*f*x*tan(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) -
 4*b**2*f) + 18*C*c**2*d*f*x*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b
2*f*tan(e + f*x) - 4*b2*f) - 9*I*C*c**2*d*f*x/(4*b**2*f*tan(e + f*x)**2
 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 6*C*c**2*d*log(tan(e + f*x)**2 + 1
)*tan(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b
2*f) - 12*I*C*c2*d*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(4*b**2*f*tan(
e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 6*C*c**2*d*log(tan(e +
f*x)**2 + 1)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f
) - 15*I*C*c**2*d*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e
 + f*x) - 4*b**2*f) - 12*C*c**2*d/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*ta
n(e + f*x) - 4*b**2*f) - 27*C*c*d**2*f*x*tan(e + f*x)**2/(4*b**2*f*tan(e +
f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 54*I*C*c*d**2*f*x*tan(e + f
*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 27*C*
c*d**2*f*x/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f)
+ 12*I*C*c*d**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)**2/(4*b**2*f*tan(e +
f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 24*C*c*d**2*log(tan(e + f*x

)**2 + 1)*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x)
- 4*b**2*f) - 12*I*C*c*d**2*log(tan(e + f*x)**2 + 1)/(4*b**2*f*tan(e + f*x)
2 - 8*I*b2*f*tan(e + f*x) - 4*b**2*f) + 12*C*c*d**2*tan(e + f*x)**3/(4*
b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 57*C*c*d**2*
tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f
) - 42*I*C*c*d**2/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b
2*f) - 15*I*C*d3*f*x*tan(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 - 8*I*b*
*2*f*tan(e + f*x) - 4*b**2*f) - 30*C*d**3*f*x*tan(e + f*x)/(4*b**2*f*tan(e
+ f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 15*I*C*d**3*f*x/(4*b**2*f
*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 8*C*d**3*log(tan(e
 + f*x)**2 + 1)*tan(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(
e + f*x) - 4*b**2*f) + 16*I*C*d**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(4
*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 8*C*d**3*lo
g(tan(e + f*x)**2 + 1)/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x)
- 4*b**2*f) + 2*C*d**3*tan(e + f*x)**4/(4*b**2*f*tan(e + f*x)**2 - 8*I*b**2
*f*tan(e + f*x) - 4*b**2*f) + 4*I*C*d**3*tan(e + f*x)**3/(4*b**2*f*tan(e +
f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 29*I*C*d**3*tan(e + f*x)/(4
*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 22*C*d**3/(
4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) - 4*b**2*f), Eq(a, -I*b)
), (-A*c**3*f*x*tan(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(
e + f*x) - 4*b**2*f) - 2*I*A*c**3*f*x*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**
2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + A*c**3*f*x/(4*b**2*f*tan(e + f*x)
2 + 8*I*b2*f*tan(e + f*x) - 4*b**2*f) - A*c**3*tan(e + f*x)/(4*b**2*f*t
an(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 2*I*A*c**3/(4*b**2*f
*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 3*I*A*c**2*d*f*x*t
an(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2
*f) + 6*A*c**2*d*f*x*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*ta
n(e + f*x) - 4*b**2*f) + 3*I*A*c**2*d*f*x/(4*b**2*f*tan(e + f*x)**2 + 8*I*b
2*f*tan(e + f*x) - 4*b2*f) - 3*I*A*c**2*d*tan(e + f*x)/(4*b**2*f*tan(e
+ f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 3*A*c*d**2*f*x*tan(e + f*
x)**2/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 6*I
*A*c*d**2*f*x*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f
*x) - 4*b**2*f) - 3*A*c*d**2*f*x/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan
(e + f*x) - 4*b**2*f) - 9*A*c*d**2*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 +
 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 6*I*A*c*d**2/(4*b**2*f*tan(e + f*x)*
*2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 3*I*A*d**3*f*x*tan(e + f*x)**2/(
4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 6*A*d**3*f
*x*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**
2*f) + 3*I*A*d**3*f*x/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) -
 4*b**2*f) + 2*A*d**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)**2/(4*b**2*f*ta
n(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 4*I*A*d**3*log(tan(e
+ f*x)**2 + 1)*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e +
f*x) - 4*b**2*f) - 2*A*d**3*log(tan(e + f*x)**2 + 1)/(4*b**2*f*tan(e + f*x)
2 + 8*I*b2*f*tan(e + f*x) - 4*b**2*f) + 5*I*A*d**3*tan(e + f*x)/(4*b**2
*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 4*A*d**3/(4*b**2

```
*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - I*B*c**3*f*x*tan
(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f
) + 2*B*c**3*f*x*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e
+ f*x) - 4*b**2*f) + I*B*c**3*f*x/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*ta
n(e + f*x) - 4*b**2*f) - I*B*c**3*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 +
8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 3*B*c**2*d*f*x*tan(e + f*x)**2/(4*b**
2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 6*I*B*c**2*d*f*
x*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2
*f) - 3*B*c**2*d*f*x/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) -
4*b**2*f) - 9*B*c**2*d*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*
tan(e + f*x) - 4*b**2*f) - 6*I*B*c**2*d/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**
2*f*tan(e + f*x) - 4*b**2*f) - 9*I*B*c*d**2*f*x*tan(e + f*x)**2/(4*b**2*f*t
an(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 18*B*c*d**2*f*x*tan(
e + f*x)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) +
9*I*B*c*d**2*f*x/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b*
*2*f) + 6*B*c*d**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)**2/(4*b**2*f*tan(e
 + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 12*I*B*c*d**2*log(tan(e
+ f*x)**2 + 1)*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e +
f*x) - 4*b**2*f) - 6*B*c*d**2*log(tan(e + f*x)**2 + 1)/(4*b**2*f*tan(e + f*
x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 15*I*B*c*d**2*tan(e + f*x)/(4
*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 12*B*c*d**2
/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 9*B*d**3
*f*x*tan(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) -
4*b**2*f) - 18*I*B*d**3*f*x*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b*
*2*f*tan(e + f*x) - 4*b**2*f) + 9*B*d**3*f*x/(4*b**2*f*tan(e + f*x)**2 + 8*
I*b**2*f*tan(e + f*x) - 4*b**2*f) - 4*I*B*d**3*log(tan(e + f*x)**2 + 1)*tan
(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f
) + 8*B*d**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**
2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 4*I*B*d**3*log(tan(e + f*x)**2 +
1)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 4*B*d*
*3*tan(e + f*x)**3/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*
b**2*f) + 19*B*d**3*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan
(e + f*x) - 4*b**2*f) + 14*I*B*d**3/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*
tan(e + f*x) - 4*b**2*f) + C*c**3*f*x*tan(e + f*x)**2/(4*b**2*f*tan(e + f*x
)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 2*I*C*c**3*f*x*tan(e + f*x)/(4
*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - C*c**3*f*x/
(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 3*C*c**3*
tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f
) - 2*I*C*c**3/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2
*f) - 9*I*C*c**2*d*f*x*tan(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2
*f*tan(e + f*x) - 4*b**2*f) + 18*C*c**2*d*f*x*tan(e + f*x)/(4*b**2*f*tan(e
+ f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 9*I*C*c**2*d*f*x/(4*b**2*
f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 6*C*c**2*d*log(ta
n(e + f*x)**2 + 1)*tan(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*t
an(e + f*x) - 4*b**2*f) + 12*I*C*c**2*d*log(tan(e + f*x)**2 + 1)*tan(e + f*
```

x)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 6*C*c*
*2*d*log(tan(e + f*x)**2 + 1)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e
+ f*x) - 4*b**2*f) + 15*I*C*c**2*d*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 +
 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 12*C*c**2*d/(4*b**2*f*tan(e + f*x)**
2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 27*C*c*d**2*f*x*tan(e + f*x)**2/(
4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 54*I*C*c*d
2*f*x*tan(e + f*x)/(4*b2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) -
4*b**2*f) + 27*C*c*d**2*f*x/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e +
f*x) - 4*b**2*f) - 12*I*C*c*d**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)**2/(
4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 24*C*c*d**
2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**
2*f*tan(e + f*x) - 4*b**2*f) + 12*I*C*c*d**2*log(tan(e + f*x)**2 + 1)/(4*b*
*2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) + 12*C*c*d**2*ta
n(e + f*x)**3/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*
f) + 57*C*c*d**2*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e
+ f*x) - 4*b**2*f) + 42*I*C*c*d**2/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*t
an(e + f*x) - 4*b**2*f) + 15*I*C*d**3*f*x*tan(e + f*x)**2/(4*b**2*f*tan(e +
 f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 30*C*d**3*f*x*tan(e + f*x)
/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 15*I*C*d
3*f*x/(4*b2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 8
*C*d**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)**2/(4*b**2*f*tan(e + f*x)**2
+ 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 16*I*C*d**3*log(tan(e + f*x)**2 + 1
)*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2
*f) + 8*C*d**3*log(tan(e + f*x)**2 + 1)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**
2*f*tan(e + f*x) - 4*b**2*f) + 2*C*d**3*tan(e + f*x)**4/(4*b**2*f*tan(e + f
*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 4*I*C*d**3*tan(e + f*x)**3/(
4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2*f) - 29*I*C*d**
3*tan(e + f*x)/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**2
*f) + 22*C*d**3/(4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) - 4*b**
2*f), Eq(a, I*b)), (x*(c + d*tan(e))**3*(A + B*tan(e) + C*tan(e)**2)/(a + b
*tan(e))**2, Eq(f, 0)), (2*A*a**5*b**2*d**3*log(a/b + tan(e + f*x))/(2*a**5
*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e
+ f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 2*A*a**5*b**2*d**3/(2*a**5*b
4*f + 2*a4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e +
f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) - 6*A*a**4*b**3*c*d**2/(2*a**5*b
4*f + 2*a4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e +
f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 2*A*a**4*b**3*d**3*log(a/b + t
an(e + f*x))*tan(e + f*x)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a
3*b6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x
)) + 2*A*a**3*b**4*c**3*f*x/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4
*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f
*x)) - 6*A*a**3*b**4*c**2*d*log(a/b + tan(e + f*x))/(2*a**5*b**4*f + 2*a**4
*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**
8*f + 2*b**9*f*tan(e + f*x)) + 3*A*a**3*b**4*c**2*d*log(tan(e + f*x)**2 + 1
)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7

```
*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 6*A*a**3*b**4*c**2*
d/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7
*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) - 6*A*a**3*b**4*c*d**
2*f*x/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*
b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 6*A*a**3*b**4*d
**3*log(a/b + tan(e + f*x))/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4
*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f
*x)) - A*a**3*b**4*d**3*log(tan(e + f*x)**2 + 1)/(2*a**5*b**4*f + 2*a**4*b*
*5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f
 + 2*b**9*f*tan(e + f*x)) + 2*A*a**3*b**4*d**3/(2*a**5*b**4*f + 2*a**4*b**5
*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f +
 2*b**9*f*tan(e + f*x)) + 2*A*a**2*b**5*c**3*f*x*tan(e + f*x)/(2*a**5*b**4*
f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x)
 + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 4*A*a**2*b**5*c**3*log(a/b + tan(e
 + f*x))/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a*
*2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) - 2*A*a**2*b**
5*c**3*log(tan(e + f*x)**2 + 1)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x)
 + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e
 + f*x)) - 2*A*a**2*b**5*c**3/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) +
 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e +
 f*x)) + 12*A*a**2*b**5*c**2*d*f*x/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f
*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*ta
n(e + f*x)) - 6*A*a**2*b**5*c**2*d*log(a/b + tan(e + f*x))*tan(e + f*x)/(2*
a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*ta
n(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 3*A*a**2*b**5*c**2*d*log
(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f
*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*ta
n(e + f*x)) - 6*A*a**2*b**5*c*d**2*f*x*tan(e + f*x)/(2*a**5*b**4*f + 2*a**4
*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**
8*f + 2*b**9*f*tan(e + f*x)) - 12*A*a**2*b**5*c*d**2*log(a/b + tan(e + f*x)
)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7
*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 6*A*a**2*b**5*c*d**
2*log(tan(e + f*x)**2 + 1)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*
a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*
x)) - 6*A*a**2*b**5*c*d**2/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*
a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*
x)) - 4*A*a**2*b**5*d**3*f*x/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) +
4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e +
f*x)) + 6*A*a**2*b**5*d**3*log(a/b + tan(e + f*x))*tan(e + f*x)/(2*a**5*b**
4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*
x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) - A*a**2*b**5*d**3*log(tan(e + f*x
)**2 + 1)*tan(e + f*x)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3
*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x))
- 2*A*a*b**6*c**3*f*x/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*
b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) +
```

```
4*A*a*b**6*c**3*log(a/b + tan(e + f*x))*tan(e + f*x)/(2*a**5*b**4*f + 2*a*
*4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b
**8*f + 2*b**9*f*tan(e + f*x)) - 2*A*a*b**6*c**3*log(tan(e + f*x)**2 + 1)*t
an(e + f*x)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4
*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 12*A*a*b*
*6*c**2*d*f*x*tan(e + f*x)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*
a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*
x)) + 6*A*a*b**6*c**2*d*log(a/b + tan(e + f*x))/(2*a**5*b**4*f + 2*a**4*b**
5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f
+ 2*b**9*f*tan(e + f*x)) - 3*A*a*b**6*c**2*d*log(tan(e + f*x)**2 + 1)/(2*a*
*5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(
e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 6*A*a*b**6*c**2*d/(2*a**5*
b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e +
 f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 6*A*a*b**6*c*d**2*f*x/(2*a**5
*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e
+ f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) - 12*A*a*b**6*c*d**2*log(a/b +
 tan(e + f*x))*tan(e + f*x)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4
*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f
*x)) + 6*A*a*b**6*c*d**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*a**5*b**4
*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x
) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) - 4*A*a*b**6*d**3*f*x*tan(e + f*x)/
(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f
*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + A*a*b**6*d**3*log(tan
(e + f*x)**2 + 1)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6
*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) - 2*A
*b**7*c**3*f*x*tan(e + f*x)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4
*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f
*x)) - 2*A*b**7*c**3/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b
**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) +
6*A*b**7*c**2*d*log(a/b + tan(e + f*x))*tan(e + f*x)/(2*a**5*b**4*f + 2*a**
4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b*
*8*f + 2*b**9*f*tan(e + f*x)) - 3*A*b**7*c**2*d*log(tan(e + f*x)**2 + 1)*ta
n(e + f*x)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*
a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 6*A*b**7*c
*d**2*f*x*tan(e + f*x)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3
*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x))
+ A*b**7*d**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*a**5*b**4*f + 2*a**4
*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**
8*f + 2*b**9*f*tan(e + f*x)) - 4*B*a**6*b*d**3*log(a/b + tan(e + f*x))/(2*a
**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan
(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) - 4*B*a**6*b*d**3/(2*a**5*b
**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e +
f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 6*B*a**5*b**2*c*d**2*log(a/b +
 tan(e + f*x))/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f
+ 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 6*B*a*
```

```
*5*b**2*c*d**2/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f
+ 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) - 4*B*a*
*5*b**2*d**3*log(a/b + tan(e + f*x))*tan(e + f*x)/(2*a**5*b**4*f + 2*a**4*b
**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*
f + 2*b**9*f*tan(e + f*x)) - 6*B*a**4*b**3*c**2*d/(2*a**5*b**4*f + 2*a**4*b
**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*
f + 2*b**9*f*tan(e + f*x)) + 6*B*a**4*b**3*c*d**2*log(a/b + tan(e + f*x))*t
an(e + f*x)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4
*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) - 8*B*a**4*
b**3*d**3*log(a/b + tan(e + f*x))/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*
x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan
(e + f*x)) + 2*B*a**4*b**3*d**3*tan(e + f*x)**2/(2*a**5*b**4*f + 2*a**4*b**
5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f
+ 2*b**9*f*tan(e + f*x)) - 6*B*a**4*b**3*d**3/(2*a**5*b**4*f + 2*a**4*b**5*
f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f +
2*b**9*f*tan(e + f*x)) - 2*B*a**3*b**4*c**3*log(a/b + tan(e + f*x))/(2*a**5
*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e
+ f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + B*a**3*b**4*c**3*log(tan(e +
 f*x)**2 + 1)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f +
 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 2*B*a**
3*b**4*c**3/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4
*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) - 6*B*a**3*
b**4*c**2*d*f*x/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f
 + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 18*B*
a**3*b**4*c*d**2*log(a/b + tan(e + f*x))/(2*a**5*b**4*f + 2*a**4*b**5*f*tan
(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**
9*f*tan(e + f*x)) - 3*B*a**3*b**4*c*d**2*log(tan(e + f*x)**2 + 1)/(2*a**5*b
**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e +
f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 6*B*a**3*b**4*c*d**2/(2*a**5*b
**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e +
f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 2*B*a**3*b**4*d**3*f*x/(2*a**5
*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e
+ f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) - 8*B*a**3*b**4*d**3*log(a/b +
 tan(e + f*x))*tan(e + f*x)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4
*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f
*x)) + 4*B*a**2*b**5*c**3*f*x/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) +
 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e +
 f*x)) - 2*B*a**2*b**5*c**3*log(a/b + tan(e + f*x))*tan(e + f*x)/(2*a**5*b*
*4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f
*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + B*a**2*b**5*c**3*log(tan(e + f*
x)**2 + 1)*tan(e + f*x)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**
3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x))
 - 6*B*a**2*b**5*c**2*d*f*x*tan(e + f*x)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan
(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**
9*f*tan(e + f*x)) - 12*B*a**2*b**5*c**2*d*log(a/b + tan(e + f*x))/(2*a**5*b
```

```
**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e +
f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 6*B*a**2*b**5*c**2*d*log(tan(e
 + f*x)**2 + 1)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f
 + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) - 6*B*a
**2*b**5*c**2*d/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f
 + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) - 12*B*
a**2*b**5*c*d**2*f*x/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b
**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) +
18*B*a**2*b**5*c*d**2*log(a/b + tan(e + f*x))*tan(e + f*x)/(2*a**5*b**4*f +
 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) +
2*a*b**8*f + 2*b**9*f*tan(e + f*x)) - 3*B*a**2*b**5*c*d**2*log(tan(e + f*x)
**2 + 1)*tan(e + f*x)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*
b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) +
 2*B*a**2*b**5*d**3*f*x*tan(e + f*x)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e +
 f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*
tan(e + f*x)) - 2*B*a**2*b**5*d**3*log(tan(e + f*x)**2 + 1)/(2*a**5*b**4*f
+ 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) +
 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 4*B*a**2*b**5*d**3*tan(e + f*x)**2/(
2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*
tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) - 2*B*a**2*b**5*d**3/(2*
a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*ta
n(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 4*B*a*b**6*c**3*f*x*tan(
e + f*x)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a*
*2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 2*B*a*b**6*c
**3*log(a/b + tan(e + f*x))/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4
*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f
*x)) - B*a*b**6*c**3*log(tan(e + f*x)**2 + 1)/(2*a**5*b**4*f + 2*a**4*b**5*
f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f +
2*b**9*f*tan(e + f*x)) + 2*B*a*b**6*c**3/(2*a**5*b**4*f + 2*a**4*b**5*f*tan
(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**
9*f*tan(e + f*x)) + 6*B*a*b**6*c**2*d*f*x/(2*a**5*b**4*f + 2*a**4*b**5*f*ta
n(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b*
*9*f*tan(e + f*x)) - 12*B*a*b**6*c**2*d*log(a/b + tan(e + f*x))*tan(e + f*x
)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7
*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 6*B*a*b**6*c**2*d*l
og(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e +
 f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*
tan(e + f*x)) - 12*B*a*b**6*c*d**2*f*x*tan(e + f*x)/(2*a**5*b**4*f + 2*a**4
*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**
8*f + 2*b**9*f*tan(e + f*x)) + 3*B*a*b**6*c*d**2*log(tan(e + f*x)**2 + 1)/(
2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*
tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) - 2*B*a*b**6*d**3*f*x/(2
*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*t
an(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) - 2*B*a*b**6*d**3*log(tan
(e + f*x)**2 + 1)*tan(e + f*x)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x)
```

+ 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e
+ f*x)) + 2*B*b**7*c**3*log(a/b + tan(e + f*x))*tan(e + f*x)/(2*a**5*b**4*f
+ 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x)
+ 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) - B*b**7*c**3*log(tan(e + f*x)**2 + 1
)*tan(e + f*x)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f
+ 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 6*B*b*
*7*c**2*d*f*x*tan(e + f*x)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*
a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*
x)) + 3*B*b**7*c*d**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*a**5*b**4*f
+ 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) +
2*a*b**8*f + 2*b**9*f*tan(e + f*x)) - 2*B*b**7*d**3*f*x*tan(e + f*x)/(2*a*
*5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(
e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 2*B*b**7*d**3*tan(e + f*x)
2/(2*a5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b*
*7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 6*C*a**7*d**3*log
(a/b + tan(e + f*x))/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b
6*f + 4*a2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) +
6*C*a**7*d**3/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f +
4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) - 12*C*a*
*6*b*c*d**2*log(a/b + tan(e + f*x))/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e +
f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*t
an(e + f*x)) - 12*C*a**6*b*c*d**2/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*
x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan
(e + f*x)) + 6*C*a**6*b*d**3*log(a/b + tan(e + f*x))*tan(e + f*x)/(2*a**5*b
4*f + 2*a4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e +
f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 6*C*a**5*b**2*c**2*d*log(a/b +
tan(e + f*x))/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f
+ 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 6*C*a*
*5*b**2*c**2*d/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f
+ 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) - 12*C*a
5*b2*c*d**2*log(a/b + tan(e + f*x))*tan(e + f*x)/(2*a**5*b**4*f + 2*a**
4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b*
*8*f + 2*b**9*f*tan(e + f*x)) + 10*C*a**5*b**2*d**3*log(a/b + tan(e + f*x))
/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*
f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) - 3*C*a**5*b**2*d**3*t
an(e + f*x)**2/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f
+ 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 10*C*a
5*b2*d**3/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f +
4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) - 2*C*a**
4*b**3*c**3/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4
*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 6*C*a**4*
b**3*c**2*d*log(a/b + tan(e + f*x))*tan(e + f*x)/(2*a**5*b**4*f + 2*a**4*b*
*5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f
+ 2*b**9*f*tan(e + f*x)) - 24*C*a**4*b**3*c*d**2*log(a/b + tan(e + f*x))/(
2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*

```
tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 6*C*a**4*b**3*c*d**2*t
an(e + f*x)**2/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f
+ 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) - 18*C*a
**4*b**3*c*d**2/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f
 + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 10*C*
a**4*b**3*d**3*log(a/b + tan(e + f*x))*tan(e + f*x)/(2*a**5*b**4*f + 2*a**4
*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**
8*f + 2*b**9*f*tan(e + f*x)) + C*a**4*b**3*d**3*tan(e + f*x)**3/(2*a**5*b**
4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*
x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) - 2*C*a**3*b**4*c**3*f*x/(2*a**5*b
**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e +
f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 18*C*a**3*b**4*c**2*d*log(a/b
+ tan(e + f*x))/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f
 + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) - 3*C*a
**3*b**4*c**2*d*log(tan(e + f*x)**2 + 1)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan
(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**
9*f*tan(e + f*x)) + 6*C*a**3*b**4*c**2*d/(2*a**5*b**4*f + 2*a**4*b**5*f*tan
(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**
9*f*tan(e + f*x)) + 6*C*a**3*b**4*c*d**2*f*x/(2*a**5*b**4*f + 2*a**4*b**5*f
*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2
*b**9*f*tan(e + f*x)) - 24*C*a**3*b**4*c*d**2*log(a/b + tan(e + f*x))*tan(e
 + f*x)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**
2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + C*a**3*b**4*d
**3*log(tan(e + f*x)**2 + 1)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) +
4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e +
f*x)) - 6*C*a**3*b**4*d**3*tan(e + f*x)**2/(2*a**5*b**4*f + 2*a**4*b**5*f*t
an(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b
**9*f*tan(e + f*x)) + 4*C*a**3*b**4*d**3/(2*a**5*b**4*f + 2*a**4*b**5*f*tan
(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**
9*f*tan(e + f*x)) - 2*C*a**2*b**5*c**3*f*x*tan(e + f*x)/(2*a**5*b**4*f + 2*
a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a
*b**8*f + 2*b**9*f*tan(e + f*x)) - 4*C*a**2*b**5*c**3*log(a/b + tan(e + f*x
))/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**
7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 2*C*a**2*b**5*c**3
*log(tan(e + f*x)**2 + 1)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a
**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x
)) - 2*C*a**2*b**5*c**3/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**
3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x))
 - 12*C*a**2*b**5*c**2*d*f*x/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) +
4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e +
f*x)) + 18*C*a**2*b**5*c**2*d*log(a/b + tan(e + f*x))*tan(e + f*x)/(2*a**5*
b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e +
 f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) - 3*C*a**2*b**5*c**2*d*log(tan(
e + f*x)**2 + 1)*tan(e + f*x)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) +
 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e +
```

f*x)) + 6*C*a**2*b**5*c*d**2*f*x*tan(e + f*x)/(2*a**5*b**4*f + 2*a**4*b**5
*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f +
2*b**9*f*tan(e + f*x)) - 6*C*a**2*b**5*c*d**2*log(tan(e + f*x)**2 + 1)/(2*
a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*ta
n(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 12*C*a**2*b**5*c*d**2*ta
n(e + f*x)**2/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f +
4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) - 6*C*a**
2*b**5*c*d**2/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f +
4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 4*C*a**
2*b**5*d**3*f*x/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f
+ 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + C*a**
2*b**5*d**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*a**5*b**4*f + 2*a**4*b
5*f*tan(e + f*x) + 4*a3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*
f + 2*b**9*f*tan(e + f*x)) + 2*C*a**2*b**5*d**3*tan(e + f*x)**3/(2*a**5*b**
4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*
x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 2*C*a*b**6*c**3*f*x/(2*a**5*b**4
*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x
) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) - 4*C*a*b**6*c**3*log(a/b + tan(e +
f*x))*tan(e + f*x)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b*
*6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 2
*C*a*b**6*c**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*a**5*b**4*f + 2*a**
4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b*
*8*f + 2*b**9*f*tan(e + f*x)) - 12*C*a*b**6*c**2*d*f*x*tan(e + f*x)/(2*a**5
*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e
+ f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 3*C*a*b**6*c**2*d*log(tan(e
+ f*x)**2 + 1)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f
+ 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) - 6*C*a*
b**6*c*d**2*f*x/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f
+ 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) - 6*C*a
*b**6*c*d**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*a**5*b**4*f + 2*a**4*
b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8
*f + 2*b**9*f*tan(e + f*x)) + 4*C*a*b**6*d**3*f*x*tan(e + f*x)/(2*a**5*b**4
*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x
) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) - C*a*b**6*d**3*log(tan(e + f*x)**2
+ 1)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*
b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) - 3*C*a*b**6*d**3
*tan(e + f*x)**2/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*
f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 2*C*
b**7*c**3*f*x*tan(e + f*x)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*
a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*
x)) + 3*C*b**7*c**2*d*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*a**5*b**4*f
+ 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) +
2*a*b**8*f + 2*b**9*f*tan(e + f*x)) - 6*C*b**7*c*d**2*f*x*tan(e + f*x)/(2*
a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*ta
n(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) + 6*C*b**7*c*d**2*tan(e +

```
f*x)**2/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e + f*x) + 4*a**3*b**6*f + 4*a**
2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*tan(e + f*x)) - C*b**7*d**3*l
og(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*a**5*b**4*f + 2*a**4*b**5*f*tan(e +
 f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*b**9*f*
tan(e + f*x)) + C*b**7*d**3*tan(e + f*x)**3/(2*a**5*b**4*f + 2*a**4*b**5*f*
tan(e + f*x) + 4*a**3*b**6*f + 4*a**2*b**7*f*tan(e + f*x) + 2*a*b**8*f + 2*
b**9*f*tan(e + f*x)), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.32 (sec) , antiderivative size = 685, normalized size of antiderivative = 1.19

$$\int \frac{(c+d\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx$$

$$= \frac{\frac{2\,(((A-C)a^2+2\,Bab-(A-C)b^2)c^3-3\,(Ba^2-2\,(A-C)ab-Bb^2)c^2d-3\,((A-C)a^2+2\,Bab-(A-C)b^2)cd^2+(Ba^2-2\,(A-C)ab-Bb^2)d^3)(fx+e)}{a^4+2\,a^2b^2+b^4} - }{}$$

```
[In] integrate((c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e
))^2,x, algorithm="maxima")
```

```
[Out] 1/2*(2*(((A - C)*a^2 + 2*B*a*b - (A - C)*b^2)*c^3 - 3*(B*a^2 - 2*(A - C)*a*
b - B*b^2)*c^2*d - 3*((A - C)*a^2 + 2*B*a*b - (A - C)*b^2)*c*d^2 + (B*a^2 -
 2*(A - C)*a*b - B*b^2)*d^3)*(f*x + e)/(a^4 + 2*a^2*b^2 + b^4) - 2*((B*a^2*
b^4 - 2*(A - C)*a*b^5 - B*b^6)*c^3 - 3*(C*a^4*b^2 - (A - 3*C)*a^2*b^4 - 2*B
*a*b^5 + A*b^6)*c^2*d + 3*(2*C*a^5*b - B*a^4*b^2 + 4*C*a^3*b^3 - 3*B*a^2*b^
4 + 2*A*a*b^5)*c*d^2 - (3*C*a^6 - 2*B*a^5*b + (A + 5*C)*a^4*b^2 - 4*B*a^3*b
^3 + 3*A*a^2*b^4)*d^3)*log(b*tan(f*x + e) + a)/(a^4*b^4 + 2*a^2*b^6 + b^8)
+ ((B*a^2 - 2*(A - C)*a*b - B*b^2)*c^3 + 3*((A - C)*a^2 + 2*B*a*b - (A - C)
*b^2)*c^2*d - 3*(B*a^2 - 2*(A - C)*a*b - B*b^2)*c*d^2 - ((A - C)*a^2 + 2*B*
a*b - (A - C)*b^2)*d^3)*log(tan(f*x + e)^2 + 1)/(a^4 + 2*a^2*b^2 + b^4) - 2
*((C*a^2*b^3 - B*a*b^4 + A*b^5)*c^3 - 3*(C*a^3*b^2 - B*a^2*b^3 + A*a*b^4)*c
^2*d + 3*(C*a^4*b - B*a^3*b^2 + A*a^2*b^3)*c*d^2 - (C*a^5 - B*a^4*b + A*a^3
*b^2)*d^3)/(a^3*b^4 + a*b^6 + (a^2*b^5 + b^7)*tan(f*x + e)) + (C*b*d^3*tan(
f*x + e)^2 + 2*(3*C*b*c*d^2 - (2*C*a - B*b)*d^3)*tan(f*x + e))/b^3)/f
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1329 vs. 2(571) = 1142.

Time = 1.13 (sec) , antiderivative size = 1329, normalized size of antiderivative = 2.32

$$\int \frac{(c+d\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx = \text{Too large to display}$$

```
[In] integrate((c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e
))^2,x, algorithm="giac")
```

```
[Out] 1/2*(2*(A*a^2*c^3 - C*a^2*c^3 + 2*B*a*b*c^3 - A*b^2*c^3 + C*b^2*c^3 - 3*B*a
^2*c^2*d + 6*A*a*b*c^2*d - 6*C*a*b*c^2*d + 3*B*b^2*c^2*d - 3*A*a^2*c*d^2 +
3*C*a^2*c*d^2 - 6*B*a*b*c*d^2 + 3*A*b^2*c*d^2 - 3*C*b^2*c*d^2 + B*a^2*d^3 -
 2*A*a*b*d^3 + 2*C*a*b*d^3 - B*b^2*d^3)*(f*x + e)/(a^4 + 2*a^2*b^2 + b^4) +
 (B*a^2*c^3 - 2*A*a*b*c^3 + 2*C*a*b*c^3 - B*b^2*c^3 + 3*A*a^2*c^2*d - 3*C*a
^2*c^2*d + 6*B*a*b*c^2*d - 3*A*b^2*c^2*d + 3*C*b^2*c^2*d - 3*B*a^2*c*d^2 +
6*A*a*b*c*d^2 - 6*C*a*b*c*d^2 + 3*B*b^2*c*d^2 - A*a^2*d^3 + C*a^2*d^3 - 2*B
*a*b*d^3 + A*b^2*d^3 - C*b^2*d^3)*log(tan(f*x + e)^2 + 1)/(a^4 + 2*a^2*b^2
+ b^4) - 2*(B*a^2*b^4*c^3 - 2*A*a*b^5*c^3 + 2*C*a*b^5*c^3 - B*b^6*c^3 - 3*C
*a^4*b^2*c^2*d + 3*A*a^2*b^4*c^2*d - 9*C*a^2*b^4*c^2*d + 6*B*a*b^5*c^2*d -
3*A*b^6*c^2*d + 6*C*a^5*b*c*d^2 - 3*B*a^4*b^2*c*d^2 + 12*C*a^3*b^3*c*d^2 -
9*B*a^2*b^4*c*d^2 + 6*A*a*b^5*c*d^2 - 3*C*a^6*d^3 + 2*B*a^5*b*d^3 - A*a^4*b
^2*d^3 - 5*C*a^4*b^2*d^3 + 4*B*a^3*b^3*d^3 - 3*A*a^2*b^4*d^3)*log(abs(b*tan
(f*x + e) + a))/(a^4*b^4 + 2*a^2*b^6 + b^8) + 2*(B*a^2*b^5*c^3*tan(f*x + e)
 - 2*A*a*b^6*c^3*tan(f*x + e) + 2*C*a*b^6*c^3*tan(f*x + e) - B*b^7*c^3*tan(
f*x + e) - 3*C*a^4*b^3*c^2*d*tan(f*x + e) + 3*A*a^2*b^5*c^2*d*tan(f*x + e)
- 9*C*a^2*b^5*c^2*d*tan(f*x + e) + 6*B*a*b^6*c^2*d*tan(f*x + e) - 3*A*b^7*c
^2*d*tan(f*x + e) + 6*C*a^5*b^2*c*d^2*tan(f*x + e) - 3*B*a^4*b^3*c*d^2*tan(
f*x + e) + 12*C*a^3*b^4*c*d^2*tan(f*x + e) - 9*B*a^2*b^5*c*d^2*tan(f*x + e)
 + 6*A*a*b^6*c*d^2*tan(f*x + e) - 3*C*a^6*b*d^3*tan(f*x + e) + 2*B*a^5*b^2*
d^3*tan(f*x + e) - A*a^4*b^3*d^3*tan(f*x + e) - 5*C*a^4*b^3*d^3*tan(f*x + e
) + 4*B*a^3*b^4*d^3*tan(f*x + e) - 3*A*a^2*b^5*d^3*tan(f*x + e) - C*a^4*b^3
*c^3 + 2*B*a^3*b^4*c^3 - 3*A*a^2*b^5*c^3 + C*a^2*b^5*c^3 - A*b^7*c^3 - 3*B*
a^4*b^3*c^2*d + 6*A*a^3*b^4*c^2*d - 6*C*a^3*b^4*c^2*d + 3*B*a^2*b^5*c^2*d +
 3*C*a^6*b*c*d^2 - 3*A*a^4*b^3*c*d^2 + 9*C*a^4*b^3*c*d^2 - 6*B*a^3*b^4*c*d^
2 + 3*A*a^2*b^5*c*d^2 - 2*C*a^7*d^3 + B*a^6*b*d^3 - 4*C*a^5*b^2*d^3 + 3*B*a
^4*b^3*d^3 - 2*A*a^3*b^4*d^3)/((a^4*b^4 + 2*a^2*b^6 + b^8)*(b*tan(f*x + e)
+ a)) + (C*b^2*d^3*tan(f*x + e)^2 + 6*C*b^2*c*d^2*tan(f*x + e) - 4*C*a*b*d^
3*tan(f*x + e) + 2*B*b^2*d^3*tan(f*x + e))/b^4)/f
```

Mupad [B] (verification not implemented)

Time = 15.06 (sec) , antiderivative size = 701, normalized size of antiderivative = 1.22

$$\int \frac{(c+d\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx$$

$$= \frac{\tan(e+f\,x)\,\left(\frac{B\,d^3+3\,C\,c\,d^2}{b^2}-\frac{2\,C\,a\,d^3}{b^3}\right)}{f}$$

$$-\frac{\ln\left(\tan(e+f\,x)+1\mathrm{i}\right)\,\left(B\,c^3-A\,d^3+C\,d^3+3\,A\,c^2\,d-3\,B\,c\,d^2-3\,C\,c^2\,d+A\,c^3\,1\mathrm{i}+B\,d^3\,1\mathrm{i}-C\,c^3\,1\mathrm{i}-\right.}{2\,f\,\left(-a^2+a\,b\,2\mathrm{i}+b^2\right)}$$

$$+\frac{\ln\left(a+b\tan(e+f\,x)\right)\,\left(b^4\,(3\,A\,a^2\,d^3-B\,a^2\,c^3-3\,A\,a^2\,c^2\,d+9\,B\,a^2\,c\,d^2+9\,C\,a^2\,c^2\,d)-b^5\,(2\,C\,a\,c^3-2\,\right.}{}$$

$$-\frac{\ln\left(\tan(e+f\,x)-\mathrm{i}\right)\,\left(A\,c^3-A\,d^3\,1\mathrm{i}+B\,c^3\,1\mathrm{i}+B\,d^3-C\,c^3+C\,d^3\,1\mathrm{i}-3\,A\,c\,d^2+A\,c^2\,d\,3\mathrm{i}-B\,c\,d^2\,3\mathrm{i}-\right.}{2\,f\,\left(-a^2\,1\mathrm{i}+2\,a\,b+b^2\,1\mathrm{i}\right)}$$

$$-\frac{-C\,a^5\,d^3+3\,C\,a^4\,b\,c\,d^2+B\,a^4\,b\,d^3-3\,C\,a^3\,b^2\,c^2\,d-3\,B\,a^3\,b^2\,c\,d^2-A\,a^3\,b^2\,d^3+C\,a^2\,b^3\,c^3+3\,B\,a^2\,b^3\,c^2\,}{b\,f\,\left(\tan\left(e+f\,x\right)\,b^4+a\,b^3\right)\,\left(a^2+b^2\right)}$$

$$+\frac{C\,d^3\tan(e+f\,x)^2}{2\,b^2\,f}$$

```
[In] int(((c + d*tan(e + f*x))^3*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a + b
*tan(e + f*x))^2,x)
```

```
[Out] (tan(e + f*x)*((B*d^3 + 3*C*c*d^2)/b^2 - (2*C*a*d^3)/b^3))/f - (log(tan(e +
 f*x) + 1i)*(A*c^3*1i - A*d^3 + B*c^3 + B*d^3*1i - C*c^3*1i + C*d^3 - A*c*d
^2*3i + 3*A*c^2*d - 3*B*c*d^2 - B*c^2*d*3i + C*c*d^2*3i - 3*C*c^2*d))/(2*f*
(a*b*2i - a^2 + b^2)) + (log(a + b*tan(e + f*x))*(b^4*(3*A*a^2*d^3 - B*a^2*
c^3 - 3*A*a^2*c^2*d + 9*B*a^2*c*d^2 + 9*C*a^2*c^2*d) - b^5*(2*C*a*c^3 - 2*A
*a*c^3 + 6*A*a*c*d^2 + 6*B*a*c^2*d) - b^3*(4*B*a^3*d^3 + 12*C*a^3*c*d^2) +
b^6*(B*c^3 + 3*A*c^2*d) - b*(2*B*a^5*d^3 + 6*C*a^5*c*d^2) + b^2*(A*a^4*d^3
+ 5*C*a^4*d^3 + 3*B*a^4*c*d^2 + 3*C*a^4*c^2*d) + 3*C*a^6*d^3))/(f*(b^8 + 2*
a^2*b^6 + a^4*b^4)) - (log(tan(e + f*x) - 1i)*(A*c^3 - A*d^3*1i + B*c^3*1i
+ B*d^3 - C*c^3 + C*d^3*1i - 3*A*c*d^2 + A*c^2*d*3i - B*c*d^2*3i - 3*B*c^2*
d + 3*C*c*d^2 - C*c^2*d*3i))/(2*f*(2*a*b - a^2*1i + b^2*1i)) - (A*b^5*c^3 -
 C*a^5*d^3 - B*a*b^4*c^3 + B*a^4*b*d^3 - A*a^3*b^2*d^3 + C*a^2*b^3*c^3 + 3*
A*a^2*b^3*c*d^2 + 3*B*a^2*b^3*c^2*d - 3*B*a^3*b^2*c*d^2 - 3*C*a^3*b^2*c^2*d
 - 3*A*a*b^4*c^2*d + 3*C*a^4*b*c*d^2)/(b*f*(a*b^3 + b^4*tan(e + f*x))*(a^2
+ b^2)) + (C*d^3*tan(e + f*x)^2)/(2*b^2*f)
```

3.69 $\int \frac{(c+d\tan(e+fx))^3(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx$

Optimal result 659
Rubi [A] (verified) 660
Mathematica [C] (verified) 664
Maple [A] (verified) 665
Fricas [B] (verification not implemented) 666
Sympy [F(-2)] 667
Maxima [A] (verification not implemented) 667
Giac [B] (verification not implemented) 668
Mupad [B] (verification not implemented) 670

Optimal result

Integrand size = 45, antiderivative size = 798

$$\int \frac{(c+d\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx =$$

$$-\frac{(3ab^2(Ac^3-c^3C-3Bc^2d-3Acd^2+3cCd^2+Bd^3)+a^3(c^3C+3Bc^2d-3cCd^2-Bd^3-A(c^3-3cd^2}{(a^2+b^2)^3}$$

$$-\frac{(b^3(Ac^3-c^3C-3Bc^2d-3Acd^2+3cCd^2+Bd^3)+3a^2b(c^3C+3Bc^2d-3cCd^2-Bd^3-A(c^3-3cd^2)}{(a^2}$$

$$-\frac{(bc-ad)\,(a^5bBd^2-3a^6Cd^2+a^4b^2d(Bc-9Cd)+a^3b^3B(c^2+3d^2)-b^6(c(cC+3Bd)-A(c^2-3d^2))}{b^4\,(a^2}$$

$$-\frac{d^2(a^3bBd-3a^4Cd-ab^3(2Ac-2cC-3Bd)+a^2b^2(Bc-6Cd)-b^4(Bc+(2A+C)d))\tan(e+fx)}{b^3\,(a^2+b^2)^2\,f}$$

$$+\frac{(a^3bBd-3a^4Cd-b^4(2Bc+3Ad)-ab^3(4Ac-4cC-5Bd)+a^2b^2(2Bc+(A-7C)d))\,(c+d\tan(e+}{2b^2\,(a^2+b^2)^2\,f(a+b\tan(e+fx))}$$

$$-\frac{(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^3}{2b\,(a^2+b^2)\,f(a+b\tan(e+fx))^2}$$

```
[Out] -(3*a*b^2*(A*c^3-3*A*c*d^2-3*B*c^2*d+B*d^3-C*c^3+3*C*c*d^2)+a^3*(c^3*C+3*B*
c^2*d-3*C*c*d^2-B*d^3-A*(c^3-3*c*d^2))-3*a^2*b*((A-C)*d*(3*c^2-d^2)+B*(c^3-
3*c*d^2))+b^3*((A-C)*d*(3*c^2-d^2)+B*(c^3-3*c*d^2)))*x/(a^2+b^2)^3-(b^3*(A*
c^3-3*A*c*d^2-3*B*c^2*d+B*d^3-C*c^3+3*C*c*d^2)+3*a^2*b*(c^3*C+3*B*c^2*d-3*C
*c*d^2-B*d^3-A*(c^3-3*c*d^2))+a^3*((A-C)*d*(3*c^2-d^2)+B*(c^3-3*c*d^2))-3*a
*b^2*((A-C)*d*(3*c^2-d^2)+B*(c^3-3*c*d^2)))*ln(cos(f*x+e))/(a^2+b^2)^3/f-(-
a*d+b*c)*(a^5*b*B*d^2-3*a^6*C*d^2+a^4*b^2*d*(B*c-9*C*d)+a^3*b^3*B*(c^2+3*d^
2)-b^6*(c*(3*B*d+C*c)-A*(c^2-3*d^2))-a*b^5*(8*c*(A-C)*d+3*B*(c^2-2*d^2))+a^
2*b^4*(3*c^2*C+6*B*c*d-10*C*d^2-A*(3*c^2-d^2)))*ln(a+b*tan(f*x+e))/b^4/(a^2
+b^2)^3/f-d^2*(a^3*b*B*d-3*a^4*C*d-a*b^3*(2*A*c-3*B*d-2*C*c)+a^2*b^2*(B*c-6
```

```
*C*d)-b^4*(B*c+(2*A+C)*d))*tan(f*x+e)/b^3/(a^2+b^2)^2/f+1/2*(a^3*b*B*d-3*a^
4*C*d-b^4*(3*A*d+2*B*c)-a*b^3*(4*A*c-5*B*d-4*C*c)+a^2*b^2*(2*B*c+(A-7*C)*d)
)*(c+d*tan(f*x+e))^2/b^2/(a^2+b^2)^2/f/(a+b*tan(f*x+e))-1/2*(A*b^2-a*(B*b-C
*a))*(c+d*tan(f*x+e))^3/b/(a^2+b^2)/f/(a+b*tan(f*x+e))^2
```

Rubi [A] (verified)

Time = 2.96 (sec) , antiderivative size = 798, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}} = 0.133$, Rules used = {3726, 3718, 3707, 3698, 31, 3556}

$$\int \frac{(c+d\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx$$

$$= -\frac{(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^3}{2b\,(a^2+b^2)\,f(a+b\tan(e+fx))^2}$$

$$+\frac{(-3Cda^4+bBda^3+b^2(2Bc+(A-7C)d)a^2-b^3(4Ac-4Cc-5Bd)a-b^4(2Bc+3Ad))\,(c+d\tan(e+}{2b^2\,(a^2+b^2)^2\,f(a+b\tan(e+fx))}$$

$$-\frac{((Cc^3+3Bdc^2-3Cd^2c-Bd^3-A(c^3-3cd^2))\,a^3-3b((A-C)d(3c^2-d^2)+B(c^3-3cd^2))\,a^2+3b^2(A}{(a^2+b^2)^3}$$

$$-\frac{(((A-C)d(3c^2-d^2)+B(c^3-3cd^2))\,a^3+3b(Cc^3+3Bdc^2-3Cd^2c-Bd^3-A(c^3-3cd^2))\,a^2-3b^2((}{(a^2+}$$

$$-\frac{(bc-ad)\,(-3Cd^2a^6+bBd^2a^5+b^2d(Bc-9Cd)a^4+b^3B(c^2+3d^2)\,a^3+b^4(3Cc^2+6Bdc-10Cd^2-A(}{b^4\,(a^2}$$

$$-\frac{d^2(-3Cda^4+bBda^3+b^2(Bc-6Cd)a^2-b^3(2Ac-2Cc-3Bd)a-b^4(Bc+(2A+C)d))\tan(e+fx)}{b^3\,(a^2+b^2)^2\,f}$$

```
[In] Int[((c + d*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a + b
*Tan[e + f*x])^3,x]
```

```
[Out] -(((3*a*b^2*(A*c^3 - c^3*C - 3*B*c^2*d - 3*A*c*d^2 + 3*c*C*d^2 + B*d^3) + a
^3*(c^3*C + 3*B*c^2*d - 3*c*C*d^2 - B*d^3 - A*(c^3 - 3*c*d^2)) - 3*a^2*b*((
A - C)*d*(3*c^2 - d^2) + B*(c^3 - 3*c*d^2)) + b^3*((A - C)*d*(3*c^2 - d^2)
+ B*(c^3 - 3*c*d^2)))*x)/(a^2 + b^2)^3) - ((b^3*(A*c^3 - c^3*C - 3*B*c^2*d
- 3*A*c*d^2 + 3*c*C*d^2 + B*d^3) + 3*a^2*b*(c^3*C + 3*B*c^2*d - 3*c*C*d^2 -
 B*d^3 - A*(c^3 - 3*c*d^2)) + a^3*((A - C)*d*(3*c^2 - d^2) + B*(c^3 - 3*c*d
^2)) - 3*a*b^2*((A - C)*d*(3*c^2 - d^2) + B*(c^3 - 3*c*d^2)))*Log[Cos[e + f
*x]])/((a^2 + b^2)^3*f) - ((b*c - a*d)*(a^5*b*B*d^2 - 3*a^6*C*d^2 + a^4*b^2
*d*(B*c - 9*C*d) + a^3*b^3*B*(c^2 + 3*d^2) - b^6*(c*(c*C + 3*B*d) - A*(c^2
- 3*d^2)) - a*b^5*(8*c*(A - C)*d + 3*B*(c^2 - 2*d^2)) + a^2*b^4*(3*c^2*C +
6*B*c*d - 10*C*d^2 - A*(3*c^2 - d^2)))*Log[a + b*Tan[e + f*x]])/(b^4*(a^2 +
 b^2)^3*f) - (d^2*(a^3*b*B*d - 3*a^4*C*d - a*b^3*(2*A*c - 2*c*C - 3*B*d) +
a^2*b^2*(B*c - 6*C*d) - b^4*(B*c + (2*A + C)*d))*Tan[e + f*x])/(b^3*(a^2 +
b^2)^2*f) + ((a^3*b*B*d - 3*a^4*C*d - b^4*(2*B*c + 3*A*d) - a*b^3*(4*A*c -
```

```
4*c*C - 5*B*d) + a^2*b^2*(2*B*c + (A - 7*C)*d))*(c + d*Tan[e + f*x])^2)/(2*
b^2*(a^2 + b^2)^2*f*(a + b*Tan[e + f*x])) - ((A*b^2 - a*(b*B - a*C))*(c + d
*Tan[e + f*x])^3)/(2*b*(a^2 + b^2)*f*(a + b*Tan[e + f*x])^2)
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3698

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3707

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[(a*A + b*B - a
*C)*(x/(a^2 + b^2)), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3726

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*d^2 + c*(c*C - B*d))*(a + b*Tan[e +
 f*x])^m*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dis
```

```
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rubi steps

$$\text{integral} = -\frac{(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^3}{2b\,(a^2+b^2)\,f(a+b\tan(e+fx))^2} + \frac{\int \frac{(c+d\tan(e+fx))^2((bB-aC)(2bc-3ad)+Ab(2ac+3bd)-2b((A-C)(bc-ad)-B(ac+bd))\tan(e+fx)+(Ab^2-abB+3a^2C+2b^2C)d\tan^2(e+fx)}{(a+b\tan(e+fx))^2}}{2b\,(a^2+b^2)}$$

$$= \frac{(a^3bBd-3a^4Cd-b^4(2Bc+3Ad)-ab^3(4Ac-4cC-5Bd)+a^2b^2(2Bc+(A-7C)d))\,(c+d\tan}{2b^2\,(a^2+b^2)^2\,f(a+b\tan(e+fx))} - \frac{(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^3}{2b\,(a^2+b^2)\,f(a+b\tan(e+fx))^2} + \frac{\int \frac{(c+d\tan(e+fx))(b(ac+2bd)((bB-aC)(2bc-3ad)+Ab(2ac+3bd))+(bc-2ad)(a^2bBd-3a^3Cd-Ab^2(2bc-ad)+2b^3(cC+Bd)+2ab^2(}{}}{}$$

$$= -\frac{d^2(a^3bBd-3a^4Cd-ab^3(2Ac-2cC-3Bd)+a^2b^2(Bc-6Cd)-b^4(Bc+(2A+C)d))\tan(e+}{b^3\,(a^2+b^2)^2\,f} + \frac{(a^3bBd-3a^4Cd-b^4(2Bc+3Ad)-ab^3(4Ac-4cC-5Bd)+a^2b^2(2Bc+(A-7C)d))\,(c+d\tan}{2b^2\,(a^2+b^2)^2\,f(a+b\tan(e+fx))} - \frac{(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^3}{2b\,(a^2+b^2)\,f(a+b\tan(e+fx))^2} - \frac{\int \frac{2(3a^5Cd^3+6a^3b^2Cd^3-a^4bd^2(3cC+Bd)-a^2b^3(Ac^3-c^3C-3Bc^2d-3Acd^2+9cCd^2+3Bd^3)-ab^4(2Bc^3+6Ac^2d-6c^2Cd-6Bcd^2-2}{}}{}$$

$$=$$

$$-\frac{(3ab^2(Ac^3-c^3C-3Bc^2d-3Acd^2+3cCd^2+Bd^3)+a^3(c^3C+3Bc^2d-3cCd^2-Bd^3-A(c^3}{(a^2}$$

$$-\frac{d^2(a^3bBd-3a^4Cd-ab^3(2Ac-2cC-3Bd)+a^2b^2(Bc-6Cd)-b^4(Bc+(2A+C)d))\tan(e}{b^3\left(a^2+b^2\right)^2f}$$

$$+\frac{(a^3bBd-3a^4Cd-b^4(2Bc+3Ad)-ab^3(4Ac-4cC-5Bd)+a^2b^2(2Bc+(A-7C)d))\,(c+d}{2b^2\left(a^2+b^2\right)^2f(a+b\tan(e+fx))}$$

$$-\frac{(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^3}{2b\,(a^2+b^2)\,f(a+b\tan(e+fx))^2}$$

$$-\frac{((bc-ad)\,(a^5bBd^2-3a^6Cd^2+a^4b^2d(Bc-9Cd)+a^3b^3B(c^2+3d^2)-b^6(c(cC+3Bd)-A(c^2}{}$$

$$+\frac{\left(2b(a^2+b^2)^2\,d^2(3bcC+bBd-3aCd)+2b(3a^5Cd^3+6a^3b^2Cd^3-a^4bd^2(3cC+Bd)-a^2b^3(Ac}{}$$

$$=$$

$$-\frac{(3ab^2(Ac^3-c^3C-3Bc^2d-3Acd^2+3cCd^2+Bd^3)+a^3(c^3C+3Bc^2d-3cCd^2-Bd^3-A(c^3}{(a^2}$$

$$+\frac{(3a^2b(Ac^3-c^3C-3Bc^2d-3Acd^2+3cCd^2+Bd^3)+b^3(c^3C+3Bc^2d-3cCd^2-Bd^3-A(c^3}{}$$

$$-\frac{d^2(a^3bBd-3a^4Cd-ab^3(2Ac-2cC-3Bd)+a^2b^2(Bc-6Cd)-b^4(Bc+(2A+C)d))\tan(e}{b^3\left(a^2+b^2\right)^2f}$$

$$+\frac{(a^3bBd-3a^4Cd-b^4(2Bc+3Ad)-ab^3(4Ac-4cC-5Bd)+a^2b^2(2Bc+(A-7C)d))\,(c+d}{2b^2\left(a^2+b^2\right)^2f(a+b\tan(e+fx))}$$

$$-\frac{(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^3}{2b\,(a^2+b^2)\,f(a+b\tan(e+fx))^2}$$

$$-\frac{((bc-ad)\,(a^5bBd^2-3a^6Cd^2+a^4b^2d(Bc-9Cd)+a^3b^3B(c^2+3d^2)-b^6(c(cC+3Bd)-A(c^2}{}$$

$$=$$

$$-\frac{(3ab^2(Ac^3-c^3C-3Bc^2d-3Acd^2+3cCd^2+Bd^3)+a^3(c^3C+3Bc^2d-3cCd^2-Bd^3-A(c^3}{(a^2}$$

$$+\frac{(3a^2b(Ac^3-c^3C-3Bc^2d-3Acd^2+3cCd^2+Bd^3)+b^3(c^3C+3Bc^2d-3cCd^2-Bd^3-A(c^3}{}$$

$$-\frac{(bc-ad)\,(a^5bBd^2-3a^6Cd^2+a^4b^2d(Bc-9Cd)+a^3b^3B(c^2+3d^2)-b^6(c(cC+3Bd)-A(c^2+}{}$$

$$-\frac{d^2(a^3bBd-3a^4Cd-ab^3(2Ac-2cC-3Bd)+a^2b^2(Bc-6Cd)-b^4(Bc+(2A+C)d))\tan(e}{b^3\left(a^2+b^2\right)^2f}$$

$$+\frac{(a^3bBd-3a^4Cd-b^4(2Bc+3Ad)-ab^3(4Ac-4cC-5Bd)+a^2b^2(2Bc+(A-7C)d))\,(c+d}{2b^2\left(a^2+b^2\right)^2f(a+b\tan(e+fx))}$$

$$-\frac{(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^3}{2b\,(a^2+b^2)\,f(a+b\tan(e+fx))^2}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 7.56 (sec) , antiderivative size = 1409, normalized size of antiderivative = 1.77

$$\int \frac{(c+d\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx = \frac{C(c+d\tan(e+fx))^3}{bf(a+b\tan(e+fx))^2}$$
$$+\frac{b(-3a^2Abc^3+Ab^3c^3+a^3Bc^3-3ab^2Bc^3+3a^2bc^3C-b^3c^3C+3a^3Ac^2d-9aAb^2c^2d+9a^2bBc^2d-3b^3Bc^2d-3a^3c^2Cd+9ab^2c^2Cd+9a^2Abcd^2-3Ab^3c}{}$$

```
[In] Integrate[((c + d*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(a + b*Tan[e + f*x])^3,x]
```

```
[Out] (C*(c + d*Tan[e + f*x])^3)/(b*f*(a + b*Tan[e + f*x])^2) + ((b*(-3*a^2*A*b*c
^3 + A*b^3*c^3 + a^3*B*c^3 - 3*a*b^2*B*c^3 + 3*a^2*b*c^3*C - b^3*c^3*C + 3*
a^3*A*c^2*d - 9*a*A*b^2*c^2*d + 9*a^2*b*B*c^2*d - 3*b^3*B*c^2*d - 3*a^3*c^2
*C*d + 9*a*b^2*c^2*C*d + 9*a^2*A*b*c*d^2 - 3*A*b^3*c*d^2 - 3*a^3*B*c*d^2 +
9*a*b^2*B*c*d^2 - 9*a^2*b*c*C*d^2 + 3*b^3*c*C*d^2 - a^3*A*d^3 + 3*a*A*b^2*d
^3 - 3*a^2*b*B*d^3 + b^3*B*d^3 + a^3*C*d^3 - 3*a*b^2*C*d^3 + I*(-(a^3*A*c^3
) + 3*a*A*b^2*c^3 - 3*a^2*b*B*c^3 + b^3*B*c^3 + a^3*c^3*C - 3*a*b^2*c^3*C -
 9*a^2*A*b*c^2*d + 3*A*b^3*c^2*d + 3*a^3*B*c^2*d - 9*a*b^2*B*c^2*d + 9*a^2*
b*c^2*C*d - 3*b^3*c^2*C*d + 3*a^3*A*c*d^2 - 9*a*A*b^2*c*d^2 + 9*a^2*b*B*c*d
^2 - 3*b^3*B*c*d^2 - 3*a^3*c*C*d^2 + 9*a*b^2*c*C*d^2 + 3*a^2*A*b*d^3 - A*b^
3*d^3 - a^3*B*d^3 + 3*a*b^2*B*d^3 - 3*a^2*b*C*d^3 + b^3*C*d^3))*Log[I - Tan
[e + f*x]])/(2*(a^2 + b^2)^3*f) - (b*(3*a^2*A*b*c^3 - A*b^3*c^3 - a^3*B*c^3
 + 3*a*b^2*B*c^3 - 3*a^2*b*c^3*C + b^3*c^3*C - 3*a^3*A*c^2*d + 9*a*A*b^2*c^
2*d - 9*a^2*b*B*c^2*d + 3*b^3*B*c^2*d + 3*a^3*c^2*C*d - 9*a*b^2*c^2*C*d - 9
*a^2*A*b*c*d^2 + 3*A*b^3*c*d^2 + 3*a^3*B*c*d^2 - 9*a*b^2*B*c*d^2 + 9*a^2*b*
c*C*d^2 - 3*b^3*c*C*d^2 + a^3*A*d^3 - 3*a*A*b^2*d^3 + 3*a^2*b*B*d^3 - b^3*B
*d^3 - a^3*C*d^3 + 3*a*b^2*C*d^3 + I*(-(a^3*A*c^3) + 3*a*A*b^2*c^3 - 3*a^2*
b*B*c^3 + b^3*B*c^3 + a^3*c^3*C - 3*a*b^2*c^3*C - 9*a^2*A*b*c^2*d + 3*A*b^3
*c^2*d + 3*a^3*B*c^2*d - 9*a*b^2*B*c^2*d + 9*a^2*b*c^2*C*d - 3*b^3*c^2*C*d
+ 3*a^3*A*c*d^2 - 9*a*A*b^2*c*d^2 + 9*a^2*b*B*c*d^2 - 3*b^3*B*c*d^2 - 3*a^3
*c*C*d^2 + 9*a*b^2*c*C*d^2 + 3*a^2*A*b*d^3 - A*b^3*d^3 - a^3*B*d^3 + 3*a*b^
2*B*d^3 - 3*a^2*b*C*d^3 + b^3*C*d^3))*Log[I + Tan[e + f*x]])/(2*(a^2 + b^2)
^3*f) - ((b*c - a*d)*(a^5*b*B*d^2 - 3*a^6*C*d^2 + a^4*b^2*d*(B*c - 9*C*d) +
 a^3*b^3*B*(c^2 + 3*d^2) - b^6*(c*(c*C + 3*B*d) - A*(c^2 - 3*d^2)) - a*b^5*
(8*c*(A - C)*d + 3*B*(c^2 - 2*d^2)) + a^2*b^4*(3*c^2*C + 6*B*c*d - 10*C*d^2
 - A*(3*c^2 - d^2)))*Log[a + b*Tan[e + f*x]])/(b^3*(a^2 + b^2)^3*f) - ((A*b
^2 - a*b*B + 3*a^2*C + 2*b^2*C)*(b*c - a*d)^3)/(2*b^3*(a^2 + b^2)*f*(a + b*
Tan[e + f*x])^2) + ((b*c - a*d)^2*(2*a^3*b*B*d - 6*a^4*C*d - 2*a*b^3*(A*c -
 c*C - 2*B*d) - b^4*(B*c + 3*(A + C)*d) + a^2*b^2*(B*c - (A + 11*C)*d)))/(b
^3*(a^2 + b^2)^2*f*(a + b*Tan[e + f*x])))/b
```

Maple [A] (verified)

Time = 0.50 (sec) , antiderivative size = 1271, normalized size of antiderivative = 1.59

method	result	size
derivativedivides	Expression too large to display	1271
default	Expression too large to display	1271
norman	Expression too large to display	2076
parallelrisch	Expression too large to display	6687
risch	Expression too large to display	6825

```
[In] int((c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^3,x
,method=_RETURNVERBOSE)
```

```
[Out] 1/f*(tan(f*x+e)*C*d^3/b^3-1/2*(-A*a^3*b^2*d^3+3*A*a^2*b^3*c*d^2-3*A*a*b^4*c
^2*d+A*b^5*c^3+B*a^4*b*d^3-3*B*a^3*b^2*c*d^2+3*B*a^2*b^3*c^2*d-B*a*b^4*c^3-
C*a^5*d^3+3*C*a^4*b*c*d^2-3*C*a^3*b^2*c^2*d+C*a^2*b^3*c^3)/b^4/(a^2+b^2)/(a
+b*tan(f*x+e))^2-1/b^4*(A*a^4*b^2*d^3-3*A*a^2*b^4*c^2*d+3*A*a^2*b^4*d^3+2*A
*a*b^5*c^3-6*A*a*b^5*c*d^2+3*A*b^6*c^2*d-2*B*a^5*b*d^3+3*B*a^4*b^2*c*d^2-4*
B*a^3*b^3*d^3-B*a^2*b^4*c^3+9*B*a^2*b^4*c*d^2-6*B*a*b^5*c^2*d+B*b^6*c^3+3*C
*a^6*d^3-6*C*a^5*b*c*d^2+3*C*a^4*b^2*c^2*d+5*C*a^4*b^2*d^3-12*C*a^3*b^3*c*d
^2+9*C*a^2*b^4*c^2*d-2*C*a*b^5*c^3)/(a^2+b^2)^2/(a+b*tan(f*x+e))+(-3*A*a^3*
b^4*c^2*d+A*a^3*b^4*d^3+3*A*a^2*b^5*c^3-9*A*a^2*b^5*c*d^2+9*A*a*b^6*c^2*d-3
*A*a*b^6*d^3-A*b^7*c^3+3*A*b^7*c*d^2+B*a^6*b*d^3+3*B*a^4*b^3*d^3-B*a^3*b^4*
c^3+3*B*a^3*b^4*c*d^2-9*B*a^2*b^5*c^2*d+6*B*a^2*b^5*d^3+3*B*a*b^6*c^3-9*B*a
*b^6*c*d^2+3*B*b^7*c^2*d-3*C*a^7*d^3+3*C*a^6*b*c*d^2-9*C*a^5*b^2*d^3+9*C*a^
4*b^3*c*d^2+3*C*a^3*b^4*c^2*d-10*C*a^3*b^4*d^3-3*C*a^2*b^5*c^3+18*C*a^2*b^5
*c*d^2-9*C*a*b^6*c^2*d+C*b^7*c^3)/b^4/(a^2+b^2)^3*ln(a+b*tan(f*x+e))+1/(a^2
+b^2)^3*(1/2*(3*A*a^3*c^2*d-A*a^3*d^3-3*A*a^2*b*c^3+9*A*a^2*b*c*d^2-9*A*a*b
^2*c^2*d+3*A*a*b^2*d^3+A*b^3*c^3-3*A*b^3*c*d^2+B*a^3*c^3-3*B*a^3*c*d^2+9*B*
a^2*b*c^2*d-3*B*a^2*b*d^3-3*B*a*b^2*c^3+9*B*a*b^2*c*d^2-3*B*b^3*c^2*d+B*b^3
*d^3-3*C*a^3*c^2*d+C*a^3*d^3+3*C*a^2*b*c^3-9*C*a^2*b*c*d^2+9*C*a*b^2*c^2*d-
3*C*a*b^2*d^3-C*b^3*c^3+3*C*b^3*c*d^2)*ln(1+tan(f*x+e)^2)+(A*a^3*c^3-3*A*a^
3*c*d^2+9*A*a^2*b*c^2*d-3*A*a^2*b*d^3-3*A*a*b^2*c^3+9*A*a*b^2*c*d^2-3*A*b^3
*c^2*d+A*b^3*d^3-3*B*a^3*c^2*d+B*a^3*d^3+3*B*a^2*b*c^3-9*B*a^2*b*c*d^2+9*B*
a*b^2*c^2*d-3*B*a*b^2*d^3-B*b^3*c^3+3*B*b^3*c*d^2-C*a^3*c^3+3*C*a^3*c*d^2-9
*C*a^2*b*c^2*d+3*C*a^2*b*d^3+3*C*a*b^2*c^3-9*C*a*b^2*c*d^2+3*C*b^3*c^2*d-C*
b^3*d^3)*arctan(tan(f*x+e))))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 2549 vs. $2(794) = 1588$.

Time = 1.44 (sec) , antiderivative size = 2549, normalized size of antiderivative = 3.19

$$\int \frac{(c+d\tan(e+fx))^3 (A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx = \text{Too large to display}$$

```
[In] integrate((c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e
))^3,x, algorithm="fricas")
```

```
[Out] 1/2*(2*(C*a^6*b^3 + 3*C*a^4*b^5 + 3*C*a^2*b^7 + C*b^9)*d^3*tan(f*x + e)^3 -
 (3*C*a^4*b^5 - 5*B*a^3*b^6 + (7*A - 3*C)*a^2*b^7 + B*a*b^8 + A*b^9)*c^3 +
3*(C*a^5*b^4 - 3*B*a^4*b^5 + 5*(A - C)*a^3*b^6 + 3*B*a^2*b^7 - A*a*b^8)*c^2
*d + 3*(C*a^6*b^3 + B*a^5*b^4 - (3*A - 7*C)*a^4*b^5 - 5*B*a^3*b^6 + 3*A*a^2
*b^7)*c*d^2 - (3*C*a^7*b^2 - B*a^6*b^3 - (A - 9*C)*a^5*b^4 - 7*B*a^4*b^5 +
5*A*a^3*b^6)*d^3 + 2*(((A - C)*a^5*b^4 + 3*B*a^4*b^5 - 3*(A - C)*a^3*b^6 -
B*a^2*b^7)*c^3 - 3*(B*a^5*b^4 - 3*(A - C)*a^4*b^5 - 3*B*a^3*b^6 + (A - C)*a
^2*b^7)*c^2*d - 3*((A - C)*a^5*b^4 + 3*B*a^4*b^5 - 3*(A - C)*a^3*b^6 - B*a^
2*b^7)*c*d^2 + (B*a^5*b^4 - 3*(A - C)*a^4*b^5 - 3*B*a^3*b^6 + (A - C)*a^2*b
^7)*d^3)*f*x + ((C*a^4*b^5 - 3*B*a^3*b^6 + 5*(A - C)*a^2*b^7 + 3*B*a*b^8 -
A*b^9)*c^3 + 3*(C*a^5*b^4 + B*a^4*b^5 - (3*A - 7*C)*a^3*b^6 - 5*B*a^2*b^7 +
 3*A*a*b^8)*c^2*d - 3*(3*C*a^6*b^3 - B*a^5*b^4 - (A - 9*C)*a^4*b^5 - 7*B*a^
3*b^6 + 5*A*a^2*b^7)*c*d^2 + (9*C*a^7*b^2 - 3*B*a^6*b^3 + (A + 23*C)*a^5*b^
4 - 9*B*a^4*b^5 + (7*A + 12*C)*a^3*b^6 + 4*C*a*b^8)*d^3 + 2*(((A - C)*a^3*b
^6 + 3*B*a^2*b^7 - 3*(A - C)*a*b^8 - B*b^9)*c^3 - 3*(B*a^3*b^6 - 3*(A - C)*
a^2*b^7 - 3*B*a*b^8 + (A - C)*b^9)*c^2*d - 3*((A - C)*a^3*b^6 + 3*B*a^2*b^7
 - 3*(A - C)*a*b^8 - B*b^9)*c*d^2 + (B*a^3*b^6 - 3*(A - C)*a^2*b^7 - 3*B*a*
b^8 + (A - C)*b^9)*d^3)*f*x)*tan(f*x + e)^2 - ((B*a^5*b^4 - 3*(A - C)*a^4*b
^5 - 3*B*a^3*b^6 + (A - C)*a^2*b^7)*c^3 + 3*((A - C)*a^5*b^4 + 3*B*a^4*b^5
- 3*(A - C)*a^3*b^6 - B*a^2*b^7)*c^2*d - 3*(C*a^8*b + 3*C*a^6*b^3 + B*a^5*b
^4 - 3*(A - 2*C)*a^4*b^5 - 3*B*a^3*b^6 + A*a^2*b^7)*c*d^2 + (3*C*a^9 - B*a^
8*b + 9*C*a^7*b^2 - 3*B*a^6*b^3 - (A - 10*C)*a^5*b^4 - 6*B*a^4*b^5 + 3*A*a^
3*b^6)*d^3 + ((B*a^3*b^6 - 3*(A - C)*a^2*b^7 - 3*B*a*b^8 + (A - C)*b^9)*c^3
 + 3*((A - C)*a^3*b^6 + 3*B*a^2*b^7 - 3*(A - C)*a*b^8 - B*b^9)*c^2*d - 3*(C
*a^6*b^3 + 3*C*a^4*b^5 + B*a^3*b^6 - 3*(A - 2*C)*a^2*b^7 - 3*B*a*b^8 + A*b^
9)*c*d^2 + (3*C*a^7*b^2 - B*a^6*b^3 + 9*C*a^5*b^4 - 3*B*a^4*b^5 - (A - 10*C
)*a^3*b^6 - 6*B*a^2*b^7 + 3*A*a*b^8)*d^3)*tan(f*x + e)^2 + 2*((B*a^4*b^5 -
3*(A - C)*a^3*b^6 - 3*B*a^2*b^7 + (A - C)*a*b^8)*c^3 + 3*((A - C)*a^4*b^5 +
 3*B*a^3*b^6 - 3*(A - C)*a^2*b^7 - B*a*b^8)*c^2*d - 3*(C*a^7*b^2 + 3*C*a^5*
b^4 + B*a^4*b^5 - 3*(A - 2*C)*a^3*b^6 - 3*B*a^2*b^7 + A*a*b^8)*c*d^2 + (3*C
*a^8*b - B*a^7*b^2 + 9*C*a^6*b^3 - 3*B*a^5*b^4 - (A - 10*C)*a^4*b^5 - 6*B*a
^3*b^6 + 3*A*a^2*b^7)*d^3)*tan(f*x + e))*log((b^2*tan(f*x + e)^2 + 2*a*b*ta
n(f*x + e) + a^2)/(tan(f*x + e)^2 + 1)) - (3*(C*a^8*b + 3*C*a^6*b^3 + 3*C*a
^4*b^5 + C*a^2*b^7)*c*d^2 - (3*C*a^9 - B*a^8*b + 9*C*a^7*b^2 - 3*B*a^6*b^3
```

```
+ 9*C*a^5*b^4 - 3*B*a^4*b^5 + 3*C*a^3*b^6 - B*a^2*b^7)*d^3 + (3*(C*a^6*b^3
+ 3*C*a^4*b^5 + 3*C*a^2*b^7 + C*b^9)*c*d^2 - (3*C*a^7*b^2 - B*a^6*b^3 + 9*C
*a^5*b^4 - 3*B*a^4*b^5 + 9*C*a^3*b^6 - 3*B*a^2*b^7 + 3*C*a*b^8 - B*b^9)*d^3
)*tan(f*x + e)^2 + 2*(3*(C*a^7*b^2 + 3*C*a^5*b^4 + 3*C*a^3*b^6 + C*a*b^8)*c
*d^2 - (3*C*a^8*b - B*a^7*b^2 + 9*C*a^6*b^3 - 3*B*a^5*b^4 + 9*C*a^4*b^5 - 3
*B*a^3*b^6 + 3*C*a^2*b^7 - B*a*b^8)*d^3)*tan(f*x + e))*log(1/(tan(f*x + e)^
2 + 1)) + 2*((C*a^5*b^4 - 2*B*a^4*b^5 + 3*(A - C)*a^3*b^6 + 3*B*a^2*b^7 - (
3*A - 2*C)*a*b^8 - B*b^9)*c^3 + 3*(B*a^5*b^4 - (2*A - 3*C)*a^4*b^5 - 3*B*a^
3*b^6 + 3*(A - C)*a^2*b^7 + 2*B*a*b^8 - A*b^9)*c^2*d - 3*(C*a^7*b^2 - (A -
3*C)*a^5*b^4 - 3*B*a^4*b^5 + (3*A - 4*C)*a^3*b^6 + 3*B*a^2*b^7 - 2*A*a*b^8)
*c*d^2 + (3*C*a^8*b - B*a^7*b^2 + 6*C*a^6*b^3 - 3*B*a^5*b^4 + (3*A - 2*C)*a
^4*b^5 + 4*B*a^3*b^6 - (3*A - C)*a^2*b^7)*d^3 + 2*(((A - C)*a^4*b^5 + 3*B*a
^3*b^6 - 3*(A - C)*a^2*b^7 - B*a*b^8)*c^3 - 3*(B*a^4*b^5 - 3*(A - C)*a^3*b^
6 - 3*B*a^2*b^7 + (A - C)*a*b^8)*c^2*d - 3*((A - C)*a^4*b^5 + 3*B*a^3*b^6 -
 3*(A - C)*a^2*b^7 - B*a*b^8)*c*d^2 + (B*a^4*b^5 - 3*(A - C)*a^3*b^6 - 3*B*
a^2*b^7 + (A - C)*a*b^8)*d^3)*f*x)*tan(f*x + e))/((a^6*b^6 + 3*a^4*b^8 + 3*
a^2*b^10 + b^12)*f*tan(f*x + e)^2 + 2*(a^7*b^5 + 3*a^5*b^7 + 3*a^3*b^9 + a*
b^11)*f*tan(f*x + e) + (a^8*b^4 + 3*a^6*b^6 + 3*a^4*b^8 + a^2*b^10)*f)
```

Sympy [F(-2)]

Exception generated.

$$\int \frac{(c+d\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx$$
$$= \text{Exception raised: AttributeError}$$

```
[In] integrate((c+d*tan(f*x+e))**3*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x
+e))**3,x)
```

```
[Out] Exception raised: AttributeError >> 'NoneType' object has no attribute 'pri
mitive'
```

Maxima [A] (verification not implemented)

none

Time = 0.39 (sec) , antiderivative size = 1119, normalized size of antiderivative = 1.40

$$\int \frac{(c+d\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx = \text{Too large to display}$$

```
[In] integrate((c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e
))^3,x, algorithm="maxima")
```

```
[Out] 1/2*(2*C*d^3*tan(f*x + e)/b^3 + 2*(((A - C)*a^3 + 3*B*a^2*b - 3*(A - C)*a*b
^2 - B*b^3)*c^3 - 3*(B*a^3 - 3*(A - C)*a^2*b - 3*B*a*b^2 + (A - C)*b^3)*c^2
*d - 3*((A - C)*a^3 + 3*B*a^2*b - 3*(A - C)*a*b^2 - B*b^3)*c*d^2 + (B*a^3 -
 3*(A - C)*a^2*b - 3*B*a*b^2 + (A - C)*b^3)*d^3)*(f*x + e)/(a^6 + 3*a^4*b^2
 + 3*a^2*b^4 + b^6) - 2*((B*a^3*b^4 - 3*(A - C)*a^2*b^5 - 3*B*a*b^6 + (A -
C)*b^7)*c^3 + 3*((A - C)*a^3*b^4 + 3*B*a^2*b^5 - 3*(A - C)*a*b^6 - B*b^7)*c
^2*d - 3*(C*a^6*b + 3*C*a^4*b^3 + B*a^3*b^4 - 3*(A - 2*C)*a^2*b^5 - 3*B*a*b
^6 + A*b^7)*c*d^2 + (3*C*a^7 - B*a^6*b + 9*C*a^5*b^2 - 3*B*a^4*b^3 - (A - 1
0*C)*a^3*b^4 - 6*B*a^2*b^5 + 3*A*a*b^6)*d^3)*log(b*tan(f*x + e) + a)/(a^6*b
^4 + 3*a^4*b^6 + 3*a^2*b^8 + b^10) + ((B*a^3 - 3*(A - C)*a^2*b - 3*B*a*b^2
+ (A - C)*b^3)*c^3 + 3*((A - C)*a^3 + 3*B*a^2*b - 3*(A - C)*a*b^2 - B*b^3)*
c^2*d - 3*(B*a^3 - 3*(A - C)*a^2*b - 3*B*a*b^2 + (A - C)*b^3)*c*d^2 - ((A -
 C)*a^3 + 3*B*a^2*b - 3*(A - C)*a*b^2 - B*b^3)*d^3)*log(tan(f*x + e)^2 + 1)
/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) - ((C*a^4*b^3 - 3*B*a^3*b^4 + (5*A - 3
*C)*a^2*b^5 + B*a*b^6 + A*b^7)*c^3 + 3*(C*a^5*b^2 + B*a^4*b^3 - (3*A - 5*C)
*a^3*b^4 - 3*B*a^2*b^5 + A*a*b^6)*c^2*d - 3*(3*C*a^6*b - B*a^5*b^2 - (A - 7
*C)*a^4*b^3 - 5*B*a^3*b^4 + 3*A*a^2*b^5)*c*d^2 + (5*C*a^7 - 3*B*a^6*b + (A
+ 9*C)*a^5*b^2 - 7*B*a^4*b^3 + 5*A*a^3*b^4)*d^3 - 2*((B*a^2*b^5 - 2*(A - C)
*a*b^6 - B*b^7)*c^3 - 3*(C*a^4*b^3 - (A - 3*C)*a^2*b^5 - 2*B*a*b^6 + A*b^7)
*c^2*d + 3*(2*C*a^5*b^2 - B*a^4*b^3 + 4*C*a^3*b^4 - 3*B*a^2*b^5 + 2*A*a*b^6
)*c*d^2 - (3*C*a^6*b - 2*B*a^5*b^2 + (A + 5*C)*a^4*b^3 - 4*B*a^3*b^4 + 3*A*
a^2*b^5)*d^3)*tan(f*x + e))/(a^6*b^4 + 2*a^4*b^6 + a^2*b^8 + (a^4*b^6 + 2*a
^2*b^8 + b^10)*tan(f*x + e)^2 + 2*(a^5*b^5 + 2*a^3*b^7 + a*b^9)*tan(f*x + e
)))/f
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 2441 vs. $2(794) = 1588$.

Time = 1.36 (sec) , antiderivative size = 2441, normalized size of antiderivative = 3.06

$$\int \frac{(c+d\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx = \text{Too large to display}$$

```
[In] integrate((c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e
))^3,x, algorithm="giac")
```

```
[Out] 1/2*(2*C*d^3*tan(f*x + e)/b^3 + 2*(A*a^3*c^3 - C*a^3*c^3 + 3*B*a^2*b*c^3 -
3*A*a*b^2*c^3 + 3*C*a*b^2*c^3 - B*b^3*c^3 - 3*B*a^3*c^2*d + 9*A*a^2*b*c^2*d
 - 9*C*a^2*b*c^2*d + 9*B*a*b^2*c^2*d - 3*A*b^3*c^2*d + 3*C*b^3*c^2*d - 3*A*
a^3*c*d^2 + 3*C*a^3*c*d^2 - 9*B*a^2*b*c*d^2 + 9*A*a*b^2*c*d^2 - 9*C*a*b^2*c
*d^2 + 3*B*b^3*c*d^2 + B*a^3*d^3 - 3*A*a^2*b*d^3 + 3*C*a^2*b*d^3 - 3*B*a*b^
2*d^3 + A*b^3*d^3 - C*b^3*d^3)*(f*x + e)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6
) + (B*a^3*c^3 - 3*A*a^2*b*c^3 + 3*C*a^2*b*c^3 - 3*B*a*b^2*c^3 + A*b^3*c^3
- C*b^3*c^3 + 3*A*a^3*c^2*d - 3*C*a^3*c^2*d + 9*B*a^2*b*c^2*d - 9*A*a*b^2*c
^2*d + 9*C*a*b^2*c^2*d - 3*B*b^3*c^2*d - 3*B*a^3*c*d^2 + 9*A*a^2*b*c*d^2 -
```

9*C*a^2*b*c*d^2 + 9*B*a*b^2*c*d^2 - 3*A*b^3*c*d^2 + 3*C*b^3*c*d^2 - A*a^3*d
^3 + C*a^3*d^3 - 3*B*a^2*b*d^3 + 3*A*a*b^2*d^3 - 3*C*a*b^2*d^3 + B*b^3*d^3)
*log(tan(f*x + e)^2 + 1)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) - 2*(B*a^3*b^4
*c^3 - 3*A*a^2*b^5*c^3 + 3*C*a^2*b^5*c^3 - 3*B*a*b^6*c^3 + A*b^7*c^3 - C*b^
7*c^3 + 3*A*a^3*b^4*c^2*d - 3*C*a^3*b^4*c^2*d + 9*B*a^2*b^5*c^2*d - 9*A*a*b
^6*c^2*d + 9*C*a*b^6*c^2*d - 3*B*b^7*c^2*d - 3*C*a^6*b*c*d^2 - 9*C*a^4*b^3*
c*d^2 - 3*B*a^3*b^4*c*d^2 + 9*A*a^2*b^5*c*d^2 - 18*C*a^2*b^5*c*d^2 + 9*B*a*
b^6*c*d^2 - 3*A*b^7*c*d^2 + 3*C*a^7*d^3 - B*a^6*b*d^3 + 9*C*a^5*b^2*d^3 - 3
*B*a^4*b^3*d^3 - A*a^3*b^4*d^3 + 10*C*a^3*b^4*d^3 - 6*B*a^2*b^5*d^3 + 3*A*a
*b^6*d^3)*log(abs(b*tan(f*x + e) + a))/(a^6*b^4 + 3*a^4*b^6 + 3*a^2*b^8 + b
^10) + (3*B*a^3*b^6*c^3*tan(f*x + e)^2 - 9*A*a^2*b^7*c^3*tan(f*x + e)^2 + 9
*C*a^2*b^7*c^3*tan(f*x + e)^2 - 9*B*a*b^8*c^3*tan(f*x + e)^2 + 3*A*b^9*c^3*
tan(f*x + e)^2 - 3*C*b^9*c^3*tan(f*x + e)^2 + 9*A*a^3*b^6*c^2*d*tan(f*x + e
)^2 - 9*C*a^3*b^6*c^2*d*tan(f*x + e)^2 + 27*B*a^2*b^7*c^2*d*tan(f*x + e)^2
- 27*A*a*b^8*c^2*d*tan(f*x + e)^2 + 27*C*a*b^8*c^2*d*tan(f*x + e)^2 - 9*B*b
^9*c^2*d*tan(f*x + e)^2 - 9*C*a^6*b^3*c*d^2*tan(f*x + e)^2 - 27*C*a^4*b^5*c
*d^2*tan(f*x + e)^2 - 9*B*a^3*b^6*c*d^2*tan(f*x + e)^2 + 27*A*a^2*b^7*c*d^2
*tan(f*x + e)^2 - 54*C*a^2*b^7*c*d^2*tan(f*x + e)^2 + 27*B*a*b^8*c*d^2*tan(
f*x + e)^2 - 9*A*b^9*c*d^2*tan(f*x + e)^2 + 9*C*a^7*b^2*d^3*tan(f*x + e)^2
- 3*B*a^6*b^3*d^3*tan(f*x + e)^2 + 27*C*a^5*b^4*d^3*tan(f*x + e)^2 - 9*B*a^
4*b^5*d^3*tan(f*x + e)^2 - 3*A*a^3*b^6*d^3*tan(f*x + e)^2 + 30*C*a^3*b^6*d^
3*tan(f*x + e)^2 - 18*B*a^2*b^7*d^3*tan(f*x + e)^2 + 9*A*a*b^8*d^3*tan(f*x
+ e)^2 + 8*B*a^4*b^5*c^3*tan(f*x + e) - 22*A*a^3*b^6*c^3*tan(f*x + e) + 22*
C*a^3*b^6*c^3*tan(f*x + e) - 18*B*a^2*b^7*c^3*tan(f*x + e) + 2*A*a*b^8*c^3*
tan(f*x + e) - 2*C*a*b^8*c^3*tan(f*x + e) - 2*B*b^9*c^3*tan(f*x + e) - 6*C*
a^6*b^3*c^2*d*tan(f*x + e) + 24*A*a^4*b^5*c^2*d*tan(f*x + e) - 42*C*a^4*b^5
*c^2*d*tan(f*x + e) + 66*B*a^3*b^6*c^2*d*tan(f*x + e) - 54*A*a^2*b^7*c^2*d*
tan(f*x + e) + 36*C*a^2*b^7*c^2*d*tan(f*x + e) - 6*B*a*b^8*c^2*d*tan(f*x +
e) - 6*A*b^9*c^2*d*tan(f*x + e) - 6*C*a^7*b^2*c*d^2*tan(f*x + e) - 6*B*a^6*
b^3*c*d^2*tan(f*x + e) - 18*C*a^5*b^4*c*d^2*tan(f*x + e) - 42*B*a^4*b^5*c*d
^2*tan(f*x + e) + 66*A*a^3*b^6*c*d^2*tan(f*x + e) - 84*C*a^3*b^6*c*d^2*tan(
f*x + e) + 36*B*a^2*b^7*c*d^2*tan(f*x + e) - 6*A*a*b^8*c*d^2*tan(f*x + e) +
12*C*a^8*b*d^3*tan(f*x + e) - 2*B*a^7*b^2*d^3*tan(f*x + e) - 2*A*a^6*b^3*d
^3*tan(f*x + e) + 38*C*a^6*b^3*d^3*tan(f*x + e) - 6*B*a^5*b^4*d^3*tan(f*x +
e) - 14*A*a^4*b^5*d^3*tan(f*x + e) + 50*C*a^4*b^5*d^3*tan(f*x + e) - 28*B*
a^3*b^6*d^3*tan(f*x + e) + 12*A*a^2*b^7*d^3*tan(f*x + e) - C*a^6*b^3*c^3 +
6*B*a^5*b^4*c^3 - 14*A*a^4*b^5*c^3 + 11*C*a^4*b^5*c^3 - 7*B*a^3*b^6*c^3 - 3
*A*a^2*b^7*c^3 - B*a*b^8*c^3 - A*b^9*c^3 - 3*C*a^7*b^2*c^2*d - 3*B*a^6*b^3*
c^2*d + 18*A*a^5*b^4*c^2*d - 27*C*a^5*b^4*c^2*d + 33*B*a^4*b^5*c^2*d - 21*A
*a^3*b^6*c^2*d + 12*C*a^3*b^6*c^2*d - 3*A*a*b^8*c^2*d - 3*B*a^7*b^2*c*d^2 -
3*A*a^6*b^3*c*d^2 + 3*C*a^6*b^3*c*d^2 - 27*B*a^5*b^4*c*d^2 + 33*A*a^4*b^5*
c*d^2 - 33*C*a^4*b^5*c*d^2 + 12*B*a^3*b^6*c*d^2 + 4*C*a^9*d^3 - A*a^7*b^2*d
^3 + 13*C*a^7*b^2*d^3 + B*a^6*b^3*d^3 - 9*A*a^5*b^4*d^3 + 21*C*a^5*b^4*d^3
- 11*B*a^4*b^5*d^3 + 4*A*a^3*b^6*d^3)/((a^6*b^4 + 3*a^4*b^6 + 3*a^2*b^8 + b
^10)*(b*tan(f*x + e) + a)^2))/f

Mupad [B] (verification not implemented)

Time = 17.39 (sec) , antiderivative size = 1172, normalized size of antiderivative = 1.47

$$\int \frac{(c+d\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx$$

$$= \frac{\ln\left(a+b\tan(e+f\,x)\right)\,\left(b^3\,\left(3\,B\,a^4\,d^3+9\,C\,c\,a^4\,d^2\right)-b^6\,\left(3\,A\,a\,d^3-3\,B\,a\,c^3-9\,A\,a\,c^2\,d+9\,B\,a\,c\,d^2+9\,C\,a\right.\right.}{}$$

$$+\frac{\ln\left(\tan(e+f\,x)+1\mathrm{i}\right)\,\left(A\,c^3+A\,d^3\,1\mathrm{i}-B\,c^3\,1\mathrm{i}+B\,d^3-C\,c^3-C\,d^3\,1\mathrm{i}-3\,A\,c\,d^2-A\,c^2\,d\,3\mathrm{i}+B\,c\,d^2\,3\mathrm{i}-\right.}{2\,f\,\left(-a^3\,1\mathrm{i}-3\,a^2\,b+a\,b^2\,3\mathrm{i}+b^3\right)}$$

$$-\frac{\frac{\tan(e+f\,x)\,\left(B\,b^6\,c^3+3\,C\,a^6\,d^3+2\,A\,a\,b^5\,c^3-2\,B\,a^5\,b\,d^3-2\,C\,a\,b^5\,c^3+3\,A\,b^6\,c^2\,d+3\,A\,a^2\,b^4\,d^3+A\,a^4\,b^2\,d^3-B\,a^2\,b^4\,c^3-4\,B\,a^3\,b^3\,d^3+5\,C\,a^4\,b^2\,d^3-\right.}{a^4+2\,a^2\,b^2+b}}{}$$

$$+\frac{\ln\left(\tan(e+f\,x)-\mathrm{i}\right)\,\left(A\,d^3-B\,c^3-C\,d^3-3\,A\,c^2\,d+3\,B\,c\,d^2+3\,C\,c^2\,d+A\,c^3\,1\mathrm{i}+B\,d^3\,1\mathrm{i}-C\,c^3\,1\mathrm{i}-A\right.}{2\,f\,\left(-a^3-a^2\,b\,3\mathrm{i}+3\,a\,b^2+b^3\,1\mathrm{i}\right)}$$

$$+\frac{C\,d^3\tan(e+f\,x)}{b^3\,f}$$

```
[In] int(((c + d*tan(e + f*x))^3*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a + b
*tan(e + f*x))^3,x)
```

```
[Out] (log(tan(e + f*x) + 1i)*(A*c^3 + A*d^3*1i - B*c^3*1i + B*d^3 - C*c^3 - C*d^
3*1i - 3*A*c*d^2 - A*c^2*d*3i + B*c*d^2*3i - 3*B*c^2*d + 3*C*c*d^2 + C*c^2*
d*3i))/(2*f*(a*b^2*3i - 3*a^2*b - a^3*1i + b^3)) - ((tan(e + f*x)*(B*b^6*c^
3 + 3*C*a^6*d^3 + 2*A*a*b^5*c^3 - 2*B*a^5*b*d^3 - 2*C*a*b^5*c^3 + 3*A*b^6*c
^2*d + 3*A*a^2*b^4*d^3 + A*a^4*b^2*d^3 - B*a^2*b^4*c^3 - 4*B*a^3*b^3*d^3 +
5*C*a^4*b^2*d^3 - 3*A*a^2*b^4*c^2*d + 9*B*a^2*b^4*c*d^2 + 3*B*a^4*b^2*c*d^2
 + 9*C*a^2*b^4*c^2*d - 12*C*a^3*b^3*c*d^2 + 3*C*a^4*b^2*c^2*d - 6*A*a*b^5*c
*d^2 - 6*B*a*b^5*c^2*d - 6*C*a^5*b*c*d^2))/(a^4 + b^4 + 2*a^2*b^2) + (A*b^7
*c^3 + 5*C*a^7*d^3 + B*a*b^6*c^3 - 3*B*a^6*b*d^3 + 5*A*a^2*b^5*c^3 + 5*A*a^
3*b^4*d^3 + A*a^5*b^2*d^3 - 3*B*a^3*b^4*c^3 - 7*B*a^4*b^3*d^3 - 3*C*a^2*b^5
*c^3 + C*a^4*b^3*c^3 + 9*C*a^5*b^2*d^3 - 9*A*a^2*b^5*c*d^2 - 9*A*a^3*b^4*c^
2*d + 3*A*a^4*b^3*c*d^2 - 9*B*a^2*b^5*c^2*d + 15*B*a^3*b^4*c*d^2 + 3*B*a^4*
b^3*c^2*d + 3*B*a^5*b^2*c*d^2 + 15*C*a^3*b^4*c^2*d - 21*C*a^4*b^3*c*d^2 + 3
*C*a^5*b^2*c^2*d + 3*A*a*b^6*c^2*d - 9*C*a^6*b*c*d^2)/(2*b*(a^4 + b^4 + 2*a
^2*b^2)))/(f*(a^2*b^3 + b^5*tan(e + f*x)^2 + 2*a*b^4*tan(e + f*x))) + (log(
a + b*tan(e + f*x))*(b^3*(3*B*a^4*d^3 + 9*C*a^4*c*d^2) - b^6*(3*A*a*d^3 - 3
*B*a*c^3 - 9*A*a*c^2*d + 9*B*a*c*d^2 + 9*C*a*c^2*d) + b^5*(3*A*a^2*c^3 + 6*
B*a^2*d^3 - 3*C*a^2*c^3 - 9*A*a^2*c*d^2 - 9*B*a^2*c^2*d + 18*C*a^2*c*d^2) +
 b^4*(A*a^3*d^3 - B*a^3*c^3 - 10*C*a^3*d^3 - 3*A*a^3*c^2*d + 3*B*a^3*c*d^2
+ 3*C*a^3*c^2*d) + b*(B*a^6*d^3 + 3*C*a^6*c*d^2) + b^7*(C*c^3 - A*c^3 + 3*A
*c*d^2 + 3*B*c^2*d) - 3*C*a^7*d^3 - 9*C*a^5*b^2*d^3))/(f*(b^10 + 3*a^2*b^8
+ 3*a^4*b^6 + a^6*b^4)) + (log(tan(e + f*x) - 1i)*(A*c^3*1i + A*d^3 - B*c^3
 + B*d^3*1i - C*c^3*1i - C*d^3 - A*c*d^2*3i - 3*A*c^2*d + 3*B*c*d^2 - B*c^2
*d*3i + C*c*d^2*3i + 3*C*c^2*d))/(2*f*(3*a*b^2 - a^2*b*3i - a^3 + b^3*1i))
+ (C*d^3*tan(e + f*x))/(b^3*f)
```

3.70 $\int \frac{(a+b\tan(e+fx))^3(A+B\tan(e+fx)+C\tan^2(e+fx))}{c+d\tan(e+fx)}\,dx$

Optimal result 671
Rubi [A] (verified) 672
Mathematica [C] (verified) 675
Maple [A] (verified) 675
Fricas [A] (verification not implemented) 676
Sympy [C] (verification not implemented) 676
Maxima [A] (verification not implemented) 680
Giac [A] (verification not implemented) 681
Mupad [B] (verification not implemented) 682

Optimal result

Integrand size = 45, antiderivative size = 337

$$\int \frac{(a+b\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{c+d\tan(e+fx)}\,dx$$
$$=\frac{(a^3(Ac-cC+Bd)-3ab^2(Ac-cC+Bd)-3a^2b(Bc-(A-C)d)+b^3(Bc-(A-C)d))\,x}{c^2+d^2}$$
$$-\frac{(3a^2b(Ac-cC+Bd)-b^3(Ac-cC+Bd)+a^3(Bc-(A-C)d)-3ab^2(Bc-(A-C)d))\log(\cos(e+}{(c^2+d^2)\,f}$$
$$-\frac{(bc-ad)^3\,(c^2C-Bcd+Ad^2)\log(c+d\tan(e+fx))}{d^4\,(c^2+d^2)\,f}$$
$$+\frac{b(b(Ab+aB-bC)d^2+(bc-ad)(bcC-bBd-aCd))\tan(e+fx)}{d^3f}$$
$$-\frac{(bcC-bBd-aCd)(a+b\tan(e+fx))^2}{2d^2f}+\frac{C(a+b\tan(e+fx))^3}{3df}$$

```
[Out] (a^3*(A*c+B*d-C*c)-3*a*b^2*(A*c+B*d-C*c)-3*a^2*b*(B*c-(A-C)*d)+b^3*(B*c-(A-
C)*d))*x/(c^2+d^2)-(3*a^2*b*(A*c+B*d-C*c)-b^3*(A*c+B*d-C*c)+a^3*(B*c-(A-C)*
d)-3*a*b^2*(B*c-(A-C)*d))*ln(cos(f*x+e))/(c^2+d^2)/f-(-a*d+b*c)^3*(A*d^2-B*
c*d+C*c^2)*ln(c+d*tan(f*x+e))/d^4/(c^2+d^2)/f+b*(b*(A*b+B*a-C*b)*d^2+(-a*d+
b*c)*(-B*b*d-C*a*d+C*b*c))*tan(f*x+e)/d^3/f-1/2*(-B*b*d-C*a*d+C*b*c)*(a+b*t
an(f*x+e))^2/d^2/f+1/3*C*(a+b*tan(f*x+e))^3/d/f
```

Rubi [A] (verified)

Time = 1.62 (sec) , antiderivative size = 337, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.133, Rules used = {3728, 3718, 3707, 3698, 31, 3556}

$$\int \frac{(a+b\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{c+d\tan(e+fx)}\,dx =$$

$$-\frac{\log(\cos(e+fx))\,(a^3(Bc-d(A-C))+3a^2b(Ac+Bd-cC)-3ab^2(Bc-d(A-C))-b^3(Ac+Bd-c}{f\,(c^2+d^2)}$$
$$+\frac{x(a^3(Ac+Bd-cC)-3a^2b(Bc-d(A-C))-3ab^2(Ac+Bd-cC)+b^3(Bc-d(A-C)))}{c^2+d^2}$$
$$-\frac{(bc-ad)^3\,(Ad^2-Bcd+c^2C)\log(c+d\tan(e+fx))}{d^4f\,(c^2+d^2)}$$
$$+\frac{b\tan(e+fx)\,(bd^2(aB+Ab-bC)+(bc-ad)(-aCd-bBd+bcC))}{d^3f}$$
$$-\frac{(-aCd-bBd+bcC)(a+b\tan(e+fx))^2}{2d^2f}+\frac{C(a+b\tan(e+fx))^3}{3df}$$

```
[In] Int[((a + b*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c + d
*Tan[e + f*x]),x]
```

```
[Out] ((a^3*(A*c - c*C + B*d) - 3*a*b^2*(A*c - c*C + B*d) - 3*a^2*b*(B*c - (A - C
)*d) + b^3*(B*c - (A - C)*d))*x)/(c^2 + d^2) - ((3*a^2*b*(A*c - c*C + B*d)
- b^3*(A*c - c*C + B*d) + a^3*(B*c - (A - C)*d) - 3*a*b^2*(B*c - (A - C)*d)
)*Log[Cos[e + f*x]])/((c^2 + d^2)*f) - ((b*c - a*d)^3*(c^2*C - B*c*d + A*d^
2)*Log[c + d*Tan[e + f*x]])/(d^4*(c^2 + d^2)*f) + (b*(b*(A*b + a*B - b*C)*d
^2 + (b*c - a*d)*(b*c*C - b*B*d - a*C*d))*Tan[e + f*x])/(d^3*f) - ((b*c*C -
 b*B*d - a*C*d)*(a + b*Tan[e + f*x])^2)/(2*d^2*f) + (C*(a + b*Tan[e + f*x])
^3)/(3*d*f)
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3698

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3707

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[(a*A + b*B - a
*C)*(x/(a^2 + b^2)), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\text{integral} = \frac{C(a+b\tan(e+fx))^3}{3df} + \frac{\int \frac{(a+b\tan(e+fx))^2(-3(bcC-aAd)+3(Ab+aB-bC)d\tan(e+fx)-3(bcC-bBd-aCd)\tan^2(e+fx))}{c+d\tan(e+fx)}\,dx}{3d}$$

$$= -\frac{(bcC-bBd-aCd)(a+b\tan(e+fx))^2}{2d^2f} + \frac{C(a+b\tan(e+fx))^3}{3df} + \frac{\int \frac{(a+b\tan(e+fx))(-6(2abcCd-a^2Ad^2-b^2c(cC-Bd))+6(a^2B-b^2B+2ab(A-C))d^2\tan(e+fx)+6(b(Ab+aB-bC)d^2+(bc-ad)(\cdots}{c+d\tan(e+fx)}}{6d^2}$$

$$= \frac{b(b(Ab+aB-bC)d^2+(bc-ad)(bcC-bBd-aCd))\tan(e+fx)}{d^3f} - \frac{(bcC-bBd-aCd)(a+b\tan(e+fx))^2}{2d^2f} + \frac{C(a+b\tan(e+fx))^3}{3df} - \frac{\int \frac{6(3a^2bcCd^2-a^3Ad^3-3ab^2cd(cC-Bd)+b^3c(c^2C-Bcd+(A-C)d^2))-6(a^3B-3ab^2B+3a^2b(A-C)-b^3(A-C))d^3\tan(e+fx)-6(a}{c+d\tan(e+fx)}}{6d^3}$$

$$= \frac{(a^3(Ac-cC+Bd)-3ab^2(Ac-cC+Bd)-3a^2b(Bc-(A-C)d)+b^3(Bc-(A-C)d))x}{c^2+d^2} + \frac{b(b(Ab+aB-bC)d^2+(bc-ad)(bcC-bBd-aCd))\tan(e+fx)}{d^3f} - \frac{(bcC-bBd-aCd)(a+b\tan(e+fx))^2}{2d^2f} + \frac{C(a+b\tan(e+fx))^3}{3df} - \frac{((bc-ad)^3(c^2C-Bcd+Ad^2))\int \frac{1+\tan^2(e+fx)}{c+d\tan(e+fx)}dx}{d^3(c^2+d^2)} + \frac{(3a^2b(Ac-cC+Bd)-b^3(Ac-cC+Bd)+a^3(Bc-(A-C)d)-3ab^2(Bc-(A-C)d))\int \tan}{c^2+d^2}$$

$$= \frac{(a^3(Ac-cC+Bd)-3ab^2(Ac-cC+Bd)-3a^2b(Bc-(A-C)d)+b^3(Bc-(A-C)d))x}{c^2+d^2} - \frac{(3a^2b(Ac-cC+Bd)-b^3(Ac-cC+Bd)+a^3(Bc-(A-C)d)-3ab^2(Bc-(A-C)d))\log(c}{(c^2+d^2)f} + \frac{b(b(Ab+aB-bC)d^2+(bc-ad)(bcC-bBd-aCd))\tan(e+fx)}{d^3f} - \frac{(bcC-bBd-aCd)(a+b\tan(e+fx))^2}{2d^2f} + \frac{C(a+b\tan(e+fx))^3}{3df} - \frac{((bc-ad)^3(c^2C-Bcd+Ad^2))\,\mathrm{Subst}\left(\int \frac{1}{c+x}dx, x, d\tan(e+fx)\right)}{d^4(c^2+d^2)f}$$

$$= \frac{(a^3(Ac-cC+Bd)-3ab^2(Ac-cC+Bd)-3a^2b(Bc-(A-C)d)+b^3(Bc-(A-C)d))x}{c^2+d^2} - \frac{(3a^2b(Ac-cC+Bd)-b^3(Ac-cC+Bd)+a^3(Bc-(A-C)d)-3ab^2(Bc-(A-C)d))\log(c}{(c^2+d^2)f} - \frac{(bc-ad)^3(c^2C-Bcd+Ad^2)\log(c+d\tan(e+fx))}{d^4(c^2+d^2)f} + \frac{b(b(Ab+aB-bC)d^2+(bc-ad)(bcC-bBd-aCd))\tan(e+fx)}{d^3f} - \frac{(bcC-bBd-aCd)(a+b\tan(e+fx))^2}{2d^2f} + \frac{C(a+b\tan(e+fx))^3}{3df}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 4.86 (sec) , antiderivative size = 258, normalized size of antiderivative = 0.77

$$\int \frac{(a+b\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{c+d\tan(e+fx)}\,dx$$
$$= \frac{\frac{3(a+ib)^3(-iA+B+iC)d^2\log(i-\tan(e+fx))}{c+id}+\frac{3(a-ib)^3(iA+B-iC)d^2\log(i+\tan(e+fx))}{c-id}+\frac{6(-bc+ad)^3(c^2C-Bcd+Ad^2)\log(c+d\tan(e+f}{d^2(c^2+d^2)}}{}$$

```
[In] Integrate[((a + b*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(c + d*Tan[e + f*x]),x]
```

```
[Out] ((3*(a + I*b)^3*((-I)*A + B + I*C)*d^2*Log[I - Tan[e + f*x]])/(c + I*d) + (
3*(a - I*b)^3*(I*A + B - I*C)*d^2*Log[I + Tan[e + f*x]])/(c - I*d) + (6*(-(
b*c) + a*d)^3*(c^2*C - B*c*d + A*d^2)*Log[c + d*Tan[e + f*x]])/(d^2*(c^2 +
d^2)) + 6*b^2*(A*b + a*B - b*C)*d*Tan[e + f*x] - (6*b*(b*c - a*d)*(-(b*c*C)
 + b*B*d + a*C*d)*Tan[e + f*x])/d - 3*(b*c*C - b*B*d - a*C*d)*(a + b*Tan[e
+ f*x])^2 + 2*C*d*(a + b*Tan[e + f*x])^3)/(6*d^2*f)
```

Maple [A] (verified)

Time = 0.24 (sec) , antiderivative size = 500, normalized size of antiderivative = 1.48

method	result
norman	$\frac{(A a^3c+3A a^2bd-3Aa b^2c-A b^3d+B a^3d-3B a^2bc-3Ba b^2d+B b^3c-C a^3c-3C a^2bd+3Ca b^2c+C b^3d)x}{c^2+d^2}+\frac{(A b^2d^2+}{}$
derivativedivides	$\frac{b\left(\frac{C b^2d^2\tan(fx+e)^3}{3}+\frac{B b^2d^2\tan(fx+e)^2}{2}+\frac{3Cab d^2\tan(fx+e)^2}{2}-\frac{C b^2cd\tan(fx+e)^2}{2}+\tan(fx+e)A b^2d^2+3\tan(fx+e)Bab d^2-\tan\right.}{d^3}$
default	$\frac{b\left(\frac{C b^2d^2\tan(fx+e)^3}{3}+\frac{B b^2d^2\tan(fx+e)^2}{2}+\frac{3Cab d^2\tan(fx+e)^2}{2}-\frac{C b^2cd\tan(fx+e)^2}{2}+\tan(fx+e)A b^2d^2+3\tan(fx+e)Bab d^2-\tan\right.}{d^3}$
parallelrisch	Expression too large to display
risch	Expression too large to display

```
[In] int((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e)),x,m
ethod=_RETURNVERBOSE)
```

```
[Out] (A*a^3*c+3*A*a^2*b*d-3*A*a*b^2*c-A*b^3*d+B*a^3*d-3*B*a^2*b*c-3*B*a*b^2*d+B*
b^3*c-C*a^3*c-3*C*a^2*b*d+3*C*a*b^2*c+C*b^3*d)/(c^2+d^2)*x+(A*b^2*d^2+3*B*a
*b*d^2-B*b^2*c*d+3*C*a^2*d^2-3*C*a*b*c*d+C*b^2*c^2-C*b^2*d^2)*b/f/d^3*tan(f
*x+e)+1/3*C*b^3/d/f*tan(f*x+e)^3+1/2*b^2*(B*b*d+3*C*a*d-C*b*c)/d^2/f*tan(f*
x+e)^2+(A*a^3*d^5-3*A*a^2*b*c*d^4+3*A*a*b^2*c^2*d^3-A*b^3*c^3*d^2-B*a^3*c*d
```

```
^4+3*B*a^2*b*c^2*d^3-3*B*a*b^2*c^3*d^2+B*b^3*c^4*d+C*a^3*c^2*d^3-3*C*a^2*b*
c^3*d^2+3*C*a*b^2*c^4*d-C*b^3*c^5)/(c^2+d^2)/d^4/f*ln(c+d*tan(f*x+e))-1/2*(
A*a^3*d-3*A*a^2*b*c-3*A*a*b^2*d+A*b^3*c-B*a^3*c-3*B*a^2*b*d+3*B*a*b^2*c+B*b
^3*d-C*a^3*d+3*C*a^2*b*c+3*C*a*b^2*d-C*b^3*c)/f/(c^2+d^2)*ln(1+tan(f*x+e)^2
)
```

Fricas [A] (verification not implemented)

none

Time = 0.77 (sec) , antiderivative size = 627, normalized size of antiderivative = 1.86

$$\int \frac{(a+b\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{c+d\tan(e+fx)}\,dx$$
$$=\frac{2\,(Cb^3c^2d^3+Cb^3d^5)\tan{(fx+e)}^3+6\,(((A-C)a^3-3\,Ba^2b-3\,(A-C)ab^2+Bb^3)cd^4+(Ba^3+3\,(A-C}{}$$

```
[In] integrate((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
)),x, algorithm="fricas")
```

```
[Out] 1/6*(2*(C*b^3*c^2*d^3 + C*b^3*d^5)*tan(f*x + e)^3 + 6*(((A - C)*a^3 - 3*B*a
^2*b - 3*(A - C)*a*b^2 + B*b^3)*c*d^4 + (B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^
2 - (A - C)*b^3)*d^5)*f*x - 3*(C*b^3*c^3*d^2 + C*b^3*c*d^4 - (3*C*a*b^2 + B
*b^3)*c^2*d^3 - (3*C*a*b^2 + B*b^3)*d^5)*tan(f*x + e)^2 - 3*(C*b^3*c^5 - A*
a^3*d^5 - (3*C*a*b^2 + B*b^3)*c^4*d + (3*C*a^2*b + 3*B*a*b^2 + A*b^3)*c^3*d
^2 - (C*a^3 + 3*B*a^2*b + 3*A*a*b^2)*c^2*d^3 + (B*a^3 + 3*A*a^2*b)*c*d^4)*l
og((d^2*tan(f*x + e)^2 + 2*c*d*tan(f*x + e) + c^2)/(tan(f*x + e)^2 + 1)) +
3*(C*b^3*c^5 - (3*C*a*b^2 + B*b^3)*c^4*d + (3*C*a^2*b + 3*B*a*b^2 + A*b^3)*
c^3*d^2 - (C*a^3 + 3*B*a^2*b + 3*A*a*b^2)*c^2*d^3 + (3*C*a^2*b + 3*B*a*b^2
+ (A - C)*b^3)*c*d^4 - (C*a^3 + 3*B*a^2*b + 3*(A - C)*a*b^2 - B*b^3)*d^5)*l
og(1/(tan(f*x + e)^2 + 1)) + 6*(C*b^3*c^4*d - (3*C*a*b^2 + B*b^3)*c^3*d^2 +
 (3*C*a^2*b + 3*B*a*b^2 + A*b^3)*c^2*d^3 - (3*C*a*b^2 + B*b^3)*c*d^4 + (3*C
*a^2*b + 3*B*a*b^2 + (A - C)*b^3)*d^5)*tan(f*x + e))/((c^2*d^4 + d^6)*f)
```

Sympy [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 20.49 (sec) , antiderivative size = 7096, normalized size of antiderivative = 21.06

$$\int \frac{(a+b\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{c+d\tan(e+fx)}\,dx=\text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))**3*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x
+e)),x)
```

```
[Out] Piecewise((zoo*x*(a + b*tan(e))**3*(A + B*tan(e) + C*tan(e)**2)/tan(e), Eq(
c, 0) & Eq(d, 0) & Eq(f, 0)), ((A*a**3*x + 3*A*a**2*b*log(tan(e + f*x)**2 +
 1)/(2*f) - 3*A*a*b**2*x + 3*A*a*b**2*tan(e + f*x)/f - A*b**3*log(tan(e + f
*x)**2 + 1)/(2*f) + A*b**3*tan(e + f*x)**2/(2*f) + B*a**3*log(tan(e + f*x)*
*2 + 1)/(2*f) - 3*B*a**2*b*x + 3*B*a**2*b*tan(e + f*x)/f - 3*B*a*b**2*log(t
an(e + f*x)**2 + 1)/(2*f) + 3*B*a*b**2*tan(e + f*x)**2/(2*f) + B*b**3*x + B
*b**3*tan(e + f*x)**3/(3*f) - B*b**3*tan(e + f*x)/f - C*a**3*x + C*a**3*tan
(e + f*x)/f - 3*C*a**2*b*log(tan(e + f*x)**2 + 1)/(2*f) + 3*C*a**2*b*tan(e
+ f*x)**2/(2*f) + 3*C*a*b**2*x + C*a*b**2*tan(e + f*x)**3/f - 3*C*a*b**2*ta
n(e + f*x)/f + C*b**3*log(tan(e + f*x)**2 + 1)/(2*f) + C*b**3*tan(e + f*x)*
*4/(4*f) - C*b**3*tan(e + f*x)**2/(2*f))/c, Eq(d, 0)), (3*I*A*a**3*f*x*tan(
e + f*x)/(6*d*f*tan(e + f*x) - 6*I*d*f) + 3*A*a**3*f*x/(6*d*f*tan(e + f*x)
- 6*I*d*f) + 3*I*A*a**3/(6*d*f*tan(e + f*x) - 6*I*d*f) + 9*A*a**2*b*f*x*tan
(e + f*x)/(6*d*f*tan(e + f*x) - 6*I*d*f) - 9*I*A*a**2*b*f*x/(6*d*f*tan(e +
f*x) - 6*I*d*f) - 9*A*a**2*b/(6*d*f*tan(e + f*x) - 6*I*d*f) + 9*I*A*a*b**2*
f*x*tan(e + f*x)/(6*d*f*tan(e + f*x) - 6*I*d*f) + 9*A*a*b**2*f*x/(6*d*f*tan
(e + f*x) - 6*I*d*f) + 9*A*a*b**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(6*
d*f*tan(e + f*x) - 6*I*d*f) - 9*I*A*a*b**2*log(tan(e + f*x)**2 + 1)/(6*d*f*
tan(e + f*x) - 6*I*d*f) - 9*I*A*a*b**2/(6*d*f*tan(e + f*x) - 6*I*d*f) - 9*A
*b**3*f*x*tan(e + f*x)/(6*d*f*tan(e + f*x) - 6*I*d*f) + 9*I*A*b**3*f*x/(6*d
*f*tan(e + f*x) - 6*I*d*f) + 3*I*A*b**3*log(tan(e + f*x)**2 + 1)*tan(e + f*
x)/(6*d*f*tan(e + f*x) - 6*I*d*f) + 3*A*b**3*log(tan(e + f*x)**2 + 1)/(6*d*
f*tan(e + f*x) - 6*I*d*f) + 6*A*b**3*tan(e + f*x)**2/(6*d*f*tan(e + f*x) -
6*I*d*f) + 9*A*b**3/(6*d*f*tan(e + f*x) - 6*I*d*f) + 3*B*a**3*f*x*tan(e + f
*x)/(6*d*f*tan(e + f*x) - 6*I*d*f) - 3*I*B*a**3*f*x/(6*d*f*tan(e + f*x) - 6
*I*d*f) - 3*B*a**3/(6*d*f*tan(e + f*x) - 6*I*d*f) + 9*I*B*a**2*b*f*x*tan(e
+ f*x)/(6*d*f*tan(e + f*x) - 6*I*d*f) + 9*B*a**2*b*f*x/(6*d*f*tan(e + f*x)
- 6*I*d*f) + 9*B*a**2*b*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(6*d*f*tan(e
+ f*x) - 6*I*d*f) - 9*I*B*a**2*b*log(tan(e + f*x)**2 + 1)/(6*d*f*tan(e + f*
x) - 6*I*d*f) - 9*I*B*a**2*b/(6*d*f*tan(e + f*x) - 6*I*d*f) - 27*B*a*b**2*f
*x*tan(e + f*x)/(6*d*f*tan(e + f*x) - 6*I*d*f) + 27*I*B*a*b**2*f*x/(6*d*f*t
an(e + f*x) - 6*I*d*f) + 9*I*B*a*b**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)
/(6*d*f*tan(e + f*x) - 6*I*d*f) + 9*B*a*b**2*log(tan(e + f*x)**2 + 1)/(6*d*
f*tan(e + f*x) - 6*I*d*f) + 18*B*a*b**2*tan(e + f*x)**2/(6*d*f*tan(e + f*x)
 - 6*I*d*f) + 27*B*a*b**2/(6*d*f*tan(e + f*x) - 6*I*d*f) - 9*I*B*b**3*f*x*t
an(e + f*x)/(6*d*f*tan(e + f*x) - 6*I*d*f) - 9*B*b**3*f*x/(6*d*f*tan(e + f*
x) - 6*I*d*f) - 6*B*b**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(6*d*f*tan(e
 + f*x) - 6*I*d*f) + 6*I*B*b**3*log(tan(e + f*x)**2 + 1)/(6*d*f*tan(e + f*x
) - 6*I*d*f) + 3*B*b**3*tan(e + f*x)**3/(6*d*f*tan(e + f*x) - 6*I*d*f) + 3*
I*B*b**3*tan(e + f*x)**2/(6*d*f*tan(e + f*x) - 6*I*d*f) + 9*I*B*b**3/(6*d*f
*tan(e + f*x) - 6*I*d*f) + 3*I*C*a**3*f*x*tan(e + f*x)/(6*d*f*tan(e + f*x)
- 6*I*d*f) + 3*C*a**3*f*x/(6*d*f*tan(e + f*x) - 6*I*d*f) + 3*C*a**3*log(tan
(e + f*x)**2 + 1)*tan(e + f*x)/(6*d*f*tan(e + f*x) - 6*I*d*f) - 3*I*C*a**3*
log(tan(e + f*x)**2 + 1)/(6*d*f*tan(e + f*x) - 6*I*d*f) - 3*I*C*a**3/(6*d*f
*tan(e + f*x) - 6*I*d*f) - 27*C*a**2*b*f*x*tan(e + f*x)/(6*d*f*tan(e + f*x)
```

 - 6*I*d*f) + 27*I*C*a**2*b*f*x/(6*d*f*tan(e + f*x) - 6*I*d*f) + 9*I*C*a**2
*b*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(6*d*f*tan(e + f*x) - 6*I*d*f) + 9
*C*a**2*b*log(tan(e + f*x)**2 + 1)/(6*d*f*tan(e + f*x) - 6*I*d*f) + 18*C*a*
*2*b*tan(e + f*x)**2/(6*d*f*tan(e + f*x) - 6*I*d*f) + 27*C*a**2*b/(6*d*f*ta
n(e + f*x) - 6*I*d*f) - 27*I*C*a*b**2*f*x*tan(e + f*x)/(6*d*f*tan(e + f*x)
 - 6*I*d*f) - 27*C*a*b**2*f*x/(6*d*f*tan(e + f*x) - 6*I*d*f) - 18*C*a*b**2*l
og(tan(e + f*x)**2 + 1)*tan(e + f*x)/(6*d*f*tan(e + f*x) - 6*I*d*f) + 18*I*
C*a*b**2*log(tan(e + f*x)**2 + 1)/(6*d*f*tan(e + f*x) - 6*I*d*f) + 9*C*a*b*
*2*tan(e + f*x)**3/(6*d*f*tan(e + f*x) - 6*I*d*f) + 9*I*C*a*b**2*tan(e + f*
x)**2/(6*d*f*tan(e + f*x) - 6*I*d*f) + 27*I*C*a*b**2/(6*d*f*tan(e + f*x) -
6*I*d*f) + 15*C*b**3*f*x*tan(e + f*x)/(6*d*f*tan(e + f*x) - 6*I*d*f) - 15*I
*C*b**3*f*x/(6*d*f*tan(e + f*x) - 6*I*d*f) - 6*I*C*b**3*log(tan(e + f*x)**2
 + 1)*tan(e + f*x)/(6*d*f*tan(e + f*x) - 6*I*d*f) - 6*C*b**3*log(tan(e + f*
x)**2 + 1)/(6*d*f*tan(e + f*x) - 6*I*d*f) + 2*C*b**3*tan(e + f*x)**4/(6*d*f
*tan(e + f*x) - 6*I*d*f) + I*C*b**3*tan(e + f*x)**3/(6*d*f*tan(e + f*x) - 6
*I*d*f) - 9*C*b**3*tan(e + f*x)**2/(6*d*f*tan(e + f*x) - 6*I*d*f) - 15*C*b*
*3/(6*d*f*tan(e + f*x) - 6*I*d*f), Eq(c, -I*d)), (-3*I*A*a**3*f*x*tan(e + f
*x)/(6*d*f*tan(e + f*x) + 6*I*d*f) + 3*A*a**3*f*x/(6*d*f*tan(e + f*x) + 6*I
*d*f) - 3*I*A*a**3/(6*d*f*tan(e + f*x) + 6*I*d*f) + 9*A*a**2*b*f*x*tan(e +
f*x)/(6*d*f*tan(e + f*x) + 6*I*d*f) + 9*I*A*a**2*b*f*x/(6*d*f*tan(e + f*x)
 + 6*I*d*f) - 9*A*a**2*b/(6*d*f*tan(e + f*x) + 6*I*d*f) - 9*I*A*a*b**2*f*x*t
an(e + f*x)/(6*d*f*tan(e + f*x) + 6*I*d*f) + 9*A*a*b**2*f*x/(6*d*f*tan(e +
f*x) + 6*I*d*f) + 9*A*a*b**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(6*d*f*t
an(e + f*x) + 6*I*d*f) + 9*I*A*a*b**2*log(tan(e + f*x)**2 + 1)/(6*d*f*tan(e
 + f*x) + 6*I*d*f) + 9*I*A*a*b**2/(6*d*f*tan(e + f*x) + 6*I*d*f) - 9*A*b**3
*f*x*tan(e + f*x)/(6*d*f*tan(e + f*x) + 6*I*d*f) - 9*I*A*b**3*f*x/(6*d*f*ta
n(e + f*x) + 6*I*d*f) - 3*I*A*b**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(6
*d*f*tan(e + f*x) + 6*I*d*f) + 3*A*b**3*log(tan(e + f*x)**2 + 1)/(6*d*f*tan
(e + f*x) + 6*I*d*f) + 6*A*b**3*tan(e + f*x)**2/(6*d*f*tan(e + f*x) + 6*I*d
*f) + 9*A*b**3/(6*d*f*tan(e + f*x) + 6*I*d*f) + 3*B*a**3*f*x*tan(e + f*x)/(
6*d*f*tan(e + f*x) + 6*I*d*f) + 3*I*B*a**3*f*x/(6*d*f*tan(e + f*x) + 6*I*d*
f) - 3*B*a**3/(6*d*f*tan(e + f*x) + 6*I*d*f) - 9*I*B*a**2*b*f*x*tan(e + f*x
)/(6*d*f*tan(e + f*x) + 6*I*d*f) + 9*B*a**2*b*f*x/(6*d*f*tan(e + f*x) + 6*I
*d*f) + 9*B*a**2*b*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(6*d*f*tan(e + f*x
) + 6*I*d*f) + 9*I*B*a**2*b*log(tan(e + f*x)**2 + 1)/(6*d*f*tan(e + f*x) +
6*I*d*f) + 9*I*B*a**2*b/(6*d*f*tan(e + f*x) + 6*I*d*f) - 27*B*a*b**2*f*x*ta
n(e + f*x)/(6*d*f*tan(e + f*x) + 6*I*d*f) - 27*I*B*a*b**2*f*x/(6*d*f*tan(e
+ f*x) + 6*I*d*f) - 9*I*B*a*b**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(6*d
*f*tan(e + f*x) + 6*I*d*f) + 9*B*a*b**2*log(tan(e + f*x)**2 + 1)/(6*d*f*tan
(e + f*x) + 6*I*d*f) + 18*B*a*b**2*tan(e + f*x)**2/(6*d*f*tan(e + f*x) + 6*
I*d*f) + 27*B*a*b**2/(6*d*f*tan(e + f*x) + 6*I*d*f) + 9*I*B*b**3*f*x*tan(e
+ f*x)/(6*d*f*tan(e + f*x) + 6*I*d*f) - 9*B*b**3*f*x/(6*d*f*tan(e + f*x) +
6*I*d*f) - 6*B*b**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(6*d*f*tan(e + f*
x) + 6*I*d*f) - 6*I*B*b**3*log(tan(e + f*x)**2 + 1)/(6*d*f*tan(e + f*x) + 6
*I*d*f) + 3*B*b**3*tan(e + f*x)**3/(6*d*f*tan(e + f*x) + 6*I*d*f) - 3*I*B*b

```
**3*tan(e + f*x)**2/(6*d*f*tan(e + f*x) + 6*I*d*f) - 9*I*B*b**3/(6*d*f*tan(
e + f*x) + 6*I*d*f) - 3*I*C*a**3*f*x*tan(e + f*x)/(6*d*f*tan(e + f*x) + 6*I
*d*f) + 3*C*a**3*f*x/(6*d*f*tan(e + f*x) + 6*I*d*f) + 3*C*a**3*log(tan(e +
f*x)**2 + 1)*tan(e + f*x)/(6*d*f*tan(e + f*x) + 6*I*d*f) + 3*I*C*a**3*log(t
an(e + f*x)**2 + 1)/(6*d*f*tan(e + f*x) + 6*I*d*f) + 3*I*C*a**3/(6*d*f*tan(
e + f*x) + 6*I*d*f) - 27*C*a**2*b*f*x*tan(e + f*x)/(6*d*f*tan(e + f*x) + 6*
I*d*f) - 27*I*C*a**2*b*f*x/(6*d*f*tan(e + f*x) + 6*I*d*f) - 9*I*C*a**2*b*lo
g(tan(e + f*x)**2 + 1)*tan(e + f*x)/(6*d*f*tan(e + f*x) + 6*I*d*f) + 9*C*a*
*2*b*log(tan(e + f*x)**2 + 1)/(6*d*f*tan(e + f*x) + 6*I*d*f) + 18*C*a**2*b*
tan(e + f*x)**2/(6*d*f*tan(e + f*x) + 6*I*d*f) + 27*C*a**2*b/(6*d*f*tan(e +
 f*x) + 6*I*d*f) + 27*I*C*a*b**2*f*x*tan(e + f*x)/(6*d*f*tan(e + f*x) + 6*I
*d*f) - 27*C*a*b**2*f*x/(6*d*f*tan(e + f*x) + 6*I*d*f) - 18*C*a*b**2*log(ta
n(e + f*x)**2 + 1)*tan(e + f*x)/(6*d*f*tan(e + f*x) + 6*I*d*f) - 18*I*C*a*b
**2*log(tan(e + f*x)**2 + 1)/(6*d*f*tan(e + f*x) + 6*I*d*f) + 9*C*a*b**2*ta
n(e + f*x)**3/(6*d*f*tan(e + f*x) + 6*I*d*f) - 9*I*C*a*b**2*tan(e + f*x)**2
/(6*d*f*tan(e + f*x) + 6*I*d*f) - 27*I*C*a*b**2/(6*d*f*tan(e + f*x) + 6*I*d
*f) + 15*C*b**3*f*x*tan(e + f*x)/(6*d*f*tan(e + f*x) + 6*I*d*f) + 15*I*C*b*
*3*f*x/(6*d*f*tan(e + f*x) + 6*I*d*f) + 6*I*C*b**3*log(tan(e + f*x)**2 + 1)
*tan(e + f*x)/(6*d*f*tan(e + f*x) + 6*I*d*f) - 6*C*b**3*log(tan(e + f*x)**2
 + 1)/(6*d*f*tan(e + f*x) + 6*I*d*f) + 2*C*b**3*tan(e + f*x)**4/(6*d*f*tan(
e + f*x) + 6*I*d*f) - I*C*b**3*tan(e + f*x)**3/(6*d*f*tan(e + f*x) + 6*I*d*
f) - 9*C*b**3*tan(e + f*x)**2/(6*d*f*tan(e + f*x) + 6*I*d*f) - 15*C*b**3/(6
*d*f*tan(e + f*x) + 6*I*d*f), Eq(c, I*d)), (x*(a + b*tan(e))**3*(A + B*tan(
e) + C*tan(e)**2)/(c + d*tan(e)), Eq(f, 0)), (6*A*a**3*c*d**4*f*x/(6*c**2*d
**4*f + 6*d**6*f) + 6*A*a**3*d**5*log(c/d + tan(e + f*x))/(6*c**2*d**4*f +
6*d**6*f) - 3*A*a**3*d**5*log(tan(e + f*x)**2 + 1)/(6*c**2*d**4*f + 6*d**6*
f) - 18*A*a**2*b*c*d**4*log(c/d + tan(e + f*x))/(6*c**2*d**4*f + 6*d**6*f)
+ 9*A*a**2*b*c*d**4*log(tan(e + f*x)**2 + 1)/(6*c**2*d**4*f + 6*d**6*f) + 1
8*A*a**2*b*d**5*f*x/(6*c**2*d**4*f + 6*d**6*f) + 18*A*a*b**2*c**2*d**3*log(
c/d + tan(e + f*x))/(6*c**2*d**4*f + 6*d**6*f) - 18*A*a*b**2*c*d**4*f*x/(6*
c**2*d**4*f + 6*d**6*f) + 9*A*a*b**2*d**5*log(tan(e + f*x)**2 + 1)/(6*c**2*
d**4*f + 6*d**6*f) - 6*A*b**3*c**3*d**2*log(c/d + tan(e + f*x))/(6*c**2*d**
4*f + 6*d**6*f) + 6*A*b**3*c**2*d**3*tan(e + f*x)/(6*c**2*d**4*f + 6*d**6*f
) - 3*A*b**3*c*d**4*log(tan(e + f*x)**2 + 1)/(6*c**2*d**4*f + 6*d**6*f) - 6
*A*b**3*d**5*f*x/(6*c**2*d**4*f + 6*d**6*f) + 6*A*b**3*d**5*tan(e + f*x)/(6
*c**2*d**4*f + 6*d**6*f) - 6*B*a**3*c*d**4*log(c/d + tan(e + f*x))/(6*c**2*
d**4*f + 6*d**6*f) + 3*B*a**3*c*d**4*log(tan(e + f*x)**2 + 1)/(6*c**2*d**4*
f + 6*d**6*f) + 6*B*a**3*d**5*f*x/(6*c**2*d**4*f + 6*d**6*f) + 18*B*a**2*b*
c**2*d**3*log(c/d + tan(e + f*x))/(6*c**2*d**4*f + 6*d**6*f) - 18*B*a**2*b*
c*d**4*f*x/(6*c**2*d**4*f + 6*d**6*f) + 9*B*a**2*b*d**5*log(tan(e + f*x)**2
 + 1)/(6*c**2*d**4*f + 6*d**6*f) - 18*B*a*b**2*c**3*d**2*log(c/d + tan(e +
f*x))/(6*c**2*d**4*f + 6*d**6*f) + 18*B*a*b**2*c**2*d**3*tan(e + f*x)/(6*c*
*2*d**4*f + 6*d**6*f) - 9*B*a*b**2*c*d**4*log(tan(e + f*x)**2 + 1)/(6*c**2*
d**4*f + 6*d**6*f) - 18*B*a*b**2*d**5*f*x/(6*c**2*d**4*f + 6*d**6*f) + 18*B
*a*b**2*d**5*tan(e + f*x)/(6*c**2*d**4*f + 6*d**6*f) + 6*B*b**3*c**4*d*log(
```

```
c/d + tan(e + f*x))/(6*c**2*d**4*f + 6*d**6*f) - 6*B*b**3*c**3*d**2*tan(e +
 f*x)/(6*c**2*d**4*f + 6*d**6*f) + 3*B*b**3*c**2*d**3*tan(e + f*x)**2/(6*c*
*2*d**4*f + 6*d**6*f) + 6*B*b**3*c*d**4*f*x/(6*c**2*d**4*f + 6*d**6*f) - 6*
B*b**3*c*d**4*tan(e + f*x)/(6*c**2*d**4*f + 6*d**6*f) - 3*B*b**3*d**5*log(t
an(e + f*x)**2 + 1)/(6*c**2*d**4*f + 6*d**6*f) + 3*B*b**3*d**5*tan(e + f*x)
**2/(6*c**2*d**4*f + 6*d**6*f) + 6*C*a**3*c**2*d**3*log(c/d + tan(e + f*x))
/(6*c**2*d**4*f + 6*d**6*f) - 6*C*a**3*c*d**4*f*x/(6*c**2*d**4*f + 6*d**6*f
) + 3*C*a**3*d**5*log(tan(e + f*x)**2 + 1)/(6*c**2*d**4*f + 6*d**6*f) - 18*
C*a**2*b*c**3*d**2*log(c/d + tan(e + f*x))/(6*c**2*d**4*f + 6*d**6*f) + 18*
C*a**2*b*c**2*d**3*tan(e + f*x)/(6*c**2*d**4*f + 6*d**6*f) - 9*C*a**2*b*c*d
**4*log(tan(e + f*x)**2 + 1)/(6*c**2*d**4*f + 6*d**6*f) - 18*C*a**2*b*d**5*
f*x/(6*c**2*d**4*f + 6*d**6*f) + 18*C*a**2*b*d**5*tan(e + f*x)/(6*c**2*d**4
*f + 6*d**6*f) + 18*C*a*b**2*c**4*d*log(c/d + tan(e + f*x))/(6*c**2*d**4*f
+ 6*d**6*f) - 18*C*a*b**2*c**3*d**2*tan(e + f*x)/(6*c**2*d**4*f + 6*d**6*f)
 + 9*C*a*b**2*c**2*d**3*tan(e + f*x)**2/(6*c**2*d**4*f + 6*d**6*f) + 18*C*a
*b**2*c*d**4*f*x/(6*c**2*d**4*f + 6*d**6*f) - 18*C*a*b**2*c*d**4*tan(e + f*
x)/(6*c**2*d**4*f + 6*d**6*f) - 9*C*a*b**2*d**5*log(tan(e + f*x)**2 + 1)/(6
*c**2*d**4*f + 6*d**6*f) + 9*C*a*b**2*d**5*tan(e + f*x)**2/(6*c**2*d**4*f +
 6*d**6*f) - 6*C*b**3*c**5*log(c/d + tan(e + f*x))/(6*c**2*d**4*f + 6*d**6*
f) + 6*C*b**3*c**4*d*tan(e + f*x)/(6*c**2*d**4*f + 6*d**6*f) - 3*C*b**3*c**
3*d**2*tan(e + f*x)**2/(6*c**2*d**4*f + 6*d**6*f) + 2*C*b**3*c**2*d**3*tan(
e + f*x)**3/(6*c**2*d**4*f + 6*d**6*f) + 3*C*b**3*c*d**4*log(tan(e + f*x)**
2 + 1)/(6*c**2*d**4*f + 6*d**6*f) - 3*C*b**3*c*d**4*tan(e + f*x)**2/(6*c**2
*d**4*f + 6*d**6*f) + 6*C*b**3*d**5*f*x/(6*c**2*d**4*f + 6*d**6*f) + 2*C*b*
*3*d**5*tan(e + f*x)**3/(6*c**2*d**4*f + 6*d**6*f) - 6*C*b**3*d**5*tan(e +
f*x)/(6*c**2*d**4*f + 6*d**6*f), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.47 (sec) , antiderivative size = 445, normalized size of antiderivative = 1.32

$$\int \frac{(a+b\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{c+d\tan(e+fx)}\,dx$$

$$= \frac{\frac{6\,(((A-C)a^3-3\,Ba^2b-3\,(A-C)ab^2+Bb^3)c+(Ba^3+3\,(A-C)a^2b-3\,Bab^2-(A-C)b^3)d)(fx+e)}{c^2+d^2} - \frac{6\,(Cb^3c^5-Aa^3d^5-(3\,Cab^2+Bb^3)c^4d+(3\,Ca^2}{}}{}$$

```
[In] integrate((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
)),x, algorithm="maxima")
```

```
[Out] 1/6*(6*(((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b^3)*c + (B*a^3 + 3*
(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*d)*(f*x + e)/(c^2 + d^2) - 6*(C*b^
3*c^5 - A*a^3*d^5 - (3*C*a*b^2 + B*b^3)*c^4*d + (3*C*a^2*b + 3*B*a*b^2 + A*
b^3)*c^3*d^2 - (C*a^3 + 3*B*a^2*b + 3*A*a*b^2)*c^2*d^3 + (B*a^3 + 3*A*a^2*b
```

```
)*c*d^4)*log(d*tan(f*x + e) + c)/(c^2*d^4 + d^6) + 3*((B*a^3 + 3*(A - C)*a^
2*b - 3*B*a*b^2 - (A - C)*b^3)*c - ((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b
^2 + B*b^3)*d)*log(tan(f*x + e)^2 + 1)/(c^2 + d^2) + (2*C*b^3*d^2*tan(f*x +
 e)^3 - 3*(C*b^3*c*d - (3*C*a*b^2 + B*b^3)*d^2)*tan(f*x + e)^2 + 6*(C*b^3*c
^2 - (3*C*a*b^2 + B*b^3)*c*d + (3*C*a^2*b + 3*B*a*b^2 + (A - C)*b^3)*d^2)*t
an(f*x + e))/d^3)/f
```

Giac [A] (verification not implemented)

none

Time = 0.96 (sec) , antiderivative size = 559, normalized size of antiderivative = 1.66

$$\int \frac{(a+b\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{c+d\tan(e+fx)}\,dx$$
$$=\frac{\frac{6\,(Aa^3c-Ca^3c-3\,Ba^2bc-3\,Aab^2c+3\,Cab^2c+Bb^3c+Ba^3d+3\,Aa^2bd-3\,Ca^2bd-3\,Bab^2d-Ab^3d+Cb^3d)(fx+e)}{c^2+d^2}+\frac{3\,(Ba^3c+3\,Aa^2bc-3\,Ca^2bc-3}{}}{}$$

```
[In] integrate((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
)),x, algorithm="giac")
```

```
[Out] 1/6*(6*(A*a^3*c - C*a^3*c - 3*B*a^2*b*c - 3*A*a*b^2*c + 3*C*a*b^2*c + B*b^3
*c + B*a^3*d + 3*A*a^2*b*d - 3*C*a^2*b*d - 3*B*a*b^2*d - A*b^3*d + C*b^3*d)
*(f*x + e)/(c^2 + d^2) + 3*(B*a^3*c + 3*A*a^2*b*c - 3*C*a^2*b*c - 3*B*a*b^2
*c - A*b^3*c + C*b^3*c - A*a^3*d + C*a^3*d + 3*B*a^2*b*d + 3*A*a*b^2*d - 3*
C*a*b^2*d - B*b^3*d)*log(tan(f*x + e)^2 + 1)/(c^2 + d^2) - 6*(C*b^3*c^5 - 3
*C*a*b^2*c^4*d - B*b^3*c^4*d + 3*C*a^2*b*c^3*d^2 + 3*B*a*b^2*c^3*d^2 + A*b^
3*c^3*d^2 - C*a^3*c^2*d^3 - 3*B*a^2*b*c^2*d^3 - 3*A*a*b^2*c^2*d^3 + B*a^3*c
*d^4 + 3*A*a^2*b*c*d^4 - A*a^3*d^5)*log(abs(d*tan(f*x + e) + c))/(c^2*d^4 +
 d^6) + (2*C*b^3*d^2*tan(f*x + e)^3 - 3*C*b^3*c*d*tan(f*x + e)^2 + 9*C*a*b^
2*d^2*tan(f*x + e)^2 + 3*B*b^3*d^2*tan(f*x + e)^2 + 6*C*b^3*c^2*tan(f*x + e
) - 18*C*a*b^2*c*d*tan(f*x + e) - 6*B*b^3*c*d*tan(f*x + e) + 18*C*a^2*b*d^2
*tan(f*x + e) + 18*B*a*b^2*d^2*tan(f*x + e) + 6*A*b^3*d^2*tan(f*x + e) - 6*
C*b^3*d^2*tan(f*x + e))/d^3)/f
```

Mupad [B] (verification not implemented)

Time = 12.13 (sec) , antiderivative size = 508, normalized size of antiderivative = 1.51

$$\int \frac{(a+b\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{c+d\tan(e+fx)}\,dx$$
$$= \frac{\tan(e+f\,x)^2\,\left(\frac{B\,b^3+3\,C\,a\,b^2}{2\,d}-\frac{C\,b^3\,c}{2\,d^2}\right)}{f} - \frac{\tan(e+f\,x)\,\left(\frac{c\,\left(\frac{B\,b^3+3\,C\,a\,b^2}{d}-\frac{C\,b^3\,c}{d^2}\right)}{d}-\frac{3\,C\,a^2\,b+3\,B\,a\,b^2+A\,b^3}{d}+\frac{C\,b^3}{d}\right)}{f} - \frac{\ln\left(c+d\tan(e+f\,x)\right)\,\left(d^4\,\left(B\,c\,a^3+3\,A\,b\,c\,a^2\right)-d^3\,\left(C\,a^3\,c^2+3\,B\,a^2\,b\,c^2+3\,A\,a\,b^2\,c^2\right)+d^2\,\left(3\,C\,a^2\,b\,c^3+\right.\right.}{f\,\left(c^2\,d^4+d^6\right)} - \frac{\ln\left(\tan(e+f\,x)+1\mathrm{i}\right)\,\left(A\,a^3+A\,b^3\,1\mathrm{i}-B\,a^3\,1\mathrm{i}+B\,b^3-C\,a^3-C\,b^3\,1\mathrm{i}-3\,A\,a\,b^2-A\,a^2\,b\,3\mathrm{i}+B\,a\,b^2\,3\mathrm{i}-\right.}{2\,f\,\left(d+c\,1\mathrm{i}\right)} - \frac{\ln\left(\tan(e+f\,x)-\mathrm{i}\right)\,\left(A\,b^3-B\,a^3-C\,b^3-3\,A\,a^2\,b+3\,B\,a\,b^2+3\,C\,a^2\,b+A\,a^3\,1\mathrm{i}+B\,b^3\,1\mathrm{i}-C\,a^3\,1\mathrm{i}-\right.}{2\,f\,\left(c+d\,1\mathrm{i}\right)} + \frac{C\,b^3\tan(e+f\,x)^3}{3\,d\,f}$$

```
[In] int(((a + b*tan(e + f*x))^3*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d
*tan(e + f*x)),x)
```

```
[Out] (tan(e + f*x)^2*((B*b^3 + 3*C*a*b^2)/(2*d) - (C*b^3*c)/(2*d^2)))/f - (tan(e
 + f*x)*((c*((B*b^3 + 3*C*a*b^2)/d - (C*b^3*c)/d^2))/d - (A*b^3 + 3*B*a*b^2
 + 3*C*a^2*b)/d + (C*b^3)/d))/f - (log(c + d*tan(e + f*x))*(d^4*(B*a^3*c +
3*A*a^2*b*c) - d^3*(C*a^3*c^2 + 3*A*a*b^2*c^2 + 3*B*a^2*b*c^2) + d^2*(A*b^3
*c^3 + 3*B*a*b^2*c^3 + 3*C*a^2*b*c^3) - d*(B*b^3*c^4 + 3*C*a*b^2*c^4) - A*a
^3*d^5 + C*b^3*c^5))/(f*(d^6 + c^2*d^4)) - (log(tan(e + f*x) + 1i)*(A*a^3 +
 A*b^3*1i - B*a^3*1i + B*b^3 - C*a^3 - C*b^3*1i - 3*A*a*b^2 - A*a^2*b*3i +
B*a*b^2*3i - 3*B*a^2*b + 3*C*a*b^2 + C*a^2*b*3i))/(2*f*(c*1i + d)) - (log(t
an(e + f*x) - 1i)*(A*a^3*1i + A*b^3 - B*a^3 + B*b^3*1i - C*a^3*1i - C*b^3 -
 A*a*b^2*3i - 3*A*a^2*b + 3*B*a*b^2 - B*a^2*b*3i + C*a*b^2*3i + 3*C*a^2*b))
/(2*f*(c + d*1i)) + (C*b^3*tan(e + f*x)^3)/(3*d*f)
```

3.71 $\int \frac{(a+b\tan(e+fx))^2(A+B\tan(e+fx)+C\tan^2(e+fx))}{c+d\tan(e+fx)}\,dx$

Optimal result 683
Rubi [A] (verified) 683
Mathematica [C] (verified) 686
Maple [A] (verified) 687
Fricas [A] (verification not implemented) 687
Sympy [C] (verification not implemented) 688
Maxima [A] (verification not implemented) 690
Giac [A] (verification not implemented) 691
Mupad [B] (verification not implemented) 692

Optimal result

Integrand size = 45, antiderivative size = 236

$$\begin{aligned}
&\int \frac{(a+b\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{c+d\tan(e+fx)}\,dx \\
&= \frac{(a^2(Ac-cC+Bd)-b^2(Ac-cC+Bd)-2ab(Bc-(A-C)d))\,x}{c^2+d^2} \\
&\quad - \frac{(2ab(Ac-cC+Bd)+a^2(Bc-(A-C)d)-b^2(Bc-(A-C)d))\log(\cos(e+fx))}{(c^2+d^2)\,f} \\
&\quad + \frac{(bc-ad)^2\,(c^2C-Bcd+Ad^2)\log(c+d\tan(e+fx))}{d^3\,(c^2+d^2)\,f} \\
&\quad - \frac{b(bcC-bBd-aCd)\tan(e+fx)}{d^2f} + \frac{C(a+b\tan(e+fx))^2}{2df}
\end{aligned}$$

```
[Out] (a^2*(A*c+B*d-C*c)-b^2*(A*c+B*d-C*c)-2*a*b*(B*c-(A-C)*d))*x/(c^2+d^2)-(2*a*
b*(A*c+B*d-C*c)+a^2*(B*c-(A-C)*d)-b^2*(B*c-(A-C)*d))*ln(cos(f*x+e))/(c^2+d^
2)/f+(-a*d+b*c)^2*(A*d^2-B*c*d+C*c^2)*ln(c+d*tan(f*x+e))/d^3/(c^2+d^2)/f-b*
(-B*b*d-C*a*d+C*b*c)*tan(f*x+e)/d^2/f+1/2*C*(a+b*tan(f*x+e))^2/d/f
```

Rubi [A] (verified)

Time = 0.88 (sec) , antiderivative size = 236, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.133, Rules used

$= \{3728, 3718, 3707, 3698, 31, 3556\}$

$$\int \frac{(a+b\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{c+d\tan(e+fx)}\,dx$$
$$= -\frac{\log(\cos(e+fx))\,(a^2(Bc-d(A-C))+2ab(Ac+Bd-cC)-b^2(Bc-d(A-C)))}{f\,(c^2+d^2)}$$
$$+\frac{x(a^2(Ac+Bd-cC)-2ab(Bc-d(A-C))-b^2(Ac+Bd-cC))}{c^2+d^2}$$
$$+\frac{(bc-ad)^2\,(Ad^2-Bcd+c^2C)\log(c+d\tan(e+fx))}{d^3f\,(c^2+d^2)}$$
$$-\frac{b\tan(e+fx)(-aCd-bBd+bcC)}{d^2f}+\frac{C(a+b\tan(e+fx))^2}{2df}$$

```
[In] Int[((a + b*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c + d
*Tan[e + f*x]),x]
```

```
[Out] ((a^2*(A*c - c*C + B*d) - b^2*(A*c - c*C + B*d) - 2*a*b*(B*c - (A - C)*d))*
x)/(c^2 + d^2) - ((2*a*b*(A*c - c*C + B*d) + a^2*(B*c - (A - C)*d) - b^2*(B
*c - (A - C)*d))*Log[Cos[e + f*x]])/((c^2 + d^2)*f) + ((b*c - a*d)^2*(c^2*C
 - B*c*d + A*d^2)*Log[c + d*Tan[e + f*x]])/(d^3*(c^2 + d^2)*f) - (b*(b*c*C
- b*B*d - a*C*d)*Tan[e + f*x])/(d^2*f) + (C*(a + b*Tan[e + f*x])^2)/(2*d*f)
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3698

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3707

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[(a*A + b*B - a
*C)*(x/(a^2 + b^2)), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
```

```
0]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \frac{C(a+b\tan(e+fx))^2}{2df} \\
&\quad + \frac{\int \frac{(a+b\tan(e+fx))(-2(bcC-aAd)+2(Ab+aB-bC)d\tan(e+fx)-2(bcC-bBd-aCd)\tan^2(e+fx))}{c+d\tan(e+fx)}\,dx}{2d} \\
&= -\frac{b(bcC-bBd-aCd)\tan(e+fx)}{d^2f} + \frac{C(a+b\tan(e+fx))^2}{2df} \\
&\quad - \frac{\int \frac{2(2abcCd-a^2Ad^2-b^2c(cC-Bd))-2(a^2B-b^2B+2ab(A-C))d^2\tan(e+fx)-2(a^2Cd^2-2abd(cC-Bd)+b^2(c^2C-Bcd+(A-C)d}{c+d\tan(e+fx)}}{2d^2}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{(a^2(Ac-cC+Bd)-b^2(Ac-cC+Bd)-2ab(Bc-(A-C)d))\,x}{c^2+d^2} \\
&\quad - \frac{b(bcC-bBd-aCd)\tan(e+fx)}{d^2f} + \frac{C(a+b\tan(e+fx))^2}{2df} \\
&\quad + \frac{((bc-ad)^2\,(c^2C-Bcd+Ad^2))\int \frac{1+\tan^2(e+fx)}{c+d\tan(e+fx)}\,dx}{d^2\,(c^2+d^2)} \\
&\quad + \frac{(2ab(Ac-cC+Bd)+a^2(Bc-(A-C)d)-b^2(Bc-(A-C)d))\int \tan(e+fx)\,dx}{c^2+d^2} \\
&= \frac{(a^2(Ac-cC+Bd)-b^2(Ac-cC+Bd)-2ab(Bc-(A-C)d))\,x}{c^2+d^2} \\
&\quad - \frac{(2ab(Ac-cC+Bd)+a^2(Bc-(A-C)d)-b^2(Bc-(A-C)d))\log(\cos(e+fx))}{(c^2+d^2)\,f} \\
&\quad - \frac{b(bcC-bBd-aCd)\tan(e+fx)}{d^2f} + \frac{C(a+b\tan(e+fx))^2}{2df} \\
&\quad + \frac{((bc-ad)^2\,(c^2C-Bcd+Ad^2))\,\mathrm{Subst}\left(\int \frac{1}{c+x}\,dx,x,d\tan(e+fx)\right)}{d^3\,(c^2+d^2)\,f} \\
&= \frac{(a^2(Ac-cC+Bd)-b^2(Ac-cC+Bd)-2ab(Bc-(A-C)d))\,x}{c^2+d^2} \\
&\quad - \frac{(2ab(Ac-cC+Bd)+a^2(Bc-(A-C)d)-b^2(Bc-(A-C)d))\log(\cos(e+fx))}{(c^2+d^2)\,f} \\
&\quad + \frac{(bc-ad)^2\,(c^2C-Bcd+Ad^2)\log(c+d\tan(e+fx))}{d^3\,(c^2+d^2)\,f} \\
&\quad - \frac{b(bcC-bBd-aCd)\tan(e+fx)}{d^2f} + \frac{C(a+b\tan(e+fx))^2}{2df}
\end{aligned}
$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 3.23 (sec) , antiderivative size = 190, normalized size of antiderivative = 0.81

$$
\int \frac{(a+b\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{c+d\tan(e+fx)}\,dx
$$

$$
= \frac{\frac{(a+ib)^2(-iA+B+iC)d\log(i-\tan(e+fx))}{c+id} + \frac{(a-ib)^2(iA+B-iC)d\log(i+\tan(e+fx))}{c-id} + \frac{2(bc-ad)^2(c^2C-Bcd+Ad^2)\log(c+d\tan(e+fx))}{d^2(c^2+d^2)} + 2}{2df}
$$

```
[In] Integrate[((a + b*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(c + d*Tan[e + f*x]),x]
[Out] (((a + I*b)^2*((-I)*A + B + I*C)*d*Log[I - Tan[e + f*x]])/(c + I*d) + ((a -
 I*b)^2*(I*A + B - I*C)*d*Log[I + Tan[e + f*x]])/(c - I*d) + (2*(b*c - a*d)
^2*(c^2*C - B*c*d + A*d^2)*Log[c + d*Tan[e + f*x]])/(d^2*(c^2 + d^2)) + (2*
b*(-(b*c*C) + b*B*d + a*C*d)*Tan[e + f*x])/d + C*(a + b*Tan[e + f*x])^2)/(2
*d*f)
```

Maple [A] (verified)

Time = 0.21 (sec) , antiderivative size = 317, normalized size of antiderivative = 1.34

method	result
derivativedivides	$\frac{\frac{b\left(\frac{\tan(fx+e)^2 Cbd}{2}+\tan(fx+e)bdB+2\tan(fx+e)Cad-\tan(fx+e)Cbc\right)}{d^2}+\frac{\left(-A\,a^2d+2Aabc+A\,b^2d+B\,a^2c+2Babd-B\,b^2c+C\,a^2d-2C\right.}{2}}{}$
default	$\frac{\frac{b\left(\frac{\tan(fx+e)^2 Cbd}{2}+\tan(fx+e)bdB+2\tan(fx+e)Cad-\tan(fx+e)Cbc\right)}{d^2}+\frac{\left(-A\,a^2d+2Aabc+A\,b^2d+B\,a^2c+2Babd-B\,b^2c+C\,a^2d-2C\right.}{2}}{}$
norman	$\frac{(A\,a^2c+2Aabd-A\,b^2c+B\,a^2d-2Babc-B\,b^2d-C\,a^2c-2Cabd+C\,b^2c)x}{c^2+d^2}+\frac{b(bdB+2Cad-Cbc)\tan(fx+e)}{d^2f}+\frac{C\,b^2\tan}{2}$
parallelrisch	$-\frac{4Bxabc\,d^3f-4C\tan(fx+e)ab\,c^2d^2-2Ax\,a^2c\,d^3f-4Axab\,d^4f+2Ax\,b^2c\,d^3f+2Cx\,a^2c\,d^3f+4Cxab\,d^4f-2Cx\,b^2c\,d^3f}{}$
risch	Expression too large to display

```
[In] int((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e)),x,m
ethod=_RETURNVERBOSE)
```

```
[Out] 1/f*(b/d^2*(1/2*tan(f*x+e)^2*C*b*d+tan(f*x+e)*b*d*B+2*tan(f*x+e)*C*a*d-tan(
f*x+e)*C*b*c)+1/(c^2+d^2)*(1/2*(-A*a^2*d+2*A*a*b*c+A*b^2*d+B*a^2*c+2*B*a*b*
d-B*b^2*c+C*a^2*d-2*C*a*b*c-C*b^2*d)*ln(1+tan(f*x+e)^2)+(A*a^2*c+2*A*a*b*d-
A*b^2*c+B*a^2*d-2*B*a*b*c-B*b^2*d-C*a^2*c-2*C*a*b*d+C*b^2*c)*arctan(tan(f*x
+e)))+1/d^3*(A*a^2*d^4-2*A*a*b*c*d^3+A*b^2*c^2*d^2-B*a^2*c*d^3+2*B*a*b*c^2*
d^2-B*b^2*c^3*d+C*a^2*c^2*d^2-2*C*a*b*c^3*d+C*b^2*c^4)/(c^2+d^2)*ln(c+d*tan
(f*x+e)))
```

Fricas [A] (verification not implemented)

none

Time = 0.47 (sec) , antiderivative size = 390, normalized size of antiderivative = 1.65

$$\int \frac{(a+b\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{c+d\tan(e+fx)}\,dx$$
$$=\frac{2\,(((A-C)a^2-2\,Bab-(A-C)b^2)cd^3+(Ba^2+2\,(A-C)ab-Bb^2)d^4)fx+(Cb^2c^2d^2+Cb^2d^4)\tan(f}{}$$

```
[In] integrate((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
)),x, algorithm="fricas")
```

```
[Out] 1/2*(2*(((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*c*d^3 + (B*a^2 + 2*(A - C)*a*
b - B*b^2)*d^4)*f*x + (C*b^2*c^2*d^2 + C*b^2*d^4)*tan(f*x + e)^2 + (C*b^2*c
^4 + A*a^2*d^4 - (2*C*a*b + B*b^2)*c^3*d + (C*a^2 + 2*B*a*b + A*b^2)*c^2*d^
```

```
2 - (B*a^2 + 2*A*a*b)*c*d^3)*log((d^2*tan(f*x + e)^2 + 2*c*d*tan(f*x + e) +
 c^2)/(tan(f*x + e)^2 + 1)) - (C*b^2*c^4 - (2*C*a*b + B*b^2)*c^3*d + (C*a^2
 + 2*B*a*b + A*b^2)*c^2*d^2 - (2*C*a*b + B*b^2)*c*d^3 + (C*a^2 + 2*B*a*b +
(A - C)*b^2)*d^4)*log(1/(tan(f*x + e)^2 + 1)) - 2*(C*b^2*c^3*d + C*b^2*c*d^
3 - (2*C*a*b + B*b^2)*c^2*d^2 - (2*C*a*b + B*b^2)*d^4)*tan(f*x + e))/((c^2*
d^3 + d^5)*f)
```

Sympy [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 2.23 (sec) , antiderivative size = 4444, normalized size of antiderivative = 18.83

$$\int \frac{(a+b\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{c+d\tan(e+fx)}\,dx=\text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))**2*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x
+e)),x)
```

```
[Out] Piecewise((zoo*x*(a + b*tan(e))**2*(A + B*tan(e) + C*tan(e)**2)/tan(e), Eq(
c, 0) & Eq(d, 0) & Eq(f, 0)), ((A*a**2*x + A*a*b*log(tan(e + f*x)**2 + 1)/f
 - A*b**2*x + A*b**2*tan(e + f*x)/f + B*a**2*log(tan(e + f*x)**2 + 1)/(2*f)
 - 2*B*a*b*x + 2*B*a*b*tan(e + f*x)/f - B*b**2*log(tan(e + f*x)**2 + 1)/(2*
f) + B*b**2*tan(e + f*x)**2/(2*f) - C*a**2*x + C*a**2*tan(e + f*x)/f - C*a*
b*log(tan(e + f*x)**2 + 1)/f + C*a*b*tan(e + f*x)**2/f + C*b**2*x + C*b**2*
tan(e + f*x)**3/(3*f) - C*b**2*tan(e + f*x)/f)/c, Eq(d, 0)), (I*A*a**2*f*x*
tan(e + f*x)/(2*d*f*tan(e + f*x) - 2*I*d*f) + A*a**2*f*x/(2*d*f*tan(e + f*x
) - 2*I*d*f) + I*A*a**2/(2*d*f*tan(e + f*x) - 2*I*d*f) + 2*A*a*b*f*x*tan(e
+ f*x)/(2*d*f*tan(e + f*x) - 2*I*d*f) - 2*I*A*a*b*f*x/(2*d*f*tan(e + f*x) -
 2*I*d*f) - 2*A*a*b/(2*d*f*tan(e + f*x) - 2*I*d*f) + I*A*b**2*f*x*tan(e + f
*x)/(2*d*f*tan(e + f*x) - 2*I*d*f) + A*b**2*f*x/(2*d*f*tan(e + f*x) - 2*I*d
*f) + A*b**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*d*f*tan(e + f*x) - 2*
I*d*f) - I*A*b**2*log(tan(e + f*x)**2 + 1)/(2*d*f*tan(e + f*x) - 2*I*d*f) -
 I*A*b**2/(2*d*f*tan(e + f*x) - 2*I*d*f) + B*a**2*f*x*tan(e + f*x)/(2*d*f*t
an(e + f*x) - 2*I*d*f) - I*B*a**2*f*x/(2*d*f*tan(e + f*x) - 2*I*d*f) - B*a*
*2/(2*d*f*tan(e + f*x) - 2*I*d*f) + 2*I*B*a*b*f*x*tan(e + f*x)/(2*d*f*tan(e
 + f*x) - 2*I*d*f) + 2*B*a*b*f*x/(2*d*f*tan(e + f*x) - 2*I*d*f) + 2*B*a*b*l
og(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*d*f*tan(e + f*x) - 2*I*d*f) - 2*I*B
*a*b*log(tan(e + f*x)**2 + 1)/(2*d*f*tan(e + f*x) - 2*I*d*f) - 2*I*B*a*b/(2
*d*f*tan(e + f*x) - 2*I*d*f) - 3*B*b**2*f*x*tan(e + f*x)/(2*d*f*tan(e + f*x
) - 2*I*d*f) + 3*I*B*b**2*f*x/(2*d*f*tan(e + f*x) - 2*I*d*f) + I*B*b**2*log
(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*d*f*tan(e + f*x) - 2*I*d*f) + B*b**2*
log(tan(e + f*x)**2 + 1)/(2*d*f*tan(e + f*x) - 2*I*d*f) + 2*B*b**2*tan(e +
f*x)**2/(2*d*f*tan(e + f*x) - 2*I*d*f) + 3*B*b**2/(2*d*f*tan(e + f*x) - 2*I
*d*f) + I*C*a**2*f*x*tan(e + f*x)/(2*d*f*tan(e + f*x) - 2*I*d*f) + C*a**2*f
*x/(2*d*f*tan(e + f*x) - 2*I*d*f) + C*a**2*log(tan(e + f*x)**2 + 1)*tan(e +
```

f*x)/(2*d*f*tan(e + f*x) - 2*I*d*f) - I*C*a**2*log(tan(e + f*x)**2 + 1)/(2
*d*f*tan(e + f*x) - 2*I*d*f) - I*C*a**2/(2*d*f*tan(e + f*x) - 2*I*d*f) - 6*
C*a*b*f*x*tan(e + f*x)/(2*d*f*tan(e + f*x) - 2*I*d*f) + 6*I*C*a*b*f*x/(2*d*
f*tan(e + f*x) - 2*I*d*f) + 2*I*C*a*b*log(tan(e + f*x)**2 + 1)*tan(e + f*x)
/(2*d*f*tan(e + f*x) - 2*I*d*f) + 2*C*a*b*log(tan(e + f*x)**2 + 1)/(2*d*f*t
an(e + f*x) - 2*I*d*f) + 4*C*a*b*tan(e + f*x)**2/(2*d*f*tan(e + f*x) - 2*I*
d*f) + 6*C*a*b/(2*d*f*tan(e + f*x) - 2*I*d*f) - 3*I*C*b**2*f*x*tan(e + f*x)
/(2*d*f*tan(e + f*x) - 2*I*d*f) - 3*C*b**2*f*x/(2*d*f*tan(e + f*x) - 2*I*d*
f) - 2*C*b**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*d*f*tan(e + f*x) - 2
*I*d*f) + 2*I*C*b**2*log(tan(e + f*x)**2 + 1)/(2*d*f*tan(e + f*x) - 2*I*d*f
) + C*b**2*tan(e + f*x)**3/(2*d*f*tan(e + f*x) - 2*I*d*f) + I*C*b**2*tan(e
+ f*x)**2/(2*d*f*tan(e + f*x) - 2*I*d*f) + 3*I*C*b**2/(2*d*f*tan(e + f*x) -
2*I*d*f), Eq(c, -I*d)), (-I*A*a**2*f*x*tan(e + f*x)/(2*d*f*tan(e + f*x) +
2*I*d*f) + A*a**2*f*x/(2*d*f*tan(e + f*x) + 2*I*d*f) - I*A*a**2/(2*d*f*tan(
e + f*x) + 2*I*d*f) + 2*A*a*b*f*x*tan(e + f*x)/(2*d*f*tan(e + f*x) + 2*I*d*
f) + 2*I*A*a*b*f*x/(2*d*f*tan(e + f*x) + 2*I*d*f) - 2*A*a*b/(2*d*f*tan(e +
f*x) + 2*I*d*f) - I*A*b**2*f*x*tan(e + f*x)/(2*d*f*tan(e + f*x) + 2*I*d*f)
+ A*b**2*f*x/(2*d*f*tan(e + f*x) + 2*I*d*f) + A*b**2*log(tan(e + f*x)**2 +
1)*tan(e + f*x)/(2*d*f*tan(e + f*x) + 2*I*d*f) + I*A*b**2*log(tan(e + f*x)*
*2 + 1)/(2*d*f*tan(e + f*x) + 2*I*d*f) + I*A*b**2/(2*d*f*tan(e + f*x) + 2*I
*d*f) + B*a**2*f*x*tan(e + f*x)/(2*d*f*tan(e + f*x) + 2*I*d*f) + I*B*a**2*f
*x/(2*d*f*tan(e + f*x) + 2*I*d*f) - B*a**2/(2*d*f*tan(e + f*x) + 2*I*d*f) -
2*I*B*a*b*f*x*tan(e + f*x)/(2*d*f*tan(e + f*x) + 2*I*d*f) + 2*B*a*b*f*x/(2
*d*f*tan(e + f*x) + 2*I*d*f) + 2*B*a*b*log(tan(e + f*x)**2 + 1)*tan(e + f*x
)/(2*d*f*tan(e + f*x) + 2*I*d*f) + 2*I*B*a*b*log(tan(e + f*x)**2 + 1)/(2*d*
f*tan(e + f*x) + 2*I*d*f) + 2*I*B*a*b/(2*d*f*tan(e + f*x) + 2*I*d*f) - 3*B*
b**2*f*x*tan(e + f*x)/(2*d*f*tan(e + f*x) + 2*I*d*f) - 3*I*B*b**2*f*x/(2*d*
f*tan(e + f*x) + 2*I*d*f) - I*B*b**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/
(2*d*f*tan(e + f*x) + 2*I*d*f) + B*b**2*log(tan(e + f*x)**2 + 1)/(2*d*f*tan
(e + f*x) + 2*I*d*f) + 2*B*b**2*tan(e + f*x)**2/(2*d*f*tan(e + f*x) + 2*I*d
*f) + 3*B*b**2/(2*d*f*tan(e + f*x) + 2*I*d*f) - I*C*a**2*f*x*tan(e + f*x)/(
2*d*f*tan(e + f*x) + 2*I*d*f) + C*a**2*f*x/(2*d*f*tan(e + f*x) + 2*I*d*f) +
C*a**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*d*f*tan(e + f*x) + 2*I*d*f
) + I*C*a**2*log(tan(e + f*x)**2 + 1)/(2*d*f*tan(e + f*x) + 2*I*d*f) + I*C*
a**2/(2*d*f*tan(e + f*x) + 2*I*d*f) - 6*C*a*b*f*x*tan(e + f*x)/(2*d*f*tan(e
+ f*x) + 2*I*d*f) - 6*I*C*a*b*f*x/(2*d*f*tan(e + f*x) + 2*I*d*f) - 2*I*C*a
*b*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*d*f*tan(e + f*x) + 2*I*d*f) + 2
*C*a*b*log(tan(e + f*x)**2 + 1)/(2*d*f*tan(e + f*x) + 2*I*d*f) + 4*C*a*b*ta
n(e + f*x)**2/(2*d*f*tan(e + f*x) + 2*I*d*f) + 6*C*a*b/(2*d*f*tan(e + f*x)
+ 2*I*d*f) + 3*I*C*b**2*f*x*tan(e + f*x)/(2*d*f*tan(e + f*x) + 2*I*d*f) - 3
*C*b**2*f*x/(2*d*f*tan(e + f*x) + 2*I*d*f) - 2*C*b**2*log(tan(e + f*x)**2 +
1)*tan(e + f*x)/(2*d*f*tan(e + f*x) + 2*I*d*f) - 2*I*C*b**2*log(tan(e + f*
x)**2 + 1)/(2*d*f*tan(e + f*x) + 2*I*d*f) + C*b**2*tan(e + f*x)**3/(2*d*f*t
an(e + f*x) + 2*I*d*f) - I*C*b**2*tan(e + f*x)**2/(2*d*f*tan(e + f*x) + 2*I
*d*f) - 3*I*C*b**2/(2*d*f*tan(e + f*x) + 2*I*d*f), Eq(c, I*d)), (x*(a + b*t

```
an(e))**2*(A + B*tan(e) + C*tan(e)**2)/(c + d*tan(e)), Eq(f, 0)), (2*A*a**2
*c*d**3*f*x/(2*c**2*d**3*f + 2*d**5*f) + 2*A*a**2*d**4*log(c/d + tan(e + f*
x))/(2*c**2*d**3*f + 2*d**5*f) - A*a**2*d**4*log(tan(e + f*x)**2 + 1)/(2*c*
*2*d**3*f + 2*d**5*f) - 4*A*a*b*c*d**3*log(c/d + tan(e + f*x))/(2*c**2*d**3
*f + 2*d**5*f) + 2*A*a*b*c*d**3*log(tan(e + f*x)**2 + 1)/(2*c**2*d**3*f + 2
*d**5*f) + 4*A*a*b*d**4*f*x/(2*c**2*d**3*f + 2*d**5*f) + 2*A*b**2*c**2*d**2
*log(c/d + tan(e + f*x))/(2*c**2*d**3*f + 2*d**5*f) - 2*A*b**2*c*d**3*f*x/(
2*c**2*d**3*f + 2*d**5*f) + A*b**2*d**4*log(tan(e + f*x)**2 + 1)/(2*c**2*d*
*3*f + 2*d**5*f) - 2*B*a**2*c*d**3*log(c/d + tan(e + f*x))/(2*c**2*d**3*f +
 2*d**5*f) + B*a**2*c*d**3*log(tan(e + f*x)**2 + 1)/(2*c**2*d**3*f + 2*d**5
*f) + 2*B*a**2*d**4*f*x/(2*c**2*d**3*f + 2*d**5*f) + 4*B*a*b*c**2*d**2*log(
c/d + tan(e + f*x))/(2*c**2*d**3*f + 2*d**5*f) - 4*B*a*b*c*d**3*f*x/(2*c**2
*d**3*f + 2*d**5*f) + 2*B*a*b*d**4*log(tan(e + f*x)**2 + 1)/(2*c**2*d**3*f
+ 2*d**5*f) - 2*B*b**2*c**3*d*log(c/d + tan(e + f*x))/(2*c**2*d**3*f + 2*d*
*5*f) + 2*B*b**2*c**2*d**2*tan(e + f*x)/(2*c**2*d**3*f + 2*d**5*f) - B*b**2
*c*d**3*log(tan(e + f*x)**2 + 1)/(2*c**2*d**3*f + 2*d**5*f) - 2*B*b**2*d**4
*f*x/(2*c**2*d**3*f + 2*d**5*f) + 2*B*b**2*d**4*tan(e + f*x)/(2*c**2*d**3*f
 + 2*d**5*f) + 2*C*a**2*c**2*d**2*log(c/d + tan(e + f*x))/(2*c**2*d**3*f +
2*d**5*f) - 2*C*a**2*c*d**3*f*x/(2*c**2*d**3*f + 2*d**5*f) + C*a**2*d**4*lo
g(tan(e + f*x)**2 + 1)/(2*c**2*d**3*f + 2*d**5*f) - 4*C*a*b*c**3*d*log(c/d
+ tan(e + f*x))/(2*c**2*d**3*f + 2*d**5*f) + 4*C*a*b*c**2*d**2*tan(e + f*x)
/(2*c**2*d**3*f + 2*d**5*f) - 2*C*a*b*c*d**3*log(tan(e + f*x)**2 + 1)/(2*c*
*2*d**3*f + 2*d**5*f) - 4*C*a*b*d**4*f*x/(2*c**2*d**3*f + 2*d**5*f) + 4*C*a
*b*d**4*tan(e + f*x)/(2*c**2*d**3*f + 2*d**5*f) + 2*C*b**2*c**4*log(c/d + t
an(e + f*x))/(2*c**2*d**3*f + 2*d**5*f) - 2*C*b**2*c**3*d*tan(e + f*x)/(2*c
**2*d**3*f + 2*d**5*f) + C*b**2*c**2*d**2*tan(e + f*x)**2/(2*c**2*d**3*f +
2*d**5*f) + 2*C*b**2*c*d**3*f*x/(2*c**2*d**3*f + 2*d**5*f) - 2*C*b**2*c*d**
3*tan(e + f*x)/(2*c**2*d**3*f + 2*d**5*f) - C*b**2*d**4*log(tan(e + f*x)**2
 + 1)/(2*c**2*d**3*f + 2*d**5*f) + C*b**2*d**4*tan(e + f*x)**2/(2*c**2*d**3
*f + 2*d**5*f), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.44 (sec) , antiderivative size = 294, normalized size of antiderivative = 1.25

$$\int \frac{(a+b\tan(e+fx))^2 \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{c+d\tan(e+fx)}\,dx$$

$$= \frac{\frac{2\left(\left((A-C)a^2-2\,Bab-(A-C)b^2\right)c+\left(Ba^2+2\,(A-C)ab-Bb^2\right)d\right)(fx+e)}{c^2+d^2}+\frac{2\left(Cb^2c^4+Aa^2d^4-\left(2\,Cab+Bb^2\right)c^3d+\left(Ca^2+2\,Bab+Ab^2\right)c^2d^2-\left(Ba^2+\right.\right.}{c^2d^3+d^5}}{}$$

```
[In] integrate((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
)),x, algorithm="maxima")
```

```
[Out] 1/2*(2*(((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*c + (B*a^2 + 2*(A - C)*a*b -
B*b^2)*d)*(f*x + e)/(c^2 + d^2) + 2*(C*b^2*c^4 + A*a^2*d^4 - (2*C*a*b + B*b
^2)*c^3*d + (C*a^2 + 2*B*a*b + A*b^2)*c^2*d^2 - (B*a^2 + 2*A*a*b)*c*d^3)*lo
g(d*tan(f*x + e) + c)/(c^2*d^3 + d^5) + ((B*a^2 + 2*(A - C)*a*b - B*b^2)*c
- ((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*d)*log(tan(f*x + e)^2 + 1)/(c^2 + d
^2) + (C*b^2*d*tan(f*x + e)^2 - 2*(C*b^2*c - (2*C*a*b + B*b^2)*d)*tan(f*x +
 e))/d^2)/f
```

Giac [A] (verification not implemented)

none

Time = 0.67 (sec) , antiderivative size = 331, normalized size of antiderivative = 1.40

$$\int \frac{(a+b\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{c+d\tan(e+fx)}\,dx$$

$$=\frac{\frac{2\left(Aa^2c-Ca^2c-2\,Babc-Ab^2c+Cb^2c+Ba^2d+2\,Aabd-2\,Cabd-Bb^2d\right)(fx+e)}{c^2+d^2}+\frac{\left(Ba^2c+2\,Aabc-2\,Cabc-Bb^2c-Aa^2d+Ca^2d+2\,Babd+Ab^2d-\right.}{c^2+d^2}}{}$$

```
[In] integrate((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
)),x, algorithm="giac")
```

```
[Out] 1/2*(2*(A*a^2*c - C*a^2*c - 2*B*a*b*c - A*b^2*c + C*b^2*c + B*a^2*d + 2*A*a
*b*d - 2*C*a*b*d - B*b^2*d)*(f*x + e)/(c^2 + d^2) + (B*a^2*c + 2*A*a*b*c -
2*C*a*b*c - B*b^2*c - A*a^2*d + C*a^2*d + 2*B*a*b*d + A*b^2*d - C*b^2*d)*lo
g(tan(f*x + e)^2 + 1)/(c^2 + d^2) + 2*(C*b^2*c^4 - 2*C*a*b*c^3*d - B*b^2*c^
3*d + C*a^2*c^2*d^2 + 2*B*a*b*c^2*d^2 + A*b^2*c^2*d^2 - B*a^2*c*d^3 - 2*A*a
*b*c*d^3 + A*a^2*d^4)*log(abs(d*tan(f*x + e) + c))/(c^2*d^3 + d^5) + (C*b^2
*d*tan(f*x + e)^2 - 2*C*b^2*c*tan(f*x + e) + 4*C*a*b*d*tan(f*x + e) + 2*B*b
^2*d*tan(f*x + e))/d^2)/f
```

Mupad [B] (verification not implemented)

Time = 10.24 (sec) , antiderivative size = 325, normalized size of antiderivative = 1.38

$$\int \frac{(a+b\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{c+d\tan(e+fx)}\,dx$$
$$=\frac{\tan(e+f\,x)\,\left(\frac{B\,b^2+2\,C\,a\,b}{d}-\frac{C\,b^2\,c}{d^2}\right)}{f}$$
$$+\frac{\ln\left(c+d\tan(e+f\,x)\right)\,\left(d^2\,(C\,a^2\,c^2+2\,B\,a\,b\,c^2+A\,b^2\,c^2)-d\,(B\,b^2\,c^3+2\,C\,a\,b\,c^3)-d^3\,(B\,c\,a^2+2\,A\,b\,c\,a\right.}{f\,\left(c^2\,d^3+d^5\right)}$$
$$+\frac{\ln\left(\tan(e+f\,x)+1\mathrm{i}\right)\,\left(A\,b^2-A\,a^2+B\,a^2\,1\mathrm{i}-B\,b^2\,1\mathrm{i}+C\,a^2-C\,b^2+A\,a\,b\,2\mathrm{i}+2\,B\,a\,b-C\,a\,b\,2\mathrm{i}\right)}{2\,f\,\left(d+c\,1\mathrm{i}\right)}$$
$$+\frac{\ln\left(\tan(e+f\,x)-\mathrm{i}\right)\,\left(B\,a^2-B\,b^2+2\,A\,a\,b-2\,C\,a\,b-A\,a^2\,1\mathrm{i}+A\,b^2\,1\mathrm{i}+C\,a^2\,1\mathrm{i}-C\,b^2\,1\mathrm{i}+B\,a\,b\,2\mathrm{i}\right)}{2\,f\,\left(c+d\,1\mathrm{i}\right)}$$
$$+\frac{C\,b^2\,{\tan(e+f\,x)}^2}{2\,d\,f}$$

```
[In] int(((a + b*tan(e + f*x))^2*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d
*tan(e + f*x)),x)
```

```
[Out] (tan(e + f*x)*((B*b^2 + 2*C*a*b)/d - (C*b^2*c)/d^2))/f + (log(c + d*tan(e +
 f*x))*(d^2*(A*b^2*c^2 + C*a^2*c^2 + 2*B*a*b*c^2) - d*(B*b^2*c^3 + 2*C*a*b*
c^3) - d^3*(B*a^2*c + 2*A*a*b*c) + A*a^2*d^4 + C*b^2*c^4))/(f*(d^5 + c^2*d^
3)) + (log(tan(e + f*x) + 1i)*(A*b^2 - A*a^2 + B*a^2*1i - B*b^2*1i + C*a^2
- C*b^2 + A*a*b*2i + 2*B*a*b - C*a*b*2i))/(2*f*(c*1i + d)) + (log(tan(e + f
*x) - 1i)*(A*b^2*1i - A*a^2*1i + B*a^2 - B*b^2 + C*a^2*1i - C*b^2*1i + 2*A*
a*b + B*a*b*2i - 2*C*a*b))/(2*f*(c + d*1i)) + (C*b^2*tan(e + f*x)^2)/(2*d*f
)
```

3.72 $\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{c+d\tan(e+fx)}\,dx$

Optimal result 693
Rubi [A] (verified) 693
Mathematica [C] (verified) 695
Maple [A] (verified) 696
Fricas [A] (verification not implemented) 696
Sympy [C] (verification not implemented) 697
Maxima [A] (verification not implemented) 698
Giac [A] (verification not implemented) 699
Mupad [B] (verification not implemented) 699

Optimal result

Integrand size = 43, antiderivative size = 156

$$\begin{aligned}&\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{c+d\tan(e+fx)}\,dx\\&=\frac{(a(Ac-cC+Bd)-b(Bc-(A-C)d))x}{c^2+d^2}\\&\quad-\frac{(Abc+aBc-bcC-aAd+bBd+aCd)\log(\cos(e+fx))}{(c^2+d^2)\,f}\\&\quad-\frac{(bc-ad)\left(c^2C-Bcd+Ad^2\right)\log(c+d\tan(e+fx))}{d^2\left(c^2+d^2\right)f}+\frac{bC\tan(e+fx)}{df}\end{aligned}$$

```
[Out] (a*(A*c+B*d-C*c)-b*(B*c-(A-C)*d))*x/(c^2+d^2)-(-A*a*d+A*b*c+B*a*c+B*b*d+C*a
*d-C*b*c)*ln(cos(f*x+e))/(c^2+d^2)/f-(-a*d+b*c)*(A*d^2-B*c*d+C*c^2)*ln(c+d*
tan(f*x+e))/d^2/(c^2+d^2)/f+b*C*tan(f*x+e)/d/f
```

Rubi [A] (verified)

Time = 0.38 (sec) , antiderivative size = 156, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.116, Rules used = {3718, 3707, 3698, 31, 3556}

$$\begin{aligned}&\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{c+d\tan(e+fx)}\,dx\\&=-\frac{(bc-ad)\left(Ad^2-Bcd+c^2C\right)\log(c+d\tan(e+fx))}{d^2f\left(c^2+d^2\right)}\\&\quad-\frac{\log(\cos(e+fx))(-aAd+aBc+aCd+Abc+bBd-bcC)}{f\left(c^2+d^2\right)}\\&\quad+\frac{x(a(Ac+Bd-cC)-b(Bc-d(A-C)))}{c^2+d^2}+\frac{bC\tan(e+fx)}{df}\end{aligned}$$

```
[In] Int[((a + b*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c + d*T
an[e + f*x]),x]
```

```
[Out] ((a*(A*c - c*C + B*d) - b*(B*c - (A - C)*d))*x)/(c^2 + d^2) - ((A*b*c + a*B
*c - b*c*C - a*A*d + b*B*d + a*C*d)*Log[Cos[e + f*x]])/((c^2 + d^2)*f) - ((
b*c - a*d)*(c^2*C - B*c*d + A*d^2)*Log[c + d*Tan[e + f*x]])/(d^2*(c^2 + d^2
)*f) + (b*C*Tan[e + f*x])/(d*f)
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3698

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3707

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[(a*A + b*B - a
*C)*(x/(a^2 + b^2)), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rubi steps

$$\text{integral} = \frac{bC\tan(e+fx)}{df} - \frac{\int \frac{bcC-aAd-(Ab+aB-bC)d\tan(e+fx)+(bcC-bBd-aCd)\tan^2(e+fx)}{c+d\tan(e+fx)}\,dx}{d}$$
$$= \frac{(a(Ac-cC+Bd)-b(Bc-(A-C)d))x}{c^2+d^2} + \frac{bC\tan(e+fx)}{df} + \frac{(Abc+aBc-bcC-aAd+bBd+aCd)\int \tan(e+fx)\,dx}{c^2+d^2} - \frac{((bc-ad)(c^2C-Bcd+Ad^2))\int \frac{1+\tan^2(e+fx)}{c+d\tan(e+fx)}\,dx}{d(c^2+d^2)}$$
$$= \frac{(a(Ac-cC+Bd)-b(Bc-(A-C)d))x}{c^2+d^2} - \frac{(Abc+aBc-bcC-aAd+bBd+aCd)\log(\cos(e+fx))}{(c^2+d^2)f} + \frac{bC\tan(e+fx)}{df} - \frac{((bc-ad)(c^2C-Bcd+Ad^2))\,\text{Subst}\left(\int \frac{1}{c+x}\,dx, x, d\tan(e+fx)\right)}{d^2(c^2+d^2)f}$$
$$= \frac{(a(Ac-cC+Bd)-b(Bc-(A-C)d))x}{c^2+d^2} - \frac{(Abc+aBc-bcC-aAd+bBd+aCd)\log(\cos(e+fx))}{(c^2+d^2)f} - \frac{(bc-ad)(c^2C-Bcd+Ad^2)\log(c+d\tan(e+fx))}{d^2(c^2+d^2)f} + \frac{bC\tan(e+fx)}{df}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 1.18 (sec) , antiderivative size = 148, normalized size of antiderivative = 0.95

$$\int \frac{(a+b\tan(e+fx))(A+B\tan(e+fx)+C\tan^2(e+fx))}{c+d\tan(e+fx)}\,dx$$
$$= \frac{\frac{(-ia+b)(A+iB-C)\log(i-\tan(e+fx))}{c+id} + \frac{(ia+b)(A-iB-C)\log(i+\tan(e+fx))}{c-id} + \frac{2(-bc+ad)(c^2C-Bcd+Ad^2)\log(c+d\tan(e+fx))}{d^2(c^2+d^2)} + 2bC}{2f}$$

```
[In] Integrate[((a + b*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c
 + d*Tan[e + f*x]),x]
```

```
[Out] ((((-I)*a + b)*(A + I*B - C)*Log[I - Tan[e + f*x]])/(c + I*d) + ((I*a + b)*
(A - I*B - C)*Log[I + Tan[e + f*x]])/(c - I*d) + (2*(-(b*c) + a*d)*(c^2*C -
 B*c*d + A*d^2)*Log[c + d*Tan[e + f*x]])/(d^2*(c^2 + d^2)) + (2*b*C*Tan[e +
 f*x])/d)/(2*f)
```

Maple [A] (verified)

Time = 0.15 (sec) , antiderivative size = 173, normalized size of antiderivative = 1.11

method	result
derivativedivides	$\frac{\frac{\tan(fx+e)Cb}{d}+\frac{\frac{(-Aad+Abc+Bac+bdB+Cad-Cbc)\ln\left(1+\tan(fx+e)^2\right)}{2}+(Aac+Abd+Bad-Bbc-Cac-Cbd)\arctan(\tan(fx+e))}{c^2+d^2}+\frac{(Ac}{}}{f}$
default	$\frac{\frac{\tan(fx+e)Cb}{d}+\frac{\frac{(-Aad+Abc+Bac+bdB+Cad-Cbc)\ln\left(1+\tan(fx+e)^2\right)}{2}+(Aac+Abd+Bad-Bbc-Cac-Cbd)\arctan(\tan(fx+e))}{c^2+d^2}+\frac{(Ac}{}}{f}$
norman	$\frac{(Aac+Abd+Bad-Bbc-Cac-Cbd)x}{c^2+d^2}+\frac{bC\tan(fx+e)}{df}+\frac{(Aa\,d^3-Abc\,d^2-Bac\,d^2+Bb\,c^2d+Ca\,c^2d-Cb\,c^3)\ln(c+d\tan}{d^2f(c^2+d^2)}$
parallelrisch	$-\frac{-2Aac\,d^2fx-2Ab\,d^3fx-2Ba\,d^3fx+2Bbc\,d^2fx+2Cac\,d^2fx+2Cb\,d^3fx+A\ln\left(1+\tan(fx+e)^2\right)a\,d^3-A\ln\left(1+\tan(fx+\right.}{}$
risch	$\frac{2ibBe}{df}+\frac{2iCae}{df}-\frac{2iCbcx}{d^2}-\frac{2idAax}{c^2+d^2}+\frac{2iAbcx}{c^2+d^2}+\frac{2iBacx}{c^2+d^2}-\frac{\ln\left(e^{2i(fx+e)}-\frac{id+c}{id-c}\right)Cb\,c^3}{d^2f(c^2+d^2)}-\frac{2iCbce}{d^2f}-\frac{2idAae}{f(c^2+d^2}$

```
[In] int((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e)),x,met
hod=_RETURNVERBOSE)
```

```
[Out] 1/f*(tan(f*x+e)*C*b/d+1/(c^2+d^2)*(1/2*(-A*a*d+A*b*c+B*a*c+B*b*d+C*a*d-C*b*
c)*ln(1+tan(f*x+e)^2)+(A*a*c+A*b*d+B*a*d-B*b*c-C*a*c-C*b*d)*arctan(tan(f*x+
e)))+1/d^2*(A*a*d^3-A*b*c*d^2-B*a*c*d^2+B*b*c^2*d+C*a*c^2*d-C*b*c^3)/(c^2+d
^2)*ln(c+d*tan(f*x+e)))
```

Fricas [A] (verification not implemented)

none

Time = 0.32 (sec) , antiderivative size = 212, normalized size of antiderivative = 1.36

$$\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{c+d\tan(e+fx)}\,dx$$
$$=\frac{2\left(((A-C)a-Bb)cd^2+(Ba+(A-C)b)d^3\right)fx-\left(Cbc^3-Aad^3-(Ca+Bb)c^2d+(Ba+Ab)cd^2\right)\log\left(\right.}{}$$

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
,x, algorithm="fricas")
```

```
[Out] 1/2*(2*(((A - C)*a - B*b)*c*d^2 + (B*a + (A - C)*b)*d^3)*f*x - (C*b*c^3 - A
*a*d^3 - (C*a + B*b)*c^2*d + (B*a + A*b)*c*d^2)*log((d^2*tan(f*x + e)^2 + 2
*c*d*tan(f*x + e) + c^2)/(tan(f*x + e)^2 + 1)) + (C*b*c^3 + C*b*c*d^2 - (C*
a + B*b)*c^2*d - (C*a + B*b)*d^3)*log(1/(tan(f*x + e)^2 + 1)) + 2*(C*b*c^2*
d + C*b*d^3)*tan(f*x + e))/((c^2*d^2 + d^4)*f)
```

Sympy [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 0.92 (sec) , antiderivative size = 2387, normalized size of antiderivative = 15.30

$$\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{c+d\tan(e+fx)}\,dx = \text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x+e)
),x)
```

```
[Out] Piecewise((zoo*x*(a + b*tan(e))*(A + B*tan(e) + C*tan(e)**2)/tan(e), Eq(c,
0) & Eq(d, 0) & Eq(f, 0)), ((A*a*x + A*b*log(tan(e + f*x)**2 + 1)/(2*f) + B
*a*log(tan(e + f*x)**2 + 1)/(2*f) - B*b*x + B*b*tan(e + f*x)/f - C*a*x + C*
a*tan(e + f*x)/f - C*b*log(tan(e + f*x)**2 + 1)/(2*f) + C*b*tan(e + f*x)**2
/(2*f))/c, Eq(d, 0)), (I*A*a*f*x*tan(e + f*x)/(2*d*f*tan(e + f*x) - 2*I*d*f
) + A*a*f*x/(2*d*f*tan(e + f*x) - 2*I*d*f) + I*A*a/(2*d*f*tan(e + f*x) - 2*
I*d*f) + A*b*f*x*tan(e + f*x)/(2*d*f*tan(e + f*x) - 2*I*d*f) - I*A*b*f*x/(2
*d*f*tan(e + f*x) - 2*I*d*f) - A*b/(2*d*f*tan(e + f*x) - 2*I*d*f) + B*a*f*x
*tan(e + f*x)/(2*d*f*tan(e + f*x) - 2*I*d*f) - I*B*a*f*x/(2*d*f*tan(e + f*x
) - 2*I*d*f) - B*a/(2*d*f*tan(e + f*x) - 2*I*d*f) + I*B*b*f*x*tan(e + f*x)/
(2*d*f*tan(e + f*x) - 2*I*d*f) + B*b*f*x/(2*d*f*tan(e + f*x) - 2*I*d*f) + B
*b*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*d*f*tan(e + f*x) - 2*I*d*f) - I
*B*b*log(tan(e + f*x)**2 + 1)/(2*d*f*tan(e + f*x) - 2*I*d*f) - I*B*b/(2*d*f
*tan(e + f*x) - 2*I*d*f) + I*C*a*f*x*tan(e + f*x)/(2*d*f*tan(e + f*x) - 2*I
*d*f) + C*a*f*x/(2*d*f*tan(e + f*x) - 2*I*d*f) + C*a*log(tan(e + f*x)**2 +
1)*tan(e + f*x)/(2*d*f*tan(e + f*x) - 2*I*d*f) - I*C*a*log(tan(e + f*x)**2
+ 1)/(2*d*f*tan(e + f*x) - 2*I*d*f) - I*C*a/(2*d*f*tan(e + f*x) - 2*I*d*f)
- 3*C*b*f*x*tan(e + f*x)/(2*d*f*tan(e + f*x) - 2*I*d*f) + 3*I*C*b*f*x/(2*d*
f*tan(e + f*x) - 2*I*d*f) + I*C*b*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*
d*f*tan(e + f*x) - 2*I*d*f) + C*b*log(tan(e + f*x)**2 + 1)/(2*d*f*tan(e + f
*x) - 2*I*d*f) + 2*C*b*tan(e + f*x)**2/(2*d*f*tan(e + f*x) - 2*I*d*f) + 3*C
*b/(2*d*f*tan(e + f*x) - 2*I*d*f), Eq(c, -I*d)), (-I*A*a*f*x*tan(e + f*x)/(
2*d*f*tan(e + f*x) + 2*I*d*f) + A*a*f*x/(2*d*f*tan(e + f*x) + 2*I*d*f) - I*
A*a/(2*d*f*tan(e + f*x) + 2*I*d*f) + A*b*f*x*tan(e + f*x)/(2*d*f*tan(e + f*
x) + 2*I*d*f) + I*A*b*f*x/(2*d*f*tan(e + f*x) + 2*I*d*f) - A*b/(2*d*f*tan(e
 + f*x) + 2*I*d*f) + B*a*f*x*tan(e + f*x)/(2*d*f*tan(e + f*x) + 2*I*d*f) +
I*B*a*f*x/(2*d*f*tan(e + f*x) + 2*I*d*f) - B*a/(2*d*f*tan(e + f*x) + 2*I*d*
f) - I*B*b*f*x*tan(e + f*x)/(2*d*f*tan(e + f*x) + 2*I*d*f) + B*b*f*x/(2*d*f
*tan(e + f*x) + 2*I*d*f) + B*b*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*d*f
*tan(e + f*x) + 2*I*d*f) + I*B*b*log(tan(e + f*x)**2 + 1)/(2*d*f*tan(e + f*
x) + 2*I*d*f) + I*B*b/(2*d*f*tan(e + f*x) + 2*I*d*f) - I*C*a*f*x*tan(e + f*
x)/(2*d*f*tan(e + f*x) + 2*I*d*f) + C*a*f*x/(2*d*f*tan(e + f*x) + 2*I*d*f)
+ C*a*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*d*f*tan(e + f*x) + 2*I*d*f)
+ I*C*a*log(tan(e + f*x)**2 + 1)/(2*d*f*tan(e + f*x) + 2*I*d*f) + I*C*a/(2*
```

```
d*f*tan(e + f*x) + 2*I*d*f) - 3*C*b*f*x*tan(e + f*x)/(2*d*f*tan(e + f*x) +
2*I*d*f) - 3*I*C*b*f*x/(2*d*f*tan(e + f*x) + 2*I*d*f) - I*C*b*log(tan(e + f
*x)**2 + 1)*tan(e + f*x)/(2*d*f*tan(e + f*x) + 2*I*d*f) + C*b*log(tan(e + f
*x)**2 + 1)/(2*d*f*tan(e + f*x) + 2*I*d*f) + 2*C*b*tan(e + f*x)**2/(2*d*f*t
an(e + f*x) + 2*I*d*f) + 3*C*b/(2*d*f*tan(e + f*x) + 2*I*d*f), Eq(c, I*d)),
 (x*(a + b*tan(e))*(A + B*tan(e) + C*tan(e)**2)/(c + d*tan(e)), Eq(f, 0)),
(2*A*a*c*d**2*f*x/(2*c**2*d**2*f + 2*d**4*f) + 2*A*a*d**3*log(c/d + tan(e +
 f*x))/(2*c**2*d**2*f + 2*d**4*f) - A*a*d**3*log(tan(e + f*x)**2 + 1)/(2*c*
*2*d**2*f + 2*d**4*f) - 2*A*b*c*d**2*log(c/d + tan(e + f*x))/(2*c**2*d**2*f
 + 2*d**4*f) + A*b*c*d**2*log(tan(e + f*x)**2 + 1)/(2*c**2*d**2*f + 2*d**4*
f) + 2*A*b*d**3*f*x/(2*c**2*d**2*f + 2*d**4*f) - 2*B*a*c*d**2*log(c/d + tan
(e + f*x))/(2*c**2*d**2*f + 2*d**4*f) + B*a*c*d**2*log(tan(e + f*x)**2 + 1)
/(2*c**2*d**2*f + 2*d**4*f) + 2*B*a*d**3*f*x/(2*c**2*d**2*f + 2*d**4*f) + 2
*B*b*c**2*d*log(c/d + tan(e + f*x))/(2*c**2*d**2*f + 2*d**4*f) - 2*B*b*c*d*
*2*f*x/(2*c**2*d**2*f + 2*d**4*f) + B*b*d**3*log(tan(e + f*x)**2 + 1)/(2*c*
*2*d**2*f + 2*d**4*f) + 2*C*a*c**2*d*log(c/d + tan(e + f*x))/(2*c**2*d**2*f
 + 2*d**4*f) - 2*C*a*c*d**2*f*x/(2*c**2*d**2*f + 2*d**4*f) + C*a*d**3*log(t
an(e + f*x)**2 + 1)/(2*c**2*d**2*f + 2*d**4*f) - 2*C*b*c**3*log(c/d + tan(e
 + f*x))/(2*c**2*d**2*f + 2*d**4*f) + 2*C*b*c**2*d*tan(e + f*x)/(2*c**2*d**
2*f + 2*d**4*f) - C*b*c*d**2*log(tan(e + f*x)**2 + 1)/(2*c**2*d**2*f + 2*d*
*4*f) - 2*C*b*d**3*f*x/(2*c**2*d**2*f + 2*d**4*f) + 2*C*b*d**3*tan(e + f*x)
/(2*c**2*d**2*f + 2*d**4*f), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.39 (sec) , antiderivative size = 178, normalized size of antiderivative = 1.14

$$\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{c+d\tan(e+fx)}\,dx$$

$$= \frac{\frac{2\,Cb\tan(fx+e)}{d} + \frac{2\,(((A-C)a-Bb)c+(Ba+(A-C)b)d)(fx+e)}{c^2+d^2} - \frac{2\,(Cbc^3-Aad^3-(Ca+Bb)c^2d+(Ba+Ab)cd^2)\log(d\tan(fx+e)+c)}{c^2d^2+d^4} + \frac{((Ba\cdots}{\cdots}}{2\,f}$$

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
,x, algorithm="maxima")
```

```
[Out] 1/2*(2*C*b*tan(f*x + e)/d + 2*(((A - C)*a - B*b)*c + (B*a + (A - C)*b)*d)*(
f*x + e)/(c^2 + d^2) - 2*(C*b*c^3 - A*a*d^3 - (C*a + B*b)*c^2*d + (B*a + A*
b)*c*d^2)*log(d*tan(f*x + e) + c)/(c^2*d^2 + d^4) + ((B*a + (A - C)*b)*c -
((A - C)*a - B*b)*d)*log(tan(f*x + e)^2 + 1)/(c^2 + d^2))/f
```

Giac [A] (verification not implemented)

none

Time = 0.53 (sec) , antiderivative size = 182, normalized size of antiderivative = 1.17

$$\int \frac{(a+b\tan(e+fx))\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{c+d\tan(e+fx)}\,dx$$
$$=\frac{\frac{2\,Cb\tan(fx+e)}{d}+\frac{2\,(Aac-Cac-Bbc+Bad+Abd-Cbd)(fx+e)}{c^2+d^2}+\frac{(Bac+Abc-Cbc-Aad+Cad+Bbd)\log\left(\tan(fx+e)^2+1\right)}{c^2+d^2}-\frac{2\,(Cbc^3-Cac}{}}{2\,f}$$

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
,x, algorithm="giac")
```

```
[Out] 1/2*(2*C*b*tan(f*x + e)/d + 2*(A*a*c - C*a*c - B*b*c + B*a*d + A*b*d - C*b*
d)*(f*x + e)/(c^2 + d^2) + (B*a*c + A*b*c - C*b*c - A*a*d + C*a*d + B*b*d)*
log(tan(f*x + e)^2 + 1)/(c^2 + d^2) - 2*(C*b*c^3 - C*a*c^2*d - B*b*c^2*d +
B*a*c*d^2 + A*b*c*d^2 - A*a*d^3)*log(abs(d*tan(f*x + e) + c))/(c^2*d^2 + d^
4))/f
```

Mupad [B] (verification not implemented)

Time = 9.17 (sec) , antiderivative size = 186, normalized size of antiderivative = 1.19

$$\int \frac{(a+b\tan(e+fx))\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{c+d\tan(e+fx)}\,dx$$
$$=\frac{\ln(\tan(e+f\,x)-\mathrm{i})\,\left(A\,b+B\,a-C\,b-A\,a\,1\mathrm{i}+B\,b\,1\mathrm{i}+C\,a\,1\mathrm{i}\right)}{2\,f\,\left(c+d\,1\mathrm{i}\right)}$$
$$+\frac{\ln(\tan(e+f\,x)+1\mathrm{i})\,\left(B\,b+A\,b\,1\mathrm{i}+B\,a\,1\mathrm{i}-A\,a+C\,a-C\,b\,1\mathrm{i}\right)}{2\,f\,\left(d+c\,1\mathrm{i}\right)}$$
$$-\frac{\ln(c+d\tan(e+f\,x))\,\left(d^2\,(A\,b\,c+B\,a\,c)-d\,(B\,b\,c^2+C\,a\,c^2)-A\,a\,d^3+C\,b\,c^3\right)}{f\,\left(c^2\,d^2+d^4\right)}$$
$$+\frac{C\,b\tan(e+f\,x)}{d\,f}$$

```
[In] int(((a + b*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d*t
an(e + f*x)),x)
```

```
[Out] (log(tan(e + f*x) - 1i)*(A*b - A*a*1i + B*a + B*b*1i + C*a*1i - C*b))/(2*f*
(c + d*1i)) + (log(tan(e + f*x) + 1i)*(A*b*1i - A*a + B*a*1i + B*b + C*a -
C*b*1i))/(2*f*(c*1i + d)) - (log(c + d*tan(e + f*x))*(d^2*(A*b*c + B*a*c) -
 d*(B*b*c^2 + C*a*c^2) - A*a*d^3 + C*b*c^3))/(f*(d^4 + c^2*d^2)) + (C*b*tan
(e + f*x))/(d*f)
```

3.73 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{c+d\tan(e+fx)}\,dx$

Optimal result 700
Rubi [A] (verified) 700
Mathematica [C] (verified) 702
Maple [A] (verified) 702
Fricas [A] (verification not implemented) 703
Sympy [C] (verification not implemented) 703
Maxima [A] (verification not implemented) 704
Giac [A] (verification not implemented) 704
Mupad [B] (verification not implemented) 705

Optimal result

Integrand size = 33, antiderivative size = 99

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{c+d\tan(e+fx)}\,dx = \frac{(Ac-cC+Bd)x}{c^2+d^2} - \frac{(Bc-(A-C)d)\log(\cos(e+fx))}{(c^2+d^2)\,f} + \frac{(c^2C-Bcd+Ad^2)\log(c+d\tan(e+fx))}{d\,(c^2+d^2)\,f}$$

```
[Out] (A*c+B*d-C*c)*x/(c^2+d^2)-(B*c-(A-C)*d)*ln(cos(f*x+e))/(c^2+d^2)/f+(A*d^2-B
*c*d+C*c^2)*ln(c+d*tan(f*x+e))/d/(c^2+d^2)/f
```

Rubi [A] (verified)

Time = 0.11 (sec) , antiderivative size = 99, normalized size of antiderivative = 1.00, number of steps used = 4, number of rules used = 4, $\frac{\text{number of rules}}{\text{integrand size}} = 0.121$, Rules used = {3707, 3698, 31, 3556}

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{c+d\tan(e+fx)}\,dx = \frac{(Ad^2-Bcd+c^2C)\log(c+d\tan(e+fx))}{df\,(c^2+d^2)} - \frac{(Bc-d(A-C))\log(\cos(e+fx))}{f\,(c^2+d^2)} + \frac{x(Ac+Bd-cC)}{c^2+d^2}$$

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/(c + d*Tan[e + f*x]),x]
```

```
[Out] ((A*c - c*C + B*d)*x)/(c^2 + d^2) - ((B*c - (A - C)*d)*Log[Cos[e + f*x]])/(
(c^2 + d^2)*f) + ((c^2*C - B*c*d + A*d^2)*Log[c + d*Tan[e + f*x]])/(d*(c^2
+ d^2)*f)
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3698

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3707

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[(a*A + b*B - a
*C)*(x/(a^2 + b^2)), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rubi steps

$$\begin{aligned}\text{integral} &= \frac{(Ac-cC+Bd)x}{c^2+d^2}-\frac{(-Bc+Ad-Cd)\int\tan(e+fx)\,dx}{c^2+d^2}\\&\quad+\frac{(c^2C-Bcd+Ad^2)\int\frac{1+\tan^2(e+fx)}{c+d\tan(e+fx)}\,dx}{c^2+d^2}\\&=\frac{(Ac-cC+Bd)x}{c^2+d^2}-\frac{(Bc-(A-C)d)\log(\cos(e+fx))}{(c^2+d^2)\,f}\\&\quad+\frac{(c^2C-Bcd+Ad^2)\,\text{Subst}\left(\int\frac{1}{c+x}\,dx,x,d\tan(e+fx)\right)}{d\,(c^2+d^2)\,f}\\&=\frac{(Ac-cC+Bd)x}{c^2+d^2}-\frac{(Bc-(A-C)d)\log(\cos(e+fx))}{(c^2+d^2)\,f}\\&\quad+\frac{(c^2C-Bcd+Ad^2)\log(c+d\tan(e+fx))}{d\,(c^2+d^2)\,f}\end{aligned}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 0.24 (sec) , antiderivative size = 117, normalized size of antiderivative = 1.18

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{c+d\tan(e+fx)}\,dx = \frac{\frac{(-iA+B+iC)\log(i-\tan(e+fx))}{c+id}+\frac{(iA+B-iC)\log(i+\tan(e+fx))}{c-id}+\frac{2(c^2C-Bcd+Ad^2)\log(c+d\tan(e+fx))}{d(c^2+d^2)}}{2f}$$

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/(c + d*Tan[e + f*x]),x]
```

```
[Out] ((((-I)*A + B + I*C)*Log[I - Tan[e + f*x]])/(c + I*d) + ((I*A + B - I*C)*Lo
g[I + Tan[e + f*x]])/(c - I*d) + (2*(c^2*C - B*c*d + A*d^2)*Log[c + d*Tan[e
 + f*x]])/(d*(c^2 + d^2)))/(2*f)
```

Maple [A] (verified)

Time = 0.10 (sec) , antiderivative size = 100, normalized size of antiderivative = 1.01

method	result
derivativedivides	$\frac{\frac{\frac{(-Ad+Bc+Cd)\ln\left(1+\tan(fx+e)^2\right)}{2}+(Ac+Bd-Cc)\arctan(\tan(fx+e))}{c^2+d^2}+\frac{\left(A\,d^2-Bcd+c^2C\right)\ln(c+d\tan(fx+e))}{\left(c^2+d^2\right)d}}{f}$
default	$\frac{\frac{\frac{(-Ad+Bc+Cd)\ln\left(1+\tan(fx+e)^2\right)}{2}+(Ac+Bd-Cc)\arctan(\tan(fx+e))}{c^2+d^2}+\frac{\left(A\,d^2-Bcd+c^2C\right)\ln(c+d\tan(fx+e))}{\left(c^2+d^2\right)d}}{f}$
norman	$\frac{(Ac+Bd-Cc)x}{c^2+d^2}+\frac{(A\,d^2-Bcd+c^2C)\ln(c+d\tan(fx+e))}{d(c^2+d^2)f}-\frac{(Ad-Bc-Cd)\ln\left(1+\tan(fx+e)^2\right)}{2f(c^2+d^2)}$
parallelrisch	$-\frac{-2Axcdf-2Bx\,d^2f+2Cxcdf+A\ln\left(1+\tan(fx+e)^2\right)d^2-2A\ln(c+d\tan(fx+e))d^2-B\ln\left(1+\tan(fx+e)^2\right)cd+2B\ln(c+\cdots}{2(c^2+d^2)df}$
risch	$\frac{ixB}{id-c}-\frac{xA}{id-c}+\frac{xC}{id-c}-\frac{2idAx}{c^2+d^2}-\frac{2idAe}{(c^2+d^2)f}+\frac{2iBcx}{c^2+d^2}+\frac{2iBce}{(c^2+d^2)f}-\frac{2ic^2Cx}{(c^2+d^2)d}-\frac{2ic^2Ce}{(c^2+d^2)df}+\frac{2iCx}{d}+\cdots$

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e)),x,method=_RETURNVERBOS
E)
```

```
[Out] 1/f*(1/(c^2+d^2)*(1/2*(-A*d+B*c+C*d)*ln(1+tan(f*x+e)^2)+(A*c+B*d-C*c)*arcta
n(tan(f*x+e)))+(A*d^2-B*c*d+C*c^2)/(c^2+d^2)/d*ln(c+d*tan(f*x+e)))
```

Fricas [A] (verification not implemented)

none

Time = 0.27 (sec) , antiderivative size = 118, normalized size of antiderivative = 1.19

$$\int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{c + d\tan(e + fx)}\,dx$$
$$= \frac{2\left((A - C)cd + Bd^2\right)fx + \left(Cc^2 - Bcd + Ad^2\right)\log\left(\frac{d^2\tan(fx+e)^2 + 2\,cd\tan(fx+e) + c^2}{\tan(fx+e)^2 + 1}\right) - \left(Cc^2 + Cd^2\right)\log\left(\frac{}{\tan(f}\right.}{2\left(c^2d + d^3\right)f}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e)),x, algorithm="fr
icas")
```

```
[Out] 1/2*(2*((A - C)*c*d + B*d^2)*f*x + (C*c^2 - B*c*d + A*d^2)*log((d^2*tan(f*x
 + e)^2 + 2*c*d*tan(f*x + e) + c^2)/(tan(f*x + e)^2 + 1)) - (C*c^2 + C*d^2)
*log(1/(tan(f*x + e)^2 + 1)))/((c^2*d + d^3)*f)
```

Sympy [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 0.58 (sec) , antiderivative size = 966, normalized size of antiderivative = 9.76

$$\int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{c + d\tan(e + fx)}\,dx = \text{Too large to display}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x+e)),x)
```

```
[Out] Piecewise((zoo*x*(A + B*tan(e) + C*tan(e)**2)/tan(e), Eq(c, 0) & Eq(d, 0) &
 Eq(f, 0)), ((A*x + B*log(tan(e + f*x)**2 + 1)/(2*f) - C*x + C*tan(e + f*x)
/f)/c, Eq(d, 0)), (I*A*f*x*tan(e + f*x)/(2*d*f*tan(e + f*x) - 2*I*d*f) + A*
f*x/(2*d*f*tan(e + f*x) - 2*I*d*f) + I*A/(2*d*f*tan(e + f*x) - 2*I*d*f) + B
*f*x*tan(e + f*x)/(2*d*f*tan(e + f*x) - 2*I*d*f) - I*B*f*x/(2*d*f*tan(e + f
*x) - 2*I*d*f) - B/(2*d*f*tan(e + f*x) - 2*I*d*f) + I*C*f*x*tan(e + f*x)/(2
*d*f*tan(e + f*x) - 2*I*d*f) + C*f*x/(2*d*f*tan(e + f*x) - 2*I*d*f) + C*log
(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*d*f*tan(e + f*x) - 2*I*d*f) - I*C*log
(tan(e + f*x)**2 + 1)/(2*d*f*tan(e + f*x) - 2*I*d*f) - I*C/(2*d*f*tan(e + f
*x) - 2*I*d*f), Eq(c, -I*d)), (-I*A*f*x*tan(e + f*x)/(2*d*f*tan(e + f*x) +
2*I*d*f) + A*f*x/(2*d*f*tan(e + f*x) + 2*I*d*f) - I*A/(2*d*f*tan(e + f*x) +
 2*I*d*f) + B*f*x*tan(e + f*x)/(2*d*f*tan(e + f*x) + 2*I*d*f) + I*B*f*x/(2*
d*f*tan(e + f*x) + 2*I*d*f) - B/(2*d*f*tan(e + f*x) + 2*I*d*f) - I*C*f*x*ta
n(e + f*x)/(2*d*f*tan(e + f*x) + 2*I*d*f) + C*f*x/(2*d*f*tan(e + f*x) + 2*I
*d*f) + C*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*d*f*tan(e + f*x) + 2*I*d
*f) + I*C*log(tan(e + f*x)**2 + 1)/(2*d*f*tan(e + f*x) + 2*I*d*f) + I*C/(2*
d*f*tan(e + f*x) + 2*I*d*f), Eq(c, I*d)), (x*(A + B*tan(e) + C*tan(e)**2)/(
```

```
c + d*tan(e)), Eq(f, 0)), (2*A*c*d*f*x/(2*c**2*d*f + 2*d**3*f) + 2*A*d**2*l
og(c/d + tan(e + f*x))/(2*c**2*d*f + 2*d**3*f) - A*d**2*log(tan(e + f*x)**2
 + 1)/(2*c**2*d*f + 2*d**3*f) - 2*B*c*d*log(c/d + tan(e + f*x))/(2*c**2*d*f
 + 2*d**3*f) + B*c*d*log(tan(e + f*x)**2 + 1)/(2*c**2*d*f + 2*d**3*f) + 2*B
*d**2*f*x/(2*c**2*d*f + 2*d**3*f) + 2*C*c**2*log(c/d + tan(e + f*x))/(2*c**
2*d*f + 2*d**3*f) - 2*C*c*d*f*x/(2*c**2*d*f + 2*d**3*f) + C*d**2*log(tan(e
+ f*x)**2 + 1)/(2*c**2*d*f + 2*d**3*f), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.40 (sec) , antiderivative size = 106, normalized size of antiderivative = 1.07

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{c+d\tan(e+fx)}\,dx = \frac{\frac{2\,((A-C)c+Bd)(fx+e)}{c^2+d^2}+\frac{2\,(Cc^2-Bcd+Ad^2)\log(d\tan(fx+e)+c)}{c^2d+d^3}+\frac{(Bc-(A-C)d)\log\left(\tan(fx+e)^2+1\right)}{c^2+d^2}}{2\,f}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e)),x, algorithm="ma
xima")
```

```
[Out] 1/2*(2*((A - C)*c + B*d)*(f*x + e)/(c^2 + d^2) + 2*(C*c^2 - B*c*d + A*d^2)*
log(d*tan(f*x + e) + c)/(c^2*d + d^3) + (B*c - (A - C)*d)*log(tan(f*x + e)^
2 + 1)/(c^2 + d^2))/f
```

Giac [A] (verification not implemented)

none

Time = 0.47 (sec) , antiderivative size = 106, normalized size of antiderivative = 1.07

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{c+d\tan(e+fx)}\,dx = \frac{\frac{2\,(Ac-Cc+Bd)(fx+e)}{c^2+d^2}+\frac{(Bc-Ad+Cd)\log\left(\tan(fx+e)^2+1\right)}{c^2+d^2}+\frac{2\,(Cc^2-Bcd+Ad^2)\log(|d\tan(fx+e)+c|)}{c^2d+d^3}}{2\,f}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e)),x, algorithm="gi
ac")
```

```
[Out] 1/2*(2*(A*c - C*c + B*d)*(f*x + e)/(c^2 + d^2) + (B*c - A*d + C*d)*log(tan(
f*x + e)^2 + 1)/(c^2 + d^2) + 2*(C*c^2 - B*c*d + A*d^2)*log(abs(d*tan(f*x +
 e) + c))/(c^2*d + d^3))/f
```

Mupad [B] (verification not implemented)

Time = 8.81 (sec) , antiderivative size = 109, normalized size of antiderivative = 1.10

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{c+d\tan(e+fx)}\,dx = \frac{\ln\left(\tan(e+f\,x)+1\mathrm{i}\right)\,\left(C-A+B\,1\mathrm{i}\right)}{2\,f\,\left(d+c\,1\mathrm{i}\right)} + \frac{\ln\left(\tan(e+f\,x)-\mathrm{i}\right)\,\left(B-A\,1\mathrm{i}+C\,1\mathrm{i}\right)}{2\,f\,\left(c+d\,1\mathrm{i}\right)} + \frac{\ln\left(c+d\tan(e+f\,x)\right)\,\left(C\,c^2-B\,c\,d+A\,d^2\right)}{d\,f\,\left(c^2+d^2\right)}$$

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/(c + d*tan(e + f*x)),x)
```

```
[Out] (log(tan(e + f*x) + 1i)*(B*1i - A + C))/(2*f*(c*1i + d)) + (log(tan(e + f*x
) - 1i)*(B - A*1i + C*1i))/(2*f*(c + d*1i)) + (log(c + d*tan(e + f*x))*(A*d
^2 + C*c^2 - B*c*d))/(d*f*(c^2 + d^2))
```

3.74 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))}\,dx$

Optimal result 706
Rubi [A] (verified) 706
Mathematica [A] (verified) 708
Maple [A] (verified) 708
Fricas [A] (verification not implemented) 709
Sympy [C] (verification not implemented) 709
Maxima [A] (verification not implemented) 721
Giac [A] (verification not implemented) 721
Mupad [B] (verification not implemented) 722

Optimal result

Integrand size = 45, antiderivative size = 165

$$\begin{aligned}\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))}\,dx &= \frac{(a(Ac-cC+Bd)+b(Bc-(A-C)d))x}{(a^2+b^2)(c^2+d^2)} \\ &\quad + \frac{(Ab^2-a(bB-aC))\log(a\cos(e+fx)+b\sin(e+fx))}{(a^2+b^2)(bc-ad)f} \\ &\quad - \frac{(c^2C-Bcd+Ad^2)\log(c\cos(e+fx)+d\sin(e+fx))}{(bc-ad)(c^2+d^2)f}\end{aligned}$$

```
[Out] (a*(A*c+B*d-C*c)+b*(B*c-(A-C)*d))*x/(a^2+b^2)/(c^2+d^2)+(A*b^2-a*(B*b-C*a))
*ln(a*cos(f*x+e)+b*sin(f*x+e))/(a^2+b^2)/(-a*d+b*c)/f-(A*d^2-B*c*d+C*c^2)*l
n(c*cos(f*x+e)+d*sin(f*x+e))/(-a*d+b*c)/(c^2+d^2)/f
```

Rubi [A] (verified)

Time = 0.31 (sec) , antiderivative size = 164, normalized size of antiderivative = 0.99, number of steps used = 3, number of rules used = 2, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.044, Rules used = {3732, 3611}

$$\begin{aligned}\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))}\,dx &= \frac{x(a(Ac+Bd-cC)-bd(A-C)+bBc)}{(a^2+b^2)(c^2+d^2)} \\ &\quad + \frac{(Ab^2-a(bB-aC))\log(a\cos(e+fx)+b\sin(e+fx))}{f(a^2+b^2)(bc-ad)} \\ &\quad - \frac{(Ad^2-Bcd+c^2C)\log(c\cos(e+fx)+d\sin(e+fx))}{f(c^2+d^2)(bc-ad)}\end{aligned}$$

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])*(c + d*Ta
n[e + f*x])),x]
```

```
[Out] ((b*B*c - b*(A - C)*d + a*(A*c - c*C + B*d))*x)/((a^2 + b^2)*(c^2 + d^2)) +
 ((A*b^2 - a*(b*B - a*C))*Log[a*Cos[e + f*x] + b*Sin[e + f*x]])/((a^2 + b^2
)*(b*c - a*d)*f) - ((c^2*C - B*c*d + A*d^2)*Log[c*Cos[e + f*x] + d*Sin[e +
f*x]])/((b*c - a*d)*(c^2 + d^2)*f)
```

Rule 3611

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c/(b*f))*Log[RemoveContent[a*Cos[e + f*x] + b*Si
n[e + f*x], x]], x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3732

```
Int[((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^
2)/(((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])), x_Symbol] :> Simp[(a*(A*c - c*C + B*d) + b*(B*c - A*d + C*d))*(x/
((a^2 + b^2)*(c^2 + d^2))), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/((b*c - a*d)
*(a^2 + b^2)), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] - Dist
[(c^2*C - B*c*d + A*d^2)/((b*c - a*d)*(c^2 + d^2)), Int[(d - c*Tan[e + f*x]
)/(c + d*Tan[e + f*x]), x], x]) /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] &&
 NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0]
```

Rubi steps

$$
\begin{aligned}
\text{integral} &= \frac{(bBc - b(A-C)d + a(Ac - cC + Bd))x}{(a^2+b^2)(c^2+d^2)} \\
&\quad + \frac{(Ab^2 - a(bB - aC)) \int \frac{b - a\tan(e+fx)}{a+b\tan(e+fx)}\,dx}{(a^2+b^2)(bc-ad)} - \frac{(c^2C - Bcd + Ad^2) \int \frac{d - c\tan(e+fx)}{c+d\tan(e+fx)}\,dx}{(bc-ad)(c^2+d^2)} \\
&= \frac{(bBc - b(A-C)d + a(Ac - cC + Bd))x}{(a^2+b^2)(c^2+d^2)} \\
&\quad + \frac{(Ab^2 - a(bB - aC))\log(a\cos(e+fx) + b\sin(e+fx))}{(a^2+b^2)(bc-ad)f} \\
&\quad - \frac{(c^2C - Bcd + Ad^2)\log(c\cos(e+fx) + d\sin(e+fx))}{(bc-ad)(c^2+d^2)f}
\end{aligned}
$$

Mathematica [A] (verified)

Time = 1.62 (sec) , antiderivative size = 313, normalized size of antiderivative = 1.90

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))}\,dx = -\frac{\frac{\left(Abc-aBc-bcC+aAd+bBd-aCd+\frac{\sqrt{-b^2}(bBc+b(-A+C)d+a(Ac-cC+Bd))}{b}\right)\log\left(\sqrt{-b^2}-b\tan(e+fx)\right)}{(a^2+b^2)(c^2+d^2)}+\frac{2(Ab^2+a(-bB+aC))\log(a+b\tan(e}{(a^2+b^2)(-bc+ad)}}{}$$

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])*(c
+ d*Tan[e + f*x])),x]
```

```
[Out] -1/2*(((A*b*c - a*B*c - b*c*C + a*A*d + b*B*d - a*C*d + (Sqrt[-b^2]*(b*B*c
+ b*(-A + C)*d + a*(A*c - c*C + B*d)))/b)*Log[Sqrt[-b^2] - b*Tan[e + f*x]])
/((a^2 + b^2)*(c^2 + d^2)) + (2*(A*b^2 + a*(-(b*B) + a*C))*Log[a + b*Tan[e
+ f*x]])/((a^2 + b^2)*(-(b*c) + a*d)) + ((A*b*c - a*B*c - b*c*C + a*A*d + b
*B*d - a*C*d + (b*(b*B*c + b*(-A + C)*d + a*(A*c - c*C + B*d)))/Sqrt[-b^2])
*Log[Sqrt[-b^2] + b*Tan[e + f*x]])/((a^2 + b^2)*(c^2 + d^2)) + (2*(c^2*C -
B*c*d + A*d^2)*Log[c + d*Tan[e + f*x]])/((b*c - a*d)*(c^2 + d^2)))/f
```

Maple [A] (verified)

Time = 0.28 (sec) , antiderivative size = 197, normalized size of antiderivative = 1.19

method	result
derivativedivides	$\frac{-\frac{\left(A b^2-Bab+C a^2\right)\ln(a+b\tan(fx+e))}{(ad-bc)\left(a^2+b^2\right)}+\frac{\frac{(-Aad-Abc+Bac-bdB+Cad+Cbc)\ln\left(1+\tan(fx+e)^2\right)}{2}+(Aac-Abd+Bad+Bbc-Cac+Cbd}{\left(a^2+b^2\right)\left(c^2+d^2\right)}}{f}$
default	$\frac{-\frac{\left(A b^2-Bab+C a^2\right)\ln(a+b\tan(fx+e))}{(ad-bc)\left(a^2+b^2\right)}+\frac{\frac{(-Aad-Abc+Bac-bdB+Cad+Cbc)\ln\left(1+\tan(fx+e)^2\right)}{2}+(Aac-Abd+Bad+Bbc-Cac+Cbd}{\left(a^2+b^2\right)\left(c^2+d^2\right)}}{f}$
norman	$\frac{(Aac-Abd+Bad+Bbc-Cac+Cbd)x}{(a^2+b^2)(c^2+d^2)}+\frac{(A d^2-Bcd+c^2C)\ln(c+d\tan(fx+e))}{f(a c^2d+a d^3-b c^3-bc d^2)}-\frac{(A b^2-Bab+C a^2)\ln(a+b\tan(fx+e}{(ad-bc)(a^2+b^2)f}$
parallelrisch	$-\frac{-2C\ln(c+d\tan(fx+e))a^2c^2-2C\ln(c+d\tan(fx+e))b^2c^2+A\ln\left(1+\tan(fx+e)^2\right)a^2d^2-A\ln\left(1+\tan(fx+e)^2\right)b^2c^2+2A}{}$
risch	$-\frac{2iBabx}{a^3d-a^2bc+a b^2d-b^3c}-\frac{xA}{iad+ibc-ac+bd}+\frac{xC}{iad+ibc-ac+bd}-\frac{2ic^2Ce}{f(a c^2d+a d^3-b c^3-bc d^2)}+\frac{2iA b^2x}{a^3d-a^2bc+a b^2d-b}$

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e)),x,met
hod=_RETURNVERBOSE)
```

```
[Out] 1/f*(-(A*b^2-B*a*b+C*a^2)/(a*d-b*c)/(a^2+b^2)*ln(a+b*tan(f*x+e))+1/(a^2+b^2
)/(c^2+d^2)*(1/2*(-A*a*d-A*b*c+B*a*c-B*b*d+C*a*d+C*b*c)*ln(1+tan(f*x+e)^2)+
```

```
(A*a*c-A*b*d+B*a*d+B*b*c-C*a*c+C*b*d)*arctan(tan(f*x+e)))+(A*d^2-B*c*d+C*c^
2)/(a*d-b*c)/(c^2+d^2)*ln(c+d*tan(f*x+e)))
```

Fricas [A] (verification not implemented)

none

Time = 0.46 (sec) , antiderivative size = 301, normalized size of antiderivative = 1.82

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))}\,dx$$
$$= \frac{2\left(\left(\left((A-C)ab+Bb^2\right)c^2-\left((A-C)a^2+(A-C)b^2\right)cd-\left(Ba^2-(A-C)ab\right)d^2\right)fx+\left(\left(Ca^2-Bab+Ab^2\right.\right.\right.}{2\left(\left(a^2b+\right.\right.}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))
,x, algorithm="fricas")
```

```
[Out] 1/2*(2*(((A - C)*a*b + B*b^2)*c^2 - ((A - C)*a^2 + (A - C)*b^2)*c*d - (B*a^
2 - (A - C)*a*b)*d^2)*f*x + ((C*a^2 - B*a*b + A*b^2)*c^2 + (C*a^2 - B*a*b +
 A*b^2)*d^2)*log((b^2*tan(f*x + e)^2 + 2*a*b*tan(f*x + e) + a^2)/(tan(f*x +
 e)^2 + 1)) - ((C*a^2 + C*b^2)*c^2 - (B*a^2 + B*b^2)*c*d + (A*a^2 + A*b^2)*
d^2)*log((d^2*tan(f*x + e)^2 + 2*c*d*tan(f*x + e) + c^2)/(tan(f*x + e)^2 +
1)))/(((a^2*b + b^3)*c^3 - (a^3 + a*b^2)*c^2*d + (a^2*b + b^3)*c*d^2 - (a^3
 + a*b^2)*d^3)*f)
```

Sympy [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 34.90 (sec) , antiderivative size = 24052, normalized size of antiderivative = 145.77

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))}\,dx = \text{Too large to display}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e)
),x)
```

```
[Out] Piecewise(((2*A*c*d*f*x/(2*c**2*d*f + 2*d**3*f) + 2*A*d**2*log(c/d + tan(e
+ f*x))/(2*c**2*d*f + 2*d**3*f) - A*d**2*log(tan(e + f*x)**2 + 1)/(2*c**2*d
*f + 2*d**3*f) - 2*B*c*d*log(c/d + tan(e + f*x))/(2*c**2*d*f + 2*d**3*f) +
B*c*d*log(tan(e + f*x)**2 + 1)/(2*c**2*d*f + 2*d**3*f) + 2*B*d**2*f*x/(2*c*
*2*d*f + 2*d**3*f) + 2*C*c**2*log(c/d + tan(e + f*x))/(2*c**2*d*f + 2*d**3*
f) - 2*C*c*d*f*x/(2*c**2*d*f + 2*d**3*f) + C*d**2*log(tan(e + f*x)**2 + 1)/
(2*c**2*d*f + 2*d**3*f))/a, Eq(b, 0)), ((2*A*a*b*f*x/(2*a**2*b*f + 2*b**3*f
) + 2*A*b**2*log(a/b + tan(e + f*x))/(2*a**2*b*f + 2*b**3*f) - A*b**2*log(t
```

an(e + f*x)**2 + 1)/(2*a**2*b*f + 2*b**3*f) - 2*B*a*b*log(a/b + tan(e + f*x
))/(2*a**2*b*f + 2*b**3*f) + B*a*b*log(tan(e + f*x)**2 + 1)/(2*a**2*b*f + 2
*b**3*f) + 2*B*b**2*f*x/(2*a**2*b*f + 2*b**3*f) + 2*C*a**2*log(a/b + tan(e
+ f*x))/(2*a**2*b*f + 2*b**3*f) - 2*C*a*b*f*x/(2*a**2*b*f + 2*b**3*f) + C*b
2*log(tan(e + f*x)2 + 1)/(2*a**2*b*f + 2*b**3*f))/c, Eq(d, 0)), (I*A*c*
*2*f*x*tan(e + f*x)/(2*b*c**3*f*tan(e + f*x) - 2*I*b*c**3*f + 2*I*b*c**2*d*
f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e + f*x) - 2*I*b*c*d**2*f
+ 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**3*f) + A*c**2*f*x/(2*b*c**3*f*tan(e +
f*x) - 2*I*b*c**3*f + 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c*d*
*2*f*tan(e + f*x) - 2*I*b*c*d**2*f + 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**3*f
) + I*A*c**2/(2*b*c**3*f*tan(e + f*x) - 2*I*b*c**3*f + 2*I*b*c**2*d*f*tan(e
 + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e + f*x) - 2*I*b*c*d**2*f + 2*I*b
*d**3*f*tan(e + f*x) + 2*b*d**3*f) - 2*A*c*d*f*x*tan(e + f*x)/(2*b*c**3*f*t
an(e + f*x) - 2*I*b*c**3*f + 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2
*b*c*d**2*f*tan(e + f*x) - 2*I*b*c*d**2*f + 2*I*b*d**3*f*tan(e + f*x) + 2*b
*d**3*f) + 2*I*A*c*d*f*x/(2*b*c**3*f*tan(e + f*x) - 2*I*b*c**3*f + 2*I*b*c*
*2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e + f*x) - 2*I*b*c*d*
*2*f + 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**3*f) + I*A*d**2*f*x*tan(e + f*x)/
(2*b*c**3*f*tan(e + f*x) - 2*I*b*c**3*f + 2*I*b*c**2*d*f*tan(e + f*x) + 2*b
*c**2*d*f + 2*b*c*d**2*f*tan(e + f*x) - 2*I*b*c*d**2*f + 2*I*b*d**3*f*tan(e
 + f*x) + 2*b*d**3*f) + A*d**2*f*x/(2*b*c**3*f*tan(e + f*x) - 2*I*b*c**3*f
+ 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e + f*x) -
2*I*b*c*d**2*f + 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**3*f) - 2*A*d**2*log(c/d
 + tan(e + f*x))*tan(e + f*x)/(2*b*c**3*f*tan(e + f*x) - 2*I*b*c**3*f + 2*I
*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e + f*x) - 2*I*b
*c*d**2*f + 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**3*f) + 2*I*A*d**2*log(c/d +
tan(e + f*x))/(2*b*c**3*f*tan(e + f*x) - 2*I*b*c**3*f + 2*I*b*c**2*d*f*tan(
e + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e + f*x) - 2*I*b*c*d**2*f + 2*I*
b*d**3*f*tan(e + f*x) + 2*b*d**3*f) + A*d**2*log(tan(e + f*x)**2 + 1)*tan(e
 + f*x)/(2*b*c**3*f*tan(e + f*x) - 2*I*b*c**3*f + 2*I*b*c**2*d*f*tan(e + f*
x) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e + f*x) - 2*I*b*c*d**2*f + 2*I*b*d**3
*f*tan(e + f*x) + 2*b*d**3*f) - I*A*d**2*log(tan(e + f*x)**2 + 1)/(2*b*c**3
*f*tan(e + f*x) - 2*I*b*c**3*f + 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f
 + 2*b*c*d**2*f*tan(e + f*x) - 2*I*b*c*d**2*f + 2*I*b*d**3*f*tan(e + f*x) +
 2*b*d**3*f) + I*A*d**2/(2*b*c**3*f*tan(e + f*x) - 2*I*b*c**3*f + 2*I*b*c**
2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e + f*x) - 2*I*b*c*d**
2*f + 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**3*f) + B*c**2*f*x*tan(e + f*x)/(2*
b*c**3*f*tan(e + f*x) - 2*I*b*c**3*f + 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c*
*2*d*f + 2*b*c*d**2*f*tan(e + f*x) - 2*I*b*c*d**2*f + 2*I*b*d**3*f*tan(e +
f*x) + 2*b*d**3*f) - I*B*c**2*f*x/(2*b*c**3*f*tan(e + f*x) - 2*I*b*c**3*f +
 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e + f*x) - 2
*I*b*c*d**2*f + 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**3*f) - B*c**2/(2*b*c**3*
f*tan(e + f*x) - 2*I*b*c**3*f + 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f
+ 2*b*c*d**2*f*tan(e + f*x) - 2*I*b*c*d**2*f + 2*I*b*d**3*f*tan(e + f*x) +
2*b*d**3*f) + 2*B*c*d*log(c/d + tan(e + f*x))*tan(e + f*x)/(2*b*c**3*f*tan(

e + f*x) - 2*I*b*c**3*f + 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*
c*d**2*f*tan(e + f*x) - 2*I*b*c*d**2*f + 2*I*b*d**3*f*tan(e + f*x) + 2*b*d*
*3*f) - 2*I*B*c*d*log(c/d + tan(e + f*x))/(2*b*c**3*f*tan(e + f*x) - 2*I*b*
c**3*f + 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e +
f*x) - 2*I*b*c*d**2*f + 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**3*f) - B*c*d*log
(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*b*c**3*f*tan(e + f*x) - 2*I*b*c**3*f
+ 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e + f*x) -
2*I*b*c*d**2*f + 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**3*f) + I*B*c*d*log(tan(
e + f*x)**2 + 1)/(2*b*c**3*f*tan(e + f*x) - 2*I*b*c**3*f + 2*I*b*c**2*d*f*t
an(e + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e + f*x) - 2*I*b*c*d**2*f + 2
*I*b*d**3*f*tan(e + f*x) + 2*b*d**3*f) - B*d**2*f*x*tan(e + f*x)/(2*b*c**3*
f*tan(e + f*x) - 2*I*b*c**3*f + 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f
+ 2*b*c*d**2*f*tan(e + f*x) - 2*I*b*c*d**2*f + 2*I*b*d**3*f*tan(e + f*x) +
2*b*d**3*f) + I*B*d**2*f*x/(2*b*c**3*f*tan(e + f*x) - 2*I*b*c**3*f + 2*I*b*
c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e + f*x) - 2*I*b*c*
d**2*f + 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**3*f) - B*d**2/(2*b*c**3*f*tan(e
 + f*x) - 2*I*b*c**3*f + 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c
*d**2*f*tan(e + f*x) - 2*I*b*c*d**2*f + 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**
3*f) + I*C*c**2*f*x*tan(e + f*x)/(2*b*c**3*f*tan(e + f*x) - 2*I*b*c**3*f +
2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e + f*x) - 2*
I*b*c*d**2*f + 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**3*f) + C*c**2*f*x/(2*b*c*
*3*f*tan(e + f*x) - 2*I*b*c**3*f + 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d
*f + 2*b*c*d**2*f*tan(e + f*x) - 2*I*b*c*d**2*f + 2*I*b*d**3*f*tan(e + f*x)
 + 2*b*d**3*f) - 2*C*c**2*log(c/d + tan(e + f*x))*tan(e + f*x)/(2*b*c**3*f*
tan(e + f*x) - 2*I*b*c**3*f + 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f +
2*b*c*d**2*f*tan(e + f*x) - 2*I*b*c*d**2*f + 2*I*b*d**3*f*tan(e + f*x) + 2*
b*d**3*f) + 2*I*C*c**2*log(c/d + tan(e + f*x))/(2*b*c**3*f*tan(e + f*x) - 2
*I*b*c**3*f + 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan
(e + f*x) - 2*I*b*c*d**2*f + 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**3*f) + C*c*
*2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*b*c**3*f*tan(e + f*x) - 2*I*b*c
3*f + 2*I*b*c2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e + f
*x) - 2*I*b*c*d**2*f + 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**3*f) - I*C*c**2*l
og(tan(e + f*x)**2 + 1)/(2*b*c**3*f*tan(e + f*x) - 2*I*b*c**3*f + 2*I*b*c**
2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e + f*x) - 2*I*b*c*d**
2*f + 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**3*f) - I*C*c**2/(2*b*c**3*f*tan(e
+ f*x) - 2*I*b*c**3*f + 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c*
d**2*f*tan(e + f*x) - 2*I*b*c*d**2*f + 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**3
*f) + 2*C*c*d*f*x*tan(e + f*x)/(2*b*c**3*f*tan(e + f*x) - 2*I*b*c**3*f + 2*
I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e + f*x) - 2*I*
b*c*d**2*f + 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**3*f) - 2*I*C*c*d*f*x/(2*b*c
3*f*tan(e + f*x) - 2*I*b*c3*f + 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*
d*f + 2*b*c*d**2*f*tan(e + f*x) - 2*I*b*c*d**2*f + 2*I*b*d**3*f*tan(e + f*x
) + 2*b*d**3*f) + I*C*d**2*f*x*tan(e + f*x)/(2*b*c**3*f*tan(e + f*x) - 2*I*
b*c**3*f + 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e
+ f*x) - 2*I*b*c*d**2*f + 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**3*f) + C*d**2*

```
f*x/(2*b*c**3*f*tan(e + f*x) - 2*I*b*c**3*f + 2*I*b*c**2*d*f*tan(e + f*x) +
 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e + f*x) - 2*I*b*c*d**2*f + 2*I*b*d**3*f*t
an(e + f*x) + 2*b*d**3*f) - I*C*d**2/(2*b*c**3*f*tan(e + f*x) - 2*I*b*c**3*
f + 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e + f*x)
- 2*I*b*c*d**2*f + 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**3*f), Eq(a, -I*b)), (
-I*A*c**2*f*x*tan(e + f*x)/(2*b*c**3*f*tan(e + f*x) + 2*I*b*c**3*f - 2*I*b*
c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e + f*x) + 2*I*b*c*
d**2*f - 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**3*f) + A*c**2*f*x/(2*b*c**3*f*t
an(e + f*x) + 2*I*b*c**3*f - 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2
*b*c*d**2*f*tan(e + f*x) + 2*I*b*c*d**2*f - 2*I*b*d**3*f*tan(e + f*x) + 2*b
*d**3*f) - I*A*c**2/(2*b*c**3*f*tan(e + f*x) + 2*I*b*c**3*f - 2*I*b*c**2*d*
f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e + f*x) + 2*I*b*c*d**2*f
- 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**3*f) - 2*A*c*d*f*x*tan(e + f*x)/(2*b*c
**3*f*tan(e + f*x) + 2*I*b*c**3*f - 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*
d*f + 2*b*c*d**2*f*tan(e + f*x) + 2*I*b*c*d**2*f - 2*I*b*d**3*f*tan(e + f*x
) + 2*b*d**3*f) - 2*I*A*c*d*f*x/(2*b*c**3*f*tan(e + f*x) + 2*I*b*c**3*f - 2
*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e + f*x) + 2*I
*b*c*d**2*f - 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**3*f) - I*A*d**2*f*x*tan(e
+ f*x)/(2*b*c**3*f*tan(e + f*x) + 2*I*b*c**3*f - 2*I*b*c**2*d*f*tan(e + f*x
) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e + f*x) + 2*I*b*c*d**2*f - 2*I*b*d**3*
f*tan(e + f*x) + 2*b*d**3*f) + A*d**2*f*x/(2*b*c**3*f*tan(e + f*x) + 2*I*b*
c**3*f - 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e +
f*x) + 2*I*b*c*d**2*f - 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**3*f) - 2*A*d**2*
log(c/d + tan(e + f*x))*tan(e + f*x)/(2*b*c**3*f*tan(e + f*x) + 2*I*b*c**3*
f - 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e + f*x)
+ 2*I*b*c*d**2*f - 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**3*f) - 2*I*A*d**2*log
(c/d + tan(e + f*x))/(2*b*c**3*f*tan(e + f*x) + 2*I*b*c**3*f - 2*I*b*c**2*d
*f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e + f*x) + 2*I*b*c*d**2*f
 - 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**3*f) + A*d**2*log(tan(e + f*x)**2 + 1
)*tan(e + f*x)/(2*b*c**3*f*tan(e + f*x) + 2*I*b*c**3*f - 2*I*b*c**2*d*f*tan
(e + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e + f*x) + 2*I*b*c*d**2*f - 2*I
*b*d**3*f*tan(e + f*x) + 2*b*d**3*f) + I*A*d**2*log(tan(e + f*x)**2 + 1)/(2
*b*c**3*f*tan(e + f*x) + 2*I*b*c**3*f - 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c
**2*d*f + 2*b*c*d**2*f*tan(e + f*x) + 2*I*b*c*d**2*f - 2*I*b*d**3*f*tan(e +
 f*x) + 2*b*d**3*f) - I*A*d**2/(2*b*c**3*f*tan(e + f*x) + 2*I*b*c**3*f - 2*
I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e + f*x) + 2*I*
b*c*d**2*f - 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**3*f) + B*c**2*f*x*tan(e + f
*x)/(2*b*c**3*f*tan(e + f*x) + 2*I*b*c**3*f - 2*I*b*c**2*d*f*tan(e + f*x) +
 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e + f*x) + 2*I*b*c*d**2*f - 2*I*b*d**3*f*t
an(e + f*x) + 2*b*d**3*f) + I*B*c**2*f*x/(2*b*c**3*f*tan(e + f*x) + 2*I*b*c
**3*f - 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e + f
*x) + 2*I*b*c*d**2*f - 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**3*f) - B*c**2/(2*
b*c**3*f*tan(e + f*x) + 2*I*b*c**3*f - 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c*
*2*d*f + 2*b*c*d**2*f*tan(e + f*x) + 2*I*b*c*d**2*f - 2*I*b*d**3*f*tan(e +
f*x) + 2*b*d**3*f) + 2*B*c*d*log(c/d + tan(e + f*x))*tan(e + f*x)/(2*b*c**3
```

```
*f*tan(e + f*x) + 2*I*b*c**3*f - 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f
 + 2*b*c*d**2*f*tan(e + f*x) + 2*I*b*c*d**2*f - 2*I*b*d**3*f*tan(e + f*x) +
 2*b*d**3*f) + 2*I*B*c*d*log(c/d + tan(e + f*x))/(2*b*c**3*f*tan(e + f*x) +
 2*I*b*c**3*f - 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*t
an(e + f*x) + 2*I*b*c*d**2*f - 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**3*f) - B*
c*d*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*b*c**3*f*tan(e + f*x) + 2*I*b*
c**3*f - 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e +
f*x) + 2*I*b*c*d**2*f - 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**3*f) - I*B*c*d*l
og(tan(e + f*x)**2 + 1)/(2*b*c**3*f*tan(e + f*x) + 2*I*b*c**3*f - 2*I*b*c**
2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e + f*x) + 2*I*b*c*d**
2*f - 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**3*f) - B*d**2*f*x*tan(e + f*x)/(2*
b*c**3*f*tan(e + f*x) + 2*I*b*c**3*f - 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c*
*2*d*f + 2*b*c*d**2*f*tan(e + f*x) + 2*I*b*c*d**2*f - 2*I*b*d**3*f*tan(e +
f*x) + 2*b*d**3*f) - I*B*d**2*f*x/(2*b*c**3*f*tan(e + f*x) + 2*I*b*c**3*f -
 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e + f*x) + 2
*I*b*c*d**2*f - 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**3*f) - B*d**2/(2*b*c**3*
f*tan(e + f*x) + 2*I*b*c**3*f - 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f
 + 2*b*c*d**2*f*tan(e + f*x) + 2*I*b*c*d**2*f - 2*I*b*d**3*f*tan(e + f*x) +
2*b*d**3*f) - I*C*c**2*f*x*tan(e + f*x)/(2*b*c**3*f*tan(e + f*x) + 2*I*b*c*
*3*f - 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e + f*
x) + 2*I*b*c*d**2*f - 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**3*f) + C*c**2*f*x/
(2*b*c**3*f*tan(e + f*x) + 2*I*b*c**3*f - 2*I*b*c**2*d*f*tan(e + f*x) + 2*b
*c**2*d*f + 2*b*c*d**2*f*tan(e + f*x) + 2*I*b*c*d**2*f - 2*I*b*d**3*f*tan(e
 + f*x) + 2*b*d**3*f) - 2*C*c**2*log(c/d + tan(e + f*x))*tan(e + f*x)/(2*b*
c**3*f*tan(e + f*x) + 2*I*b*c**3*f - 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2
*d*f + 2*b*c*d**2*f*tan(e + f*x) + 2*I*b*c*d**2*f - 2*I*b*d**3*f*tan(e + f*
x) + 2*b*d**3*f) - 2*I*C*c**2*log(c/d + tan(e + f*x))/(2*b*c**3*f*tan(e + f
*x) + 2*I*b*c**3*f - 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c*d**
2*f*tan(e + f*x) + 2*I*b*c*d**2*f - 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**3*f)
 + C*c**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*b*c**3*f*tan(e + f*x) +
2*I*b*c**3*f - 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*ta
n(e + f*x) + 2*I*b*c*d**2*f - 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**3*f) + I*C
*c**2*log(tan(e + f*x)**2 + 1)/(2*b*c**3*f*tan(e + f*x) + 2*I*b*c**3*f - 2*
I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e + f*x) + 2*I*
b*c*d**2*f - 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**3*f) + I*C*c**2/(2*b*c**3*f
*tan(e + f*x) + 2*I*b*c**3*f - 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f +
 2*b*c*d**2*f*tan(e + f*x) + 2*I*b*c*d**2*f - 2*I*b*d**3*f*tan(e + f*x) + 2
*b*d**3*f) + 2*C*c*d*f*x*tan(e + f*x)/(2*b*c**3*f*tan(e + f*x) + 2*I*b*c**3
*f - 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e + f*x)
 + 2*I*b*c*d**2*f - 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**3*f) + 2*I*C*c*d*f*x
/(2*b*c**3*f*tan(e + f*x) + 2*I*b*c**3*f - 2*I*b*c**2*d*f*tan(e + f*x) + 2*
b*c**2*d*f + 2*b*c*d**2*f*tan(e + f*x) + 2*I*b*c*d**2*f - 2*I*b*d**3*f*tan(
e + f*x) + 2*b*d**3*f) - I*C*d**2*f*x*tan(e + f*x)/(2*b*c**3*f*tan(e + f*x)
 + 2*I*b*c**3*f - 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f
*tan(e + f*x) + 2*I*b*c*d**2*f - 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**3*f) +
```

C*d**2*f*x/(2*b*c**3*f*tan(e + f*x) + 2*I*b*c**3*f - 2*I*b*c**2*d*f*tan(e +
f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e + f*x) + 2*I*b*c*d**2*f - 2*I*b*d
3*f*tan(e + f*x) + 2*b*d3*f) + I*C*d**2/(2*b*c**3*f*tan(e + f*x) + 2*I*
b*c**3*f - 2*I*b*c**2*d*f*tan(e + f*x) + 2*b*c**2*d*f + 2*b*c*d**2*f*tan(e
+ f*x) + 2*I*b*c*d**2*f - 2*I*b*d**3*f*tan(e + f*x) + 2*b*d**3*f), Eq(a, I*
b)), (2*A*c**3*d*f*x/(2*b*c**5*f + 2*b*c**4*d*f*tan(e + f*x) + 4*b*c**3*d**
2*f + 4*b*c**2*d**3*f*tan(e + f*x) + 2*b*c*d**4*f + 2*b*d**5*f*tan(e + f*x)
) + 2*A*c**2*d**2*f*x*tan(e + f*x)/(2*b*c**5*f + 2*b*c**4*d*f*tan(e + f*x)
+ 4*b*c**3*d**2*f + 4*b*c**2*d**3*f*tan(e + f*x) + 2*b*c*d**4*f + 2*b*d**5*
f*tan(e + f*x)) + 4*A*c**2*d**2*log(c/d + tan(e + f*x))/(2*b*c**5*f + 2*b*c
4*d*f*tan(e + f*x) + 4*b*c3*d**2*f + 4*b*c**2*d**3*f*tan(e + f*x) + 2*b
*c*d**4*f + 2*b*d**5*f*tan(e + f*x)) - 2*A*c**2*d**2*log(tan(e + f*x)**2 +
1)/(2*b*c**5*f + 2*b*c**4*d*f*tan(e + f*x) + 4*b*c**3*d**2*f + 4*b*c**2*d**
3*f*tan(e + f*x) + 2*b*c*d**4*f + 2*b*d**5*f*tan(e + f*x)) - 2*A*c**2*d**2/
(2*b*c**5*f + 2*b*c**4*d*f*tan(e + f*x) + 4*b*c**3*d**2*f + 4*b*c**2*d**3*f
*tan(e + f*x) + 2*b*c*d**4*f + 2*b*d**5*f*tan(e + f*x)) - 2*A*c*d**3*f*x/(2
*b*c**5*f + 2*b*c**4*d*f*tan(e + f*x) + 4*b*c**3*d**2*f + 4*b*c**2*d**3*f*t
an(e + f*x) + 2*b*c*d**4*f + 2*b*d**5*f*tan(e + f*x)) + 4*A*c*d**3*log(c/d
+ tan(e + f*x))*tan(e + f*x)/(2*b*c**5*f + 2*b*c**4*d*f*tan(e + f*x) + 4*b*
c**3*d**2*f + 4*b*c**2*d**3*f*tan(e + f*x) + 2*b*c*d**4*f + 2*b*d**5*f*tan(
e + f*x)) - 2*A*c*d**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*b*c**5*f +
2*b*c**4*d*f*tan(e + f*x) + 4*b*c**3*d**2*f + 4*b*c**2*d**3*f*tan(e + f*x)
+ 2*b*c*d**4*f + 2*b*d**5*f*tan(e + f*x)) - 2*A*d**4*f*x*tan(e + f*x)/(2*b*
c**5*f + 2*b*c**4*d*f*tan(e + f*x) + 4*b*c**3*d**2*f + 4*b*c**2*d**3*f*tan(
e + f*x) + 2*b*c*d**4*f + 2*b*d**5*f*tan(e + f*x)) - 2*A*d**4/(2*b*c**5*f +
2*b*c**4*d*f*tan(e + f*x) + 4*b*c**3*d**2*f + 4*b*c**2*d**3*f*tan(e + f*x)
+ 2*b*c*d**4*f + 2*b*d**5*f*tan(e + f*x)) - 2*B*c**3*d*log(c/d + tan(e + f
*x))/(2*b*c**5*f + 2*b*c**4*d*f*tan(e + f*x) + 4*b*c**3*d**2*f + 4*b*c**2*d
3*f*tan(e + f*x) + 2*b*c*d4*f + 2*b*d**5*f*tan(e + f*x)) + B*c**3*d*log
(tan(e + f*x)**2 + 1)/(2*b*c**5*f + 2*b*c**4*d*f*tan(e + f*x) + 4*b*c**3*d*
*2*f + 4*b*c**2*d**3*f*tan(e + f*x) + 2*b*c*d**4*f + 2*b*d**5*f*tan(e + f*x
)) + 2*B*c**3*d/(2*b*c**5*f + 2*b*c**4*d*f*tan(e + f*x) + 4*b*c**3*d**2*f +
4*b*c**2*d**3*f*tan(e + f*x) + 2*b*c*d**4*f + 2*b*d**5*f*tan(e + f*x)) + 4
*B*c**2*d**2*f*x/(2*b*c**5*f + 2*b*c**4*d*f*tan(e + f*x) + 4*b*c**3*d**2*f
+ 4*b*c**2*d**3*f*tan(e + f*x) + 2*b*c*d**4*f + 2*b*d**5*f*tan(e + f*x)) -
2*B*c**2*d**2*log(c/d + tan(e + f*x))*tan(e + f*x)/(2*b*c**5*f + 2*b*c**4*d
*f*tan(e + f*x) + 4*b*c**3*d**2*f + 4*b*c**2*d**3*f*tan(e + f*x) + 2*b*c*d*
*4*f + 2*b*d**5*f*tan(e + f*x)) + B*c**2*d**2*log(tan(e + f*x)**2 + 1)*tan(
e + f*x)/(2*b*c**5*f + 2*b*c**4*d*f*tan(e + f*x) + 4*b*c**3*d**2*f + 4*b*c*
*2*d**3*f*tan(e + f*x) + 2*b*c*d**4*f + 2*b*d**5*f*tan(e + f*x)) + 4*B*c*d*
*3*f*x*tan(e + f*x)/(2*b*c**5*f + 2*b*c**4*d*f*tan(e + f*x) + 4*b*c**3*d**2
*f + 4*b*c**2*d**3*f*tan(e + f*x) + 2*b*c*d**4*f + 2*b*d**5*f*tan(e + f*x))
+ 2*B*c*d**3*log(c/d + tan(e + f*x))/(2*b*c**5*f + 2*b*c**4*d*f*tan(e + f*
x) + 4*b*c**3*d**2*f + 4*b*c**2*d**3*f*tan(e + f*x) + 2*b*c*d**4*f + 2*b*d*
*5*f*tan(e + f*x)) - B*c*d**3*log(tan(e + f*x)**2 + 1)/(2*b*c**5*f + 2*b*c*

```
*4*d*f*tan(e + f*x) + 4*b*c**3*d**2*f + 4*b*c**2*d**3*f*tan(e + f*x) + 2*b*
c*d**4*f + 2*b*d**5*f*tan(e + f*x)) + 2*B*c*d**3/(2*b*c**5*f + 2*b*c**4*d*f
*tan(e + f*x) + 4*b*c**3*d**2*f + 4*b*c**2*d**3*f*tan(e + f*x) + 2*b*c*d**4
*f + 2*b*d**5*f*tan(e + f*x)) + 2*B*d**4*log(c/d + tan(e + f*x))*tan(e + f*
x)/(2*b*c**5*f + 2*b*c**4*d*f*tan(e + f*x) + 4*b*c**3*d**2*f + 4*b*c**2*d**
3*f*tan(e + f*x) + 2*b*c*d**4*f + 2*b*d**5*f*tan(e + f*x)) - B*d**4*log(tan
(e + f*x)**2 + 1)*tan(e + f*x)/(2*b*c**5*f + 2*b*c**4*d*f*tan(e + f*x) + 4*
b*c**3*d**2*f + 4*b*c**2*d**3*f*tan(e + f*x) + 2*b*c*d**4*f + 2*b*d**5*f*ta
n(e + f*x)) - 2*C*c**4/(2*b*c**5*f + 2*b*c**4*d*f*tan(e + f*x) + 4*b*c**3*d
**2*f + 4*b*c**2*d**3*f*tan(e + f*x) + 2*b*c*d**4*f + 2*b*d**5*f*tan(e + f*
x)) - 2*C*c**3*d*f*x/(2*b*c**5*f + 2*b*c**4*d*f*tan(e + f*x) + 4*b*c**3*d**
2*f + 4*b*c**2*d**3*f*tan(e + f*x) + 2*b*c*d**4*f + 2*b*d**5*f*tan(e + f*x)
) - 2*C*c**2*d**2*f*x*tan(e + f*x)/(2*b*c**5*f + 2*b*c**4*d*f*tan(e + f*x)
+ 4*b*c**3*d**2*f + 4*b*c**2*d**3*f*tan(e + f*x) + 2*b*c*d**4*f + 2*b*d**5*
f*tan(e + f*x)) - 4*C*c**2*d**2*log(c/d + tan(e + f*x))/(2*b*c**5*f + 2*b*c
**4*d*f*tan(e + f*x) + 4*b*c**3*d**2*f + 4*b*c**2*d**3*f*tan(e + f*x) + 2*b
*c*d**4*f + 2*b*d**5*f*tan(e + f*x)) + 2*C*c**2*d**2*log(tan(e + f*x)**2 +
1)/(2*b*c**5*f + 2*b*c**4*d*f*tan(e + f*x) + 4*b*c**3*d**2*f + 4*b*c**2*d**
3*f*tan(e + f*x) + 2*b*c*d**4*f + 2*b*d**5*f*tan(e + f*x)) - 2*C*c**2*d**2/
(2*b*c**5*f + 2*b*c**4*d*f*tan(e + f*x) + 4*b*c**3*d**2*f + 4*b*c**2*d**3*f
*tan(e + f*x) + 2*b*c*d**4*f + 2*b*d**5*f*tan(e + f*x)) + 2*C*c*d**3*f*x/(2
*b*c**5*f + 2*b*c**4*d*f*tan(e + f*x) + 4*b*c**3*d**2*f + 4*b*c**2*d**3*f*t
an(e + f*x) + 2*b*c*d**4*f + 2*b*d**5*f*tan(e + f*x)) - 4*C*c*d**3*log(c/d
+ tan(e + f*x))*tan(e + f*x)/(2*b*c**5*f + 2*b*c**4*d*f*tan(e + f*x) + 4*b*
c**3*d**2*f + 4*b*c**2*d**3*f*tan(e + f*x) + 2*b*c*d**4*f + 2*b*d**5*f*tan(
e + f*x)) + 2*C*c*d**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*b*c**5*f +
2*b*c**4*d*f*tan(e + f*x) + 4*b*c**3*d**2*f + 4*b*c**2*d**3*f*tan(e + f*x)
+ 2*b*c*d**4*f + 2*b*d**5*f*tan(e + f*x)) + 2*C*d**4*f*x*tan(e + f*x)/(2*b*
c**5*f + 2*b*c**4*d*f*tan(e + f*x) + 4*b*c**3*d**2*f + 4*b*c**2*d**3*f*tan(
e + f*x) + 2*b*c*d**4*f + 2*b*d**5*f*tan(e + f*x)), Eq(a, b*c/d)), (I*A*a**
2*f*x*tan(e + f*x)/(2*a**3*d*f*tan(e + f*x) - 2*I*a**3*d*f + 2*I*a**2*b*d*f
*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(e + f*x) - 2*I*a*b**2*d*f +
 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d*f) + A*a**2*f*x/(2*a**3*d*f*tan(e + f
*x) - 2*I*a**3*d*f + 2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b**2*
d*f*tan(e + f*x) - 2*I*a*b**2*d*f + 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d*f)
 + I*A*a**2/(2*a**3*d*f*tan(e + f*x) - 2*I*a**3*d*f + 2*I*a**2*b*d*f*tan(e
+ f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(e + f*x) - 2*I*a*b**2*d*f + 2*I*b*
*3*d*f*tan(e + f*x) + 2*b**3*d*f) - 2*A*a*b*f*x*tan(e + f*x)/(2*a**3*d*f*ta
n(e + f*x) - 2*I*a**3*d*f + 2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*
a*b**2*d*f*tan(e + f*x) - 2*I*a*b**2*d*f + 2*I*b**3*d*f*tan(e + f*x) + 2*b*
*3*d*f) + 2*I*A*a*b*f*x/(2*a**3*d*f*tan(e + f*x) - 2*I*a**3*d*f + 2*I*a**2*
b*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(e + f*x) - 2*I*a*b**2*
d*f + 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d*f) + I*A*b**2*f*x*tan(e + f*x)/(
2*a**3*d*f*tan(e + f*x) - 2*I*a**3*d*f + 2*I*a**2*b*d*f*tan(e + f*x) + 2*a*
*2*b*d*f + 2*a*b**2*d*f*tan(e + f*x) - 2*I*a*b**2*d*f + 2*I*b**3*d*f*tan(e
```

```
+ f*x) + 2*b**3*d*f) + A*b**2*f*x/(2*a**3*d*f*tan(e + f*x) - 2*I*a**3*d*f +
 2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(e + f*x) - 2
*I*a*b**2*d*f + 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d*f) - 2*A*b**2*log(a/b
+ tan(e + f*x))*tan(e + f*x)/(2*a**3*d*f*tan(e + f*x) - 2*I*a**3*d*f + 2*I*
a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(e + f*x) - 2*I*a*
b**2*d*f + 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d*f) + 2*I*A*b**2*log(a/b + t
an(e + f*x))/(2*a**3*d*f*tan(e + f*x) - 2*I*a**3*d*f + 2*I*a**2*b*d*f*tan(e
 + f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(e + f*x) - 2*I*a*b**2*d*f + 2*I*b
**3*d*f*tan(e + f*x) + 2*b**3*d*f) + A*b**2*log(tan(e + f*x)**2 + 1)*tan(e
+ f*x)/(2*a**3*d*f*tan(e + f*x) - 2*I*a**3*d*f + 2*I*a**2*b*d*f*tan(e + f*x
) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(e + f*x) - 2*I*a*b**2*d*f + 2*I*b**3*d*
f*tan(e + f*x) + 2*b**3*d*f) - I*A*b**2*log(tan(e + f*x)**2 + 1)/(2*a**3*d*
f*tan(e + f*x) - 2*I*a**3*d*f + 2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f
+ 2*a*b**2*d*f*tan(e + f*x) - 2*I*a*b**2*d*f + 2*I*b**3*d*f*tan(e + f*x) +
2*b**3*d*f) + I*A*b**2/(2*a**3*d*f*tan(e + f*x) - 2*I*a**3*d*f + 2*I*a**2*b
*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(e + f*x) - 2*I*a*b**2*d
*f + 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d*f) + B*a**2*f*x*tan(e + f*x)/(2*a
**3*d*f*tan(e + f*x) - 2*I*a**3*d*f + 2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*
b*d*f + 2*a*b**2*d*f*tan(e + f*x) - 2*I*a*b**2*d*f + 2*I*b**3*d*f*tan(e + f
*x) + 2*b**3*d*f) - I*B*a**2*f*x/(2*a**3*d*f*tan(e + f*x) - 2*I*a**3*d*f +
2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(e + f*x) - 2*
I*a*b**2*d*f + 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d*f) - B*a**2/(2*a**3*d*f
*tan(e + f*x) - 2*I*a**3*d*f + 2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f +
 2*a*b**2*d*f*tan(e + f*x) - 2*I*a*b**2*d*f + 2*I*b**3*d*f*tan(e + f*x) + 2
*b**3*d*f) + 2*B*a*b*log(a/b + tan(e + f*x))*tan(e + f*x)/(2*a**3*d*f*tan(e
 + f*x) - 2*I*a**3*d*f + 2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b
**2*d*f*tan(e + f*x) - 2*I*a*b**2*d*f + 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*
d*f) - 2*I*B*a*b*log(a/b + tan(e + f*x))/(2*a**3*d*f*tan(e + f*x) - 2*I*a**
3*d*f + 2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(e + f
*x) - 2*I*a*b**2*d*f + 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d*f) - B*a*b*log(
tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*a**3*d*f*tan(e + f*x) - 2*I*a**3*d*f +
 2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(e + f*x) - 2
*I*a*b**2*d*f + 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d*f) + I*B*a*b*log(tan(e
 + f*x)**2 + 1)/(2*a**3*d*f*tan(e + f*x) - 2*I*a**3*d*f + 2*I*a**2*b*d*f*ta
n(e + f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(e + f*x) - 2*I*a*b**2*d*f + 2*
I*b**3*d*f*tan(e + f*x) + 2*b**3*d*f) - B*b**2*f*x*tan(e + f*x)/(2*a**3*d*f
*tan(e + f*x) - 2*I*a**3*d*f + 2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f +
 2*a*b**2*d*f*tan(e + f*x) - 2*I*a*b**2*d*f + 2*I*b**3*d*f*tan(e + f*x) + 2
*b**3*d*f) + I*B*b**2*f*x/(2*a**3*d*f*tan(e + f*x) - 2*I*a**3*d*f + 2*I*a**
2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(e + f*x) - 2*I*a*b**
2*d*f + 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d*f) - B*b**2/(2*a**3*d*f*tan(e
+ f*x) - 2*I*a**3*d*f + 2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b*
*2*d*f*tan(e + f*x) - 2*I*a*b**2*d*f + 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d
*f) + I*C*a**2*f*x*tan(e + f*x)/(2*a**3*d*f*tan(e + f*x) - 2*I*a**3*d*f + 2
*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(e + f*x) - 2*I
```

*a*b**2*d*f + 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d*f) + C*a**2*f*x/(2*a**3*
d*f*tan(e + f*x) - 2*I*a**3*d*f + 2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*
f + 2*a*b**2*d*f*tan(e + f*x) - 2*I*a*b**2*d*f + 2*I*b**3*d*f*tan(e + f*x)
+ 2*b**3*d*f) - 2*C*a**2*log(a/b + tan(e + f*x))*tan(e + f*x)/(2*a**3*d*f*t
an(e + f*x) - 2*I*a**3*d*f + 2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2
*a*b**2*d*f*tan(e + f*x) - 2*I*a*b**2*d*f + 2*I*b**3*d*f*tan(e + f*x) + 2*b
3*d*f) + 2*I*C*a2*log(a/b + tan(e + f*x))/(2*a**3*d*f*tan(e + f*x) - 2*
I*a**3*d*f + 2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(
e + f*x) - 2*I*a*b**2*d*f + 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d*f) + C*a**
2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*a**3*d*f*tan(e + f*x) - 2*I*a**3
*d*f + 2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(e + f*
x) - 2*I*a*b**2*d*f + 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d*f) - I*C*a**2*lo
g(tan(e + f*x)**2 + 1)/(2*a**3*d*f*tan(e + f*x) - 2*I*a**3*d*f + 2*I*a**2*b
*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(e + f*x) - 2*I*a*b**2*d
*f + 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d*f) - I*C*a**2/(2*a**3*d*f*tan(e +
 f*x) - 2*I*a**3*d*f + 2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b**
2*d*f*tan(e + f*x) - 2*I*a*b**2*d*f + 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d*
f) + 2*C*a*b*f*x*tan(e + f*x)/(2*a**3*d*f*tan(e + f*x) - 2*I*a**3*d*f + 2*I
*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(e + f*x) - 2*I*a
*b**2*d*f + 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d*f) - 2*I*C*a*b*f*x/(2*a**3
*d*f*tan(e + f*x) - 2*I*a**3*d*f + 2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d
*f + 2*a*b**2*d*f*tan(e + f*x) - 2*I*a*b**2*d*f + 2*I*b**3*d*f*tan(e + f*x)
 + 2*b**3*d*f) + I*C*b**2*f*x*tan(e + f*x)/(2*a**3*d*f*tan(e + f*x) - 2*I*a
3*d*f + 2*I*a2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(e +
 f*x) - 2*I*a*b**2*d*f + 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d*f) + C*b**2*f
*x/(2*a**3*d*f*tan(e + f*x) - 2*I*a**3*d*f + 2*I*a**2*b*d*f*tan(e + f*x) +
2*a**2*b*d*f + 2*a*b**2*d*f*tan(e + f*x) - 2*I*a*b**2*d*f + 2*I*b**3*d*f*ta
n(e + f*x) + 2*b**3*d*f) - I*C*b**2/(2*a**3*d*f*tan(e + f*x) - 2*I*a**3*d*f
 + 2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(e + f*x) -
 2*I*a*b**2*d*f + 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d*f), Eq(c, -I*d)), (-
I*A*a**2*f*x*tan(e + f*x)/(2*a**3*d*f*tan(e + f*x) + 2*I*a**3*d*f - 2*I*a**
2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(e + f*x) + 2*I*a*b**
2*d*f - 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d*f) + A*a**2*f*x/(2*a**3*d*f*ta
n(e + f*x) + 2*I*a**3*d*f - 2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*
a*b**2*d*f*tan(e + f*x) + 2*I*a*b**2*d*f - 2*I*b**3*d*f*tan(e + f*x) + 2*b*
*3*d*f) - I*A*a**2/(2*a**3*d*f*tan(e + f*x) + 2*I*a**3*d*f - 2*I*a**2*b*d*f
*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(e + f*x) + 2*I*a*b**2*d*f -
 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d*f) - 2*A*a*b*f*x*tan(e + f*x)/(2*a**3
*d*f*tan(e + f*x) + 2*I*a**3*d*f - 2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d
*f + 2*a*b**2*d*f*tan(e + f*x) + 2*I*a*b**2*d*f - 2*I*b**3*d*f*tan(e + f*x)
 + 2*b**3*d*f) - 2*I*A*a*b*f*x/(2*a**3*d*f*tan(e + f*x) + 2*I*a**3*d*f - 2*
I*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(e + f*x) + 2*I*
a*b**2*d*f - 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d*f) - I*A*b**2*f*x*tan(e +
 f*x)/(2*a**3*d*f*tan(e + f*x) + 2*I*a**3*d*f - 2*I*a**2*b*d*f*tan(e + f*x)
 + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(e + f*x) + 2*I*a*b**2*d*f - 2*I*b**3*d*f

*tan(e + f*x) + 2*b**3*d*f) + A*b**2*f*x/(2*a**3*d*f*tan(e + f*x) + 2*I*a**
3*d*f - 2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(e + f
*x) + 2*I*a*b**2*d*f - 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d*f) - 2*A*b**2*l
og(a/b + tan(e + f*x))*tan(e + f*x)/(2*a**3*d*f*tan(e + f*x) + 2*I*a**3*d*f
 - 2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(e + f*x) +
 2*I*a*b**2*d*f - 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d*f) - 2*I*A*b**2*log(
a/b + tan(e + f*x))/(2*a**3*d*f*tan(e + f*x) + 2*I*a**3*d*f - 2*I*a**2*b*d*
f*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(e + f*x) + 2*I*a*b**2*d*f
 - 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d*f) + A*b**2*log(tan(e + f*x)**2 + 1)
*tan(e + f*x)/(2*a**3*d*f*tan(e + f*x) + 2*I*a**3*d*f - 2*I*a**2*b*d*f*tan(
e + f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(e + f*x) + 2*I*a*b**2*d*f - 2*I*
b**3*d*f*tan(e + f*x) + 2*b**3*d*f) + I*A*b**2*log(tan(e + f*x)**2 + 1)/(2*
a**3*d*f*tan(e + f*x) + 2*I*a**3*d*f - 2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2
*b*d*f + 2*a*b**2*d*f*tan(e + f*x) + 2*I*a*b**2*d*f - 2*I*b**3*d*f*tan(e +
f*x) + 2*b**3*d*f) - I*A*b**2/(2*a**3*d*f*tan(e + f*x) + 2*I*a**3*d*f - 2*I
*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(e + f*x) + 2*I*a
*b**2*d*f - 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d*f) + B*a**2*f*x*tan(e + f*
x)/(2*a**3*d*f*tan(e + f*x) + 2*I*a**3*d*f - 2*I*a**2*b*d*f*tan(e + f*x) +
2*a**2*b*d*f + 2*a*b**2*d*f*tan(e + f*x) + 2*I*a*b**2*d*f - 2*I*b**3*d*f*ta
n(e + f*x) + 2*b**3*d*f) + I*B*a**2*f*x/(2*a**3*d*f*tan(e + f*x) + 2*I*a**3
*d*f - 2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(e + f*
x) + 2*I*a*b**2*d*f - 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d*f) - B*a**2/(2*a
3*d*f*tan(e + f*x) + 2*I*a3*d*f - 2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*
b*d*f + 2*a*b**2*d*f*tan(e + f*x) + 2*I*a*b**2*d*f - 2*I*b**3*d*f*tan(e + f
*x) + 2*b**3*d*f) + 2*B*a*b*log(a/b + tan(e + f*x))*tan(e + f*x)/(2*a**3*d*
f*tan(e + f*x) + 2*I*a**3*d*f - 2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f
 + 2*a*b**2*d*f*tan(e + f*x) + 2*I*a*b**2*d*f - 2*I*b**3*d*f*tan(e + f*x) +
2*b**3*d*f) + 2*I*B*a*b*log(a/b + tan(e + f*x))/(2*a**3*d*f*tan(e + f*x) +
2*I*a**3*d*f - 2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*ta
n(e + f*x) + 2*I*a*b**2*d*f - 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d*f) - B*a
*b*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*a**3*d*f*tan(e + f*x) + 2*I*a**
3*d*f - 2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(e + f
*x) + 2*I*a*b**2*d*f - 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d*f) - I*B*a*b*lo
g(tan(e + f*x)**2 + 1)/(2*a**3*d*f*tan(e + f*x) + 2*I*a**3*d*f - 2*I*a**2*b
*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(e + f*x) + 2*I*a*b**2*d
*f - 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d*f) - B*b**2*f*x*tan(e + f*x)/(2*a
3*d*f*tan(e + f*x) + 2*I*a3*d*f - 2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*
b*d*f + 2*a*b**2*d*f*tan(e + f*x) + 2*I*a*b**2*d*f - 2*I*b**3*d*f*tan(e + f
*x) + 2*b**3*d*f) - I*B*b**2*f*x/(2*a**3*d*f*tan(e + f*x) + 2*I*a**3*d*f -
2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(e + f*x) + 2*
I*a*b**2*d*f - 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d*f) - B*b**2/(2*a**3*d*f
*tan(e + f*x) + 2*I*a**3*d*f - 2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f +
 2*a*b**2*d*f*tan(e + f*x) + 2*I*a*b**2*d*f - 2*I*b**3*d*f*tan(e + f*x) + 2
*b**3*d*f) - I*C*a**2*f*x*tan(e + f*x)/(2*a**3*d*f*tan(e + f*x) + 2*I*a**3*
d*f - 2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(e + f*x

) + 2*I*a*b**2*d*f - 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d*f) + C*a**2*f*x/(
2*a**3*d*f*tan(e + f*x) + 2*I*a**3*d*f - 2*I*a**2*b*d*f*tan(e + f*x) + 2*a*
*2*b*d*f + 2*a*b**2*d*f*tan(e + f*x) + 2*I*a*b**2*d*f - 2*I*b**3*d*f*tan(e
+ f*x) + 2*b**3*d*f) - 2*C*a**2*log(a/b + tan(e + f*x))*tan(e + f*x)/(2*a**
3*d*f*tan(e + f*x) + 2*I*a**3*d*f - 2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*
d*f + 2*a*b**2*d*f*tan(e + f*x) + 2*I*a*b**2*d*f - 2*I*b**3*d*f*tan(e + f*x
) + 2*b**3*d*f) - 2*I*C*a**2*log(a/b + tan(e + f*x))/(2*a**3*d*f*tan(e + f*
x) + 2*I*a**3*d*f - 2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b**2*d
*f*tan(e + f*x) + 2*I*a*b**2*d*f - 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d*f)
+ C*a**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*a**3*d*f*tan(e + f*x) + 2
*I*a**3*d*f - 2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan
(e + f*x) + 2*I*a*b**2*d*f - 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d*f) + I*C*
a**2*log(tan(e + f*x)**2 + 1)/(2*a**3*d*f*tan(e + f*x) + 2*I*a**3*d*f - 2*I
*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(e + f*x) + 2*I*a
*b**2*d*f - 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d*f) + I*C*a**2/(2*a**3*d*f*
tan(e + f*x) + 2*I*a**3*d*f - 2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f +
2*a*b**2*d*f*tan(e + f*x) + 2*I*a*b**2*d*f - 2*I*b**3*d*f*tan(e + f*x) + 2*
b**3*d*f) + 2*C*a*b*f*x*tan(e + f*x)/(2*a**3*d*f*tan(e + f*x) + 2*I*a**3*d*
f - 2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(e + f*x)
+ 2*I*a*b**2*d*f - 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d*f) + 2*I*C*a*b*f*x/
(2*a**3*d*f*tan(e + f*x) + 2*I*a**3*d*f - 2*I*a**2*b*d*f*tan(e + f*x) + 2*a
2*b*d*f + 2*a*b2*d*f*tan(e + f*x) + 2*I*a*b**2*d*f - 2*I*b**3*d*f*tan(e
 + f*x) + 2*b**3*d*f) - I*C*b**2*f*x*tan(e + f*x)/(2*a**3*d*f*tan(e + f*x)
+ 2*I*a**3*d*f - 2*I*a**2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*
tan(e + f*x) + 2*I*a*b**2*d*f - 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d*f) + C
*b**2*f*x/(2*a**3*d*f*tan(e + f*x) + 2*I*a**3*d*f - 2*I*a**2*b*d*f*tan(e +
f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(e + f*x) + 2*I*a*b**2*d*f - 2*I*b**3
*d*f*tan(e + f*x) + 2*b**3*d*f) + I*C*b**2/(2*a**3*d*f*tan(e + f*x) + 2*I*a
3*d*f - 2*I*a2*b*d*f*tan(e + f*x) + 2*a**2*b*d*f + 2*a*b**2*d*f*tan(e +
 f*x) + 2*I*a*b**2*d*f - 2*I*b**3*d*f*tan(e + f*x) + 2*b**3*d*f), Eq(c, I*d
)), (x*(A + B*tan(e) + C*tan(e)**2)/((a + b*tan(e))*(c + d*tan(e))), Eq(f,
0)), (2*A*a**2*c*d*f*x/(2*a**3*c**2*d*f + 2*a**3*d**3*f - 2*a**2*b*c**3*f -
 2*a**2*b*c*d**2*f + 2*a*b**2*c**2*d*f + 2*a*b**2*d**3*f - 2*b**3*c**3*f -
2*b**3*c*d**2*f) + 2*A*a**2*d**2*log(c/d + tan(e + f*x))/(2*a**3*c**2*d*f +
 2*a**3*d**3*f - 2*a**2*b*c**3*f - 2*a**2*b*c*d**2*f + 2*a*b**2*c**2*d*f +
2*a*b**2*d**3*f - 2*b**3*c**3*f - 2*b**3*c*d**2*f) - A*a**2*d**2*log(tan(e
+ f*x)**2 + 1)/(2*a**3*c**2*d*f + 2*a**3*d**3*f - 2*a**2*b*c**3*f - 2*a**2*
b*c*d**2*f + 2*a*b**2*c**2*d*f + 2*a*b**2*d**3*f - 2*b**3*c**3*f - 2*b**3*c
*d**2*f) - 2*A*a*b*c**2*f*x/(2*a**3*c**2*d*f + 2*a**3*d**3*f - 2*a**2*b*c**
3*f - 2*a**2*b*c*d**2*f + 2*a*b**2*c**2*d*f + 2*a*b**2*d**3*f - 2*b**3*c**3
*f - 2*b**3*c*d**2*f) - 2*A*a*b*d**2*f*x/(2*a**3*c**2*d*f + 2*a**3*d**3*f -
 2*a**2*b*c**3*f - 2*a**2*b*c*d**2*f + 2*a*b**2*c**2*d*f + 2*a*b**2*d**3*f
- 2*b**3*c**3*f - 2*b**3*c*d**2*f) - 2*A*b**2*c**2*log(a/b + tan(e + f*x))/
(2*a**3*c**2*d*f + 2*a**3*d**3*f - 2*a**2*b*c**3*f - 2*a**2*b*c*d**2*f + 2*
a*b**2*c**2*d*f + 2*a*b**2*d**3*f - 2*b**3*c**3*f - 2*b**3*c*d**2*f) + A*b*

```
*2*c**2*log(tan(e + f*x)**2 + 1)/(2*a**3*c**2*d*f + 2*a**3*d**3*f - 2*a**2*
b*c**3*f - 2*a**2*b*c*d**2*f + 2*a*b**2*c**2*d*f + 2*a*b**2*d**3*f - 2*b**3
*c**3*f - 2*b**3*c*d**2*f) + 2*A*b**2*c*d*f*x/(2*a**3*c**2*d*f + 2*a**3*d**
3*f - 2*a**2*b*c**3*f - 2*a**2*b*c*d**2*f + 2*a*b**2*c**2*d*f + 2*a*b**2*d*
*3*f - 2*b**3*c**3*f - 2*b**3*c*d**2*f) - 2*A*b**2*d**2*log(a/b + tan(e + f
*x))/(2*a**3*c**2*d*f + 2*a**3*d**3*f - 2*a**2*b*c**3*f - 2*a**2*b*c*d**2*f
 + 2*a*b**2*c**2*d*f + 2*a*b**2*d**3*f - 2*b**3*c**3*f - 2*b**3*c*d**2*f) +
 2*A*b**2*d**2*log(c/d + tan(e + f*x))/(2*a**3*c**2*d*f + 2*a**3*d**3*f - 2
*a**2*b*c**3*f - 2*a**2*b*c*d**2*f + 2*a*b**2*c**2*d*f + 2*a*b**2*d**3*f -
2*b**3*c**3*f - 2*b**3*c*d**2*f) - 2*B*a**2*c*d*log(c/d + tan(e + f*x))/(2*
a**3*c**2*d*f + 2*a**3*d**3*f - 2*a**2*b*c**3*f - 2*a**2*b*c*d**2*f + 2*a*b
**2*c**2*d*f + 2*a*b**2*d**3*f - 2*b**3*c**3*f - 2*b**3*c*d**2*f) + B*a**2*
c*d*log(tan(e + f*x)**2 + 1)/(2*a**3*c**2*d*f + 2*a**3*d**3*f - 2*a**2*b*c*
*3*f - 2*a**2*b*c*d**2*f + 2*a*b**2*c**2*d*f + 2*a*b**2*d**3*f - 2*b**3*c**
3*f - 2*b**3*c*d**2*f) + 2*B*a**2*d**2*f*x/(2*a**3*c**2*d*f + 2*a**3*d**3*f
 - 2*a**2*b*c**3*f - 2*a**2*b*c*d**2*f + 2*a*b**2*c**2*d*f + 2*a*b**2*d**3*
f - 2*b**3*c**3*f - 2*b**3*c*d**2*f) + 2*B*a*b*c**2*log(a/b + tan(e + f*x))
/(2*a**3*c**2*d*f + 2*a**3*d**3*f - 2*a**2*b*c**3*f - 2*a**2*b*c*d**2*f + 2
*a*b**2*c**2*d*f + 2*a*b**2*d**3*f - 2*b**3*c**3*f - 2*b**3*c*d**2*f) - B*a
*b*c**2*log(tan(e + f*x)**2 + 1)/(2*a**3*c**2*d*f + 2*a**3*d**3*f - 2*a**2*
b*c**3*f - 2*a**2*b*c*d**2*f + 2*a*b**2*c**2*d*f + 2*a*b**2*d**3*f - 2*b**3
*c**3*f - 2*b**3*c*d**2*f) + 2*B*a*b*d**2*log(a/b + tan(e + f*x))/(2*a**3*c
**2*d*f + 2*a**3*d**3*f - 2*a**2*b*c**3*f - 2*a**2*b*c*d**2*f + 2*a*b**2*c*
*2*d*f + 2*a*b**2*d**3*f - 2*b**3*c**3*f - 2*b**3*c*d**2*f) - B*a*b*d**2*lo
g(tan(e + f*x)**2 + 1)/(2*a**3*c**2*d*f + 2*a**3*d**3*f - 2*a**2*b*c**3*f -
 2*a**2*b*c*d**2*f + 2*a*b**2*c**2*d*f + 2*a*b**2*d**3*f - 2*b**3*c**3*f -
2*b**3*c*d**2*f) - 2*B*b**2*c**2*f*x/(2*a**3*c**2*d*f + 2*a**3*d**3*f - 2*a
**2*b*c**3*f - 2*a**2*b*c*d**2*f + 2*a*b**2*c**2*d*f + 2*a*b**2*d**3*f - 2*
b**3*c**3*f - 2*b**3*c*d**2*f) - 2*B*b**2*c*d*log(c/d + tan(e + f*x))/(2*a*
*3*c**2*d*f + 2*a**3*d**3*f - 2*a**2*b*c**3*f - 2*a**2*b*c*d**2*f + 2*a*b**
2*c**2*d*f + 2*a*b**2*d**3*f - 2*b**3*c**3*f - 2*b**3*c*d**2*f) + B*b**2*c*
d*log(tan(e + f*x)**2 + 1)/(2*a**3*c**2*d*f + 2*a**3*d**3*f - 2*a**2*b*c**3
*f - 2*a**2*b*c*d**2*f + 2*a*b**2*c**2*d*f + 2*a*b**2*d**3*f - 2*b**3*c**3*
f - 2*b**3*c*d**2*f) - 2*C*a**2*c**2*log(a/b + tan(e + f*x))/(2*a**3*c**2*d
*f + 2*a**3*d**3*f - 2*a**2*b*c**3*f - 2*a**2*b*c*d**2*f + 2*a*b**2*c**2*d*
f + 2*a*b**2*d**3*f - 2*b**3*c**3*f - 2*b**3*c*d**2*f) + 2*C*a**2*c**2*log(
c/d + tan(e + f*x))/(2*a**3*c**2*d*f + 2*a**3*d**3*f - 2*a**2*b*c**3*f - 2*
a**2*b*c*d**2*f + 2*a*b**2*c**2*d*f + 2*a*b**2*d**3*f - 2*b**3*c**3*f - 2*b
**3*c*d**2*f) - 2*C*a**2*c*d*f*x/(2*a**3*c**2*d*f + 2*a**3*d**3*f - 2*a**2*
b*c**3*f - 2*a**2*b*c*d**2*f + 2*a*b**2*c**2*d*f + 2*a*b**2*d**3*f - 2*b**3
*c**3*f - 2*b**3*c*d**2*f) - 2*C*a**2*d**2*log(a/b + tan(e + f*x))/(2*a**3*
c**2*d*f + 2*a**3*d**3*f - 2*a**2*b*c**3*f - 2*a**2*b*c*d**2*f + 2*a*b**2*c
**2*d*f + 2*a*b**2*d**3*f - 2*b**3*c**3*f - 2*b**3*c*d**2*f) + C*a**2*d**2*
log(tan(e + f*x)**2 + 1)/(2*a**3*c**2*d*f + 2*a**3*d**3*f - 2*a**2*b*c**3*f
 - 2*a**2*b*c*d**2*f + 2*a*b**2*c**2*d*f + 2*a*b**2*d**3*f - 2*b**3*c**3*f
```

```
- 2*b**3*c*d**2*f) + 2*C*a*b*c**2*f*x/(2*a**3*c**2*d*f + 2*a**3*d**3*f - 2*
a**2*b*c**3*f - 2*a**2*b*c*d**2*f + 2*a*b**2*c**2*d*f + 2*a*b**2*d**3*f - 2
*b**3*c**3*f - 2*b**3*c*d**2*f) + 2*C*a*b*d**2*f*x/(2*a**3*c**2*d*f + 2*a**
3*d**3*f - 2*a**2*b*c**3*f - 2*a**2*b*c*d**2*f + 2*a*b**2*c**2*d*f + 2*a*b*
*2*d**3*f - 2*b**3*c**3*f - 2*b**3*c*d**2*f) + 2*C*b**2*c**2*log(c/d + tan(
e + f*x))/(2*a**3*c**2*d*f + 2*a**3*d**3*f - 2*a**2*b*c**3*f - 2*a**2*b*c*d
**2*f + 2*a*b**2*c**2*d*f + 2*a*b**2*d**3*f - 2*b**3*c**3*f - 2*b**3*c*d**2
*f) - C*b**2*c**2*log(tan(e + f*x)**2 + 1)/(2*a**3*c**2*d*f + 2*a**3*d**3*f
 - 2*a**2*b*c**3*f - 2*a**2*b*c*d**2*f + 2*a*b**2*c**2*d*f + 2*a*b**2*d**3*
f - 2*b**3*c**3*f - 2*b**3*c*d**2*f) - 2*C*b**2*c*d*f*x/(2*a**3*c**2*d*f +
2*a**3*d**3*f - 2*a**2*b*c**3*f - 2*a**2*b*c*d**2*f + 2*a*b**2*c**2*d*f + 2
*a*b**2*d**3*f - 2*b**3*c**3*f - 2*b**3*c*d**2*f), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.56 (sec) , antiderivative size = 243, normalized size of antiderivative = 1.47

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))}\,dx = \frac{\frac{2\left(((A-C)a+Bb)c+(Ba-(A-C)b)d\right)(fx+e)}{(a^2+b^2)c^2+(a^2+b^2)d^2}+\frac{2\left(Ca^2-Bab+Ab^2\right)\log(b\tan(fx+e)+a)}{(a^2b+b^3)c-(a^3+ab^2)d}-\frac{2\left(Cc^2-Bcd+Ad^2\right)\log(d\tan(fx+e)+c)}{bc^3-ac^2d+bcd^2-ad^3}+\frac{((Ba}{}}{2f}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))
,x, algorithm="maxima")
```

```
[Out] 1/2*(2*(((A - C)*a + B*b)*c + (B*a - (A - C)*b)*d)*(f*x + e)/((a^2 + b^2)*c
^2 + (a^2 + b^2)*d^2) + 2*(C*a^2 - B*a*b + A*b^2)*log(b*tan(f*x + e) + a)/(
(a^2*b + b^3)*c - (a^3 + a*b^2)*d) - 2*(C*c^2 - B*c*d + A*d^2)*log(d*tan(f*
x + e) + c)/(b*c^3 - a*c^2*d + b*c*d^2 - a*d^3) + ((B*a - (A - C)*b)*c - ((
A - C)*a + B*b)*d)*log(tan(f*x + e)^2 + 1)/((a^2 + b^2)*c^2 + (a^2 + b^2)*d
^2))/f
```

Giac [A] (verification not implemented)

none

Time = 0.60 (sec) , antiderivative size = 268, normalized size of antiderivative = 1.62

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))}\,dx = \frac{\frac{2\,(Aac-Cac+Bbc+Bad-Abd+Cbd)(fx+e)}{a^2c^2+b^2c^2+a^2d^2+b^2d^2}+\frac{(Bac-Abc+Cbc-Aad+Cad-Bbd)\log\left(\tan(fx+e)^2+1\right)}{a^2c^2+b^2c^2+a^2d^2+b^2d^2}+\frac{2\left(Ca^2b-Bab^2+Ab^3\right)\log(|b\tan(f}{a^2b^2c+b^4c-a^3bd-ab^3d}}{2f}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))
,x, algorithm="giac")
```

```
[Out] 1/2*(2*(A*a*c - C*a*c + B*b*c + B*a*d - A*b*d + C*b*d)*(f*x + e)/(a^2*c^2 +
 b^2*c^2 + a^2*d^2 + b^2*d^2) + (B*a*c - A*b*c + C*b*c - A*a*d + C*a*d - B*
b*d)*log(tan(f*x + e)^2 + 1)/(a^2*c^2 + b^2*c^2 + a^2*d^2 + b^2*d^2) + 2*(C
*a^2*b - B*a*b^2 + A*b^3)*log(abs(b*tan(f*x + e) + a))/(a^2*b^2*c + b^4*c -
 a^3*b*d - a*b^3*d) - 2*(C*c^2*d - B*c*d^2 + A*d^3)*log(abs(d*tan(f*x + e)
+ c))/(b*c^3*d - a*c^2*d^2 + b*c*d^3 - a*d^4))/f
```

Mupad [B] (verification not implemented)

Time = 20.81 (sec) , antiderivative size = 196, normalized size of antiderivative = 1.19

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))}\,dx = \frac{\ln(c+d\tan(e+f\,x))\,\left(C\,c^2-B\,c\,d+A\,d^2\right)}{f\,\left(a\,d-b\,c\right)\,\left(c^2+d^2\right)}+\frac{\ln(\tan(e+f\,x)+1\mathrm{i})\,\left(C-A+B\,1\mathrm{i}\right)}{2\,f\,\left(a\,c\,1\mathrm{i}+a\,d+b\,c-b\,d\,1\mathrm{i}\right)} - \frac{\ln(a+b\tan(e+f\,x))\,\left(C\,a^2-B\,a\,b+A\,b^2\right)}{f\,\left(d\,a^3-c\,a^2\,b+d\,a\,b^2-c\,b^3\right)}-\frac{\ln(\tan(e+f\,x)-\mathrm{i})\,\left(A-C+B\,1\mathrm{i}\right)}{2\,f\,\left(a\,d-a\,c\,1\mathrm{i}+b\,c+b\,d\,1\mathrm{i}\right)}$$

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))*(c + d*ta
n(e + f*x))),x)
```

```
[Out] (log(tan(e + f*x) + 1i)*(B*1i - A + C))/(2*f*(a*c*1i + a*d + b*c - b*d*1i))
 - (log(tan(e + f*x) - 1i)*(A + B*1i - C))/(2*f*(a*d - a*c*1i + b*c + b*d*1
i)) - (log(a + b*tan(e + f*x))*(A*b^2 + C*a^2 - B*a*b))/(f*(a^3*d - b^3*c -
 a^2*b*c + a*b^2*d)) + (log(c + d*tan(e + f*x))*(A*d^2 + C*c^2 - B*c*d))/(f
*(a*d - b*c)*(c^2 + d^2))
```

3.75 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))}\,dx$

Optimal result 723
Rubi [A] (verified) 724
Mathematica [B] (verified) 726
Maple [A] (verified) 727
Fricas [B] (verification not implemented) 727
Sympy [F(-2)] 728
Maxima [A] (verification not implemented) 728
Giac [B] (verification not implemented) 729
Mupad [B] (verification not implemented) 730

Optimal result

Integrand size = 45, antiderivative size = 281

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))}\,dx$$
$$= \frac{(a^2(Ac-cC+Bd)-b^2(Ac-cC+Bd)+2ab(Bc-(A-C)d))\,x}{(a^2+b^2)^2\,(c^2+d^2)}$$
$$+ \frac{(2ab^3c(A-C)+2a^3bBd-a^4Cd+b^4(Bc-Ad)-a^2b^2(Bc+3Ad-Cd))\log(a\cos(e+fx)+b\sin(e}{(a^2+b^2)^2\,(bc-ad)^2f}$$
$$+ \frac{d(c^2C-Bcd+Ad^2)\log(c\cos(e+fx)+d\sin(e+fx))}{(bc-ad)^2\,(c^2+d^2)\,f}$$
$$- \frac{Ab^2-a(bB-aC)}{(a^2+b^2)\,(bc-ad)f(a+b\tan(e+fx))}$$

```
[Out] (a^2*(A*c+B*d-C*c)-b^2*(A*c+B*d-C*c)+2*a*b*(B*c-(A-C)*d))*x/(a^2+b^2)^2/(c^
2+d^2)+(2*a*b^3*c*(A-C)+2*a^3*b*B*d-a^4*C*d+b^4*(-A*d+B*c)-a^2*b^2*(3*A*d+B
*c-C*d))*ln(a*cos(f*x+e)+b*sin(f*x+e))/(a^2+b^2)^2/(-a*d+b*c)^2/f+d*(A*d^2-
B*c*d+C*c^2)*ln(c*cos(f*x+e)+d*sin(f*x+e))/(-a*d+b*c)^2/(c^2+d^2)/f+(-A*b^2
+a*(B*b-C*a))/(a^2+b^2)/(-a*d+b*c)/f/(a+b*tan(f*x+e))
```

Rubi [A] (verified)

Time = 0.87 (sec) , antiderivative size = 281, normalized size of antiderivative = 1.00, number of steps used = 4, number of rules used = 3, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.067, Rules used = {3730, 3732, 3611}

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))}\,dx$$
$$=\frac{x(a^2(Ac+Bd-cC)+2ab(Bc-d(A-C))-b^2(Ac+Bd-cC))}{(a^2+b^2)^2(c^2+d^2)}$$
$$-\frac{Ab^2-a(bB-aC)}{f(a^2+b^2)(bc-ad)(a+b\tan(e+fx))}$$
$$+\frac{(a^4(-C)d+2a^3bBd-a^2b^2(3Ad+Bc-Cd)+2ab^3c(A-C)+b^4(Bc-Ad))\log(a\cos(e+fx)+b\sin(}{f(a^2+b^2)^2(bc-ad)^2}$$
$$+\frac{d(Ad^2-Bcd+c^2C)\log(c\cos(e+fx)+d\sin(e+fx))}{f(c^2+d^2)(bc-ad)^2}$$

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^2*(c + d*
Tan[e + f*x])),x]
```

```
[Out] ((a^2*(A*c - c*C + B*d) - b^2*(A*c - c*C + B*d) + 2*a*b*(B*c - (A - C)*d))*
x)/((a^2 + b^2)^2*(c^2 + d^2)) + ((2*a*b^3*c*(A - C) + 2*a^3*b*B*d - a^4*C*
d + b^4*(B*c - A*d) - a^2*b^2*(B*c + 3*A*d - C*d))*Log[a*Cos[e + f*x] + b*S
in[e + f*x]])/((a^2 + b^2)^2*(b*c - a*d)^2*f) + (d*(c^2*C - B*c*d + A*d^2)*
Log[c*Cos[e + f*x] + d*Sin[e + f*x]])/((b*c - a*d)^2*(c^2 + d^2)*f) - (A*b^
2 - a*(b*B - a*C))/((a^2 + b^2)*(b*c - a*d)*f*(a + b*Tan[e + f*x]))
```

Rule 3611

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c/(b*f))*Log[RemoveContent[a*Cos[e + f*x] + b*Si
n[e + f*x], x]], x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3730

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 - a*(b*B - a*C))*(a + b*Tan[e +
 f*x])^(m + 1)*((c + d*Tan[e + f*x])^(n + 1)/(f*(m + 1)*(b*c - a*d)*(a^2 +
b^2))), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
```

```
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3732

```
Int[((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^
2)/(((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])), x_Symbol] :> Simp[(a*(A*c - c*C + B*d) + b*(B*c - A*d + C*d))*(x/
((a^2 + b^2)*(c^2 + d^2))), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/((b*c - a*d)
*(a^2 + b^2)), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] - Dist
[(c^2*C - B*c*d + A*d^2)/((b*c - a*d)*(c^2 + d^2)), Int[(d - c*Tan[e + f*x]
)/(c + d*Tan[e + f*x]), x], x]) /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] &&
 NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0]
```

Rubi steps

$$
\begin{aligned}
\text{integral} &= -\frac{Ab^2-a(bB-aC)}{(a^2+b^2)\,(bc-ad)f(a+b\tan(e+fx))} \\
&\quad -\frac{\int \frac{-abc(A-C)+a^2Ad-b^2(Bc-Ad)+(Ab-aB-bC)(bc-ad)\tan(e+fx)+\left(Ab^2-a(bB-aC)\right)d\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))}\,dx}{(a^2+b^2)\,(bc-ad)} \\
&= \frac{(a^2(Ac-cC+Bd)-b^2(Ac-cC+Bd)+2ab(Bc-(A-C)d))\,x}{(a^2+b^2)^2\,(c^2+d^2)} \\
&\quad -\frac{Ab^2-a(bB-aC)}{(a^2+b^2)\,(bc-ad)f(a+b\tan(e+fx))} \\
&\quad +\frac{(d(c^2C-Bcd+Ad^2))\int\frac{d-c\tan(e+fx)}{c+d\tan(e+fx)}\,dx}{(bc-ad)^2\,(c^2+d^2)} \\
&\quad +\frac{(2ab^3c(A-C)+2a^3bBd-a^4Cd+b^4(Bc-Ad)-a^2b^2(Bc+3Ad-Cd))\int\frac{b-a\tan(e+fx)}{a+b\tan(e+fx)}\,dx}{(a^2+b^2)^2\,(bc-ad)^2} \\
&= \frac{(a^2(Ac-cC+Bd)-b^2(Ac-cC+Bd)+2ab(Bc-(A-C)d))\,x}{(a^2+b^2)^2\,(c^2+d^2)} \\
&\quad +\frac{(2ab^3c(A-C)+2a^3bBd-a^4Cd+b^4(Bc-Ad)-a^2b^2(Bc+3Ad-Cd))\log(a\cos(e+fx)+}{(a^2+b^2)^2\,(bc-ad)^2f} \\
&\quad +\frac{d(c^2C-Bcd+Ad^2)\log(c\cos(e+fx)+d\sin(e+fx))}{(bc-ad)^2\,(c^2+d^2)\,f} \\
&\quad -\frac{Ab^2-a(bB-aC)}{(a^2+b^2)\,(bc-ad)f(a+b\tan(e+fx))}
\end{aligned}
$$

Mathematica [B] (verified)

Leaf count is larger than twice the leaf count of optimal. 572 vs. $2(281) = 562$.

Time = 7.29 (sec) , antiderivative size = 572, normalized size of antiderivative = 2.04

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))}\,dx =$$

$$-\frac{b(bc-ad)\left(2aAbc-a^2Bc+b^2Bc-2abcC+a^2Ad-Ab^2d+2abBd-a^2Cd+b^2Cd+\frac{\sqrt{-b^2}\left(a^2(Ac-cC+Bd)-b^2(Ac-cC+Bd)+2ab(Bc-(A-C)d)\right)}{b}\right)\log\Big(\sqrt{}}{2(a^2+b^2)(c^2+d^2)}$$

$$-\frac{Ab^2-a(bB-aC)}{(a^2+b^2)\,(bc-ad)f(a+b\tan(e+fx))}$$

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^2*(
c + d*Tan[e + f*x])),x]
```

```
[Out] -(((b*(b*c - a*d)*(2*a*A*b*c - a^2*B*c + b^2*B*c - 2*a*b*c*C + a^2*A*d - A*
b^2*d + 2*a*b*B*d - a^2*C*d + b^2*C*d + (Sqrt[-b^2]*(a^2*(A*c - c*C + B*d)
- b^2*(A*c - c*C + B*d) + 2*a*b*(B*c - (A - C)*d)))/b)*Log[Sqrt[-b^2] - b*T
an[e + f*x]])/(2*(a^2 + b^2)*(c^2 + d^2)) - (b*(2*a*b^3*c*(A - C) + 2*a^3*b
*B*d - a^4*C*d + b^4*(B*c - A*d) - a^2*b^2*(B*c + 3*A*d - C*d))*Log[a + b*T
an[e + f*x]])/((a^2 + b^2)*(b*c - a*d)) + (b*(b*c - a*d)*(2*a*A*b*c - a^2*B
*c + b^2*B*c - 2*a*b*c*C + a^2*A*d - A*b^2*d + 2*a*b*B*d - a^2*C*d + b^2*C*
d - (Sqrt[-b^2]*(a^2*(A*c - c*C + B*d) - b^2*(A*c - c*C + B*d) + 2*a*b*(B*c
 - (A - C)*d)))/b)*Log[Sqrt[-b^2] + b*Tan[e + f*x]])/(2*(a^2 + b^2)*(c^2 +
d^2)) - (b*(a^2 + b^2)*d*(c^2*C - B*c*d + A*d^2)*Log[c + d*Tan[e + f*x]])/(
(b*c - a*d)*(c^2 + d^2)))/(b*(a^2 + b^2)*(b*c - a*d)*f)) - (A*b^2 - a*(b*B
- a*C))/((a^2 + b^2)*(b*c - a*d)*f*(a + b*Tan[e + f*x]))
```

Maple [A] (verified)

Time = 0.52 (sec) , antiderivative size = 364, normalized size of antiderivative = 1.30

method	result
derivativedivides	$\frac{-\frac{\left(3A\,a^2b^2d-2Aa\,b^3c+A\,b^4d-2a^3bBd+B\,a^2b^2c-B\,b^4c+a^4Cd-C\,a^2b^2d+2Ca\,b^3c\right)\ln(a+b\tan(fx+e))}{(ad-bc)^2\left(a^2+b^2\right)^2}+\frac{A\,b^2-Bab+C\,a^2}{(ad-bc)\left(a^2+b^2\right)(a+b\tan(}$
default	$\frac{-\frac{\left(3A\,a^2b^2d-2Aa\,b^3c+A\,b^4d-2a^3bBd+B\,a^2b^2c-B\,b^4c+a^4Cd-C\,a^2b^2d+2Ca\,b^3c\right)\ln(a+b\tan(fx+e))}{(ad-bc)^2\left(a^2+b^2\right)^2}+\frac{A\,b^2-Bab+C\,a^2}{(ad-bc)\left(a^2+b^2\right)(a+b\tan(}$
norman	$\frac{\frac{a\left(A\,a^2c-2Aabd-A\,b^2c+B\,a^2d+2Babc-B\,b^2d-C\,a^2c+2Cabd+C\,b^2c\right)x}{\left(a^4+2a^2b^2+b^4\right)\left(c^2+d^2\right)}+\frac{A\,b^3-Ba\,b^2+C\,a^2b}{bf(ad-bc)\left(a^2+b^2\right)}+\frac{\left(A\,a^2c-2Aabd-A\,b^2c+B\,a^2d+2B}{\left(a^4+2\right.}}{a+b\tan(fx+e)}$
parallelrisch	Expression too large to display
risch	Expression too large to display

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e)),x,m
ethod=_RETURNVERBOSE)
```

```
[Out] 1/f*(-(3*A*a^2*b^2*d-2*A*a*b^3*c+A*b^4*d-2*B*a^3*b*d+B*a^2*b^2*c-B*b^4*c+C*
a^4*d-C*a^2*b^2*d+2*C*a*b^3*c)/(a*d-b*c)^2/(a^2+b^2)^2*ln(a+b*tan(f*x+e))+(
A*b^2-B*a*b+C*a^2)/(a*d-b*c)/(a^2+b^2)/(a+b*tan(f*x+e))+1/(a^2+b^2)^2/(c^2+
d^2)*(1/2*(-A*a^2*d-2*A*a*b*c+A*b^2*d+B*a^2*c-2*B*a*b*d-B*b^2*c+C*a^2*d+2*C
*a*b*c-C*b^2*d)*ln(1+tan(f*x+e)^2)+(A*a^2*c-2*A*a*b*d-A*b^2*c+B*a^2*d+2*B*a
*b*c-B*b^2*d-C*a^2*c+2*C*a*b*d+C*b^2*c)*arctan(tan(f*x+e)))+(A*d^2-B*c*d+C*
c^2)*d/(a*d-b*c)^2/(c^2+d^2)*ln(c+d*tan(f*x+e)))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1345 vs. 2(280) = 560.

Time = 1.08 (sec) , antiderivative size = 1345, normalized size of antiderivative = 4.79

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))}\,dx = \text{Too large to display}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e
)),x, algorithm="fricas")
```

```
[Out] -1/2*(2*(C*a^2*b^3 - B*a*b^4 + A*b^5)*c^3 - 2*(C*a^3*b^2 - B*a^2*b^3 + A*a*
b^4)*c^2*d + 2*(C*a^2*b^3 - B*a*b^4 + A*b^5)*c*d^2 - 2*(C*a^3*b^2 - B*a^2*b
^3 + A*a*b^4)*d^3 - 2*(((A - C)*a^3*b^2 + 2*B*a^2*b^3 - (A - C)*a*b^4)*c^3
- (2*(A - C)*a^4*b + 3*B*a^3*b^2 + B*a*b^4)*c^2*d + ((A - C)*a^5 + 3*(A - C
)*a^3*b^2 + 2*B*a^2*b^3)*c*d^2 + (B*a^5 - 2*(A - C)*a^4*b - B*a^3*b^2)*d^3)
*f*x + ((B*a^3*b^2 - 2*(A - C)*a^2*b^3 - B*a*b^4)*c^3 + (C*a^5 - 2*B*a^4*b
```

```
+ (3*A - C)*a^3*b^2 + A*a*b^4)*c^2*d + (B*a^3*b^2 - 2*(A - C)*a^2*b^3 - B*a
*b^4)*c*d^2 + (C*a^5 - 2*B*a^4*b + (3*A - C)*a^3*b^2 + A*a*b^4)*d^3 + ((B*a
^2*b^3 - 2*(A - C)*a*b^4 - B*b^5)*c^3 + (C*a^4*b - 2*B*a^3*b^2 + (3*A - C)*
a^2*b^3 + A*b^5)*c^2*d + (B*a^2*b^3 - 2*(A - C)*a*b^4 - B*b^5)*c*d^2 + (C*a
^4*b - 2*B*a^3*b^2 + (3*A - C)*a^2*b^3 + A*b^5)*d^3)*tan(f*x + e))*log((b^2
*tan(f*x + e)^2 + 2*a*b*tan(f*x + e) + a^2)/(tan(f*x + e)^2 + 1)) - ((C*a^5
 + 2*C*a^3*b^2 + C*a*b^4)*c^2*d - (B*a^5 + 2*B*a^3*b^2 + B*a*b^4)*c*d^2 + (
A*a^5 + 2*A*a^3*b^2 + A*a*b^4)*d^3 + ((C*a^4*b + 2*C*a^2*b^3 + C*b^5)*c^2*d
 - (B*a^4*b + 2*B*a^2*b^3 + B*b^5)*c*d^2 + (A*a^4*b + 2*A*a^2*b^3 + A*b^5)*
d^3)*tan(f*x + e))*log((d^2*tan(f*x + e)^2 + 2*c*d*tan(f*x + e) + c^2)/(tan
(f*x + e)^2 + 1)) - 2*((C*a^3*b^2 - B*a^2*b^3 + A*a*b^4)*c^3 - (C*a^4*b - B
*a^3*b^2 + A*a^2*b^3)*c^2*d + (C*a^3*b^2 - B*a^2*b^3 + A*a*b^4)*c*d^2 - (C*
a^4*b - B*a^3*b^2 + A*a^2*b^3)*d^3 + (((A - C)*a^2*b^3 + 2*B*a*b^4 - (A - C
)*b^5)*c^3 - (2*(A - C)*a^3*b^2 + 3*B*a^2*b^3 + B*b^5)*c^2*d + ((A - C)*a^4
*b + 3*(A - C)*a^2*b^3 + 2*B*a*b^4)*c*d^2 + (B*a^4*b - 2*(A - C)*a^3*b^2 -
B*a^2*b^3)*d^3)*f*x)*tan(f*x + e))/(((a^4*b^3 + 2*a^2*b^5 + b^7)*c^4 - 2*(a
^5*b^2 + 2*a^3*b^4 + a*b^6)*c^3*d + (a^6*b + 3*a^4*b^3 + 3*a^2*b^5 + b^7)*c
^2*d^2 - 2*(a^5*b^2 + 2*a^3*b^4 + a*b^6)*c*d^3 + (a^6*b + 2*a^4*b^3 + a^2*b
^5)*d^4)*f*tan(f*x + e) + ((a^5*b^2 + 2*a^3*b^4 + a*b^6)*c^4 - 2*(a^6*b + 2
*a^4*b^3 + a^2*b^5)*c^3*d + (a^7 + 3*a^5*b^2 + 3*a^3*b^4 + a*b^6)*c^2*d^2 -
 2*(a^6*b + 2*a^4*b^3 + a^2*b^5)*c*d^3 + (a^7 + 2*a^5*b^2 + a^3*b^4)*d^4)*f
)
```

Sympy [F(-2)]

Exception generated.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))}\,dx = \text{Exception raised: NotImplementedError}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e))**2/(c+d*tan(f*x
+e)),x)
```

```
[Out] Exception raised: NotImplementedError >> no valid subset found
```

Maxima [A] (verification not implemented)

none

Time = 0.33 (sec) , antiderivative size = 520, normalized size of antiderivative = 1.85

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))}\,dx$$
$$= \frac{\frac{2\left(\left(\left(A-C\right)a^2+2\,Bab-\left(A-C\right)b^2\right)c+\left(Ba^2-2\,\left(A-C\right)ab-Bb^2\right)d\right)(fx+e)}{\left(a^4+2\,a^2b^2+b^4\right)c^2+\left(a^4+2\,a^2b^2+b^4\right)d^2} - \frac{2\left(\left(Ba^2b^2-2\,\left(A-C\right)ab^3-Bb^4\right)c+\left(Ca^4-2\,Ba^3b+\left(3\,A-C\right)a^2b^2+Ab^4\right)\cdots}{\left(a^4b^2+2\,a^2b^4+b^6\right)c^2-2\,\left(a^5b+2\,a^3b^3+ab^5\right)cd+\left(a^6+2\,a^4b^2\cdots\right.}}{\cdots}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e
)),x, algorithm="maxima")
```

```
[Out] 1/2*(2*(((A - C)*a^2 + 2*B*a*b - (A - C)*b^2)*c + (B*a^2 - 2*(A - C)*a*b -
B*b^2)*d)*(f*x + e)/((a^4 + 2*a^2*b^2 + b^4)*c^2 + (a^4 + 2*a^2*b^2 + b^4)*
d^2) - 2*((B*a^2*b^2 - 2*(A - C)*a*b^3 - B*b^4)*c + (C*a^4 - 2*B*a^3*b + (3
*A - C)*a^2*b^2 + A*b^4)*d)*log(b*tan(f*x + e) + a)/((a^4*b^2 + 2*a^2*b^4 +
 b^6)*c^2 - 2*(a^5*b + 2*a^3*b^3 + a*b^5)*c*d + (a^6 + 2*a^4*b^2 + a^2*b^4)
*d^2) + 2*(C*c^2*d - B*c*d^2 + A*d^3)*log(d*tan(f*x + e) + c)/(b^2*c^4 - 2*
a*b*c^3*d - 2*a*b*c*d^3 + a^2*d^4 + (a^2 + b^2)*c^2*d^2) + ((B*a^2 - 2*(A -
 C)*a*b - B*b^2)*c - ((A - C)*a^2 + 2*B*a*b - (A - C)*b^2)*d)*log(tan(f*x +
 e)^2 + 1)/((a^4 + 2*a^2*b^2 + b^4)*c^2 + (a^4 + 2*a^2*b^2 + b^4)*d^2) - 2*
(C*a^2 - B*a*b + A*b^2)/((a^3*b + a*b^3)*c - (a^4 + a^2*b^2)*d + ((a^2*b^2
+ b^4)*c - (a^3*b + a*b^3)*d)*tan(f*x + e)))/f
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 832 vs. $2(280) = 560$.

Time = 0.77 (sec) , antiderivative size = 832, normalized size of antiderivative = 2.96

$$\int \frac{A + B\tan(e+fx) + C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))}\,dx$$
$$= \frac{\frac{2\left(Aa^2c-Ca^2c+2\,Babc-Ab^2c+Cb^2c+Ba^2d-2\,Aabd+2\,Cabd-Bb^2d\right)(fx+e)}{a^4c^2+2\,a^2b^2c^2+b^4c^2+a^4d^2+2\,a^2b^2d^2+b^4d^2} + \frac{\left(Ba^2c-2\,Aabc+2\,Cabc-Bb^2c-Aa^2d+Ca^2d-2\,Babd+Ab^2d-\right.}{a^4c^2+2\,a^2b^2c^2+b^4c^2+a^4d^2+2\,a^2b^2d^2+}}{}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e
)),x, algorithm="giac")
```

```
[Out] 1/2*(2*(A*a^2*c - C*a^2*c + 2*B*a*b*c - A*b^2*c + C*b^2*c + B*a^2*d - 2*A*a
*b*d + 2*C*a*b*d - B*b^2*d)*(f*x + e)/(a^4*c^2 + 2*a^2*b^2*c^2 + b^4*c^2 +
a^4*d^2 + 2*a^2*b^2*d^2 + b^4*d^2) + (B*a^2*c - 2*A*a*b*c + 2*C*a*b*c - B*b
^2*c - A*a^2*d + C*a^2*d - 2*B*a*b*d + A*b^2*d - C*b^2*d)*log(tan(f*x + e)^
2 + 1)/(a^4*c^2 + 2*a^2*b^2*c^2 + b^4*c^2 + a^4*d^2 + 2*a^2*b^2*d^2 + b^4*d
^2) - 2*(B*a^2*b^3*c - 2*A*a*b^4*c + 2*C*a*b^4*c - B*b^5*c + C*a^4*b*d - 2*
B*a^3*b^2*d + 3*A*a^2*b^3*d - C*a^2*b^3*d + A*b^5*d)*log(abs(b*tan(f*x + e)
 + a))/(a^4*b^3*c^2 + 2*a^2*b^5*c^2 + b^7*c^2 - 2*a^5*b^2*c*d - 4*a^3*b^4*c
*d - 2*a*b^6*c*d + a^6*b*d^2 + 2*a^4*b^3*d^2 + a^2*b^5*d^2) + 2*(C*c^2*d^2
- B*c*d^3 + A*d^4)*log(abs(d*tan(f*x + e) + c))/(b^2*c^4*d - 2*a*b*c^3*d^2
+ a^2*c^2*d^3 + b^2*c^2*d^3 - 2*a*b*c*d^4 + a^2*d^5) + 2*(B*a^2*b^3*c*tan(f
*x + e) - 2*A*a*b^4*c*tan(f*x + e) + 2*C*a*b^4*c*tan(f*x + e) - B*b^5*c*tan
(f*x + e) + C*a^4*b*d*tan(f*x + e) - 2*B*a^3*b^2*d*tan(f*x + e) + 3*A*a^2*b
^3*d*tan(f*x + e) - C*a^2*b^3*d*tan(f*x + e) + A*b^5*d*tan(f*x + e) - C*a^4
*b*c + 2*B*a^3*b^2*c - 3*A*a^2*b^3*c + C*a^2*b^3*c - A*b^5*c + 2*C*a^5*d -
3*B*a^4*b*d + 4*A*a^3*b^2*d - B*a^2*b^3*d + 2*A*a*b^4*d)/((a^4*b^2*c^2 + 2*
```

```
a^2*b^4*c^2 + b^6*c^2 - 2*a^5*b*c*d - 4*a^3*b^3*c*d - 2*a*b^5*c*d + a^6*d^2
 + 2*a^4*b^2*d^2 + a^2*b^4*d^2)*(b*tan(f*x + e) + a)))/f
```

Mupad [B] (verification not implemented)

Time = 60.06 (sec) , antiderivative size = 393, normalized size of antiderivative = 1.40

$$\int \frac{A + B\tan(e+fx) + C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))}\,dx$$
$$= \frac{\ln\left(\tan(e+f\,x)-\mathrm{i}\right)\,\left(B - A\,1\mathrm{i} + C\,1\mathrm{i}\right)}{2\,f\,\left(a^2\,c - b^2\,c - 2\,a\,b\,d + a^2\,d\,1\mathrm{i} - b^2\,d\,1\mathrm{i} + a\,b\,c\,2\mathrm{i}\right)}$$
$$- \frac{\ln\left(\tan(e+f\,x)+1\mathrm{i}\right)\,\left(A\,1\mathrm{i} + B - C\,1\mathrm{i}\right)}{2\,f\,\left(b^2\,c - a^2\,c + 2\,a\,b\,d + a^2\,d\,1\mathrm{i} - b^2\,d\,1\mathrm{i} + a\,b\,c\,2\mathrm{i}\right)}$$
$$- \frac{\ln\left(a+b\tan(e+f\,x)\right)\,\left(C\,d\,a^4 - 2\,B\,d\,a^3\,b + \left(3\,A\,d + B\,c - C\,d\right)\,a^2\,b^2 + \left(2\,C\,c - 2\,A\,c\right)\,a\,b^3 + \left(A\,d - B\right.\right.}{f\,\left(a^6\,d^2 - 2\,a^5\,b\,c\,d + a^4\,b^2\,c^2 + 2\,a^4\,b^2\,d^2 - 4\,a^3\,b^3\,c\,d + 2\,a^2\,b^4\,c^2 + a^2\,b^4\,d^2 - 2\,a\,b^5\,c\,d + b^6\,c^2\right)}$$
$$+ \frac{C\,a^2 - B\,a\,b + A\,b^2}{f\,\left(a\,d - b\,c\right)\,\left(a^2+b^2\right)\,\left(a + b\tan\left(e+f\,x\right)\right)}$$
$$+ \frac{d\,\ln\left(c+d\tan(e+f\,x)\right)\,\left(C\,c^2 - B\,c\,d + A\,d^2\right)}{f\left(a\,d - b\,c\right)^2\,\left(c^2+d^2\right)}$$

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))^2*(c + d*
tan(e + f*x))),x)
```

```
[Out] (log(tan(e + f*x) - 1i)*(B - A*1i + C*1i))/(2*f*(a^2*c + a^2*d*1i - b^2*c -
 b^2*d*1i + a*b*c*2i - 2*a*b*d)) - (log(tan(e + f*x) + 1i)*(A*1i + B - C*1i
))/(2*f*(a^2*d*1i - a^2*c + b^2*c - b^2*d*1i + a*b*c*2i + 2*a*b*d)) - (log(
a + b*tan(e + f*x))*(b^4*(A*d - B*c) + a^2*b^2*(3*A*d + B*c - C*d) + C*a^4*
d - a*b^3*(2*A*c - 2*C*c) - 2*B*a^3*b*d))/(f*(a^6*d^2 + b^6*c^2 + 2*a^2*b^4
*c^2 + a^4*b^2*c^2 + a^2*b^4*d^2 + 2*a^4*b^2*d^2 - 2*a*b^5*c*d - 2*a^5*b*c*
d - 4*a^3*b^3*c*d)) + (A*b^2 + C*a^2 - B*a*b)/(f*(a*d - b*c)*(a^2 + b^2)*(a
 + b*tan(e + f*x))) + (d*log(c + d*tan(e + f*x))*(A*d^2 + C*c^2 - B*c*d))/(
f*(a*d - b*c)^2*(c^2 + d^2))
```

3.76 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^3(c+d\tan(e+fx))}\,dx$

Optimal result 731
Rubi [A] (verified) 732
Mathematica [A] (verified) 734
Maple [A] (verified) 735
Fricas [B] (verification not implemented) 736
Sympy [F(-2)] 738
Maxima [B] (verification not implemented) 738
Giac [B] (verification not implemented) 739
Mupad [B] (verification not implemented) 740

Optimal result

Integrand size = 45, antiderivative size = 477

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^3(c+d\tan(e+fx))}\,dx$$

$$= \frac{(a^3(Ac-cC+Bd)-3ab^2(Ac-cC+Bd)+3a^2b(Bc-(A-C)d)-b^3(Bc-(A-C)d))\,x}{(a^2+b^2)^3\,(c^2+d^2)}$$

$$+\frac{(3ab^5Bc^2-3a^5bBd^2+a^6Cd^2+3a^4b^2d(Bc+2Ad-Cd)+b^6(c(cC-Bd)-A(c^2-d^2))-a^3b^3(8c(A}{(a^2+b^2)^3\,(bc-}$$

$$-\frac{d^2(c^2C-Bcd+Ad^2)\log(c\cos(e+fx)+d\sin(e+fx))}{(bc-ad)^3\,(c^2+d^2)\,f}$$

$$-\frac{Ab^2-a(bB-aC)}{2\,(a^2+b^2)\,(bc-ad)f(a+b\tan(e+fx))^2}$$

$$-\frac{2ab^3c(A-C)+2a^3bBd-a^4Cd+b^4(Bc-Ad)-a^2b^2(Bc+3Ad-Cd)}{(a^2+b^2)^2\,(bc-ad)^2f(a+b\tan(e+fx))}$$

```
[Out] (a^3*(A*c+B*d-C*c)-3*a*b^2*(A*c+B*d-C*c)+3*a^2*b*(B*c-(A-C)*d)-b^3*(B*c-(A-
C)*d))*x/(a^2+b^2)^3/(c^2+d^2)+(3*a*b^5*B*c^2-3*a^5*b*B*d^2+a^6*C*d^2+3*a^4
*b^2*d*(2*A*d+B*c-C*d)+b^6*(c*(-B*d+C*c)-A*(c^2-d^2))-a^3*b^3*(8*c*(A-C)*d+
B*(c^2-d^2))-3*a^2*b^4*(c*(2*B*d+C*c)-A*(c^2+d^2)))*ln(a*cos(f*x+e)+b*sin(f
*x+e))/(a^2+b^2)^3/(-a*d+b*c)^3/f-d^2*(A*d^2-B*c*d+C*c^2)*ln(c*cos(f*x+e)+d
*sin(f*x+e))/(-a*d+b*c)^3/(c^2+d^2)/f+1/2*(-A*b^2+a*(B*b-C*a))/(a^2+b^2)/(-
a*d+b*c)/f/(a+b*tan(f*x+e))^2+(-2*a*b^3*c*(A-C)-2*a^3*b*B*d+a^4*C*d-b^4*(-A
*d+B*c)+a^2*b^2*(3*A*d+B*c-C*d))/(a^2+b^2)^2/(-a*d+b*c)^2/f/(a+b*tan(f*x+e)
)
```

Rubi [A] (verified)

Time = 1.94 (sec) , antiderivative size = 477, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 3, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.067, Rules used = {3730, 3732, 3611}

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^3(c+d\tan(e+fx))}\,dx$$
$$= -\frac{Ab^2-a(bB-aC)}{2f\left(a^2+b^2\right)(bc-ad)(a+b\tan(e+fx))^2}$$
$$+\frac{x(a^3(Ac+Bd-cC)+3a^2b(Bc-d(A-C))-3ab^2(Ac+Bd-cC)-b^3(Bc-d(A-C)))}{\left(a^2+b^2\right)^3\left(c^2+d^2\right)}$$
$$-\frac{a^4(-C)d+2a^3bBd-a^2b^2(3Ad+Bc-Cd)+2ab^3c(A-C)+b^4(Bc-Ad)}{f\left(a^2+b^2\right)^2(bc-ad)^2(a+b\tan(e+fx))}$$
$$+\frac{(a^6Cd^2-3a^5bBd^2+3a^4b^2d(2Ad+Bc-Cd)-a^3b^3(8cd(A-C)+B(c^2-d^2))-3a^2b^4(c(2Bd+cC)-}{f\left(a^2+b^2\right)^3(bc-}$$
$$-\frac{d^2(Ad^2-Bcd+c^2C)\log(c\cos(e+fx)+d\sin(e+fx))}{f\left(c^2+d^2\right)(bc-ad)^3}$$

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^3*(c + d*
Tan[e + f*x])),x]
```

```
[Out] ((a^3*(A*c - c*C + B*d) - 3*a*b^2*(A*c - c*C + B*d) + 3*a^2*b*(B*c - (A - C
)*d) - b^3*(B*c - (A - C)*d))*x)/((a^2 + b^2)^3*(c^2 + d^2)) + ((3*a*b^5*B*
c^2 - 3*a^5*b*B*d^2 + a^6*C*d^2 + 3*a^4*b^2*d*(B*c + 2*A*d - C*d) + b^6*(c*
(c*C - B*d) - A*(c^2 - d^2)) - a^3*b^3*(8*c*(A - C)*d + B*(c^2 - d^2)) - 3*
a^2*b^4*(c*(c*C + 2*B*d) - A*(c^2 + d^2)))*Log[a*Cos[e + f*x] + b*Sin[e + f
*x]])/((a^2 + b^2)^3*(b*c - a*d)^3*f) - (d^2*(c^2*C - B*c*d + A*d^2)*Log[c*
Cos[e + f*x] + d*Sin[e + f*x]])/((b*c - a*d)^3*(c^2 + d^2)*f) - (A*b^2 - a*
(b*B - a*C))/(2*(a^2 + b^2)*(b*c - a*d)*f*(a + b*Tan[e + f*x])^2) - (2*a*b^
3*c*(A - C) + 2*a^3*b*B*d - a^4*C*d + b^4*(B*c - A*d) - a^2*b^2*(B*c + 3*A*
d - C*d))/((a^2 + b^2)^2*(b*c - a*d)^2*f*(a + b*Tan[e + f*x]))
```

Rule 3611

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c/(b*f))*Log[RemoveContent[a*Cos[e + f*x] + b*Si
n[e + f*x], x]], x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3730

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 - a*(b*B - a*C))*(a + b*Tan[e +
```

```
 f*x])^(m + 1)*((c + d*Tan[e + f*x])^(n + 1)/(f*(m + 1)*(b*c - a*d)*(a^2 +
b^2))), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3732

```
Int[((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^
2)/(((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])), x_Symbol] :> Simp[(a*(A*c - c*C + B*d) + b*(B*c - A*d + C*d))*(x/
((a^2 + b^2)*(c^2 + d^2))), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/((b*c - a*d)
*(a^2 + b^2)), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] - Dist
[(c^2*C - B*c*d + A*d^2)/((b*c - a*d)*(c^2 + d^2)), Int[(d - c*Tan[e + f*x]
)/(c + d*Tan[e + f*x]), x], x]) /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] &&
 NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0]
```

Rubi steps

integral

$$= -\frac{Ab^2 - a(bB - aC)}{2(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))^2} - \frac{\int \frac{-2\left(abc(A-C)-a^2Ad+b^2(Bc-Ad)\right)+2(Ab-aB-bC)(bc-ad)\tan(e+fx)+2\left(Ab^2-a(bB-aC)\right)d\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))}\,dx}{2(a^2+b^2)(bc-ad)}$$

$$= -\frac{Ab^2 - a(bB - aC)}{2(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))^2} - \frac{2ab^3c(A-C)+2a^3bBd-a^4Cd+b^4(Bc-Ad)-a^2b^2(Bc+3Ad-Cd)}{(a^2+b^2)^2(bc-ad)^2f(a+b\tan(e+fx))} + \frac{\int \frac{2\left(2ab^3Bc^2-2a^3bc(A-C)d+a^4Ad^2+b^4\left(c(cC-Bd)-A(c^2-d^2)\right)-a^2b^2\left(c(cC+3Bd)-A(c^2+2d^2)\right)\right)+2\left(a^2B-b^2B-2ab(A-C)\right)(bc-ad)^2\,\mathrm{ta}}{(a+b\tan(e+fx))(c+d\tan(e+fx))}}{2(a^2+b^2)^2(bc-ad)^2}$$

$$= \frac{(a^3(Ac - cC + Bd) - 3ab^2(Ac - cC + Bd) + 3a^2b(Bc - (A - C)d) - b^3(Bc - (A - C)d))\,x}{(a^2+b^2)^3\,(c^2+d^2)} - \frac{Ab^2 - a(bB - aC)}{2\,(a^2+b^2)\,(bc-ad)f(a+b\tan(e+fx))^2} - \frac{2ab^3c(A-C) + 2a^3bBd - a^4Cd + b^4(Bc - Ad) - a^2b^2(Bc + 3Ad - Cd)}{(a^2+b^2)^2\,(bc-ad)^2f(a+b\tan(e+fx))} - \frac{(d^2(c^2C - Bcd + Ad^2))\int \frac{d - c\tan(e+fx)}{c + d\tan(e+fx)}\,dx}{(bc-ad)^3\,(c^2+d^2)} + \frac{(3ab^5Bc^2 - 3a^5bBd^2 + a^6Cd^2 + 3a^4b^2d(Bc + 2Ad - Cd) + b^6(c(cC - Bd) - A(c^2 - d^2)) - a^3b^3(8c(A -}{(a^2+b^2)^3\,(bc-ad)^3}$$

$$= \frac{(a^3(Ac - cC + Bd) - 3ab^2(Ac - cC + Bd) + 3a^2b(Bc - (A - C)d) - b^3(Bc - (A - C)d))\,x}{(a^2+b^2)^3\,(c^2+d^2)} + \frac{(3ab^5Bc^2 - 3a^5bBd^2 + a^6Cd^2 + 3a^4b^2d(Bc + 2Ad - Cd) + b^6(c(cC - Bd) - A(c^2 - d^2)) - a^3b^3(8c(A -}{(a^2+b^2)^3\,(bc - a} - \frac{d^2(c^2C - Bcd + Ad^2)\log(c\cos(e+fx) + d\sin(e+fx))}{(bc-ad)^3\,(c^2+d^2)\,f} - \frac{Ab^2 - a(bB - aC)}{2\,(a^2+b^2)\,(bc-ad)f(a+b\tan(e+fx))^2} - \frac{2ab^3c(A-C) + 2a^3bBd - a^4Cd + b^4(Bc - Ad) - a^2b^2(Bc + 3Ad - Cd)}{(a^2+b^2)^2\,(bc-ad)^2f(a+b\tan(e+fx))}$$

Mathematica [A] (verified)

Time = 8.92 (sec) , antiderivative size = 898, normalized size of antiderivative = 1.88

$$\int \frac{A + B\tan(e+fx) + C\tan^2(e+fx)}{(a+b\tan(e+fx))^3(c+d\tan(e+fx))}\,dx = -\frac{Ab^2 - a(bB - aC)}{2\,(a^2+b^2)\,(bc-ad)f(a+b\tan(e+fx))^2} - \frac{-\frac{-\frac{b(bc-ad)^2\left(3a^2Abc - Ab^3c - a^3Bc + 3ab^2Bc - 3a^2bcC + b^3cC + a^3Ad - 3aAb^2d + 3a^2bBd - b^3Bd - a^3Cd + 3ab^2Cd + \frac{\sqrt{-b^2}\left(a^3(Ac - cC + Bd) - 3ab^2(Ac - cC + Bd\right.}{b}\right.}{(a^2+b^2)(c^2+d^2)}}{}}{}$$

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^3*(
c + d*Tan[e + f*x])),x]
```

```
[Out] -1/2*(A*b^2 - a*(b*B - a*C))/((a^2 + b^2)*(b*c - a*d)*f*(a + b*Tan[e + f*x]
)^2) - (-((-((b*(b*c - a*d)^2*(3*a^2*A*b*c - A*b^3*c - a^3*B*c + 3*a*b^2*B*
c - 3*a^2*b*c*C + b^3*c*C + a^3*A*d - 3*a*A*b^2*d + 3*a^2*b*B*d - b^3*B*d -
 a^3*C*d + 3*a*b^2*C*d + (Sqrt[-b^2]*(a^3*(A*c - c*C + B*d) - 3*a*b^2*(A*c
- c*C + B*d) + 3*a^2*b*(B*c - (A - C)*d) - b^3*(B*c - (A - C)*d)))/b)*Log[S
```

```
qrt[-b^2] - b*Tan[e + f*x]])/((a^2 + b^2)*(c^2 + d^2))) + (2*b*(3*a*b^5*B*c
^2 - 3*a^5*b*B*d^2 + a^6*C*d^2 + 3*a^4*b^2*d*(B*c + 2*A*d - C*d) + b^6*(c*(
c*C - B*d) - A*(c^2 - d^2)) - a^3*b^3*(8*c*(A - C)*d + B*(c^2 - d^2)) - 3*a
^2*b^4*(c*(c*C + 2*B*d) - A*(c^2 + d^2)))*Log[a + b*Tan[e + f*x]])/((a^2 +
b^2)*(b*c - a*d)) - (b*(b*c - a*d)^2*(3*a^2*A*b*c - A*b^3*c - a^3*B*c + 3*a
*b^2*B*c - 3*a^2*b*c*C + b^3*c*C + a^3*A*d - 3*a*A*b^2*d + 3*a^2*b*B*d - b^
3*B*d - a^3*C*d + 3*a*b^2*C*d - (Sqrt[-b^2]*(a^3*(A*c - c*C + B*d) - 3*a*b^
2*(A*c - c*C + B*d) + 3*a^2*b*(B*c - (A - C)*d) - b^3*(B*c - (A - C)*d)))/b
)*Log[Sqrt[-b^2] + b*Tan[e + f*x]])/((a^2 + b^2)*(c^2 + d^2)) - (2*b*(a^2 +
 b^2)^2*d^2*(c^2*C - B*c*d + A*d^2)*Log[c + d*Tan[e + f*x]])/((b*c - a*d)*(
c^2 + d^2)))/(b*(a^2 + b^2)*(b*c - a*d)*f)) - (-(a*(-2*a*(A*b^2 - a*(b*B -
a*C))*d + 2*b*(A*b - a*B - b*C)*(b*c - a*d))) - 2*b^2*(a*b*c*(A - C) - a^2*
A*d + b^2*(B*c - A*d)))/((a^2 + b^2)*(b*c - a*d)*f*(a + b*Tan[e + f*x])))/(
2*(a^2 + b^2)*(b*c - a*d))
```

Maple [A] (verified)

Time = 1.58 (sec) , antiderivative size = 647, normalized size of antiderivative = 1.36

method	result
derivativedivides	$\frac{\frac{3A\,a^2b^2d-2Aa\,b^3c+A\,b^4d-2a^3bBd+B\,a^2b^2c-B\,b^4c+a^4Cd-C\,a^2b^2d+2Ca\,b^3c}{(ad-bc)^2\left(a^2+b^2\right)^2(a+b\tan(fx+e))}-\frac{\left(6A\,a^4b^2d^2-8A\,a^3b^3cd+3A\,a^2b^4c^2+3A\,a^2b^4d^2-\right.}{}}{}$
default	$\frac{\frac{3A\,a^2b^2d-2Aa\,b^3c+A\,b^4d-2a^3bBd+B\,a^2b^2c-B\,b^4c+a^4Cd-C\,a^2b^2d+2Ca\,b^3c}{(ad-bc)^2\left(a^2+b^2\right)^2(a+b\tan(fx+e))}-\frac{\left(6A\,a^4b^2d^2-8A\,a^3b^3cd+3A\,a^2b^4c^2+3A\,a^2b^4d^2-\right.}{}}{}$
norman	Expression too large to display
parallelrisch	Expression too large to display
risch	Expression too large to display

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^3/(c+d*tan(f*x+e)),x,m
ethod=_RETURNVERBOSE)
```

```
[Out] 1/f*((3*A*a^2*b^2*d-2*A*a*b^3*c+A*b^4*d-2*B*a^3*b*d+B*a^2*b^2*c-B*b^4*c+C*a
^4*d-C*a^2*b^2*d+2*C*a*b^3*c)/(a*d-b*c)^2/(a^2+b^2)^2/(a+b*tan(f*x+e))-(6*A
*a^4*b^2*d^2-8*A*a^3*b^3*c*d+3*A*a^2*b^4*c^2+3*A*a^2*b^4*d^2-A*b^6*c^2+A*b^
6*d^2-3*B*a^5*b*d^2+3*B*a^4*b^2*c*d-B*a^3*b^3*c^2+B*a^3*b^3*d^2-6*B*a^2*b^4
*c*d+3*B*a*b^5*c^2-B*b^6*c*d+C*a^6*d^2-3*C*a^4*b^2*d^2+8*C*a^3*b^3*c*d-3*C*
a^2*b^4*c^2+C*b^6*c^2)/(a*d-b*c)^3/(a^2+b^2)^3*ln(a+b*tan(f*x+e))+1/2*(A*b^
2-B*a*b+C*a^2)/(a*d-b*c)/(a^2+b^2)/(a+b*tan(f*x+e))^2+1/(a^2+b^2)^3/(c^2+d^
2)*(1/2*(-A*a^3*d-3*A*a^2*b*c+3*A*a*b^2*d+A*b^3*c+B*a^3*c-3*B*a^2*b*d-3*B*a
*b^2*c+B*b^3*d+C*a^3*d+3*C*a^2*b*c-3*C*a*b^2*d-C*b^3*c)*ln(1+tan(f*x+e)^2)+
(A*a^3*c-3*A*a^2*b*d-3*A*a*b^2*c+A*b^3*d+B*a^3*d+3*B*a^2*b*c-3*B*a*b^2*d-B*
b^3*c-C*a^3*c+3*C*a^2*b*d+3*C*a*b^2*c-C*b^3*d)*arctan(tan(f*x+e)))+(A*d^2-B
*c*d+C*c^2)*d^2/(a*d-b*c)^3/(c^2+d^2)*ln(c+d*tan(f*x+e)))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 3643 vs. $2(475) = 950$.

Time = 3.39 (sec) , antiderivative size = 3643, normalized size of antiderivative = 7.64

$$\int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{(a + b\tan(e + fx))^3(c + d\tan(e + fx))}\,dx = \text{Too large to display}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^3/(c+d*tan(f*x+e
)),x, algorithm="fricas")
```

```
[Out] -1/2*((3*C*a^4*b^4 - 5*B*a^3*b^5 + (7*A - 3*C)*a^2*b^6 + B*a*b^7 + A*b^8)*c
^4 - 4*(2*C*a^5*b^3 - 3*B*a^4*b^4 + (4*A - C)*a^3*b^5 + A*a*b^7)*c^3*d + (5
*C*a^6*b^2 - 7*B*a^5*b^3 + (9*A + 2*C)*a^4*b^4 - 6*B*a^3*b^5 + (10*A - 3*C)
*a^2*b^6 + B*a*b^7 + A*b^8)*c^2*d^2 - 4*(2*C*a^5*b^3 - 3*B*a^4*b^4 + (4*A -
 C)*a^3*b^5 + A*a*b^7)*c*d^3 + (5*C*a^6*b^2 - 7*B*a^5*b^3 + (9*A - C)*a^4*b
^4 - B*a^3*b^5 + 3*A*a^2*b^6)*d^4 - 2*(((A - C)*a^5*b^3 + 3*B*a^4*b^4 - 3*(
A - C)*a^3*b^5 - B*a^2*b^6)*c^4 - (3*(A - C)*a^6*b^2 + 8*B*a^5*b^3 - 6*(A -
 C)*a^4*b^4 - (A - C)*a^2*b^6)*c^3*d + 3*((A - C)*a^7*b + 2*B*a^6*b^2 + 2*B
*a^4*b^4 - (A - C)*a^3*b^5)*c^2*d^2 - ((A - C)*a^8 + 6*(A - C)*a^6*b^2 + 8*
B*a^5*b^3 - 3*(A - C)*a^4*b^4)*c*d^3 - (B*a^8 - 3*(A - C)*a^7*b - 3*B*a^6*b
^2 + (A - C)*a^5*b^3)*d^4)*f*x - ((C*a^4*b^4 - 3*B*a^3*b^5 + 5*(A - C)*a^2*
b^6 + 3*B*a*b^7 - A*b^8)*c^4 - 4*(C*a^5*b^3 - 2*B*a^4*b^4 + (3*A - 2*C)*a^3
*b^5 + B*a^2*b^6)*c^3*d + (3*C*a^6*b^2 - 5*B*a^5*b^3 + (7*A - 2*C)*a^4*b^4
- 2*B*a^3*b^5 + (6*A - 5*C)*a^2*b^6 + 3*B*a*b^7 - A*b^8)*c^2*d^2 - 4*(C*a^5
*b^3 - 2*B*a^4*b^4 + (3*A - 2*C)*a^3*b^5 + B*a^2*b^6)*c*d^3 + (3*C*a^6*b^2
- 5*B*a^5*b^3 + (7*A - 3*C)*a^4*b^4 + B*a^3*b^5 + A*a^2*b^6)*d^4 + 2*(((A -
 C)*a^3*b^5 + 3*B*a^2*b^6 - 3*(A - C)*a*b^7 - B*b^8)*c^4 - (3*(A - C)*a^4*b
^4 + 8*B*a^3*b^5 - 6*(A - C)*a^2*b^6 - (A - C)*b^8)*c^3*d + 3*((A - C)*a^5*
b^3 + 2*B*a^4*b^4 + 2*B*a^2*b^6 - (A - C)*a*b^7)*c^2*d^2 - ((A - C)*a^6*b^2
 + 6*(A - C)*a^4*b^4 + 8*B*a^3*b^5 - 3*(A - C)*a^2*b^6)*c*d^3 - (B*a^6*b^2
- 3*(A - C)*a^5*b^3 - 3*B*a^4*b^4 + (A - C)*a^3*b^5)*d^4)*f*x)*tan(f*x + e)
^2 + ((B*a^5*b^3 - 3*(A - C)*a^4*b^4 - 3*B*a^3*b^5 + (A - C)*a^2*b^6)*c^4 -
 (3*B*a^6*b^2 - 8*(A - C)*a^5*b^3 - 6*B*a^4*b^4 - B*a^2*b^6)*c^3*d - (C*a^8
 - 3*B*a^7*b + 3*(2*A - C)*a^6*b^2 + 3*(2*A - C)*a^4*b^4 + 3*B*a^3*b^5 + C*
a^2*b^6)*c^2*d^2 - (3*B*a^6*b^2 - 8*(A - C)*a^5*b^3 - 6*B*a^4*b^4 - B*a^2*b
^6)*c*d^3 - (C*a^8 - 3*B*a^7*b + 3*(2*A - C)*a^6*b^2 + B*a^5*b^3 + 3*A*a^4*
b^4 + A*a^2*b^6)*d^4 + ((B*a^3*b^5 - 3*(A - C)*a^2*b^6 - 3*B*a*b^7 + (A - C
)*b^8)*c^4 - (3*B*a^4*b^4 - 8*(A - C)*a^3*b^5 - 6*B*a^2*b^6 - B*b^8)*c^3*d
- (C*a^6*b^2 - 3*B*a^5*b^3 + 3*(2*A - C)*a^4*b^4 + 3*(2*A - C)*a^2*b^6 + 3*
B*a*b^7 + C*b^8)*c^2*d^2 - (3*B*a^4*b^4 - 8*(A - C)*a^3*b^5 - 6*B*a^2*b^6 -
 B*b^8)*c*d^3 - (C*a^6*b^2 - 3*B*a^5*b^3 + 3*(2*A - C)*a^4*b^4 + B*a^3*b^5
+ 3*A*a^2*b^6 + A*b^8)*d^4)*tan(f*x + e)^2 + 2*((B*a^4*b^4 - 3*(A - C)*a^3*
b^5 - 3*B*a^2*b^6 + (A - C)*a*b^7)*c^4 - (3*B*a^5*b^3 - 8*(A - C)*a^4*b^4 -
 6*B*a^3*b^5 - B*a*b^7)*c^3*d - (C*a^7*b - 3*B*a^6*b^2 + 3*(2*A - C)*a^5*b^
```

3 + 3*(2*A - C)*a^3*b^5 + 3*B*a^2*b^6 + C*a*b^7)*c^2*d^2 - (3*B*a^5*b^3 - 8
*(A - C)*a^4*b^4 - 6*B*a^3*b^5 - B*a*b^7)*c*d^3 - (C*a^7*b - 3*B*a^6*b^2 +
3*(2*A - C)*a^5*b^3 + B*a^4*b^4 + 3*A*a^3*b^5 + A*a*b^7)*d^4)*tan(f*x + e))
*log((b^2*tan(f*x + e)^2 + 2*a*b*tan(f*x + e) + a^2)/(tan(f*x + e)^2 + 1))
+ ((C*a^8 + 3*C*a^6*b^2 + 3*C*a^4*b^4 + C*a^2*b^6)*c^2*d^2 - (B*a^8 + 3*B*a
^6*b^2 + 3*B*a^4*b^4 + B*a^2*b^6)*c*d^3 + (A*a^8 + 3*A*a^6*b^2 + 3*A*a^4*b^
4 + A*a^2*b^6)*d^4 + ((C*a^6*b^2 + 3*C*a^4*b^4 + 3*C*a^2*b^6 + C*b^8)*c^2*d
^2 - (B*a^6*b^2 + 3*B*a^4*b^4 + 3*B*a^2*b^6 + B*b^8)*c*d^3 + (A*a^6*b^2 + 3
*A*a^4*b^4 + 3*A*a^2*b^6 + A*b^8)*d^4)*tan(f*x + e)^2 + 2*((C*a^7*b + 3*C*a
^5*b^3 + 3*C*a^3*b^5 + C*a*b^7)*c^2*d^2 - (B*a^7*b + 3*B*a^5*b^3 + 3*B*a^3*
b^5 + B*a*b^7)*c*d^3 + (A*a^7*b + 3*A*a^5*b^3 + 3*A*a^3*b^5 + A*a*b^7)*d^4)
*tan(f*x + e))*log((d^2*tan(f*x + e)^2 + 2*c*d*tan(f*x + e) + c^2)/(tan(f*x
 + e)^2 + 1)) - 2*((C*a^5*b^3 - 2*B*a^4*b^4 + 3*(A - C)*a^3*b^5 + 3*B*a^2*b
^6 - (3*A - 2*C)*a*b^7 - B*b^8)*c^4 - (3*C*a^6*b^2 - 5*B*a^5*b^3 + (7*A - 6
*C)*a^4*b^4 + 6*B*a^3*b^5 - 3*(2*A - C)*a^2*b^6 - B*a*b^7 - A*b^8)*c^3*d +
(2*C*a^7*b - 3*B*a^6*b^2 + 2*(2*A - C)*a^5*b^3 + B*a^4*b^4 - 2*C*a^3*b^5 +
3*B*a^2*b^6 - 2*(2*A - C)*a*b^7 - B*b^8)*c^2*d^2 - (3*C*a^6*b^2 - 5*B*a^5*b
^3 + (7*A - 6*C)*a^4*b^4 + 6*B*a^3*b^5 - 3*(2*A - C)*a^2*b^6 - B*a*b^7 - A*
b^8)*c*d^3 + (2*C*a^7*b - 3*B*a^6*b^2 + (4*A - 3*C)*a^5*b^3 + 3*B*a^4*b^4 -
 (3*A - C)*a^3*b^5 - A*a*b^7)*d^4 + 2*(((A - C)*a^4*b^4 + 3*B*a^3*b^5 - 3*(
A - C)*a^2*b^6 - B*a*b^7)*c^4 - (3*(A - C)*a^5*b^3 + 8*B*a^4*b^4 - 6*(A - C
)*a^3*b^5 - (A - C)*a*b^7)*c^3*d + 3*((A - C)*a^6*b^2 + 2*B*a^5*b^3 + 2*B*a
^3*b^5 - (A - C)*a^2*b^6)*c^2*d^2 - ((A - C)*a^7*b + 6*(A - C)*a^5*b^3 + 8*
B*a^4*b^4 - 3*(A - C)*a^3*b^5)*c*d^3 - (B*a^7*b - 3*(A - C)*a^6*b^2 - 3*B*a
^5*b^3 + (A - C)*a^4*b^4)*d^4)*f*x)*tan(f*x + e))/(((a^6*b^5 + 3*a^4*b^7 +
3*a^2*b^9 + b^11)*c^5 - 3*(a^7*b^4 + 3*a^5*b^6 + 3*a^3*b^8 + a*b^10)*c^4*d
+ (3*a^8*b^3 + 10*a^6*b^5 + 12*a^4*b^7 + 6*a^2*b^9 + b^11)*c^3*d^2 - (a^9*b
^2 + 6*a^7*b^4 + 12*a^5*b^6 + 10*a^3*b^8 + 3*a*b^10)*c^2*d^3 + 3*(a^8*b^3 +
 3*a^6*b^5 + 3*a^4*b^7 + a^2*b^9)*c*d^4 - (a^9*b^2 + 3*a^7*b^4 + 3*a^5*b^6
+ a^3*b^8)*d^5)*f*tan(f*x + e)^2 + 2*((a^7*b^4 + 3*a^5*b^6 + 3*a^3*b^8 + a*
b^10)*c^5 - 3*(a^8*b^3 + 3*a^6*b^5 + 3*a^4*b^7 + a^2*b^9)*c^4*d + (3*a^9*b^
2 + 10*a^7*b^4 + 12*a^5*b^6 + 6*a^3*b^8 + a*b^10)*c^3*d^2 - (a^10*b + 6*a^8
*b^3 + 12*a^6*b^5 + 10*a^4*b^7 + 3*a^2*b^9)*c^2*d^3 + 3*(a^9*b^2 + 3*a^7*b^
4 + 3*a^5*b^6 + a^3*b^8)*c*d^4 - (a^10*b + 3*a^8*b^3 + 3*a^6*b^5 + a^4*b^7)
*d^5)*f*tan(f*x + e) + ((a^8*b^3 + 3*a^6*b^5 + 3*a^4*b^7 + a^2*b^9)*c^5 - 3
*(a^9*b^2 + 3*a^7*b^4 + 3*a^5*b^6 + a^3*b^8)*c^4*d + (3*a^10*b + 10*a^8*b^3
 + 12*a^6*b^5 + 6*a^4*b^7 + a^2*b^9)*c^3*d^2 - (a^11 + 6*a^9*b^2 + 12*a^7*b
^4 + 10*a^5*b^6 + 3*a^3*b^8)*c^2*d^3 + 3*(a^10*b + 3*a^8*b^3 + 3*a^6*b^5 +
a^4*b^7)*c*d^4 - (a^11 + 3*a^9*b^2 + 3*a^7*b^4 + a^5*b^6)*d^5)*f)

Sympy [F(-2)]

Exception generated.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^3(c+d\tan(e+fx))}\,dx = \text{Exception raised: NotImplementedError}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e))**3/(c+d*tan(f*x
+e)),x)
```

```
[Out] Exception raised: NotImplementedError >> no valid subset found
```

Maxima [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1096 vs. $2(475) = 950$.

Time = 0.38 (sec) , antiderivative size = 1096, normalized size of antiderivative = 2.30

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^3(c+d\tan(e+fx))}\,dx = \text{Too large to display}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^3/(c+d*tan(f*x+e
)),x, algorithm="maxima")
```

```
[Out] 1/2*(2*(((A - C)*a^3 + 3*B*a^2*b - 3*(A - C)*a*b^2 - B*b^3)*c + (B*a^3 - 3*
(A - C)*a^2*b - 3*B*a*b^2 + (A - C)*b^3)*d)*(f*x + e)/((a^6 + 3*a^4*b^2 + 3
*a^2*b^4 + b^6)*c^2 + (a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6)*d^2) - 2*((B*a^3*
b^3 - 3*(A - C)*a^2*b^4 - 3*B*a*b^5 + (A - C)*b^6)*c^2 - (3*B*a^4*b^2 - 8*(
A - C)*a^3*b^3 - 6*B*a^2*b^4 - B*b^6)*c*d - (C*a^6 - 3*B*a^5*b + 3*(2*A - C
)*a^4*b^2 + B*a^3*b^3 + 3*A*a^2*b^4 + A*b^6)*d^2)*log(b*tan(f*x + e) + a)/(
(a^6*b^3 + 3*a^4*b^5 + 3*a^2*b^7 + b^9)*c^3 - 3*(a^7*b^2 + 3*a^5*b^4 + 3*a^
3*b^6 + a*b^8)*c^2*d + 3*(a^8*b + 3*a^6*b^3 + 3*a^4*b^5 + a^2*b^7)*c*d^2 -
(a^9 + 3*a^7*b^2 + 3*a^5*b^4 + a^3*b^6)*d^3) - 2*(C*c^2*d^2 - B*c*d^3 + A*d
^4)*log(d*tan(f*x + e) + c)/(b^3*c^5 - 3*a*b^2*c^4*d + 3*a^2*b*c*d^4 - a^3*
d^5 + (3*a^2*b + b^3)*c^3*d^2 - (a^3 + 3*a*b^2)*c^2*d^3) + ((B*a^3 - 3*(A -
 C)*a^2*b - 3*B*a*b^2 + (A - C)*b^3)*c - ((A - C)*a^3 + 3*B*a^2*b - 3*(A -
C)*a*b^2 - B*b^3)*d)*log(tan(f*x + e)^2 + 1)/((a^6 + 3*a^4*b^2 + 3*a^2*b^4
+ b^6)*c^2 + (a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6)*d^2) - ((C*a^4*b - 3*B*a^3
*b^2 + (5*A - 3*C)*a^2*b^3 + B*a*b^4 + A*b^5)*c - (3*C*a^5 - 5*B*a^4*b + (7
*A - C)*a^3*b^2 - B*a^2*b^3 + 3*A*a*b^4)*d - 2*((B*a^2*b^3 - 2*(A - C)*a*b^
4 - B*b^5)*c + (C*a^4*b - 2*B*a^3*b^2 + (3*A - C)*a^2*b^3 + A*b^5)*d)*tan(f
*x + e))/((a^6*b^2 + 2*a^4*b^4 + a^2*b^6)*c^2 - 2*(a^7*b + 2*a^5*b^3 + a^3*
b^5)*c*d + (a^8 + 2*a^6*b^2 + a^4*b^4)*d^2 + ((a^4*b^4 + 2*a^2*b^6 + b^8)*c
^2 - 2*(a^5*b^3 + 2*a^3*b^5 + a*b^7)*c*d + (a^6*b^2 + 2*a^4*b^4 + a^2*b^6)*
d^2)*tan(f*x + e)^2 + 2*((a^5*b^3 + 2*a^3*b^5 + a*b^7)*c^2 - 2*(a^6*b^2 + 2
*a^4*b^4 + a^2*b^6)*c*d + (a^7*b + 2*a^5*b^3 + a^3*b^5)*d^2)*tan(f*x + e)))
/f
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 2080 vs. $2(475) = 950$.

Time = 1.09 (sec) , antiderivative size = 2080, normalized size of antiderivative = 4.36

$$\int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{(a + b\tan(e + fx))^3(c + d\tan(e + fx))}\,dx = \text{Too large to display}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^3/(c+d*tan(f*x+e
)),x, algorithm="giac")
```

```
[Out] 1/2*(2*(A*a^3*c - C*a^3*c + 3*B*a^2*b*c - 3*A*a*b^2*c + 3*C*a*b^2*c - B*b^3
*c + B*a^3*d - 3*A*a^2*b*d + 3*C*a^2*b*d - 3*B*a*b^2*d + A*b^3*d - C*b^3*d)
*(f*x + e)/(a^6*c^2 + 3*a^4*b^2*c^2 + 3*a^2*b^4*c^2 + b^6*c^2 + a^6*d^2 + 3
*a^4*b^2*d^2 + 3*a^2*b^4*d^2 + b^6*d^2) + (B*a^3*c - 3*A*a^2*b*c + 3*C*a^2*
b*c - 3*B*a*b^2*c + A*b^3*c - C*b^3*c - A*a^3*d + C*a^3*d - 3*B*a^2*b*d + 3
*A*a*b^2*d - 3*C*a*b^2*d + B*b^3*d)*log(tan(f*x + e)^2 + 1)/(a^6*c^2 + 3*a^
4*b^2*c^2 + 3*a^2*b^4*c^2 + b^6*c^2 + a^6*d^2 + 3*a^4*b^2*d^2 + 3*a^2*b^4*d
^2 + b^6*d^2) - 2*(B*a^3*b^4*c^2 - 3*A*a^2*b^5*c^2 + 3*C*a^2*b^5*c^2 - 3*B*
a*b^6*c^2 + A*b^7*c^2 - C*b^7*c^2 - 3*B*a^4*b^3*c*d + 8*A*a^3*b^4*c*d - 8*C
*a^3*b^4*c*d + 6*B*a^2*b^5*c*d + B*b^7*c*d - C*a^6*b*d^2 + 3*B*a^5*b^2*d^2
- 6*A*a^4*b^3*d^2 + 3*C*a^4*b^3*d^2 - B*a^3*b^4*d^2 - 3*A*a^2*b^5*d^2 - A*b
^7*d^2)*log(abs(b*tan(f*x + e) + a))/(a^6*b^4*c^3 + 3*a^4*b^6*c^3 + 3*a^2*b
^8*c^3 + b^10*c^3 - 3*a^7*b^3*c^2*d - 9*a^5*b^5*c^2*d - 9*a^3*b^7*c^2*d - 3
*a*b^9*c^2*d + 3*a^8*b^2*c*d^2 + 9*a^6*b^4*c*d^2 + 9*a^4*b^6*c*d^2 + 3*a^2*
b^8*c*d^2 - a^9*b*d^3 - 3*a^7*b^3*d^3 - 3*a^5*b^5*d^3 - a^3*b^7*d^3) - 2*(C
*c^2*d^3 - B*c*d^4 + A*d^5)*log(abs(d*tan(f*x + e) + c))/(b^3*c^5*d - 3*a*b
^2*c^4*d^2 + 3*a^2*b*c^3*d^3 + b^3*c^3*d^3 - a^3*c^2*d^4 - 3*a*b^2*c^2*d^4
+ 3*a^2*b*c*d^5 - a^3*d^6) + (3*B*a^3*b^5*c^2*tan(f*x + e)^2 - 9*A*a^2*b^6*
c^2*tan(f*x + e)^2 + 9*C*a^2*b^6*c^2*tan(f*x + e)^2 - 9*B*a*b^7*c^2*tan(f*x
 + e)^2 + 3*A*b^8*c^2*tan(f*x + e)^2 - 3*C*b^8*c^2*tan(f*x + e)^2 - 9*B*a^4
*b^4*c*d*tan(f*x + e)^2 + 24*A*a^3*b^5*c*d*tan(f*x + e)^2 - 24*C*a^3*b^5*c*
d*tan(f*x + e)^2 + 18*B*a^2*b^6*c*d*tan(f*x + e)^2 + 3*B*b^8*c*d*tan(f*x +
e)^2 - 3*C*a^6*b^2*d^2*tan(f*x + e)^2 + 9*B*a^5*b^3*d^2*tan(f*x + e)^2 - 18
*A*a^4*b^4*d^2*tan(f*x + e)^2 + 9*C*a^4*b^4*d^2*tan(f*x + e)^2 - 3*B*a^3*b^
5*d^2*tan(f*x + e)^2 - 9*A*a^2*b^6*d^2*tan(f*x + e)^2 - 3*A*b^8*d^2*tan(f*x
 + e)^2 + 8*B*a^4*b^4*c^2*tan(f*x + e) - 22*A*a^3*b^5*c^2*tan(f*x + e) + 22
*C*a^3*b^5*c^2*tan(f*x + e) - 18*B*a^2*b^6*c^2*tan(f*x + e) + 2*A*a*b^7*c^2
*tan(f*x + e) - 2*C*a*b^7*c^2*tan(f*x + e) - 2*B*b^8*c^2*tan(f*x + e) + 2*C
*a^6*b^2*c*d*tan(f*x + e) - 24*B*a^5*b^3*c*d*tan(f*x + e) + 58*A*a^4*b^4*c*
d*tan(f*x + e) - 52*C*a^4*b^4*c*d*tan(f*x + e) + 32*B*a^3*b^5*c*d*tan(f*x +
 e) + 12*A*a^2*b^6*c*d*tan(f*x + e) - 6*C*a^2*b^6*c*d*tan(f*x + e) + 8*B*a*
b^7*c*d*tan(f*x + e) + 2*A*b^8*c*d*tan(f*x + e) - 8*C*a^7*b*d^2*tan(f*x + e
) + 22*B*a^6*b^2*d^2*tan(f*x + e) - 42*A*a^5*b^3*d^2*tan(f*x + e) + 18*C*a^
5*b^3*d^2*tan(f*x + e) - 2*B*a^4*b^4*d^2*tan(f*x + e) - 26*A*a^3*b^5*d^2*ta
```

```
n(f*x + e) + 2*C*a^3*b^5*d^2*tan(f*x + e) - 8*A*a*b^7*d^2*tan(f*x + e) - C*
a^6*b^2*c^2 + 6*B*a^5*b^3*c^2 - 14*A*a^4*b^4*c^2 + 11*C*a^4*b^4*c^2 - 7*B*a
^3*b^5*c^2 - 3*A*a^2*b^6*c^2 - B*a*b^7*c^2 - A*b^8*c^2 + 4*C*a^7*b*c*d - 17
*B*a^6*b^2*c*d + 36*A*a^5*b^3*c*d - 24*C*a^5*b^3*c*d + 10*B*a^4*b^4*c*d + 1
6*A*a^3*b^5*c*d - 4*C*a^3*b^5*c*d + 3*B*a^2*b^6*c*d + 4*A*a*b^7*c*d - 6*C*a
^8*d^2 + 14*B*a^7*b*d^2 - 25*A*a^6*b^2*d^2 + 7*C*a^6*b^2*d^2 + 3*B*a^5*b^3*
d^2 - 19*A*a^4*b^4*d^2 + C*a^4*b^4*d^2 + B*a^3*b^5*d^2 - 6*A*a^2*b^6*d^2)/(
(a^6*b^3*c^3 + 3*a^4*b^5*c^3 + 3*a^2*b^7*c^3 + b^9*c^3 - 3*a^7*b^2*c^2*d -
9*a^5*b^4*c^2*d - 9*a^3*b^6*c^2*d - 3*a*b^8*c^2*d + 3*a^8*b*c*d^2 + 9*a^6*b
^3*c*d^2 + 9*a^4*b^5*c*d^2 + 3*a^2*b^7*c*d^2 - a^9*d^3 - 3*a^7*b^2*d^3 - 3*
a^5*b^4*d^3 - a^3*b^6*d^3)*(b*tan(f*x + e) + a)^2))/f
```

Mupad [B] (verification not implemented)

Time = 22.52 (sec) , antiderivative size = 65819, normalized size of antiderivative = 137.99

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^3(c+d\tan(e+fx))}\,dx = \text{Too large to display}$$

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))^3*(c + d*
tan(e + f*x))),x)
```

```
[Out] -(((A*b^5*c - 3*C*a^5*d - 3*A*a*b^4*d + B*a*b^4*c + 5*B*a^4*b*d + C*a^4*b*c
 + 5*A*a^2*b^3*c - 7*A*a^3*b^2*d - 3*B*a^3*b^2*c + B*a^2*b^3*d - 3*C*a^2*b^
3*c + C*a^3*b^2*d)/(2*(a^2*d^2 + b^2*c^2 - 2*a*b*c*d)*(a^4 + b^4 + 2*a^2*b^
2)) - (tan(e + f*x)*(A*b^5*d - B*b^5*c - 2*A*a*b^4*c + 2*C*a*b^4*c + C*a^4*
b*d + 3*A*a^2*b^3*d + B*a^2*b^3*c - 2*B*a^3*b^2*d - C*a^2*b^3*d))/((a^2*d^2
 + b^2*c^2 - 2*a*b*c*d)*(a^4 + b^4 + 2*a^2*b^2)))/(a^2 + b^2*tan(e + f*x)^2
 + 2*a*b*tan(e + f*x)) - symsum(log(- (A^3*b^8*c^2*d^4 - 4*A^3*a^2*b^6*d^6
- 7*A^3*a^4*b^4*d^6 - A^3*b^8*d^6 + A^2*C*b^8*d^6 - 3*A^3*a^2*b^6*c^2*d^4 -
 B^3*a^3*b^5*c^2*d^4 - C^3*a^2*b^6*c^2*d^4 - 2*C^3*a^3*b^5*c^3*d^3 + 7*C^3*
a^4*b^4*c^2*d^4 + A^2*B*a*b^7*d^6 + A^2*B*b^8*c*d^5 + A^3*a*b^7*c*d^5 + C^3
*a^7*b*c*d^5 - A*B^2*a^2*b^6*d^6 - 3*A*B^2*a^6*b^2*d^6 + 2*A^2*B*a^3*b^5*d^
6 + 9*A^2*B*a^5*b^3*d^6 - A*C^2*a^2*b^6*d^6 - 4*A*C^2*a^4*b^4*d^6 + A*C^2*a
^6*b^2*d^6 + 5*A^2*C*a^2*b^6*d^6 + 11*A^2*C*a^4*b^4*d^6 - A^2*C*a^6*b^2*d^6
 + A*C^2*b^8*c^2*d^4 - 2*A^2*C*b^8*c^2*d^4 - B*C^2*b^8*c^3*d^3 + B^2*C*b^8*
c^2*d^4 + 9*A^3*a^3*b^5*c*d^5 - B^3*a*b^7*c^2*d^4 + B^3*a^2*b^6*c*d^5 + B^3
*a^4*b^4*c*d^5 + 2*C^3*a*b^7*c^3*d^3 - 3*C^3*a^5*b^3*c*d^5 + A*B*C*a^7*b*d^
6 - 2*A*B*C*b^8*c*d^5 + 3*A*B^2*a^2*b^6*c^2*d^4 - A*B^2*a^4*b^4*c^2*d^4 + 3
*A^2*B*a^3*b^5*c^2*d^4 - A*C^2*a^2*b^6*c^2*d^4 + 4*A*C^2*a^3*b^5*c^3*d^3 -
14*A*C^2*a^4*b^4*c^2*d^4 + 5*A^2*C*a^2*b^6*c^2*d^4 - 2*A^2*C*a^3*b^5*c^3*d^
3 + 7*A^2*C*a^4*b^4*c^2*d^4 + 6*B*C^2*a^2*b^6*c^3*d^3 - 15*B*C^2*a^3*b^5*c^
2*d^4 - B*C^2*a^4*b^4*c^3*d^3 + 3*B*C^2*a^5*b^3*c^2*d^4 + 5*B^2*C*a^2*b^6*c
^2*d^4 + 2*B^2*C*a^3*b^5*c^3*d^3 - 4*B^2*C*a^4*b^4*c^2*d^4 + A*B*C*a^3*b^5*
d^6 - 6*A*B*C*a^5*b^3*d^6 + A*B*C*b^8*c^3*d^3 + 2*A*C^2*a*b^7*c*d^5 - A*C^2
```

*a^7*b*c*d^5 - 3*A^2*C*a*b^7*c*d^5 - 5*A*B^2*a^3*b^5*c*d^5 + 3*A*B^2*a^5*b^
3*c*d^5 - 5*A^2*B*a*b^7*c^2*d^4 + 7*A^2*B*a^2*b^6*c*d^5 - 10*A^2*B*a^4*b^4*
c*d^5 - 4*A*C^2*a*b^7*c^3*d^3 + 12*A*C^2*a^3*b^5*c*d^5 + 9*A*C^2*a^5*b^3*c*
d^5 + 2*A^2*C*a*b^7*c^3*d^3 - 21*A^2*C*a^3*b^5*c*d^5 - 6*A^2*C*a^5*b^3*c*d^
5 - 2*B*C^2*a*b^7*c^2*d^4 + B*C^2*a^2*b^6*c*d^5 + 5*B*C^2*a^4*b^4*c*d^5 - 4
*B*C^2*a^6*b^2*c*d^5 - 2*B^2*C*a*b^7*c^3*d^3 - B^2*C*a^3*b^5*c*d^5 + 3*B^2*
C*a^5*b^3*c*d^5 - 6*A*B*C*a^2*b^6*c^3*d^3 + 12*A*B*C*a^3*b^5*c^2*d^4 + A*B*
C*a^4*b^4*c^3*d^3 - 3*A*B*C*a^5*b^3*c^2*d^4 + 7*A*B*C*a*b^7*c^2*d^4 - 11*A*
B*C*a^2*b^6*c*d^5 + 2*A*B*C*a^4*b^4*c*d^5 + 3*A*B*C*a^6*b^2*c*d^5)/(a^12*d^
4 + b^12*c^4 + 4*a^2*b^10*c^4 + 6*a^4*b^8*c^4 + 4*a^6*b^6*c^4 + a^8*b^4*c^4
 + a^4*b^8*d^4 + 4*a^6*b^6*d^4 + 6*a^8*b^4*d^4 + 4*a^10*b^2*d^4 - 4*a^3*b^9
*c*d^3 - 16*a^3*b^9*c^3*d - 16*a^5*b^7*c*d^3 - 24*a^5*b^7*c^3*d - 24*a^7*b^
5*c*d^3 - 16*a^7*b^5*c^3*d - 16*a^9*b^3*c*d^3 - 4*a^9*b^3*c^3*d + 6*a^2*b^1
0*c^2*d^2 + 24*a^4*b^8*c^2*d^2 + 36*a^6*b^6*c^2*d^2 + 24*a^8*b^4*c^2*d^2 +
6*a^10*b^2*c^2*d^2 - 4*a*b^11*c^3*d - 4*a^11*b*c*d^3) - root(480*a^11*b^7*c
*d^9*f^4 + 480*a^7*b^11*c^9*d*f^4 + 360*a^13*b^5*c*d^9*f^4 + 360*a^9*b^9*c^
9*d*f^4 + 360*a^9*b^9*c*d^9*f^4 + 360*a^5*b^13*c^9*d*f^4 + 144*a^15*b^3*c*d
^9*f^4 + 144*a^11*b^7*c^9*d*f^4 + 144*a^7*b^11*c*d^9*f^4 + 144*a^3*b^15*c^9
*d*f^4 + 48*a^17*b*c^3*d^7*f^4 + 48*a*b^17*c^7*d^3*f^4 + 24*a^17*b*c^5*d^5*
f^4 + 24*a^13*b^5*c^9*d*f^4 + 24*a^5*b^13*c*d^9*f^4 + 24*a*b^17*c^5*d^5*f^4
 + 24*a^17*b*c*d^9*f^4 + 24*a*b^17*c^9*d*f^4 + 3920*a^9*b^9*c^5*d^5*f^4 - 3
360*a^10*b^8*c^4*d^6*f^4 - 3360*a^8*b^10*c^6*d^4*f^4 + 3024*a^11*b^7*c^5*d^
5*f^4 - 3024*a^10*b^8*c^6*d^4*f^4 - 3024*a^8*b^10*c^4*d^6*f^4 + 3024*a^7*b^
11*c^5*d^5*f^4 + 2320*a^9*b^9*c^7*d^3*f^4 + 2320*a^9*b^9*c^3*d^7*f^4 - 2240
*a^12*b^6*c^4*d^6*f^4 - 2240*a^6*b^12*c^6*d^4*f^4 + 2160*a^11*b^7*c^3*d^7*f
^4 + 2160*a^7*b^11*c^7*d^3*f^4 - 1624*a^12*b^6*c^6*d^4*f^4 - 1624*a^6*b^12*
c^4*d^6*f^4 + 1488*a^11*b^7*c^7*d^3*f^4 + 1488*a^7*b^11*c^3*d^7*f^4 + 1344*
a^13*b^5*c^5*d^5*f^4 + 1344*a^5*b^13*c^5*d^5*f^4 - 1320*a^10*b^8*c^2*d^8*f^
4 - 1320*a^8*b^10*c^8*d^2*f^4 + 1200*a^13*b^5*c^3*d^7*f^4 + 1200*a^5*b^13*c
^7*d^3*f^4 - 1060*a^12*b^6*c^2*d^8*f^4 - 1060*a^6*b^12*c^8*d^2*f^4 - 948*a^
10*b^8*c^8*d^2*f^4 - 948*a^8*b^10*c^2*d^8*f^4 - 840*a^14*b^4*c^4*d^6*f^4 -
840*a^4*b^14*c^6*d^4*f^4 + 528*a^13*b^5*c^7*d^3*f^4 + 528*a^5*b^13*c^3*d^7*
f^4 - 480*a^14*b^4*c^6*d^4*f^4 - 480*a^14*b^4*c^2*d^8*f^4 - 480*a^4*b^14*c^
8*d^2*f^4 - 480*a^4*b^14*c^4*d^6*f^4 + 368*a^15*b^3*c^3*d^7*f^4 - 368*a^12*
b^6*c^8*d^2*f^4 - 368*a^6*b^12*c^2*d^8*f^4 + 368*a^3*b^15*c^7*d^3*f^4 + 304
*a^15*b^3*c^5*d^5*f^4 + 304*a^3*b^15*c^5*d^5*f^4 - 144*a^16*b^2*c^4*d^6*f^4
 - 144*a^2*b^16*c^6*d^4*f^4 - 108*a^16*b^2*c^2*d^8*f^4 - 108*a^2*b^16*c^8*d
^2*f^4 + 80*a^15*b^3*c^7*d^3*f^4 + 80*a^3*b^15*c^3*d^7*f^4 - 60*a^16*b^2*c^
6*d^4*f^4 - 60*a^14*b^4*c^8*d^2*f^4 - 60*a^4*b^14*c^2*d^8*f^4 - 60*a^2*b^16
*c^4*d^6*f^4 - 8*b^18*c^8*d^2*f^4 - 4*b^18*c^6*d^4*f^4 - 8*a^18*c^2*d^8*f^4
 - 4*a^18*c^4*d^6*f^4 - 80*a^12*b^6*d^10*f^4 - 60*a^14*b^4*d^10*f^4 - 60*a^
10*b^8*d^10*f^4 - 24*a^16*b^2*d^10*f^4 - 24*a^8*b^10*d^10*f^4 - 4*a^6*b^12*
d^10*f^4 - 80*a^6*b^12*c^10*f^4 - 60*a^8*b^10*c^10*f^4 - 60*a^4*b^14*c^10*f
^4 - 24*a^10*b^8*c^10*f^4 - 24*a^2*b^16*c^10*f^4 - 4*a^12*b^6*c^10*f^4 - 4*
b^18*c^10*f^4 - 4*a^18*d^10*f^4 - 12*A*C*a^11*b*c*d^7*f^2 - 12*A*C*a*b^11*c

^7*d*f^2 - 912*B*C*a^5*b^7*c^4*d^4*f^2 - 792*B*C*a^8*b^4*c^3*d^5*f^2 + 792*
B*C*a^4*b^8*c^5*d^3*f^2 + 720*B*C*a^7*b^5*c^4*d^4*f^2 - 480*B*C*a^5*b^7*c^6
*d^2*f^2 - 408*B*C*a^5*b^7*c^2*d^6*f^2 + 384*B*C*a^7*b^5*c^2*d^6*f^2 - 336*
B*C*a^8*b^4*c^5*d^3*f^2 + 324*B*C*a^4*b^8*c^3*d^5*f^2 + 312*B*C*a^7*b^5*c^6
*d^2*f^2 - 248*B*C*a^3*b^9*c^6*d^2*f^2 + 216*B*C*a^9*b^3*c^2*d^6*f^2 - 196*
B*C*a^3*b^9*c^4*d^4*f^2 + 132*B*C*a^9*b^3*c^4*d^4*f^2 + 80*B*C*a^6*b^6*c^3*
d^5*f^2 - 64*B*C*a^6*b^6*c^5*d^3*f^2 - 36*B*C*a^2*b^10*c^3*d^5*f^2 - 28*B*C
*a^3*b^9*c^2*d^6*f^2 + 12*B*C*a^10*b^2*c^5*d^3*f^2 - 12*B*C*a^10*b^2*c^3*d^
5*f^2 - 12*B*C*a^2*b^10*c^5*d^3*f^2 - 4*B*C*a^9*b^3*c^6*d^2*f^2 - 1468*A*C*
a^6*b^6*c^4*d^4*f^2 + 996*A*C*a^7*b^5*c^3*d^5*f^2 + 900*A*C*a^5*b^7*c^5*d^3
*f^2 - 676*A*C*a^6*b^6*c^6*d^2*f^2 - 660*A*C*a^6*b^6*c^2*d^6*f^2 + 636*A*C*
a^5*b^7*c^3*d^5*f^2 + 540*A*C*a^7*b^5*c^5*d^3*f^2 - 236*A*C*a^3*b^9*c^5*d^3
*f^2 - 204*A*C*a^9*b^3*c^3*d^5*f^2 + 156*A*C*a^10*b^2*c^2*d^6*f^2 + 132*A*C
*a^2*b^10*c^6*d^2*f^2 - 72*A*C*a^9*b^3*c^5*d^3*f^2 - 72*A*C*a^4*b^8*c^6*d^2
*f^2 + 66*A*C*a^4*b^8*c^2*d^6*f^2 + 54*A*C*a^10*b^2*c^4*d^4*f^2 + 54*A*C*a^
2*b^10*c^4*d^4*f^2 - 48*A*C*a^8*b^4*c^2*d^6*f^2 - 48*A*C*a^4*b^8*c^4*d^4*f^
2 + 42*A*C*a^8*b^4*c^6*d^2*f^2 - 40*A*C*a^3*b^9*c^3*d^5*f^2 - 36*A*C*a^8*b^
4*c^4*d^4*f^2 + 24*A*C*a^2*b^10*c^2*d^6*f^2 + 960*A*B*a^5*b^7*c^4*d^4*f^2 -
864*A*B*a^4*b^8*c^5*d^3*f^2 + 756*A*B*a^8*b^4*c^3*d^5*f^2 - 744*A*B*a^7*b^
5*c^4*d^4*f^2 - 528*A*B*a^4*b^8*c^3*d^5*f^2 + 504*A*B*a^5*b^7*c^6*d^2*f^2 -
432*A*B*a^7*b^5*c^2*d^6*f^2 + 432*A*B*a^5*b^7*c^2*d^6*f^2 + 348*A*B*a^8*b^
4*c^5*d^3*f^2 - 312*A*B*a^7*b^5*c^6*d^2*f^2 - 284*A*B*a^9*b^3*c^2*d^6*f^2 +
280*A*B*a^3*b^9*c^6*d^2*f^2 + 264*A*B*a^3*b^9*c^4*d^4*f^2 - 240*A*B*a^6*b^
6*c^3*d^5*f^2 - 172*A*B*a^9*b^3*c^4*d^4*f^2 + 68*A*B*a^3*b^9*c^2*d^6*f^2 -
60*A*B*a^2*b^10*c^3*d^5*f^2 + 24*A*B*a^6*b^6*c^5*d^3*f^2 - 24*A*B*a^2*b^10*
c^5*d^3*f^2 + 12*A*B*a^10*b^2*c^3*d^5*f^2 + 360*B*C*a^4*b^8*c^7*d*f^2 - 336
*B*C*a^8*b^4*c*d^7*f^2 + 168*B*C*a^6*b^6*c*d^7*f^2 - 136*B*C*a^6*b^6*c^7*d*
f^2 - 36*B*C*a^11*b*c^2*d^6*f^2 + 36*B*C*a*b^11*c^6*d^2*f^2 + 24*B*C*a^10*b
^2*c*d^7*f^2 - 24*B*C*a^2*b^10*c^7*d*f^2 - 12*B*C*a^11*b*c^4*d^4*f^2 + 12*B
*C*a^4*b^8*c*d^7*f^2 + 12*B*C*a*b^11*c^4*d^4*f^2 + 444*A*C*a^7*b^5*c*d^7*f^
2 + 348*A*C*a^5*b^7*c^7*d*f^2 - 164*A*C*a^3*b^9*c^7*d*f^2 - 132*A*C*a^9*b^3
*c*d^7*f^2 + 84*A*C*a^5*b^7*c*d^7*f^2 + 32*A*C*a^3*b^9*c*d^7*f^2 - 12*A*C*a
^11*b*c^3*d^5*f^2 - 12*A*C*a^7*b^5*c^7*d*f^2 - 12*A*C*a*b^11*c^5*d^3*f^2 -
360*A*B*a^4*b^8*c^7*d*f^2 + 288*A*B*a^8*b^4*c*d^7*f^2 - 288*A*B*a^6*b^6*c*d
^7*f^2 - 144*A*B*a^4*b^8*c*d^7*f^2 + 136*A*B*a^6*b^6*c^7*d*f^2 - 60*A*B*a^2
*b^10*c*d^7*f^2 - 36*A*B*a^10*b^2*c*d^7*f^2 + 24*A*B*a^2*b^10*c^7*d*f^2 - 2
4*A*B*a*b^11*c^6*d^2*f^2 + 12*A*B*a^11*b*c^2*d^6*f^2 + 12*A*B*a*b^11*c^4*d^
4*f^2 + 12*A*B*a*b^11*c^2*d^6*f^2 - 8*B*C*b^12*c^5*d^3*f^2 - 8*B*C*b^12*c^3
*d^5*f^2 + 8*A*C*b^12*c^2*d^6*f^2 - 4*B*C*a^12*c^3*d^5*f^2 + 4*A*C*b^12*c^4
*d^4*f^2 - 2*A*C*b^12*c^6*d^2*f^2 + 80*B*C*a^9*b^3*d^8*f^2 - 24*B*C*a^7*b^5
*d^8*f^2 + 6*A*C*a^12*c^2*d^6*f^2 + 4*A*B*b^12*c^5*d^3*f^2 - 4*A*B*b^12*c^3
*d^5*f^2 - 90*A*C*a^8*b^4*d^8*f^2 - 80*B*C*a^3*b^9*c^8*f^2 + 54*A*C*a^10*b^
2*d^8*f^2 - 30*A*C*a^6*b^6*d^8*f^2 + 24*B*C*a^5*b^7*c^8*f^2 - 12*A*C*a^4*b^
8*d^8*f^2 - 112*A*B*a^9*b^3*d^8*f^2 - 66*A*C*a^4*b^8*c^8*f^2 + 54*A*C*a^2*b
^10*c^8*f^2 + 4*A*B*a^3*b^9*d^8*f^2 + 2*A*C*a^6*b^6*c^8*f^2 + 80*A*B*a^3*b^

9*c^8*f^2 - 24*A*B*a^5*b^7*c^8*f^2 + 726*C^2*a^6*b^6*c^4*d^4*f^2 - 402*C^2*
a^7*b^5*c^3*d^5*f^2 - 402*C^2*a^5*b^7*c^5*d^3*f^2 + 322*C^2*a^6*b^6*c^6*d^2
*f^2 + 322*C^2*a^6*b^6*c^2*d^6*f^2 - 222*C^2*a^7*b^5*c^5*d^3*f^2 - 222*C^2*
a^5*b^7*c^3*d^5*f^2 + 134*C^2*a^9*b^3*c^3*d^5*f^2 + 134*C^2*a^3*b^9*c^5*d^3
*f^2 - 66*C^2*a^10*b^2*c^2*d^6*f^2 - 66*C^2*a^2*b^10*c^6*d^2*f^2 + 52*C^2*a
^9*b^3*c^5*d^3*f^2 + 52*C^2*a^3*b^9*c^3*d^5*f^2 - 27*C^2*a^8*b^4*c^6*d^2*f^
2 - 27*C^2*a^4*b^8*c^2*d^6*f^2 + 24*C^2*a^8*b^4*c^4*d^4*f^2 + 24*C^2*a^8*b^
4*c^2*d^6*f^2 + 24*C^2*a^4*b^8*c^6*d^2*f^2 + 24*C^2*a^4*b^8*c^4*d^4*f^2 - 1
5*C^2*a^10*b^2*c^4*d^4*f^2 - 15*C^2*a^2*b^10*c^4*d^4*f^2 - 570*B^2*a^6*b^6*
c^4*d^4*f^2 + 366*B^2*a^7*b^5*c^3*d^5*f^2 + 318*B^2*a^5*b^7*c^5*d^3*f^2 - 2
62*B^2*a^6*b^6*c^6*d^2*f^2 - 222*B^2*a^6*b^6*c^2*d^6*f^2 - 210*B^2*a^3*b^9*
c^5*d^3*f^2 + 186*B^2*a^7*b^5*c^5*d^3*f^2 + 162*B^2*a^5*b^7*c^3*d^5*f^2 - 1
42*B^2*a^9*b^3*c^3*d^5*f^2 + 132*B^2*a^4*b^8*c^4*d^4*f^2 + 117*B^2*a^4*b^8*
c^2*d^6*f^2 + 102*B^2*a^2*b^10*c^6*d^2*f^2 - 96*B^2*a^3*b^9*c^3*d^5*f^2 + 9
0*B^2*a^10*b^2*c^2*d^6*f^2 + 81*B^2*a^2*b^10*c^4*d^4*f^2 - 56*B^2*a^9*b^3*c
^5*d^3*f^2 + 48*B^2*a^8*b^4*c^4*d^4*f^2 + 48*B^2*a^4*b^8*c^6*d^2*f^2 + 45*B
^2*a^8*b^4*c^6*d^2*f^2 + 36*B^2*a^8*b^4*c^2*d^6*f^2 + 36*B^2*a^2*b^10*c^2*d
^6*f^2 + 33*B^2*a^10*b^2*c^4*d^4*f^2 + 822*A^2*a^6*b^6*c^4*d^4*f^2 - 594*A^
2*a^7*b^5*c^3*d^5*f^2 + 498*A^2*a^6*b^6*c^2*d^6*f^2 - 498*A^2*a^5*b^7*c^5*d
^3*f^2 - 414*A^2*a^5*b^7*c^3*d^5*f^2 + 354*A^2*a^6*b^6*c^6*d^2*f^2 - 318*A^
2*a^7*b^5*c^5*d^3*f^2 + 144*A^2*a^8*b^4*c^2*d^6*f^2 + 102*A^2*a^3*b^9*c^5*d
^3*f^2 + 84*A^2*a^4*b^8*c^4*d^4*f^2 + 81*A^2*a^4*b^8*c^2*d^6*f^2 + 72*A^2*a
^8*b^4*c^4*d^4*f^2 + 70*A^2*a^9*b^3*c^3*d^5*f^2 - 66*A^2*a^2*b^10*c^6*d^2*f
^2 + 48*A^2*a^4*b^8*c^6*d^2*f^2 - 42*A^2*a^10*b^2*c^2*d^6*f^2 + 24*A^2*a^2*
b^10*c^2*d^6*f^2 + 20*A^2*a^9*b^3*c^5*d^3*f^2 - 15*A^2*a^10*b^2*c^4*d^4*f^2
 - 15*A^2*a^8*b^4*c^6*d^2*f^2 - 15*A^2*a^2*b^10*c^4*d^4*f^2 - 12*A^2*a^3*b^
9*c^3*d^5*f^2 - 8*B*C*b^12*c^7*d*f^2 + 4*B*C*a^12*c*d^7*f^2 - 24*B*C*a^11*b
*d^8*f^2 + 8*A*B*b^12*c^7*d*f^2 - 8*A*B*b^12*c*d^7*f^2 + 24*B*C*a*b^11*c^8*
f^2 - 8*A*B*a^12*c*d^7*f^2 + 12*A*B*a^11*b*d^8*f^2 - 24*A*B*a*b^11*c^8*f^2
 - 174*C^2*a^7*b^5*c*d^7*f^2 - 174*C^2*a^5*b^7*c^7*d*f^2 + 82*C^2*a^9*b^3*c*
d^7*f^2 + 82*C^2*a^3*b^9*c^7*d*f^2 + 6*C^2*a^11*b*c^3*d^5*f^2 + 6*C^2*a^7*b
^5*c^7*d*f^2 + 6*C^2*a^5*b^7*c*d^7*f^2 + 6*C^2*a*b^11*c^5*d^3*f^2 + 162*B^2
*a^7*b^5*c*d^7*f^2 + 138*B^2*a^5*b^7*c^7*d*f^2 - 118*B^2*a^3*b^9*c^7*d*f^2
 - 86*B^2*a^9*b^3*c*d^7*f^2 - 30*B^2*a*b^11*c^5*d^3*f^2 - 18*B^2*a^7*b^5*c^7
*d*f^2 - 18*B^2*a^5*b^7*c*d^7*f^2 - 12*B^2*a*b^11*c^3*d^5*f^2 - 6*B^2*a^11*
b*c^3*d^5*f^2 - 4*B^2*a^3*b^9*c*d^7*f^2 - 270*A^2*a^7*b^5*c*d^7*f^2 - 174*A
^2*a^5*b^7*c^7*d*f^2 - 90*A^2*a^5*b^7*c*d^7*f^2 + 82*A^2*a^3*b^9*c^7*d*f^2
 + 50*A^2*a^9*b^3*c*d^7*f^2 - 32*A^2*a^3*b^9*c*d^7*f^2 + 6*A^2*a^11*b*c^3*d^
5*f^2 + 6*A^2*a^7*b^5*c^7*d*f^2 + 6*A^2*a*b^11*c^5*d^3*f^2 + 6*C^2*a^11*b*c
*d^7*f^2 + 6*C^2*a*b^11*c^7*d*f^2 - 18*B^2*a*b^11*c^7*d*f^2 - 6*B^2*a^11*b*
c*d^7*f^2 + 6*A^2*a^11*b*c*d^7*f^2 + 6*A^2*a*b^11*c^7*d*f^2 - 6*A*C*b^12*c^
8*f^2 - 2*A*C*a^12*d^8*f^2 + 4*C^2*b^12*c^4*d^4*f^2 + 3*C^2*b^12*c^6*d^2*f^
2 + 4*C^2*a^12*c^4*d^4*f^2 + 4*B^2*b^12*c^4*d^4*f^2 + 4*B^2*b^12*c^2*d^6*f^
2 + 3*C^2*a^12*c^2*d^6*f^2 + 3*B^2*b^12*c^6*d^2*f^2 + 33*C^2*a^8*b^4*d^8*f^
2 - 27*C^2*a^10*b^2*d^8*f^2 - 4*A^2*b^12*c^4*d^4*f^2 + 3*B^2*a^12*c^2*d^6*f

^2 - C^2*a^6*b^6*d^8*f^2 - A^2*b^12*c^6*d^2*f^2 + 33*C^2*a^4*b^8*c^8*f^2 +
33*B^2*a^10*b^2*d^8*f^2 - 27*C^2*a^2*b^10*c^8*f^2 - 27*B^2*a^8*b^4*d^8*f^2
+ 3*B^2*a^6*b^6*d^8*f^2 - C^2*a^6*b^6*c^8*f^2 - A^2*a^12*c^2*d^6*f^2 + 117*
A^2*a^8*b^4*d^8*f^2 + 111*A^2*a^6*b^6*d^8*f^2 + 72*A^2*a^4*b^8*d^8*f^2 + 33
*B^2*a^2*b^10*c^8*f^2 - 27*B^2*a^4*b^8*c^8*f^2 + 24*A^2*a^2*b^10*d^8*f^2 +
3*B^2*a^6*b^6*c^8*f^2 - 3*A^2*a^10*b^2*d^8*f^2 + 33*A^2*a^4*b^8*c^8*f^2 - 2
7*A^2*a^2*b^10*c^8*f^2 - A^2*a^6*b^6*c^8*f^2 + 3*C^2*b^12*c^8*f^2 + 3*C^2*a
^12*d^8*f^2 + 4*A^2*b^12*d^8*f^2 - B^2*b^12*c^8*f^2 - B^2*a^12*d^8*f^2 + 3*
A^2*b^12*c^8*f^2 + 3*A^2*a^12*d^8*f^2 - 24*A*B*C*a*b^8*c*d^6*f + 342*A*B*C*
a^4*b^5*c^2*d^5*f - 186*A*B*C*a^5*b^4*c^3*d^4*f - 66*A*B*C*a^2*b^7*c^4*d^3*
f + 48*A*B*C*a^2*b^7*c^2*d^5*f + 42*A*B*C*a^6*b^3*c^2*d^5*f + 26*A*B*C*a^3*
b^6*c^5*d^2*f + 24*A*B*C*a^6*b^3*c^4*d^3*f - 18*A*B*C*a^7*b^2*c^3*d^4*f - 1
8*A*B*C*a^4*b^5*c^4*d^3*f - 8*A*B*C*a^3*b^6*c^3*d^4*f + 6*A*B*C*a^5*b^4*c^5
*d^2*f - 128*A*B*C*a^3*b^6*c*d^6*f + 126*A*B*C*a^7*b^2*c*d^6*f + 72*A*B*C*a
*b^8*c^3*d^4*f - 36*A*B*C*a^8*b*c^2*d^5*f - 36*A*B*C*a*b^8*c^5*d^2*f + 30*A
*B*C*a^2*b^7*c^6*d*f - 12*A*B*C*a^5*b^4*c*d^6*f - 12*A*B*C*a^4*b^5*c^6*d*f
- 21*B^2*C*a^8*b*c*d^6*f - 3*B^2*C*a*b^8*c^6*d*f + 21*A^2*C*a^8*b*c*d^6*f -
21*A*C^2*a^8*b*c*d^6*f - 9*A^2*C*a*b^8*c^6*d*f + 9*A*C^2*a*b^8*c^6*d*f + 3
6*A^2*B*a*b^8*c*d^6*f + 21*A*B^2*a^8*b*c*d^6*f + 3*A*B^2*a*b^8*c^6*d*f + 16
*A*B*C*b^9*c^4*d^3*f - 16*A*B*C*b^9*c^2*d^5*f - 78*A*B*C*a^6*b^3*d^7*f + 24
*A*B*C*a^4*b^5*d^7*f + 2*A*B*C*a^3*b^6*c^7*f - 237*B^2*C*a^4*b^5*c^3*d^4*f
+ 165*B*C^2*a^5*b^4*c^3*d^4*f + 92*B^2*C*a^3*b^6*c^2*d^5*f - 81*B^2*C*a^7*b
^2*c^2*d^5*f + 77*B^2*C*a^3*b^6*c^4*d^3*f - 75*B*C^2*a^4*b^5*c^2*d^5*f + 69
*B^2*C*a^5*b^4*c^4*d^3*f + 69*B*C^2*a^4*b^5*c^4*d^3*f - 68*B*C^2*a^3*b^6*c^
3*d^4*f - 63*B^2*C*a^4*b^5*c^5*d^2*f - 61*B*C^2*a^6*b^3*c^2*d^5*f + 57*B*C^
2*a^2*b^7*c^4*d^3*f - 53*B*C^2*a^3*b^6*c^5*d^2*f - 44*B*C^2*a^6*b^3*c^4*d^3
*f - 36*B^2*C*a^2*b^7*c^3*d^4*f + 35*B^2*C*a^6*b^3*c^3*d^4*f - 33*B^2*C*a^5
*b^4*c^2*d^5*f + 33*B^2*C*a^2*b^7*c^5*d^2*f + 33*B*C^2*a^7*b^2*c^3*d^4*f -
12*B^2*C*a^7*b^2*c^4*d^3*f + 9*B*C^2*a^5*b^4*c^5*d^2*f + 4*B^2*C*a^6*b^3*c^
5*d^2*f + 225*A^2*C*a^5*b^4*c^2*d^5*f - 105*A*C^2*a^5*b^4*c^2*d^5*f - 99*A^
2*C*a^4*b^5*c^3*d^4*f - 81*A^2*C*a^4*b^5*c^5*d^2*f + 67*A^2*C*a^3*b^6*c^4*d
^3*f - 59*A*C^2*a^3*b^6*c^4*d^3*f - 57*A*C^2*a^7*b^2*c^2*d^5*f + 57*A*C^2*a
^2*b^7*c^5*d^2*f + 51*A^2*C*a^5*b^4*c^4*d^3*f + 48*A^2*C*a^2*b^7*c^3*d^4*f
+ 45*A*C^2*a^4*b^5*c^5*d^2*f - 35*A^2*C*a^6*b^3*c^3*d^4*f + 33*A^2*C*a^7*b^
2*c^2*d^5*f - 33*A^2*C*a^2*b^7*c^5*d^2*f + 33*A*C^2*a^5*b^4*c^4*d^3*f + 27*
A*C^2*a^6*b^3*c^3*d^4*f + 24*A*C^2*a^3*b^6*c^2*d^5*f - 24*A*C^2*a^2*b^7*c^3
*d^4*f - 21*A*C^2*a^4*b^5*c^3*d^4*f - 16*A^2*C*a^3*b^6*c^2*d^5*f - 243*A^2*
B*a^4*b^5*c^2*d^5*f - 156*A*B^2*a^3*b^6*c^2*d^5*f + 141*A*B^2*a^4*b^5*c^3*d
^4*f + 108*A^2*B*a^3*b^6*c^3*d^4*f - 105*A*B^2*a^3*b^6*c^4*d^3*f + 84*A*B^2
*a^2*b^7*c^3*d^4*f + 81*A*B^2*a^5*b^4*c^2*d^5*f + 51*A^2*B*a^6*b^3*c^2*d^5*
f - 51*A^2*B*a^4*b^5*c^4*d^3*f - 48*A^2*B*a^2*b^7*c^2*d^5*f + 45*A^2*B*a^5*
b^4*c^3*d^4*f + 39*A*B^2*a^4*b^5*c^5*d^2*f - 35*A*B^2*a^6*b^3*c^3*d^4*f + 3
3*A*B^2*a^7*b^2*c^2*d^5*f + 27*A^2*B*a^3*b^6*c^5*d^2*f - 21*A*B^2*a^5*b^4*c
^4*d^3*f + 20*A^2*B*a^6*b^3*c^4*d^3*f - 15*A^2*B*a^7*b^2*c^3*d^4*f - 15*A^2
*B*a^5*b^4*c^5*d^2*f + 9*A^2*B*a^2*b^7*c^4*d^3*f + 3*A*B^2*a^2*b^7*c^5*d^2*

f + 2*A*B*C*b^9*c^6*d*f - 6*A*B*C*a^9*c*d^6*f + 18*A*B*C*a^8*b*d^7*f - 6*A*
B*C*a*b^8*c^7*f + 63*B^2*C*a^6*b^3*c*d^6*f - 48*B^2*C*a*b^8*c^4*d^3*f + 42*
B*C^2*a^8*b*c^2*d^5*f + 42*B*C^2*a^5*b^4*c*d^6*f - 39*B*C^2*a^7*b^2*c*d^6*f
 + 30*B*C^2*a*b^8*c^5*d^2*f - 24*B^2*C*a^4*b^5*c*d^6*f - 24*B*C^2*a*b^8*c^3
*d^4*f + 17*B^2*C*a^3*b^6*c^6*d*f - 15*B*C^2*a^2*b^7*c^6*d*f + 12*B^2*C*a^8
*b*c^3*d^4*f + 12*B^2*C*a*b^8*c^2*d^5*f + 6*B*C^2*a^4*b^5*c^6*d*f - 192*A^2
*C*a^4*b^5*c*d^6*f - 99*A^2*C*a^6*b^3*c*d^6*f + 84*A*C^2*a^4*b^5*c*d^6*f +
59*A*C^2*a^6*b^3*c*d^6*f + 51*A^2*C*a^3*b^6*c^6*d*f - 51*A*C^2*a^3*b^6*c^6*
d*f - 36*A^2*C*a*b^8*c^2*d^5*f - 24*A*C^2*a*b^8*c^4*d^3*f + 24*A*C^2*a*b^8*
c^2*d^5*f + 12*A^2*C*a*b^8*c^4*d^3*f + 12*A*C^2*a^8*b*c^3*d^4*f + 160*A^2*B
*a^3*b^6*c*d^6*f - 99*A*B^2*a^6*b^3*c*d^6*f - 87*A^2*B*a^7*b^2*c*d^6*f - 72
*A*B^2*a^4*b^5*c*d^6*f - 48*A*B^2*a*b^8*c^2*d^5*f - 36*A^2*B*a*b^8*c^3*d^4*
f + 24*A*B^2*a*b^8*c^4*d^3*f - 17*A*B^2*a^3*b^6*c^6*d*f - 15*A^2*B*a^2*b^7*
c^6*d*f + 12*A*B^2*a^2*b^7*c*d^6*f + 6*A^2*B*a^8*b*c^2*d^5*f - 6*A^2*B*a^5*
b^4*c*d^6*f + 6*A^2*B*a^4*b^5*c^6*d*f + 6*A^2*B*a*b^8*c^5*d^2*f + 12*B^2*C*
b^9*c^3*d^4*f - 12*B*C^2*b^9*c^4*d^3*f - 12*A^2*C*b^9*c^3*d^4*f - 8*A*C^2*b
^9*c^5*d^2*f + 8*A*C^2*b^9*c^3*d^4*f + 4*B^2*C*a^9*c^2*d^5*f + 4*A^2*C*b^9*
c^5*d^2*f - 4*B*C^2*a^9*c^3*d^4*f + 12*A^2*B*b^9*c^2*d^5*f - 8*A*B^2*b^9*c^
3*d^4*f - 4*A^2*B*b^9*c^4*d^3*f + 4*A*C^2*a^9*c^2*d^5*f + 3*B^2*C*a^7*b^2*d
^7*f - B*C^2*a^6*b^3*d^7*f + 96*A^2*C*a^5*b^4*d^7*f - 39*A^2*C*a^7*b^2*d^7*
f - 36*A*C^2*a^5*b^4*d^7*f + 32*A^2*C*a^3*b^6*d^7*f + 15*A*C^2*a^7*b^2*d^7*
f - 3*B^2*C*a^2*b^7*c^7*f - B*C^2*a^3*b^6*c^7*f + 111*A^2*B*a^6*b^3*d^7*f -
 39*A*B^2*a^7*b^2*d^7*f + 24*A*B^2*a^5*b^4*d^7*f - 9*A^2*C*a^2*b^7*c^7*f +
9*A*C^2*a^2*b^7*c^7*f - 4*A*B^2*a^3*b^6*d^7*f + 3*A*B^2*a^2*b^7*c^7*f - A^2
*B*a^3*b^6*c^7*f + 3*C^3*a^8*b*c*d^6*f - 3*C^3*a*b^8*c^6*d*f - 3*A^3*a^8*b*
c*d^6*f + 3*A^3*a*b^8*c^6*d*f - B*C^2*b^9*c^6*d*f + 4*A^2*C*b^9*c*d^6*f + 3
*B*C^2*a^9*c*d^6*f + 8*A*B^2*b^9*c*d^6*f + 3*B*C^2*a^8*b*d^7*f - A^2*B*b^9*
c^6*d*f + 12*A^2*C*a*b^8*d^7*f + 3*B*C^2*a*b^8*c^7*f - A^2*B*a^9*c*d^6*f -
9*A^2*B*a^8*b*d^7*f + 3*A^2*B*a*b^8*c^7*f - 39*C^3*a^5*b^4*c^4*d^3*f + 39*C
^3*a^4*b^5*c^3*d^4*f + 27*C^3*a^7*b^2*c^2*d^5*f - 27*C^3*a^2*b^7*c^5*d^2*f
- 17*C^3*a^6*b^3*c^3*d^4*f + 17*C^3*a^3*b^6*c^4*d^3*f + 3*C^3*a^5*b^4*c^2*d
^5*f - 3*C^3*a^4*b^5*c^5*d^2*f - 63*B^3*a^5*b^4*c^3*d^4*f + 57*B^3*a^4*b^5*
c^2*d^5*f - 51*B^3*a^2*b^7*c^4*d^3*f + 48*B^3*a^3*b^6*c^3*d^4*f + 31*B^3*a^
6*b^3*c^2*d^5*f + 27*B^3*a^3*b^6*c^5*d^2*f + 16*B^3*a^6*b^3*c^4*d^3*f - 15*
B^3*a^5*b^4*c^5*d^2*f - 12*B^3*a^2*b^7*c^2*d^5*f + 9*B^3*a^4*b^5*c^4*d^3*f
- 3*B^3*a^7*b^2*c^3*d^4*f - 123*A^3*a^5*b^4*c^2*d^5*f + 81*A^3*a^4*b^5*c^3*
d^4*f - 45*A^3*a^5*b^4*c^4*d^3*f + 39*A^3*a^4*b^5*c^5*d^2*f + 25*A^3*a^6*b^
3*c^3*d^4*f - 25*A^3*a^3*b^6*c^4*d^3*f - 24*A^3*a^2*b^7*c^3*d^4*f - 8*A^3*a
^3*b^6*c^2*d^5*f - 3*A^3*a^7*b^2*c^2*d^5*f + 3*A^3*a^2*b^7*c^5*d^2*f - 17*C
^3*a^6*b^3*c*d^6*f + 17*C^3*a^3*b^6*c^6*d*f - 12*C^3*a^8*b*c^3*d^4*f + 12*C
^3*a*b^8*c^4*d^3*f + 24*B^3*a*b^8*c^3*d^4*f + 21*B^3*a^7*b^2*c*d^6*f - 18*B
^3*a^5*b^4*c*d^6*f - 15*B^3*a^2*b^7*c^6*d*f - 6*B^3*a^8*b*c^2*d^5*f + 6*B^3
*a^4*b^5*c^6*d*f + 6*B^3*a*b^8*c^5*d^2*f + 4*B^3*a^3*b^6*c*d^6*f + 108*A^3*
a^4*b^5*c*d^6*f + 57*A^3*a^6*b^3*c*d^6*f - 17*A^3*a^3*b^6*c^6*d*f + 12*A^3*
a*b^8*c^2*d^5*f + 4*C^3*b^9*c^5*d^2*f - 4*C^3*a^9*c^2*d^5*f - 4*B^3*b^9*c^2

*d^5*f + 4*A^3*b^9*c^3*d^4*f + 3*C^3*a^7*b^2*d^7*f - 3*C^3*a^2*b^7*c^7*f -
B^3*a^6*b^3*d^7*f - 60*A^3*a^5*b^4*d^7*f - 32*A^3*a^3*b^6*d^7*f + 21*A^3*a^
7*b^2*d^7*f - B^3*a^3*b^6*c^7*f + 3*A^3*a^2*b^7*c^7*f - B^3*b^9*c^6*d*f - 4
*A^3*b^9*c*d^6*f - B^3*a^9*c*d^6*f + 3*B^3*a^8*b*d^7*f - 12*A^3*a*b^8*d^7*f
 + 3*B^3*a*b^8*c^7*f - B^2*C*a^9*d^7*f - 4*A^2*B*b^9*d^7*f + 3*A^2*C*b^9*c^
7*f - 3*A*C^2*b^9*c^7*f - A*C^2*a^9*d^7*f - A*B^2*b^9*c^7*f - C^3*a^9*d^7*f
 - A^3*b^9*c^7*f + B^2*C*b^9*c^7*f + A^2*C*a^9*d^7*f + A*B^2*a^9*d^7*f + C^
3*b^9*c^7*f + A^3*a^9*d^7*f - 6*A*B^2*C*a^5*b*c*d^5 - 21*A^2*B*C*a^3*b^3*c^
2*d^4 + 21*A*B*C^2*a^3*b^3*c^2*d^4 + 12*A*B^2*C*a^4*b^2*c^2*d^4 - 12*A*B^2*
C*a^2*b^4*c^2*d^4 - 10*A*B^2*C*a^3*b^3*c^3*d^3 - 6*A*B*C^2*a^4*b^2*c^3*d^3
+ 3*A^2*B*C*a^4*b^2*c^3*d^3 + 3*A^2*B*C*a^2*b^4*c^3*d^3 + 3*A*B^2*C*a^2*b^4
*c^4*d^2 + 3*A*B*C^2*a^2*b^4*c^3*d^3 + 2*A*B*C^2*a^3*b^3*c^4*d^2 - A^2*B*C*
a^3*b^3*c^4*d^2 + 18*A^2*B*C*a^2*b^4*c*d^5 + 10*A*B^2*C*a^3*b^3*c*d^5 + 9*A
^2*B*C*a^4*b^2*c*d^5 - 9*A*B*C^2*a^4*b^2*c*d^5 - 9*A*B*C^2*a^2*b^4*c*d^5 -
6*A^2*B*C*a*b^5*c^2*d^4 + 6*A*B^2*C*a*b^5*c^3*d^3 + 6*A*B*C^2*a^5*b*c^2*d^4
 - 6*A*B*C^2*a*b^5*c^4*d^2 - 3*A^2*B*C*a^5*b*c^2*d^4 + 3*A^2*B*C*a*b^5*c^4*
d^2 + 3*A*B*C^2*a*b^5*c^2*d^4 - 3*B^3*C*a^5*b*c^2*d^4 + 3*B^3*C*a^4*b^2*c*d
^5 + 3*B^3*C*a*b^5*c^4*d^2 + 3*B^2*C^2*a^5*b*c*d^5 - 3*B*C^3*a^5*b*c^2*d^4
+ 3*B*C^3*a^4*b^2*c*d^5 + 3*B*C^3*a*b^5*c^4*d^2 + 24*A^3*C*a^3*b^3*c*d^5 +
8*A*C^3*a^3*b^3*c*d^5 - 9*A^3*B*a^2*b^4*c*d^5 - 9*A*B^3*a^2*b^4*c*d^5 - 3*A
^3*B*a^4*b^2*c*d^5 + 3*A^3*B*a*b^5*c^2*d^4 + 3*A^2*B^2*a^5*b*c*d^5 - 3*A*B^
3*a^4*b^2*c*d^5 + 3*A*B^3*a*b^5*c^2*d^4 + 5*A*B*C^2*b^6*c^3*d^3 - 4*A^2*B*C
*b^6*c^3*d^3 - A*B^2*C*b^6*c^4*d^2 - 3*A*B^2*C*a^4*b^2*d^6 - 2*A^2*B*C*a^3*
b^3*d^6 + 9*B^2*C^2*a^3*b^3*c^3*d^3 - 6*B^2*C^2*a^4*b^2*c^2*d^4 + 6*B^2*C^2
*a^2*b^4*c^2*d^4 - 3*B^2*C^2*a^2*b^4*c^4*d^2 + 24*A^2*C^2*a^3*b^3*c^3*d^3 -
 15*A^2*C^2*a^4*b^2*c^2*d^4 - 9*A^2*C^2*a^2*b^4*c^4*d^2 + 3*A^2*C^2*a^2*b^4
*c^2*d^4 + 9*A^2*B^2*a^2*b^4*c^2*d^4 - 3*A^2*B^2*a^4*b^2*c^2*d^4 + 4*A^2*B*
C*b^6*c*d^5 - 2*A*B*C^2*b^6*c*d^5 + 2*A*B*C^2*a^6*c*d^5 - A^2*B*C*a^6*c*d^5
 + 6*A^2*B*C*a^5*b*d^6 - 3*A*B*C^2*a^5*b*d^6 - 7*B^3*C*a^3*b^3*c^2*d^4 - 7*
B*C^3*a^3*b^3*c^2*d^4 + 3*B^3*C*a^4*b^2*c^3*d^3 - 3*B^3*C*a^2*b^4*c^3*d^3 -
 3*B^2*C^2*a*b^5*c^3*d^3 + 3*B*C^3*a^4*b^2*c^3*d^3 - 3*B*C^3*a^2*b^4*c^3*d^
3 - B^3*C*a^3*b^3*c^4*d^2 - B^2*C^2*a^3*b^3*c*d^5 - B*C^3*a^3*b^3*c^4*d^2 -
 24*A^2*C^2*a^3*b^3*c*d^5 - 24*A*C^3*a^3*b^3*c^3*d^3 + 12*A*C^3*a^4*b^2*c^2
*d^4 + 9*A*C^3*a^2*b^4*c^4*d^2 - 8*A^3*C*a^3*b^3*c^3*d^3 + 6*A^3*C*a^4*b^2*
c^2*d^4 - 6*A^3*C*a^2*b^4*c^2*d^4 + 3*A^3*C*a^2*b^4*c^4*d^2 - 9*A^2*B^2*a^3
*b^3*c*d^5 + 7*A^3*B*a^3*b^3*c^2*d^4 + 7*A*B^3*a^3*b^3*c^2*d^4 - 3*A^3*B*a^
2*b^4*c^3*d^3 - 3*A^2*B^2*a*b^5*c^3*d^3 - 3*A*B^3*a^2*b^4*c^3*d^3 - 5*A^2*C
^2*b^6*c^2*d^4 + 3*A^2*C^2*b^6*c^4*d^2 + 12*A^2*C^2*a^4*b^2*d^6 + 3*A^2*C^2
*a^2*b^4*d^6 + 6*A^2*B^2*a^4*b^2*d^6 + 3*A^2*B^2*a^2*b^4*d^6 + A*B*C^2*a^3*
b^3*d^6 - 3*B^4*a*b^5*c^3*d^3 - B^4*a^3*b^3*c*d^5 + A^2*B^2*a^3*b^3*c^3*d^3
 - 8*A^4*a^3*b^3*c*d^5 - 2*B^3*C*b^6*c^3*d^3 - 2*B*C^3*b^6*c^3*d^3 + 4*A^3*
C*b^6*c^2*d^4 - 3*A*C^3*b^6*c^4*d^2 + 2*A*C^3*b^6*c^2*d^4 - A^3*C*b^6*c^4*d
^2 - 2*A*C^3*a^6*c^2*d^4 - 15*A^3*C*a^4*b^2*d^6 - 6*A^3*C*a^2*b^4*d^6 - 3*A
*C^3*a^4*b^2*d^6 + 3*B^4*a^5*b*c*d^5 - B^3*C*a^6*c*d^5 - B*C^3*a^6*c*d^5 -
2*A^3*B*b^6*c*d^5 - 2*A*B^3*b^6*c*d^5 - 3*A^3*B*a^5*b*d^6 - 3*A*B^3*a^5*b*d

^6 + 8*C^4*a^3*b^3*c^3*d^3 - 3*C^4*a^4*b^2*c^2*d^4 - 3*C^4*a^2*b^4*c^4*d^2
+ 6*B^4*a^2*b^4*c^2*d^4 - 3*B^4*a^4*b^2*c^2*d^4 + 3*A^4*a^2*b^4*c^2*d^4 + B
^2*C^2*b^6*c^4*d^2 + B^2*C^2*b^6*c^2*d^4 + B^2*C^2*a^6*c^2*d^4 + A^2*C^2*a^
6*c^2*d^4 - 2*A^3*C*b^6*d^6 + A^3*B*b^6*c^3*d^3 + A*B^3*b^6*c^3*d^3 + A^3*B
*a^3*b^3*d^6 + A*B^3*a^3*b^3*d^6 - A^4*b^6*c^2*d^4 + 6*A^4*a^4*b^2*d^6 + 3*
A^4*a^2*b^4*d^6 - 2*A^2*C^2*a^6*d^6 + A*B^2*C*a^6*d^6 + B^4*a^3*b^3*c^3*d^3
 + A^3*C*a^6*d^6 + A*C^3*a^6*d^6 + C^4*b^6*c^4*d^2 + C^4*a^6*c^2*d^4 + B^4*
b^6*c^2*d^4 + A^2*C^2*b^6*d^6 + A^2*B^2*b^6*d^6 + A^4*b^6*d^6, f, k)*(root(
480*a^11*b^7*c*d^9*f^4 + 480*a^7*b^11*c^9*d*f^4 + 360*a^13*b^5*c*d^9*f^4 +
360*a^9*b^9*c^9*d*f^4 + 360*a^9*b^9*c*d^9*f^4 + 360*a^5*b^13*c^9*d*f^4 + 14
4*a^15*b^3*c*d^9*f^4 + 144*a^11*b^7*c^9*d*f^4 + 144*a^7*b^11*c*d^9*f^4 + 14
4*a^3*b^15*c^9*d*f^4 + 48*a^17*b*c^3*d^7*f^4 + 48*a*b^17*c^7*d^3*f^4 + 24*a
^17*b*c^5*d^5*f^4 + 24*a^13*b^5*c^9*d*f^4 + 24*a^5*b^13*c*d^9*f^4 + 24*a*b^
17*c^5*d^5*f^4 + 24*a^17*b*c*d^9*f^4 + 24*a*b^17*c^9*d*f^4 + 3920*a^9*b^9*c
^5*d^5*f^4 - 3360*a^10*b^8*c^4*d^6*f^4 - 3360*a^8*b^10*c^6*d^4*f^4 + 3024*a
^11*b^7*c^5*d^5*f^4 - 3024*a^10*b^8*c^6*d^4*f^4 - 3024*a^8*b^10*c^4*d^6*f^4
 + 3024*a^7*b^11*c^5*d^5*f^4 + 2320*a^9*b^9*c^7*d^3*f^4 + 2320*a^9*b^9*c^3*
d^7*f^4 - 2240*a^12*b^6*c^4*d^6*f^4 - 2240*a^6*b^12*c^6*d^4*f^4 + 2160*a^11
*b^7*c^3*d^7*f^4 + 2160*a^7*b^11*c^7*d^3*f^4 - 1624*a^12*b^6*c^6*d^4*f^4 -
1624*a^6*b^12*c^4*d^6*f^4 + 1488*a^11*b^7*c^7*d^3*f^4 + 1488*a^7*b^11*c^3*d
^7*f^4 + 1344*a^13*b^5*c^5*d^5*f^4 + 1344*a^5*b^13*c^5*d^5*f^4 - 1320*a^10*
b^8*c^2*d^8*f^4 - 1320*a^8*b^10*c^8*d^2*f^4 + 1200*a^13*b^5*c^3*d^7*f^4 + 1
200*a^5*b^13*c^7*d^3*f^4 - 1060*a^12*b^6*c^2*d^8*f^4 - 1060*a^6*b^12*c^8*d^
2*f^4 - 948*a^10*b^8*c^8*d^2*f^4 - 948*a^8*b^10*c^2*d^8*f^4 - 840*a^14*b^4*
c^4*d^6*f^4 - 840*a^4*b^14*c^6*d^4*f^4 + 528*a^13*b^5*c^7*d^3*f^4 + 528*a^5
*b^13*c^3*d^7*f^4 - 480*a^14*b^4*c^6*d^4*f^4 - 480*a^14*b^4*c^2*d^8*f^4 - 4
80*a^4*b^14*c^8*d^2*f^4 - 480*a^4*b^14*c^4*d^6*f^4 + 368*a^15*b^3*c^3*d^7*f
^4 - 368*a^12*b^6*c^8*d^2*f^4 - 368*a^6*b^12*c^2*d^8*f^4 + 368*a^3*b^15*c^7
*d^3*f^4 + 304*a^15*b^3*c^5*d^5*f^4 + 304*a^3*b^15*c^5*d^5*f^4 - 144*a^16*b
^2*c^4*d^6*f^4 - 144*a^2*b^16*c^6*d^4*f^4 - 108*a^16*b^2*c^2*d^8*f^4 - 108*
a^2*b^16*c^8*d^2*f^4 + 80*a^15*b^3*c^7*d^3*f^4 + 80*a^3*b^15*c^3*d^7*f^4 -
60*a^16*b^2*c^6*d^4*f^4 - 60*a^14*b^4*c^8*d^2*f^4 - 60*a^4*b^14*c^2*d^8*f^4
 - 60*a^2*b^16*c^4*d^6*f^4 - 8*b^18*c^8*d^2*f^4 - 4*b^18*c^6*d^4*f^4 - 8*a^
18*c^2*d^8*f^4 - 4*a^18*c^4*d^6*f^4 - 80*a^12*b^6*d^10*f^4 - 60*a^14*b^4*d^
10*f^4 - 60*a^10*b^8*d^10*f^4 - 24*a^16*b^2*d^10*f^4 - 24*a^8*b^10*d^10*f^4
 - 4*a^6*b^12*d^10*f^4 - 80*a^6*b^12*c^10*f^4 - 60*a^8*b^10*c^10*f^4 - 60*a
^4*b^14*c^10*f^4 - 24*a^10*b^8*c^10*f^4 - 24*a^2*b^16*c^10*f^4 - 4*a^12*b^6
*c^10*f^4 - 4*b^18*c^10*f^4 - 4*a^18*d^10*f^4 - 12*A*C*a^11*b*c*d^7*f^2 - 1
2*A*C*a*b^11*c^7*d*f^2 - 912*B*C*a^5*b^7*c^4*d^4*f^2 - 792*B*C*a^8*b^4*c^3*
d^5*f^2 + 792*B*C*a^4*b^8*c^5*d^3*f^2 + 720*B*C*a^7*b^5*c^4*d^4*f^2 - 480*B
*C*a^5*b^7*c^6*d^2*f^2 - 408*B*C*a^5*b^7*c^2*d^6*f^2 + 384*B*C*a^7*b^5*c^2*
d^6*f^2 - 336*B*C*a^8*b^4*c^5*d^3*f^2 + 324*B*C*a^4*b^8*c^3*d^5*f^2 + 312*B
*C*a^7*b^5*c^6*d^2*f^2 - 248*B*C*a^3*b^9*c^6*d^2*f^2 + 216*B*C*a^9*b^3*c^2*
d^6*f^2 - 196*B*C*a^3*b^9*c^4*d^4*f^2 + 132*B*C*a^9*b^3*c^4*d^4*f^2 + 80*B*
C*a^6*b^6*c^3*d^5*f^2 - 64*B*C*a^6*b^6*c^5*d^3*f^2 - 36*B*C*a^2*b^10*c^3*d^

5*f^2 - 28*B*C*a^3*b^9*c^2*d^6*f^2 + 12*B*C*a^10*b^2*c^5*d^3*f^2 - 12*B*C*a
^10*b^2*c^3*d^5*f^2 - 12*B*C*a^2*b^10*c^5*d^3*f^2 - 4*B*C*a^9*b^3*c^6*d^2*f
^2 - 1468*A*C*a^6*b^6*c^4*d^4*f^2 + 996*A*C*a^7*b^5*c^3*d^5*f^2 + 900*A*C*a
^5*b^7*c^5*d^3*f^2 - 676*A*C*a^6*b^6*c^6*d^2*f^2 - 660*A*C*a^6*b^6*c^2*d^6*
f^2 + 636*A*C*a^5*b^7*c^3*d^5*f^2 + 540*A*C*a^7*b^5*c^5*d^3*f^2 - 236*A*C*a
^3*b^9*c^5*d^3*f^2 - 204*A*C*a^9*b^3*c^3*d^5*f^2 + 156*A*C*a^10*b^2*c^2*d^6
*f^2 + 132*A*C*a^2*b^10*c^6*d^2*f^2 - 72*A*C*a^9*b^3*c^5*d^3*f^2 - 72*A*C*a
^4*b^8*c^6*d^2*f^2 + 66*A*C*a^4*b^8*c^2*d^6*f^2 + 54*A*C*a^10*b^2*c^4*d^4*f
^2 + 54*A*C*a^2*b^10*c^4*d^4*f^2 - 48*A*C*a^8*b^4*c^2*d^6*f^2 - 48*A*C*a^4*
b^8*c^4*d^4*f^2 + 42*A*C*a^8*b^4*c^6*d^2*f^2 - 40*A*C*a^3*b^9*c^3*d^5*f^2 -
36*A*C*a^8*b^4*c^4*d^4*f^2 + 24*A*C*a^2*b^10*c^2*d^6*f^2 + 960*A*B*a^5*b^7
*c^4*d^4*f^2 - 864*A*B*a^4*b^8*c^5*d^3*f^2 + 756*A*B*a^8*b^4*c^3*d^5*f^2 -
744*A*B*a^7*b^5*c^4*d^4*f^2 - 528*A*B*a^4*b^8*c^3*d^5*f^2 + 504*A*B*a^5*b^7
*c^6*d^2*f^2 - 432*A*B*a^7*b^5*c^2*d^6*f^2 + 432*A*B*a^5*b^7*c^2*d^6*f^2 +
348*A*B*a^8*b^4*c^5*d^3*f^2 - 312*A*B*a^7*b^5*c^6*d^2*f^2 - 284*A*B*a^9*b^3
*c^2*d^6*f^2 + 280*A*B*a^3*b^9*c^6*d^2*f^2 + 264*A*B*a^3*b^9*c^4*d^4*f^2 -
240*A*B*a^6*b^6*c^3*d^5*f^2 - 172*A*B*a^9*b^3*c^4*d^4*f^2 + 68*A*B*a^3*b^9*
c^2*d^6*f^2 - 60*A*B*a^2*b^10*c^3*d^5*f^2 + 24*A*B*a^6*b^6*c^5*d^3*f^2 - 24
*A*B*a^2*b^10*c^5*d^3*f^2 + 12*A*B*a^10*b^2*c^3*d^5*f^2 + 360*B*C*a^4*b^8*c
^7*d*f^2 - 336*B*C*a^8*b^4*c*d^7*f^2 + 168*B*C*a^6*b^6*c*d^7*f^2 - 136*B*C*
a^6*b^6*c^7*d*f^2 - 36*B*C*a^11*b*c^2*d^6*f^2 + 36*B*C*a*b^11*c^6*d^2*f^2 +
24*B*C*a^10*b^2*c*d^7*f^2 - 24*B*C*a^2*b^10*c^7*d*f^2 - 12*B*C*a^11*b*c^4*
d^4*f^2 + 12*B*C*a^4*b^8*c*d^7*f^2 + 12*B*C*a*b^11*c^4*d^4*f^2 + 444*A*C*a^
7*b^5*c*d^7*f^2 + 348*A*C*a^5*b^7*c^7*d*f^2 - 164*A*C*a^3*b^9*c^7*d*f^2 - 1
32*A*C*a^9*b^3*c*d^7*f^2 + 84*A*C*a^5*b^7*c*d^7*f^2 + 32*A*C*a^3*b^9*c*d^7*
f^2 - 12*A*C*a^11*b*c^3*d^5*f^2 - 12*A*C*a^7*b^5*c^7*d*f^2 - 12*A*C*a*b^11*
c^5*d^3*f^2 - 360*A*B*a^4*b^8*c^7*d*f^2 + 288*A*B*a^8*b^4*c*d^7*f^2 - 288*A
*B*a^6*b^6*c*d^7*f^2 - 144*A*B*a^4*b^8*c*d^7*f^2 + 136*A*B*a^6*b^6*c^7*d*f^
2 - 60*A*B*a^2*b^10*c*d^7*f^2 - 36*A*B*a^10*b^2*c*d^7*f^2 + 24*A*B*a^2*b^10
*c^7*d*f^2 - 24*A*B*a*b^11*c^6*d^2*f^2 + 12*A*B*a^11*b*c^2*d^6*f^2 + 12*A*B
*a*b^11*c^4*d^4*f^2 + 12*A*B*a*b^11*c^2*d^6*f^2 - 8*B*C*b^12*c^5*d^3*f^2 -
8*B*C*b^12*c^3*d^5*f^2 + 8*A*C*b^12*c^2*d^6*f^2 - 4*B*C*a^12*c^3*d^5*f^2 +
4*A*C*b^12*c^4*d^4*f^2 - 2*A*C*b^12*c^6*d^2*f^2 + 80*B*C*a^9*b^3*d^8*f^2 -
24*B*C*a^7*b^5*d^8*f^2 + 6*A*C*a^12*c^2*d^6*f^2 + 4*A*B*b^12*c^5*d^3*f^2 -
4*A*B*b^12*c^3*d^5*f^2 - 90*A*C*a^8*b^4*d^8*f^2 - 80*B*C*a^3*b^9*c^8*f^2 +
54*A*C*a^10*b^2*d^8*f^2 - 30*A*C*a^6*b^6*d^8*f^2 + 24*B*C*a^5*b^7*c^8*f^2 -
12*A*C*a^4*b^8*d^8*f^2 - 112*A*B*a^9*b^3*d^8*f^2 - 66*A*C*a^4*b^8*c^8*f^2
+ 54*A*C*a^2*b^10*c^8*f^2 + 4*A*B*a^3*b^9*d^8*f^2 + 2*A*C*a^6*b^6*c^8*f^2 +
80*A*B*a^3*b^9*c^8*f^2 - 24*A*B*a^5*b^7*c^8*f^2 + 726*C^2*a^6*b^6*c^4*d^4*
f^2 - 402*C^2*a^7*b^5*c^3*d^5*f^2 - 402*C^2*a^5*b^7*c^5*d^3*f^2 + 322*C^2*a
^6*b^6*c^6*d^2*f^2 + 322*C^2*a^6*b^6*c^2*d^6*f^2 - 222*C^2*a^7*b^5*c^5*d^3*
f^2 - 222*C^2*a^5*b^7*c^3*d^5*f^2 + 134*C^2*a^9*b^3*c^3*d^5*f^2 + 134*C^2*a
^3*b^9*c^5*d^3*f^2 - 66*C^2*a^10*b^2*c^2*d^6*f^2 - 66*C^2*a^2*b^10*c^6*d^2*
f^2 + 52*C^2*a^9*b^3*c^5*d^3*f^2 + 52*C^2*a^3*b^9*c^3*d^5*f^2 - 27*C^2*a^8*
b^4*c^6*d^2*f^2 - 27*C^2*a^4*b^8*c^2*d^6*f^2 + 24*C^2*a^8*b^4*c^4*d^4*f^2 +

24*C^2*a^8*b^4*c^2*d^6*f^2 + 24*C^2*a^4*b^8*c^6*d^2*f^2 + 24*C^2*a^4*b^8*c
^4*d^4*f^2 - 15*C^2*a^10*b^2*c^4*d^4*f^2 - 15*C^2*a^2*b^10*c^4*d^4*f^2 - 57
0*B^2*a^6*b^6*c^4*d^4*f^2 + 366*B^2*a^7*b^5*c^3*d^5*f^2 + 318*B^2*a^5*b^7*c
^5*d^3*f^2 - 262*B^2*a^6*b^6*c^6*d^2*f^2 - 222*B^2*a^6*b^6*c^2*d^6*f^2 - 21
0*B^2*a^3*b^9*c^5*d^3*f^2 + 186*B^2*a^7*b^5*c^5*d^3*f^2 + 162*B^2*a^5*b^7*c
^3*d^5*f^2 - 142*B^2*a^9*b^3*c^3*d^5*f^2 + 132*B^2*a^4*b^8*c^4*d^4*f^2 + 11
7*B^2*a^4*b^8*c^2*d^6*f^2 + 102*B^2*a^2*b^10*c^6*d^2*f^2 - 96*B^2*a^3*b^9*c
^3*d^5*f^2 + 90*B^2*a^10*b^2*c^2*d^6*f^2 + 81*B^2*a^2*b^10*c^4*d^4*f^2 - 56
*B^2*a^9*b^3*c^5*d^3*f^2 + 48*B^2*a^8*b^4*c^4*d^4*f^2 + 48*B^2*a^4*b^8*c^6*
d^2*f^2 + 45*B^2*a^8*b^4*c^6*d^2*f^2 + 36*B^2*a^8*b^4*c^2*d^6*f^2 + 36*B^2*
a^2*b^10*c^2*d^6*f^2 + 33*B^2*a^10*b^2*c^4*d^4*f^2 + 822*A^2*a^6*b^6*c^4*d^
4*f^2 - 594*A^2*a^7*b^5*c^3*d^5*f^2 + 498*A^2*a^6*b^6*c^2*d^6*f^2 - 498*A^2
*a^5*b^7*c^5*d^3*f^2 - 414*A^2*a^5*b^7*c^3*d^5*f^2 + 354*A^2*a^6*b^6*c^6*d^
2*f^2 - 318*A^2*a^7*b^5*c^5*d^3*f^2 + 144*A^2*a^8*b^4*c^2*d^6*f^2 + 102*A^2
*a^3*b^9*c^5*d^3*f^2 + 84*A^2*a^4*b^8*c^4*d^4*f^2 + 81*A^2*a^4*b^8*c^2*d^6*
f^2 + 72*A^2*a^8*b^4*c^4*d^4*f^2 + 70*A^2*a^9*b^3*c^3*d^5*f^2 - 66*A^2*a^2*
b^10*c^6*d^2*f^2 + 48*A^2*a^4*b^8*c^6*d^2*f^2 - 42*A^2*a^10*b^2*c^2*d^6*f^2
 + 24*A^2*a^2*b^10*c^2*d^6*f^2 + 20*A^2*a^9*b^3*c^5*d^3*f^2 - 15*A^2*a^10*b
^2*c^4*d^4*f^2 - 15*A^2*a^8*b^4*c^6*d^2*f^2 - 15*A^2*a^2*b^10*c^4*d^4*f^2 -
 12*A^2*a^3*b^9*c^3*d^5*f^2 - 8*B*C*b^12*c^7*d*f^2 + 4*B*C*a^12*c*d^7*f^2 -
 24*B*C*a^11*b*d^8*f^2 + 8*A*B*b^12*c^7*d*f^2 - 8*A*B*b^12*c*d^7*f^2 + 24*B
*C*a*b^11*c^8*f^2 - 8*A*B*a^12*c*d^7*f^2 + 12*A*B*a^11*b*d^8*f^2 - 24*A*B*a
*b^11*c^8*f^2 - 174*C^2*a^7*b^5*c*d^7*f^2 - 174*C^2*a^5*b^7*c^7*d*f^2 + 82*
C^2*a^9*b^3*c*d^7*f^2 + 82*C^2*a^3*b^9*c^7*d*f^2 + 6*C^2*a^11*b*c^3*d^5*f^2
 + 6*C^2*a^7*b^5*c^7*d*f^2 + 6*C^2*a^5*b^7*c*d^7*f^2 + 6*C^2*a*b^11*c^5*d^3
*f^2 + 162*B^2*a^7*b^5*c*d^7*f^2 + 138*B^2*a^5*b^7*c^7*d*f^2 - 118*B^2*a^3*
b^9*c^7*d*f^2 - 86*B^2*a^9*b^3*c*d^7*f^2 - 30*B^2*a*b^11*c^5*d^3*f^2 - 18*B
^2*a^7*b^5*c^7*d*f^2 - 18*B^2*a^5*b^7*c*d^7*f^2 - 12*B^2*a*b^11*c^3*d^5*f^2
 - 6*B^2*a^11*b*c^3*d^5*f^2 - 4*B^2*a^3*b^9*c*d^7*f^2 - 270*A^2*a^7*b^5*c*d
^7*f^2 - 174*A^2*a^5*b^7*c^7*d*f^2 - 90*A^2*a^5*b^7*c*d^7*f^2 + 82*A^2*a^3*
b^9*c^7*d*f^2 + 50*A^2*a^9*b^3*c*d^7*f^2 - 32*A^2*a^3*b^9*c*d^7*f^2 + 6*A^2
*a^11*b*c^3*d^5*f^2 + 6*A^2*a^7*b^5*c^7*d*f^2 + 6*A^2*a*b^11*c^5*d^3*f^2 +
6*C^2*a^11*b*c*d^7*f^2 + 6*C^2*a*b^11*c^7*d*f^2 - 18*B^2*a*b^11*c^7*d*f^2 -
 6*B^2*a^11*b*c*d^7*f^2 + 6*A^2*a^11*b*c*d^7*f^2 + 6*A^2*a*b^11*c^7*d*f^2 -
 6*A*C*b^12*c^8*f^2 - 2*A*C*a^12*d^8*f^2 + 4*C^2*b^12*c^4*d^4*f^2 + 3*C^2*b
^12*c^6*d^2*f^2 + 4*C^2*a^12*c^4*d^4*f^2 + 4*B^2*b^12*c^4*d^4*f^2 + 4*B^2*b
^12*c^2*d^6*f^2 + 3*C^2*a^12*c^2*d^6*f^2 + 3*B^2*b^12*c^6*d^2*f^2 + 33*C^2*
a^8*b^4*d^8*f^2 - 27*C^2*a^10*b^2*d^8*f^2 - 4*A^2*b^12*c^4*d^4*f^2 + 3*B^2*
a^12*c^2*d^6*f^2 - C^2*a^6*b^6*d^8*f^2 - A^2*b^12*c^6*d^2*f^2 + 33*C^2*a^4*
b^8*c^8*f^2 + 33*B^2*a^10*b^2*d^8*f^2 - 27*C^2*a^2*b^10*c^8*f^2 - 27*B^2*a^
8*b^4*d^8*f^2 + 3*B^2*a^6*b^6*d^8*f^2 - C^2*a^6*b^6*c^8*f^2 - A^2*a^12*c^2*
d^6*f^2 + 117*A^2*a^8*b^4*d^8*f^2 + 111*A^2*a^6*b^6*d^8*f^2 + 72*A^2*a^4*b^
8*d^8*f^2 + 33*B^2*a^2*b^10*c^8*f^2 - 27*B^2*a^4*b^8*c^8*f^2 + 24*A^2*a^2*b
^10*d^8*f^2 + 3*B^2*a^6*b^6*c^8*f^2 - 3*A^2*a^10*b^2*d^8*f^2 + 33*A^2*a^4*b
^8*c^8*f^2 - 27*A^2*a^2*b^10*c^8*f^2 - A^2*a^6*b^6*c^8*f^2 + 3*C^2*b^12*c^8

*f^2 + 3*C^2*a^12*d^8*f^2 + 4*A^2*b^12*d^8*f^2 - B^2*b^12*c^8*f^2 - B^2*a^1
2*d^8*f^2 + 3*A^2*b^12*c^8*f^2 + 3*A^2*a^12*d^8*f^2 - 24*A*B*C*a*b^8*c*d^6*
f + 342*A*B*C*a^4*b^5*c^2*d^5*f - 186*A*B*C*a^5*b^4*c^3*d^4*f - 66*A*B*C*a^
2*b^7*c^4*d^3*f + 48*A*B*C*a^2*b^7*c^2*d^5*f + 42*A*B*C*a^6*b^3*c^2*d^5*f +
 26*A*B*C*a^3*b^6*c^5*d^2*f + 24*A*B*C*a^6*b^3*c^4*d^3*f - 18*A*B*C*a^7*b^2
*c^3*d^4*f - 18*A*B*C*a^4*b^5*c^4*d^3*f - 8*A*B*C*a^3*b^6*c^3*d^4*f + 6*A*B
*C*a^5*b^4*c^5*d^2*f - 128*A*B*C*a^3*b^6*c*d^6*f + 126*A*B*C*a^7*b^2*c*d^6*
f + 72*A*B*C*a*b^8*c^3*d^4*f - 36*A*B*C*a^8*b*c^2*d^5*f - 36*A*B*C*a*b^8*c^
5*d^2*f + 30*A*B*C*a^2*b^7*c^6*d*f - 12*A*B*C*a^5*b^4*c*d^6*f - 12*A*B*C*a^
4*b^5*c^6*d*f - 21*B^2*C*a^8*b*c*d^6*f - 3*B^2*C*a*b^8*c^6*d*f + 21*A^2*C*a
^8*b*c*d^6*f - 21*A*C^2*a^8*b*c*d^6*f - 9*A^2*C*a*b^8*c^6*d*f + 9*A*C^2*a*b
^8*c^6*d*f + 36*A^2*B*a*b^8*c*d^6*f + 21*A*B^2*a^8*b*c*d^6*f + 3*A*B^2*a*b^
8*c^6*d*f + 16*A*B*C*b^9*c^4*d^3*f - 16*A*B*C*b^9*c^2*d^5*f - 78*A*B*C*a^6*
b^3*d^7*f + 24*A*B*C*a^4*b^5*d^7*f + 2*A*B*C*a^3*b^6*c^7*f - 237*B^2*C*a^4*
b^5*c^3*d^4*f + 165*B*C^2*a^5*b^4*c^3*d^4*f + 92*B^2*C*a^3*b^6*c^2*d^5*f -
81*B^2*C*a^7*b^2*c^2*d^5*f + 77*B^2*C*a^3*b^6*c^4*d^3*f - 75*B*C^2*a^4*b^5*
c^2*d^5*f + 69*B^2*C*a^5*b^4*c^4*d^3*f + 69*B*C^2*a^4*b^5*c^4*d^3*f - 68*B*
C^2*a^3*b^6*c^3*d^4*f - 63*B^2*C*a^4*b^5*c^5*d^2*f - 61*B*C^2*a^6*b^3*c^2*d
^5*f + 57*B*C^2*a^2*b^7*c^4*d^3*f - 53*B*C^2*a^3*b^6*c^5*d^2*f - 44*B*C^2*a
^6*b^3*c^4*d^3*f - 36*B^2*C*a^2*b^7*c^3*d^4*f + 35*B^2*C*a^6*b^3*c^3*d^4*f
- 33*B^2*C*a^5*b^4*c^2*d^5*f + 33*B^2*C*a^2*b^7*c^5*d^2*f + 33*B*C^2*a^7*b^
2*c^3*d^4*f - 12*B^2*C*a^7*b^2*c^4*d^3*f + 9*B*C^2*a^5*b^4*c^5*d^2*f + 4*B^
2*C*a^6*b^3*c^5*d^2*f + 225*A^2*C*a^5*b^4*c^2*d^5*f - 105*A*C^2*a^5*b^4*c^2
*d^5*f - 99*A^2*C*a^4*b^5*c^3*d^4*f - 81*A^2*C*a^4*b^5*c^5*d^2*f + 67*A^2*C
*a^3*b^6*c^4*d^3*f - 59*A*C^2*a^3*b^6*c^4*d^3*f - 57*A*C^2*a^7*b^2*c^2*d^5*
f + 57*A*C^2*a^2*b^7*c^5*d^2*f + 51*A^2*C*a^5*b^4*c^4*d^3*f + 48*A^2*C*a^2*
b^7*c^3*d^4*f + 45*A*C^2*a^4*b^5*c^5*d^2*f - 35*A^2*C*a^6*b^3*c^3*d^4*f + 3
3*A^2*C*a^7*b^2*c^2*d^5*f - 33*A^2*C*a^2*b^7*c^5*d^2*f + 33*A*C^2*a^5*b^4*c
^4*d^3*f + 27*A*C^2*a^6*b^3*c^3*d^4*f + 24*A*C^2*a^3*b^6*c^2*d^5*f - 24*A*C
^2*a^2*b^7*c^3*d^4*f - 21*A*C^2*a^4*b^5*c^3*d^4*f - 16*A^2*C*a^3*b^6*c^2*d^
5*f - 243*A^2*B*a^4*b^5*c^2*d^5*f - 156*A*B^2*a^3*b^6*c^2*d^5*f + 141*A*B^2
*a^4*b^5*c^3*d^4*f + 108*A^2*B*a^3*b^6*c^3*d^4*f - 105*A*B^2*a^3*b^6*c^4*d^
3*f + 84*A*B^2*a^2*b^7*c^3*d^4*f + 81*A*B^2*a^5*b^4*c^2*d^5*f + 51*A^2*B*a^
6*b^3*c^2*d^5*f - 51*A^2*B*a^4*b^5*c^4*d^3*f - 48*A^2*B*a^2*b^7*c^2*d^5*f +
 45*A^2*B*a^5*b^4*c^3*d^4*f + 39*A*B^2*a^4*b^5*c^5*d^2*f - 35*A*B^2*a^6*b^3
*c^3*d^4*f + 33*A*B^2*a^7*b^2*c^2*d^5*f + 27*A^2*B*a^3*b^6*c^5*d^2*f - 21*A
*B^2*a^5*b^4*c^4*d^3*f + 20*A^2*B*a^6*b^3*c^4*d^3*f - 15*A^2*B*a^7*b^2*c^3*
d^4*f - 15*A^2*B*a^5*b^4*c^5*d^2*f + 9*A^2*B*a^2*b^7*c^4*d^3*f + 3*A*B^2*a^
2*b^7*c^5*d^2*f + 2*A*B*C*b^9*c^6*d*f - 6*A*B*C*a^9*c*d^6*f + 18*A*B*C*a^8*
b*d^7*f - 6*A*B*C*a*b^8*c^7*f + 63*B^2*C*a^6*b^3*c*d^6*f - 48*B^2*C*a*b^8*c
^4*d^3*f + 42*B*C^2*a^8*b*c^2*d^5*f + 42*B*C^2*a^5*b^4*c*d^6*f - 39*B*C^2*a
^7*b^2*c*d^6*f + 30*B*C^2*a*b^8*c^5*d^2*f - 24*B^2*C*a^4*b^5*c*d^6*f - 24*B
*C^2*a*b^8*c^3*d^4*f + 17*B^2*C*a^3*b^6*c^6*d*f - 15*B*C^2*a^2*b^7*c^6*d*f
+ 12*B^2*C*a^8*b*c^3*d^4*f + 12*B^2*C*a*b^8*c^2*d^5*f + 6*B*C^2*a^4*b^5*c^6
*d*f - 192*A^2*C*a^4*b^5*c*d^6*f - 99*A^2*C*a^6*b^3*c*d^6*f + 84*A*C^2*a^4*

b^5*c*d^6*f + 59*A*C^2*a^6*b^3*c*d^6*f + 51*A^2*C*a^3*b^6*c^6*d*f - 51*A*C^
2*a^3*b^6*c^6*d*f - 36*A^2*C*a*b^8*c^2*d^5*f - 24*A*C^2*a*b^8*c^4*d^3*f + 2
4*A*C^2*a*b^8*c^2*d^5*f + 12*A^2*C*a*b^8*c^4*d^3*f + 12*A*C^2*a^8*b*c^3*d^4
*f + 160*A^2*B*a^3*b^6*c*d^6*f - 99*A*B^2*a^6*b^3*c*d^6*f - 87*A^2*B*a^7*b^
2*c*d^6*f - 72*A*B^2*a^4*b^5*c*d^6*f - 48*A*B^2*a*b^8*c^2*d^5*f - 36*A^2*B*
a*b^8*c^3*d^4*f + 24*A*B^2*a*b^8*c^4*d^3*f - 17*A*B^2*a^3*b^6*c^6*d*f - 15*
A^2*B*a^2*b^7*c^6*d*f + 12*A*B^2*a^2*b^7*c*d^6*f + 6*A^2*B*a^8*b*c^2*d^5*f
- 6*A^2*B*a^5*b^4*c*d^6*f + 6*A^2*B*a^4*b^5*c^6*d*f + 6*A^2*B*a*b^8*c^5*d^2
*f + 12*B^2*C*b^9*c^3*d^4*f - 12*B*C^2*b^9*c^4*d^3*f - 12*A^2*C*b^9*c^3*d^4
*f - 8*A*C^2*b^9*c^5*d^2*f + 8*A*C^2*b^9*c^3*d^4*f + 4*B^2*C*a^9*c^2*d^5*f
+ 4*A^2*C*b^9*c^5*d^2*f - 4*B*C^2*a^9*c^3*d^4*f + 12*A^2*B*b^9*c^2*d^5*f -
8*A*B^2*b^9*c^3*d^4*f - 4*A^2*B*b^9*c^4*d^3*f + 4*A*C^2*a^9*c^2*d^5*f + 3*B
^2*C*a^7*b^2*d^7*f - B*C^2*a^6*b^3*d^7*f + 96*A^2*C*a^5*b^4*d^7*f - 39*A^2*
C*a^7*b^2*d^7*f - 36*A*C^2*a^5*b^4*d^7*f + 32*A^2*C*a^3*b^6*d^7*f + 15*A*C^
2*a^7*b^2*d^7*f - 3*B^2*C*a^2*b^7*c^7*f - B*C^2*a^3*b^6*c^7*f + 111*A^2*B*a
^6*b^3*d^7*f - 39*A*B^2*a^7*b^2*d^7*f + 24*A*B^2*a^5*b^4*d^7*f - 9*A^2*C*a^
2*b^7*c^7*f + 9*A*C^2*a^2*b^7*c^7*f - 4*A*B^2*a^3*b^6*d^7*f + 3*A*B^2*a^2*b
^7*c^7*f - A^2*B*a^3*b^6*c^7*f + 3*C^3*a^8*b*c*d^6*f - 3*C^3*a*b^8*c^6*d*f
- 3*A^3*a^8*b*c*d^6*f + 3*A^3*a*b^8*c^6*d*f - B*C^2*b^9*c^6*d*f + 4*A^2*C*b
^9*c*d^6*f + 3*B*C^2*a^9*c*d^6*f + 8*A*B^2*b^9*c*d^6*f + 3*B*C^2*a^8*b*d^7*
f - A^2*B*b^9*c^6*d*f + 12*A^2*C*a*b^8*d^7*f + 3*B*C^2*a*b^8*c^7*f - A^2*B*
a^9*c*d^6*f - 9*A^2*B*a^8*b*d^7*f + 3*A^2*B*a*b^8*c^7*f - 39*C^3*a^5*b^4*c^
4*d^3*f + 39*C^3*a^4*b^5*c^3*d^4*f + 27*C^3*a^7*b^2*c^2*d^5*f - 27*C^3*a^2*
b^7*c^5*d^2*f - 17*C^3*a^6*b^3*c^3*d^4*f + 17*C^3*a^3*b^6*c^4*d^3*f + 3*C^3
*a^5*b^4*c^2*d^5*f - 3*C^3*a^4*b^5*c^5*d^2*f - 63*B^3*a^5*b^4*c^3*d^4*f + 5
7*B^3*a^4*b^5*c^2*d^5*f - 51*B^3*a^2*b^7*c^4*d^3*f + 48*B^3*a^3*b^6*c^3*d^4
*f + 31*B^3*a^6*b^3*c^2*d^5*f + 27*B^3*a^3*b^6*c^5*d^2*f + 16*B^3*a^6*b^3*c
^4*d^3*f - 15*B^3*a^5*b^4*c^5*d^2*f - 12*B^3*a^2*b^7*c^2*d^5*f + 9*B^3*a^4*
b^5*c^4*d^3*f - 3*B^3*a^7*b^2*c^3*d^4*f - 123*A^3*a^5*b^4*c^2*d^5*f + 81*A^
3*a^4*b^5*c^3*d^4*f - 45*A^3*a^5*b^4*c^4*d^3*f + 39*A^3*a^4*b^5*c^5*d^2*f +
25*A^3*a^6*b^3*c^3*d^4*f - 25*A^3*a^3*b^6*c^4*d^3*f - 24*A^3*a^2*b^7*c^3*d
^4*f - 8*A^3*a^3*b^6*c^2*d^5*f - 3*A^3*a^7*b^2*c^2*d^5*f + 3*A^3*a^2*b^7*c^
5*d^2*f - 17*C^3*a^6*b^3*c*d^6*f + 17*C^3*a^3*b^6*c^6*d*f - 12*C^3*a^8*b*c^
3*d^4*f + 12*C^3*a*b^8*c^4*d^3*f + 24*B^3*a*b^8*c^3*d^4*f + 21*B^3*a^7*b^2*
c*d^6*f - 18*B^3*a^5*b^4*c*d^6*f - 15*B^3*a^2*b^7*c^6*d*f - 6*B^3*a^8*b*c^2
*d^5*f + 6*B^3*a^4*b^5*c^6*d*f + 6*B^3*a*b^8*c^5*d^2*f + 4*B^3*a^3*b^6*c*d^
6*f + 108*A^3*a^4*b^5*c*d^6*f + 57*A^3*a^6*b^3*c*d^6*f - 17*A^3*a^3*b^6*c^6
*d*f + 12*A^3*a*b^8*c^2*d^5*f + 4*C^3*b^9*c^5*d^2*f - 4*C^3*a^9*c^2*d^5*f -
4*B^3*b^9*c^2*d^5*f + 4*A^3*b^9*c^3*d^4*f + 3*C^3*a^7*b^2*d^7*f - 3*C^3*a^
2*b^7*c^7*f - B^3*a^6*b^3*d^7*f - 60*A^3*a^5*b^4*d^7*f - 32*A^3*a^3*b^6*d^7
*f + 21*A^3*a^7*b^2*d^7*f - B^3*a^3*b^6*c^7*f + 3*A^3*a^2*b^7*c^7*f - B^3*b
^9*c^6*d*f - 4*A^3*b^9*c*d^6*f - B^3*a^9*c*d^6*f + 3*B^3*a^8*b*d^7*f - 12*A
^3*a*b^8*d^7*f + 3*B^3*a*b^8*c^7*f - B^2*C*a^9*d^7*f - 4*A^2*B*b^9*d^7*f +
3*A^2*C*b^9*c^7*f - 3*A*C^2*b^9*c^7*f - A*C^2*a^9*d^7*f - A*B^2*b^9*c^7*f -
C^3*a^9*d^7*f - A^3*b^9*c^7*f + B^2*C*b^9*c^7*f + A^2*C*a^9*d^7*f + A*B^2*

a^9*d^7*f + C^3*b^9*c^7*f + A^3*a^9*d^7*f - 6*A*B^2*C*a^5*b*c*d^5 - 21*A^2*
B*C*a^3*b^3*c^2*d^4 + 21*A*B*C^2*a^3*b^3*c^2*d^4 + 12*A*B^2*C*a^4*b^2*c^2*d
^4 - 12*A*B^2*C*a^2*b^4*c^2*d^4 - 10*A*B^2*C*a^3*b^3*c^3*d^3 - 6*A*B*C^2*a^
4*b^2*c^3*d^3 + 3*A^2*B*C*a^4*b^2*c^3*d^3 + 3*A^2*B*C*a^2*b^4*c^3*d^3 + 3*A
*B^2*C*a^2*b^4*c^4*d^2 + 3*A*B*C^2*a^2*b^4*c^3*d^3 + 2*A*B*C^2*a^3*b^3*c^4*
d^2 - A^2*B*C*a^3*b^3*c^4*d^2 + 18*A^2*B*C*a^2*b^4*c*d^5 + 10*A*B^2*C*a^3*b
^3*c*d^5 + 9*A^2*B*C*a^4*b^2*c*d^5 - 9*A*B*C^2*a^4*b^2*c*d^5 - 9*A*B*C^2*a^
2*b^4*c*d^5 - 6*A^2*B*C*a*b^5*c^2*d^4 + 6*A*B^2*C*a*b^5*c^3*d^3 + 6*A*B*C^2
*a^5*b*c^2*d^4 - 6*A*B*C^2*a*b^5*c^4*d^2 - 3*A^2*B*C*a^5*b*c^2*d^4 + 3*A^2*
B*C*a*b^5*c^4*d^2 + 3*A*B*C^2*a*b^5*c^2*d^4 - 3*B^3*C*a^5*b*c^2*d^4 + 3*B^3
*C*a^4*b^2*c*d^5 + 3*B^3*C*a*b^5*c^4*d^2 + 3*B^2*C^2*a^5*b*c*d^5 - 3*B*C^3*
a^5*b*c^2*d^4 + 3*B*C^3*a^4*b^2*c*d^5 + 3*B*C^3*a*b^5*c^4*d^2 + 24*A^3*C*a^
3*b^3*c*d^5 + 8*A*C^3*a^3*b^3*c*d^5 - 9*A^3*B*a^2*b^4*c*d^5 - 9*A*B^3*a^2*b
^4*c*d^5 - 3*A^3*B*a^4*b^2*c*d^5 + 3*A^3*B*a*b^5*c^2*d^4 + 3*A^2*B^2*a^5*b*
c*d^5 - 3*A*B^3*a^4*b^2*c*d^5 + 3*A*B^3*a*b^5*c^2*d^4 + 5*A*B*C^2*b^6*c^3*d
^3 - 4*A^2*B*C*b^6*c^3*d^3 - A*B^2*C*b^6*c^4*d^2 - 3*A*B^2*C*a^4*b^2*d^6 -
2*A^2*B*C*a^3*b^3*d^6 + 9*B^2*C^2*a^3*b^3*c^3*d^3 - 6*B^2*C^2*a^4*b^2*c^2*d
^4 + 6*B^2*C^2*a^2*b^4*c^2*d^4 - 3*B^2*C^2*a^2*b^4*c^4*d^2 + 24*A^2*C^2*a^3
*b^3*c^3*d^3 - 15*A^2*C^2*a^4*b^2*c^2*d^4 - 9*A^2*C^2*a^2*b^4*c^4*d^2 + 3*A
^2*C^2*a^2*b^4*c^2*d^4 + 9*A^2*B^2*a^2*b^4*c^2*d^4 - 3*A^2*B^2*a^4*b^2*c^2*
d^4 + 4*A^2*B*C*b^6*c*d^5 - 2*A*B*C^2*b^6*c*d^5 + 2*A*B*C^2*a^6*c*d^5 - A^2
*B*C*a^6*c*d^5 + 6*A^2*B*C*a^5*b*d^6 - 3*A*B*C^2*a^5*b*d^6 - 7*B^3*C*a^3*b^
3*c^2*d^4 - 7*B*C^3*a^3*b^3*c^2*d^4 + 3*B^3*C*a^4*b^2*c^3*d^3 - 3*B^3*C*a^2
*b^4*c^3*d^3 - 3*B^2*C^2*a*b^5*c^3*d^3 + 3*B*C^3*a^4*b^2*c^3*d^3 - 3*B*C^3*
a^2*b^4*c^3*d^3 - B^3*C*a^3*b^3*c^4*d^2 - B^2*C^2*a^3*b^3*c*d^5 - B*C^3*a^3
*b^3*c^4*d^2 - 24*A^2*C^2*a^3*b^3*c*d^5 - 24*A*C^3*a^3*b^3*c^3*d^3 + 12*A*C
^3*a^4*b^2*c^2*d^4 + 9*A*C^3*a^2*b^4*c^4*d^2 - 8*A^3*C*a^3*b^3*c^3*d^3 + 6*
A^3*C*a^4*b^2*c^2*d^4 - 6*A^3*C*a^2*b^4*c^2*d^4 + 3*A^3*C*a^2*b^4*c^4*d^2 -
9*A^2*B^2*a^3*b^3*c*d^5 + 7*A^3*B*a^3*b^3*c^2*d^4 + 7*A*B^3*a^3*b^3*c^2*d^
4 - 3*A^3*B*a^2*b^4*c^3*d^3 - 3*A^2*B^2*a*b^5*c^3*d^3 - 3*A*B^3*a^2*b^4*c^3
*d^3 - 5*A^2*C^2*b^6*c^2*d^4 + 3*A^2*C^2*b^6*c^4*d^2 + 12*A^2*C^2*a^4*b^2*d
^6 + 3*A^2*C^2*a^2*b^4*d^6 + 6*A^2*B^2*a^4*b^2*d^6 + 3*A^2*B^2*a^2*b^4*d^6
+ A*B*C^2*a^3*b^3*d^6 - 3*B^4*a*b^5*c^3*d^3 - B^4*a^3*b^3*c*d^5 + A^2*B^2*a
^3*b^3*c^3*d^3 - 8*A^4*a^3*b^3*c*d^5 - 2*B^3*C*b^6*c^3*d^3 - 2*B*C^3*b^6*c^
3*d^3 + 4*A^3*C*b^6*c^2*d^4 - 3*A*C^3*b^6*c^4*d^2 + 2*A*C^3*b^6*c^2*d^4 - A
^3*C*b^6*c^4*d^2 - 2*A*C^3*a^6*c^2*d^4 - 15*A^3*C*a^4*b^2*d^6 - 6*A^3*C*a^2
*b^4*d^6 - 3*A*C^3*a^4*b^2*d^6 + 3*B^4*a^5*b*c*d^5 - B^3*C*a^6*c*d^5 - B*C^
3*a^6*c*d^5 - 2*A^3*B*b^6*c*d^5 - 2*A*B^3*b^6*c*d^5 - 3*A^3*B*a^5*b*d^6 - 3
*A*B^3*a^5*b*d^6 + 8*C^4*a^3*b^3*c^3*d^3 - 3*C^4*a^4*b^2*c^2*d^4 - 3*C^4*a^
2*b^4*c^4*d^2 + 6*B^4*a^2*b^4*c^2*d^4 - 3*B^4*a^4*b^2*c^2*d^4 + 3*A^4*a^2*b
^4*c^2*d^4 + B^2*C^2*b^6*c^4*d^2 + B^2*C^2*b^6*c^2*d^4 + B^2*C^2*a^6*c^2*d^
4 + A^2*C^2*a^6*c^2*d^4 - 2*A^3*C*b^6*d^6 + A^3*B*b^6*c^3*d^3 + A*B^3*b^6*c
^3*d^3 + A^3*B*a^3*b^3*d^6 + A*B^3*a^3*b^3*d^6 - A^4*b^6*c^2*d^4 + 6*A^4*a^
4*b^2*d^6 + 3*A^4*a^2*b^4*d^6 - 2*A^2*C^2*a^6*d^6 + A*B^2*C*a^6*d^6 + B^4*a
^3*b^3*c^3*d^3 + A^3*C*a^6*d^6 + A*C^3*a^6*d^6 + C^4*b^6*c^4*d^2 + C^4*a^6*

c^2*d^4 + B^4*b^6*c^2*d^4 + A^2*C^2*b^6*d^6 + A^2*B^2*b^6*d^6 + A^4*b^6*d^6
, f, k)*((B*b^14*c^7*d - B*a^13*b*d^8 - 4*A*a^2*b^12*d^8 - 16*A*a^4*b^10*d^
8 - 35*A*a^6*b^8*d^8 - 33*A*a^8*b^6*d^8 - 5*A*a^10*b^4*d^8 + 5*A*a^12*b^2*d
^8 - 4*B*a^5*b^9*d^8 + 3*B*a^7*b^7*d^8 + 17*B*a^9*b^5*d^8 + 9*B*a^11*b^3*d^
8 - 4*A*b^14*c^2*d^6 + 4*A*b^14*c^4*d^4 - 3*A*b^14*c^6*d^2 + 11*C*a^6*b^8*d
^8 + 17*C*a^8*b^6*d^8 + C*a^10*b^4*d^8 - 5*C*a^12*b^2*d^8 + 4*B*b^14*c^3*d^
5 - 4*B*b^14*c^5*d^3 - 4*C*b^14*c^4*d^4 + 3*C*b^14*c^6*d^2 - 6*A*a*b^13*c^5
*d^3 + 40*A*a^3*b^11*c*d^7 + 3*A*a^3*b^11*c^7*d + 122*A*a^5*b^9*c*d^7 + 3*A
*a^5*b^9*c^7*d + 175*A*a^7*b^7*c*d^7 + A*a^7*b^7*c^7*d + 105*A*a^9*b^5*c*d^
7 + 21*A*a^11*b^3*c*d^7 - 8*B*a*b^13*c^2*d^6 - 4*B*a*b^13*c^4*d^4 + 5*B*a*b
^13*c^6*d^2 + 4*B*a^2*b^12*c*d^7 + 3*B*a^2*b^12*c^7*d + 32*B*a^4*b^10*c*d^7
 + 3*B*a^4*b^10*c^7*d + 31*B*a^6*b^8*c*d^7 + B*a^6*b^8*c^7*d - 27*B*a^8*b^6
*c*d^7 - 39*B*a^10*b^4*c*d^7 - 9*B*a^12*b^2*c*d^7 + 8*C*a*b^13*c^3*d^5 + 10
*C*a*b^13*c^5*d^3 - 3*C*a^3*b^11*c^7*d - 38*C*a^5*b^9*c*d^7 - 3*C*a^5*b^9*c
^7*d - 79*C*a^7*b^7*c*d^7 - C*a^7*b^7*c^7*d - 41*C*a^9*b^5*c*d^7 + 3*C*a^11
*b^3*c*d^7 - 28*A*a^2*b^12*c^2*d^6 + 43*A*a^2*b^12*c^4*d^4 + A*a^2*b^12*c^6
*d^2 - 4*A*a^3*b^11*c^3*d^5 - 35*A*a^3*b^11*c^5*d^3 - 117*A*a^4*b^10*c^2*d^
6 + 69*A*a^4*b^10*c^4*d^4 + 5*A*a^4*b^10*c^6*d^2 + 67*A*a^5*b^9*c^3*d^5 - 3
7*A*a^5*b^9*c^5*d^3 - 245*A*a^6*b^8*c^2*d^6 + 5*A*a^6*b^8*c^4*d^4 - 5*A*a^6
*b^8*c^6*d^2 + 161*A*a^7*b^7*c^3*d^5 + 7*A*a^7*b^7*c^5*d^3 - 237*A*a^8*b^6*
c^2*d^6 - 45*A*a^8*b^6*c^4*d^4 - 6*A*a^8*b^6*c^6*d^2 + 105*A*a^9*b^5*c^3*d^
5 + 15*A*a^9*b^5*c^5*d^3 - 91*A*a^10*b^4*c^2*d^6 - 20*A*a^10*b^4*c^4*d^4 +
15*A*a^11*b^3*c^3*d^5 - 6*A*a^12*b^2*c^2*d^6 + 44*B*a^2*b^12*c^3*d^5 - 11*B
*a^2*b^12*c^5*d^3 - 64*B*a^3*b^11*c^2*d^6 - 71*B*a^3*b^11*c^4*d^4 - B*a^3*b
^11*c^6*d^2 + 187*B*a^4*b^10*c^3*d^5 + 23*B*a^4*b^10*c^5*d^3 - 145*B*a^5*b^
9*c^2*d^6 - 173*B*a^5*b^9*c^4*d^4 - 17*B*a^5*b^9*c^6*d^2 + 273*B*a^6*b^8*c^
3*d^5 + 63*B*a^6*b^8*c^5*d^3 - 115*B*a^7*b^7*c^2*d^6 - 149*B*a^7*b^7*c^4*d^
4 - 11*B*a^7*b^7*c^6*d^2 + 141*B*a^8*b^6*c^3*d^5 + 33*B*a^8*b^6*c^5*d^3 - 1
1*B*a^9*b^5*c^2*d^6 - 43*B*a^9*b^5*c^4*d^4 + 15*B*a^10*b^4*c^3*d^5 + 15*B*a
^11*b^3*c^2*d^6 - 4*C*a^2*b^12*c^2*d^6 - 47*C*a^2*b^12*c^4*d^4 - C*a^2*b^12
*c^6*d^2 + 36*C*a^3*b^11*c^3*d^5 + 51*C*a^3*b^11*c^5*d^3 + 25*C*a^4*b^10*c^
2*d^6 - 85*C*a^4*b^10*c^4*d^4 - 5*C*a^4*b^10*c^6*d^2 - 19*C*a^5*b^9*c^3*d^5
 + 61*C*a^5*b^9*c^5*d^3 + 117*C*a^6*b^8*c^2*d^6 - 29*C*a^6*b^8*c^4*d^4 + 5*
C*a^6*b^8*c^6*d^2 - 129*C*a^7*b^7*c^3*d^5 + 9*C*a^7*b^7*c^5*d^3 + 145*C*a^8
*b^6*c^2*d^6 + 29*C*a^8*b^6*c^4*d^4 + 6*C*a^8*b^6*c^6*d^2 - 97*C*a^9*b^5*c^
3*d^5 - 11*C*a^9*b^5*c^5*d^3 + 59*C*a^10*b^4*c^2*d^6 + 16*C*a^10*b^4*c^4*d^
4 - 15*C*a^11*b^3*c^3*d^5 + 2*C*a^12*b^2*c^2*d^6 + 8*A*a*b^13*c*d^7 + A*a*b
^13*c^7*d + A*a^13*b*c*d^7 - C*a*b^13*c^7*d + 3*C*a^13*b*c*d^7)/(a^12*d^4 +
 b^12*c^4 + 4*a^2*b^10*c^4 + 6*a^4*b^8*c^4 + 4*a^6*b^6*c^4 + a^8*b^4*c^4 +
a^4*b^8*d^4 + 4*a^6*b^6*d^4 + 6*a^8*b^4*d^4 + 4*a^10*b^2*d^4 - 4*a^3*b^9*c*
d^3 - 16*a^3*b^9*c^3*d - 16*a^5*b^7*c*d^3 - 24*a^5*b^7*c^3*d - 24*a^7*b^5*c
*d^3 - 16*a^7*b^5*c^3*d - 16*a^9*b^3*c*d^3 - 4*a^9*b^3*c^3*d + 6*a^2*b^10*c
^2*d^2 + 24*a^4*b^8*c^2*d^2 + 36*a^6*b^6*c^2*d^2 + 24*a^8*b^4*c^2*d^2 + 6*a
^10*b^2*c^2*d^2 - 4*a*b^11*c^3*d - 4*a^11*b*c*d^3) + root(480*a^11*b^7*c*d^
9*f^4 + 480*a^7*b^11*c^9*d*f^4 + 360*a^13*b^5*c*d^9*f^4 + 360*a^9*b^9*c^9*d

*f^4 + 360*a^9*b^9*c*d^9*f^4 + 360*a^5*b^13*c^9*d*f^4 + 144*a^15*b^3*c*d^9*
f^4 + 144*a^11*b^7*c^9*d*f^4 + 144*a^7*b^11*c*d^9*f^4 + 144*a^3*b^15*c^9*d*
f^4 + 48*a^17*b*c^3*d^7*f^4 + 48*a*b^17*c^7*d^3*f^4 + 24*a^17*b*c^5*d^5*f^4
 + 24*a^13*b^5*c^9*d*f^4 + 24*a^5*b^13*c*d^9*f^4 + 24*a*b^17*c^5*d^5*f^4 +
24*a^17*b*c*d^9*f^4 + 24*a*b^17*c^9*d*f^4 + 3920*a^9*b^9*c^5*d^5*f^4 - 3360
*a^10*b^8*c^4*d^6*f^4 - 3360*a^8*b^10*c^6*d^4*f^4 + 3024*a^11*b^7*c^5*d^5*f
^4 - 3024*a^10*b^8*c^6*d^4*f^4 - 3024*a^8*b^10*c^4*d^6*f^4 + 3024*a^7*b^11*
c^5*d^5*f^4 + 2320*a^9*b^9*c^7*d^3*f^4 + 2320*a^9*b^9*c^3*d^7*f^4 - 2240*a^
12*b^6*c^4*d^6*f^4 - 2240*a^6*b^12*c^6*d^4*f^4 + 2160*a^11*b^7*c^3*d^7*f^4
+ 2160*a^7*b^11*c^7*d^3*f^4 - 1624*a^12*b^6*c^6*d^4*f^4 - 1624*a^6*b^12*c^4
*d^6*f^4 + 1488*a^11*b^7*c^7*d^3*f^4 + 1488*a^7*b^11*c^3*d^7*f^4 + 1344*a^1
3*b^5*c^5*d^5*f^4 + 1344*a^5*b^13*c^5*d^5*f^4 - 1320*a^10*b^8*c^2*d^8*f^4 -
 1320*a^8*b^10*c^8*d^2*f^4 + 1200*a^13*b^5*c^3*d^7*f^4 + 1200*a^5*b^13*c^7*
d^3*f^4 - 1060*a^12*b^6*c^2*d^8*f^4 - 1060*a^6*b^12*c^8*d^2*f^4 - 948*a^10*
b^8*c^8*d^2*f^4 - 948*a^8*b^10*c^2*d^8*f^4 - 840*a^14*b^4*c^4*d^6*f^4 - 840
*a^4*b^14*c^6*d^4*f^4 + 528*a^13*b^5*c^7*d^3*f^4 + 528*a^5*b^13*c^3*d^7*f^4
 - 480*a^14*b^4*c^6*d^4*f^4 - 480*a^14*b^4*c^2*d^8*f^4 - 480*a^4*b^14*c^8*d
^2*f^4 - 480*a^4*b^14*c^4*d^6*f^4 + 368*a^15*b^3*c^3*d^7*f^4 - 368*a^12*b^6
*c^8*d^2*f^4 - 368*a^6*b^12*c^2*d^8*f^4 + 368*a^3*b^15*c^7*d^3*f^4 + 304*a^
15*b^3*c^5*d^5*f^4 + 304*a^3*b^15*c^5*d^5*f^4 - 144*a^16*b^2*c^4*d^6*f^4 -
144*a^2*b^16*c^6*d^4*f^4 - 108*a^16*b^2*c^2*d^8*f^4 - 108*a^2*b^16*c^8*d^2*
f^4 + 80*a^15*b^3*c^7*d^3*f^4 + 80*a^3*b^15*c^3*d^7*f^4 - 60*a^16*b^2*c^6*d
^4*f^4 - 60*a^14*b^4*c^8*d^2*f^4 - 60*a^4*b^14*c^2*d^8*f^4 - 60*a^2*b^16*c^
4*d^6*f^4 - 8*b^18*c^8*d^2*f^4 - 4*b^18*c^6*d^4*f^4 - 8*a^18*c^2*d^8*f^4 -
4*a^18*c^4*d^6*f^4 - 80*a^12*b^6*d^10*f^4 - 60*a^14*b^4*d^10*f^4 - 60*a^10*
b^8*d^10*f^4 - 24*a^16*b^2*d^10*f^4 - 24*a^8*b^10*d^10*f^4 - 4*a^6*b^12*d^1
0*f^4 - 80*a^6*b^12*c^10*f^4 - 60*a^8*b^10*c^10*f^4 - 60*a^4*b^14*c^10*f^4
- 24*a^10*b^8*c^10*f^4 - 24*a^2*b^16*c^10*f^4 - 4*a^12*b^6*c^10*f^4 - 4*b^1
8*c^10*f^4 - 4*a^18*d^10*f^4 - 12*A*C*a^11*b*c*d^7*f^2 - 12*A*C*a*b^11*c^7*
d*f^2 - 912*B*C*a^5*b^7*c^4*d^4*f^2 - 792*B*C*a^8*b^4*c^3*d^5*f^2 + 792*B*C
*a^4*b^8*c^5*d^3*f^2 + 720*B*C*a^7*b^5*c^4*d^4*f^2 - 480*B*C*a^5*b^7*c^6*d^
2*f^2 - 408*B*C*a^5*b^7*c^2*d^6*f^2 + 384*B*C*a^7*b^5*c^2*d^6*f^2 - 336*B*C
*a^8*b^4*c^5*d^3*f^2 + 324*B*C*a^4*b^8*c^3*d^5*f^2 + 312*B*C*a^7*b^5*c^6*d^
2*f^2 - 248*B*C*a^3*b^9*c^6*d^2*f^2 + 216*B*C*a^9*b^3*c^2*d^6*f^2 - 196*B*C
*a^3*b^9*c^4*d^4*f^2 + 132*B*C*a^9*b^3*c^4*d^4*f^2 + 80*B*C*a^6*b^6*c^3*d^5
*f^2 - 64*B*C*a^6*b^6*c^5*d^3*f^2 - 36*B*C*a^2*b^10*c^3*d^5*f^2 - 28*B*C*a^
3*b^9*c^2*d^6*f^2 + 12*B*C*a^10*b^2*c^5*d^3*f^2 - 12*B*C*a^10*b^2*c^3*d^5*f
^2 - 12*B*C*a^2*b^10*c^5*d^3*f^2 - 4*B*C*a^9*b^3*c^6*d^2*f^2 - 1468*A*C*a^6
*b^6*c^4*d^4*f^2 + 996*A*C*a^7*b^5*c^3*d^5*f^2 + 900*A*C*a^5*b^7*c^5*d^3*f^
2 - 676*A*C*a^6*b^6*c^6*d^2*f^2 - 660*A*C*a^6*b^6*c^2*d^6*f^2 + 636*A*C*a^5
*b^7*c^3*d^5*f^2 + 540*A*C*a^7*b^5*c^5*d^3*f^2 - 236*A*C*a^3*b^9*c^5*d^3*f^
2 - 204*A*C*a^9*b^3*c^3*d^5*f^2 + 156*A*C*a^10*b^2*c^2*d^6*f^2 + 132*A*C*a^
2*b^10*c^6*d^2*f^2 - 72*A*C*a^9*b^3*c^5*d^3*f^2 - 72*A*C*a^4*b^8*c^6*d^2*f^
2 + 66*A*C*a^4*b^8*c^2*d^6*f^2 + 54*A*C*a^10*b^2*c^4*d^4*f^2 + 54*A*C*a^2*b
^10*c^4*d^4*f^2 - 48*A*C*a^8*b^4*c^2*d^6*f^2 - 48*A*C*a^4*b^8*c^4*d^4*f^2 +

42*A*C*a^8*b^4*c^6*d^2*f^2 - 40*A*C*a^3*b^9*c^3*d^5*f^2 - 36*A*C*a^8*b^4*c
^4*d^4*f^2 + 24*A*C*a^2*b^10*c^2*d^6*f^2 + 960*A*B*a^5*b^7*c^4*d^4*f^2 - 86
4*A*B*a^4*b^8*c^5*d^3*f^2 + 756*A*B*a^8*b^4*c^3*d^5*f^2 - 744*A*B*a^7*b^5*c
^4*d^4*f^2 - 528*A*B*a^4*b^8*c^3*d^5*f^2 + 504*A*B*a^5*b^7*c^6*d^2*f^2 - 43
2*A*B*a^7*b^5*c^2*d^6*f^2 + 432*A*B*a^5*b^7*c^2*d^6*f^2 + 348*A*B*a^8*b^4*c
^5*d^3*f^2 - 312*A*B*a^7*b^5*c^6*d^2*f^2 - 284*A*B*a^9*b^3*c^2*d^6*f^2 + 28
0*A*B*a^3*b^9*c^6*d^2*f^2 + 264*A*B*a^3*b^9*c^4*d^4*f^2 - 240*A*B*a^6*b^6*c
^3*d^5*f^2 - 172*A*B*a^9*b^3*c^4*d^4*f^2 + 68*A*B*a^3*b^9*c^2*d^6*f^2 - 60*
A*B*a^2*b^10*c^3*d^5*f^2 + 24*A*B*a^6*b^6*c^5*d^3*f^2 - 24*A*B*a^2*b^10*c^5
*d^3*f^2 + 12*A*B*a^10*b^2*c^3*d^5*f^2 + 360*B*C*a^4*b^8*c^7*d*f^2 - 336*B*
C*a^8*b^4*c*d^7*f^2 + 168*B*C*a^6*b^6*c*d^7*f^2 - 136*B*C*a^6*b^6*c^7*d*f^2
 - 36*B*C*a^11*b*c^2*d^6*f^2 + 36*B*C*a*b^11*c^6*d^2*f^2 + 24*B*C*a^10*b^2*
c*d^7*f^2 - 24*B*C*a^2*b^10*c^7*d*f^2 - 12*B*C*a^11*b*c^4*d^4*f^2 + 12*B*C*
a^4*b^8*c*d^7*f^2 + 12*B*C*a*b^11*c^4*d^4*f^2 + 444*A*C*a^7*b^5*c*d^7*f^2 +
 348*A*C*a^5*b^7*c^7*d*f^2 - 164*A*C*a^3*b^9*c^7*d*f^2 - 132*A*C*a^9*b^3*c*
d^7*f^2 + 84*A*C*a^5*b^7*c*d^7*f^2 + 32*A*C*a^3*b^9*c*d^7*f^2 - 12*A*C*a^11
*b*c^3*d^5*f^2 - 12*A*C*a^7*b^5*c^7*d*f^2 - 12*A*C*a*b^11*c^5*d^3*f^2 - 360
*A*B*a^4*b^8*c^7*d*f^2 + 288*A*B*a^8*b^4*c*d^7*f^2 - 288*A*B*a^6*b^6*c*d^7*
f^2 - 144*A*B*a^4*b^8*c*d^7*f^2 + 136*A*B*a^6*b^6*c^7*d*f^2 - 60*A*B*a^2*b^
10*c*d^7*f^2 - 36*A*B*a^10*b^2*c*d^7*f^2 + 24*A*B*a^2*b^10*c^7*d*f^2 - 24*A
*B*a*b^11*c^6*d^2*f^2 + 12*A*B*a^11*b*c^2*d^6*f^2 + 12*A*B*a*b^11*c^4*d^4*f
^2 + 12*A*B*a*b^11*c^2*d^6*f^2 - 8*B*C*b^12*c^5*d^3*f^2 - 8*B*C*b^12*c^3*d^
5*f^2 + 8*A*C*b^12*c^2*d^6*f^2 - 4*B*C*a^12*c^3*d^5*f^2 + 4*A*C*b^12*c^4*d^
4*f^2 - 2*A*C*b^12*c^6*d^2*f^2 + 80*B*C*a^9*b^3*d^8*f^2 - 24*B*C*a^7*b^5*d^
8*f^2 + 6*A*C*a^12*c^2*d^6*f^2 + 4*A*B*b^12*c^5*d^3*f^2 - 4*A*B*b^12*c^3*d^
5*f^2 - 90*A*C*a^8*b^4*d^8*f^2 - 80*B*C*a^3*b^9*c^8*f^2 + 54*A*C*a^10*b^2*d
^8*f^2 - 30*A*C*a^6*b^6*d^8*f^2 + 24*B*C*a^5*b^7*c^8*f^2 - 12*A*C*a^4*b^8*d
^8*f^2 - 112*A*B*a^9*b^3*d^8*f^2 - 66*A*C*a^4*b^8*c^8*f^2 + 54*A*C*a^2*b^10
*c^8*f^2 + 4*A*B*a^3*b^9*d^8*f^2 + 2*A*C*a^6*b^6*c^8*f^2 + 80*A*B*a^3*b^9*c
^8*f^2 - 24*A*B*a^5*b^7*c^8*f^2 + 726*C^2*a^6*b^6*c^4*d^4*f^2 - 402*C^2*a^7
*b^5*c^3*d^5*f^2 - 402*C^2*a^5*b^7*c^5*d^3*f^2 + 322*C^2*a^6*b^6*c^6*d^2*f^
2 + 322*C^2*a^6*b^6*c^2*d^6*f^2 - 222*C^2*a^7*b^5*c^5*d^3*f^2 - 222*C^2*a^5
*b^7*c^3*d^5*f^2 + 134*C^2*a^9*b^3*c^3*d^5*f^2 + 134*C^2*a^3*b^9*c^5*d^3*f^
2 - 66*C^2*a^10*b^2*c^2*d^6*f^2 - 66*C^2*a^2*b^10*c^6*d^2*f^2 + 52*C^2*a^9*
b^3*c^5*d^3*f^2 + 52*C^2*a^3*b^9*c^3*d^5*f^2 - 27*C^2*a^8*b^4*c^6*d^2*f^2 -
 27*C^2*a^4*b^8*c^2*d^6*f^2 + 24*C^2*a^8*b^4*c^4*d^4*f^2 + 24*C^2*a^8*b^4*c
^2*d^6*f^2 + 24*C^2*a^4*b^8*c^6*d^2*f^2 + 24*C^2*a^4*b^8*c^4*d^4*f^2 - 15*C
^2*a^10*b^2*c^4*d^4*f^2 - 15*C^2*a^2*b^10*c^4*d^4*f^2 - 570*B^2*a^6*b^6*c^4
*d^4*f^2 + 366*B^2*a^7*b^5*c^3*d^5*f^2 + 318*B^2*a^5*b^7*c^5*d^3*f^2 - 262*
B^2*a^6*b^6*c^6*d^2*f^2 - 222*B^2*a^6*b^6*c^2*d^6*f^2 - 210*B^2*a^3*b^9*c^5
*d^3*f^2 + 186*B^2*a^7*b^5*c^5*d^3*f^2 + 162*B^2*a^5*b^7*c^3*d^5*f^2 - 142*
B^2*a^9*b^3*c^3*d^5*f^2 + 132*B^2*a^4*b^8*c^4*d^4*f^2 + 117*B^2*a^4*b^8*c^2
*d^6*f^2 + 102*B^2*a^2*b^10*c^6*d^2*f^2 - 96*B^2*a^3*b^9*c^3*d^5*f^2 + 90*B
^2*a^10*b^2*c^2*d^6*f^2 + 81*B^2*a^2*b^10*c^4*d^4*f^2 - 56*B^2*a^9*b^3*c^5*
d^3*f^2 + 48*B^2*a^8*b^4*c^4*d^4*f^2 + 48*B^2*a^4*b^8*c^6*d^2*f^2 + 45*B^2*

a^8*b^4*c^6*d^2*f^2 + 36*B^2*a^8*b^4*c^2*d^6*f^2 + 36*B^2*a^2*b^10*c^2*d^6*
f^2 + 33*B^2*a^10*b^2*c^4*d^4*f^2 + 822*A^2*a^6*b^6*c^4*d^4*f^2 - 594*A^2*a
^7*b^5*c^3*d^5*f^2 + 498*A^2*a^6*b^6*c^2*d^6*f^2 - 498*A^2*a^5*b^7*c^5*d^3*
f^2 - 414*A^2*a^5*b^7*c^3*d^5*f^2 + 354*A^2*a^6*b^6*c^6*d^2*f^2 - 318*A^2*a
^7*b^5*c^5*d^3*f^2 + 144*A^2*a^8*b^4*c^2*d^6*f^2 + 102*A^2*a^3*b^9*c^5*d^3*
f^2 + 84*A^2*a^4*b^8*c^4*d^4*f^2 + 81*A^2*a^4*b^8*c^2*d^6*f^2 + 72*A^2*a^8*
b^4*c^4*d^4*f^2 + 70*A^2*a^9*b^3*c^3*d^5*f^2 - 66*A^2*a^2*b^10*c^6*d^2*f^2
+ 48*A^2*a^4*b^8*c^6*d^2*f^2 - 42*A^2*a^10*b^2*c^2*d^6*f^2 + 24*A^2*a^2*b^1
0*c^2*d^6*f^2 + 20*A^2*a^9*b^3*c^5*d^3*f^2 - 15*A^2*a^10*b^2*c^4*d^4*f^2 -
15*A^2*a^8*b^4*c^6*d^2*f^2 - 15*A^2*a^2*b^10*c^4*d^4*f^2 - 12*A^2*a^3*b^9*c
^3*d^5*f^2 - 8*B*C*b^12*c^7*d*f^2 + 4*B*C*a^12*c*d^7*f^2 - 24*B*C*a^11*b*d^
8*f^2 + 8*A*B*b^12*c^7*d*f^2 - 8*A*B*b^12*c*d^7*f^2 + 24*B*C*a*b^11*c^8*f^2
 - 8*A*B*a^12*c*d^7*f^2 + 12*A*B*a^11*b*d^8*f^2 - 24*A*B*a*b^11*c^8*f^2 - 1
74*C^2*a^7*b^5*c*d^7*f^2 - 174*C^2*a^5*b^7*c^7*d*f^2 + 82*C^2*a^9*b^3*c*d^7
*f^2 + 82*C^2*a^3*b^9*c^7*d*f^2 + 6*C^2*a^11*b*c^3*d^5*f^2 + 6*C^2*a^7*b^5*
c^7*d*f^2 + 6*C^2*a^5*b^7*c*d^7*f^2 + 6*C^2*a*b^11*c^5*d^3*f^2 + 162*B^2*a^
7*b^5*c*d^7*f^2 + 138*B^2*a^5*b^7*c^7*d*f^2 - 118*B^2*a^3*b^9*c^7*d*f^2 - 8
6*B^2*a^9*b^3*c*d^7*f^2 - 30*B^2*a*b^11*c^5*d^3*f^2 - 18*B^2*a^7*b^5*c^7*d*
f^2 - 18*B^2*a^5*b^7*c*d^7*f^2 - 12*B^2*a*b^11*c^3*d^5*f^2 - 6*B^2*a^11*b*c
^3*d^5*f^2 - 4*B^2*a^3*b^9*c*d^7*f^2 - 270*A^2*a^7*b^5*c*d^7*f^2 - 174*A^2*
a^5*b^7*c^7*d*f^2 - 90*A^2*a^5*b^7*c*d^7*f^2 + 82*A^2*a^3*b^9*c^7*d*f^2 + 5
0*A^2*a^9*b^3*c*d^7*f^2 - 32*A^2*a^3*b^9*c*d^7*f^2 + 6*A^2*a^11*b*c^3*d^5*f
^2 + 6*A^2*a^7*b^5*c^7*d*f^2 + 6*A^2*a*b^11*c^5*d^3*f^2 + 6*C^2*a^11*b*c*d^
7*f^2 + 6*C^2*a*b^11*c^7*d*f^2 - 18*B^2*a*b^11*c^7*d*f^2 - 6*B^2*a^11*b*c*d
^7*f^2 + 6*A^2*a^11*b*c*d^7*f^2 + 6*A^2*a*b^11*c^7*d*f^2 - 6*A*C*b^12*c^8*f
^2 - 2*A*C*a^12*d^8*f^2 + 4*C^2*b^12*c^4*d^4*f^2 + 3*C^2*b^12*c^6*d^2*f^2 +
 4*C^2*a^12*c^4*d^4*f^2 + 4*B^2*b^12*c^4*d^4*f^2 + 4*B^2*b^12*c^2*d^6*f^2 +
 3*C^2*a^12*c^2*d^6*f^2 + 3*B^2*b^12*c^6*d^2*f^2 + 33*C^2*a^8*b^4*d^8*f^2 -
 27*C^2*a^10*b^2*d^8*f^2 - 4*A^2*b^12*c^4*d^4*f^2 + 3*B^2*a^12*c^2*d^6*f^2
- C^2*a^6*b^6*d^8*f^2 - A^2*b^12*c^6*d^2*f^2 + 33*C^2*a^4*b^8*c^8*f^2 + 33*
B^2*a^10*b^2*d^8*f^2 - 27*C^2*a^2*b^10*c^8*f^2 - 27*B^2*a^8*b^4*d^8*f^2 + 3
*B^2*a^6*b^6*d^8*f^2 - C^2*a^6*b^6*c^8*f^2 - A^2*a^12*c^2*d^6*f^2 + 117*A^2
*a^8*b^4*d^8*f^2 + 111*A^2*a^6*b^6*d^8*f^2 + 72*A^2*a^4*b^8*d^8*f^2 + 33*B^
2*a^2*b^10*c^8*f^2 - 27*B^2*a^4*b^8*c^8*f^2 + 24*A^2*a^2*b^10*d^8*f^2 + 3*B
^2*a^6*b^6*c^8*f^2 - 3*A^2*a^10*b^2*d^8*f^2 + 33*A^2*a^4*b^8*c^8*f^2 - 27*A
^2*a^2*b^10*c^8*f^2 - A^2*a^6*b^6*c^8*f^2 + 3*C^2*b^12*c^8*f^2 + 3*C^2*a^12
*d^8*f^2 + 4*A^2*b^12*d^8*f^2 - B^2*b^12*c^8*f^2 - B^2*a^12*d^8*f^2 + 3*A^2
*b^12*c^8*f^2 + 3*A^2*a^12*d^8*f^2 - 24*A*B*C*a*b^8*c*d^6*f + 342*A*B*C*a^4
*b^5*c^2*d^5*f - 186*A*B*C*a^5*b^4*c^3*d^4*f - 66*A*B*C*a^2*b^7*c^4*d^3*f +
 48*A*B*C*a^2*b^7*c^2*d^5*f + 42*A*B*C*a^6*b^3*c^2*d^5*f + 26*A*B*C*a^3*b^6
*c^5*d^2*f + 24*A*B*C*a^6*b^3*c^4*d^3*f - 18*A*B*C*a^7*b^2*c^3*d^4*f - 18*A
*B*C*a^4*b^5*c^4*d^3*f - 8*A*B*C*a^3*b^6*c^3*d^4*f + 6*A*B*C*a^5*b^4*c^5*d^
2*f - 128*A*B*C*a^3*b^6*c*d^6*f + 126*A*B*C*a^7*b^2*c*d^6*f + 72*A*B*C*a*b^
8*c^3*d^4*f - 36*A*B*C*a^8*b*c^2*d^5*f - 36*A*B*C*a*b^8*c^5*d^2*f + 30*A*B*
C*a^2*b^7*c^6*d*f - 12*A*B*C*a^5*b^4*c*d^6*f - 12*A*B*C*a^4*b^5*c^6*d*f - 2

1*B^2*C*a^8*b*c*d^6*f - 3*B^2*C*a*b^8*c^6*d*f + 21*A^2*C*a^8*b*c*d^6*f - 21
*A*C^2*a^8*b*c*d^6*f - 9*A^2*C*a*b^8*c^6*d*f + 9*A*C^2*a*b^8*c^6*d*f + 36*A
^2*B*a*b^8*c*d^6*f + 21*A*B^2*a^8*b*c*d^6*f + 3*A*B^2*a*b^8*c^6*d*f + 16*A*
B*C*b^9*c^4*d^3*f - 16*A*B*C*b^9*c^2*d^5*f - 78*A*B*C*a^6*b^3*d^7*f + 24*A*
B*C*a^4*b^5*d^7*f + 2*A*B*C*a^3*b^6*c^7*f - 237*B^2*C*a^4*b^5*c^3*d^4*f + 1
65*B*C^2*a^5*b^4*c^3*d^4*f + 92*B^2*C*a^3*b^6*c^2*d^5*f - 81*B^2*C*a^7*b^2*
c^2*d^5*f + 77*B^2*C*a^3*b^6*c^4*d^3*f - 75*B*C^2*a^4*b^5*c^2*d^5*f + 69*B^
2*C*a^5*b^4*c^4*d^3*f + 69*B*C^2*a^4*b^5*c^4*d^3*f - 68*B*C^2*a^3*b^6*c^3*d
^4*f - 63*B^2*C*a^4*b^5*c^5*d^2*f - 61*B*C^2*a^6*b^3*c^2*d^5*f + 57*B*C^2*a
^2*b^7*c^4*d^3*f - 53*B*C^2*a^3*b^6*c^5*d^2*f - 44*B*C^2*a^6*b^3*c^4*d^3*f
 - 36*B^2*C*a^2*b^7*c^3*d^4*f + 35*B^2*C*a^6*b^3*c^3*d^4*f - 33*B^2*C*a^5*b^
4*c^2*d^5*f + 33*B^2*C*a^2*b^7*c^5*d^2*f + 33*B*C^2*a^7*b^2*c^3*d^4*f - 12*
B^2*C*a^7*b^2*c^4*d^3*f + 9*B*C^2*a^5*b^4*c^5*d^2*f + 4*B^2*C*a^6*b^3*c^5*d
^2*f + 225*A^2*C*a^5*b^4*c^2*d^5*f - 105*A*C^2*a^5*b^4*c^2*d^5*f - 99*A^2*C
*a^4*b^5*c^3*d^4*f - 81*A^2*C*a^4*b^5*c^5*d^2*f + 67*A^2*C*a^3*b^6*c^4*d^3*
f - 59*A*C^2*a^3*b^6*c^4*d^3*f - 57*A*C^2*a^7*b^2*c^2*d^5*f + 57*A*C^2*a^2*
b^7*c^5*d^2*f + 51*A^2*C*a^5*b^4*c^4*d^3*f + 48*A^2*C*a^2*b^7*c^3*d^4*f + 4
5*A*C^2*a^4*b^5*c^5*d^2*f - 35*A^2*C*a^6*b^3*c^3*d^4*f + 33*A^2*C*a^7*b^2*c
^2*d^5*f - 33*A^2*C*a^2*b^7*c^5*d^2*f + 33*A*C^2*a^5*b^4*c^4*d^3*f + 27*A*C
^2*a^6*b^3*c^3*d^4*f + 24*A*C^2*a^3*b^6*c^2*d^5*f - 24*A*C^2*a^2*b^7*c^3*d^
4*f - 21*A*C^2*a^4*b^5*c^3*d^4*f - 16*A^2*C*a^3*b^6*c^2*d^5*f - 243*A^2*B*a
^4*b^5*c^2*d^5*f - 156*A*B^2*a^3*b^6*c^2*d^5*f + 141*A*B^2*a^4*b^5*c^3*d^4*
f + 108*A^2*B*a^3*b^6*c^3*d^4*f - 105*A*B^2*a^3*b^6*c^4*d^3*f + 84*A*B^2*a^
2*b^7*c^3*d^4*f + 81*A*B^2*a^5*b^4*c^2*d^5*f + 51*A^2*B*a^6*b^3*c^2*d^5*f -
 51*A^2*B*a^4*b^5*c^4*d^3*f - 48*A^2*B*a^2*b^7*c^2*d^5*f + 45*A^2*B*a^5*b^4
*c^3*d^4*f + 39*A*B^2*a^4*b^5*c^5*d^2*f - 35*A*B^2*a^6*b^3*c^3*d^4*f + 33*A
*B^2*a^7*b^2*c^2*d^5*f + 27*A^2*B*a^3*b^6*c^5*d^2*f - 21*A*B^2*a^5*b^4*c^4*
d^3*f + 20*A^2*B*a^6*b^3*c^4*d^3*f - 15*A^2*B*a^7*b^2*c^3*d^4*f - 15*A^2*B*
a^5*b^4*c^5*d^2*f + 9*A^2*B*a^2*b^7*c^4*d^3*f + 3*A*B^2*a^2*b^7*c^5*d^2*f +
 2*A*B*C*b^9*c^6*d*f - 6*A*B*C*a^9*c*d^6*f + 18*A*B*C*a^8*b*d^7*f - 6*A*B*C
*a*b^8*c^7*f + 63*B^2*C*a^6*b^3*c*d^6*f - 48*B^2*C*a*b^8*c^4*d^3*f + 42*B*C
^2*a^8*b*c^2*d^5*f + 42*B*C^2*a^5*b^4*c*d^6*f - 39*B*C^2*a^7*b^2*c*d^6*f +
30*B*C^2*a*b^8*c^5*d^2*f - 24*B^2*C*a^4*b^5*c*d^6*f - 24*B*C^2*a*b^8*c^3*d^
4*f + 17*B^2*C*a^3*b^6*c^6*d*f - 15*B*C^2*a^2*b^7*c^6*d*f + 12*B^2*C*a^8*b*
c^3*d^4*f + 12*B^2*C*a*b^8*c^2*d^5*f + 6*B*C^2*a^4*b^5*c^6*d*f - 192*A^2*C*
a^4*b^5*c*d^6*f - 99*A^2*C*a^6*b^3*c*d^6*f + 84*A*C^2*a^4*b^5*c*d^6*f + 59*
A*C^2*a^6*b^3*c*d^6*f + 51*A^2*C*a^3*b^6*c^6*d*f - 51*A*C^2*a^3*b^6*c^6*d*f
 - 36*A^2*C*a*b^8*c^2*d^5*f - 24*A*C^2*a*b^8*c^4*d^3*f + 24*A*C^2*a*b^8*c^2
*d^5*f + 12*A^2*C*a*b^8*c^4*d^3*f + 12*A*C^2*a^8*b*c^3*d^4*f + 160*A^2*B*a^
3*b^6*c*d^6*f - 99*A*B^2*a^6*b^3*c*d^6*f - 87*A^2*B*a^7*b^2*c*d^6*f - 72*A*
B^2*a^4*b^5*c*d^6*f - 48*A*B^2*a*b^8*c^2*d^5*f - 36*A^2*B*a*b^8*c^3*d^4*f +
 24*A*B^2*a*b^8*c^4*d^3*f - 17*A*B^2*a^3*b^6*c^6*d*f - 15*A^2*B*a^2*b^7*c^6
*d*f + 12*A*B^2*a^2*b^7*c*d^6*f + 6*A^2*B*a^8*b*c^2*d^5*f - 6*A^2*B*a^5*b^4
*c*d^6*f + 6*A^2*B*a^4*b^5*c^6*d*f + 6*A^2*B*a*b^8*c^5*d^2*f + 12*B^2*C*b^9
*c^3*d^4*f - 12*B*C^2*b^9*c^4*d^3*f - 12*A^2*C*b^9*c^3*d^4*f - 8*A*C^2*b^9*

c^5*d^2*f + 8*A*C^2*b^9*c^3*d^4*f + 4*B^2*C*a^9*c^2*d^5*f + 4*A^2*C*b^9*c^5
*d^2*f - 4*B*C^2*a^9*c^3*d^4*f + 12*A^2*B*b^9*c^2*d^5*f - 8*A*B^2*b^9*c^3*d
^4*f - 4*A^2*B*b^9*c^4*d^3*f + 4*A*C^2*a^9*c^2*d^5*f + 3*B^2*C*a^7*b^2*d^7*
f - B*C^2*a^6*b^3*d^7*f + 96*A^2*C*a^5*b^4*d^7*f - 39*A^2*C*a^7*b^2*d^7*f -
 36*A*C^2*a^5*b^4*d^7*f + 32*A^2*C*a^3*b^6*d^7*f + 15*A*C^2*a^7*b^2*d^7*f -
 3*B^2*C*a^2*b^7*c^7*f - B*C^2*a^3*b^6*c^7*f + 111*A^2*B*a^6*b^3*d^7*f - 39
*A*B^2*a^7*b^2*d^7*f + 24*A*B^2*a^5*b^4*d^7*f - 9*A^2*C*a^2*b^7*c^7*f + 9*A
*C^2*a^2*b^7*c^7*f - 4*A*B^2*a^3*b^6*d^7*f + 3*A*B^2*a^2*b^7*c^7*f - A^2*B*
a^3*b^6*c^7*f + 3*C^3*a^8*b*c*d^6*f - 3*C^3*a*b^8*c^6*d*f - 3*A^3*a^8*b*c*d
^6*f + 3*A^3*a*b^8*c^6*d*f - B*C^2*b^9*c^6*d*f + 4*A^2*C*b^9*c*d^6*f + 3*B*
C^2*a^9*c*d^6*f + 8*A*B^2*b^9*c*d^6*f + 3*B*C^2*a^8*b*d^7*f - A^2*B*b^9*c^6
*d*f + 12*A^2*C*a*b^8*d^7*f + 3*B*C^2*a*b^8*c^7*f - A^2*B*a^9*c*d^6*f - 9*A
^2*B*a^8*b*d^7*f + 3*A^2*B*a*b^8*c^7*f - 39*C^3*a^5*b^4*c^4*d^3*f + 39*C^3*
a^4*b^5*c^3*d^4*f + 27*C^3*a^7*b^2*c^2*d^5*f - 27*C^3*a^2*b^7*c^5*d^2*f - 1
7*C^3*a^6*b^3*c^3*d^4*f + 17*C^3*a^3*b^6*c^4*d^3*f + 3*C^3*a^5*b^4*c^2*d^5*
f - 3*C^3*a^4*b^5*c^5*d^2*f - 63*B^3*a^5*b^4*c^3*d^4*f + 57*B^3*a^4*b^5*c^2
*d^5*f - 51*B^3*a^2*b^7*c^4*d^3*f + 48*B^3*a^3*b^6*c^3*d^4*f + 31*B^3*a^6*b
^3*c^2*d^5*f + 27*B^3*a^3*b^6*c^5*d^2*f + 16*B^3*a^6*b^3*c^4*d^3*f - 15*B^3
*a^5*b^4*c^5*d^2*f - 12*B^3*a^2*b^7*c^2*d^5*f + 9*B^3*a^4*b^5*c^4*d^3*f - 3
*B^3*a^7*b^2*c^3*d^4*f - 123*A^3*a^5*b^4*c^2*d^5*f + 81*A^3*a^4*b^5*c^3*d^4
*f - 45*A^3*a^5*b^4*c^4*d^3*f + 39*A^3*a^4*b^5*c^5*d^2*f + 25*A^3*a^6*b^3*c
^3*d^4*f - 25*A^3*a^3*b^6*c^4*d^3*f - 24*A^3*a^2*b^7*c^3*d^4*f - 8*A^3*a^3*
b^6*c^2*d^5*f - 3*A^3*a^7*b^2*c^2*d^5*f + 3*A^3*a^2*b^7*c^5*d^2*f - 17*C^3*
a^6*b^3*c*d^6*f + 17*C^3*a^3*b^6*c^6*d*f - 12*C^3*a^8*b*c^3*d^4*f + 12*C^3*
a*b^8*c^4*d^3*f + 24*B^3*a*b^8*c^3*d^4*f + 21*B^3*a^7*b^2*c*d^6*f - 18*B^3*
a^5*b^4*c*d^6*f - 15*B^3*a^2*b^7*c^6*d*f - 6*B^3*a^8*b*c^2*d^5*f + 6*B^3*a^
4*b^5*c^6*d*f + 6*B^3*a*b^8*c^5*d^2*f + 4*B^3*a^3*b^6*c*d^6*f + 108*A^3*a^4
*b^5*c*d^6*f + 57*A^3*a^6*b^3*c*d^6*f - 17*A^3*a^3*b^6*c^6*d*f + 12*A^3*a*b
^8*c^2*d^5*f + 4*C^3*b^9*c^5*d^2*f - 4*C^3*a^9*c^2*d^5*f - 4*B^3*b^9*c^2*d^
5*f + 4*A^3*b^9*c^3*d^4*f + 3*C^3*a^7*b^2*d^7*f - 3*C^3*a^2*b^7*c^7*f - B^3
*a^6*b^3*d^7*f - 60*A^3*a^5*b^4*d^7*f - 32*A^3*a^3*b^6*d^7*f + 21*A^3*a^7*b
^2*d^7*f - B^3*a^3*b^6*c^7*f + 3*A^3*a^2*b^7*c^7*f - B^3*b^9*c^6*d*f - 4*A^
3*b^9*c*d^6*f - B^3*a^9*c*d^6*f + 3*B^3*a^8*b*d^7*f - 12*A^3*a*b^8*d^7*f +
3*B^3*a*b^8*c^7*f - B^2*C*a^9*d^7*f - 4*A^2*B*b^9*d^7*f + 3*A^2*C*b^9*c^7*f
 - 3*A*C^2*b^9*c^7*f - A*C^2*a^9*d^7*f - A*B^2*b^9*c^7*f - C^3*a^9*d^7*f -
A^3*b^9*c^7*f + B^2*C*b^9*c^7*f + A^2*C*a^9*d^7*f + A*B^2*a^9*d^7*f + C^3*b
^9*c^7*f + A^3*a^9*d^7*f - 6*A*B^2*C*a^5*b*c*d^5 - 21*A^2*B*C*a^3*b^3*c^2*d
^4 + 21*A*B*C^2*a^3*b^3*c^2*d^4 + 12*A*B^2*C*a^4*b^2*c^2*d^4 - 12*A*B^2*C*a
^2*b^4*c^2*d^4 - 10*A*B^2*C*a^3*b^3*c^3*d^3 - 6*A*B*C^2*a^4*b^2*c^3*d^3 + 3
*A^2*B*C*a^4*b^2*c^3*d^3 + 3*A^2*B*C*a^2*b^4*c^3*d^3 + 3*A*B^2*C*a^2*b^4*c^
4*d^2 + 3*A*B*C^2*a^2*b^4*c^3*d^3 + 2*A*B*C^2*a^3*b^3*c^4*d^2 - A^2*B*C*a^3
*b^3*c^4*d^2 + 18*A^2*B*C*a^2*b^4*c*d^5 + 10*A*B^2*C*a^3*b^3*c*d^5 + 9*A^2*
B*C*a^4*b^2*c*d^5 - 9*A*B*C^2*a^4*b^2*c*d^5 - 9*A*B*C^2*a^2*b^4*c*d^5 - 6*A
^2*B*C*a*b^5*c^2*d^4 + 6*A*B^2*C*a*b^5*c^3*d^3 + 6*A*B*C^2*a^5*b*c^2*d^4 -
6*A*B*C^2*a*b^5*c^4*d^2 - 3*A^2*B*C*a^5*b*c^2*d^4 + 3*A^2*B*C*a*b^5*c^4*d^2

+ 3*A*B*C^2*a*b^5*c^2*d^4 - 3*B^3*C*a^5*b*c^2*d^4 + 3*B^3*C*a^4*b^2*c*d^5
+ 3*B^3*C*a*b^5*c^4*d^2 + 3*B^2*C^2*a^5*b*c*d^5 - 3*B*C^3*a^5*b*c^2*d^4 + 3
*B*C^3*a^4*b^2*c*d^5 + 3*B*C^3*a*b^5*c^4*d^2 + 24*A^3*C*a^3*b^3*c*d^5 + 8*A
*C^3*a^3*b^3*c*d^5 - 9*A^3*B*a^2*b^4*c*d^5 - 9*A*B^3*a^2*b^4*c*d^5 - 3*A^3*
B*a^4*b^2*c*d^5 + 3*A^3*B*a*b^5*c^2*d^4 + 3*A^2*B^2*a^5*b*c*d^5 - 3*A*B^3*a
^4*b^2*c*d^5 + 3*A*B^3*a*b^5*c^2*d^4 + 5*A*B*C^2*b^6*c^3*d^3 - 4*A^2*B*C*b^
6*c^3*d^3 - A*B^2*C*b^6*c^4*d^2 - 3*A*B^2*C*a^4*b^2*d^6 - 2*A^2*B*C*a^3*b^3
*d^6 + 9*B^2*C^2*a^3*b^3*c^3*d^3 - 6*B^2*C^2*a^4*b^2*c^2*d^4 + 6*B^2*C^2*a^
2*b^4*c^2*d^4 - 3*B^2*C^2*a^2*b^4*c^4*d^2 + 24*A^2*C^2*a^3*b^3*c^3*d^3 - 15
*A^2*C^2*a^4*b^2*c^2*d^4 - 9*A^2*C^2*a^2*b^4*c^4*d^2 + 3*A^2*C^2*a^2*b^4*c^
2*d^4 + 9*A^2*B^2*a^2*b^4*c^2*d^4 - 3*A^2*B^2*a^4*b^2*c^2*d^4 + 4*A^2*B*C*b
^6*c*d^5 - 2*A*B*C^2*b^6*c*d^5 + 2*A*B*C^2*a^6*c*d^5 - A^2*B*C*a^6*c*d^5 +
6*A^2*B*C*a^5*b*d^6 - 3*A*B*C^2*a^5*b*d^6 - 7*B^3*C*a^3*b^3*c^2*d^4 - 7*B*C
^3*a^3*b^3*c^2*d^4 + 3*B^3*C*a^4*b^2*c^3*d^3 - 3*B^3*C*a^2*b^4*c^3*d^3 - 3*
B^2*C^2*a*b^5*c^3*d^3 + 3*B*C^3*a^4*b^2*c^3*d^3 - 3*B*C^3*a^2*b^4*c^3*d^3 -
B^3*C*a^3*b^3*c^4*d^2 - B^2*C^2*a^3*b^3*c*d^5 - B*C^3*a^3*b^3*c^4*d^2 - 24
*A^2*C^2*a^3*b^3*c*d^5 - 24*A*C^3*a^3*b^3*c^3*d^3 + 12*A*C^3*a^4*b^2*c^2*d^
4 + 9*A*C^3*a^2*b^4*c^4*d^2 - 8*A^3*C*a^3*b^3*c^3*d^3 + 6*A^3*C*a^4*b^2*c^2
*d^4 - 6*A^3*C*a^2*b^4*c^2*d^4 + 3*A^3*C*a^2*b^4*c^4*d^2 - 9*A^2*B^2*a^3*b^
3*c*d^5 + 7*A^3*B*a^3*b^3*c^2*d^4 + 7*A*B^3*a^3*b^3*c^2*d^4 - 3*A^3*B*a^2*b
^4*c^3*d^3 - 3*A^2*B^2*a*b^5*c^3*d^3 - 3*A*B^3*a^2*b^4*c^3*d^3 - 5*A^2*C^2*
b^6*c^2*d^4 + 3*A^2*C^2*b^6*c^4*d^2 + 12*A^2*C^2*a^4*b^2*d^6 + 3*A^2*C^2*a^
2*b^4*d^6 + 6*A^2*B^2*a^4*b^2*d^6 + 3*A^2*B^2*a^2*b^4*d^6 + A*B*C^2*a^3*b^3
*d^6 - 3*B^4*a*b^5*c^3*d^3 - B^4*a^3*b^3*c*d^5 + A^2*B^2*a^3*b^3*c^3*d^3 -
8*A^4*a^3*b^3*c*d^5 - 2*B^3*C*b^6*c^3*d^3 - 2*B*C^3*b^6*c^3*d^3 + 4*A^3*C*b
^6*c^2*d^4 - 3*A*C^3*b^6*c^4*d^2 + 2*A*C^3*b^6*c^2*d^4 - A^3*C*b^6*c^4*d^2
- 2*A*C^3*a^6*c^2*d^4 - 15*A^3*C*a^4*b^2*d^6 - 6*A^3*C*a^2*b^4*d^6 - 3*A*C^
3*a^4*b^2*d^6 + 3*B^4*a^5*b*c*d^5 - B^3*C*a^6*c*d^5 - B*C^3*a^6*c*d^5 - 2*A
^3*B*b^6*c*d^5 - 2*A*B^3*b^6*c*d^5 - 3*A^3*B*a^5*b*d^6 - 3*A*B^3*a^5*b*d^6
+ 8*C^4*a^3*b^3*c^3*d^3 - 3*C^4*a^4*b^2*c^2*d^4 - 3*C^4*a^2*b^4*c^4*d^2 + 6
*B^4*a^2*b^4*c^2*d^4 - 3*B^4*a^4*b^2*c^2*d^4 + 3*A^4*a^2*b^4*c^2*d^4 + B^2*
C^2*b^6*c^4*d^2 + B^2*C^2*b^6*c^2*d^4 + B^2*C^2*a^6*c^2*d^4 + A^2*C^2*a^6*c
^2*d^4 - 2*A^3*C*b^6*d^6 + A^3*B*b^6*c^3*d^3 + A*B^3*b^6*c^3*d^3 + A^3*B*a^
3*b^3*d^6 + A*B^3*a^3*b^3*d^6 - A^4*b^6*c^2*d^4 + 6*A^4*a^4*b^2*d^6 + 3*A^4
*a^2*b^4*d^6 - 2*A^2*C^2*a^6*d^6 + A*B^2*C*a^6*d^6 + B^4*a^3*b^3*c^3*d^3 +
A^3*C*a^6*d^6 + A*C^3*a^6*d^6 + C^4*b^6*c^4*d^2 + C^4*a^6*c^2*d^4 + B^4*b^6
*c^2*d^4 + A^2*C^2*b^6*d^6 + A^2*B^2*b^6*d^6 + A^4*b^6*d^6, f, k)*((4*a^5*b
^12*d^9 + 12*a^7*b^10*d^9 + 8*a^9*b^8*d^9 - 8*a^11*b^6*d^9 - 12*a^13*b^4*d^
9 - 4*a^15*b^2*d^9 + 4*b^17*c^5*d^4 - 4*b^17*c^7*d^2 - 12*a*b^16*c^4*d^5 +
28*a*b^16*c^6*d^3 + 32*a^3*b^14*c^8*d - 12*a^4*b^13*c*d^8 + 48*a^5*b^12*c^8
*d - 20*a^6*b^11*c*d^8 + 32*a^7*b^10*c^8*d + 48*a^8*b^9*c*d^8 + 8*a^9*b^8*c
^8*d + 152*a^10*b^7*c*d^8 + 148*a^12*b^5*c*d^8 + 60*a^14*b^3*c*d^8 + 8*a^2*
b^15*c^3*d^6 - 44*a^2*b^15*c^5*d^4 - 52*a^2*b^15*c^7*d^2 + 8*a^3*b^14*c^2*d
^7 - 12*a^3*b^14*c^4*d^5 + 172*a^3*b^14*c^6*d^3 + 68*a^4*b^13*c^3*d^6 - 248
*a^4*b^13*c^5*d^4 - 168*a^4*b^13*c^7*d^2 - 28*a^5*b^12*c^2*d^7 + 40*a^5*b^1

2*c^4*d^5 + 408*a^5*b^12*c^6*d^3 + 252*a^6*b^11*c^3*d^6 - 472*a^6*b^11*c^5*
d^4 - 232*a^6*b^11*c^7*d^2 - 228*a^7*b^10*c^2*d^7 + 40*a^7*b^10*c^4*d^5 + 4
72*a^7*b^10*c^6*d^3 + 488*a^8*b^9*c^3*d^6 - 428*a^8*b^9*c^5*d^4 - 148*a^8*b
^9*c^7*d^2 - 472*a^9*b^8*c^2*d^7 - 60*a^9*b^8*c^4*d^5 + 268*a^9*b^8*c^6*d^3
 + 512*a^10*b^7*c^3*d^6 - 188*a^10*b^7*c^5*d^4 - 36*a^10*b^7*c^7*d^2 - 448*
a^11*b^6*c^2*d^7 - 92*a^11*b^6*c^4*d^5 + 60*a^11*b^6*c^6*d^3 + 276*a^12*b^5
*c^3*d^6 - 32*a^12*b^5*c^5*d^4 - 204*a^13*b^4*c^2*d^7 - 32*a^13*b^4*c^4*d^5
 + 60*a^14*b^3*c^3*d^6 - 36*a^15*b^2*c^2*d^7 + 8*a*b^16*c^8*d + 8*a^16*b*c*
d^8)/(a^12*d^4 + b^12*c^4 + 4*a^2*b^10*c^4 + 6*a^4*b^8*c^4 + 4*a^6*b^6*c^4
+ a^8*b^4*c^4 + a^4*b^8*d^4 + 4*a^6*b^6*d^4 + 6*a^8*b^4*d^4 + 4*a^10*b^2*d^
4 - 4*a^3*b^9*c*d^3 - 16*a^3*b^9*c^3*d - 16*a^5*b^7*c*d^3 - 24*a^5*b^7*c^3*
d - 24*a^7*b^5*c*d^3 - 16*a^7*b^5*c^3*d - 16*a^9*b^3*c*d^3 - 4*a^9*b^3*c^3*
d + 6*a^2*b^10*c^2*d^2 + 24*a^4*b^8*c^2*d^2 + 36*a^6*b^6*c^2*d^2 + 24*a^8*b
^4*c^2*d^2 + 6*a^10*b^2*c^2*d^2 - 4*a*b^11*c^3*d - 4*a^11*b*c*d^3) + (tan(e
 + f*x)*(6*a^16*b*d^9 + 6*b^17*c^8*d + 8*a^4*b^13*d^9 + 38*a^6*b^11*d^9 + 7
8*a^8*b^9*d^9 + 92*a^10*b^7*d^9 + 68*a^12*b^5*d^9 + 30*a^14*b^3*d^9 + 8*b^1
7*c^4*d^5 + 6*b^17*c^6*d^3 - 32*a*b^16*c^3*d^6 - 20*a*b^16*c^5*d^4 - 20*a*b
^16*c^7*d^2 + 22*a^2*b^15*c^8*d - 32*a^3*b^14*c*d^8 + 28*a^4*b^13*c^8*d - 1
48*a^5*b^12*c*d^8 + 12*a^6*b^11*c^8*d - 292*a^7*b^10*c*d^8 - 2*a^8*b^9*c^8*
d - 328*a^9*b^8*c*d^8 - 2*a^10*b^7*c^8*d - 232*a^11*b^6*c*d^8 - 100*a^13*b^
4*c*d^8 - 20*a^15*b^2*c*d^8 - 2*a^16*b*c^2*d^7 + 48*a^2*b^15*c^2*d^7 + 58*a
^2*b^15*c^4*d^5 + 32*a^2*b^15*c^6*d^3 - 152*a^3*b^14*c^3*d^6 - 28*a^3*b^14*
c^5*d^4 - 68*a^3*b^14*c^7*d^2 + 218*a^4*b^13*c^2*d^7 + 60*a^4*b^13*c^4*d^5
+ 38*a^4*b^13*c^6*d^3 - 236*a^5*b^12*c^3*d^6 + 128*a^5*b^12*c^5*d^4 - 72*a^
5*b^12*c^7*d^2 + 400*a^6*b^11*c^2*d^7 - 210*a^6*b^11*c^4*d^5 - 48*a^6*b^11*
c^6*d^3 - 52*a^7*b^10*c^3*d^6 + 392*a^7*b^10*c^5*d^4 - 8*a^7*b^10*c^7*d^2 +
 378*a^8*b^9*c^2*d^7 - 560*a^8*b^9*c^4*d^5 - 142*a^8*b^9*c^6*d^3 + 232*a^9*
b^8*c^3*d^6 + 428*a^9*b^8*c^5*d^4 + 28*a^9*b^8*c^7*d^2 + 192*a^10*b^7*c^2*d
^7 - 522*a^10*b^7*c^4*d^5 - 112*a^10*b^7*c^6*d^3 + 256*a^11*b^6*c^3*d^6 + 2
12*a^11*b^6*c^5*d^4 + 12*a^11*b^6*c^7*d^2 + 46*a^12*b^5*c^2*d^7 - 212*a^12*
b^5*c^4*d^5 - 30*a^12*b^5*c^6*d^3 + 100*a^13*b^4*c^3*d^6 + 40*a^13*b^4*c^5*
d^4 - 30*a^14*b^3*c^4*d^5 + 12*a^15*b^2*c^3*d^6))/(a^12*d^4 + b^12*c^4 + 4*
a^2*b^10*c^4 + 6*a^4*b^8*c^4 + 4*a^6*b^6*c^4 + a^8*b^4*c^4 + a^4*b^8*d^4 +
4*a^6*b^6*d^4 + 6*a^8*b^4*d^4 + 4*a^10*b^2*d^4 - 4*a^3*b^9*c*d^3 - 16*a^3*b
^9*c^3*d - 16*a^5*b^7*c*d^3 - 24*a^5*b^7*c^3*d - 24*a^7*b^5*c*d^3 - 16*a^7*
b^5*c^3*d - 16*a^9*b^3*c*d^3 - 4*a^9*b^3*c^3*d + 6*a^2*b^10*c^2*d^2 + 24*a^
4*b^8*c^2*d^2 + 36*a^6*b^6*c^2*d^2 + 24*a^8*b^4*c^2*d^2 + 6*a^10*b^2*c^2*d^
2 - 4*a*b^11*c^3*d - 4*a^11*b*c*d^3)) + (tan(e + f*x)*(3*A*a^13*b*d^8 - 3*A
*b^14*c^7*d + C*a^13*b*d^8 + 3*C*b^14*c^7*d + 8*A*a^3*b^11*d^8 + 24*A*a^5*b
^9*d^8 + 51*A*a^7*b^7*d^8 + 65*A*a^9*b^5*d^8 + 33*A*a^11*b^3*d^8 - 4*B*a^4*
b^10*d^8 + 7*B*a^6*b^8*d^8 + 21*B*a^8*b^6*d^8 + 5*B*a^10*b^4*d^8 - 5*B*a^12
*b^2*d^8 + 8*A*b^14*c^3*d^5 - 8*A*b^14*c^5*d^3 + 12*C*a^5*b^9*d^8 + 13*C*a^
7*b^7*d^8 - 9*C*a^9*b^5*d^8 - 9*C*a^11*b^3*d^8 - 12*B*b^14*c^4*d^4 - B*b^14
*c^6*d^2 + 12*C*b^14*c^5*d^3 - 8*A*a*b^13*c^2*d^6 + 8*A*a*b^13*c^4*d^4 + 13
*A*a*b^13*c^6*d^2 - 8*A*a^2*b^12*c*d^7 - A*a^2*b^12*c^7*d + 8*A*a^4*b^10*c*

d^7 + 7*A*a^4*b^10*c^7*d + 3*A*a^6*b^8*c*d^7 + 5*A*a^6*b^8*c^7*d - 63*A*a^8
*b^6*c*d^7 - 63*A*a^10*b^4*c*d^7 - 13*A*a^12*b^2*c*d^7 + 24*B*a*b^13*c^3*d^
5 + 30*B*a*b^13*c^5*d^3 + 8*B*a^3*b^11*c*d^7 + 13*B*a^3*b^11*c^7*d - 50*B*a
^5*b^9*c*d^7 + 5*B*a^5*b^9*c^7*d - 143*B*a^7*b^7*c*d^7 - B*a^7*b^7*c^7*d -
105*B*a^9*b^5*c*d^7 - 21*B*a^11*b^3*c*d^7 - 12*C*a*b^13*c^4*d^4 - 13*C*a*b^
13*c^6*d^2 + C*a^2*b^12*c^7*d - 44*C*a^4*b^10*c*d^7 - 7*C*a^4*b^10*c^7*d -
67*C*a^6*b^8*c*d^7 - 5*C*a^6*b^8*c^7*d + 7*C*a^8*b^6*c*d^7 + 39*C*a^10*b^4*
c*d^7 + 9*C*a^12*b^2*c*d^7 + 64*A*a^2*b^12*c^3*d^5 - 7*A*a^2*b^12*c^5*d^3 -
96*A*a^3*b^11*c^2*d^6 - 87*A*a^3*b^11*c^4*d^4 - A*a^3*b^11*c^6*d^2 + 263*A
*a^4*b^10*c^3*d^5 + 67*A*a^4*b^10*c^5*d^3 - 233*A*a^5*b^9*c^2*d^6 - 253*A*a
^5*b^9*c^4*d^4 - 41*A*a^5*b^9*c^6*d^2 + 381*A*a^6*b^8*c^3*d^5 + 123*A*a^6*b
^8*c^5*d^3 - 195*A*a^7*b^7*c^2*d^6 - 213*A*a^7*b^7*c^4*d^4 - 27*A*a^7*b^7*c
^6*d^2 + 189*A*a^8*b^6*c^3*d^5 + 57*A*a^8*b^6*c^5*d^3 - 35*A*a^9*b^5*c^2*d^
6 - 55*A*a^9*b^5*c^4*d^4 + 15*A*a^10*b^4*c^3*d^5 + 15*A*a^11*b^3*c^2*d^6 -
16*B*a^2*b^12*c^2*d^6 - 119*B*a^2*b^12*c^4*d^4 - 37*B*a^2*b^12*c^6*d^2 + 11
6*B*a^3*b^11*c^3*d^5 + 115*B*a^3*b^11*c^5*d^3 + 17*B*a^4*b^10*c^2*d^6 - 209
*B*a^4*b^10*c^4*d^4 - 65*B*a^4*b^10*c^6*d^2 + 85*B*a^5*b^9*c^3*d^5 + 125*B*
a^5*b^9*c^5*d^3 + 161*B*a^6*b^8*c^2*d^6 - 89*B*a^6*b^8*c^4*d^4 - 23*B*a^6*b
^8*c^6*d^2 - 97*B*a^7*b^7*c^3*d^5 + 25*B*a^7*b^7*c^5*d^3 + 213*B*a^8*b^6*c^
2*d^6 + 33*B*a^8*b^6*c^4*d^4 + 6*B*a^8*b^6*c^6*d^2 - 105*B*a^9*b^5*c^3*d^5
- 15*B*a^9*b^5*c^5*d^3 + 91*B*a^10*b^4*c^2*d^6 + 20*B*a^10*b^4*c^4*d^4 - 15
*B*a^11*b^3*c^3*d^5 + 6*B*a^12*b^2*c^2*d^6 - 32*C*a^2*b^12*c^3*d^5 + 23*C*a
^2*b^12*c^5*d^3 + 64*C*a^3*b^11*c^2*d^6 + 71*C*a^3*b^11*c^4*d^4 + C*a^3*b^1
1*c^6*d^2 - 215*C*a^4*b^10*c^3*d^5 - 43*C*a^4*b^10*c^5*d^3 + 185*C*a^5*b^9*
c^2*d^6 + 229*C*a^5*b^9*c^4*d^4 + 41*C*a^5*b^9*c^6*d^2 - 349*C*a^6*b^8*c^3*
d^5 - 107*C*a^6*b^8*c^5*d^3 + 163*C*a^7*b^7*c^2*d^6 + 197*C*a^7*b^7*c^4*d^4
+ 27*C*a^7*b^7*c^6*d^2 - 181*C*a^8*b^6*c^3*d^5 - 53*C*a^8*b^6*c^5*d^3 + 27
*C*a^9*b^5*c^2*d^6 + 51*C*a^9*b^5*c^4*d^4 - 15*C*a^10*b^4*c^3*d^5 - 15*C*a^
11*b^3*c^2*d^6 + 7*B*a*b^13*c^7*d - B*a^13*b*c*d^7))/(a^12*d^4 + b^12*c^4 +
4*a^2*b^10*c^4 + 6*a^4*b^8*c^4 + 4*a^6*b^6*c^4 + a^8*b^4*c^4 + a^4*b^8*d^4
+ 4*a^6*b^6*d^4 + 6*a^8*b^4*d^4 + 4*a^10*b^2*d^4 - 4*a^3*b^9*c*d^3 - 16*a^
3*b^9*c^3*d - 16*a^5*b^7*c*d^3 - 24*a^5*b^7*c^3*d - 24*a^7*b^5*c*d^3 - 16*a
^7*b^5*c^3*d - 16*a^9*b^3*c*d^3 - 4*a^9*b^3*c^3*d + 6*a^2*b^10*c^2*d^2 + 24
*a^4*b^8*c^2*d^2 + 36*a^6*b^6*c^2*d^2 + 24*a^8*b^4*c^2*d^2 + 6*a^10*b^2*c^2
*d^2 - 4*a*b^11*c^3*d - 4*a^11*b*c*d^3)) - (A^2*a^9*b^2*d^7 - 45*A^2*a^5*b^
6*d^7 - 24*A^2*a^7*b^4*d^7 - 28*A^2*a^3*b^8*d^7 - B^2*a^5*b^6*d^7 - 3*B^2*a
^9*b^2*d^7 + 4*A^2*b^11*c^3*d^4 - A^2*b^11*c^5*d^2 - C^2*a^5*b^6*d^7 - 4*C^
2*a^7*b^4*d^7 + C^2*a^9*b^2*d^7 - B^2*b^11*c^5*d^2 - C^2*b^11*c^5*d^2 - 4*A
^2*a*b^10*d^7 - 4*A^2*b^11*c*d^6 - 26*A^2*a^2*b^9*c^3*d^4 + 10*A^2*a^2*b^9*
c^5*d^2 + 14*A^2*a^3*b^8*c^2*d^5 - 24*A^2*a^3*b^8*c^4*d^3 + 72*A^2*a^4*b^7*
c^3*d^4 - 13*A^2*a^4*b^7*c^5*d^2 - 154*A^2*a^5*b^6*c^2*d^5 + 33*A^2*a^5*b^6
*c^4*d^3 - 42*A^2*a^6*b^5*c^3*d^4 + 28*A^2*a^7*b^4*c^2*d^5 + 34*B^2*a^2*b^9
*c^3*d^4 - 14*B^2*a^2*b^9*c^5*d^2 - 46*B^2*a^3*b^8*c^2*d^5 + 36*B^2*a^3*b^8
*c^4*d^3 - 68*B^2*a^4*b^7*c^3*d^4 + 11*B^2*a^4*b^7*c^5*d^2 + 102*B^2*a^5*b^
6*c^2*d^5 - 27*B^2*a^5*b^6*c^4*d^3 + 42*B^2*a^6*b^5*c^3*d^4 - 52*B^2*a^7*b^

4*c^2*d^5 - 22*C^2*a^2*b^9*c^3*d^4 + 10*C^2*a^2*b^9*c^5*d^2 + 10*C^2*a^3*b^
8*c^2*d^5 - 24*C^2*a^3*b^8*c^4*d^3 + 92*C^2*a^4*b^7*c^3*d^4 - 13*C^2*a^4*b^
7*c^5*d^2 - 134*C^2*a^5*b^6*c^2*d^5 + 33*C^2*a^5*b^6*c^4*d^3 - 30*C^2*a^6*b
^5*c^3*d^4 + 48*C^2*a^7*b^4*c^2*d^5 + 4*C^2*a^9*b^2*c^2*d^5 - 4*A*B*a^2*b^9
*d^7 + 4*A*B*a^4*b^7*d^7 + 19*A*B*a^6*b^5*d^7 + 18*A*B*a^8*b^3*d^7 + 12*A*C
*a^3*b^8*d^7 + 22*A*C*a^5*b^6*d^7 + 12*A*C*a^7*b^4*d^7 - 6*A*C*a^9*b^2*d^7
+ 4*A*B*b^11*c^2*d^5 + B*C*a^6*b^5*d^7 - 6*B*C*a^8*b^3*d^7 - 4*A*C*b^11*c^3
*d^4 + 2*A*C*b^11*c^5*d^2 - 2*A^2*a*b^10*c^6*d + 2*B^2*a*b^10*c^6*d - 2*C^2
*a*b^10*c^6*d + 4*C^2*a^10*b*c*d^6 + 8*A^2*a*b^10*c^2*d^5 + 3*A^2*a*b^10*c^
4*d^3 + 8*A^2*a^2*b^9*c*d^6 + 2*A^2*a^3*b^8*c^6*d + 63*A^2*a^4*b^7*c*d^6 +
130*A^2*a^6*b^5*c*d^6 - 9*A^2*a^8*b^3*c*d^6 - 12*B^2*a*b^10*c^2*d^5 + 3*B^2
*a*b^10*c^4*d^3 + 4*B^2*a^2*b^9*c*d^6 - 2*B^2*a^3*b^8*c^6*d + 3*B^2*a^4*b^7
*c*d^6 - 50*B^2*a^6*b^5*c*d^6 + 39*B^2*a^8*b^3*c*d^6 + 3*C^2*a*b^10*c^4*d^3
+ 2*C^2*a^3*b^8*c^6*d + 3*C^2*a^4*b^7*c*d^6 + 54*C^2*a^6*b^5*c*d^6 - 33*C^
2*a^8*b^3*c*d^6 - A*B*a^10*b*d^7 - A*B*b^11*c^6*d + B*C*a^10*b*d^7 + B*C*b^
11*c^6*d + 16*A*B*a*b^10*c*d^6 + 4*A*C*a*b^10*c^6*d - 24*A*B*a*b^10*c^3*d^4
+ 6*A*B*a*b^10*c^5*d^2 + 6*A*B*a^2*b^9*c^6*d + 56*A*B*a^3*b^8*c*d^6 - A*B*
a^4*b^7*c^6*d + 70*A*B*a^5*b^6*c*d^6 - 140*A*B*a^7*b^4*c*d^6 + 6*A*B*a^9*b^
2*c*d^6 - 4*A*C*a*b^10*c^2*d^5 - 6*A*C*a*b^10*c^4*d^3 - 20*A*C*a^2*b^9*c*d^
6 - 4*A*C*a^3*b^8*c^6*d - 74*A*C*a^4*b^7*c*d^6 - 176*A*C*a^6*b^5*c*d^6 + 54
*A*C*a^8*b^3*c*d^6 + 12*B*C*a*b^10*c^3*d^4 - 6*B*C*a*b^10*c^5*d^2 - 6*B*C*a
^2*b^9*c^6*d - 12*B*C*a^3*b^8*c*d^6 + B*C*a^4*b^7*c^6*d - 50*B*C*a^5*b^6*c*
d^6 + 112*B*C*a^7*b^4*c*d^6 - 26*B*C*a^9*b^2*c*d^6 - 20*A*B*a^2*b^9*c^2*d^5
- 15*A*B*a^2*b^9*c^4*d^3 + 100*A*B*a^3*b^8*c^3*d^4 - 36*A*B*a^3*b^8*c^5*d^
2 - 195*A*B*a^4*b^7*c^2*d^5 + 90*A*B*a^4*b^7*c^4*d^3 - 144*A*B*a^5*b^6*c^3*
d^4 + 6*A*B*a^5*b^6*c^5*d^2 + 190*A*B*a^6*b^5*c^2*d^5 - 15*A*B*a^6*b^5*c^4*
d^3 + 20*A*B*a^7*b^4*c^3*d^4 - 15*A*B*a^8*b^3*c^2*d^5 + 48*A*C*a^2*b^9*c^3*
d^4 - 20*A*C*a^2*b^9*c^5*d^2 - 8*A*C*a^3*b^8*c^2*d^5 + 48*A*C*a^3*b^8*c^4*d
^3 - 164*A*C*a^4*b^7*c^3*d^4 + 26*A*C*a^4*b^7*c^5*d^2 + 312*A*C*a^5*b^6*c^2
*d^5 - 66*A*C*a^5*b^6*c^4*d^3 + 72*A*C*a^6*b^5*c^3*d^4 - 60*A*C*a^7*b^4*c^2
*d^5 + 16*B*C*a^2*b^9*c^2*d^5 + 15*B*C*a^2*b^9*c^4*d^3 - 120*B*C*a^3*b^8*c^
3*d^4 + 36*B*C*a^3*b^8*c^5*d^2 + 175*B*C*a^4*b^7*c^2*d^5 - 90*B*C*a^4*b^7*c
^4*d^3 + 140*B*C*a^5*b^6*c^3*d^4 - 6*B*C*a^5*b^6*c^5*d^2 - 202*B*C*a^6*b^5*
c^2*d^5 + 15*B*C*a^6*b^5*c^4*d^3 - 16*B*C*a^7*b^4*c^3*d^4 + 15*B*C*a^8*b^3*
c^2*d^5)/(a^12*d^4 + b^12*c^4 + 4*a^2*b^10*c^4 + 6*a^4*b^8*c^4 + 4*a^6*b^6*
c^4 + a^8*b^4*c^4 + a^4*b^8*d^4 + 4*a^6*b^6*d^4 + 6*a^8*b^4*d^4 + 4*a^10*b^
2*d^4 - 4*a^3*b^9*c*d^3 - 16*a^3*b^9*c^3*d - 16*a^5*b^7*c*d^3 - 24*a^5*b^7*
c^3*d - 24*a^7*b^5*c*d^3 - 16*a^7*b^5*c^3*d - 16*a^9*b^3*c*d^3 - 4*a^9*b^3*
c^3*d + 6*a^2*b^10*c^2*d^2 + 24*a^4*b^8*c^2*d^2 + 36*a^6*b^6*c^2*d^2 + 24*a
^8*b^4*c^2*d^2 + 6*a^10*b^2*c^2*d^2 - 4*a*b^11*c^3*d - 4*a^11*b*c*d^3) + (t
an(e + f*x)*(2*A^2*b^11*d^7 + 6*A^2*a^2*b^9*d^7 - 12*A^2*a^4*b^7*d^7 - 66*A
^2*a^6*b^5*d^7 + 18*A^2*a^8*b^3*d^7 - 2*B^2*a^4*b^7*d^7 + 29*B^2*a^6*b^5*d^
7 - 36*B^2*a^8*b^3*d^7 - 6*A^2*b^11*c^2*d^5 + 2*A^2*b^11*c^4*d^3 + 2*C^2*a^
4*b^7*d^7 - 32*C^2*a^6*b^5*d^7 + 30*C^2*a^8*b^3*d^7 + 2*B^2*b^11*c^2*d^5 +
2*B^2*b^11*c^4*d^3 + 4*C^2*b^11*c^4*d^3 + B^2*a^10*b*d^7 - 4*C^2*a^10*b*d^7

+ B^2*b^11*c^6*d + 38*A^2*a^2*b^9*c^2*d^5 + 4*A^2*a^2*b^9*c^4*d^3 - 16*A^2
*a^3*b^8*c^3*d^4 - 24*A^2*a^3*b^8*c^5*d^2 - 2*A^2*a^4*b^7*c^2*d^5 + 62*A^2*
a^4*b^7*c^4*d^3 - 88*A^2*a^5*b^6*c^3*d^4 + 78*A^2*a^6*b^5*c^2*d^5 - 8*B^2*a
^2*b^9*c^2*d^5 + 19*B^2*a^2*b^9*c^4*d^3 - 46*B^2*a^3*b^8*c^3*d^4 + 12*B^2*a
^3*b^8*c^5*d^2 + 83*B^2*a^4*b^7*c^2*d^5 - 28*B^2*a^4*b^7*c^4*d^3 + 30*B^2*a
^5*b^6*c^3*d^4 - 6*B^2*a^5*b^6*c^5*d^2 - 22*B^2*a^6*b^5*c^2*d^5 + 15*B^2*a^
6*b^5*c^4*d^3 - 18*B^2*a^7*b^4*c^3*d^4 + 9*B^2*a^8*b^3*c^2*d^5 + 12*C^2*a^2
*b^9*c^2*d^5 + 2*C^2*a^2*b^9*c^4*d^3 - 24*C^2*a^3*b^8*c^5*d^2 - 82*C^2*a^4*
b^7*c^2*d^5 + 52*C^2*a^4*b^7*c^4*d^3 - 56*C^2*a^5*b^6*c^3*d^4 + 22*C^2*a^6*
b^5*c^2*d^5 - 6*C^2*a^6*b^5*c^4*d^3 + 16*C^2*a^7*b^4*c^3*d^4 - 6*C^2*a^8*b^
3*c^2*d^5 - 6*A*B*a^3*b^8*d^7 - 18*A*B*a^5*b^6*d^7 + 114*A*B*a^7*b^4*d^7 -
10*A*B*a^9*b^2*d^7 + 14*A*C*a^4*b^7*d^7 + 94*A*C*a^6*b^5*d^7 - 54*A*C*a^8*b
^3*d^7 + 2*A*B*b^11*c^3*d^4 + 24*B*C*a^5*b^6*d^7 - 84*B*C*a^7*b^4*d^7 + 28*
B*C*a^9*b^2*d^7 + 4*A*C*b^11*c^2*d^5 - 6*A*C*b^11*c^4*d^3 - 4*B*C*b^11*c^3*
d^4 - 8*A^2*a*b^10*c*d^6 - 8*A^2*a*b^10*c^3*d^4 + 4*A^2*a^2*b^9*c^6*d - 40*
A^2*a^3*b^8*c*d^6 + 72*A^2*a^5*b^6*c*d^6 - 48*A^2*a^7*b^4*c*d^6 - 14*B^2*a*
b^10*c^3*d^4 - 6*B^2*a*b^10*c^5*d^2 - 2*B^2*a^2*b^9*c^6*d + 14*B^2*a^3*b^8*
c*d^6 + B^2*a^4*b^7*c^6*d - 100*B^2*a^5*b^6*c*d^6 + 38*B^2*a^7*b^4*c*d^6 -
8*C^2*a*b^10*c^3*d^4 + 4*C^2*a^2*b^9*c^6*d - 8*C^2*a^3*b^8*c*d^6 + 104*C^2*
a^5*b^6*c*d^6 - 48*C^2*a^7*b^4*c*d^6 - 8*C^2*a^9*b^2*c*d^6 + 2*C^2*a^10*b*c
^2*d^5 + 2*A*C*a^10*b*d^7 - 4*A*B*b^11*c*d^6 + 4*A*B*a*b^10*c^6*d - 4*B*C*a
*b^10*c^6*d - 2*B*C*a^10*b*c*d^6 + 30*A*B*a*b^10*c^2*d^5 - 10*A*B*a^2*b^9*c
*d^6 - 4*A*B*a^3*b^8*c^6*d + 114*A*B*a^4*b^7*c*d^6 - 166*A*B*a^6*b^5*c*d^6
+ 18*A*B*a^8*b^3*c*d^6 + 16*A*C*a*b^10*c^3*d^4 - 8*A*C*a^2*b^9*c^6*d + 16*A
*C*a^3*b^8*c*d^6 - 224*A*C*a^5*b^6*c*d^6 + 64*A*C*a^7*b^4*c*d^6 + 6*B*C*a*b
^10*c^4*d^3 + 4*B*C*a^3*b^8*c^6*d - 106*B*C*a^4*b^7*c*d^6 + 194*B*C*a^6*b^5
*c*d^6 - 6*B*C*a^8*b^3*c*d^6 - 2*A*B*a^2*b^9*c^3*d^4 - 24*A*B*a^2*b^9*c^5*d
^2 - 54*A*B*a^3*b^8*c^2*d^5 + 60*A*B*a^3*b^8*c^4*d^3 - 90*A*B*a^4*b^7*c^3*d
^4 + 24*A*B*a^4*b^7*c^5*d^2 + 118*A*B*a^5*b^6*c^2*d^5 - 60*A*B*a^5*b^6*c^4*
d^3 + 74*A*B*a^6*b^5*c^3*d^4 - 46*A*B*a^7*b^4*c^2*d^5 - 56*A*C*a^2*b^9*c^2*
d^5 - 6*A*C*a^2*b^9*c^4*d^3 + 16*A*C*a^3*b^8*c^3*d^4 + 48*A*C*a^3*b^8*c^5*d
^2 + 80*A*C*a^4*b^7*c^2*d^5 - 114*A*C*a^4*b^7*c^4*d^3 + 144*A*C*a^5*b^6*c^3
*d^4 - 96*A*C*a^6*b^5*c^2*d^5 + 6*A*C*a^6*b^5*c^4*d^3 - 16*A*C*a^7*b^4*c^3*
d^4 + 12*A*C*a^8*b^3*c^2*d^5 - 14*B*C*a^2*b^9*c^3*d^4 + 24*B*C*a^2*b^9*c^5*
d^2 + 106*B*C*a^3*b^8*c^2*d^5 - 50*B*C*a^3*b^8*c^4*d^3 + 70*B*C*a^4*b^7*c^3
*d^4 - 24*B*C*a^4*b^7*c^5*d^2 - 110*B*C*a^5*b^6*c^2*d^5 + 62*B*C*a^5*b^6*c^
4*d^3 - 74*B*C*a^6*b^5*c^3*d^4 + 26*B*C*a^7*b^4*c^2*d^5 - 2*B*C*a^7*b^4*c^4
*d^3 + 6*B*C*a^8*b^3*c^3*d^4 - 6*B*C*a^9*b^2*c^2*d^5))/(a^12*d^4 + b^12*c^4
+ 4*a^2*b^10*c^4 + 6*a^4*b^8*c^4 + 4*a^6*b^6*c^4 + a^8*b^4*c^4 + a^4*b^8*d
^4 + 4*a^6*b^6*d^4 + 6*a^8*b^4*d^4 + 4*a^10*b^2*d^4 - 4*a^3*b^9*c*d^3 - 16*
a^3*b^9*c^3*d - 16*a^5*b^7*c*d^3 - 24*a^5*b^7*c^3*d - 24*a^7*b^5*c*d^3 - 16
*a^7*b^5*c^3*d - 16*a^9*b^3*c*d^3 - 4*a^9*b^3*c^3*d + 6*a^2*b^10*c^2*d^2 +
24*a^4*b^8*c^2*d^2 + 36*a^6*b^6*c^2*d^2 + 24*a^8*b^4*c^2*d^2 + 6*a^10*b^2*c
^2*d^2 - 4*a*b^11*c^3*d - 4*a^11*b*c*d^3)) - (tan(e + f*x)*(B^3*a^4*b^4*d^6
- A^3*a^3*b^5*d^6 - 3*B^3*a^6*b^2*d^6 - 3*C^3*a^5*b^3*d^6 - B^3*b^8*c^2*d^

4 - A^2*B*b^8*d^6 - A^3*a*b^7*d^6 + A^3*b^8*c*d^5 + C^3*a^7*b*d^6 + 2*B^3*a
^2*b^6*c^2*d^4 - B^3*a^4*b^4*c^2*d^4 + 4*C^3*a^2*b^6*c^3*d^3 - 12*C^3*a^3*b
^5*c^2*d^4 - A*C^2*a^7*b*d^6 + A^2*C*a*b^7*d^6 + 2*A*B^2*b^8*c*d^5 + B^2*C*
a^7*b*d^6 - A^2*C*b^8*c*d^5 + A*B^2*a^3*b^5*d^6 + 9*A*B^2*a^5*b^3*d^6 - 3*A
^2*B*a^2*b^6*d^6 - 6*A^2*B*a^4*b^4*d^6 + 2*A*C^2*a^3*b^5*d^6 + 9*A*C^2*a^5*
b^3*d^6 - A^2*C*a^3*b^5*d^6 - 6*A^2*C*a^5*b^3*d^6 + B*C^2*a^4*b^4*d^6 - 3*B
*C^2*a^6*b^2*d^6 - 3*B^2*C*a^5*b^3*d^6 + B^2*C*b^8*c^3*d^3 + A^3*a^2*b^6*c*
d^5 - 5*B^3*a^3*b^5*c*d^5 + 3*B^3*a^5*b^3*c*d^5 + 11*C^3*a^4*b^4*c*d^5 - C^
3*a^6*b^2*c*d^5 + 4*A*B^2*a^3*b^5*c^2*d^4 - 4*A^2*B*a^2*b^6*c^2*d^4 - 8*A*C
^2*a^2*b^6*c^3*d^3 + 24*A*C^2*a^3*b^5*c^2*d^4 + 4*A^2*C*a^2*b^6*c^3*d^3 - 1
2*A^2*C*a^3*b^5*c^2*d^4 + 8*B*C^2*a^2*b^6*c^2*d^4 + 4*B*C^2*a^3*b^5*c^3*d^3
 - 12*B*C^2*a^4*b^4*c^2*d^4 - 2*B^2*C*a^2*b^6*c^3*d^3 + 2*B^2*C*a^3*b^5*c^2
*d^4 + B^2*C*a^4*b^4*c^3*d^3 - 3*B^2*C*a^5*b^3*c^2*d^4 + 2*A*B*C*a^4*b^4*d^
6 + 2*A*B*C*a^6*b^2*d^6 + A^2*B*a*b^7*c*d^5 - B*C^2*a^7*b*c*d^5 - 4*A*B^2*a
*b^7*c^2*d^4 + 7*A*B^2*a^2*b^6*c*d^5 - 11*A*B^2*a^4*b^4*c*d^5 + 9*A^2*B*a^3
*b^5*c*d^5 - 2*A*C^2*a^2*b^6*c*d^5 - 25*A*C^2*a^4*b^4*c*d^5 + A*C^2*a^6*b^2
*c*d^5 + A^2*C*a^2*b^6*c*d^5 + 14*A^2*C*a^4*b^4*c*d^5 - 4*B*C^2*a*b^7*c^3*d
^3 - 6*B*C^2*a^3*b^5*c*d^5 + 9*B*C^2*a^5*b^3*c*d^5 + B^2*C*a*b^7*c^2*d^4 +
7*B^2*C*a^4*b^4*c*d^5 + 3*B^2*C*a^6*b^2*c*d^5 - 4*A*B*C*a^2*b^6*c^2*d^4 - 4
*A*B*C*a^3*b^5*c^3*d^3 + 12*A*B*C*a^4*b^4*c^2*d^4 - 2*A*B*C*a*b^7*c*d^5 + 4
*A*B*C*a*b^7*c^3*d^3 - 6*A*B*C*a^3*b^5*c*d^5 - 12*A*B*C*a^5*b^3*c*d^5))/(a^
12*d^4 + b^12*c^4 + 4*a^2*b^10*c^4 + 6*a^4*b^8*c^4 + 4*a^6*b^6*c^4 + a^8*b^
4*c^4 + a^4*b^8*d^4 + 4*a^6*b^6*d^4 + 6*a^8*b^4*d^4 + 4*a^10*b^2*d^4 - 4*a^
3*b^9*c*d^3 - 16*a^3*b^9*c^3*d - 16*a^5*b^7*c*d^3 - 24*a^5*b^7*c^3*d - 24*a
^7*b^5*c*d^3 - 16*a^7*b^5*c^3*d - 16*a^9*b^3*c*d^3 - 4*a^9*b^3*c^3*d + 6*a^
2*b^10*c^2*d^2 + 24*a^4*b^8*c^2*d^2 + 36*a^6*b^6*c^2*d^2 + 24*a^8*b^4*c^2*d
^2 + 6*a^10*b^2*c^2*d^2 - 4*a*b^11*c^3*d - 4*a^11*b*c*d^3))*root(480*a^11*b
^7*c*d^9*f^4 + 480*a^7*b^11*c^9*d*f^4 + 360*a^13*b^5*c*d^9*f^4 + 360*a^9*b^
9*c^9*d*f^4 + 360*a^9*b^9*c*d^9*f^4 + 360*a^5*b^13*c^9*d*f^4 + 144*a^15*b^3
*c*d^9*f^4 + 144*a^11*b^7*c^9*d*f^4 + 144*a^7*b^11*c*d^9*f^4 + 144*a^3*b^15
*c^9*d*f^4 + 48*a^17*b*c^3*d^7*f^4 + 48*a*b^17*c^7*d^3*f^4 + 24*a^17*b*c^5*
d^5*f^4 + 24*a^13*b^5*c^9*d*f^4 + 24*a^5*b^13*c*d^9*f^4 + 24*a*b^17*c^5*d^5
*f^4 + 24*a^17*b*c*d^9*f^4 + 24*a*b^17*c^9*d*f^4 + 3920*a^9*b^9*c^5*d^5*f^4
 - 3360*a^10*b^8*c^4*d^6*f^4 - 3360*a^8*b^10*c^6*d^4*f^4 + 3024*a^11*b^7*c^
5*d^5*f^4 - 3024*a^10*b^8*c^6*d^4*f^4 - 3024*a^8*b^10*c^4*d^6*f^4 + 3024*a^
7*b^11*c^5*d^5*f^4 + 2320*a^9*b^9*c^7*d^3*f^4 + 2320*a^9*b^9*c^3*d^7*f^4 -
2240*a^12*b^6*c^4*d^6*f^4 - 2240*a^6*b^12*c^6*d^4*f^4 + 2160*a^11*b^7*c^3*d
^7*f^4 + 2160*a^7*b^11*c^7*d^3*f^4 - 1624*a^12*b^6*c^6*d^4*f^4 - 1624*a^6*b
^12*c^4*d^6*f^4 + 1488*a^11*b^7*c^7*d^3*f^4 + 1488*a^7*b^11*c^3*d^7*f^4 + 1
344*a^13*b^5*c^5*d^5*f^4 + 1344*a^5*b^13*c^5*d^5*f^4 - 1320*a^10*b^8*c^2*d^
8*f^4 - 1320*a^8*b^10*c^8*d^2*f^4 + 1200*a^13*b^5*c^3*d^7*f^4 + 1200*a^5*b^
13*c^7*d^3*f^4 - 1060*a^12*b^6*c^2*d^8*f^4 - 1060*a^6*b^12*c^8*d^2*f^4 - 94
8*a^10*b^8*c^8*d^2*f^4 - 948*a^8*b^10*c^2*d^8*f^4 - 840*a^14*b^4*c^4*d^6*f^
4 - 840*a^4*b^14*c^6*d^4*f^4 + 528*a^13*b^5*c^7*d^3*f^4 + 528*a^5*b^13*c^3*
d^7*f^4 - 480*a^14*b^4*c^6*d^4*f^4 - 480*a^14*b^4*c^2*d^8*f^4 - 480*a^4*b^1

4*c^8*d^2*f^4 - 480*a^4*b^14*c^4*d^6*f^4 + 368*a^15*b^3*c^3*d^7*f^4 - 368*a
^12*b^6*c^8*d^2*f^4 - 368*a^6*b^12*c^2*d^8*f^4 + 368*a^3*b^15*c^7*d^3*f^4 +
304*a^15*b^3*c^5*d^5*f^4 + 304*a^3*b^15*c^5*d^5*f^4 - 144*a^16*b^2*c^4*d^6
*f^4 - 144*a^2*b^16*c^6*d^4*f^4 - 108*a^16*b^2*c^2*d^8*f^4 - 108*a^2*b^16*c
^8*d^2*f^4 + 80*a^15*b^3*c^7*d^3*f^4 + 80*a^3*b^15*c^3*d^7*f^4 - 60*a^16*b^
2*c^6*d^4*f^4 - 60*a^14*b^4*c^8*d^2*f^4 - 60*a^4*b^14*c^2*d^8*f^4 - 60*a^2*
b^16*c^4*d^6*f^4 - 8*b^18*c^8*d^2*f^4 - 4*b^18*c^6*d^4*f^4 - 8*a^18*c^2*d^8
*f^4 - 4*a^18*c^4*d^6*f^4 - 80*a^12*b^6*d^10*f^4 - 60*a^14*b^4*d^10*f^4 - 6
0*a^10*b^8*d^10*f^4 - 24*a^16*b^2*d^10*f^4 - 24*a^8*b^10*d^10*f^4 - 4*a^6*b
^12*d^10*f^4 - 80*a^6*b^12*c^10*f^4 - 60*a^8*b^10*c^10*f^4 - 60*a^4*b^14*c^
10*f^4 - 24*a^10*b^8*c^10*f^4 - 24*a^2*b^16*c^10*f^4 - 4*a^12*b^6*c^10*f^4
- 4*b^18*c^10*f^4 - 4*a^18*d^10*f^4 - 12*A*C*a^11*b*c*d^7*f^2 - 12*A*C*a*b^
11*c^7*d*f^2 - 912*B*C*a^5*b^7*c^4*d^4*f^2 - 792*B*C*a^8*b^4*c^3*d^5*f^2 +
792*B*C*a^4*b^8*c^5*d^3*f^2 + 720*B*C*a^7*b^5*c^4*d^4*f^2 - 480*B*C*a^5*b^7
*c^6*d^2*f^2 - 408*B*C*a^5*b^7*c^2*d^6*f^2 + 384*B*C*a^7*b^5*c^2*d^6*f^2 -
336*B*C*a^8*b^4*c^5*d^3*f^2 + 324*B*C*a^4*b^8*c^3*d^5*f^2 + 312*B*C*a^7*b^5
*c^6*d^2*f^2 - 248*B*C*a^3*b^9*c^6*d^2*f^2 + 216*B*C*a^9*b^3*c^2*d^6*f^2 -
196*B*C*a^3*b^9*c^4*d^4*f^2 + 132*B*C*a^9*b^3*c^4*d^4*f^2 + 80*B*C*a^6*b^6*
c^3*d^5*f^2 - 64*B*C*a^6*b^6*c^5*d^3*f^2 - 36*B*C*a^2*b^10*c^3*d^5*f^2 - 28
*B*C*a^3*b^9*c^2*d^6*f^2 + 12*B*C*a^10*b^2*c^5*d^3*f^2 - 12*B*C*a^10*b^2*c^
3*d^5*f^2 - 12*B*C*a^2*b^10*c^5*d^3*f^2 - 4*B*C*a^9*b^3*c^6*d^2*f^2 - 1468*
A*C*a^6*b^6*c^4*d^4*f^2 + 996*A*C*a^7*b^5*c^3*d^5*f^2 + 900*A*C*a^5*b^7*c^5
*d^3*f^2 - 676*A*C*a^6*b^6*c^6*d^2*f^2 - 660*A*C*a^6*b^6*c^2*d^6*f^2 + 636*
A*C*a^5*b^7*c^3*d^5*f^2 + 540*A*C*a^7*b^5*c^5*d^3*f^2 - 236*A*C*a^3*b^9*c^5
*d^3*f^2 - 204*A*C*a^9*b^3*c^3*d^5*f^2 + 156*A*C*a^10*b^2*c^2*d^6*f^2 + 132
*A*C*a^2*b^10*c^6*d^2*f^2 - 72*A*C*a^9*b^3*c^5*d^3*f^2 - 72*A*C*a^4*b^8*c^6
*d^2*f^2 + 66*A*C*a^4*b^8*c^2*d^6*f^2 + 54*A*C*a^10*b^2*c^4*d^4*f^2 + 54*A*
C*a^2*b^10*c^4*d^4*f^2 - 48*A*C*a^8*b^4*c^2*d^6*f^2 - 48*A*C*a^4*b^8*c^4*d^
4*f^2 + 42*A*C*a^8*b^4*c^6*d^2*f^2 - 40*A*C*a^3*b^9*c^3*d^5*f^2 - 36*A*C*a^
8*b^4*c^4*d^4*f^2 + 24*A*C*a^2*b^10*c^2*d^6*f^2 + 960*A*B*a^5*b^7*c^4*d^4*f
^2 - 864*A*B*a^4*b^8*c^5*d^3*f^2 + 756*A*B*a^8*b^4*c^3*d^5*f^2 - 744*A*B*a^
7*b^5*c^4*d^4*f^2 - 528*A*B*a^4*b^8*c^3*d^5*f^2 + 504*A*B*a^5*b^7*c^6*d^2*f
^2 - 432*A*B*a^7*b^5*c^2*d^6*f^2 + 432*A*B*a^5*b^7*c^2*d^6*f^2 + 348*A*B*a^
8*b^4*c^5*d^3*f^2 - 312*A*B*a^7*b^5*c^6*d^2*f^2 - 284*A*B*a^9*b^3*c^2*d^6*f
^2 + 280*A*B*a^3*b^9*c^6*d^2*f^2 + 264*A*B*a^3*b^9*c^4*d^4*f^2 - 240*A*B*a^
6*b^6*c^3*d^5*f^2 - 172*A*B*a^9*b^3*c^4*d^4*f^2 + 68*A*B*a^3*b^9*c^2*d^6*f^
2 - 60*A*B*a^2*b^10*c^3*d^5*f^2 + 24*A*B*a^6*b^6*c^5*d^3*f^2 - 24*A*B*a^2*b
^10*c^5*d^3*f^2 + 12*A*B*a^10*b^2*c^3*d^5*f^2 + 360*B*C*a^4*b^8*c^7*d*f^2 -
336*B*C*a^8*b^4*c*d^7*f^2 + 168*B*C*a^6*b^6*c*d^7*f^2 - 136*B*C*a^6*b^6*c^
7*d*f^2 - 36*B*C*a^11*b*c^2*d^6*f^2 + 36*B*C*a*b^11*c^6*d^2*f^2 + 24*B*C*a^
10*b^2*c*d^7*f^2 - 24*B*C*a^2*b^10*c^7*d*f^2 - 12*B*C*a^11*b*c^4*d^4*f^2 +
12*B*C*a^4*b^8*c*d^7*f^2 + 12*B*C*a*b^11*c^4*d^4*f^2 + 444*A*C*a^7*b^5*c*d^
7*f^2 + 348*A*C*a^5*b^7*c^7*d*f^2 - 164*A*C*a^3*b^9*c^7*d*f^2 - 132*A*C*a^9
*b^3*c*d^7*f^2 + 84*A*C*a^5*b^7*c*d^7*f^2 + 32*A*C*a^3*b^9*c*d^7*f^2 - 12*A
*C*a^11*b*c^3*d^5*f^2 - 12*A*C*a^7*b^5*c^7*d*f^2 - 12*A*C*a*b^11*c^5*d^3*f^

2 - 360*A*B*a^4*b^8*c^7*d*f^2 + 288*A*B*a^8*b^4*c*d^7*f^2 - 288*A*B*a^6*b^6
*c*d^7*f^2 - 144*A*B*a^4*b^8*c*d^7*f^2 + 136*A*B*a^6*b^6*c^7*d*f^2 - 60*A*B
*a^2*b^10*c*d^7*f^2 - 36*A*B*a^10*b^2*c*d^7*f^2 + 24*A*B*a^2*b^10*c^7*d*f^2
 - 24*A*B*a*b^11*c^6*d^2*f^2 + 12*A*B*a^11*b*c^2*d^6*f^2 + 12*A*B*a*b^11*c^
4*d^4*f^2 + 12*A*B*a*b^11*c^2*d^6*f^2 - 8*B*C*b^12*c^5*d^3*f^2 - 8*B*C*b^12
*c^3*d^5*f^2 + 8*A*C*b^12*c^2*d^6*f^2 - 4*B*C*a^12*c^3*d^5*f^2 + 4*A*C*b^12
*c^4*d^4*f^2 - 2*A*C*b^12*c^6*d^2*f^2 + 80*B*C*a^9*b^3*d^8*f^2 - 24*B*C*a^7
*b^5*d^8*f^2 + 6*A*C*a^12*c^2*d^6*f^2 + 4*A*B*b^12*c^5*d^3*f^2 - 4*A*B*b^12
*c^3*d^5*f^2 - 90*A*C*a^8*b^4*d^8*f^2 - 80*B*C*a^3*b^9*c^8*f^2 + 54*A*C*a^1
0*b^2*d^8*f^2 - 30*A*C*a^6*b^6*d^8*f^2 + 24*B*C*a^5*b^7*c^8*f^2 - 12*A*C*a^
4*b^8*d^8*f^2 - 112*A*B*a^9*b^3*d^8*f^2 - 66*A*C*a^4*b^8*c^8*f^2 + 54*A*C*a
^2*b^10*c^8*f^2 + 4*A*B*a^3*b^9*d^8*f^2 + 2*A*C*a^6*b^6*c^8*f^2 + 80*A*B*a^
3*b^9*c^8*f^2 - 24*A*B*a^5*b^7*c^8*f^2 + 726*C^2*a^6*b^6*c^4*d^4*f^2 - 402*
C^2*a^7*b^5*c^3*d^5*f^2 - 402*C^2*a^5*b^7*c^5*d^3*f^2 + 322*C^2*a^6*b^6*c^6
*d^2*f^2 + 322*C^2*a^6*b^6*c^2*d^6*f^2 - 222*C^2*a^7*b^5*c^5*d^3*f^2 - 222*
C^2*a^5*b^7*c^3*d^5*f^2 + 134*C^2*a^9*b^3*c^3*d^5*f^2 + 134*C^2*a^3*b^9*c^5
*d^3*f^2 - 66*C^2*a^10*b^2*c^2*d^6*f^2 - 66*C^2*a^2*b^10*c^6*d^2*f^2 + 52*C
^2*a^9*b^3*c^5*d^3*f^2 + 52*C^2*a^3*b^9*c^3*d^5*f^2 - 27*C^2*a^8*b^4*c^6*d^
2*f^2 - 27*C^2*a^4*b^8*c^2*d^6*f^2 + 24*C^2*a^8*b^4*c^4*d^4*f^2 + 24*C^2*a^
8*b^4*c^2*d^6*f^2 + 24*C^2*a^4*b^8*c^6*d^2*f^2 + 24*C^2*a^4*b^8*c^4*d^4*f^2
 - 15*C^2*a^10*b^2*c^4*d^4*f^2 - 15*C^2*a^2*b^10*c^4*d^4*f^2 - 570*B^2*a^6*
b^6*c^4*d^4*f^2 + 366*B^2*a^7*b^5*c^3*d^5*f^2 + 318*B^2*a^5*b^7*c^5*d^3*f^2
 - 262*B^2*a^6*b^6*c^6*d^2*f^2 - 222*B^2*a^6*b^6*c^2*d^6*f^2 - 210*B^2*a^3*
b^9*c^5*d^3*f^2 + 186*B^2*a^7*b^5*c^5*d^3*f^2 + 162*B^2*a^5*b^7*c^3*d^5*f^2
 - 142*B^2*a^9*b^3*c^3*d^5*f^2 + 132*B^2*a^4*b^8*c^4*d^4*f^2 + 117*B^2*a^4*
b^8*c^2*d^6*f^2 + 102*B^2*a^2*b^10*c^6*d^2*f^2 - 96*B^2*a^3*b^9*c^3*d^5*f^2
 + 90*B^2*a^10*b^2*c^2*d^6*f^2 + 81*B^2*a^2*b^10*c^4*d^4*f^2 - 56*B^2*a^9*b
^3*c^5*d^3*f^2 + 48*B^2*a^8*b^4*c^4*d^4*f^2 + 48*B^2*a^4*b^8*c^6*d^2*f^2 +
45*B^2*a^8*b^4*c^6*d^2*f^2 + 36*B^2*a^8*b^4*c^2*d^6*f^2 + 36*B^2*a^2*b^10*c
^2*d^6*f^2 + 33*B^2*a^10*b^2*c^4*d^4*f^2 + 822*A^2*a^6*b^6*c^4*d^4*f^2 - 59
4*A^2*a^7*b^5*c^3*d^5*f^2 + 498*A^2*a^6*b^6*c^2*d^6*f^2 - 498*A^2*a^5*b^7*c
^5*d^3*f^2 - 414*A^2*a^5*b^7*c^3*d^5*f^2 + 354*A^2*a^6*b^6*c^6*d^2*f^2 - 31
8*A^2*a^7*b^5*c^5*d^3*f^2 + 144*A^2*a^8*b^4*c^2*d^6*f^2 + 102*A^2*a^3*b^9*c
^5*d^3*f^2 + 84*A^2*a^4*b^8*c^4*d^4*f^2 + 81*A^2*a^4*b^8*c^2*d^6*f^2 + 72*A
^2*a^8*b^4*c^4*d^4*f^2 + 70*A^2*a^9*b^3*c^3*d^5*f^2 - 66*A^2*a^2*b^10*c^6*d
^2*f^2 + 48*A^2*a^4*b^8*c^6*d^2*f^2 - 42*A^2*a^10*b^2*c^2*d^6*f^2 + 24*A^2*
a^2*b^10*c^2*d^6*f^2 + 20*A^2*a^9*b^3*c^5*d^3*f^2 - 15*A^2*a^10*b^2*c^4*d^4
*f^2 - 15*A^2*a^8*b^4*c^6*d^2*f^2 - 15*A^2*a^2*b^10*c^4*d^4*f^2 - 12*A^2*a^
3*b^9*c^3*d^5*f^2 - 8*B*C*b^12*c^7*d*f^2 + 4*B*C*a^12*c*d^7*f^2 - 24*B*C*a^
11*b*d^8*f^2 + 8*A*B*b^12*c^7*d*f^2 - 8*A*B*b^12*c*d^7*f^2 + 24*B*C*a*b^11*
c^8*f^2 - 8*A*B*a^12*c*d^7*f^2 + 12*A*B*a^11*b*d^8*f^2 - 24*A*B*a*b^11*c^8*
f^2 - 174*C^2*a^7*b^5*c*d^7*f^2 - 174*C^2*a^5*b^7*c^7*d*f^2 + 82*C^2*a^9*b^
3*c*d^7*f^2 + 82*C^2*a^3*b^9*c^7*d*f^2 + 6*C^2*a^11*b*c^3*d^5*f^2 + 6*C^2*a
^7*b^5*c^7*d*f^2 + 6*C^2*a^5*b^7*c*d^7*f^2 + 6*C^2*a*b^11*c^5*d^3*f^2 + 162
*B^2*a^7*b^5*c*d^7*f^2 + 138*B^2*a^5*b^7*c^7*d*f^2 - 118*B^2*a^3*b^9*c^7*d*

f^2 - 86*B^2*a^9*b^3*c*d^7*f^2 - 30*B^2*a*b^11*c^5*d^3*f^2 - 18*B^2*a^7*b^5
*c^7*d*f^2 - 18*B^2*a^5*b^7*c*d^7*f^2 - 12*B^2*a*b^11*c^3*d^5*f^2 - 6*B^2*a
^11*b*c^3*d^5*f^2 - 4*B^2*a^3*b^9*c*d^7*f^2 - 270*A^2*a^7*b^5*c*d^7*f^2 - 1
74*A^2*a^5*b^7*c^7*d*f^2 - 90*A^2*a^5*b^7*c*d^7*f^2 + 82*A^2*a^3*b^9*c^7*d*
f^2 + 50*A^2*a^9*b^3*c*d^7*f^2 - 32*A^2*a^3*b^9*c*d^7*f^2 + 6*A^2*a^11*b*c^
3*d^5*f^2 + 6*A^2*a^7*b^5*c^7*d*f^2 + 6*A^2*a*b^11*c^5*d^3*f^2 + 6*C^2*a^11
*b*c*d^7*f^2 + 6*C^2*a*b^11*c^7*d*f^2 - 18*B^2*a*b^11*c^7*d*f^2 - 6*B^2*a^1
1*b*c*d^7*f^2 + 6*A^2*a^11*b*c*d^7*f^2 + 6*A^2*a*b^11*c^7*d*f^2 - 6*A*C*b^1
2*c^8*f^2 - 2*A*C*a^12*d^8*f^2 + 4*C^2*b^12*c^4*d^4*f^2 + 3*C^2*b^12*c^6*d^
2*f^2 + 4*C^2*a^12*c^4*d^4*f^2 + 4*B^2*b^12*c^4*d^4*f^2 + 4*B^2*b^12*c^2*d^
6*f^2 + 3*C^2*a^12*c^2*d^6*f^2 + 3*B^2*b^12*c^6*d^2*f^2 + 33*C^2*a^8*b^4*d^
8*f^2 - 27*C^2*a^10*b^2*d^8*f^2 - 4*A^2*b^12*c^4*d^4*f^2 + 3*B^2*a^12*c^2*d
^6*f^2 - C^2*a^6*b^6*d^8*f^2 - A^2*b^12*c^6*d^2*f^2 + 33*C^2*a^4*b^8*c^8*f^
2 + 33*B^2*a^10*b^2*d^8*f^2 - 27*C^2*a^2*b^10*c^8*f^2 - 27*B^2*a^8*b^4*d^8*
f^2 + 3*B^2*a^6*b^6*d^8*f^2 - C^2*a^6*b^6*c^8*f^2 - A^2*a^12*c^2*d^6*f^2 +
117*A^2*a^8*b^4*d^8*f^2 + 111*A^2*a^6*b^6*d^8*f^2 + 72*A^2*a^4*b^8*d^8*f^2
+ 33*B^2*a^2*b^10*c^8*f^2 - 27*B^2*a^4*b^8*c^8*f^2 + 24*A^2*a^2*b^10*d^8*f^
2 + 3*B^2*a^6*b^6*c^8*f^2 - 3*A^2*a^10*b^2*d^8*f^2 + 33*A^2*a^4*b^8*c^8*f^2
 - 27*A^2*a^2*b^10*c^8*f^2 - A^2*a^6*b^6*c^8*f^2 + 3*C^2*b^12*c^8*f^2 + 3*C
^2*a^12*d^8*f^2 + 4*A^2*b^12*d^8*f^2 - B^2*b^12*c^8*f^2 - B^2*a^12*d^8*f^2
+ 3*A^2*b^12*c^8*f^2 + 3*A^2*a^12*d^8*f^2 - 24*A*B*C*a*b^8*c*d^6*f + 342*A*
B*C*a^4*b^5*c^2*d^5*f - 186*A*B*C*a^5*b^4*c^3*d^4*f - 66*A*B*C*a^2*b^7*c^4*
d^3*f + 48*A*B*C*a^2*b^7*c^2*d^5*f + 42*A*B*C*a^6*b^3*c^2*d^5*f + 26*A*B*C*
a^3*b^6*c^5*d^2*f + 24*A*B*C*a^6*b^3*c^4*d^3*f - 18*A*B*C*a^7*b^2*c^3*d^4*f
 - 18*A*B*C*a^4*b^5*c^4*d^3*f - 8*A*B*C*a^3*b^6*c^3*d^4*f + 6*A*B*C*a^5*b^4
*c^5*d^2*f - 128*A*B*C*a^3*b^6*c*d^6*f + 126*A*B*C*a^7*b^2*c*d^6*f + 72*A*B
*C*a*b^8*c^3*d^4*f - 36*A*B*C*a^8*b*c^2*d^5*f - 36*A*B*C*a*b^8*c^5*d^2*f +
30*A*B*C*a^2*b^7*c^6*d*f - 12*A*B*C*a^5*b^4*c*d^6*f - 12*A*B*C*a^4*b^5*c^6*
d*f - 21*B^2*C*a^8*b*c*d^6*f - 3*B^2*C*a*b^8*c^6*d*f + 21*A^2*C*a^8*b*c*d^6
*f - 21*A*C^2*a^8*b*c*d^6*f - 9*A^2*C*a*b^8*c^6*d*f + 9*A*C^2*a*b^8*c^6*d*f
 + 36*A^2*B*a*b^8*c*d^6*f + 21*A*B^2*a^8*b*c*d^6*f + 3*A*B^2*a*b^8*c^6*d*f
+ 16*A*B*C*b^9*c^4*d^3*f - 16*A*B*C*b^9*c^2*d^5*f - 78*A*B*C*a^6*b^3*d^7*f
+ 24*A*B*C*a^4*b^5*d^7*f + 2*A*B*C*a^3*b^6*c^7*f - 237*B^2*C*a^4*b^5*c^3*d^
4*f + 165*B*C^2*a^5*b^4*c^3*d^4*f + 92*B^2*C*a^3*b^6*c^2*d^5*f - 81*B^2*C*a
^7*b^2*c^2*d^5*f + 77*B^2*C*a^3*b^6*c^4*d^3*f - 75*B*C^2*a^4*b^5*c^2*d^5*f
+ 69*B^2*C*a^5*b^4*c^4*d^3*f + 69*B*C^2*a^4*b^5*c^4*d^3*f - 68*B*C^2*a^3*b^
6*c^3*d^4*f - 63*B^2*C*a^4*b^5*c^5*d^2*f - 61*B*C^2*a^6*b^3*c^2*d^5*f + 57*
B*C^2*a^2*b^7*c^4*d^3*f - 53*B*C^2*a^3*b^6*c^5*d^2*f - 44*B*C^2*a^6*b^3*c^4
*d^3*f - 36*B^2*C*a^2*b^7*c^3*d^4*f + 35*B^2*C*a^6*b^3*c^3*d^4*f - 33*B^2*C
*a^5*b^4*c^2*d^5*f + 33*B^2*C*a^2*b^7*c^5*d^2*f + 33*B*C^2*a^7*b^2*c^3*d^4*
f - 12*B^2*C*a^7*b^2*c^4*d^3*f + 9*B*C^2*a^5*b^4*c^5*d^2*f + 4*B^2*C*a^6*b^
3*c^5*d^2*f + 225*A^2*C*a^5*b^4*c^2*d^5*f - 105*A*C^2*a^5*b^4*c^2*d^5*f - 9
9*A^2*C*a^4*b^5*c^3*d^4*f - 81*A^2*C*a^4*b^5*c^5*d^2*f + 67*A^2*C*a^3*b^6*c
^4*d^3*f - 59*A*C^2*a^3*b^6*c^4*d^3*f - 57*A*C^2*a^7*b^2*c^2*d^5*f + 57*A*C
^2*a^2*b^7*c^5*d^2*f + 51*A^2*C*a^5*b^4*c^4*d^3*f + 48*A^2*C*a^2*b^7*c^3*d^

4*f + 45*A*C^2*a^4*b^5*c^5*d^2*f - 35*A^2*C*a^6*b^3*c^3*d^4*f + 33*A^2*C*a^
7*b^2*c^2*d^5*f - 33*A^2*C*a^2*b^7*c^5*d^2*f + 33*A*C^2*a^5*b^4*c^4*d^3*f +
27*A*C^2*a^6*b^3*c^3*d^4*f + 24*A*C^2*a^3*b^6*c^2*d^5*f - 24*A*C^2*a^2*b^7
*c^3*d^4*f - 21*A*C^2*a^4*b^5*c^3*d^4*f - 16*A^2*C*a^3*b^6*c^2*d^5*f - 243*
A^2*B*a^4*b^5*c^2*d^5*f - 156*A*B^2*a^3*b^6*c^2*d^5*f + 141*A*B^2*a^4*b^5*c
^3*d^4*f + 108*A^2*B*a^3*b^6*c^3*d^4*f - 105*A*B^2*a^3*b^6*c^4*d^3*f + 84*A
*B^2*a^2*b^7*c^3*d^4*f + 81*A*B^2*a^5*b^4*c^2*d^5*f + 51*A^2*B*a^6*b^3*c^2*
d^5*f - 51*A^2*B*a^4*b^5*c^4*d^3*f - 48*A^2*B*a^2*b^7*c^2*d^5*f + 45*A^2*B*
a^5*b^4*c^3*d^4*f + 39*A*B^2*a^4*b^5*c^5*d^2*f - 35*A*B^2*a^6*b^3*c^3*d^4*f
+ 33*A*B^2*a^7*b^2*c^2*d^5*f + 27*A^2*B*a^3*b^6*c^5*d^2*f - 21*A*B^2*a^5*b
^4*c^4*d^3*f + 20*A^2*B*a^6*b^3*c^4*d^3*f - 15*A^2*B*a^7*b^2*c^3*d^4*f - 15
*A^2*B*a^5*b^4*c^5*d^2*f + 9*A^2*B*a^2*b^7*c^4*d^3*f + 3*A*B^2*a^2*b^7*c^5*
d^2*f + 2*A*B*C*b^9*c^6*d*f - 6*A*B*C*a^9*c*d^6*f + 18*A*B*C*a^8*b*d^7*f -
6*A*B*C*a*b^8*c^7*f + 63*B^2*C*a^6*b^3*c*d^6*f - 48*B^2*C*a*b^8*c^4*d^3*f +
42*B*C^2*a^8*b*c^2*d^5*f + 42*B*C^2*a^5*b^4*c*d^6*f - 39*B*C^2*a^7*b^2*c*d
^6*f + 30*B*C^2*a*b^8*c^5*d^2*f - 24*B^2*C*a^4*b^5*c*d^6*f - 24*B*C^2*a*b^8
*c^3*d^4*f + 17*B^2*C*a^3*b^6*c^6*d*f - 15*B*C^2*a^2*b^7*c^6*d*f + 12*B^2*C
*a^8*b*c^3*d^4*f + 12*B^2*C*a*b^8*c^2*d^5*f + 6*B*C^2*a^4*b^5*c^6*d*f - 192
*A^2*C*a^4*b^5*c*d^6*f - 99*A^2*C*a^6*b^3*c*d^6*f + 84*A*C^2*a^4*b^5*c*d^6*
f + 59*A*C^2*a^6*b^3*c*d^6*f + 51*A^2*C*a^3*b^6*c^6*d*f - 51*A*C^2*a^3*b^6*
c^6*d*f - 36*A^2*C*a*b^8*c^2*d^5*f - 24*A*C^2*a*b^8*c^4*d^3*f + 24*A*C^2*a*
b^8*c^2*d^5*f + 12*A^2*C*a*b^8*c^4*d^3*f + 12*A*C^2*a^8*b*c^3*d^4*f + 160*A
^2*B*a^3*b^6*c*d^6*f - 99*A*B^2*a^6*b^3*c*d^6*f - 87*A^2*B*a^7*b^2*c*d^6*f
- 72*A*B^2*a^4*b^5*c*d^6*f - 48*A*B^2*a*b^8*c^2*d^5*f - 36*A^2*B*a*b^8*c^3*
d^4*f + 24*A*B^2*a*b^8*c^4*d^3*f - 17*A*B^2*a^3*b^6*c^6*d*f - 15*A^2*B*a^2*
b^7*c^6*d*f + 12*A*B^2*a^2*b^7*c*d^6*f + 6*A^2*B*a^8*b*c^2*d^5*f - 6*A^2*B*
a^5*b^4*c*d^6*f + 6*A^2*B*a^4*b^5*c^6*d*f + 6*A^2*B*a*b^8*c^5*d^2*f + 12*B^
2*C*b^9*c^3*d^4*f - 12*B*C^2*b^9*c^4*d^3*f - 12*A^2*C*b^9*c^3*d^4*f - 8*A*C
^2*b^9*c^5*d^2*f + 8*A*C^2*b^9*c^3*d^4*f + 4*B^2*C*a^9*c^2*d^5*f + 4*A^2*C*
b^9*c^5*d^2*f - 4*B*C^2*a^9*c^3*d^4*f + 12*A^2*B*b^9*c^2*d^5*f - 8*A*B^2*b^
9*c^3*d^4*f - 4*A^2*B*b^9*c^4*d^3*f + 4*A*C^2*a^9*c^2*d^5*f + 3*B^2*C*a^7*b
^2*d^7*f - B*C^2*a^6*b^3*d^7*f + 96*A^2*C*a^5*b^4*d^7*f - 39*A^2*C*a^7*b^2*
d^7*f - 36*A*C^2*a^5*b^4*d^7*f + 32*A^2*C*a^3*b^6*d^7*f + 15*A*C^2*a^7*b^2*
d^7*f - 3*B^2*C*a^2*b^7*c^7*f - B*C^2*a^3*b^6*c^7*f + 111*A^2*B*a^6*b^3*d^7
*f - 39*A*B^2*a^7*b^2*d^7*f + 24*A*B^2*a^5*b^4*d^7*f - 9*A^2*C*a^2*b^7*c^7*
f + 9*A*C^2*a^2*b^7*c^7*f - 4*A*B^2*a^3*b^6*d^7*f + 3*A*B^2*a^2*b^7*c^7*f -
A^2*B*a^3*b^6*c^7*f + 3*C^3*a^8*b*c*d^6*f - 3*C^3*a*b^8*c^6*d*f - 3*A^3*a^
8*b*c*d^6*f + 3*A^3*a*b^8*c^6*d*f - B*C^2*b^9*c^6*d*f + 4*A^2*C*b^9*c*d^6*f
+ 3*B*C^2*a^9*c*d^6*f + 8*A*B^2*b^9*c*d^6*f + 3*B*C^2*a^8*b*d^7*f - A^2*B*
b^9*c^6*d*f + 12*A^2*C*a*b^8*d^7*f + 3*B*C^2*a*b^8*c^7*f - A^2*B*a^9*c*d^6*
f - 9*A^2*B*a^8*b*d^7*f + 3*A^2*B*a*b^8*c^7*f - 39*C^3*a^5*b^4*c^4*d^3*f +
39*C^3*a^4*b^5*c^3*d^4*f + 27*C^3*a^7*b^2*c^2*d^5*f - 27*C^3*a^2*b^7*c^5*d^
2*f - 17*C^3*a^6*b^3*c^3*d^4*f + 17*C^3*a^3*b^6*c^4*d^3*f + 3*C^3*a^5*b^4*c
^2*d^5*f - 3*C^3*a^4*b^5*c^5*d^2*f - 63*B^3*a^5*b^4*c^3*d^4*f + 57*B^3*a^4*
b^5*c^2*d^5*f - 51*B^3*a^2*b^7*c^4*d^3*f + 48*B^3*a^3*b^6*c^3*d^4*f + 31*B^

3*a^6*b^3*c^2*d^5*f + 27*B^3*a^3*b^6*c^5*d^2*f + 16*B^3*a^6*b^3*c^4*d^3*f -
15*B^3*a^5*b^4*c^5*d^2*f - 12*B^3*a^2*b^7*c^2*d^5*f + 9*B^3*a^4*b^5*c^4*d^
3*f - 3*B^3*a^7*b^2*c^3*d^4*f - 123*A^3*a^5*b^4*c^2*d^5*f + 81*A^3*a^4*b^5*
c^3*d^4*f - 45*A^3*a^5*b^4*c^4*d^3*f + 39*A^3*a^4*b^5*c^5*d^2*f + 25*A^3*a^
6*b^3*c^3*d^4*f - 25*A^3*a^3*b^6*c^4*d^3*f - 24*A^3*a^2*b^7*c^3*d^4*f - 8*A
^3*a^3*b^6*c^2*d^5*f - 3*A^3*a^7*b^2*c^2*d^5*f + 3*A^3*a^2*b^7*c^5*d^2*f -
17*C^3*a^6*b^3*c*d^6*f + 17*C^3*a^3*b^6*c^6*d*f - 12*C^3*a^8*b*c^3*d^4*f +
12*C^3*a*b^8*c^4*d^3*f + 24*B^3*a*b^8*c^3*d^4*f + 21*B^3*a^7*b^2*c*d^6*f -
18*B^3*a^5*b^4*c*d^6*f - 15*B^3*a^2*b^7*c^6*d*f - 6*B^3*a^8*b*c^2*d^5*f + 6
*B^3*a^4*b^5*c^6*d*f + 6*B^3*a*b^8*c^5*d^2*f + 4*B^3*a^3*b^6*c*d^6*f + 108*
A^3*a^4*b^5*c*d^6*f + 57*A^3*a^6*b^3*c*d^6*f - 17*A^3*a^3*b^6*c^6*d*f + 12*
A^3*a*b^8*c^2*d^5*f + 4*C^3*b^9*c^5*d^2*f - 4*C^3*a^9*c^2*d^5*f - 4*B^3*b^9
*c^2*d^5*f + 4*A^3*b^9*c^3*d^4*f + 3*C^3*a^7*b^2*d^7*f - 3*C^3*a^2*b^7*c^7*
f - B^3*a^6*b^3*d^7*f - 60*A^3*a^5*b^4*d^7*f - 32*A^3*a^3*b^6*d^7*f + 21*A^
3*a^7*b^2*d^7*f - B^3*a^3*b^6*c^7*f + 3*A^3*a^2*b^7*c^7*f - B^3*b^9*c^6*d*f
- 4*A^3*b^9*c*d^6*f - B^3*a^9*c*d^6*f + 3*B^3*a^8*b*d^7*f - 12*A^3*a*b^8*d
^7*f + 3*B^3*a*b^8*c^7*f - B^2*C*a^9*d^7*f - 4*A^2*B*b^9*d^7*f + 3*A^2*C*b^
9*c^7*f - 3*A*C^2*b^9*c^7*f - A*C^2*a^9*d^7*f - A*B^2*b^9*c^7*f - C^3*a^9*d
^7*f - A^3*b^9*c^7*f + B^2*C*b^9*c^7*f + A^2*C*a^9*d^7*f + A*B^2*a^9*d^7*f
+ C^3*b^9*c^7*f + A^3*a^9*d^7*f - 6*A*B^2*C*a^5*b*c*d^5 - 21*A^2*B*C*a^3*b^
3*c^2*d^4 + 21*A*B*C^2*a^3*b^3*c^2*d^4 + 12*A*B^2*C*a^4*b^2*c^2*d^4 - 12*A*
B^2*C*a^2*b^4*c^2*d^4 - 10*A*B^2*C*a^3*b^3*c^3*d^3 - 6*A*B*C^2*a^4*b^2*c^3*
d^3 + 3*A^2*B*C*a^4*b^2*c^3*d^3 + 3*A^2*B*C*a^2*b^4*c^3*d^3 + 3*A*B^2*C*a^2
*b^4*c^4*d^2 + 3*A*B*C^2*a^2*b^4*c^3*d^3 + 2*A*B*C^2*a^3*b^3*c^4*d^2 - A^2*
B*C*a^3*b^3*c^4*d^2 + 18*A^2*B*C*a^2*b^4*c*d^5 + 10*A*B^2*C*a^3*b^3*c*d^5 +
9*A^2*B*C*a^4*b^2*c*d^5 - 9*A*B*C^2*a^4*b^2*c*d^5 - 9*A*B*C^2*a^2*b^4*c*d^
5 - 6*A^2*B*C*a*b^5*c^2*d^4 + 6*A*B^2*C*a*b^5*c^3*d^3 + 6*A*B*C^2*a^5*b*c^2
*d^4 - 6*A*B*C^2*a*b^5*c^4*d^2 - 3*A^2*B*C*a^5*b*c^2*d^4 + 3*A^2*B*C*a*b^5*
c^4*d^2 + 3*A*B*C^2*a*b^5*c^2*d^4 - 3*B^3*C*a^5*b*c^2*d^4 + 3*B^3*C*a^4*b^2
*c*d^5 + 3*B^3*C*a*b^5*c^4*d^2 + 3*B^2*C^2*a^5*b*c*d^5 - 3*B*C^3*a^5*b*c^2*
d^4 + 3*B*C^3*a^4*b^2*c*d^5 + 3*B*C^3*a*b^5*c^4*d^2 + 24*A^3*C*a^3*b^3*c*d^
5 + 8*A*C^3*a^3*b^3*c*d^5 - 9*A^3*B*a^2*b^4*c*d^5 - 9*A*B^3*a^2*b^4*c*d^5 -
3*A^3*B*a^4*b^2*c*d^5 + 3*A^3*B*a*b^5*c^2*d^4 + 3*A^2*B^2*a^5*b*c*d^5 - 3*
A*B^3*a^4*b^2*c*d^5 + 3*A*B^3*a*b^5*c^2*d^4 + 5*A*B*C^2*b^6*c^3*d^3 - 4*A^2
*B*C*b^6*c^3*d^3 - A*B^2*C*b^6*c^4*d^2 - 3*A*B^2*C*a^4*b^2*d^6 - 2*A^2*B*C*
a^3*b^3*d^6 + 9*B^2*C^2*a^3*b^3*c^3*d^3 - 6*B^2*C^2*a^4*b^2*c^2*d^4 + 6*B^2
*C^2*a^2*b^4*c^2*d^4 - 3*B^2*C^2*a^2*b^4*c^4*d^2 + 24*A^2*C^2*a^3*b^3*c^3*d
^3 - 15*A^2*C^2*a^4*b^2*c^2*d^4 - 9*A^2*C^2*a^2*b^4*c^4*d^2 + 3*A^2*C^2*a^2
*b^4*c^2*d^4 + 9*A^2*B^2*a^2*b^4*c^2*d^4 - 3*A^2*B^2*a^4*b^2*c^2*d^4 + 4*A^
2*B*C*b^6*c*d^5 - 2*A*B*C^2*b^6*c*d^5 + 2*A*B*C^2*a^6*c*d^5 - A^2*B*C*a^6*c
*d^5 + 6*A^2*B*C*a^5*b*d^6 - 3*A*B*C^2*a^5*b*d^6 - 7*B^3*C*a^3*b^3*c^2*d^4
- 7*B*C^3*a^3*b^3*c^2*d^4 + 3*B^3*C*a^4*b^2*c^3*d^3 - 3*B^3*C*a^2*b^4*c^3*d
^3 - 3*B^2*C^2*a*b^5*c^3*d^3 + 3*B*C^3*a^4*b^2*c^3*d^3 - 3*B*C^3*a^2*b^4*c^
3*d^3 - B^3*C*a^3*b^3*c^4*d^2 - B^2*C^2*a^3*b^3*c*d^5 - B*C^3*a^3*b^3*c^4*d
^2 - 24*A^2*C^2*a^3*b^3*c*d^5 - 24*A*C^3*a^3*b^3*c^3*d^3 + 12*A*C^3*a^4*b^2

```
*c^2*d^4 + 9*A*C^3*a^2*b^4*c^4*d^2 - 8*A^3*C*a^3*b^3*c^3*d^3 + 6*A^3*C*a^4*
b^2*c^2*d^4 - 6*A^3*C*a^2*b^4*c^2*d^4 + 3*A^3*C*a^2*b^4*c^4*d^2 - 9*A^2*B^2
*a^3*b^3*c*d^5 + 7*A^3*B*a^3*b^3*c^2*d^4 + 7*A*B^3*a^3*b^3*c^2*d^4 - 3*A^3*
B*a^2*b^4*c^3*d^3 - 3*A^2*B^2*a*b^5*c^3*d^3 - 3*A*B^3*a^2*b^4*c^3*d^3 - 5*A
^2*C^2*b^6*c^2*d^4 + 3*A^2*C^2*b^6*c^4*d^2 + 12*A^2*C^2*a^4*b^2*d^6 + 3*A^2
*C^2*a^2*b^4*d^6 + 6*A^2*B^2*a^4*b^2*d^6 + 3*A^2*B^2*a^2*b^4*d^6 + A*B*C^2*
a^3*b^3*d^6 - 3*B^4*a*b^5*c^3*d^3 - B^4*a^3*b^3*c*d^5 + A^2*B^2*a^3*b^3*c^3
*d^3 - 8*A^4*a^3*b^3*c*d^5 - 2*B^3*C*b^6*c^3*d^3 - 2*B*C^3*b^6*c^3*d^3 + 4*
A^3*C*b^6*c^2*d^4 - 3*A*C^3*b^6*c^4*d^2 + 2*A*C^3*b^6*c^2*d^4 - A^3*C*b^6*c
^4*d^2 - 2*A*C^3*a^6*c^2*d^4 - 15*A^3*C*a^4*b^2*d^6 - 6*A^3*C*a^2*b^4*d^6 -
 3*A*C^3*a^4*b^2*d^6 + 3*B^4*a^5*b*c*d^5 - B^3*C*a^6*c*d^5 - B*C^3*a^6*c*d^
5 - 2*A^3*B*b^6*c*d^5 - 2*A*B^3*b^6*c*d^5 - 3*A^3*B*a^5*b*d^6 - 3*A*B^3*a^5
*b*d^6 + 8*C^4*a^3*b^3*c^3*d^3 - 3*C^4*a^4*b^2*c^2*d^4 - 3*C^4*a^2*b^4*c^4*
d^2 + 6*B^4*a^2*b^4*c^2*d^4 - 3*B^4*a^4*b^2*c^2*d^4 + 3*A^4*a^2*b^4*c^2*d^4
 + B^2*C^2*b^6*c^4*d^2 + B^2*C^2*b^6*c^2*d^4 + B^2*C^2*a^6*c^2*d^4 + A^2*C^
2*a^6*c^2*d^4 - 2*A^3*C*b^6*d^6 + A^3*B*b^6*c^3*d^3 + A*B^3*b^6*c^3*d^3 + A
^3*B*a^3*b^3*d^6 + A*B^3*a^3*b^3*d^6 - A^4*b^6*c^2*d^4 + 6*A^4*a^4*b^2*d^6
+ 3*A^4*a^2*b^4*d^6 - 2*A^2*C^2*a^6*d^6 + A*B^2*C*a^6*d^6 + B^4*a^3*b^3*c^3
*d^3 + A^3*C*a^6*d^6 + A*C^3*a^6*d^6 + C^4*b^6*c^4*d^2 + C^4*a^6*c^2*d^4 +
B^4*b^6*c^2*d^4 + A^2*C^2*b^6*d^6 + A^2*B^2*b^6*d^6 + A^4*b^6*d^6, f, k), k
, 1, 4))/f
```

3.77 $\int \frac{(a+b\tan(e+fx))^3(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^2}\,dx$

Optimal result 771
Rubi [A] (verified) 772
Mathematica [C] (verified) 776
Maple [A] (verified) 777
Fricas [B] (verification not implemented) 777
Sympy [C] (verification not implemented) 778
Maxima [A] (verification not implemented) 791
Giac [B] (verification not implemented) 792
Mupad [B] (verification not implemented) 793

Optimal result

Integrand size = 45, antiderivative size = 579

$$\int \frac{(a+b\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^2}\,dx =$$
$$-\frac{(a^3(c^2C-2Bcd-Cd^2-A(c^2-d^2))-3ab^2(c^2C-2Bcd-Cd^2-A(c^2-d^2))-3a^2b(2c(A-C)d-}{(c^2+d^2)^2}$$
$$+\frac{(3a^2b(c^2C-2Bcd-Cd^2-A(c^2-d^2))-b^3(c^2C-2Bcd-Cd^2-A(c^2-d^2))+a^3(2c(A-C)d-B(}{(c^2+d^2)^2\,f}$$
$$+\frac{(bc-ad)^2\,(b(3c^4C-2Bc^3d+c^2(A+5C)d^2-4Bcd^3+3Ad^4)+ad^2(2c(A-C)d-B(c^2-d^2)))\log(c}{d^4\,(c^2+d^2)^2\,f}$$
$$+\frac{b^2(ad(3c^2C-Bcd+(A+2C)d^2)-b(3c^3C-2Bc^2d+c(A+2C)d^2-Bd^3))\tan(e+fx)}{d^3\,(c^2+d^2)\,f}$$
$$+\frac{b(3c^2C-2Bcd+(2A+C)d^2)\,(a+b\tan(e+fx))^2}{2d^2\,(c^2+d^2)\,f}$$
$$-\frac{(c^2C-Bcd+Ad^2)\,(a+b\tan(e+fx))^3}{d\,(c^2+d^2)\,f(c+d\tan(e+fx))}$$

```
[Out] -(a^3*(c^2*C-2*B*c*d-C*d^2-A*(c^2-d^2))-3*a*b^2*(c^2*C-2*B*c*d-C*d^2-A*(c^2
-d^2))-3*a^2*b*(2*c*(A-C)*d-B*(c^2-d^2))+b^3*(2*c*(A-C)*d-B*(c^2-d^2)))*x/(
c^2+d^2)^2+(3*a^2*b*(c^2*C-2*B*c*d-C*d^2-A*(c^2-d^2))-b^3*(c^2*C-2*B*c*d-C*
d^2-A*(c^2-d^2))+a^3*(2*c*(A-C)*d-B*(c^2-d^2))-3*a*b^2*(2*c*(A-C)*d-B*(c^2-
d^2)))*ln(cos(f*x+e))/(c^2+d^2)^2/f+(-a*d+b*c)^2*(b*(3*c^4*C-2*B*c^3*d+c^2*
(A+5*C)*d^2-4*B*c*d^3+3*A*d^4)+a*d^2*(2*c*(A-C)*d-B*(c^2-d^2)))*ln(c+d*tan(
f*x+e))/d^4/(c^2+d^2)^2/f+b^2*(a*d*(3*c^2*C-B*c*d+(A+2*C)*d^2)-b*(3*c^3*C-2
*B*c^2*d+c*(A+2*C)*d^2-B*d^3))*tan(f*x+e)/d^3/(c^2+d^2)/f+1/2*b*(3*c^2*C-2*
B*c*d+(2*A+C)*d^2)*(a+b*tan(f*x+e))^2/d^2/(c^2+d^2)/f-(A*d^2-B*c*d+C*c^2)*(
a+b*tan(f*x+e))^3/d/(c^2+d^2)/f/(c+d*tan(f*x+e))
```

Rubi [A] (verified)

Time = 2.52 (sec) , antiderivative size = 579, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 7, $\frac{\text{number of rules}}{\text{integrand size}} = 0.156$, Rules used = {3726, 3728, 3718, 3707, 3698, 31, 3556}

$$\int \frac{(a+b\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^2}\,dx$$
$$= \frac{\log(\cos(e+fx))\left(a^3(2cd(A-C)-B(c^2-d^2))+3a^2b(-A(c^2-d^2)-2Bcd+c^2C-Cd^2)-3ab^2(2cd(A\right.}{f\left(c^2+d^2\right)^2}$$
$$-\frac{x(a^3(-A(c^2-d^2)-2Bcd+c^2C-Cd^2)-3a^2b(2cd(A-C)-B(c^2-d^2))-3ab^2(-A(c^2-d^2)-2Bc}{\left(c^2+d^2\right)^2}$$
$$+\frac{b^2\tan(e+fx)\left(ad(d^2(A+2C)-Bcd+3c^2C)-b(cd^2(A+2C)-2Bc^2d-Bd^3+3c^3C)\right)}{d^3f\left(c^2+d^2\right)}$$
$$-\frac{\left(Ad^2-Bcd+c^2C\right)(a+b\tan(e+fx))^3}{df\left(c^2+d^2\right)(c+d\tan(e+fx))}$$
$$+\frac{b(d^2(2A+C)-2Bcd+3c^2C)\,(a+b\tan(e+fx))^2}{2d^2f\left(c^2+d^2\right)}$$
$$+\frac{(bc-ad)^2\left(ad^2(2cd(A-C)-B(c^2-d^2))+b(c^2d^2(A+5C)+3Ad^4-2Bc^3d-4Bcd^3+3c^4C)\right)\log(c+}{d^4f\left(c^2+d^2\right)^2}$$

```
[In] Int[((a + b*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c + d
*Tan[e + f*x])^2,x]
```

```
[Out] -(((a^3*(c^2*C - 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - 3*a*b^2*(c^2*C - 2*B*c*
d - C*d^2 - A*(c^2 - d^2)) - 3*a^2*b*(2*c*(A - C)*d - B*(c^2 - d^2)) + b^3*
(2*c*(A - C)*d - B*(c^2 - d^2)))*x)/(c^2 + d^2)^2) + ((3*a^2*b*(c^2*C - 2*B
*c*d - C*d^2 - A*(c^2 - d^2)) - b^3*(c^2*C - 2*B*c*d - C*d^2 - A*(c^2 - d^2
)) + a^3*(2*c*(A - C)*d - B*(c^2 - d^2)) - 3*a*b^2*(2*c*(A - C)*d - B*(c^2
- d^2)))*Log[Cos[e + f*x]])/((c^2 + d^2)^2*f) + ((b*c - a*d)^2*(b*(3*c^4*C
- 2*B*c^3*d + c^2*(A + 5*C)*d^2 - 4*B*c*d^3 + 3*A*d^4) + a*d^2*(2*c*(A - C)
*d - B*(c^2 - d^2)))*Log[c + d*Tan[e + f*x]])/(d^4*(c^2 + d^2)^2*f) + (b^2*
(a*d*(3*c^2*C - B*c*d + (A + 2*C)*d^2) - b*(3*c^3*C - 2*B*c^2*d + c*(A + 2*
C)*d^2 - B*d^3))*Tan[e + f*x])/(d^3*(c^2 + d^2)*f) + (b*(3*c^2*C - 2*B*c*d
+ (2*A + C)*d^2)*(a + b*Tan[e + f*x])^2)/(2*d^2*(c^2 + d^2)*f) - ((c^2*C -
B*c*d + A*d^2)*(a + b*Tan[e + f*x])^3)/(d*(c^2 + d^2)*f*(c + d*Tan[e + f*x]
))
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3698

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3707

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[(a*A + b*B - a
*C)*(x/(a^2 + b^2)), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3726

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*d^2 + c*(c*C - B*d))*(a + b*Tan[e +
 f*x])^m*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
```

```
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\text{integral} = -\frac{(c^2C - Bcd + Ad^2)(a + b\tan(e+fx))^3}{d(c^2+d^2)f(c+d\tan(e+fx))} + \frac{\int \frac{(a+b\tan(e+fx))^2(Ad(ac+3bd)+(3bc-ad)(cC-Bd)+d((A-C)(bc-ad)+B(ac+bd))\tan(e+fx)+b(3c^2C-2Bcd+(2A+C)d^2)\tan^2(e+fx))}{c+d\tan(e+fx)}}{d(c^2+d^2)}$$

$$= \frac{b(3c^2C - 2Bcd + (2A+C)d^2)(a+b\tan(e+fx))^2}{2d^2(c^2+d^2)f} - \frac{(c^2C - Bcd + Ad^2)(a+b\tan(e+fx))^3}{d(c^2+d^2)f(c+d\tan(e+fx))} + \frac{\int \frac{(a+b\tan(e+fx))(-2(b^2c(3c^2C-2Bcd+(2A+C)d^2)-ad(Ad(ac+3bd)+(3bc-ad)(cC-Bd)))+2d^2(2ab(Ac-cC+Bd)+a^2(Bc-(A}{c+d\tan(e}}{2d^2(c^2}$$

$$= \frac{b^2(ad(3c^2C - Bcd + (A+2C)d^2) - b(3c^3C - 2Bc^2d + c(A+2C)d^2 - Bd^3))\tan(e+fx)}{d^3(c^2+d^2)f} + \frac{b(3c^2C - 2Bcd + (2A+C)d^2)(a+b\tan(e+fx))^2}{2d^2(c^2+d^2)f} - \frac{(c^2C - Bcd + Ad^2)(a+b\tan(e+fx))^3}{d(c^2+d^2)f(c+d\tan(e+fx))} - \int \frac{-2(a^3d^3(Ac-cC+Bd)+3a^2bd^2(c^2C-Bcd+Ad^2)-3ab^2cd(2c^2C-Bcd+(A+C)d^2)+b^3c(3c^3C-2Bc^2d+c(A+2C)d^2-Bd^3))-2d^3}{}$$

$$= -\frac{(a^3(c^2C-2Bcd-Cd^2-A(c^2-d^2))-3ab^2(c^2C-2Bcd-Cd^2-A(c^2-d^2))-3a^2b(2c(A-}{(c^2+d^2)^2}$$
$$+\frac{b^2(ad(3c^2C-Bcd+(A+2C)d^2)-b(3c^3C-2Bc^2d+c(A+2C)d^2-Bd^3))\tan(e+fx)}{d^3(c^2+d^2)f}$$
$$+\frac{b(3c^2C-2Bcd+(2A+C)d^2)(a+b\tan(e+fx))^2}{2d^2(c^2+d^2)f}$$
$$-\frac{(c^2C-Bcd+Ad^2)(a+b\tan(e+fx))^3}{d(c^2+d^2)f(c+d\tan(e+fx))}$$
$$-\frac{(3a^2b(c^2C-2Bcd-Cd^2-A(c^2-d^2))-b^3(c^2C-2Bcd-Cd^2-A(c^2-d^2))+a^3(2c(A-C}{(c^2+d^2)^2}$$
$$+\frac{((bc-ad)^2(b(3c^4C-2Bc^3d+c^2(A+5C)d^2-4Bcd^3+3Ad^4)+ad^2(2c(A-C)d-B(c^2-d^2}{d^3(c^2+d^2)^2}$$

$$= -\frac{(a^3(c^2C-2Bcd-Cd^2-A(c^2-d^2))-3ab^2(c^2C-2Bcd-Cd^2-A(c^2-d^2))-3a^2b(2c(A-}{(c^2+d^2)^2}$$
$$+\frac{(3a^2b(c^2C-2Bcd-Cd^2-A(c^2-d^2))-b^3(c^2C-2Bcd-Cd^2-A(c^2-d^2))+a^3(2c(A-C}{(c^2+d^2)^2f}$$
$$+\frac{b^2(ad(3c^2C-Bcd+(A+2C)d^2)-b(3c^3C-2Bc^2d+c(A+2C)d^2-Bd^3))\tan(e+fx)}{d^3(c^2+d^2)f}$$
$$+\frac{b(3c^2C-2Bcd+(2A+C)d^2)(a+b\tan(e+fx))^2}{2d^2(c^2+d^2)f}$$
$$-\frac{(c^2C-Bcd+Ad^2)(a+b\tan(e+fx))^3}{d(c^2+d^2)f(c+d\tan(e+fx))}$$
$$+\frac{((bc-ad)^2(b(3c^4C-2Bc^3d+c^2(A+5C)d^2-4Bcd^3+3Ad^4)+ad^2(2c(A-C)d-B(c^2-d^2}{d^4(c^2+d^2)^2f}$$

$$= -\frac{(a^3(c^2C-2Bcd-Cd^2-A(c^2-d^2))-3ab^2(c^2C-2Bcd-Cd^2-A(c^2-d^2))-3a^2b(2c(A-}{(c^2+d^2)^2}$$
$$+\frac{(3a^2b(c^2C-2Bcd-Cd^2-A(c^2-d^2))-b^3(c^2C-2Bcd-Cd^2-A(c^2-d^2))+a^3(2c(A-C}{(c^2+d^2)^2f}$$
$$+\frac{(bc-ad)^2(b(3c^4C-2Bc^3d+c^2(A+5C)d^2-4Bcd^3+3Ad^4)+ad^2(2c(A-C)d-B(c^2-d^2)}{d^4(c^2+d^2)^2f}$$
$$+\frac{b^2(ad(3c^2C-Bcd+(A+2C)d^2)-b(3c^3C-2Bc^2d+c(A+2C)d^2-Bd^3))\tan(e+fx)}{d^3(c^2+d^2)f}$$
$$+\frac{b(3c^2C-2Bcd+(2A+C)d^2)(a+b\tan(e+fx))^2}{2d^2(c^2+d^2)f}$$
$$-\frac{(c^2C-Bcd+Ad^2)(a+b\tan(e+fx))^3}{d(c^2+d^2)f(c+d\tan(e+fx))}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 7.60 (sec) , antiderivative size = 1022, normalized size of antiderivative = 1.77

$$\int \frac{(a+b\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^2}\,dx=\frac{C(a+b\tan(e+fx))^3}{2df(c+d\tan(e+fx))} + \frac{\frac{(-3bcC+2bBd+3aCd)(a+b\tan(e+fx))^2}{df(c+d\tan(e+fx))}+\frac{2\left(-\frac{d^2\left(-3a^2Abc^2+Ab^3c^2-a^3Bc^2+3ab^2Bc^2+3a^2bc^2C-b^3c^2C+2a^3Acd-6aAb^2cd-6a^2bBcd+2b^3Bcd-2\right.}{}\right.}{}}{}$$

```
[In] Integrate[((a + b*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(c + d*Tan[e + f*x])^2,x]
```

```
[Out] (C*(a + b*Tan[e + f*x])^3)/(2*d*f*(c + d*Tan[e + f*x])) + (((-3*b*c*C + 2*b
*B*d + 3*a*C*d)*(a + b*Tan[e + f*x])^2)/(d*f*(c + d*Tan[e + f*x])) + (2*(-1
/2*(d^2*(-3*a^2*A*b*c^2 + A*b^3*c^2 - a^3*B*c^2 + 3*a*b^2*B*c^2 + 3*a^2*b*c
^2*C - b^3*c^2*C + 2*a^3*A*c*d - 6*a*A*b^2*c*d - 6*a^2*b*B*c*d + 2*b^3*B*c*
d - 2*a^3*c*C*d + 6*a*b^2*c*C*d + 3*a^2*A*b*d^2 - A*b^3*d^2 + a^3*B*d^2 - 3
*a*b^2*B*d^2 - 3*a^2*b*C*d^2 + b^3*C*d^2 + I*(a^3*A*c^2 - 3*a*A*b^2*c^2 - 3
*a^2*b*B*c^2 + b^3*B*c^2 - a^3*c^2*C + 3*a*b^2*c^2*C + 6*a^2*A*b*c*d - 2*A*
b^3*c*d + 2*a^3*B*c*d - 6*a*b^2*B*c*d - 6*a^2*b*c*C*d + 2*b^3*c*C*d - a^3*A
*d^2 + 3*a*A*b^2*d^2 + 3*a^2*b*B*d^2 - b^3*B*d^2 + a^3*C*d^2 - 3*a*b^2*C*d^
2))*Log[I - Tan[e + f*x]])/((c^2 + d^2)^2*f) + (d^2*(3*a^2*A*b*c^2 - A*b^3*
c^2 + a^3*B*c^2 - 3*a*b^2*B*c^2 - 3*a^2*b*c^2*C + b^3*c^2*C - 2*a^3*A*c*d +
 6*a*A*b^2*c*d + 6*a^2*b*B*c*d - 2*b^3*B*c*d + 2*a^3*c*C*d - 6*a*b^2*c*C*d
- 3*a^2*A*b*d^2 + A*b^3*d^2 - a^3*B*d^2 + 3*a*b^2*B*d^2 + 3*a^2*b*C*d^2 - b
^3*C*d^2 + I*(a^3*A*c^2 - 3*a*A*b^2*c^2 - 3*a^2*b*B*c^2 + b^3*B*c^2 - a^3*c
^2*C + 3*a*b^2*c^2*C + 6*a^2*A*b*c*d - 2*A*b^3*c*d + 2*a^3*B*c*d - 6*a*b^2*
B*c*d - 6*a^2*b*c*C*d + 2*b^3*c*C*d - a^3*A*d^2 + 3*a*A*b^2*d^2 + 3*a^2*b*B
*d^2 - b^3*B*d^2 + a^3*C*d^2 - 3*a*b^2*C*d^2))*Log[I + Tan[e + f*x]])/(2*(c
^2 + d^2)^2*f) + ((b*c - a*d)^2*(b*(3*c^4*C - 2*B*c^3*d + c^2*(A + 5*C)*d^2
 - 4*B*c*d^3 + 3*A*d^4) + a*d^2*(2*c*(A - C)*d - B*(c^2 - d^2)))*Log[c + d*
Tan[e + f*x]])/(d^2*(c^2 + d^2)^2*f) - ((b*c - a*d)^2*(a*d*(3*c^2*C - B*c*d
 + (A + 2*C)*d^2) - b*(3*c^3*C - 2*B*c^2*d + c*(A + 2*C)*d^2 - B*d^3)))/(d^
2*(c^2 + d^2)*f*(c + d*Tan[e + f*x]))))/d)/(2*d)
```

Maple [A] (verified)

Time = 0.34 (sec) , antiderivative size = 829, normalized size of antiderivative = 1.43

method	result
derivativedivides	$\frac{\frac{b^2\left(\frac{\tan(fx+e)^2 Cbd}{2}+\tan(fx+e)bdB+3\tan(fx+e)Cad-2\tan(fx+e)Cbc\right)}{d^3}+\frac{\left(-2Aa^3cd+3Aa^2bc^2-3Aa^2bd^2+6Aab^2cd-Ab^3c^2+\right.}{}}{}$
default	$\frac{\frac{b^2\left(\frac{\tan(fx+e)^2 Cbd}{2}+\tan(fx+e)bdB+3\tan(fx+e)Cad-2\tan(fx+e)Cbc\right)}{d^3}+\frac{\left(-2Aa^3cd+3Aa^2bc^2-3Aa^2bd^2+6Aab^2cd-Ab^3c^2+\right.}{}}{}$
norman	Expression too large to display
parallelrisch	Expression too large to display
risch	Expression too large to display

```
[In] int((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^2,x
,method=_RETURNVERBOSE)
```

```
[Out] 1/f*(b^2/d^3*(1/2*tan(f*x+e)^2*C*b*d+tan(f*x+e)*b*d*B+3*tan(f*x+e)*C*a*d-2*
tan(f*x+e)*C*b*c)+1/(c^2+d^2)^2*(1/2*(-2*A*a^3*c*d+3*A*a^2*b*c^2-3*A*a^2*b*
d^2+6*A*a*b^2*c*d-A*b^3*c^2+A*b^3*d^2+B*a^3*c^2-B*a^3*d^2+6*B*a^2*b*c*d-3*B
*a*b^2*c^2+3*B*a*b^2*d^2-2*B*b^3*c*d+2*C*a^3*c*d-3*C*a^2*b*c^2+3*C*a^2*b*d^
2-6*C*a*b^2*c*d+C*b^3*c^2-C*b^3*d^2)*ln(1+tan(f*x+e)^2)+(A*a^3*c^2-A*a^3*d^
2+6*A*a^2*b*c*d-3*A*a*b^2*c^2+3*A*a*b^2*d^2-2*A*b^3*c*d+2*B*a^3*c*d-3*B*a^2
*b*c^2+3*B*a^2*b*d^2-6*B*a*b^2*c*d+B*b^3*c^2-B*b^3*d^2-C*a^3*c^2+C*a^3*d^2-
6*C*a^2*b*c*d+3*C*a*b^2*c^2-3*C*a*b^2*d^2+2*C*b^3*c*d)*arctan(tan(f*x+e)))-
1/d^4*(A*a^3*d^5-3*A*a^2*b*c*d^4+3*A*a*b^2*c^2*d^3-A*b^3*c^3*d^2-B*a^3*c*d^
4+3*B*a^2*b*c^2*d^3-3*B*a*b^2*c^3*d^2+B*b^3*c^4*d+C*a^3*c^2*d^3-3*C*a^2*b*c
^3*d^2+3*C*a*b^2*c^4*d-C*b^3*c^5)/(c^2+d^2)/(c+d*tan(f*x+e))+1/d^4*(2*A*a^3
*c*d^5-3*A*a^2*b*c^2*d^4+3*A*a^2*b*d^6-6*A*a*b^2*c*d^5+A*b^3*c^4*d^2+3*A*b^
3*c^2*d^4-B*a^3*c^2*d^4+B*a^3*d^6-6*B*a^2*b*c*d^5+3*B*a*b^2*c^4*d^2+9*B*a*b
^2*c^2*d^4-2*B*b^3*c^5*d-4*B*b^3*c^3*d^3-2*C*a^3*c*d^5+3*C*a^2*b*c^4*d^2+9*
C*a^2*b*c^2*d^4-6*C*a*b^2*c^5*d-12*C*a*b^2*c^3*d^3+3*C*b^3*c^6+5*C*b^3*c^4*
d^2)/(c^2+d^2)^2*ln(c+d*tan(f*x+e)))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1477 vs. $2(577) = 1154$.

Time = 1.10 (sec) , antiderivative size = 1477, normalized size of antiderivative = 2.55

$$\int \frac{(a+b\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^2}\,dx = \text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^2,x, algorithm="fricas")
```

```
[Out] 1/2*(3*C*b^3*c^5*d^2 - 2*A*a^3*d^7 - 2*(3*C*a*b^2 + B*b^3)*c^4*d^3 + 2*(3*C
*a^2*b + 3*B*a*b^2 + (A + C)*b^3)*c^3*d^4 - 2*(C*a^3 + 3*B*a^2*b + 3*A*a*b^
2)*c^2*d^5 + (2*B*a^3 + 6*A*a^2*b + C*b^3)*c*d^6 + (C*b^3*c^4*d^3 + 2*C*b^3
*c^2*d^5 + C*b^3*d^7)*tan(f*x + e)^3 + 2*(((A - C)*a^3 - 3*B*a^2*b - 3*(A -
 C)*a*b^2 + B*b^3)*c^3*d^4 + 2*(B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A -
C)*b^3)*c^2*d^5 - ((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b^3)*c*d^6
)*f*x - (3*C*b^3*c^5*d^2 + 6*C*b^3*c^3*d^4 + 3*C*b^3*c*d^6 - 2*(3*C*a*b^2 +
 B*b^3)*c^4*d^3 - 4*(3*C*a*b^2 + B*b^3)*c^2*d^5 - 2*(3*C*a*b^2 + B*b^3)*d^7
)*tan(f*x + e)^2 + (3*C*b^3*c^7 - 2*(3*C*a*b^2 + B*b^3)*c^6*d + (3*C*a^2*b
+ 3*B*a*b^2 + (A + 5*C)*b^3)*c^5*d^2 - 4*(3*C*a*b^2 + B*b^3)*c^4*d^3 - (B*a
^3 + 3*(A - 3*C)*a^2*b - 9*B*a*b^2 - 3*A*b^3)*c^3*d^4 + 2*((A - C)*a^3 - 3*
B*a^2*b - 3*A*a*b^2)*c^2*d^5 + (B*a^3 + 3*A*a^2*b)*c*d^6 + (3*C*b^3*c^6*d -
 2*(3*C*a*b^2 + B*b^3)*c^5*d^2 + (3*C*a^2*b + 3*B*a*b^2 + (A + 5*C)*b^3)*c^
4*d^3 - 4*(3*C*a*b^2 + B*b^3)*c^3*d^4 - (B*a^3 + 3*(A - 3*C)*a^2*b - 9*B*a*
b^2 - 3*A*b^3)*c^2*d^5 + 2*((A - C)*a^3 - 3*B*a^2*b - 3*A*a*b^2)*c*d^6 + (B
*a^3 + 3*A*a^2*b)*d^7)*tan(f*x + e))*log((d^2*tan(f*x + e)^2 + 2*c*d*tan(f*
x + e) + c^2)/(tan(f*x + e)^2 + 1)) - (3*C*b^3*c^7 - 2*(3*C*a*b^2 + B*b^3)*
c^6*d + (3*C*a^2*b + 3*B*a*b^2 + (A + 5*C)*b^3)*c^5*d^2 - 4*(3*C*a*b^2 + B*
b^3)*c^4*d^3 + (6*C*a^2*b + 6*B*a*b^2 + (2*A + C)*b^3)*c^3*d^4 - 2*(3*C*a*b
^2 + B*b^3)*c^2*d^5 + (3*C*a^2*b + 3*B*a*b^2 + (A - C)*b^3)*c*d^6 + (3*C*b^
3*c^6*d - 2*(3*C*a*b^2 + B*b^3)*c^5*d^2 + (3*C*a^2*b + 3*B*a*b^2 + (A + 5*C
)*b^3)*c^4*d^3 - 4*(3*C*a*b^2 + B*b^3)*c^3*d^4 + (6*C*a^2*b + 6*B*a*b^2 + (
2*A + C)*b^3)*c^2*d^5 - 2*(3*C*a*b^2 + B*b^3)*c*d^6 + (3*C*a^2*b + 3*B*a*b^
2 + (A - C)*b^3)*d^7)*tan(f*x + e))*log(1/(tan(f*x + e)^2 + 1)) - (6*C*b^3*
c^6*d - C*b^3*d^7 - 4*(3*C*a*b^2 + B*b^3)*c^5*d^2 + (6*C*a^2*b + 6*B*a*b^2
+ (2*A + 7*C)*b^3)*c^4*d^3 - 2*(C*a^3 + 3*B*a^2*b + 3*(A + 2*C)*a*b^2 + 2*B
*b^3)*c^3*d^4 + 2*(B*a^3 + 3*A*a^2*b + C*b^3)*c^2*d^5 - 2*(A*a^3 + 3*C*a*b^
2 + B*b^3)*c*d^6 - 2*(((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b^3)*c
^2*d^5 + 2*(B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*c*d^6 - ((A
- C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b^3)*d^7)*f*x)*tan(f*x + e))/((c
^4*d^5 + 2*c^2*d^7 + d^9)*f*tan(f*x + e) + (c^5*d^4 + 2*c^3*d^6 + c*d^8)*f)
```

Sympy [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 27.02 (sec) , antiderivative size = 24300, normalized size of antiderivative = 41.97

$$\int \frac{(a+b\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^2}\,dx = \text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))**3*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x
+e))**2,x)
```

```
[Out] Piecewise((zoo*x*(a + b*tan(e))**3*(A + B*tan(e) + C*tan(e)**2)/tan(e)**2,
Eq(c, 0) & Eq(d, 0) & Eq(f, 0)), ((A*a**3*x + 3*A*a**2*b*log(tan(e + f*x)**
```

2 + 1)/(2*f) - 3*A*a*b**2*x + 3*A*a*b**2*tan(e + f*x)/f - A*b**3*log(tan(e + f*x)**2 + 1)/(2*f) + A*b**3*tan(e + f*x)**2/(2*f) + B*a**3*log(tan(e + f*x)**2 + 1)/(2*f) - 3*B*a**2*b*x + 3*B*a**2*b*tan(e + f*x)/f - 3*B*a*b**2*log(tan(e + f*x)**2 + 1)/(2*f) + 3*B*a*b**2*tan(e + f*x)**2/(2*f) + B*b**3*x + B*b**3*tan(e + f*x)**3/(3*f) - B*b**3*tan(e + f*x)/f - C*a**3*x + C*a**3*tan(e + f*x)/f - 3*C*a**2*b*log(tan(e + f*x)**2 + 1)/(2*f) + 3*C*a**2*b*tan(e + f*x)**2/(2*f) + 3*C*a*b**2*x + C*a*b**2*tan(e + f*x)**3/f - 3*C*a*b**2*tan(e + f*x)/f + C*b**3*log(tan(e + f*x)**2 + 1)/(2*f) + C*b**3*tan(e + f*x)**4/(4*f) - C*b**3*tan(e + f*x)**2/(2*f))/c**2, Eq(d, 0)), (-A*a**3*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 2*I*A*a**3*f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + A*a**3*f*x/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - A*a**3*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 2*I*A*a**3/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 3*I*A*a**2*b*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 6*A*a**2*b*f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 3*I*A*a**2*b*f*x/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 3*I*A*a**2*b*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 3*A*a*b**2*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 6*I*A*a*b**2*f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 3*A*a*b**2*f*x/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 9*A*a*b**2*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 6*I*A*a*b**2/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 3*I*A*b**3*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 6*A*b**3*f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 3*I*A*b**3*f*x/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 2*A*b**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 4*I*A*b**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 2*A*b**3*log(tan(e + f*x)**2 + 1)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 5*I*A*b**3*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 4*A*b**3/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + I*B*a**3*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 2*B*a**3*f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - I*B*a**3*f*x/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + I*B*a**3*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 3*B*a**2*b*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 6*I*B*a**2*b*f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 3*B*a**2*b*f*x/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 9*B*a**2*b*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*

d**2*f) + 6*I*B*a**2*b/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x)
- 4*d**2*f) + 9*I*B*a*b**2*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 -
8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 18*B*a*b**2*f*x*tan(e + f*x)/(4*d**2*
f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 9*I*B*a*b**2*f*x/
(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 6*B*a*b**
2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 - 8*I*
d**2*f*tan(e + f*x) - 4*d**2*f) - 12*I*B*a*b**2*log(tan(e + f*x)**2 + 1)*ta
n(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f)
- 6*B*a*b**2*log(tan(e + f*x)**2 + 1)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*
f*tan(e + f*x) - 4*d**2*f) - 15*I*B*a*b**2*tan(e + f*x)/(4*d**2*f*tan(e + f
*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 12*B*a*b**2/(4*d**2*f*tan(e
+ f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 9*B*b**3*f*x*tan(e + f*x)
2/(4*d2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 18*I*
B*b**3*f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x)
 - 4*d**2*f) + 9*B*b**3*f*x/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e +
f*x) - 4*d**2*f) + 4*I*B*b**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)**2/(4*d
2*f*tan(e + f*x)2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 8*B*b**3*log(
tan(e + f*x)**2 + 1)*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*ta
n(e + f*x) - 4*d**2*f) - 4*I*B*b**3*log(tan(e + f*x)**2 + 1)/(4*d**2*f*tan(
e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 4*B*b**3*tan(e + f*x)**
3/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 19*B*b*
*3*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**
2*f) - 14*I*B*b**3/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*
d**2*f) + C*a**3*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f
*tan(e + f*x) - 4*d**2*f) - 2*I*C*a**3*f*x*tan(e + f*x)/(4*d**2*f*tan(e + f
*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - C*a**3*f*x/(4*d**2*f*tan(e +
 f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 3*C*a**3*tan(e + f*x)/(4*d
2*f*tan(e + f*x)2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 2*I*C*a**3/(4
*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 9*I*C*a**2*
b*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) -
 4*d**2*f) + 18*C*a**2*b*f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d
2*f*tan(e + f*x) - 4*d2*f) - 9*I*C*a**2*b*f*x/(4*d**2*f*tan(e + f*x)**2
 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 6*C*a**2*b*log(tan(e + f*x)**2 + 1
)*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d
2*f) - 12*I*C*a2*b*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(4*d**2*f*tan(
e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 6*C*a**2*b*log(tan(e +
f*x)**2 + 1)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f
) - 15*I*C*a**2*b*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e
 + f*x) - 4*d**2*f) - 12*C*a**2*b/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*ta
n(e + f*x) - 4*d**2*f) - 27*C*a*b**2*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e +
f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 54*I*C*a*b**2*f*x*tan(e + f
*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 27*C*
a*b**2*f*x/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f)
+ 12*I*C*a*b**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)**2/(4*d**2*f*tan(e +
f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 24*C*a*b**2*log(tan(e + f*x

)**2 + 1)*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x)
- 4*d**2*f) - 12*I*C*a*b**2*log(tan(e + f*x)**2 + 1)/(4*d**2*f*tan(e + f*x)
2 - 8*I*d2*f*tan(e + f*x) - 4*d**2*f) + 12*C*a*b**2*tan(e + f*x)**3/(4*
d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 57*C*a*b**2*
tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f
) - 42*I*C*a*b**2/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d
2*f) - 15*I*C*b3*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 - 8*I*d*
*2*f*tan(e + f*x) - 4*d**2*f) - 30*C*b**3*f*x*tan(e + f*x)/(4*d**2*f*tan(e
+ f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 15*I*C*b**3*f*x/(4*d**2*f
*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 8*C*b**3*log(tan(e
+ f*x)**2 + 1)*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(
e + f*x) - 4*d**2*f) + 16*I*C*b**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(4
*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 8*C*b**3*lo
g(tan(e + f*x)**2 + 1)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x)
- 4*d**2*f) + 2*C*b**3*tan(e + f*x)**4/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2
*f*tan(e + f*x) - 4*d**2*f) + 4*I*C*b**3*tan(e + f*x)**3/(4*d**2*f*tan(e +
f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 29*I*C*b**3*tan(e + f*x)/(4
*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 22*C*b**3/(
4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f), Eq(c, -I*d)
), (-A*a**3*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(
e + f*x) - 4*d**2*f) - 2*I*A*a**3*f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**
2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + A*a**3*f*x/(4*d**2*f*tan(e + f*x)
2 + 8*I*d2*f*tan(e + f*x) - 4*d**2*f) - A*a**3*tan(e + f*x)/(4*d**2*f*t
an(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 2*I*A*a**3/(4*d**2*f
*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 3*I*A*a**2*b*f*x*t
an(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2
*f) + 6*A*a**2*b*f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*ta
n(e + f*x) - 4*d**2*f) + 3*I*A*a**2*b*f*x/(4*d**2*f*tan(e + f*x)**2 + 8*I*d
2*f*tan(e + f*x) - 4*d2*f) - 3*I*A*a**2*b*tan(e + f*x)/(4*d**2*f*tan(e
+ f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 3*A*a*b**2*f*x*tan(e + f*
x)**2/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 6*I
*A*a*b**2*f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f
*x) - 4*d**2*f) - 3*A*a*b**2*f*x/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan
(e + f*x) - 4*d**2*f) - 9*A*a*b**2*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 +
8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 6*I*A*a*b**2/(4*d**2*f*tan(e + f*x)*
*2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 3*I*A*b**3*f*x*tan(e + f*x)**2/(
4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 6*A*b**3*f
*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**
2*f) + 3*I*A*b**3*f*x/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) -
4*d**2*f) + 2*A*b**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)**2/(4*d**2*f*ta
n(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 4*I*A*b**3*log(tan(e
+ f*x)**2 + 1)*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e +
f*x) - 4*d**2*f) - 2*A*b**3*log(tan(e + f*x)**2 + 1)/(4*d**2*f*tan(e + f*x)
2 + 8*I*d2*f*tan(e + f*x) - 4*d**2*f) + 5*I*A*b**3*tan(e + f*x)/(4*d**2
*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 4*A*b**3/(4*d**2

```
*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - I*B*a**3*f*x*tan
(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f
) + 2*B*a**3*f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e
+ f*x) - 4*d**2*f) + I*B*a**3*f*x/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*ta
n(e + f*x) - 4*d**2*f) - I*B*a**3*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 +
8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 3*B*a**2*b*f*x*tan(e + f*x)**2/(4*d**
2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 6*I*B*a**2*b*f*
x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2
*f) - 3*B*a**2*b*f*x/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) -
4*d**2*f) - 9*B*a**2*b*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*
tan(e + f*x) - 4*d**2*f) - 6*I*B*a**2*b/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**
2*f*tan(e + f*x) - 4*d**2*f) - 9*I*B*a*b**2*f*x*tan(e + f*x)**2/(4*d**2*f*t
an(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 18*B*a*b**2*f*x*tan(
e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) +
9*I*B*a*b**2*f*x/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d*
*2*f) + 6*B*a*b**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)**2/(4*d**2*f*tan(e
 + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 12*I*B*a*b**2*log(tan(e
+ f*x)**2 + 1)*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e +
f*x) - 4*d**2*f) - 6*B*a*b**2*log(tan(e + f*x)**2 + 1)/(4*d**2*f*tan(e + f*
x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 15*I*B*a*b**2*tan(e + f*x)/(4
*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 12*B*a*b**2
/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 9*B*b**3
*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) -
4*d**2*f) - 18*I*B*b**3*f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d*
*2*f*tan(e + f*x) - 4*d**2*f) + 9*B*b**3*f*x/(4*d**2*f*tan(e + f*x)**2 + 8*
I*d**2*f*tan(e + f*x) - 4*d**2*f) - 4*I*B*b**3*log(tan(e + f*x)**2 + 1)*tan
(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f
) + 8*B*b**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**
2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 4*I*B*b**3*log(tan(e + f*x)**2 +
1)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 4*B*b*
*3*tan(e + f*x)**3/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*
d**2*f) + 19*B*b**3*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan
(e + f*x) - 4*d**2*f) + 14*I*B*b**3/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*
tan(e + f*x) - 4*d**2*f) + C*a**3*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x
)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 2*I*C*a**3*f*x*tan(e + f*x)/(4
*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - C*a**3*f*x/
(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 3*C*a**3*
tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f
) - 2*I*C*a**3/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2
*f) - 9*I*C*a**2*b*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2
*f*tan(e + f*x) - 4*d**2*f) + 18*C*a**2*b*f*x*tan(e + f*x)/(4*d**2*f*tan(e
+ f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 9*I*C*a**2*b*f*x/(4*d**2*
f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 6*C*a**2*b*log(ta
n(e + f*x)**2 + 1)*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*t
an(e + f*x) - 4*d**2*f) + 12*I*C*a**2*b*log(tan(e + f*x)**2 + 1)*tan(e + f*
```

x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 6*C*a*
*2*b*log(tan(e + f*x)**2 + 1)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e
+ f*x) - 4*d**2*f) + 15*I*C*a**2*b*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 +
 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 12*C*a**2*b/(4*d**2*f*tan(e + f*x)**
2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 27*C*a*b**2*f*x*tan(e + f*x)**2/(
4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 54*I*C*a*b
2*f*x*tan(e + f*x)/(4*d2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) -
4*d**2*f) + 27*C*a*b**2*f*x/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e +
f*x) - 4*d**2*f) - 12*I*C*a*b**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)**2/(
4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 24*C*a*b**
2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**
2*f*tan(e + f*x) - 4*d**2*f) + 12*I*C*a*b**2*log(tan(e + f*x)**2 + 1)/(4*d*
*2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 12*C*a*b**2*ta
n(e + f*x)**3/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*
f) + 57*C*a*b**2*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e
+ f*x) - 4*d**2*f) + 42*I*C*a*b**2/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*t
an(e + f*x) - 4*d**2*f) + 15*I*C*b**3*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e +
 f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 30*C*b**3*f*x*tan(e + f*x)
/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 15*I*C*b
3*f*x/(4*d2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 8
*C*b**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2
+ 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 16*I*C*b**3*log(tan(e + f*x)**2 + 1
)*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2
*f) + 8*C*b**3*log(tan(e + f*x)**2 + 1)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**
2*f*tan(e + f*x) - 4*d**2*f) + 2*C*b**3*tan(e + f*x)**4/(4*d**2*f*tan(e + f
*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 4*I*C*b**3*tan(e + f*x)**3/(
4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 29*I*C*b**
3*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2
*f) + 22*C*b**3/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**
2*f), Eq(c, I*d)), (x*(a + b*tan(e))**3*(A + B*tan(e) + C*tan(e)**2)/(c + d
*tan(e))**2, Eq(f, 0)), (2*A*a**3*c**3*d**4*f*x/(2*c**5*d**4*f + 2*c**4*d**
5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f
+ 2*d**9*f*tan(e + f*x)) + 2*A*a**3*c**2*d**5*f*x*tan(e + f*x)/(2*c**5*d**4
*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x
) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 4*A*a**3*c**2*d**5*log(c/d + tan(
e + f*x))/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c
2*d7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) - 2*A*a**3*c*
*2*d**5*log(tan(e + f*x)**2 + 1)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x
) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(
e + f*x)) - 2*A*a**3*c**2*d**5/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x)
+ 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e
+ f*x)) - 2*A*a**3*c*d**6*f*x/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) +
 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e +
 f*x)) + 4*A*a**3*c*d**6*log(c/d + tan(e + f*x))*tan(e + f*x)/(2*c**5*d**4*
f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x)

```
 + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) - 2*A*a**3*c*d**6*log(tan(e + f*x)**
2 + 1)*tan(e + f*x)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d*
*6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) - 2
*A*a**3*d**7*f*x*tan(e + f*x)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) +
 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e +
 f*x)) - 2*A*a**3*d**7/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3
*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x))
- 6*A*a**2*b*c**3*d**4*log(c/d + tan(e + f*x))/(2*c**5*d**4*f + 2*c**4*d**5
*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f +
 2*d**9*f*tan(e + f*x)) + 3*A*a**2*b*c**3*d**4*log(tan(e + f*x)**2 + 1)/(2*
c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*ta
n(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 6*A*a**2*b*c**3*d**4/(2*
c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*ta
n(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 12*A*a**2*b*c**2*d**5*f*
x/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7
*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) - 6*A*a**2*b*c**2*d**
5*log(c/d + tan(e + f*x))*tan(e + f*x)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e
 + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*
f*tan(e + f*x)) + 3*A*a**2*b*c**2*d**5*log(tan(e + f*x)**2 + 1)*tan(e + f*x
)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7
*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 12*A*a**2*b*c*d**6*
f*x*tan(e + f*x)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*
f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 6*A*
a**2*b*c*d**6*log(c/d + tan(e + f*x))/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e
+ f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f
*tan(e + f*x)) - 3*A*a**2*b*c*d**6*log(tan(e + f*x)**2 + 1)/(2*c**5*d**4*f
+ 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) +
 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 6*A*a**2*b*c*d**6/(2*c**5*d**4*f + 2
*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*
c*d**8*f + 2*d**9*f*tan(e + f*x)) + 6*A*a**2*b*d**7*log(c/d + tan(e + f*x))
*tan(e + f*x)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f +
 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) - 3*A*a**
2*b*d**7*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*c**5*d**4*f + 2*c**4*d**5
*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f +
 2*d**9*f*tan(e + f*x)) - 6*A*a*b**2*c**4*d**3/(2*c**5*d**4*f + 2*c**4*d**5
*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f +
 2*d**9*f*tan(e + f*x)) - 6*A*a*b**2*c**3*d**4*f*x/(2*c**5*d**4*f + 2*c**4*
d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8
*f + 2*d**9*f*tan(e + f*x)) - 6*A*a*b**2*c**2*d**5*f*x*tan(e + f*x)/(2*c**5
*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e
+ f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) - 12*A*a*b**2*c**2*d**5*log(c/
d + tan(e + f*x))/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6
*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 6*A
*a*b**2*c**2*d**5*log(tan(e + f*x)**2 + 1)/(2*c**5*d**4*f + 2*c**4*d**5*f*t
an(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d
```

```
**9*f*tan(e + f*x)) - 6*A*a*b**2*c**2*d**5/(2*c**5*d**4*f + 2*c**4*d**5*f*t
an(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d
**9*f*tan(e + f*x)) + 6*A*a*b**2*c*d**6*f*x/(2*c**5*d**4*f + 2*c**4*d**5*f*
tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*
d**9*f*tan(e + f*x)) - 12*A*a*b**2*c*d**6*log(c/d + tan(e + f*x))*tan(e + f
*x)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d*
*7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 6*A*a*b**2*c*d**6
*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e
 + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*
f*tan(e + f*x)) + 6*A*a*b**2*d**7*f*x*tan(e + f*x)/(2*c**5*d**4*f + 2*c**4*
d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8
*f + 2*d**9*f*tan(e + f*x)) + 2*A*b**3*c**5*d**2*log(c/d + tan(e + f*x))/(2
*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*t
an(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 2*A*b**3*c**5*d**2/(2*c
**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan
(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 2*A*b**3*c**4*d**3*log(c/
d + tan(e + f*x))*tan(e + f*x)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x)
+ 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e
+ f*x)) + 6*A*b**3*c**3*d**4*log(c/d + tan(e + f*x))/(2*c**5*d**4*f + 2*c**
4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d*
*8*f + 2*d**9*f*tan(e + f*x)) - A*b**3*c**3*d**4*log(tan(e + f*x)**2 + 1)/(
2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*
tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 2*A*b**3*c**3*d**4/(2*
c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*ta
n(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) - 4*A*b**3*c**2*d**5*f*x/(
2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*
tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 6*A*b**3*c**2*d**5*log
(c/d + tan(e + f*x))*tan(e + f*x)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*
x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan
(e + f*x)) - A*b**3*c**2*d**5*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*c**5
*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e
+ f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) - 4*A*b**3*c*d**6*f*x*tan(e +
f*x)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d
**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + A*b**3*c*d**6*lo
g(tan(e + f*x)**2 + 1)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3
*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x))
+ A*b**3*d**7*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*c**5*d**4*f + 2*c**4
*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**
8*f + 2*d**9*f*tan(e + f*x)) - 2*B*a**3*c**3*d**4*log(c/d + tan(e + f*x))/(
2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*
tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + B*a**3*c**3*d**4*log(t
an(e + f*x)**2 + 1)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d*
*6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 2
*B*a**3*c**3*d**4/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6
*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 4*B
```

*a**3*c**2*d**5*f*x/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d*
*6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) - 2
*B*a**3*c**2*d**5*log(c/d + tan(e + f*x))*tan(e + f*x)/(2*c**5*d**4*f + 2*c
4*d5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*
d**8*f + 2*d**9*f*tan(e + f*x)) + B*a**3*c**2*d**5*log(tan(e + f*x)**2 + 1)
*tan(e + f*x)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f +
4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 4*B*a**
3*c*d**6*f*x*tan(e + f*x)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c
3*d6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x
)) + 2*B*a**3*c*d**6*log(c/d + tan(e + f*x))/(2*c**5*d**4*f + 2*c**4*d**5*f
*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2
*d**9*f*tan(e + f*x)) - B*a**3*c*d**6*log(tan(e + f*x)**2 + 1)/(2*c**5*d**4
*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x
) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 2*B*a**3*c*d**6/(2*c**5*d**4*f +
2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2
*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 2*B*a**3*d**7*log(c/d + tan(e + f*x))*
tan(e + f*x)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f +
4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) - B*a**3*d
7*log(tan(e + f*x)2 + 1)*tan(e + f*x)/(2*c**5*d**4*f + 2*c**4*d**5*f*ta
n(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d*
*9*f*tan(e + f*x)) - 6*B*a**2*b*c**4*d**3/(2*c**5*d**4*f + 2*c**4*d**5*f*ta
n(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d*
*9*f*tan(e + f*x)) - 6*B*a**2*b*c**3*d**4*f*x/(2*c**5*d**4*f + 2*c**4*d**5*
f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f +
2*d**9*f*tan(e + f*x)) - 6*B*a**2*b*c**2*d**5*f*x*tan(e + f*x)/(2*c**5*d**4
*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x
) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) - 12*B*a**2*b*c**2*d**5*log(c/d + t
an(e + f*x))/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f +
4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 6*B*a**2
*b*c**2*d**5*log(tan(e + f*x)**2 + 1)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e
+ f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f
*tan(e + f*x)) - 6*B*a**2*b*c**2*d**5/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e
+ f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f
*tan(e + f*x)) + 6*B*a**2*b*c*d**6*f*x/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e
+ f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*
f*tan(e + f*x)) - 12*B*a**2*b*c*d**6*log(c/d + tan(e + f*x))*tan(e + f*x)/(
2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*
tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 6*B*a**2*b*c*d**6*log(
tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*
x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan
(e + f*x)) + 6*B*a**2*b*d**7*f*x*tan(e + f*x)/(2*c**5*d**4*f + 2*c**4*d**5*
f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f +
2*d**9*f*tan(e + f*x)) + 6*B*a*b**2*c**5*d**2*log(c/d + tan(e + f*x))/(2*c*
*5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(
e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 6*B*a*b**2*c**5*d**2/(2*c*

```
*5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(
e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 6*B*a*b**2*c**4*d**3*log(c
/d + tan(e + f*x))*tan(e + f*x)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x)
 + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e
 + f*x)) + 18*B*a*b**2*c**3*d**4*log(c/d + tan(e + f*x))/(2*c**5*d**4*f + 2
*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*
c*d**8*f + 2*d**9*f*tan(e + f*x)) - 3*B*a*b**2*c**3*d**4*log(tan(e + f*x)**
2 + 1)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2
*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 6*B*a*b**2*c**
3*d**4/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2
*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) - 12*B*a*b**2*c*
*2*d**5*f*x/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4
*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 18*B*a*b*
*2*c**2*d**5*log(c/d + tan(e + f*x))*tan(e + f*x)/(2*c**5*d**4*f + 2*c**4*d
**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*
f + 2*d**9*f*tan(e + f*x)) - 3*B*a*b**2*c**2*d**5*log(tan(e + f*x)**2 + 1)*
tan(e + f*x)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f +
4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) - 12*B*a*b
**2*c*d**6*f*x*tan(e + f*x)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4
*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f
*x)) + 3*B*a*b**2*c*d**6*log(tan(e + f*x)**2 + 1)/(2*c**5*d**4*f + 2*c**4*d
**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*
f + 2*d**9*f*tan(e + f*x)) + 3*B*a*b**2*d**7*log(tan(e + f*x)**2 + 1)*tan(e
 + f*x)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**
2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) - 4*B*b**3*c**6
*d*log(c/d + tan(e + f*x))/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*
c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*
x)) - 4*B*b**3*c**6*d/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*
d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) -
 4*B*b**3*c**5*d**2*log(c/d + tan(e + f*x))*tan(e + f*x)/(2*c**5*d**4*f + 2
*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*
c*d**8*f + 2*d**9*f*tan(e + f*x)) - 8*B*b**3*c**4*d**3*log(c/d + tan(e + f*
x))/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d*
*7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 2*B*b**3*c**4*d**
3*tan(e + f*x)**2/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6
*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) - 6*B
*b**3*c**4*d**3/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f
 + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 2*B*b
**3*c**3*d**4*f*x/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6
*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) - 8*B
*b**3*c**3*d**4*log(c/d + tan(e + f*x))*tan(e + f*x)/(2*c**5*d**4*f + 2*c**
4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d*
*8*f + 2*d**9*f*tan(e + f*x)) + 2*B*b**3*c**2*d**5*f*x*tan(e + f*x)/(2*c**5
*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e
+ f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) - 2*B*b**3*c**2*d**5*log(tan(e
```

```
 + f*x)**2 + 1)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f
 + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 4*B*b
**3*c**2*d**5*tan(e + f*x)**2/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) +
 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e +
 f*x)) - 2*B*b**3*c**2*d**5/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4
*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f
*x)) - 2*B*b**3*c*d**6*f*x/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*
c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*
x)) - 2*B*b**3*c*d**6*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*c**5*d**4*f
+ 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) +
 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) - 2*B*b**3*d**7*f*x*tan(e + f*x)/(2*c*
*5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(
e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 2*B*b**3*d**7*tan(e + f*x)
**2/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d*
*7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) - 2*C*a**3*c**4*d**
3/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7
*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) - 2*C*a**3*c**3*d**4*
f*x/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d*
*7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) - 2*C*a**3*c**2*d**
5*f*x*tan(e + f*x)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**
6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) - 4*
C*a**3*c**2*d**5*log(c/d + tan(e + f*x))/(2*c**5*d**4*f + 2*c**4*d**5*f*tan
(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**
9*f*tan(e + f*x)) + 2*C*a**3*c**2*d**5*log(tan(e + f*x)**2 + 1)/(2*c**5*d**
4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*
x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) - 2*C*a**3*c**2*d**5/(2*c**5*d**4*
f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x)
 + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 2*C*a**3*c*d**6*f*x/(2*c**5*d**4*f
 + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x)
+ 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) - 4*C*a**3*c*d**6*log(c/d + tan(e + f
*x))*tan(e + f*x)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6
*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 2*C
*a**3*c*d**6*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*c**5*d**4*f + 2*c**4*
d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8
*f + 2*d**9*f*tan(e + f*x)) + 2*C*a**3*d**7*f*x*tan(e + f*x)/(2*c**5*d**4*f
 + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x)
+ 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 6*C*a**2*b*c**5*d**2*log(c/d + tan(
e + f*x))/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c
**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 6*C*a**2*b*
c**5*d**2/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c
**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 6*C*a**2*b*
c**4*d**3*log(c/d + tan(e + f*x))*tan(e + f*x)/(2*c**5*d**4*f + 2*c**4*d**5
*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f +
 2*d**9*f*tan(e + f*x)) + 18*C*a**2*b*c**3*d**4*log(c/d + tan(e + f*x))/(2*
c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*ta
```

n(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) - 3*C*a**2*b*c**3*d**4*log
(tan(e + f*x)**2 + 1)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*
d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) +
6*C*a**2*b*c**3*d**4/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*
d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) -
12*C*a**2*b*c**2*d**5*f*x/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*
c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*
x)) + 18*C*a**2*b*c**2*d**5*log(c/d + tan(e + f*x))*tan(e + f*x)/(2*c**5*d*
*4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f
*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) - 3*C*a**2*b*c**2*d**5*log(tan(e
+ f*x)**2 + 1)*tan(e + f*x)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4
*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f
*x)) - 12*C*a**2*b*c*d**6*f*x*tan(e + f*x)/(2*c**5*d**4*f + 2*c**4*d**5*f*t
an(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d
9*f*tan(e + f*x)) + 3*C*a2*b*c*d**6*log(tan(e + f*x)**2 + 1)/(2*c**5*d*
*4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f
*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 3*C*a**2*b*d**7*log(tan(e + f*x
)**2 + 1)*tan(e + f*x)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3
*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x))
- 12*C*a*b**2*c**6*d*log(c/d + tan(e + f*x))/(2*c**5*d**4*f + 2*c**4*d**5*f
*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2
*d**9*f*tan(e + f*x)) - 12*C*a*b**2*c**6*d/(2*c**5*d**4*f + 2*c**4*d**5*f*t
an(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d
9*f*tan(e + f*x)) - 12*C*a*b2*c**5*d**2*log(c/d + tan(e + f*x))*tan(e +
f*x)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*
d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) - 24*C*a*b**2*c**
4*d**3*log(c/d + tan(e + f*x))/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x)
+ 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e
+ f*x)) + 6*C*a*b**2*c**4*d**3*tan(e + f*x)**2/(2*c**5*d**4*f + 2*c**4*d**5
*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f +
2*d**9*f*tan(e + f*x)) - 18*C*a*b**2*c**4*d**3/(2*c**5*d**4*f + 2*c**4*d**
5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f
+ 2*d**9*f*tan(e + f*x)) + 6*C*a*b**2*c**3*d**4*f*x/(2*c**5*d**4*f + 2*c**4
*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**
8*f + 2*d**9*f*tan(e + f*x)) - 24*C*a*b**2*c**3*d**4*log(c/d + tan(e + f*x)
)*tan(e + f*x)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f
+ 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 6*C*a*
b**2*c**2*d**5*f*x*tan(e + f*x)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x)
+ 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e
+ f*x)) - 6*C*a*b**2*c**2*d**5*log(tan(e + f*x)**2 + 1)/(2*c**5*d**4*f + 2
*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*
c*d**8*f + 2*d**9*f*tan(e + f*x)) + 12*C*a*b**2*c**2*d**5*tan(e + f*x)**2/(
2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*
tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) - 6*C*a*b**2*c**2*d**5/(
2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*

tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) - 6*C*a*b**2*c*d**6*f*x/
(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f
*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) - 6*C*a*b**2*c*d**6*log
(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f
*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*ta
n(e + f*x)) - 6*C*a*b**2*d**7*f*x*tan(e + f*x)/(2*c**5*d**4*f + 2*c**4*d**5
*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f +
2*d**9*f*tan(e + f*x)) + 6*C*a*b**2*d**7*tan(e + f*x)**2/(2*c**5*d**4*f +
2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2
*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 6*C*b**3*c**7*log(c/d + tan(e + f*x))/
(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f
*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 6*C*b**3*c**7/(2*c**5
*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e
+ f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 6*C*b**3*c**6*d*log(c/d + ta
n(e + f*x))*tan(e + f*x)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c*
*3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)
) + 10*C*b**3*c**5*d**2*log(c/d + tan(e + f*x))/(2*c**5*d**4*f + 2*c**4*d**
5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f
+ 2*d**9*f*tan(e + f*x)) - 3*C*b**3*c**5*d**2*tan(e + f*x)**2/(2*c**5*d**4*
f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x)
+ 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 10*C*b**3*c**5*d**2/(2*c**5*d**4*f
+ 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x)
+ 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 10*C*b**3*c**4*d**3*log(c/d + tan(e
+ f*x))*tan(e + f*x)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*
d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) +
C*b**3*c**4*d**3*tan(e + f*x)**3/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*
x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan
(e + f*x)) + C*b**3*c**3*d**4*log(tan(e + f*x)**2 + 1)/(2*c**5*d**4*f + 2*c
4*d5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*
d**8*f + 2*d**9*f*tan(e + f*x)) - 6*C*b**3*c**3*d**4*tan(e + f*x)**2/(2*c**
5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e
+ f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 4*C*b**3*c**3*d**4/(2*c**5*
d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e +
f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + 4*C*b**3*c**2*d**5*f*x/(2*c**
5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e
+ f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) + C*b**3*c**2*d**5*log(tan(e
+ f*x)**2 + 1)*tan(e + f*x)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4
*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f
*x)) + 2*C*b**3*c**2*d**5*tan(e + f*x)**3/(2*c**5*d**4*f + 2*c**4*d**5*f*ta
n(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d*
*9*f*tan(e + f*x)) + 4*C*b**3*c*d**6*f*x*tan(e + f*x)/(2*c**5*d**4*f + 2*c*
*4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d
8*f + 2*d9*f*tan(e + f*x)) - C*b**3*c*d**6*log(tan(e + f*x)**2 + 1)/(2*
c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*ta
n(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) - 3*C*b**3*c*d**6*tan(e +

```
f*x)**2/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e + f*x) + 4*c**3*d**6*f + 4*c**
2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*tan(e + f*x)) - C*b**3*d**7*l
og(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*c**5*d**4*f + 2*c**4*d**5*f*tan(e +
 f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*d**9*f*
tan(e + f*x)) + C*b**3*d**7*tan(e + f*x)**3/(2*c**5*d**4*f + 2*c**4*d**5*f*
tan(e + f*x) + 4*c**3*d**6*f + 4*c**2*d**7*f*tan(e + f*x) + 2*c*d**8*f + 2*
d**9*f*tan(e + f*x)), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.34 (sec) , antiderivative size = 684, normalized size of antiderivative = 1.18

$$\int \frac{(a+b\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^2}\,dx$$

$$= \frac{\frac{2\left(\left(((A-C)a^3-3\,Ba^2b-3\,(A-C)ab^2+Bb^3\right)c^2+2\left(Ba^3+3\,(A-C)a^2b-3\,Bab^2-(A-C)b^3\right)cd-\left(((A-C)a^3-3\,Ba^2b-3\,(A-C)ab^2+Bb^3\right)d^2\right)(fx+e)}{c^4+2\,c^2d^2+d^4}}{}$$

```
[In] integrate((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^2,x, algorithm="maxima")
```

```
[Out] 1/2*(2*(((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b^3)*c^2 + 2*(B*a^3
+ 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*c*d - ((A - C)*a^3 - 3*B*a^2*b
 - 3*(A - C)*a*b^2 + B*b^3)*d^2)*(f*x + e)/(c^4 + 2*c^2*d^2 + d^4) + 2*(3*C
*b^3*c^6 - 2*(3*C*a*b^2 + B*b^3)*c^5*d + (3*C*a^2*b + 3*B*a*b^2 + (A + 5*C)
*b^3)*c^4*d^2 - 4*(3*C*a*b^2 + B*b^3)*c^3*d^3 - (B*a^3 + 3*(A - 3*C)*a^2*b
- 9*B*a*b^2 - 3*A*b^3)*c^2*d^4 + 2*((A - C)*a^3 - 3*B*a^2*b - 3*A*a*b^2)*c*
d^5 + (B*a^3 + 3*A*a^2*b)*d^6)*log(d*tan(f*x + e) + c)/(c^4*d^4 + 2*c^2*d^6
 + d^8) + ((B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*c^2 - 2*((A
- C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b^3)*c*d - (B*a^3 + 3*(A - C)*a^
2*b - 3*B*a*b^2 - (A - C)*b^3)*d^2)*log(tan(f*x + e)^2 + 1)/(c^4 + 2*c^2*d^
2 + d^4) + 2*(C*b^3*c^5 - A*a^3*d^5 - (3*C*a*b^2 + B*b^3)*c^4*d + (3*C*a^2*
b + 3*B*a*b^2 + A*b^3)*c^3*d^2 - (C*a^3 + 3*B*a^2*b + 3*A*a*b^2)*c^2*d^3 +
(B*a^3 + 3*A*a^2*b)*c*d^4)/(c^3*d^4 + c*d^6 + (c^2*d^5 + d^7)*tan(f*x + e))
 + (C*b^3*d*tan(f*x + e)^2 - 2*(2*C*b^3*c - (3*C*a*b^2 + B*b^3)*d)*tan(f*x
+ e))/d^3)/f
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1327 vs. 2(577) = 1154.

Time = 1.14 (sec) , antiderivative size = 1327, normalized size of antiderivative = 2.29

$$\int \frac{(a+b\tan(e+fx))^3 (A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^2}\,dx = \text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^2,x, algorithm="giac")
```

```
[Out] 1/2*(2*(A*a^3*c^2 - C*a^3*c^2 - 3*B*a^2*b*c^2 - 3*A*a*b^2*c^2 + 3*C*a*b^2*c
^2 + B*b^3*c^2 + 2*B*a^3*c*d + 6*A*a^2*b*c*d - 6*C*a^2*b*c*d - 6*B*a*b^2*c*
d - 2*A*b^3*c*d + 2*C*b^3*c*d - A*a^3*d^2 + C*a^3*d^2 + 3*B*a^2*b*d^2 + 3*A
*a*b^2*d^2 - 3*C*a*b^2*d^2 - B*b^3*d^2)*(f*x + e)/(c^4 + 2*c^2*d^2 + d^4) +
 (B*a^3*c^2 + 3*A*a^2*b*c^2 - 3*C*a^2*b*c^2 - 3*B*a*b^2*c^2 - A*b^3*c^2 + C
*b^3*c^2 - 2*A*a^3*c*d + 2*C*a^3*c*d + 6*B*a^2*b*c*d + 6*A*a*b^2*c*d - 6*C*
a*b^2*c*d - 2*B*b^3*c*d - B*a^3*d^2 - 3*A*a^2*b*d^2 + 3*C*a^2*b*d^2 + 3*B*a
*b^2*d^2 + A*b^3*d^2 - C*b^3*d^2)*log(tan(f*x + e)^2 + 1)/(c^4 + 2*c^2*d^2
+ d^4) + 2*(3*C*b^3*c^6 - 6*C*a*b^2*c^5*d - 2*B*b^3*c^5*d + 3*C*a^2*b*c^4*d
^2 + 3*B*a*b^2*c^4*d^2 + A*b^3*c^4*d^2 + 5*C*b^3*c^4*d^2 - 12*C*a*b^2*c^3*d
^3 - 4*B*b^3*c^3*d^3 - B*a^3*c^2*d^4 - 3*A*a^2*b*c^2*d^4 + 9*C*a^2*b*c^2*d^
4 + 9*B*a*b^2*c^2*d^4 + 3*A*b^3*c^2*d^4 + 2*A*a^3*c*d^5 - 2*C*a^3*c*d^5 - 6
*B*a^2*b*c*d^5 - 6*A*a*b^2*c*d^5 + B*a^3*d^6 + 3*A*a^2*b*d^6)*log(abs(d*tan
(f*x + e) + c))/(c^4*d^4 + 2*c^2*d^6 + d^8) - 2*(3*C*b^3*c^6*d*tan(f*x + e)
 - 6*C*a*b^2*c^5*d^2*tan(f*x + e) - 2*B*b^3*c^5*d^2*tan(f*x + e) + 3*C*a^2*
b*c^4*d^3*tan(f*x + e) + 3*B*a*b^2*c^4*d^3*tan(f*x + e) + A*b^3*c^4*d^3*tan
(f*x + e) + 5*C*b^3*c^4*d^3*tan(f*x + e) - 12*C*a*b^2*c^3*d^4*tan(f*x + e)
- 4*B*b^3*c^3*d^4*tan(f*x + e) - B*a^3*c^2*d^5*tan(f*x + e) - 3*A*a^2*b*c^2
*d^5*tan(f*x + e) + 9*C*a^2*b*c^2*d^5*tan(f*x + e) + 9*B*a*b^2*c^2*d^5*tan(
f*x + e) + 3*A*b^3*c^2*d^5*tan(f*x + e) + 2*A*a^3*c*d^6*tan(f*x + e) - 2*C*
a^3*c*d^6*tan(f*x + e) - 6*B*a^2*b*c*d^6*tan(f*x + e) - 6*A*a*b^2*c*d^6*tan
(f*x + e) + B*a^3*d^7*tan(f*x + e) + 3*A*a^2*b*d^7*tan(f*x + e) + 2*C*b^3*c
^7 - 3*C*a*b^2*c^6*d - B*b^3*c^6*d + 4*C*b^3*c^5*d^2 + C*a^3*c^4*d^3 + 3*B*
a^2*b*c^4*d^3 + 3*A*a*b^2*c^4*d^3 - 9*C*a*b^2*c^4*d^3 - 3*B*b^3*c^4*d^3 - 2
*B*a^3*c^3*d^4 - 6*A*a^2*b*c^3*d^4 + 6*C*a^2*b*c^3*d^4 + 6*B*a*b^2*c^3*d^4
+ 2*A*b^3*c^3*d^4 + 3*A*a^3*c^2*d^5 - C*a^3*c^2*d^5 - 3*B*a^2*b*c^2*d^5 - 3
*A*a*b^2*c^2*d^5 + A*a^3*d^7)/((c^4*d^4 + 2*c^2*d^6 + d^8)*(d*tan(f*x + e)
+ c)) + (C*b^3*d^2*tan(f*x + e)^2 - 4*C*b^3*c*d*tan(f*x + e) + 6*C*a*b^2*d^
2*tan(f*x + e) + 2*B*b^3*d^2*tan(f*x + e))/d^4)/f
```

Mupad [B] (verification not implemented)

Time = 15.62 (sec) , antiderivative size = 701, normalized size of antiderivative = 1.21

$$\int \frac{(a+b\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^2}\,dx$$

$$= \frac{\tan(e+f\,x)\,\left(\frac{B\,b^3+3\,C\,a\,b^2}{d^2}-\frac{2\,C\,b^3\,c}{d^3}\right)}{f}$$

$$-\frac{\ln\left(\tan(e+f\,x)+1\mathrm{i}\right)\,\left(B\,a^3-A\,b^3+C\,b^3+3\,A\,a^2\,b-3\,B\,a\,b^2-3\,C\,a^2\,b+A\,a^3\,1\mathrm{i}+B\,b^3\,1\mathrm{i}-C\,a^3\,1\mathrm{i}\right.}{2\,f\,\left(-c^2+c\,d\,2\mathrm{i}+d^2\right)}$$

$$+\frac{\ln\left(c+d\tan(e+f\,x)\right)\,\left(d^4\,\left(3\,A\,b^3\,c^2-B\,a^3\,c^2-3\,A\,a^2\,b\,c^2+9\,B\,a\,b^2\,c^2+9\,C\,a^2\,b\,c^2\right)-d^5\,\left(2\,C\,a^3\,c-2\right.\right.}{}$$

$$-\frac{\ln\left(\tan(e+f\,x)-\mathrm{i}\right)\,\left(A\,a^3-A\,b^3\,1\mathrm{i}+B\,a^3\,1\mathrm{i}+B\,b^3-C\,a^3+C\,b^3\,1\mathrm{i}-3\,A\,a\,b^2+A\,a^2\,b\,3\mathrm{i}-B\,a\,b^2\,3\mathrm{i}\right.}{2\,f\,\left(-c^2\,1\mathrm{i}+2\,c\,d+d^2\,1\mathrm{i}\right)}$$

$$-\frac{C\,a^3\,c^2\,d^3-B\,a^3\,c\,d^4+A\,a^3\,d^5-3\,C\,a^2\,b\,c^3\,d^2+3\,B\,a^2\,b\,c^2\,d^3-3\,A\,a^2\,b\,c\,d^4+3\,C\,a\,b^2\,c^4\,d-3\,B\,a\,b^2\,c^3}{d\,f\,\left(\tan\left(e+f\,x\right)\,d^4+c\,d^3\right)\,\left(c^2+d^2\right)}$$

$$+\frac{C\,b^3\tan(e+f\,x)^2}{2\,d^2\,f}$$

```
[In] int(((a + b*tan(e + f*x))^3*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d
*tan(e + f*x))^2,x)
```

```
[Out] (tan(e + f*x)*((B*b^3 + 3*C*a*b^2)/d^2 - (2*C*b^3*c)/d^3))/f - (log(tan(e +
 f*x) + 1i)*(A*a^3*1i - A*b^3 + B*a^3 + B*b^3*1i - C*a^3*1i + C*b^3 - A*a*b
^2*3i + 3*A*a^2*b - 3*B*a*b^2 - B*a^2*b*3i + C*a*b^2*3i - 3*C*a^2*b))/(2*f*
(c*d*2i - c^2 + d^2)) + (log(c + d*tan(e + f*x))*(d^4*(3*A*b^3*c^2 - B*a^3*
c^2 - 3*A*a^2*b*c^2 + 9*B*a*b^2*c^2 + 9*C*a^2*b*c^2) - d^5*(2*C*a^3*c - 2*A
*a^3*c + 6*A*a*b^2*c + 6*B*a^2*b*c) - d^3*(4*B*b^3*c^3 + 12*C*a*b^2*c^3) +
d^6*(B*a^3 + 3*A*a^2*b) - d*(2*B*b^3*c^5 + 6*C*a*b^2*c^5) + d^2*(A*b^3*c^4
+ 5*C*b^3*c^4 + 3*B*a*b^2*c^4 + 3*C*a^2*b*c^4) + 3*C*b^3*c^6))/(f*(d^8 + 2*
c^2*d^6 + c^4*d^4)) - (log(tan(e + f*x) - 1i)*(A*a^3 - A*b^3*1i + B*a^3*1i
+ B*b^3 - C*a^3 + C*b^3*1i - 3*A*a*b^2 + A*a^2*b*3i - B*a*b^2*3i - 3*B*a^2*
b + 3*C*a*b^2 - C*a^2*b*3i))/(2*f*(2*c*d - c^2*1i + d^2*1i)) - (A*a^3*d^5 -
 C*b^3*c^5 - B*a^3*c*d^4 + B*b^3*c^4*d - A*b^3*c^3*d^2 + C*a^3*c^2*d^3 + 3*
A*a*b^2*c^2*d^3 - 3*B*a*b^2*c^3*d^2 + 3*B*a^2*b*c^2*d^3 - 3*C*a^2*b*c^3*d^2
 - 3*A*a^2*b*c*d^4 + 3*C*a*b^2*c^4*d)/(d*f*(c*d^3 + d^4*tan(e + f*x))*(c^2
+ d^2)) + (C*b^3*tan(e + f*x)^2)/(2*d^2*f)
```

3.78 $\int \frac{(a+b\tan(e+fx))^2(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^2}\,dx$

Optimal result 794
Rubi [A] (verified) 795
Mathematica [C] (verified) 798
Maple [A] (verified) 798
Fricas [B] (verification not implemented) 799
Sympy [C] (verification not implemented) 800
Maxima [A] (verification not implemented) 808
Giac [B] (verification not implemented) 809
Mupad [B] (verification not implemented) 809

Optimal result

Integrand size = 45, antiderivative size = 417

$$\int \frac{(a+b\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^2}\,dx =$$
$$-\frac{(a^2(c^2C-2Bcd-Cd^2-A(c^2-d^2))-b^2(c^2C-2Bcd-Cd^2-A(c^2-d^2))-2ab(2c(A-C)d-B(c^2}{(c^2+d^2)^2}$$
$$+\frac{(2ab(c^2C-2Bcd-Cd^2-A(c^2-d^2))+a^2(2c(A-C)d-B(c^2-d^2))-b^2(2c(A-C)d-B(c^2-d^2))}{(c^2+d^2)^2\,f}$$
$$-\frac{(bc-ad)\,(b(2c^4C-Bc^3d+4c^2Cd^2-3Bcd^3+2Ad^4)+ad^2(2c(A-C)d-B(c^2-d^2)))\log(c+d\tan(e}{d^3\,(c^2+d^2)^2\,f}$$
$$+\frac{b^2(2c^2C-Bcd+(A+C)d^2)\tan(e+fx)}{d^2\,(c^2+d^2)\,f}-\frac{(c^2C-Bcd+Ad^2)\,(a+b\tan(e+fx))^2}{d\,(c^2+d^2)\,f(c+d\tan(e+fx))}$$

```
[Out] -(a^2*(c^2*C-2*B*c*d-C*d^2-A*(c^2-d^2))-b^2*(c^2*C-2*B*c*d-C*d^2-A*(c^2-d^2
))-2*a*b*(2*c*(A-C)*d-B*(c^2-d^2)))*x/(c^2+d^2)^2+(2*a*b*(c^2*C-2*B*c*d-C*d
^2-A*(c^2-d^2))+a^2*(2*c*(A-C)*d-B*(c^2-d^2))-b^2*(2*c*(A-C)*d-B*(c^2-d^2))
)*ln(cos(f*x+e))/(c^2+d^2)^2/f-(-a*d+b*c)*(b*(2*A*d^4-B*c^3*d-3*B*c*d^3+2*C
*c^4+4*C*c^2*d^2)+a*d^2*(2*c*(A-C)*d-B*(c^2-d^2)))*ln(c+d*tan(f*x+e))/d^3/(
c^2+d^2)^2/f+b^2*(2*c^2*C-B*c*d+(A+C)*d^2)*tan(f*x+e)/d^2/(c^2+d^2)/f-(A*d^
2-B*c*d+C*c^2)*(a+b*tan(f*x+e))^2/d/(c^2+d^2)/f/(c+d*tan(f*x+e))
```

Rubi [A] (verified)

Time = 1.20 (sec) , antiderivative size = 417, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}} = 0.133$, Rules used = {3726, 3718, 3707, 3698, 31, 3556}

$$\int \frac{(a+b\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^2}\,dx$$
$$= \frac{\log(\cos(e+fx))\,(a^2(2cd(A-C)-B(c^2-d^2))+2ab(-A(c^2-d^2)-2Bcd+c^2C-Cd^2)-b^2(2cd(A-}{f\,(c^2+d^2)^2}$$
$$-\frac{x(a^2(-A(c^2-d^2)-2Bcd+c^2C-Cd^2)-2ab(2cd(A-C)-B(c^2-d^2))-b^2(-A(c^2-d^2)-2Bcd}{(c^2+d^2)^2}$$
$$-\frac{(Ad^2-Bcd+c^2C)\,(a+b\tan(e+fx))^2}{df\,(c^2+d^2)\,(c+d\tan(e+fx))}$$
$$-\frac{(bc-ad)\,(ad^2(2cd(A-C)-B(c^2-d^2))+b(2Ad^4-Bc^3d-3Bcd^3+2c^4C+4c^2Cd^2))\log(c+d\tan(}{d^3f\,(c^2+d^2)^2}$$
$$+\frac{b^2\tan(e+fx)\,(d^2(A+C)-Bcd+2c^2C)}{d^2f\,(c^2+d^2)}$$

```
[In] Int[((a + b*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c + d
*Tan[e + f*x])^2,x]
```

```
[Out] -(((a^2*(c^2*C - 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - b^2*(c^2*C - 2*B*c*d -
C*d^2 - A*(c^2 - d^2)) - 2*a*b*(2*c*(A - C)*d - B*(c^2 - d^2)))*x)/(c^2 + d
^2)^2) + ((2*a*b*(c^2*C - 2*B*c*d - C*d^2 - A*(c^2 - d^2)) + a^2*(2*c*(A -
C)*d - B*(c^2 - d^2)) - b^2*(2*c*(A - C)*d - B*(c^2 - d^2)))*Log[Cos[e + f*
x]])/((c^2 + d^2)^2*f) - ((b*c - a*d)*(b*(2*c^4*C - B*c^3*d + 4*c^2*C*d^2 -
 3*B*c*d^3 + 2*A*d^4) + a*d^2*(2*c*(A - C)*d - B*(c^2 - d^2)))*Log[c + d*Ta
n[e + f*x]])/(d^3*(c^2 + d^2)^2*f) + (b^2*(2*c^2*C - B*c*d + (A + C)*d^2)*T
an[e + f*x])/(d^2*(c^2 + d^2)*f) - ((c^2*C - B*c*d + A*d^2)*(a + b*Tan[e +
f*x])^2)/(d*(c^2 + d^2)*f*(c + d*Tan[e + f*x]))
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3698

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
```

```
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3707

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[(a*A + b*B - a
*C)*(x/(a^2 + b^2)), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3726

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*d^2 + c*(c*C - B*d))*(a + b*Tan[e +
 f*x])^m*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rubi steps

$$\text{integral} = -\frac{(c^2C - Bcd + Ad^2)\,(a + b\tan(e + fx))^2}{d\,(c^2 + d^2)\,f(c + d\tan(e + fx))} + \frac{\int \frac{(a+b\tan(e+fx))\left(Ad(ac+2bd)+(2bc-ad)(cC-Bd)+d((A-C)(bc-ad)+B(ac+bd))\tan(e+fx)+b\left(2c^2C-Bcd+(A+C)d^2\right)\tan^2(e+fx)\right)}{c+d\tan(e+fx)}\,dx}{d\,(c^2 + d^2)}$$

$$=\frac{b^2(2c^2C-Bcd+(A+C)d^2)\tan(e+fx)}{d^2(c^2+d^2)f}-\frac{(c^2C-Bcd+Ad^2)(a+b\tan(e+fx))^2}{d(c^2+d^2)f(c+d\tan(e+fx))}$$

$$-\frac{\int\frac{b^2c(2c^2C-Bcd+(A+C)d^2)-ad(Ad(ac+2bd)+(2bc-ad)(cC-Bd))-d^2(2ab(Ac-cC+Bd)+a^2(Bc-(A-C)d)-b^2(Bc-(A-C)d}{c+d\tan(e+fx)}}{d^2(c^2+d^2)}$$

$$=$$

$$-\frac{(a^2(c^2C-2Bcd-Cd^2-A(c^2-d^2))-b^2(c^2C-2Bcd-Cd^2-A(c^2-d^2))-2ab(2c(A-C)}{(c^2+d^2)^2}$$

$$+\frac{b^2(2c^2C-Bcd+(A+C)d^2)\tan(e+fx)}{d^2(c^2+d^2)f}$$

$$-\frac{(c^2C-Bcd+Ad^2)(a+b\tan(e+fx))^2}{d(c^2+d^2)f(c+d\tan(e+fx))}$$

$$-\frac{(2ab(c^2C-2Bcd-Cd^2-A(c^2-d^2))+a^2(2c(A-C)d-B(c^2-d^2))-b^2(2c(A-C)d-B(}{(c^2+d^2)^2}$$

$$-\frac{((bc-ad)(b(2c^4C-Bc^3d+4c^2Cd^2-3Bcd^3+2Ad^4)+ad^2(2c(A-C)d-B(c^2-d^2))))\int\frac{1+}{c+}}{d^2(c^2+d^2)^2}$$

$$=$$

$$-\frac{(a^2(c^2C-2Bcd-Cd^2-A(c^2-d^2))-b^2(c^2C-2Bcd-Cd^2-A(c^2-d^2))-2ab(2c(A-C)}{(c^2+d^2)^2}$$

$$+\frac{(2ab(c^2C-2Bcd-Cd^2-A(c^2-d^2))+a^2(2c(A-C)d-B(c^2-d^2))-b^2(2c(A-C)d-B(}{(c^2+d^2)^2f}$$

$$+\frac{b^2(2c^2C-Bcd+(A+C)d^2)\tan(e+fx)}{d^2(c^2+d^2)f}$$

$$-\frac{(c^2C-Bcd+Ad^2)(a+b\tan(e+fx))^2}{d(c^2+d^2)f(c+d\tan(e+fx))}$$

$$-\frac{((bc-ad)(b(2c^4C-Bc^3d+4c^2Cd^2-3Bcd^3+2Ad^4)+ad^2(2c(A-C)d-B(c^2-d^2))))\text{Sub}}{d^3(c^2+d^2)^2f}$$

$$=$$

$$-\frac{(a^2(c^2C-2Bcd-Cd^2-A(c^2-d^2))-b^2(c^2C-2Bcd-Cd^2-A(c^2-d^2))-2ab(2c(A-C)}{(c^2+d^2)^2}$$

$$+\frac{(2ab(c^2C-2Bcd-Cd^2-A(c^2-d^2))+a^2(2c(A-C)d-B(c^2-d^2))-b^2(2c(A-C)d-B(}{(c^2+d^2)^2f}$$

$$-\frac{(bc-ad)(b(2c^4C-Bc^3d+4c^2Cd^2-3Bcd^3+2Ad^4)+ad^2(2c(A-C)d-B(c^2-d^2)))\log(c}{d^3(c^2+d^2)^2f}$$

$$+\frac{b^2(2c^2C-Bcd+(A+C)d^2)\tan(e+fx)}{d^2(c^2+d^2)f}$$

$$-\frac{(c^2C-Bcd+Ad^2)(a+b\tan(e+fx))^2}{d(c^2+d^2)f(c+d\tan(e+fx))}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 5.57 (sec) , antiderivative size = 277, normalized size of antiderivative = 0.66

$$\int \frac{(a+b\tan(e+fx))^2 \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^2}\,dx$$
$$= \frac{\frac{(a+ib)^2(-iA+B+iC)\log(i-\tan(e+fx))}{(c+id)^2} + \frac{(a-ib)^2(iA+B-iC)\log(i+\tan(e+fx))}{(c-id)^2} + \frac{2(-bc+ad)\left(b\left(2c^4C-Bc^3d+4c^2Cd^2-3Bcd^3+2Ad^4\right)+ad}{d^3(c^2+d^2}}{2f}$$

```
[In] Integrate[((a + b*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(c + d*Tan[e + f*x])^2,x]
```

```
[Out] (((a + I*b)^2*((-I)*A + B + I*C)*Log[I - Tan[e + f*x]])/(c + I*d)^2 + ((a -
 I*b)^2*(I*A + B - I*C)*Log[I + Tan[e + f*x]])/(c - I*d)^2 + (2*(-(b*c) + a
*d)*(b*(2*c^4*C - B*c^3*d + 4*c^2*C*d^2 - 3*B*c*d^3 + 2*A*d^4) + a*d^2*(2*c
*(A - C)*d + B*(-c^2 + d^2)))*Log[c + d*Tan[e + f*x]])/(d^3*(c^2 + d^2)^2)
- (2*(b*c - a*d)^2*(2*c^2*C - B*c*d + (A + C)*d^2))/(d^3*(c^2 + d^2)*(c + d
*Tan[e + f*x])) + (2*C*(a + b*Tan[e + f*x])^2)/(d*(c + d*Tan[e + f*x])))/(2
*f)
```

Maple [A] (verified)

Time = 0.27 (sec) , antiderivative size = 552, normalized size of antiderivative = 1.32

method	result
derivativedivides	$\frac{\frac{\tan(fx+e)C\,b^2}{d^2}+\frac{\frac{\left(-2A\,a^2cd+2Aab\,c^2-2Aab\,d^2+2A\,b^2cd+B\,a^2c^2-B\,a^2d^2+4Babcd-B\,b^2c^2+B\,b^2d^2+2C\,a^2cd-2Cab\,c^2+2Cab\,d^2-2}{2}}{}}{}$
default	$\frac{\frac{\tan(fx+e)C\,b^2}{d^2}+\frac{\frac{\left(-2A\,a^2cd+2Aab\,c^2-2Aab\,d^2+2A\,b^2cd+B\,a^2c^2-B\,a^2d^2+4Babcd-B\,b^2c^2+B\,b^2d^2+2C\,a^2cd-2Cab\,c^2+2Cab\,d^2-2}{2}}{}}{}$
norman	$\frac{\frac{c\left(A\,a^2c^2-A\,a^2d^2+4Aabcd-A\,b^2c^2+A\,b^2d^2+2B\,a^2cd-2Bab\,c^2+2Bab\,d^2-2B\,b^2cd-C\,a^2c^2+a^2C\,d^2-4Cabcd+C\,b^2c^2-C\,b^2d^2\right)x}{c^4+2c^2d^2+d^4}+C}{}$
parallelrisch	Expression too large to display
risch	Expression too large to display

```
[In] int((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^2,x
,method=_RETURNVERBOSE)
```

```
[Out] 1/f*(tan(f*x+e)*C*b^2/d^2+1/(c^2+d^2)^2*(1/2*(-2*A*a^2*c*d+2*A*a*b*c^2-2*A*
a*b*d^2+2*A*b^2*c*d+B*a^2*c^2-B*a^2*d^2+4*B*a*b*c*d-B*b^2*c^2+B*b^2*d^2+2*C
*a^2*c*d-2*C*a*b*c^2+2*C*a*b*d^2-2*C*b^2*c*d)*ln(1+tan(f*x+e)^2)+(A*a^2*c^2
```

```
-A*a^2*d^2+4*A*a*b*c*d-A*b^2*c^2+A*b^2*d^2+2*B*a^2*c*d-2*B*a*b*c^2+2*B*a*b*
d^2-2*B*b^2*c*d-C*a^2*c^2+C*a^2*d^2-4*C*a*b*c*d+C*b^2*c^2-C*b^2*d^2)*arctan
(tan(f*x+e)))-1/d^3*(A*a^2*d^4-2*A*a*b*c*d^3+A*b^2*c^2*d^2-B*a^2*c*d^3+2*B*
a*b*c^2*d^2-B*b^2*c^3*d+C*a^2*c^2*d^2-2*C*a*b*c^3*d+C*b^2*c^4)/(c^2+d^2)/(c
+d*tan(f*x+e))+1/d^3*(2*A*a^2*c*d^4-2*A*a*b*c^2*d^3+2*A*a*b*d^5-2*A*b^2*c*d
^4-B*a^2*c^2*d^3+B*a^2*d^5-4*B*a*b*c*d^4+B*b^2*c^4*d+3*B*b^2*c^2*d^3-2*C*a^
2*c*d^4+2*C*a*b*c^4*d+6*C*a*b*c^2*d^3-2*C*b^2*c^5-4*C*b^2*c^3*d^2)/(c^2+d^2
)^2*ln(c+d*tan(f*x+e)))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 939 vs. $2(417) = 834$.

Time = 0.56 (sec) , antiderivative size = 939, normalized size of antiderivative = 2.25

$$\int \frac{(a+b\tan(e+fx))^2 (A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^2}\,dx =$$
$$-\frac{2\,Cb^2c^4d^2+2\,Aa^2d^6-2\,(2\,Cab+Bb^2)c^3d^3+2\,(Ca^2+2\,Bab+Ab^2)c^2d^4-2\,(Ba^2+2\,Aab)cd^5-2\,(((}{}$$

```
[In] integrate((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^2,x, algorithm="fricas")
```

```
[Out] -1/2*(2*C*b^2*c^4*d^2 + 2*A*a^2*d^6 - 2*(2*C*a*b + B*b^2)*c^3*d^3 + 2*(C*a^
2 + 2*B*a*b + A*b^2)*c^2*d^4 - 2*(B*a^2 + 2*A*a*b)*c*d^5 - 2*(((A - C)*a^2
- 2*B*a*b - (A - C)*b^2)*c^3*d^3 + 2*(B*a^2 + 2*(A - C)*a*b - B*b^2)*c^2*d^
4 - ((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*c*d^5)*f*x - 2*(C*b^2*c^4*d^2 + 2
*C*b^2*c^2*d^4 + C*b^2*d^6)*tan(f*x + e)^2 + (2*C*b^2*c^6 + 4*C*b^2*c^4*d^2
 - (2*C*a*b + B*b^2)*c^5*d + (B*a^2 + 2*(A - 3*C)*a*b - 3*B*b^2)*c^3*d^3 -
2*((A - C)*a^2 - 2*B*a*b - A*b^2)*c^2*d^4 - (B*a^2 + 2*A*a*b)*c*d^5 + (2*C*
b^2*c^5*d + 4*C*b^2*c^3*d^3 - (2*C*a*b + B*b^2)*c^4*d^2 + (B*a^2 + 2*(A - 3
*C)*a*b - 3*B*b^2)*c^2*d^4 - 2*((A - C)*a^2 - 2*B*a*b - A*b^2)*c*d^5 - (B*a
^2 + 2*A*a*b)*d^6)*tan(f*x + e))*log((d^2*tan(f*x + e)^2 + 2*c*d*tan(f*x +
e) + c^2)/(tan(f*x + e)^2 + 1)) - (2*C*b^2*c^6 + 4*C*b^2*c^4*d^2 + 2*C*b^2*
c^2*d^4 - (2*C*a*b + B*b^2)*c^5*d - 2*(2*C*a*b + B*b^2)*c^3*d^3 - (2*C*a*b
+ B*b^2)*c*d^5 + (2*C*b^2*c^5*d + 4*C*b^2*c^3*d^3 + 2*C*b^2*c*d^5 - (2*C*a*
b + B*b^2)*c^4*d^2 - 2*(2*C*a*b + B*b^2)*c^2*d^4 - (2*C*a*b + B*b^2)*d^6)*t
an(f*x + e))*log(1/(tan(f*x + e)^2 + 1)) - 2*(2*C*b^2*c^5*d - (2*C*a*b + B*
b^2)*c^4*d^2 + (C*a^2 + 2*B*a*b + (A + 2*C)*b^2)*c^3*d^3 - (B*a^2 + 2*A*a*b
)*c^2*d^4 + (A*a^2 + C*b^2)*c*d^5 + (((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*
c^2*d^4 + 2*(B*a^2 + 2*(A - C)*a*b - B*b^2)*c*d^5 - ((A - C)*a^2 - 2*B*a*b
- (A - C)*b^2)*d^6)*f*x)*tan(f*x + e))/((c^4*d^4 + 2*c^2*d^6 + d^8)*f*tan(f
*x + e) + (c^5*d^3 + 2*c^3*d^5 + c*d^7)*f)
```

Sympy [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 3.01 (sec) , antiderivative size = 16225, normalized size of antiderivative = 38.91

$$\int \frac{(a+b\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^2}\,dx = \text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))**2*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x
+e))**2,x)
```

```
[Out] Piecewise((zoo*x*(a + b*tan(e))**2*(A + B*tan(e) + C*tan(e)**2)/tan(e)**2,
Eq(c, 0) & Eq(d, 0) & Eq(f, 0)), ((A*a**2*x + A*a*b*log(tan(e + f*x)**2 + 1
)/f - A*b**2*x + A*b**2*tan(e + f*x)/f + B*a**2*log(tan(e + f*x)**2 + 1)/(2
*f) - 2*B*a*b*x + 2*B*a*b*tan(e + f*x)/f - B*b**2*log(tan(e + f*x)**2 + 1)/
(2*f) + B*b**2*tan(e + f*x)**2/(2*f) - C*a**2*x + C*a**2*tan(e + f*x)/f - C
*a*b*log(tan(e + f*x)**2 + 1)/f + C*a*b*tan(e + f*x)**2/f + C*b**2*x + C*b*
*2*tan(e + f*x)**3/(3*f) - C*b**2*tan(e + f*x)/f)/c**2, Eq(d, 0)), (-A*a**2
*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) -
4*d**2*f) + 2*I*A*a**2*f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**
2*f*tan(e + f*x) - 4*d**2*f) + A*a**2*f*x/(4*d**2*f*tan(e + f*x)**2 - 8*I*d
**2*f*tan(e + f*x) - 4*d**2*f) - A*a**2*tan(e + f*x)/(4*d**2*f*tan(e + f*x)
**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 2*I*A*a**2/(4*d**2*f*tan(e + f*
x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 2*I*A*a*b*f*x*tan(e + f*x)**2
/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 4*A*a*b*
f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d*
*2*f) - 2*I*A*a*b*f*x/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) -
 4*d**2*f) + 2*I*A*a*b*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*
tan(e + f*x) - 4*d**2*f) + A*b**2*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x
)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 2*I*A*b**2*f*x*tan(e + f*x)/(4
*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - A*b**2*f*x/
(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 3*A*b**2*
tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f
) + 2*I*A*b**2/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2
*f) + I*B*a**2*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*t
an(e + f*x) - 4*d**2*f) + 2*B*a**2*f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)*
*2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - I*B*a**2*f*x/(4*d**2*f*tan(e + f
*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + I*B*a**2*tan(e + f*x)/(4*d**
2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 2*B*a*b*f*x*tan
(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f
) - 4*I*B*a*b*f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e
 + f*x) - 4*d**2*f) - 2*B*a*b*f*x/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*ta
n(e + f*x) - 4*d**2*f) - 6*B*a*b*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8
*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 4*I*B*a*b/(4*d**2*f*tan(e + f*x)**2 -
8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 3*I*B*b**2*f*x*tan(e + f*x)**2/(4*d**
```

```
2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 6*B*b**2*f*x*ta
n(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f)
 - 3*I*B*b**2*f*x/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d*
*2*f) + 2*B*b**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)**2/(4*d**2*f*tan(e +
 f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 4*I*B*b**2*log(tan(e + f*x
)**2 + 1)*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x)
 - 4*d**2*f) - 2*B*b**2*log(tan(e + f*x)**2 + 1)/(4*d**2*f*tan(e + f*x)**2 -
 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 5*I*B*b**2*tan(e + f*x)/(4*d**2*f*ta
n(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 4*B*b**2/(4*d**2*f*ta
n(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + C*a**2*f*x*tan(e + f*
x)**2/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 2*I
*C*a**2*f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x
) - 4*d**2*f) - C*a**2*f*x/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f
*x) - 4*d**2*f) - 3*C*a**2*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**
2*f*tan(e + f*x) - 4*d**2*f) + 2*I*C*a**2/(4*d**2*f*tan(e + f*x)**2 - 8*I*d
**2*f*tan(e + f*x) - 4*d**2*f) + 6*I*C*a*b*f*x*tan(e + f*x)**2/(4*d**2*f*ta
n(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 12*C*a*b*f*x*tan(e +
f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 6*I*
C*a*b*f*x/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) +
 4*C*a*b*log(tan(e + f*x)**2 + 1)*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2
 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 8*I*C*a*b*log(tan(e + f*x)**2 + 1)
*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*
f) - 4*C*a*b*log(tan(e + f*x)**2 + 1)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*
f*tan(e + f*x) - 4*d**2*f) - 10*I*C*a*b*tan(e + f*x)/(4*d**2*f*tan(e + f*x)
**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 8*C*a*b/(4*d**2*f*tan(e + f*x)*
*2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 9*C*b**2*f*x*tan(e + f*x)**2/(4*
d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 18*I*C*b**2*
f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d*
*2*f) + 9*C*b**2*f*x/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) -
4*d**2*f) + 4*I*C*b**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)**2/(4*d**2*f*t
an(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 8*C*b**2*log(tan(e +
 f*x)**2 + 1)*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f
*x) - 4*d**2*f) - 4*I*C*b**2*log(tan(e + f*x)**2 + 1)/(4*d**2*f*tan(e + f*x
)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 4*C*b**2*tan(e + f*x)**3/(4*d*
*2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 19*C*b**2*tan(
e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) -
14*I*C*b**2/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f)
, Eq(c, -I*d)), (-A*a**2*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 + 8*
I*d**2*f*tan(e + f*x) - 4*d**2*f) - 2*I*A*a**2*f*x*tan(e + f*x)/(4*d**2*f*t
an(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + A*a**2*f*x/(4*d**2*f
*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - A*a**2*tan(e + f*x
)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 2*I*A*a
**2/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 2*I*A
*a*b*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x
) - 4*d**2*f) + 4*A*a*b*f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d*
```

```
*2*f*tan(e + f*x) - 4*d**2*f) + 2*I*A*a*b*f*x/(4*d**2*f*tan(e + f*x)**2 + 8
*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 2*I*A*a*b*tan(e + f*x)/(4*d**2*f*tan(e
 + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + A*b**2*f*x*tan(e + f*x)*
*2/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 2*I*A*
b**2*f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) -
 4*d**2*f) - A*b**2*f*x/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x)
 - 4*d**2*f) - 3*A*b**2*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f
*tan(e + f*x) - 4*d**2*f) - 2*I*A*b**2/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2
*f*tan(e + f*x) - 4*d**2*f) - I*B*a**2*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e
+ f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 2*B*a**2*f*x*tan(e + f*x)
/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + I*B*a**2
*f*x/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - I*B*
a**2*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d
**2*f) + 2*B*a*b*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f
*tan(e + f*x) - 4*d**2*f) + 4*I*B*a*b*f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*
x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 2*B*a*b*f*x/(4*d**2*f*tan(e +
 f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 6*B*a*b*tan(e + f*x)/(4*d*
*2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 4*I*B*a*b/(4*d
**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 3*I*B*b**2*f*
x*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d
**2*f) + 6*B*b**2*f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*t
an(e + f*x) - 4*d**2*f) + 3*I*B*b**2*f*x/(4*d**2*f*tan(e + f*x)**2 + 8*I*d*
*2*f*tan(e + f*x) - 4*d**2*f) + 2*B*b**2*log(tan(e + f*x)**2 + 1)*tan(e + f
*x)**2/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 4*
I*B*b**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 +
8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 2*B*b**2*log(tan(e + f*x)**2 + 1)/(4*
d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 5*I*B*b**2*t
an(e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f)
 - 4*B*b**2/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f)
 + C*a**2*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e
+ f*x) - 4*d**2*f) + 2*I*C*a**2*f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2
+ 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - C*a**2*f*x/(4*d**2*f*tan(e + f*x)**
2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 3*C*a**2*tan(e + f*x)/(4*d**2*f*t
an(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 2*I*C*a**2/(4*d**2*f
*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 6*I*C*a*b*f*x*tan(
e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f)
 + 12*C*a*b*f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e +
 f*x) - 4*d**2*f) + 6*I*C*a*b*f*x/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*ta
n(e + f*x) - 4*d**2*f) + 4*C*a*b*log(tan(e + f*x)**2 + 1)*tan(e + f*x)**2/(
4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 8*I*C*a*b*
log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*
f*tan(e + f*x) - 4*d**2*f) - 4*C*a*b*log(tan(e + f*x)**2 + 1)/(4*d**2*f*tan
(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 10*I*C*a*b*tan(e + f*x
)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 8*C*a*b
/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 9*C*b**2
```

*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 18*I*C*b**2*f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d* *2*f*tan(e + f*x) - 4*d**2*f) + 9*C*b**2*f*x/(4*d**2*f*tan(e + f*x)**2 + 8* I*d**2*f*tan(e + f*x) - 4*d**2*f) - 4*I*C*b**2*log(tan(e + f*x)**2 + 1)*tan (e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 8*C*b**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(4*d**2*f*tan(e + f*x)** 2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 4*I*C*b**2*log(tan(e + f*x)**2 + 1)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 4*C*b* *2*tan(e + f*x)**3/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4* d**2*f) + 19*C*b**2*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan (e + f*x) - 4*d**2*f) + 14*I*C*b**2/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f* tan(e + f*x) - 4*d**2*f), Eq(c, I*d)), (x*(a + b*tan(e))**2*(A + B*tan(e) + C*tan(e)**2)/(c + d*tan(e))**2, Eq(f, 0)), (2*A*a**2*c**3*d**3*f*x/(2*c**5 *d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) + 2*A*a**2*c**2*d**4*f*x*tan(e + f*x)/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c** 2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) + 4*A*a**2*c**2 *d**4*log(c/d + tan(e + f*x))/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) - 2*A*a**2*c**2*d**4*log(tan(e + f*x)**2 + 1)/(2*c**5*d**3*f + 2*c** 4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d* *7*f + 2*d**8*f*tan(e + f*x)) - 2*A*a**2*c**2*d**4/(2*c**5*d**3*f + 2*c**4* d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7 *f + 2*d**8*f*tan(e + f*x)) - 2*A*a**2*c*d**5*f*x/(2*c**5*d**3*f + 2*c**4*d **4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7* f + 2*d**8*f*tan(e + f*x)) + 4*A*a**2*c*d**5*log(c/d + tan(e + f*x))*tan(e + f*x)/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2 *d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) - 2*A*a**2*c*d** 5*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8 *f*tan(e + f*x)) - 2*A*a**2*d**6*f*x*tan(e + f*x)/(2*c**5*d**3*f + 2*c**4*d **4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7* f + 2*d**8*f*tan(e + f*x)) - 2*A*a**2*d**6/(2*c**5*d**3*f + 2*c**4*d**4*f*t an(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d **8*f*tan(e + f*x)) - 4*A*a*b*c**3*d**3*log(c/d + tan(e + f*x))/(2*c**5*d** 3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f* x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) + 2*A*a*b*c**3*d**3*log(tan(e + f* x)**2 + 1)/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4* c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) + 4*A*a*b*c* *3*d**3/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c** 2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) + 8*A*a*b*c**2* d**4*f*x/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c* *2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) - 4*A*a*b*c**2 *d**4*log(c/d + tan(e + f*x))*tan(e + f*x)/(2*c**5*d**3*f + 2*c**4*d**4*f*t an(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d

```
**8*f*tan(e + f*x)) + 2*A*a*b*c**2*d**4*log(tan(e + f*x)**2 + 1)*tan(e + f*
x)/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**
6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) + 8*A*a*b*c*d**5*f*x
*tan(e + f*x)/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f +
 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) + 4*A*a*b
*c*d**5*log(c/d + tan(e + f*x))/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x)
 + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e
 + f*x)) - 2*A*a*b*c*d**5*log(tan(e + f*x)**2 + 1)/(2*c**5*d**3*f + 2*c**4*
d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7
*f + 2*d**8*f*tan(e + f*x)) + 4*A*a*b*c*d**5/(2*c**5*d**3*f + 2*c**4*d**4*f
*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2
*d**8*f*tan(e + f*x)) + 4*A*a*b*d**6*log(c/d + tan(e + f*x))*tan(e + f*x)/(
2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*
tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) - 2*A*a*b*d**6*log(tan(e
 + f*x)**2 + 1)*tan(e + f*x)/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) +
4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e +
f*x)) - 2*A*b**2*c**4*d**2/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*
c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*
x)) - 2*A*b**2*c**3*d**3*f*x/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) +
4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e +
f*x)) - 2*A*b**2*c**2*d**4*f*x*tan(e + f*x)/(2*c**5*d**3*f + 2*c**4*d**4*f*
tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*
d**8*f*tan(e + f*x)) - 4*A*b**2*c**2*d**4*log(c/d + tan(e + f*x))/(2*c**5*d
**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e +
f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) + 2*A*b**2*c**2*d**4*log(tan(e +
 f*x)**2 + 1)/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f +
 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) - 2*A*b**
2*c**2*d**4/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4
*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) + 2*A*b**2*
c*d**5*f*x/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*
c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) - 4*A*b**2*c
*d**5*log(c/d + tan(e + f*x))*tan(e + f*x)/(2*c**5*d**3*f + 2*c**4*d**4*f*t
an(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d
**8*f*tan(e + f*x)) + 2*A*b**2*c*d**5*log(tan(e + f*x)**2 + 1)*tan(e + f*x)
/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*
f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) + 2*A*b**2*d**6*f*x*ta
n(e + f*x)/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*
c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) - 2*B*a**2*c
**3*d**3*log(c/d + tan(e + f*x))/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x
) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(
e + f*x)) + B*a**2*c**3*d**3*log(tan(e + f*x)**2 + 1)/(2*c**5*d**3*f + 2*c*
*4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d
**7*f + 2*d**8*f*tan(e + f*x)) + 2*B*a**2*c**3*d**3/(2*c**5*d**3*f + 2*c**4
*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**
7*f + 2*d**8*f*tan(e + f*x)) + 4*B*a**2*c**2*d**4*f*x/(2*c**5*d**3*f + 2*c*
```

*4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d
7*f + 2*d8*f*tan(e + f*x)) - 2*B*a**2*c**2*d**4*log(c/d + tan(e + f*x))
*tan(e + f*x)/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f +
4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) + B*a**2*
c**2*d**4*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*c**5*d**3*f + 2*c**4*d**
4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f
+ 2*d**8*f*tan(e + f*x)) + 4*B*a**2*c*d**5*f*x*tan(e + f*x)/(2*c**5*d**3*f
+ 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) +
2*c*d**7*f + 2*d**8*f*tan(e + f*x)) + 2*B*a**2*c*d**5*log(c/d + tan(e + f*
x))/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d*
*6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) - B*a**2*c*d**5*log
(tan(e + f*x)**2 + 1)/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*
d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) +
2*B*a**2*c*d**5/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*
f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) + 2*B*
a**2*d**6*log(c/d + tan(e + f*x))*tan(e + f*x)/(2*c**5*d**3*f + 2*c**4*d**4
*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f +
2*d**8*f*tan(e + f*x)) - B*a**2*d**6*log(tan(e + f*x)**2 + 1)*tan(e + f*x)
/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*
f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) - 4*B*a*b*c**4*d**2/(2
*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*t
an(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) - 4*B*a*b*c**3*d**3*f*x/(
2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*
tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) - 4*B*a*b*c**2*d**4*f*x*
tan(e + f*x)/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f +
4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) - 8*B*a*b*
c**2*d**4*log(c/d + tan(e + f*x))/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*
x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan
(e + f*x)) + 4*B*a*b*c**2*d**4*log(tan(e + f*x)**2 + 1)/(2*c**5*d**3*f + 2*
c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c
*d**7*f + 2*d**8*f*tan(e + f*x)) - 4*B*a*b*c**2*d**4/(2*c**5*d**3*f + 2*c**
4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d*
*7*f + 2*d**8*f*tan(e + f*x)) + 4*B*a*b*c*d**5*f*x/(2*c**5*d**3*f + 2*c**4*
d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7
*f + 2*d**8*f*tan(e + f*x)) - 8*B*a*b*c*d**5*log(c/d + tan(e + f*x))*tan(e
+ f*x)/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2
*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) + 4*B*a*b*c*d**5
*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e
+ f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*
f*tan(e + f*x)) + 4*B*a*b*d**6*f*x*tan(e + f*x)/(2*c**5*d**3*f + 2*c**4*d**
4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f
+ 2*d**8*f*tan(e + f*x)) + 2*B*b**2*c**5*d*log(c/d + tan(e + f*x))/(2*c**5*
d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e +
f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) + 2*B*b**2*c**5*d/(2*c**5*d**3*
f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x)

+ 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) + 2*B*b**2*c**4*d**2*log(c/d + tan(e + f*x))*tan(e + f*x)/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) + 6*B*b**2*c**3*d**3*log(c/d + tan(e + f*x))/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) - B*b**2*c**3*d**3*log(tan(e + f*x)**2 + 1)/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) + 2*B*b**2*c**3*d**3/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) - 4*B*b**2*c**2*d**4*f*x/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) + 6*B*b**2*c**2*d**4*log(c/d + tan(e + f*x))*tan(e + f*x)/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) - B*b**2*c**2*d**4*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) - 4*B*b**2*c*d**5*f*x*tan(e + f*x)/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) + B*b**2*c*d**5*log(tan(e + f*x)**2 + 1)/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) + B*b**2*d**6*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) - 2*C*a**2*c**4*d**2/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) - 2*C*a**2*c**3*d**3*f*x/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) - 2*C*a**2*c**2*d**4*f*x*tan(e + f*x)/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) - 4*C*a**2*c**2*d**4*log(c/d + tan(e + f*x))/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) + 2*C*a**2*c**2*d**4*log(tan(e + f*x)**2 + 1)/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) - 2*C*a**2*c**2*d**4/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) + 2*C*a**2*c*d**5*f*x/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) - 4*C*a**2*c*d**5*log(c/d + tan(e + f*x))*tan(e + f*x)/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) + 2*C*a**2*c*d**5*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) + 2*C*a**2*d**6*f*x*tan(e + f*x)/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)

) + 4*C*a*b*c**5*d*log(c/d + tan(e + f*x))/(2*c**5*d**3*f + 2*c**4*d**4*f*t
an(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d
8*f*tan(e + f*x)) + 4*C*a*b*c5*d/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e +
 f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*
tan(e + f*x)) + 4*C*a*b*c**4*d**2*log(c/d + tan(e + f*x))*tan(e + f*x)/(2*c
5*d3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan
(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) + 12*C*a*b*c**3*d**3*log(c/
d + tan(e + f*x))/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5
*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) - 2*C
*a*b*c**3*d**3*log(tan(e + f*x)**2 + 1)/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(
e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8
*f*tan(e + f*x)) + 4*C*a*b*c**3*d**3/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e +
 f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*
tan(e + f*x)) - 8*C*a*b*c**2*d**4*f*x/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e
+ f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f
*tan(e + f*x)) + 12*C*a*b*c**2*d**4*log(c/d + tan(e + f*x))*tan(e + f*x)/(2
*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*t
an(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) - 2*C*a*b*c**2*d**4*log(t
an(e + f*x)**2 + 1)*tan(e + f*x)/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x
) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(
e + f*x)) - 8*C*a*b*c*d**5*f*x*tan(e + f*x)/(2*c**5*d**3*f + 2*c**4*d**4*f*
tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*
d**8*f*tan(e + f*x)) + 2*C*a*b*c*d**5*log(tan(e + f*x)**2 + 1)/(2*c**5*d**3
*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x
) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) + 2*C*a*b*d**6*log(tan(e + f*x)**2
+ 1)*tan(e + f*x)/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5
*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) - 4*C
*b**2*c**6*log(c/d + tan(e + f*x))/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f
*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*ta
n(e + f*x)) - 4*C*b**2*c**6/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4
*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f
*x)) - 4*C*b**2*c**5*d*log(c/d + tan(e + f*x))*tan(e + f*x)/(2*c**5*d**3*f
+ 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) +
 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) - 8*C*b**2*c**4*d**2*log(c/d + tan(e +
 f*x))/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2
*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) + 2*C*b**2*c**4*
d**2*tan(e + f*x)**2/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d
5*f + 4*c2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) -
6*C*b**2*c**4*d**2/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**
5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) + 2*
C*b**2*c**3*d**3*f*x/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d
5*f + 4*c2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) -
8*C*b**2*c**3*d**3*log(c/d + tan(e + f*x))*tan(e + f*x)/(2*c**5*d**3*f + 2*
c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c
*d**7*f + 2*d**8*f*tan(e + f*x)) + 2*C*b**2*c**2*d**4*f*x*tan(e + f*x)/(2*c

```
**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan
(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) - 2*C*b**2*c**2*d**4*log(ta
n(e + f*x)**2 + 1)/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**
5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) + 4*
C*b**2*c**2*d**4*tan(e + f*x)**2/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x
) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(
e + f*x)) - 2*C*b**2*c**2*d**4/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x)
+ 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e
+ f*x)) - 2*C*b**2*c*d**5*f*x/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) +
 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e +
 f*x)) - 2*C*b**2*c*d**5*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*c**5*d**3
*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*tan(e + f*x
) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) - 2*C*b**2*d**6*f*x*tan(e + f*x)/(2
*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2*d**6*f*t
an(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)) + 2*C*b**2*d**6*tan(e + f
*x)**2/(2*c**5*d**3*f + 2*c**4*d**4*f*tan(e + f*x) + 4*c**3*d**5*f + 4*c**2
*d**6*f*tan(e + f*x) + 2*c*d**7*f + 2*d**8*f*tan(e + f*x)), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.34 (sec) , antiderivative size = 493, normalized size of antiderivative = 1.18

$$\int \frac{(a+b\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^2}\,dx$$

$$= \frac{\frac{2\,Cb^2\tan(fx+e)}{d^2} + \frac{2\,(((A-C)a^2-2\,Bab-(A-C)b^2)c^2+2\,(Ba^2+2\,(A-C)ab-Bb^2)cd-((A-C)a^2-2\,Bab-(A-C)b^2)d^2)(fx+e)}{c^4+2\,c^2d^2+d^4} - \frac{2\,(2\,Cb^2c^5}{}}{}$$

```
[In] integrate((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^2,x, algorithm="maxima")
```

```
[Out] 1/2*(2*C*b^2*tan(f*x + e)/d^2 + 2*(((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*c^
2 + 2*(B*a^2 + 2*(A - C)*a*b - B*b^2)*c*d - ((A - C)*a^2 - 2*B*a*b - (A - C
)*b^2)*d^2)*(f*x + e)/(c^4 + 2*c^2*d^2 + d^4) - 2*(2*C*b^2*c^5 + 4*C*b^2*c^
3*d^2 - (2*C*a*b + B*b^2)*c^4*d + (B*a^2 + 2*(A - 3*C)*a*b - 3*B*b^2)*c^2*d
^3 - 2*((A - C)*a^2 - 2*B*a*b - A*b^2)*c*d^4 - (B*a^2 + 2*A*a*b)*d^5)*log(d
*tan(f*x + e) + c)/(c^4*d^3 + 2*c^2*d^5 + d^7) + ((B*a^2 + 2*(A - C)*a*b -
B*b^2)*c^2 - 2*((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*c*d - (B*a^2 + 2*(A -
C)*a*b - B*b^2)*d^2)*log(tan(f*x + e)^2 + 1)/(c^4 + 2*c^2*d^2 + d^4) - 2*(C
*b^2*c^4 + A*a^2*d^4 - (2*C*a*b + B*b^2)*c^3*d + (C*a^2 + 2*B*a*b + A*b^2)*
c^2*d^2 - (B*a^2 + 2*A*a*b)*c*d^3)/(c^3*d^3 + c*d^5 + (c^2*d^4 + d^6)*tan(f
*x + e)))/f
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 893 vs. $2(417) = 834$.

Time = 0.78 (sec) , antiderivative size = 893, normalized size of antiderivative = 2.14

$$\int \frac{(a+b\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^2}\,dx$$
$$= \frac{\frac{2\,Cb^2\tan(fx+e)}{d^2} + \frac{2\,(Aa^2c^2-Ca^2c^2-2\,Babc^2-Ab^2c^2+Cb^2c^2+2\,Ba^2cd+4\,Aabcd-4\,Cabcd-2\,Bb^2cd-Aa^2d^2+Ca^2d^2+2\,Babd^2+Ab^2d^2-Cb^2}{c^4+2\,c^2d^2+d^4}}{}$$

```
[In] integrate((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^2,x, algorithm="giac")
```

```
[Out] 1/2*(2*C*b^2*tan(f*x + e)/d^2 + 2*(A*a^2*c^2 - C*a^2*c^2 - 2*B*a*b*c^2 - A*
b^2*c^2 + C*b^2*c^2 + 2*B*a^2*c*d + 4*A*a*b*c*d - 4*C*a*b*c*d - 2*B*b^2*c*d
 - A*a^2*d^2 + C*a^2*d^2 + 2*B*a*b*d^2 + A*b^2*d^2 - C*b^2*d^2)*(f*x + e)/(
c^4 + 2*c^2*d^2 + d^4) + (B*a^2*c^2 + 2*A*a*b*c^2 - 2*C*a*b*c^2 - B*b^2*c^2
 - 2*A*a^2*c*d + 2*C*a^2*c*d + 4*B*a*b*c*d + 2*A*b^2*c*d - 2*C*b^2*c*d - B*
a^2*d^2 - 2*A*a*b*d^2 + 2*C*a*b*d^2 + B*b^2*d^2)*log(tan(f*x + e)^2 + 1)/(c
^4 + 2*c^2*d^2 + d^4) - 2*(2*C*b^2*c^5 - 2*C*a*b*c^4*d - B*b^2*c^4*d + 4*C*
b^2*c^3*d^2 + B*a^2*c^2*d^3 + 2*A*a*b*c^2*d^3 - 6*C*a*b*c^2*d^3 - 3*B*b^2*c
^2*d^3 - 2*A*a^2*c*d^4 + 2*C*a^2*c*d^4 + 4*B*a*b*c*d^4 + 2*A*b^2*c*d^4 - B*
a^2*d^5 - 2*A*a*b*d^5)*log(abs(d*tan(f*x + e) + c))/(c^4*d^3 + 2*c^2*d^5 +
d^7) + 2*(2*C*b^2*c^5*d*tan(f*x + e) - 2*C*a*b*c^4*d^2*tan(f*x + e) - B*b^2
*c^4*d^2*tan(f*x + e) + 4*C*b^2*c^3*d^3*tan(f*x + e) + B*a^2*c^2*d^4*tan(f*
x + e) + 2*A*a*b*c^2*d^4*tan(f*x + e) - 6*C*a*b*c^2*d^4*tan(f*x + e) - 3*B*
b^2*c^2*d^4*tan(f*x + e) - 2*A*a^2*c*d^5*tan(f*x + e) + 2*C*a^2*c*d^5*tan(f
*x + e) + 4*B*a*b*c*d^5*tan(f*x + e) + 2*A*b^2*c*d^5*tan(f*x + e) - B*a^2*d
^6*tan(f*x + e) - 2*A*a*b*d^6*tan(f*x + e) + C*b^2*c^6 - C*a^2*c^4*d^2 - 2*
B*a*b*c^4*d^2 - A*b^2*c^4*d^2 + 3*C*b^2*c^4*d^2 + 2*B*a^2*c^3*d^3 + 4*A*a*b
*c^3*d^3 - 4*C*a*b*c^3*d^3 - 2*B*b^2*c^3*d^3 - 3*A*a^2*c^2*d^4 + C*a^2*c^2*
d^4 + 2*B*a*b*c^2*d^4 + A*b^2*c^2*d^4 - A*a^2*d^6)/((c^4*d^3 + 2*c^2*d^5 +
d^7)*(d*tan(f*x + e) + c)))/f
```

Mupad [B] (verification not implemented)

Time = 33.60 (sec) , antiderivative size = 3958, normalized size of antiderivative = 9.49

$$\int \frac{(a+b\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^2}\,dx = \text{Too large to display}$$

```
[In] int(((a + b*tan(e + f*x))^2*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d
*tan(e + f*x))^2,x)
```

[Out] (log((2*C^2*b^4*c^5 - 2*C^2*a^2*b^2*c^5 + 4*C^2*b^4*c^3*d^2 - A*B*a^4*d^5 - 2*A*C*b^4*c^5 + B*C*a^4*d^5 + 2*A^2*a*b^3*d^5 - 2*A^2*a^3*b*d^5 - A^2*a^4*c*d^4 + 2*B^2*a^3*b*d^5 - A^2*b^4*c*d^4 + B^2*a^4*c*d^4 + B^2*b^4*c*d^4 - C^2*a^4*c*d^4 + C^2*b^4*c*d^4 - 4*C^2*a^2*b^2*c^3*d^2 + 5*A*B*a^2*b^2*d^5 + 2*A*C*a^2*b^2*c^5 + A*B*a^4*c^2*d^3 + 3*A*B*b^4*c^2*d^3 - B*C*a^2*b^2*d^5 - 4*A*C*b^4*c^3*d^2 - B*C*a^4*c^2*d^3 - 3*B*C*b^4*c^2*d^3 + 2*B^2*a*b^3*c^4*d - 2*C^2*a*b^3*c^4*d + 2*C^2*a^3*b*c^4*d - 2*A^2*a*b^3*c^2*d^3 + 6*A^2*a^2*b^2*c*d^4 + 2*A^2*a^3*b*c^2*d^3 + 6*B^2*a*b^3*c^2*d^3 - 6*B^2*a^2*b^2*c*d^4 - 2*B^2*a^3*b*c^2*d^3 - 6*C^2*a*b^3*c^2*d^3 + 4*C^2*a^2*b^2*c*d^4 + 6*C^2*a^3*b*c^2*d^3 - 2*A*C*a*b^3*d^5 + 2*A*C*a^3*b*d^5 - 4*B*C*a*b^3*c^5 + A*B*b^4*c^4*d + 2*A*C*a^4*c*d^4 - B*C*b^4*c^4*d - 8*A*B*a*b^3*c*d^4 + 8*A*B*a^3*b*c*d^4 + 2*A*C*a*b^3*c^4*d - 2*A*C*a^3*b*c^4*d + 4*B*C*a*b^3*c*d^4 - 8*B*C*a^3*b*c*d^4 - A*B*a^2*b^2*c^4*d + 8*A*C*a*b^3*c^2*d^3 - 10*A*C*a^2*b^2*c*d^4 - 8*A*C*a^3*b*c^2*d^3 - 8*B*C*a*b^3*c^3*d^2 + 5*B*C*a^2*b^2*c^4*d - 8*A*B*a^2*b^2*c^2*d^3 + 4*A*C*a^2*b^2*c^3*d^2 + 16*B*C*a^2*b^2*c^2*d^3)/(d^2*(c^2 + d^2)^2) + ((a*1i - b)^2*((A*b^2*d^2 - A*a^2*d^2 + C*a^2*d^2 - 8*C*b^2*c^2 - C*b^2*d^2 + 2*B*a*b*d^2 + 4*B*b^2*c*d + 8*C*a*b*c*d)/d - (tan(e + f*x)*(3*B*a^2*d^5 - 5*B*b^2*d^5 - 4*C*b^2*c^5 + 6*A*a*b*d^5 - 10*C*a*b*d^5 + 4*A*a^2*c*d^4 - 4*A*b^2*c*d^4 + 2*B*b^2*c^4*d - 4*C*a^2*c*d^4 + 8*C*b^2*c*d^4 - B*a^2*c^2*d^3 + B*b^2*c^2*d^3 - 8*B*a*b*c*d^4 + 4*C*a*b*c^4*d - 2*A*a*b*c^2*d^3 + 2*C*a*b*c^2*d^3))/(d^2*(c^2 + d^2)) + (d*(a*1i - b)^2*(4*c*d - c^2*tan(e + f*x) + 3*d^2*tan(e + f*x))*(A + B*1i - C)*1i)/(c*1i - d)^2)*(A + B*1i - C)*1i)/(2*(c*1i - d)^2) + (tan(e + f*x)*(A^2*a^4*d^5 + A^2*b^4*d^5 + B^2*b^4*d^5 + C^2*a^4*d^5 + C^2*b^4*d^5 - 2*A^2*a^2*b^2*d^5 + 3*B^2*a^2*b^2*d^5 + B^2*a^4*c^2*d^3 + 2*C^2*a^2*b^2*d^5 + 3*B^2*b^4*c^2*d^3 - 2*A*C*a^4*d^5 - 2*A*C*b^4*d^5 - 2*B*C*b^4*c^5 - 4*C^2*a*b^3*c^5 + B^2*b^4*c^4*d + 4*A^2*a^2*b^2*c^2*d^3 - 4*B^2*a^2*b^2*c^2*d^3 + 12*C^2*a^2*b^2*c^2*d^3 + 2*B*C*a^2*b^2*c^5 - 4*B*C*b^4*c^3*d^2 + 4*A^2*a*b^3*c*d^4 - 4*A^2*a^3*b*c*d^4 - 4*B^2*a*b^3*c*d^4 + 4*B^2*a^3*b*c*d^4 - 4*C^2*a^3*b*c*d^4 - B^2*a^2*b^2*c^4*d - 8*C^2*a*b^3*c^3*d^2 + 4*C^2*a^2*b^2*c^4*d + 2*A*B*a*b^3*d^5 - 4*A*B*a^3*b*d^5 + 4*A*C*a*b^3*c^5 - 2*A*B*a^4*c*d^4 - 2*A*B*b^4*c*d^4 + 2*B*C*a^3*b*d^5 + 2*B*C*a^4*c*d^4 - 2*A*B*a*b^3*c^4*d - 4*A*C*a*b^3*c*d^4 + 8*A*C*a^3*b*c*d^4 + 4*B*C*a*b^3*c^4*d - 2*B*C*a^3*b*c^4*d - 8*A*B*a*b^3*c^2*d^3 + 12*A*B*a^2*b^2*c*d^4 + 4*A*B*a^3*b*c^2*d^3 + 8*A*C*a*b^3*c^3*d^2 - 4*A*C*a^2*b^2*c^4*d + 12*B*C*a*b^3*c^2*d^3 - 10*B*C*a^2*b^2*c*d^4 - 8*B*C*a^3*b*c^2*d^3 - 16*A*C*a^2*b^2*c^2*d^3 + 4*B*C*a^2*b^2*c^3*d^2))/(d^2*(c^2 + d^2)^2))*(A*b^2 - A*a^2 - B*a^2*1i + B*b^2*1i + C*a^2 - C*b^2 - A*a*b*2i + 2*B*a*b + C*a*b*2i))/(2*f*(2*c*d - c^2*1i + d^2*1i)) + (log((2*C^2*b^4*c^5 - 2*C^2*a^2*b^2*c^5 + 4*C^2*b^4*c^3*d^2 - A*B*a^4*d^5 - 2*A*C*b^4*c^5 + B*C*a^4*d^5 + 2*A^2*a*b^3*d^5 - 2*A^2*a^3*b*d^5 - A^2*a^4*c*d^4 + 2*B^2*a^3*b*d^5 - A^2*b^4*c*d^4 + B^2*a^4*c*d^4 + B^2*b^4*c*d^4 - C^2*a^4*c*d^4 + C^2*b^4*c*d^4 - 4*C^2*a^2*b^2*c^3*d^2 + 5*A*B*a^2*b^2*d^5 + 2*A*C*a^2*b^2*c^5 + A*B*a^4*c^2*d^3 + 3*A*B*b^4*c^2*d^3 - B*C*a^2*b^2*d^5 - 4*A*C*b^4*c^3*d^2 - B*C*a^4*c^2*d^3 - 3*B*C*b^4*c^2*d^3 + 2*B^2*a*b^3*c^4*d - 2*C^2*a*b^3*c^4*d + 2*C^2*a^3*b*c^4*d - 2*A^2*a*b^3*c^2*d^3 + 6*A^2*a^2*b^2*c*d^4 + 2*A^2*a^3*b*c^

2*d^3 + 6*B^2*a*b^3*c^2*d^3 - 6*B^2*a^2*b^2*c*d^4 - 2*B^2*a^3*b*c^2*d^3 - 6 *C^2*a*b^3*c^2*d^3 + 4*C^2*a^2*b^2*c*d^4 + 6*C^2*a^3*b*c^2*d^3 - 2*A*C*a*b^ 3*d^5 + 2*A*C*a^3*b*d^5 - 4*B*C*a*b^3*c^5 + A*B*b^4*c^4*d + 2*A*C*a^4*c*d^4 - B*C*b^4*c^4*d - 8*A*B*a*b^3*c*d^4 + 8*A*B*a^3*b*c*d^4 + 2*A*C*a*b^3*c^4* d - 2*A*C*a^3*b*c^4*d + 4*B*C*a*b^3*c*d^4 - 8*B*C*a^3*b*c*d^4 - A*B*a^2*b^2 *c^4*d + 8*A*C*a*b^3*c^2*d^3 - 10*A*C*a^2*b^2*c*d^4 - 8*A*C*a^3*b*c^2*d^3 - 8*B*C*a*b^3*c^3*d^2 + 5*B*C*a^2*b^2*c^4*d - 8*A*B*a^2*b^2*c^2*d^3 + 4*A*C* a^2*b^2*c^3*d^2 + 16*B*C*a^2*b^2*c^2*d^3)/(d^2*(c^2 + d^2)^2) + (tan(e + f* x)*(A^2*a^4*d^5 + A^2*b^4*d^5 + B^2*b^4*d^5 + C^2*a^4*d^5 + C^2*b^4*d^5 - 2 *A^2*a^2*b^2*d^5 + 3*B^2*a^2*b^2*d^5 + B^2*a^4*c^2*d^3 + 2*C^2*a^2*b^2*d^5 + 3*B^2*b^4*c^2*d^3 - 2*A*C*a^4*d^5 - 2*A*C*b^4*d^5 - 2*B*C*b^4*c^5 - 4*C^2 *a*b^3*c^5 + B^2*b^4*c^4*d + 4*A^2*a^2*b^2*c^2*d^3 - 4*B^2*a^2*b^2*c^2*d^3 + 12*C^2*a^2*b^2*c^2*d^3 + 2*B*C*a^2*b^2*c^5 - 4*B*C*b^4*c^3*d^2 + 4*A^2*a* b^3*c*d^4 - 4*A^2*a^3*b*c*d^4 - 4*B^2*a*b^3*c*d^4 + 4*B^2*a^3*b*c*d^4 - 4*C ^2*a^3*b*c*d^4 - B^2*a^2*b^2*c^4*d - 8*C^2*a*b^3*c^3*d^2 + 4*C^2*a^2*b^2*c^ 4*d + 2*A*B*a*b^3*d^5 - 4*A*B*a^3*b*d^5 + 4*A*C*a*b^3*c^5 - 2*A*B*a^4*c*d^4 - 2*A*B*b^4*c*d^4 + 2*B*C*a^3*b*d^5 + 2*B*C*a^4*c*d^4 - 2*A*B*a*b^3*c^4*d - 4*A*C*a*b^3*c*d^4 + 8*A*C*a^3*b*c*d^4 + 4*B*C*a*b^3*c^4*d - 2*B*C*a^3*b*c ^4*d - 8*A*B*a*b^3*c^2*d^3 + 12*A*B*a^2*b^2*c*d^4 + 4*A*B*a^3*b*c^2*d^3 + 8 *A*C*a*b^3*c^3*d^2 - 4*A*C*a^2*b^2*c^4*d + 12*B*C*a*b^3*c^2*d^3 - 10*B*C*a^ 2*b^2*c*d^4 - 8*B*C*a^3*b*c^2*d^3 - 16*A*C*a^2*b^2*c^2*d^3 + 4*B*C*a^2*b^2* c^3*d^2))/(d^2*(c^2 + d^2)^2) + ((a*1i + b)^2*((tan(e + f*x)*(3*B*a^2*d^5 - 5*B*b^2*d^5 - 4*C*b^2*c^5 + 6*A*a*b*d^5 - 10*C*a*b*d^5 + 4*A*a^2*c*d^4 - 4 *A*b^2*c*d^4 + 2*B*b^2*c^4*d - 4*C*a^2*c*d^4 + 8*C*b^2*c*d^4 - B*a^2*c^2*d^ 3 + B*b^2*c^2*d^3 - 8*B*a*b*c*d^4 + 4*C*a*b*c^4*d - 2*A*a*b*c^2*d^3 + 2*C*a *b*c^2*d^3))/(d^2*(c^2 + d^2)) - (A*b^2*d^2 - A*a^2*d^2 + C*a^2*d^2 - 8*C*b ^2*c^2 - C*b^2*d^2 + 2*B*a*b*d^2 + 4*B*b^2*c*d + 8*C*a*b*c*d)/d + (d*(a*1i + b)^2*(4*c*d - c^2*tan(e + f*x) + 3*d^2*tan(e + f*x))*(A*1i + B - C*1i))/(c*1i + d)^2)*(A*1i + B - C*1i))/(2*(c*1i + d)^2))*(A*b^2*1i - A*a^2*1i - B* a^2 + B*b^2 + C*a^2*1i - C*b^2*1i - 2*A*a*b + B*a*b*2i + 2*C*a*b))/(2*f*(c* d*2i - c^2 + d^2)) - (log(c + d*tan(e + f*x))*(d^3*(B*a^2*c^2 - 3*B*b^2*c^2 + 2*A*a*b*c^2 - 6*C*a*b*c^2) - d^5*(B*a^2 + 2*A*a*b) - d*(B*b^2*c^4 + 2*C* a*b*c^4) + d^4*(2*A*b^2*c - 2*A*a^2*c + 2*C*a^2*c + 4*B*a*b*c) + 2*C*b^2*c^ 5 + 4*C*b^2*c^3*d^2))/(f*(d^7 + 2*c^2*d^5 + c^4*d^3)) + (C*b^2*tan(e + f*x))/(d^2*f) - (A*a^2*d^4 + C*b^2*c^4 - B*a^2*c*d^3 - B*b^2*c^3*d + A*b^2*c^2* d^2 + C*a^2*c^2*d^2 - 2*A*a*b*c*d^3 - 2*C*a*b*c^3*d + 2*B*a*b*c^2*d^2)/(d*f *(c*d^2 + d^3*tan(e + f*x))*(c^2 + d^2))

3.79 $\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^2}\,dx$

Optimal result . . . 812
Rubi [A] (verified) . . . 813
Mathematica [C] (verified) . . . 815
Maple [A] (verified) . . . 816
Fricas [A] (verification not implemented) . . . 816
Sympy [C] (verification not implemented) . . . 817
Maxima [A] (verification not implemented) . . . 822
Giac [A] (verification not implemented) . . . 823
Mupad [B] (verification not implemented) . . . 823

Optimal result

Integrand size = 43, antiderivative size = 292

$$\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^2}\,dx$$
$$= -\frac{(a(c^2C-2Bcd-Cd^2-A(c^2-d^2))-b(2c(A-C)d-B(c^2-d^2)))\,x}{(c^2+d^2)^2}$$
$$-\frac{(a(Bc^2+2cCd-Bd^2)-b(c^2C-2Bcd-Cd^2)-A(2acd-b(c^2-d^2)))\log(\cos(e+fx))}{(c^2+d^2)^2 f}$$
$$+\frac{(b(c^4C-c^2(A-3C)d^2-2Bcd^3+Ad^4)+ad^2(2c(A-C)d-B(c^2-d^2)))\log(c+d\tan(e+fx))}{d^2(c^2+d^2)^2 f}$$
$$+\frac{(bc-ad)\left(c^2C-Bcd+Ad^2\right)}{d^2(c^2+d^2)\,f(c+d\tan(e+fx))}$$

```
[Out] -(a*(c^2*C-2*B*c*d-C*d^2-A*(c^2-d^2))-b*(2*c*(A-C)*d-B*(c^2-d^2)))*x/(c^2+d
^2)^2-(a*(B*c^2-B*d^2+2*C*c*d)-b*(-2*B*c*d+C*c^2-C*d^2)-A*(2*a*c*d-b*(c^2-d
^2)))*ln(cos(f*x+e))/(c^2+d^2)^2/f+(b*(c^4*C-c^2*(A-3*C)*d^2-2*B*c*d^3+A*d^
4)+a*d^2*(2*c*(A-C)*d-B*(c^2-d^2)))*ln(c+d*tan(f*x+e))/d^2/(c^2+d^2)^2/f+(-
a*d+b*c)*(A*d^2-B*c*d+C*c^2)/d^2/(c^2+d^2)/f/(c+d*tan(f*x+e))
```

Rubi [A] (verified)

Time = 0.55 (sec) , antiderivative size = 288, normalized size of antiderivative = 0.99, number of steps used = 5, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}} = 0.116$, Rules used = {3716, 3707, 3698, 31, 3556}

$$\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^2}\,dx$$
$$= \frac{(bc-ad)\left(Ad^2-Bcd+c^2C\right)}{d^2f\left(c^2+d^2\right)(c+d\tan(e+fx))} + \frac{\log(\cos(e+fx))\left(2aAcd-aB(c^2-d^2)-2acCd-Ab(c^2-d^2)+b(-2Bcd+c^2C-Cd^2)\right)}{f\left(c^2+d^2\right)^2} - \frac{x(a(-A(c^2-d^2)-2Bcd+c^2C-Cd^2)-b(2cd(A-C)-B(c^2-d^2)))}{\left(c^2+d^2\right)^2} + \frac{\left(ad^2(2cd(A-C)-B(c^2-d^2))+b(-c^2d^2(A-3C)+Ad^4-2Bcd^3+c^4C)\right)\log(c+d\tan(e+fx))}{d^2f\left(c^2+d^2\right)^2}$$

```
[In] Int[((a + b*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c + d*T
an[e + f*x])^2,x]
```

```
[Out] -(((a*(c^2*C - 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - b*(2*c*(A - C)*d - B*(c^2
 - d^2)))*x)/(c^2 + d^2)^2) + ((2*a*A*c*d - 2*a*c*C*d - A*b*(c^2 - d^2) - a
*B*(c^2 - d^2) + b*(c^2*C - 2*B*c*d - C*d^2))*Log[Cos[e + f*x]])/((c^2 + d^
2)^2*f) + ((b*(c^4*C - c^2*(A - 3*C)*d^2 - 2*B*c*d^3 + A*d^4) + a*d^2*(2*c*
(A - C)*d - B*(c^2 - d^2)))*Log[c + d*Tan[e + f*x]])/(d^2*(c^2 + d^2)^2*f)
+ ((b*c - a*d)*(c^2*C - B*c*d + A*d^2))/(d^2*(c^2 + d^2)*f*(c + d*Tan[e + f
*x]))
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3698

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3707

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[(a*A + b*B - a
*C)*(x/(a^2 + b^2)), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3716

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.
)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(-(b*c - a*d))*(c^2*C - B*c*d + A*d^2)*((c
+ d*Tan[e + f*x])^(n + 1)/(d^2*f*(n + 1)*(c^2 + d^2))), x] + Dist[1/(d*(c^2
 + d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)*Simp[a*d*(A*c - c*C + B*d) + b*(
c^2*C - B*c*d + A*d^2) + d*(A*b*c + a*B*c - b*c*C - a*A*d + b*B*d + a*C*d)*
Tan[e + f*x] + b*C*(c^2 + d^2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c
, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] && LtQ[n,
 -1]
```

Rubi steps

$$\text{integral} = \frac{(bc-ad)\left(c^2C-Bcd+Ad^2\right)}{d^2\left(c^2+d^2\right)f(c+d\tan(e+fx))} + \frac{\int \frac{ad(Ac-cC+Bd)+b\left(c^2C-Bcd+Ad^2\right)+d(Abc+aBc-bcC-aAd+bBd+aCd)\tan(e+fx)+bC\left(c^2+d^2\right)\tan^2(e+fx)}{c+d\tan(e+fx)}\,dx}{d\left(c^2+d^2\right)}$$

$$= -\frac{\left(a\left(c^2C-2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-b\left(2c(A-C)d-B\left(c^2-d^2\right)\right)\right)x}{\left(c^2+d^2\right)^2} + \frac{(bc-ad)\left(c^2C-Bcd+Ad^2\right)}{d^2\left(c^2+d^2\right)f(c+d\tan(e+fx))} - \frac{\left(2aAcd-2acCd-Ab\left(c^2-d^2\right)-aB\left(c^2-d^2\right)+b\left(c^2C-2Bcd-Cd^2\right)\right)\int \tan(e+fx)\,dx}{\left(c^2+d^2\right)^2} + \frac{\left(b\left(c^4C-c^2(A-3C)d^2-2Bcd^3+Ad^4\right)+ad^2\left(2c(A-C)d-B\left(c^2-d^2\right)\right)\right)\int \frac{1+\tan^2(e+fx)}{c+d\tan(e+fx)}\,dx}{d\left(c^2+d^2\right)^2}$$

$$= -\frac{(a(c^2C - 2Bcd - Cd^2 - A(c^2 - d^2)) - b(2c(A - C)d - B(c^2 - d^2)))\,x}{(c^2 + d^2)^2}$$
$$+ \frac{(2aAcd - 2acCd - Ab(c^2 - d^2) - aB(c^2 - d^2) + b(c^2C - 2Bcd - Cd^2))\log(\cos(e + fx))}{(c^2 + d^2)^2 f}$$
$$+ \frac{(bc - ad)\,(c^2C - Bcd + Ad^2)}{d^2\,(c^2 + d^2)\,f(c + d\tan(e + fx))}$$
$$+ \frac{(b(c^4C - c^2(A - 3C)d^2 - 2Bcd^3 + Ad^4) + ad^2(2c(A - C)d - B(c^2 - d^2)))\,\mathrm{Subst}\left(\int \frac{1}{c+x}\,dx, x, c$$
$$d^2\,(c^2 + d^2)^2 f$$

$$= -\frac{(a(c^2C - 2Bcd - Cd^2 - A(c^2 - d^2)) - b(2c(A - C)d - B(c^2 - d^2)))\,x}{(c^2 + d^2)^2}$$
$$+ \frac{(2aAcd - 2acCd - Ab(c^2 - d^2) - aB(c^2 - d^2) + b(c^2C - 2Bcd - Cd^2))\log(\cos(e + fx))}{(c^2 + d^2)^2 f}$$
$$+ \frac{(b(c^4C - c^2(A - 3C)d^2 - 2Bcd^3 + Ad^4) + ad^2(2c(A - C)d - B(c^2 - d^2)))\log(c + d\tan(e + f}{d^2\,(c^2 + d^2)^2 f}$$
$$+ \frac{(bc - ad)\,(c^2C - Bcd + Ad^2)}{d^2\,(c^2 + d^2)\,f(c + d\tan(e + fx))}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 2.48 (sec) , antiderivative size = 221, normalized size of antiderivative = 0.76

$$\int \frac{(a + b\tan(e + fx))\,(A + B\tan(e + fx) + C\tan^2(e + fx))}{(c + d\tan(e + fx))^2}\,dx$$
$$= \frac{\frac{(-ia+b)(A+iB-C)\log(i-\tan(e+fx))}{(c+id)^2} + \frac{(ia+b)(A-iB-C)\log(i+\tan(e+fx))}{(c-id)^2} + \frac{2\left(b(c^4C-c^2(A-3C)d^2-2Bcd^3+Ad^4)+ad^2(2c(A-C)d+B\right.}{d^2(c^2+d^2)^2}}{2f}$$

```
[In] Integrate[((a + b*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c
 + d*Tan[e + f*x])^2,x]
```

```
[Out] ((((-I)*a + b)*(A + I*B - C)*Log[I - Tan[e + f*x]])/(c + I*d)^2 + ((I*a + b
)*(A - I*B - C)*Log[I + Tan[e + f*x]])/(c - I*d)^2 + (2*(b*(c^4*C - c^2*(A
- 3*C)*d^2 - 2*B*c*d^3 + A*d^4) + a*d^2*(2*c*(A - C)*d + B*(-c^2 + d^2)))*L
og[c + d*Tan[e + f*x]])/(d^2*(c^2 + d^2)^2) + (2*(b*c - a*d)*(c^2*C - B*c*d
 + A*d^2))/(d^2*(c^2 + d^2)*(c + d*Tan[e + f*x])))/(2*f)
```

Maple [A] (verified)

Time = 0.24 (sec) , antiderivative size = 321, normalized size of antiderivative = 1.10

method	result
derivativedivides	$\frac{\left(-2Aacd+Abc^2-Abd^2+Bac^2-Bad^2+2Bbcd+2Cacd-Cbc^2+Cbd^2\right)\ln\left(1+\tan(fx+e)^2\right)}{2}+\left(Aac^2-Aad^2+2Abcd+2Bacd-Bbc^2+B\ldots \quad / \left(c^2+d^2\right)^2$
default	$\frac{\left(-2Aacd+Abc^2-Abd^2+Bac^2-Bad^2+2Bbcd+2Cacd-Cbc^2+Cbd^2\right)\ln\left(1+\tan(fx+e)^2\right)}{2}+\left(Aac^2-Aad^2+2Abcd+2Bacd-Bbc^2+B\ldots \quad / \left(c^2+d^2\right)^2$
norman	$\frac{c\left(Aac^2-Aad^2+2Abcd+2Bacd-Bbc^2+Bbd^2-Cac^2+Cad^2-2Cbcd\right)x}{c^4+2c^2d^2+d^4}+\frac{d\left(Aac^2-Aad^2+2Abcd+2Bacd-Bbc^2+Bbd^2-Cac^2+Ca\ldots\right.}{c^4+2c^2d^2+d^4} \quad / \left(c+d\tan(fx+e)\right)$
parallelrisch	Expression too large to display
risch	Expression too large to display

```
[In] int((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^2,x,m
ethod=_RETURNVERBOSE)
```

```
[Out] 1/f*(1/(c^2+d^2)^2*(1/2*(-2*A*a*c*d+A*b*c^2-A*b*d^2+B*a*c^2-B*a*d^2+2*B*b*c
*d+2*C*a*c*d-C*b*c^2+C*b*d^2)*ln(1+tan(f*x+e)^2)+(A*a*c^2-A*a*d^2+2*A*b*c*d
+2*B*a*c*d-B*b*c^2+B*b*d^2-C*a*c^2+C*a*d^2-2*C*b*c*d)*arctan(tan(f*x+e)))-(
A*a*d^3-A*b*c*d^2-B*a*c*d^2+B*b*c^2*d+C*a*c^2*d-C*b*c^3)/d^2/(c^2+d^2)/(c+d
*tan(f*x+e))+(2*A*a*c*d^3-A*b*c^2*d^2+A*b*d^4-B*a*c^2*d^2+B*a*d^4-2*B*b*c*d
^3-2*C*a*c*d^3+C*b*c^4+3*C*b*c^2*d^2)/(c^2+d^2)^2/d^2*ln(c+d*tan(f*x+e)))
```

Fricas [A] (verification not implemented)

none

Time = 0.34 (sec) , antiderivative size = 505, normalized size of antiderivative = 1.73

$$\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^2}\,dx$$
$$= \frac{2Cbc^3d^2-2Aad^5-2(Ca+Bb)c^2d^3+2(Ba+Ab)cd^4+2(((A-C)a-Bb)c^3d^2+2(Ba+(A-C)b)c^2\ldots}{\ldots}$$

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^2,x, algorithm="fricas")
```

```
[Out] 1/2*(2*C*b*c^3*d^2 - 2*A*a*d^5 - 2*(C*a + B*b)*c^2*d^3 + 2*(B*a + A*b)*c*d^
4 + 2*(((A - C)*a - B*b)*c^3*d^2 + 2*(B*a + (A - C)*b)*c^2*d^3 - ((A - C)*a
 - B*b)*c*d^4)*f*x + (C*b*c^5 - (B*a + (A - 3*C)*b)*c^3*d^2 + 2*((A - C)*a
- B*b)*c^2*d^3 + (B*a + A*b)*c*d^4 + (C*b*c^4*d - (B*a + (A - 3*C)*b)*c^2*d
```

```
^3 + 2*((A - C)*a - B*b)*c*d^4 + (B*a + A*b)*d^5)*tan(f*x + e))*log((d^2*ta
n(f*x + e)^2 + 2*c*d*tan(f*x + e) + c^2)/(tan(f*x + e)^2 + 1)) - (C*b*c^5 +
 2*C*b*c^3*d^2 + C*b*c*d^4 + (C*b*c^4*d + 2*C*b*c^2*d^3 + C*b*d^5)*tan(f*x
+ e))*log(1/(tan(f*x + e)^2 + 1)) - 2*(C*b*c^4*d - A*a*c*d^4 - (C*a + B*b)*
c^3*d^2 + (B*a + A*b)*c^2*d^3 - (((A - C)*a - B*b)*c^2*d^3 + 2*(B*a + (A -
C)*b)*c*d^4 - ((A - C)*a - B*b)*d^5)*f*x)*tan(f*x + e))/((c^4*d^3 + 2*c^2*d
^5 + d^7)*f*tan(f*x + e) + (c^5*d^2 + 2*c^3*d^4 + c*d^6)*f)
```

Sympy [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 1.62 (sec) , antiderivative size = 9721, normalized size of antiderivative = 33.29

$$\int \frac{(a+b\tan(e+fx))\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^2}\,dx = \text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x+e)
)**2,x)
```

```
[Out] Piecewise((zoo*x*(a + b*tan(e))*(A + B*tan(e) + C*tan(e)**2)/tan(e)**2, Eq(
c, 0) & Eq(d, 0) & Eq(f, 0)), ((A*a*x + A*b*log(tan(e + f*x)**2 + 1)/(2*f)
+ B*a*log(tan(e + f*x)**2 + 1)/(2*f) - B*b*x + B*b*tan(e + f*x)/f - C*a*x +
 C*a*tan(e + f*x)/f - C*b*log(tan(e + f*x)**2 + 1)/(2*f) + C*b*tan(e + f*x)
**2/(2*f))/c**2, Eq(d, 0)), (-A*a*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x
)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 2*I*A*a*f*x*tan(e + f*x)/(4*d*
*2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + A*a*f*x/(4*d**
2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - A*a*tan(e + f*x
)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 2*I*A*a
/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + I*A*b*f*
x*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d
**2*f) + 2*A*b*f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(
e + f*x) - 4*d**2*f) - I*A*b*f*x/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan
(e + f*x) - 4*d**2*f) + I*A*b*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*
d**2*f*tan(e + f*x) - 4*d**2*f) + I*B*a*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e
 + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 2*B*a*f*x*tan(e + f*x)/(
4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - I*B*a*f*x/
(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + I*B*a*tan
(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) +
 B*b*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x
) - 4*d**2*f) - 2*I*B*b*f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d*
*2*f*tan(e + f*x) - 4*d**2*f) - B*b*f*x/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**
2*f*tan(e + f*x) - 4*d**2*f) - 3*B*b*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2
 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 2*I*B*b/(4*d**2*f*tan(e + f*x)**2
- 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + C*a*f*x*tan(e + f*x)**2/(4*d**2*f*t
an(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 2*I*C*a*f*x*tan(e +
```

```
f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - C*a*
f*x/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 3*C*a
*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*
f) + 2*I*C*a/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f
) + 3*I*C*b*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(
e + f*x) - 4*d**2*f) + 6*C*b*f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8
*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 3*I*C*b*f*x/(4*d**2*f*tan(e + f*x)**2
- 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 2*C*b*log(tan(e + f*x)**2 + 1)*tan(
e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f)
 - 4*I*C*b*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2
- 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 2*C*b*log(tan(e + f*x)**2 + 1)/(4*d
**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 5*I*C*b*tan(e
 + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 4
*C*b/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f), Eq(c,
 -I*d)), (-A*a*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*t
an(e + f*x) - 4*d**2*f) - 2*I*A*a*f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**
2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + A*a*f*x/(4*d**2*f*tan(e + f*x)**2
 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - A*a*tan(e + f*x)/(4*d**2*f*tan(e +
 f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 2*I*A*a/(4*d**2*f*tan(e +
f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - I*A*b*f*x*tan(e + f*x)**2/(
4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 2*A*b*f*x*
tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f
) + I*A*b*f*x/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*
f) - I*A*b*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x)
 - 4*d**2*f) - I*B*a*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 + 8*I*d*
*2*f*tan(e + f*x) - 4*d**2*f) + 2*B*a*f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*
x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + I*B*a*f*x/(4*d**2*f*tan(e + f
*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - I*B*a*tan(e + f*x)/(4*d**2*f
*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + B*b*f*x*tan(e + f*
x)**2/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 2*I
*B*b*f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) -
 4*d**2*f) - B*b*f*x/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) -
4*d**2*f) - 3*B*b*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e
 + f*x) - 4*d**2*f) - 2*I*B*b/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e
+ f*x) - 4*d**2*f) + C*a*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 + 8*
I*d**2*f*tan(e + f*x) - 4*d**2*f) + 2*I*C*a*f*x*tan(e + f*x)/(4*d**2*f*tan(
e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - C*a*f*x/(4*d**2*f*tan(e
 + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 3*C*a*tan(e + f*x)/(4*d*
*2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 2*I*C*a/(4*d**
2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 3*I*C*b*f*x*tan
(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f
) + 6*C*b*f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f
*x) - 4*d**2*f) + 3*I*C*b*f*x/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e
+ f*x) - 4*d**2*f) + 2*C*b*log(tan(e + f*x)**2 + 1)*tan(e + f*x)**2/(4*d**2
*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 4*I*C*b*log(tan(
```

```
e + f*x)**2 + 1)*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e
+ f*x) - 4*d**2*f) - 2*C*b*log(tan(e + f*x)**2 + 1)/(4*d**2*f*tan(e + f*x)*
*2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 5*I*C*b*tan(e + f*x)/(4*d**2*f*t
an(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 4*C*b/(4*d**2*f*tan(
e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f), Eq(c, I*d)), (x*(a + b*t
an(e))*(A + B*tan(e) + C*tan(e)**2)/(c + d*tan(e))**2, Eq(f, 0)), (2*A*a*c*
*3*d**2*f*x/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4
*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) + 2*A*a*c**
2*d**3*f*x*tan(e + f*x)/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**
3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x))
 + 4*A*a*c**2*d**3*log(c/d + tan(e + f*x))/(2*c**5*d**2*f + 2*c**4*d**3*f*t
an(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d
**7*f*tan(e + f*x)) - 2*A*a*c**2*d**3*log(tan(e + f*x)**2 + 1)/(2*c**5*d**2
*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x
) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) - 2*A*a*c**2*d**3/(2*c**5*d**2*f +
2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2
*c*d**6*f + 2*d**7*f*tan(e + f*x)) - 2*A*a*c*d**4*f*x/(2*c**5*d**2*f + 2*c*
*4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d
**6*f + 2*d**7*f*tan(e + f*x)) + 4*A*a*c*d**4*log(c/d + tan(e + f*x))*tan(e
 + f*x)/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**
2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) - 2*A*a*c*d**4*
log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e
+ f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f
*tan(e + f*x)) - 2*A*a*d**5*f*x*tan(e + f*x)/(2*c**5*d**2*f + 2*c**4*d**3*f
*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2
*d**7*f*tan(e + f*x)) - 2*A*a*d**5/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f
*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*ta
n(e + f*x)) - 2*A*b*c**3*d**2*log(c/d + tan(e + f*x))/(2*c**5*d**2*f + 2*c*
*4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d
**6*f + 2*d**7*f*tan(e + f*x)) + A*b*c**3*d**2*log(tan(e + f*x)**2 + 1)/(2*
c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*ta
n(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) + 2*A*b*c**3*d**2/(2*c**5*
d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e +
 f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) + 4*A*b*c**2*d**3*f*x/(2*c**5*d
**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e +
f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) - 2*A*b*c**2*d**3*log(c/d + tan(
e + f*x))*tan(e + f*x)/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3
*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x))
+ A*b*c**2*d**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*c**5*d**2*f + 2*c*
*4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d
**6*f + 2*d**7*f*tan(e + f*x)) + 4*A*b*c*d**4*f*x*tan(e + f*x)/(2*c**5*d**2
*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x
) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) + 2*A*b*c*d**4*log(c/d + tan(e + f*
x))/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d*
*5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) - A*b*c*d**4*log(ta
```

n(e + f*x)**2 + 1)/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**
4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) + 2*
A*b*c*d**4/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*
c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) + 2*A*b*d**5
*log(c/d + tan(e + f*x))*tan(e + f*x)/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e
+ f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f
*tan(e + f*x)) - A*b*d**5*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*c**5*d**
2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*
x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) - 2*B*a*c**3*d**2*log(c/d + tan(e
+ f*x))/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**
2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) + B*a*c**3*d**2
*log(tan(e + f*x)**2 + 1)/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c
3*d4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x
)) + 2*B*a*c**3*d**2/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d
4*f + 4*c2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) +
4*B*a*c**2*d**3*f*x/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d*
*4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) - 2
*B*a*c**2*d**3*log(c/d + tan(e + f*x))*tan(e + f*x)/(2*c**5*d**2*f + 2*c**4
*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**
6*f + 2*d**7*f*tan(e + f*x)) + B*a*c**2*d**3*log(tan(e + f*x)**2 + 1)*tan(e
 + f*x)/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**
2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) + 4*B*a*c*d**4*
f*x*tan(e + f*x)/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*
f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) + 2*B*
a*c*d**4*log(c/d + tan(e + f*x))/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x
) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(
e + f*x)) - B*a*c*d**4*log(tan(e + f*x)**2 + 1)/(2*c**5*d**2*f + 2*c**4*d**
3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f
+ 2*d**7*f*tan(e + f*x)) + 2*B*a*c*d**4/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(
e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7
*f*tan(e + f*x)) + 2*B*a*d**5*log(c/d + tan(e + f*x))*tan(e + f*x)/(2*c**5*
d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e +
 f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) - B*a*d**5*log(tan(e + f*x)**2
+ 1)*tan(e + f*x)/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4
*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) - 2*B
*b*c**4*d/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c
2*d5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) - 2*B*b*c**3*
d**2*f*x/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c*
*2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) - 2*B*b*c**2*d
3*f*x*tan(e + f*x)/(2*c5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d
4*f + 4*c2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) -
4*B*b*c**2*d**3*log(c/d + tan(e + f*x))/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(
e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7
*f*tan(e + f*x)) + 2*B*b*c**2*d**3*log(tan(e + f*x)**2 + 1)/(2*c**5*d**2*f
+ 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) +

2*c*d**6*f + 2*d**7*f*tan(e + f*x)) - 2*B*b*c**2*d**3/(2*c**5*d**2*f + 2*c
4*d3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*
d**6*f + 2*d**7*f*tan(e + f*x)) + 2*B*b*c*d**4*f*x/(2*c**5*d**2*f + 2*c**4*
d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6
*f + 2*d**7*f*tan(e + f*x)) - 4*B*b*c*d**4*log(c/d + tan(e + f*x))*tan(e +
f*x)/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d
5*f*tan(e + f*x) + 2*c*d6*f + 2*d**7*f*tan(e + f*x)) + 2*B*b*c*d**4*log
(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f
*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*ta
n(e + f*x)) + 2*B*b*d**5*f*x*tan(e + f*x)/(2*c**5*d**2*f + 2*c**4*d**3*f*ta
n(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d*
*7*f*tan(e + f*x)) - 2*C*a*c**4*d/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*
x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan
(e + f*x)) - 2*C*a*c**3*d**2*f*x/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x
) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(
e + f*x)) - 2*C*a*c**2*d**3*f*x*tan(e + f*x)/(2*c**5*d**2*f + 2*c**4*d**3*f
*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2
*d**7*f*tan(e + f*x)) - 4*C*a*c**2*d**3*log(c/d + tan(e + f*x))/(2*c**5*d**
2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*
x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) + 2*C*a*c**2*d**3*log(tan(e + f*x)
2 + 1)/(2*c5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c*
*2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) - 2*C*a*c**2*d
3/(2*c5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d*
*5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) + 2*C*a*c*d**4*f*x/
(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f
*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) - 4*C*a*c*d**4*log(c/d
+ tan(e + f*x))*tan(e + f*x)/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) +
4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e +
f*x)) + 2*C*a*c*d**4*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*c**5*d**2*f +
2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) +
2*c*d**6*f + 2*d**7*f*tan(e + f*x)) + 2*C*a*d**5*f*x*tan(e + f*x)/(2*c**5*d
2*f + 2*c4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e +
f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) + 2*C*b*c**5*log(c/d + tan(e + f
*x))/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d
5*f*tan(e + f*x) + 2*c*d6*f + 2*d**7*f*tan(e + f*x)) + 2*C*b*c**5/(2*c*
*5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(
e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) + 2*C*b*c**4*d*log(c/d + tan
(e + f*x))*tan(e + f*x)/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**
3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x))
+ 6*C*b*c**3*d**2*log(c/d + tan(e + f*x))/(2*c**5*d**2*f + 2*c**4*d**3*f*t
an(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d
7*f*tan(e + f*x)) - C*b*c3*d**2*log(tan(e + f*x)**2 + 1)/(2*c**5*d**2*f
+ 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x)
+ 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) + 2*C*b*c**3*d**2/(2*c**5*d**2*f + 2*
c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c

```
*d**6*f + 2*d**7*f*tan(e + f*x)) - 4*C*b*c**2*d**3*f*x/(2*c**5*d**2*f + 2*c
**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*
d**6*f + 2*d**7*f*tan(e + f*x)) + 6*C*b*c**2*d**3*log(c/d + tan(e + f*x))*t
an(e + f*x)/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4
*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) - C*b*c**2*
d**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*c**5*d**2*f + 2*c**4*d**3*f*t
an(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d
**7*f*tan(e + f*x)) - 4*C*b*c*d**4*f*x*tan(e + f*x)/(2*c**5*d**2*f + 2*c**4
*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**
6*f + 2*d**7*f*tan(e + f*x)) + C*b*c*d**4*log(tan(e + f*x)**2 + 1)/(2*c**5*
d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e +
 f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) + C*b*d**5*log(tan(e + f*x)**2
+ 1)*tan(e + f*x)/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4
*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)), True
))
```

Maxima [A] (verification not implemented)

none

Time = 0.32 (sec) , antiderivative size = 319, normalized size of antiderivative = 1.09

$$\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^2}\,dx$$

$$= \frac{\frac{2\left(((A-C)a-Bb)c^2+2\left(Ba+(A-C)b\right)cd-\left((A-C)a-Bb\right)d^2\right)(fx+e)}{c^4+2c^2d^2+d^4} + \frac{2\left(Cbc^4-\left(Ba+(A-3C)b\right)c^2d^2+2\left((A-C)a-Bb\right)cd^3+(Ba+Ab)d^4\right)\log(d}{c^4d^2+2c^2d^4+d^6}}{}$$

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^2,x, algorithm="maxima")
```

```
[Out] 1/2*(2*(((A - C)*a - B*b)*c^2 + 2*(B*a + (A - C)*b)*c*d - ((A - C)*a - B*b)
*d^2)*(f*x + e)/(c^4 + 2*c^2*d^2 + d^4) + 2*(C*b*c^4 - (B*a + (A - 3*C)*b)*
c^2*d^2 + 2*((A - C)*a - B*b)*c*d^3 + (B*a + A*b)*d^4)*log(d*tan(f*x + e) +
 c)/(c^4*d^2 + 2*c^2*d^4 + d^6) + ((B*a + (A - C)*b)*c^2 - 2*((A - C)*a - B
*b)*c*d - (B*a + (A - C)*b)*d^2)*log(tan(f*x + e)^2 + 1)/(c^4 + 2*c^2*d^2 +
 d^4) + 2*(C*b*c^3 - A*a*d^3 - (C*a + B*b)*c^2*d + (B*a + A*b)*c*d^2)/(c^3*
d^2 + c*d^4 + (c^2*d^3 + d^5)*tan(f*x + e)))/f
```

Giac [A] (verification not implemented)

none

Time = 0.62 (sec) , antiderivative size = 515, normalized size of antiderivative = 1.76

$$\int \frac{(a+b\tan(e+fx))\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^2}\,dx$$
$$= \frac{\frac{2\,(Aac^2-Cac^2-Bbc^2+2\,Bacd+2\,Abcd-2\,Cbcd-Aad^2+Cad^2+Bbd^2)(fx+e)}{c^4+2\,c^2d^2+d^4} + \frac{(Bac^2+Abc^2-Cbc^2-2\,Aacd+2\,Cacd+2\,Bbcd-Bad^2-Abd^2+}{c^4+2\,c^2d^2+d^4}}{}$$

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^2,x, algorithm="giac")
```

```
[Out] 1/2*(2*(A*a*c^2 - C*a*c^2 - B*b*c^2 + 2*B*a*c*d + 2*A*b*c*d - 2*C*b*c*d - A
*a*d^2 + C*a*d^2 + B*b*d^2)*(f*x + e)/(c^4 + 2*c^2*d^2 + d^4) + (B*a*c^2 +
A*b*c^2 - C*b*c^2 - 2*A*a*c*d + 2*C*a*c*d + 2*B*b*c*d - B*a*d^2 - A*b*d^2 +
 C*b*d^2)*log(tan(f*x + e)^2 + 1)/(c^4 + 2*c^2*d^2 + d^4) + 2*(C*b*c^4 - B*
a*c^2*d^2 - A*b*c^2*d^2 + 3*C*b*c^2*d^2 + 2*A*a*c*d^3 - 2*C*a*c*d^3 - 2*B*b
*c*d^3 + B*a*d^4 + A*b*d^4)*log(abs(d*tan(f*x + e) + c))/(c^4*d^2 + 2*c^2*d
^4 + d^6) - 2*(C*b*c^4*tan(f*x + e) - B*a*c^2*d^2*tan(f*x + e) - A*b*c^2*d^
2*tan(f*x + e) + 3*C*b*c^2*d^2*tan(f*x + e) + 2*A*a*c*d^3*tan(f*x + e) - 2*
C*a*c*d^3*tan(f*x + e) - 2*B*b*c*d^3*tan(f*x + e) + B*a*d^4*tan(f*x + e) +
A*b*d^4*tan(f*x + e) + C*a*c^4 + B*b*c^4 - 2*B*a*c^3*d - 2*A*b*c^3*d + 2*C*
b*c^3*d + 3*A*a*c^2*d^2 - C*a*c^2*d^2 - B*b*c^2*d^2 + A*a*d^4)/((c^4*d + 2*
c^2*d^3 + d^5)*(d*tan(f*x + e) + c)))/f
```

Mupad [B] (verification not implemented)

Time = 21.23 (sec) , antiderivative size = 1875, normalized size of antiderivative = 6.42

$$\int \frac{(a+b\tan(e+fx))\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^2}\,dx = \text{Too large to display}$$

```
[In] int(((a + b*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d*t
an(e + f*x))^2,x)
```

```
[Out] (log(c + d*tan(e + f*x))*(d^4*(A*b + B*a) - d^3*(2*B*b*c - 2*A*a*c + 2*C*a*
c) - d^2*(A*b*c^2 + B*a*c^2 - 3*C*b*c^2) + C*b*c^4))/(f*(d^6 + 2*c^2*d^4 +
c^4*d^2)) - (log((A*B*b^2*d^4 - A*B*a^2*d^4 + B*C*a^2*d^4 + B*C*b^2*c^4 - A
^2*a*b*d^4 + B^2*a*b*d^4 + C^2*a*b*c^4 - A^2*a^2*c*d^3 + A^2*b^2*c*d^3 + B^
2*a^2*c*d^3 - B^2*b^2*c*d^3 - C^2*a^2*c*d^3 + C^2*b^2*c*d^3 + A*B*a^2*c^2*d
^2 - A*B*b^2*c^2*d^2 - B*C*a^2*c^2*d^2 + 3*B*C*b^2*c^2*d^2 + A^2*a*b*c^2*d^
2 - B^2*a*b*c^2*d^2 + 3*C^2*a*b*c^2*d^2 - A*C*a*b*c^4 + A*C*a*b*d^4 + 2*A*C
*a^2*c*d^3 - 2*A*C*b^2*c*d^3 - 4*A*C*a*b*c^2*d^2 + 4*A*B*a*b*c*d^3 - 4*B*C*
```

a*b*c*d^3)/(d*(c^2 + d^2)^2) + (tan(e + f*x)*(A^2*a^2*d^4 + B^2*b^2*d^4 + C^2*a^2*d^4 + C^2*b^2*c^4 + C^2*b^2*d^4 + A^2*b^2*c^2*d^2 + B^2*a^2*c^2*d^2 + 3*C^2*b^2*c^2*d^2 - 2*A*C*a^2*d^4 - A*C*b^2*c^4 - A*C*b^2*d^4 - 4*A*C*b^2*c^2*d^2 - 2*A*B*a*b*d^4 - B*C*a*b*c^4 + B*C*a*b*d^4 - 2*A*B*a^2*c*d^3 + 2*A*B*b^2*c*d^3 + 2*B*C*a^2*c*d^3 - 2*B*C*b^2*c*d^3 - 2*A^2*a*b*c*d^3 + 2*B^2*a*b*c*d^3 - 2*C^2*a*b*c*d^3 + 2*A*B*a*b*c^2*d^2 - 4*B*C*a*b*c^2*d^2 + 4*A*C*a*b*c*d^3))/(d*(c^2 + d^2)^2) + ((a*1i + b)*(B*1i - A + C)*(A*a*d - B*b*d - C*a*d - 4*C*b*c + (tan(e + f*x)*(3*A*b*d^4 + 3*B*a*d^4 + 2*C*b*c^4 - 5*C*b*d^4 + 4*A*a*c*d^3 - 4*B*b*c*d^3 - 4*C*a*c*d^3 - A*b*c^2*d^2 - B*a*c^2*d^2 + C*b*c^2*d^2))/(d*(c^2 + d^2)) + (d*(a*1i + b)*(4*c*d - c^2*tan(e + f*x) + 3*d^2*tan(e + f*x))*(B*1i - A + C))/(c*1i + d)^2))/(2*(c*1i + d)^2))*(A*a*1i + A*b + B*a - B*b*1i - C*a*1i - C*b))/(2*f*(c*d*2i - c^2 + d^2)) - (log((A*B*b^2*d^4 - A*B*a^2*d^4 + B*C*a^2*d^4 + B*C*b^2*c^4 - A^2*a*b*d^4 + B^2*a*b*d^4 + C^2*a*b*c^4 - A^2*a^2*c*d^3 + A^2*b^2*c*d^3 + B^2*a^2*c*d^3 - B^2*b^2*c*d^3 - C^2*a^2*c*d^3 + C^2*b^2*c*d^3 + A*B*a^2*c^2*d^2 - A*B*b^2*c^2*d^2 - B*C*a^2*c^2*d^2 + 3*B*C*b^2*c^2*d^2 + A^2*a*b*c^2*d^2 - B^2*a*b*c^2*d^2 + 3*C^2*a*b*c^2*d^2 - A*C*a*b*c^4 + A*C*a*b*d^4 + 2*A*C*a^2*c*d^3 - 2*A*C*b^2*c*d^3 - 4*A*C*a*b*c^2*d^2 + 4*A*B*a*b*c*d^3 - 4*B*C*a*b*c*d^3)/(d*(c^2 + d^2)^2) + (tan(e + f*x)*(A^2*a^2*d^4 + B^2*b^2*d^4 + C^2*a^2*d^4 + C^2*b^2*c^4 + C^2*b^2*d^4 + A^2*b^2*c^2*d^2 + B^2*a^2*c^2*d^2 + 3*C^2*b^2*c^2*d^2 - 2*A*C*a^2*d^4 - A*C*b^2*c^4 - A*C*b^2*d^4 - 4*A*C*b^2*c^2*d^2 - 2*A*B*a*b*d^4 - B*C*a*b*c^4 + B*C*a*b*d^4 - 2*A*B*a^2*c*d^3 + 2*A*B*b^2*c*d^3 + 2*B*C*a^2*c*d^3 - 2*B*C*b^2*c*d^3 - 2*A^2*a*b*c*d^3 + 2*B^2*a*b*c*d^3 - 2*C^2*a*b*c*d^3 + 2*A*B*a*b*c^2*d^2 - 4*B*C*a*b*c^2*d^2 + 4*A*C*a*b*c*d^3))/(d*(c^2 + d^2)^2) + ((a + b*1i)*(A + B*1i - C)*(A*a*d - B*b*d - C*a*d - 4*C*b*c + (tan(e + f*x)*(3*A*b*d^4 + 3*B*a*d^4 + 2*C*b*c^4 - 5*C*b*d^4 + 4*A*a*c*d^3 - 4*B*b*c*d^3 - 4*C*a*c*d^3 - A*b*c^2*d^2 - B*a*c^2*d^2 + C*b*c^2*d^2))/(d*(c^2 + d^2)) + (d*(a + b*1i)*(4*c*d - c^2*tan(e + f*x) + 3*d^2*tan(e + f*x))*(A + B*1i - C)*1i)/(c*1i - d)^2)*1i)/(2*(c*1i - d)^2))*(A*a + A*b*1i + B*a*1i - B*b - C*a - C*b*1i))/(2*f*(2*c*d - c^2*1i + d^2*1i)) - (A*a*d^3 - C*b*c^3 - A*b*c*d^2 - B*a*c*d^2 + B*b*c^2*d + C*a*c^2*d)/(d^2*f*(c^2 + d^2)*(c + d*tan(e + f*x)))

3.80 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^2}\,dx$

Optimal result 825
Rubi [A] (verified) 826
Mathematica [C] (verified) 827
Maple [A] (verified) 828
Fricas [A] (verification not implemented) 828
Sympy [C] (verification not implemented) 829
Maxima [A] (verification not implemented) 831
Giac [B] (verification not implemented) 832
Mupad [B] (verification not implemented) 832

Optimal result

Integrand size = 33, antiderivative size = 140

$$\begin{aligned}\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^2}\,dx &= -\frac{(c^2C-2Bcd-Cd^2-A(c^2-d^2))\,x}{(c^2+d^2)^2} \\ &\quad+\frac{(2c(A-C)d-B(c^2-d^2))\log(c\cos(e+fx)+d\sin(e+fx))}{(c^2+d^2)^2 f} \\ &\quad-\frac{c^2C-Bcd+Ad^2}{d\,(c^2+d^2)\,f(c+d\tan(e+fx))}\end{aligned}$$

```
[Out] -(c^2*C-2*B*c*d-C*d^2-A*(c^2-d^2))*x/(c^2+d^2)^2+(2*c*(A-C)*d-B*(c^2-d^2))*
ln(c*cos(f*x+e)+d*sin(f*x+e))/(c^2+d^2)^2/f+(-A*d^2+B*c*d-C*c^2)/d/(c^2+d^2
)/f/(c+d*tan(f*x+e))
```

Rubi [A] (verified)

Time = 0.25 (sec) , antiderivative size = 140, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 3, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.091, Rules used = {3709, 3612, 3611}

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^2}\,dx$$
$$= -\frac{Ad^2-Bcd+c^2C}{df\,(c^2+d^2)\,(c+d\tan(e+fx))} + \frac{(2cd(A-C)-B(c^2-d^2))\log(c\cos(e+fx)+d\sin(e+fx))}{f\,(c^2+d^2)^2} - \frac{x(-A(c^2-d^2)-2Bcd+c^2C-Cd^2)}{(c^2+d^2)^2}$$

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/(c + d*Tan[e + f*x])^2,x]
```

```
[Out] -(((c^2*C - 2*B*c*d - C*d^2 - A*(c^2 - d^2))*x)/(c^2 + d^2)^2) + ((2*c*(A -
 C)*d - B*(c^2 - d^2))*Log[c*Cos[e + f*x] + d*Sin[e + f*x]])/((c^2 + d^2)^2
*f) - (c^2*C - B*c*d + A*d^2)/(d*(c^2 + d^2)*f*(c + d*Tan[e + f*x]))
```

Rule 3611

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c/(b*f))*Log[RemoveContent[a*Cos[e + f*x] + b*Si
n[e + f*x], x]], x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3612

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[(a*c + b*d)*(x/(a^2 + b^2)), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3709

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 -
 a*b*B + a^2*C)*((a + b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1)*(a^2 + b^2))), x
] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[b*B + a*(A -
C) - (A*b - a*B - b*C)*Tan[e + f*x], x], x], x] /; FreeQ[{a, b, e, f, A, B,
 C}, x] && NeQ[A*b^2 - a*b*B + a^2*C, 0] && LtQ[m, -1] && NeQ[a^2 + b^2, 0]
```

Rubi steps

$$
\begin{aligned}
\text{integral} &= -\frac{c^2C - Bcd + Ad^2}{d\,(c^2+d^2)\,f(c+d\tan(e+fx))} + \frac{\int \frac{Ac-cC+Bd+(Bc-(A-C)d)\tan(e+fx)}{c+d\tan(e+fx)}\,dx}{c^2+d^2} \\
&= -\frac{(c^2C - 2Bcd - Cd^2 - A(c^2-d^2))\,x}{(c^2+d^2)^2} - \frac{c^2C - Bcd + Ad^2}{d\,(c^2+d^2)\,f(c+d\tan(e+fx))} \\
&\quad + \frac{(2c(A-C)d - B(c^2-d^2))\int \frac{d-c\tan(e+fx)}{c+d\tan(e+fx)}\,dx}{(c^2+d^2)^2} \\
&= -\frac{(c^2C - 2Bcd - Cd^2 - A(c^2-d^2))\,x}{(c^2+d^2)^2} \\
&\quad + \frac{(2c(A-C)d - B(c^2-d^2))\log(c\cos(e+fx)+d\sin(e+fx))}{(c^2+d^2)^2\,f} \\
&\quad - \frac{c^2C - Bcd + Ad^2}{d\,(c^2+d^2)\,f(c+d\tan(e+fx))}
\end{aligned}
$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 2.71 (sec) , antiderivative size = 207, normalized size of antiderivative = 1.48

$$
\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^2}\,dx
$$

$$
= \frac{\frac{B((-ic-d)\log(i-\tan(e+fx))+i(c+id)\log(i+\tan(e+fx))+2d\log(c+d\tan(e+fx)))}{c^2+d^2} - \frac{2C}{c+d\tan(e+fx)} + (Bc+(-A+C)d)\left(\frac{i\log(i-\cdots}{(\cdots}\right.}{2df}
$$

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/(c + d*Tan[e + f*x])^2,x]
```

```
[Out] ((B*(((-I)*c - d)*Log[I - Tan[e + f*x]] + I*(c + I*d)*Log[I + Tan[e + f*x]]
 + 2*d*Log[c + d*Tan[e + f*x]]))/(c^2 + d^2) - (2*C)/(c + d*Tan[e + f*x]) +
 (B*c + (-A + C)*d)*((I*Log[I - Tan[e + f*x]])/(c + I*d)^2 - (I*Log[I + Tan
[e + f*x]])/(c - I*d)^2 + (2*d*(-2*c*Log[c + d*Tan[e + f*x]] + (c^2 + d^2)/
(c + d*Tan[e + f*x])))/(c^2 + d^2)^2))/(2*d*f)
```

Maple [A] (verified)

Time = 0.08 (sec) , antiderivative size = 173, normalized size of antiderivative = 1.24

method	result
derivativedivides	$\frac{\frac{\frac{\left(-2Adc+B\,c^{2}-d^{2}B+2Ccd\right)\ln\left(1+\tan(fx+e)^{2}\right)}{2}+\left(A\,c^{2}-A\,d^{2}+2Bcd-c^{2}C+C\,d^{2}\right)\arctan(\tan(fx+e))}{\left(c^{2}+d^{2}\right)^{2}}-\frac{A\,d^{2}-Bcd+c^{2}C}{\left(c^{2}+d^{2}\right)d(c+d\tan(fx+e))}+}{f}$
default	$\frac{\frac{\frac{\left(-2Adc+B\,c^{2}-d^{2}B+2Ccd\right)\ln\left(1+\tan(fx+e)^{2}\right)}{2}+\left(A\,c^{2}-A\,d^{2}+2Bcd-c^{2}C+C\,d^{2}\right)\arctan(\tan(fx+e))}{\left(c^{2}+d^{2}\right)^{2}}-\frac{A\,d^{2}-Bcd+c^{2}C}{\left(c^{2}+d^{2}\right)d(c+d\tan(fx+e))}+}{f}$
norman	$\frac{\frac{c\left(A\,c^{2}-A\,d^{2}+2Bcd-c^{2}C+C\,d^{2}\right)x}{c^{4}+2c^{2}d^{2}+d^{4}}+\frac{d\left(A\,c^{2}-A\,d^{2}+2Bcd-c^{2}C+C\,d^{2}\right)x\tan(fx+e)}{c^{4}+2c^{2}d^{2}+d^{4}}-\frac{A\,d^{2}-Bcd+c^{2}C}{\left(c^{2}+d^{2}\right)df}}{c+d\tan(fx+e)}+\frac{(2Adc-B\,c^{2}+d^{2}B-2C}{f(c^{4}+2c^{2}}$
parallelrisch	$-\frac{2A\,c^{2}d^{2}+2Ax\tan(fx+e)d^{4}f-2B\,c^{3}d-2Bc\,d^{3}+2C\,c^{2}d^{2}+2c^{4}C+2A\,d^{4}-2Cx\tan(fx+e)d^{4}f-2Ax\,c^{3}df+2Axc\,d^{3}f-4B}{}$
risch	$\frac{4iCcde}{f(c^{4}+2c^{2}d^{2}+d^{4})}-\frac{xA}{2icd-c^{2}+d^{2}}+\frac{xC}{2icd-c^{2}+d^{2}}+\frac{2iA\,d^{2}}{(id+c)f(-id+c)^{2}\left(-id\,\mathrm{e}^{2i(fx+e)}+c\,\mathrm{e}^{2i(fx+e)}+id+c\right)}-\frac{4i}{f(c^{4}+2}$

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^2,x,method=_RETURNVERB
OSE)
```

```
[Out] 1/f*(1/(c^2+d^2)^2*(1/2*(-2*A*c*d+B*c^2-B*d^2+2*C*c*d)*ln(1+tan(f*x+e)^2)+(
A*c^2-A*d^2+2*B*c*d-C*c^2+C*d^2)*arctan(tan(f*x+e)))-(A*d^2-B*c*d+C*c^2)/(c
^2+d^2)/d/(c+d*tan(f*x+e))+(2*A*c*d-B*c^2+B*d^2-2*C*c*d)/(c^2+d^2)^2*ln(c+d
*tan(f*x+e)))
```

Fricas [A] (verification not implemented)

none

Time = 0.26 (sec) , antiderivative size = 256, normalized size of antiderivative = 1.83

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^2}\,dx =$$
$$-\frac{2\,Cc^2d-2\,Bcd^2+2\,Ad^3-2\left((A-C)c^3+2\,Bc^2d-(A-C)cd^2\right)fx+\left(Bc^3-2\,(A-C)c^2d-Bcd^2+(\right.}{2\,(}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^2,x, algorithm="
fricas")
```

```
[Out] -1/2*(2*C*c^2*d - 2*B*c*d^2 + 2*A*d^3 - 2*((A - C)*c^3 + 2*B*c^2*d - (A - C
)*c*d^2)*f*x + (B*c^3 - 2*(A - C)*c^2*d - B*c*d^2 + (B*c^2*d - 2*(A - C)*c*
d^2 - B*d^3)*tan(f*x + e))*log((d^2*tan(f*x + e)^2 + 2*c*d*tan(f*x + e) + c
^2)/(tan(f*x + e)^2 + 1)) - 2*(C*c^3 - B*c^2*d + A*c*d^2 + ((A - C)*c^2*d +
 2*B*c*d^2 - (A - C)*d^3)*f*x)*tan(f*x + e))/((c^4*d + 2*c^2*d^3 + d^5)*f*t
an(f*x + e) + (c^5 + 2*c^3*d^2 + c*d^4)*f)
```

Sympy [C] (verification not implemented)

Result contains complex when optimal does not.

Time = 0.92 (sec) , antiderivative size = 4396, normalized size of antiderivative = 31.40

$$\int \frac{A + B\tan(e+fx) + C\tan^2(e+fx)}{(c+d\tan(e+fx))^2}\,dx = \text{Too large to display}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x+e))**2,x)
```

```
[Out] Piecewise((zoo*x*(A + B*tan(e) + C*tan(e)**2)/tan(e)**2, Eq(c, 0) & Eq(d, 0
) & Eq(f, 0)), ((A*x + B*log(tan(e + f*x)**2 + 1)/(2*f) - C*x + C*tan(e + f
*x)/f)/c**2, Eq(d, 0)), (-A*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 -
 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 2*I*A*f*x*tan(e + f*x)/(4*d**2*f*tan
(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + A*f*x/(4*d**2*f*tan(e
+ f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - A*tan(e + f*x)/(4*d**2*f*
tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 2*I*A/(4*d**2*f*tan
(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + I*B*f*x*tan(e + f*x)**
2/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 2*B*f*x
*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*
f) - I*B*f*x/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f
) + I*B*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) -
4*d**2*f) + C*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*ta
n(e + f*x) - 4*d**2*f) - 2*I*C*f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 -
 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - C*f*x/(4*d**2*f*tan(e + f*x)**2 - 8*
I*d**2*f*tan(e + f*x) - 4*d**2*f) - 3*C*tan(e + f*x)/(4*d**2*f*tan(e + f*x)
**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 2*I*C/(4*d**2*f*tan(e + f*x)**2
 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f), Eq(c, -I*d)), (-A*f*x*tan(e + f*x)*
*2/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 2*I*A*
f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d*
*2*f) + A*f*x/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*
f) - A*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4
*d**2*f) - 2*I*A/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d*
*2*f) - I*B*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(
e + f*x) - 4*d**2*f) + 2*B*f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I
*d**2*f*tan(e + f*x) - 4*d**2*f) + I*B*f*x/(4*d**2*f*tan(e + f*x)**2 + 8*I*
d**2*f*tan(e + f*x) - 4*d**2*f) - I*B*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**
2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + C*f*x*tan(e + f*x)**2/(4*d**2*f*t
an(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 2*I*C*f*x*tan(e + f*
x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - C*f*x/
(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 3*C*tan(e
 + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 2
*I*C/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f), Eq(c,
 I*d)), (x*(A + B*tan(e) + C*tan(e)**2)/(c + d*tan(e))**2, Eq(f, 0)), (2*A*
c**3*d*f*x/(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f + 4*c**
```

```
2*d**4*f*tan(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) + 2*A*c**2*d**2
*f*x*tan(e + f*x)/(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f
+ 4*c**2*d**4*f*tan(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) + 4*A*c*
*2*d**2*log(c/d + tan(e + f*x))/(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*x) +
4*c**3*d**3*f + 4*c**2*d**4*f*tan(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan(e +
f*x)) - 2*A*c**2*d**2*log(tan(e + f*x)**2 + 1)/(2*c**5*d*f + 2*c**4*d**2*f*
tan(e + f*x) + 4*c**3*d**3*f + 4*c**2*d**4*f*tan(e + f*x) + 2*c*d**5*f + 2*
d**6*f*tan(e + f*x)) - 2*A*c**2*d**2/(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*
x) + 4*c**3*d**3*f + 4*c**2*d**4*f*tan(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan
(e + f*x)) - 2*A*c*d**3*f*x/(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c*
*3*d**3*f + 4*c**2*d**4*f*tan(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)
) + 4*A*c*d**3*log(c/d + tan(e + f*x))*tan(e + f*x)/(2*c**5*d*f + 2*c**4*d*
*2*f*tan(e + f*x) + 4*c**3*d**3*f + 4*c**2*d**4*f*tan(e + f*x) + 2*c*d**5*f
 + 2*d**6*f*tan(e + f*x)) - 2*A*c*d**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x
)/(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f + 4*c**2*d**4*f*
tan(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) - 2*A*d**4*f*x*tan(e + f
*x)/(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f + 4*c**2*d**4*
f*tan(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) - 2*A*d**4/(2*c**5*d*f
 + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f + 4*c**2*d**4*f*tan(e + f*x)
+ 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) - 2*B*c**3*d*log(c/d + tan(e + f*x))/
(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f + 4*c**2*d**4*f*ta
n(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) + B*c**3*d*log(tan(e + f*x
)**2 + 1)/(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f + 4*c**2
*d**4*f*tan(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) + 2*B*c**3*d/(2*
c**5*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f + 4*c**2*d**4*f*tan(e
 + f*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) + 4*B*c**2*d**2*f*x/(2*c**5*d
*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f + 4*c**2*d**4*f*tan(e + f*x
) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) - 2*B*c**2*d**2*log(c/d + tan(e + f
*x))*tan(e + f*x)/(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f
+ 4*c**2*d**4*f*tan(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) + B*c**2
*d**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*c**5*d*f + 2*c**4*d**2*f*tan
(e + f*x) + 4*c**3*d**3*f + 4*c**2*d**4*f*tan(e + f*x) + 2*c*d**5*f + 2*d**
6*f*tan(e + f*x)) + 4*B*c*d**3*f*x*tan(e + f*x)/(2*c**5*d*f + 2*c**4*d**2*f
*tan(e + f*x) + 4*c**3*d**3*f + 4*c**2*d**4*f*tan(e + f*x) + 2*c*d**5*f + 2
*d**6*f*tan(e + f*x)) + 2*B*c*d**3*log(c/d + tan(e + f*x))/(2*c**5*d*f + 2*
c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f + 4*c**2*d**4*f*tan(e + f*x) + 2*c
*d**5*f + 2*d**6*f*tan(e + f*x)) - B*c*d**3*log(tan(e + f*x)**2 + 1)/(2*c**
5*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f + 4*c**2*d**4*f*tan(e +
f*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) + 2*B*c*d**3/(2*c**5*d*f + 2*c**
4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f + 4*c**2*d**4*f*tan(e + f*x) + 2*c*d*
*5*f + 2*d**6*f*tan(e + f*x)) + 2*B*d**4*log(c/d + tan(e + f*x))*tan(e + f*
x)/(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f + 4*c**2*d**4*f
*tan(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) - B*d**4*log(tan(e + f*
x)**2 + 1)*tan(e + f*x)/(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d
**3*f + 4*c**2*d**4*f*tan(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) -
```

```
2*C*c**4/(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f + 4*c**2*
d**4*f*tan(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) - 2*C*c**3*d*f*x/
(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f + 4*c**2*d**4*f*ta
n(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) - 2*C*c**2*d**2*f*x*tan(e
+ f*x)/(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f + 4*c**2*d*
*4*f*tan(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) - 4*C*c**2*d**2*log
(c/d + tan(e + f*x))/(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3
*f + 4*c**2*d**4*f*tan(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) + 2*C
*c**2*d**2*log(tan(e + f*x)**2 + 1)/(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*x
) + 4*c**3*d**3*f + 4*c**2*d**4*f*tan(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan(
e + f*x)) - 2*C*c**2*d**2/(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3
*d**3*f + 4*c**2*d**4*f*tan(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x))
+ 2*C*c*d**3*f*x/(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f +
 4*c**2*d**4*f*tan(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) - 4*C*c*d
**3*log(c/d + tan(e + f*x))*tan(e + f*x)/(2*c**5*d*f + 2*c**4*d**2*f*tan(e
+ f*x) + 4*c**3*d**3*f + 4*c**2*d**4*f*tan(e + f*x) + 2*c*d**5*f + 2*d**6*f
*tan(e + f*x)) + 2*C*c*d**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*c**5*d
*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f + 4*c**2*d**4*f*tan(e + f*x
) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) + 2*C*d**4*f*x*tan(e + f*x)/(2*c**5
*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f + 4*c**2*d**4*f*tan(e + f
*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)), True))
```

Maxima [A] (verification not implemented)

none

Time = 0.34 (sec) , antiderivative size = 205, normalized size of antiderivative = 1.46

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^2}\,dx = \frac{\frac{2\left((A-C)c^2+2\,Bcd-(A-C)d^2\right)(fx+e)}{c^4+2\,c^2d^2+d^4} - \frac{2\left(Bc^2-2\,(A-C)cd-Bd^2\right)\log(d\tan(fx+e)+c)}{c^4+2\,c^2d^2+d^4} + \frac{\left(Bc^2-2\,(A-C)cd-Bd^2\right)\log\left(\tan(fx+e)^2+1\right)}{c^4+2\,c^2d^2+d^4}}{2\,f}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^2,x, algorithm="
maxima")
```

```
[Out] 1/2*(2*((A - C)*c^2 + 2*B*c*d - (A - C)*d^2)*(f*x + e)/(c^4 + 2*c^2*d^2 + d
^4) - 2*(B*c^2 - 2*(A - C)*c*d - B*d^2)*log(d*tan(f*x + e) + c)/(c^4 + 2*c^
2*d^2 + d^4) + (B*c^2 - 2*(A - C)*c*d - B*d^2)*log(tan(f*x + e)^2 + 1)/(c^4
 + 2*c^2*d^2 + d^4) - 2*(C*c^2 - B*c*d + A*d^2)/(c^3*d + c*d^3 + (c^2*d^2 +
 d^4)*tan(f*x + e)))/f
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 291 vs. 2(140) = 280.

Time = 0.54 (sec) , antiderivative size = 291, normalized size of antiderivative = 2.08

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^2}\,dx$$
$$= \frac{\frac{2\,(Ac^2-Cc^2+2\,Bcd-Ad^2+Cd^2)(fx+e)}{c^4+2\,c^2d^2+d^4} + \frac{(Bc^2-2\,Acd+2\,Ccd-Bd^2)\log\left(\tan(fx+e)^2+1\right)}{c^4+2\,c^2d^2+d^4} - \frac{2\,(Bc^2d-2\,Acd^2+2\,Ccd^2-Bd^3)\log(|d\tan(fx+}{c^4d+2\,c^2d^3+d^5}}{2\,f}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^2,x, algorithm="
giac")
```

```
[Out] 1/2*(2*(A*c^2 - C*c^2 + 2*B*c*d - A*d^2 + C*d^2)*(f*x + e)/(c^4 + 2*c^2*d^2
 + d^4) + (B*c^2 - 2*A*c*d + 2*C*c*d - B*d^2)*log(tan(f*x + e)^2 + 1)/(c^4
+ 2*c^2*d^2 + d^4) - 2*(B*c^2*d - 2*A*c*d^2 + 2*C*c*d^2 - B*d^3)*log(abs(d*
tan(f*x + e) + c))/(c^4*d + 2*c^2*d^3 + d^5) + 2*(B*c^2*d^2*tan(f*x + e) -
2*A*c*d^3*tan(f*x + e) + 2*C*c*d^3*tan(f*x + e) - B*d^4*tan(f*x + e) - C*c^
4 + 2*B*c^3*d - 3*A*c^2*d^2 + C*c^2*d^2 - A*d^4)/((c^4*d + 2*c^2*d^3 + d^5)
*(d*tan(f*x + e) + c)))/f
```

Mupad [B] (verification not implemented)

Time = 10.55 (sec) , antiderivative size = 184, normalized size of antiderivative = 1.31

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^2}\,dx$$
$$= \frac{\ln\left(c+d\tan(e+f\,x)\right)\,\left(-B\,c^2+\left(2\,A-2\,C\right)\,c\,d+B\,d^2\right)}{f\,\left(c^4+2\,c^2\,d^2+d^4\right)} - \frac{\ln\left(\tan(e+f\,x)-\mathrm{i}\right)\,\left(A-C+B\,1\mathrm{i}\right)}{2\,f\,\left(-c^2\,1\mathrm{i}+2\,c\,d+d^2\,1\mathrm{i}\right)} - \frac{\ln\left(\tan(e+f\,x)+1\mathrm{i}\right)\,\left(A\,1\mathrm{i}+B-C\,1\mathrm{i}\right)}{2\,f\,\left(-c^2+c\,d\,2\mathrm{i}+d^2\right)} - \frac{C\,c^2-B\,c\,d+A\,d^2}{d\,f\,\left(c^2+d^2\right)\,\left(c+d\tan\left(e+f\,x\right)\right)}$$

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/(c + d*tan(e + f*x))^2,x)
```

```
[Out] (log(c + d*tan(e + f*x))*(B*d^2 - B*c^2 + c*d*(2*A - 2*C)))/(f*(c^4 + d^4 +
 2*c^2*d^2)) - (log(tan(e + f*x) - 1i)*(A + B*1i - C))/(2*f*(2*c*d - c^2*1i
 + d^2*1i)) - (log(tan(e + f*x) + 1i)*(A*1i + B - C*1i))/(2*f*(c*d*2i - c^2
 + d^2)) - (A*d^2 + C*c^2 - B*c*d)/(d*f*(c^2 + d^2)*(c + d*tan(e + f*x)))
```

3.81 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^2} \, dx$

Optimal result 833
Rubi [A] (verified) 834
Mathematica [B] (verified) 836
Maple [A] (verified) 837
Fricas [B] (verification not implemented) 837
Sympy [F(-2)] 838
Maxima [A] (verification not implemented) 838
Giac [B] (verification not implemented) 839
Mupad [B] (verification not implemented) 840

Optimal result

Integrand size = 45, antiderivative size = 293

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^2} \, dx$$
$$= -\frac{(a(c^2C-2Bcd-Cd^2-A(c^2-d^2))+b(2c(A-C)d-B(c^2-d^2)))\,x}{(a^2+b^2)\,(c^2+d^2)^2}$$
$$+\frac{b(Ab^2-a(bB-aC))\log(a\cos(e+fx)+b\sin(e+fx))}{(a^2+b^2)\,(bc-ad)^2 f}$$
$$-\frac{(b(c^4C-2Bc^3d+c^2(3A-C)d^2+Ad^4)-ad^2(2c(A-C)d-B(c^2-d^2)))\log(c\cos(e+fx)+d\sin(e}{(bc-ad)^2\,(c^2+d^2)^2 f}$$
$$+\frac{c^2C-Bcd+Ad^2}{(bc-ad)\,(c^2+d^2)\,f(c+d\tan(e+fx))}$$

```
[Out] -(a*(c^2*C-2*B*c*d-C*d^2-A*(c^2-d^2))+b*(2*c*(A-C)*d-B*(c^2-d^2)))*x/(a^2+b
^2)/(c^2+d^2)^2+b*(A*b^2-a*(B*b-C*a))*ln(a*cos(f*x+e)+b*sin(f*x+e))/(a^2+b^
2)/(-a*d+b*c)^2/f-(b*(c^4*C-2*B*c^3*d+c^2*(3*A-C)*d^2+A*d^4)-a*d^2*(2*c*(A-
C)*d-B*(c^2-d^2)))*ln(c*cos(f*x+e)+d*sin(f*x+e))/(-a*d+b*c)^2/(c^2+d^2)^2/f
+(A*d^2-B*c*d+C*c^2)/(-a*d+b*c)/(c^2+d^2)/f/(c+d*tan(f*x+e))
```

Rubi [A] (verified)

Time = 0.90 (sec) , antiderivative size = 293, normalized size of antiderivative = 1.00, number of steps used = 4, number of rules used = 3, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.067, Rules used = {3730, 3732, 3611}

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^2}\,dx$$
$$= -\frac{x(a(-A(c^2-d^2)-2Bcd+c^2C-Cd^2)+b(2cd(A-C)-B(c^2-d^2)))}{(a^2+b^2)\,(c^2+d^2)^2} + \frac{b(Ab^2-a(bB-aC))\log(a\cos(e+fx)+b\sin(e+fx))}{f\,(a^2+b^2)\,(bc-ad)^2} + \frac{Ad^2-Bcd+c^2C}{f\,(c^2+d^2)\,(bc-ad)(c+d\tan(e+fx))} - \frac{(b(c^2d^2(3A-C)+Ad^4-2Bc^3d+c^4C)-ad^2(2cd(A-C)-B(c^2-d^2)))\log(c\cos(e+fx)+d\sin(e+ }{f\,(c^2+d^2)^2\,(bc-ad)^2}$$

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])*(c + d*Ta
n[e + f*x])^2),x]
```

```
[Out] -(((a*(c^2*C - 2*B*c*d - C*d^2 - A*(c^2 - d^2)) + b*(2*c*(A - C)*d - B*(c^2
 - d^2)))*x)/((a^2 + b^2)*(c^2 + d^2)^2)) + (b*(A*b^2 - a*(b*B - a*C))*Log[
a*Cos[e + f*x] + b*Sin[e + f*x]])/((a^2 + b^2)*(b*c - a*d)^2*f) - ((b*(c^4*
C - 2*B*c^3*d + c^2*(3*A - C)*d^2 + A*d^4) - a*d^2*(2*c*(A - C)*d - B*(c^2
- d^2)))*Log[c*Cos[e + f*x] + d*Sin[e + f*x]])/((b*c - a*d)^2*(c^2 + d^2)^2
*f) + (c^2*C - B*c*d + A*d^2)/((b*c - a*d)*(c^2 + d^2)*f*(c + d*Tan[e + f*x
]))
```

Rule 3611

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c/(b*f))*Log[RemoveContent[a*Cos[e + f*x] + b*Si
n[e + f*x], x]], x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3730

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 - a*(b*B - a*C))*(a + b*Tan[e +
 f*x])^(m + 1)*((c + d*Tan[e + f*x])^(n + 1)/(f*(m + 1)*(b*c - a*d)*(a^2 +
b^2))), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
```

```
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3732

```
Int[((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^
2)/(((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])), x_Symbol] :> Simp[(a*(A*c - c*C + B*d) + b*(B*c - A*d + C*d))*(x/
((a^2 + b^2)*(c^2 + d^2))), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/((b*c - a*d)
*(a^2 + b^2)), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] - Dist
[(c^2*C - B*c*d + A*d^2)/((b*c - a*d)*(c^2 + d^2)), Int[(d - c*Tan[e + f*x]
)/(c + d*Tan[e + f*x]), x], x]) /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] &&
 NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0]
```

Rubi steps

integral

$$= \frac{c^2C - Bcd + Ad^2}{(bc-ad)\left(c^2+d^2\right)f(c+d\tan(e+fx))} + \frac{\int \frac{-aAcd+ad(cC-Bd)+Ab\left(c^2+d^2\right)+(bc-ad)(Bc-(A-C)d)\tan(e+fx)+b\left(c^2C-Bcd+Ad^2\right)\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))}\,dx}{(bc-ad)\left(c^2+d^2\right)}$$

$$= -\frac{\left(a\left(c^2C-2Bcd-Cd^2-A\left(c^2-d^2\right)\right)+b\left(2c(A-C)d-B\left(c^2-d^2\right)\right)\right)x}{\left(a^2+b^2\right)\left(c^2+d^2\right)^2} + \frac{c^2C-Bcd+Ad^2}{(bc-ad)\left(c^2+d^2\right)f(c+d\tan(e+fx))} + \frac{\left(b\left(Ab^2-a(bB-aC)\right)\right)\int\frac{b-a\tan(e+fx)}{a+b\tan(e+fx)}\,dx}{\left(a^2+b^2\right)(bc-ad)^2} - \frac{\left(b\left(c^4C-2Bc^3d+c^2(3A-C)d^2+Ad^4\right)-ad^2\left(2c(A-C)d-B\left(c^2-d^2\right)\right)\right)\int\frac{d-c\tan(e+fx)}{c+d\tan(e+fx)}\,dx}{(bc-ad)^2\left(c^2+d^2\right)^2}$$

$$= -\frac{\left(a\left(c^2C-2Bcd-Cd^2-A\left(c^2-d^2\right)\right)+b\left(2c(A-C)d-B\left(c^2-d^2\right)\right)\right)x}{\left(a^2+b^2\right)\left(c^2+d^2\right)^2} + \frac{b\left(Ab^2-a(bB-aC)\right)\log(a\cos(e+fx)+b\sin(e+fx))}{\left(a^2+b^2\right)(bc-ad)^2f} - \frac{\left(b\left(c^4C-2Bc^3d+c^2(3A-C)d^2+Ad^4\right)-ad^2\left(2c(A-C)d-B\left(c^2-d^2\right)\right)\right)\log(c\cos(e+fx)+d\sin(e}{(bc-ad)^2\left(c^2+d^2\right)^2f} + \frac{c^2C-Bcd+Ad^2}{(bc-ad)\left(c^2+d^2\right)f(c+d\tan(e+fx))}$$

Mathematica [B] (verified)

Leaf count is larger than twice the leaf count of optimal. 592 vs. $2(293) = 586$.

Time = 7.59 (sec) , antiderivative size = 592, normalized size of antiderivative = 2.02

$$\int \frac{A + B\tan(e+fx) + C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^2}\,dx =$$

$$-\frac{-\frac{b(bc-ad)\left(Abc^2-aBc^2-bc^2C+2aAcd+2bBcd-2acCd-Abd^2+aBd^2+bCd^2-\frac{\sqrt{-b^2}\left(a\left(c^2C-2Bcd-Cd^2-A\left(c^2-d^2\right)\right)+b\left(2c(A-C)d-B\left(c^2-d^2\right)\right)\right.}{b}\right.}{2(a^2+b^2)(c^2+d^2)}}{}$$

$$-\frac{Ad^2-c(-cC+Bd)}{(-bc+ad)\left(c^2+d^2\right)f(c+d\tan(e+fx))}$$

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])*(c
+ d*Tan[e + f*x])^2),x]
```

```
[Out] -((-1/2*(b*(b*c - a*d)*(A*b*c^2 - a*B*c^2 - b*c^2*C + 2*a*A*c*d + 2*b*B*c*d
 - 2*a*c*C*d - A*b*d^2 + a*B*d^2 + b*C*d^2 - (Sqrt[-b^2]*(a*(c^2*C - 2*B*c*
d - C*d^2 - A*(c^2 - d^2)) + b*(2*c*(A - C)*d - B*(c^2 - d^2))))/b)*Log[Sqr
t[-b^2] - b*Tan[e + f*x]])/((a^2 + b^2)*(c^2 + d^2)) + (b^2*(A*b^2 - a*(b*B
 - a*C))*(c^2 + d^2)*Log[a + b*Tan[e + f*x]])/((a^2 + b^2)*(b*c - a*d)) - (
b*(b*c - a*d)*(A*b*c^2 - a*B*c^2 - b*c^2*C + 2*a*A*c*d + 2*b*B*c*d - 2*a*c*
C*d - A*b*d^2 + a*B*d^2 + b*C*d^2 + (Sqrt[-b^2]*(a*(c^2*C - 2*B*c*d - C*d^2
 - A*(c^2 - d^2)) + b*(2*c*(A - C)*d - B*(c^2 - d^2))))/b)*Log[Sqrt[-b^2] +
 b*Tan[e + f*x]])/(2*(a^2 + b^2)*(c^2 + d^2)) - (b*(b*(c^4*C - 2*B*c^3*d +
c^2*(3*A - C)*d^2 + A*d^4) - a*d^2*(2*c*(A - C)*d - B*(c^2 - d^2)))*Log[c +
 d*Tan[e + f*x]])/((b*c - a*d)*(c^2 + d^2)))/(b*(-(b*c) + a*d)*(c^2 + d^2)*
f)) - (A*d^2 - c*(-(c*C) + B*d))/((-(b*c) + a*d)*(c^2 + d^2)*f*(c + d*Tan[e
 + f*x]))
```

Maple [A] (verified)

Time = 0.40 (sec) , antiderivative size = 365, normalized size of antiderivative = 1.25

method	result
derivativedivides	$\frac{\frac{(A b^2-Bab+C a^2) b \ln(a+b\tan(fx+e))}{(ad-bc)^2(a^2+b^2)}+\frac{\frac{(-2Aacd-Ab c^2+Ab d^2+Ba c^2-Ba d^2-2Bbcd+2Cacd+Cb c^2-Cb d^2)\ln(1+\tan(fx+e)^2)}{2}}{(a^2+b^2)}}{}$
default	$\frac{\frac{(A b^2-Bab+C a^2) b \ln(a+b\tan(fx+e))}{(ad-bc)^2(a^2+b^2)}+\frac{\frac{(-2Aacd-Ab c^2+Ab d^2+Ba c^2-Ba d^2-2Bbcd+2Cacd+Cb c^2-Cb d^2)\ln(1+\tan(fx+e)^2)}{2}}{(a^2+b^2)}}{}$
norman	$\frac{\frac{(Aa c^2-Aa d^2-2Abcd+2Bacd+Bb c^2-Bb d^2-Ca c^2+Ca d^2+2Cbcd)cx}{(a^2+b^2)(c^4+2c^2d^2+d^4)}+\frac{(Aa c^2-Aa d^2-2Abcd+2Bacd+Bb c^2-Bb d^2-Ca c^2+Ca}{(a^2+b^2)(c^4+2c^2d^2+d^4)}}{c+d\tan(fx+e)}$
parallelrisch	Expression too large to display
risch	Expression too large to display

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))^2,x,m
ethod=_RETURNVERBOSE)
```

```
[Out] 1/f*((A*b^2-B*a*b+C*a^2)*b/(a*d-b*c)^2/(a^2+b^2)*ln(a+b*tan(f*x+e))+1/(a^2+
b^2)/(c^2+d^2)^2*(1/2*(-2*A*a*c*d-A*b*c^2+A*b*d^2+B*a*c^2-B*a*d^2-2*B*b*c*d
+2*C*a*c*d+C*b*c^2-C*b*d^2)*ln(1+tan(f*x+e)^2)+(A*a*c^2-A*a*d^2-2*A*b*c*d+2
*B*a*c*d+B*b*c^2-B*b*d^2-C*a*c^2+C*a*d^2+2*C*b*c*d)*arctan(tan(f*x+e)))+(2*
A*a*c*d^3-3*A*b*c^2*d^2-A*b*d^4-B*a*c^2*d^2+B*a*d^4+2*B*b*c^3*d-2*C*a*c*d^3
-C*b*c^4+C*b*c^2*d^2)/(a*d-b*c)^2/(c^2+d^2)^2*ln(c+d*tan(f*x+e))-(A*d^2-B*c
*d+C*c^2)/(a*d-b*c)/(c^2+d^2)/(c+d*tan(f*x+e)))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1275 vs. 2(291) = 582.

Time = 1.11 (sec) , antiderivative size = 1275, normalized size of antiderivative = 4.35

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^2}\,dx = \text{Too large to display}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))
^2,x, algorithm="fricas")
```

```
[Out] 1/2*(2*(C*a^2*b + C*b^3)*c^3*d^2 - 2*(C*a^3 + B*a^2*b + C*a*b^2 + B*b^3)*c^
2*d^3 + 2*(B*a^3 + A*a^2*b + B*a*b^2 + A*b^3)*c*d^4 - 2*(A*a^3 + A*a*b^2)*d
^5 + 2*(((A - C)*a*b^2 + B*b^3)*c^5 - 2*((A - C)*a^2*b + (A - C)*b^3)*c^4*d
 + ((A - C)*a^3 - 3*B*a^2*b + 3*(A - C)*a*b^2 - B*b^3)*c^3*d^2 + 2*(B*a^3 +
 B*a*b^2)*c^2*d^3 - ((A - C)*a^3 + B*a^2*b)*c*d^4)*f*x + ((C*a^2*b - B*a*b^
2 + A*b^3)*c^5 + 2*(C*a^2*b - B*a*b^2 + A*b^3)*c^3*d^2 + (C*a^2*b - B*a*b^2
```

```
 + A*b^3)*c*d^4 + ((C*a^2*b - B*a*b^2 + A*b^3)*c^4*d + 2*(C*a^2*b - B*a*b^2
 + A*b^3)*c^2*d^3 + (C*a^2*b - B*a*b^2 + A*b^3)*d^5)*tan(f*x + e))*log((b^2
*tan(f*x + e)^2 + 2*a*b*tan(f*x + e) + a^2)/(tan(f*x + e)^2 + 1)) - ((C*a^2
*b + C*b^3)*c^5 - 2*(B*a^2*b + B*b^3)*c^4*d + (B*a^3 + (3*A - C)*a^2*b + B*
a*b^2 + (3*A - C)*b^3)*c^3*d^2 - 2*((A - C)*a^3 + (A - C)*a*b^2)*c^2*d^3 -
(B*a^3 - A*a^2*b + B*a*b^2 - A*b^3)*c*d^4 + ((C*a^2*b + C*b^3)*c^4*d - 2*(B
*a^2*b + B*b^3)*c^3*d^2 + (B*a^3 + (3*A - C)*a^2*b + B*a*b^2 + (3*A - C)*b^
3)*c^2*d^3 - 2*((A - C)*a^3 + (A - C)*a*b^2)*c*d^4 - (B*a^3 - A*a^2*b + B*a
*b^2 - A*b^3)*d^5)*tan(f*x + e))*log((d^2*tan(f*x + e)^2 + 2*c*d*tan(f*x +
e) + c^2)/(tan(f*x + e)^2 + 1)) - 2*((C*a^2*b + C*b^3)*c^4*d - (C*a^3 + B*a
^2*b + C*a*b^2 + B*b^3)*c^3*d^2 + (B*a^3 + A*a^2*b + B*a*b^2 + A*b^3)*c^2*d
^3 - (A*a^3 + A*a*b^2)*c*d^4 - (((A - C)*a*b^2 + B*b^3)*c^4*d - 2*((A - C)*
a^2*b + (A - C)*b^3)*c^3*d^2 + ((A - C)*a^3 - 3*B*a^2*b + 3*(A - C)*a*b^2 -
 B*b^3)*c^2*d^3 + 2*(B*a^3 + B*a*b^2)*c*d^4 - ((A - C)*a^3 + B*a^2*b)*d^5)*
f*x)*tan(f*x + e))/(((a^2*b^2 + b^4)*c^6*d - 2*(a^3*b + a*b^3)*c^5*d^2 + (a
^4 + 3*a^2*b^2 + 2*b^4)*c^4*d^3 - 4*(a^3*b + a*b^3)*c^3*d^4 + (2*a^4 + 3*a^
2*b^2 + b^4)*c^2*d^5 - 2*(a^3*b + a*b^3)*c*d^6 + (a^4 + a^2*b^2)*d^7)*f*tan
(f*x + e) + ((a^2*b^2 + b^4)*c^7 - 2*(a^3*b + a*b^3)*c^6*d + (a^4 + 3*a^2*b
^2 + 2*b^4)*c^5*d^2 - 4*(a^3*b + a*b^3)*c^4*d^3 + (2*a^4 + 3*a^2*b^2 + b^4)
*c^3*d^4 - 2*(a^3*b + a*b^3)*c^2*d^5 + (a^4 + a^2*b^2)*c*d^6)*f)
```

Sympy [F(-2)]

Exception generated.

$$\int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{(a + b\tan(e + fx))(c + d\tan(e + fx))^2}\,dx = \text{Exception raised: NotImplementedError}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e)
)**2,x)
```

```
[Out] Exception raised: NotImplementedError >> no valid subset found
```

Maxima [A] (verification not implemented)

none

Time = 0.35 (sec) , antiderivative size = 513, normalized size of antiderivative = 1.75

$$\int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{(a + b\tan(e + fx))(c + d\tan(e + fx))^2}\,dx$$
$$= \frac{\frac{2\left(((A-C)a+Bb)c^2+2\left(Ba-(A-C)b\right)cd-((A-C)a+Bb)d^2\right)(fx+e)}{(a^2+b^2)c^4+2\,(a^2+b^2)c^2d^2+(a^2+b^2)d^4} + \frac{2\left(Ca^2b-Bab^2+Ab^3\right)\log(b\tan(fx+e)+a)}{(a^2b^2+b^4)c^2-2\,(a^3b+ab^3)cd+(a^4+a^2b^2)d^2} - \frac{2\left(Cbc^4-2\,Bbc^3d-2\,(A-\right.}{b^2c^6-2\,abc^5d-}}{}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))
^2,x, algorithm="maxima")
```

```
[Out] 1/2*(2*(((A - C)*a + B*b)*c^2 + 2*(B*a - (A - C)*b)*c*d - ((A - C)*a + B*b)
*d^2)*(f*x + e)/((a^2 + b^2)*c^4 + 2*(a^2 + b^2)*c^2*d^2 + (a^2 + b^2)*d^4)
 + 2*(C*a^2*b - B*a*b^2 + A*b^3)*log(b*tan(f*x + e) + a)/((a^2*b^2 + b^4)*c
^2 - 2*(a^3*b + a*b^3)*c*d + (a^4 + a^2*b^2)*d^2) - 2*(C*b*c^4 - 2*B*b*c^3*
d - 2*(A - C)*a*c*d^3 + (B*a + (3*A - C)*b)*c^2*d^2 - (B*a - A*b)*d^4)*log(
d*tan(f*x + e) + c)/(b^2*c^6 - 2*a*b*c^5*d - 4*a*b*c^3*d^3 - 2*a*b*c*d^5 +
a^2*d^6 + (a^2 + 2*b^2)*c^4*d^2 + (2*a^2 + b^2)*c^2*d^4) + ((B*a - (A - C)*
b)*c^2 - 2*((A - C)*a + B*b)*c*d - (B*a - (A - C)*b)*d^2)*log(tan(f*x + e)^
2 + 1)/((a^2 + b^2)*c^4 + 2*(a^2 + b^2)*c^2*d^2 + (a^2 + b^2)*d^4) + 2*(C*c
^2 - B*c*d + A*d^2)/(b*c^4 - a*c^3*d + b*c^2*d^2 - a*c*d^3 + (b*c^3*d - a*c
^2*d^2 + b*c*d^3 - a*d^4)*tan(f*x + e)))/f
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 832 vs. $2(291) = 582$.

Time = 0.77 (sec) , antiderivative size = 832, normalized size of antiderivative = 2.84

$$\int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{(a + b\tan(e + fx))(c + d\tan(e + fx))^2}\, dx$$
$$= \frac{\frac{2\left(Aac^2 - Cac^2 + Bbc^2 + 2\,Bacd - 2\,Abcd + 2\,Cbcd - Aad^2 + Cad^2 - Bbd^2\right)(fx+e)}{a^2c^4 + b^2c^4 + 2\,a^2c^2d^2 + 2\,b^2c^2d^2 + a^2d^4 + b^2d^4} + \frac{\left(Bac^2 - Abc^2 + Cbc^2 - 2\,Aacd + 2\,Cacd - 2\,Bbcd - Bad^2 + Abd^2 - \right.}{a^2c^4 + b^2c^4 + 2\,a^2c^2d^2 + 2\,b^2c^2d^2 + a^2d^4 +}}{}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))
^2,x, algorithm="giac")
```

```
[Out] 1/2*(2*(A*a*c^2 - C*a*c^2 + B*b*c^2 + 2*B*a*c*d - 2*A*b*c*d + 2*C*b*c*d - A
*a*d^2 + C*a*d^2 - B*b*d^2)*(f*x + e)/(a^2*c^4 + b^2*c^4 + 2*a^2*c^2*d^2 +
2*b^2*c^2*d^2 + a^2*d^4 + b^2*d^4) + (B*a*c^2 - A*b*c^2 + C*b*c^2 - 2*A*a*c
*d + 2*C*a*c*d - 2*B*b*c*d - B*a*d^2 + A*b*d^2 - C*b*d^2)*log(tan(f*x + e)^
2 + 1)/(a^2*c^4 + b^2*c^4 + 2*a^2*c^2*d^2 + 2*b^2*c^2*d^2 + a^2*d^4 + b^2*d
^4) + 2*(C*a^2*b^2 - B*a*b^3 + A*b^4)*log(abs(b*tan(f*x + e) + a))/(a^2*b^3
*c^2 + b^5*c^2 - 2*a^3*b^2*c*d - 2*a*b^4*c*d + a^4*b*d^2 + a^2*b^3*d^2) - 2
*(C*b*c^4*d - 2*B*b*c^3*d^2 + B*a*c^2*d^3 + 3*A*b*c^2*d^3 - C*b*c^2*d^3 - 2
*A*a*c*d^4 + 2*C*a*c*d^4 - B*a*d^5 + A*b*d^5)*log(abs(d*tan(f*x + e) + c))/
(b^2*c^6*d - 2*a*b*c^5*d^2 + a^2*c^4*d^3 + 2*b^2*c^4*d^3 - 4*a*b*c^3*d^4 +
2*a^2*c^2*d^5 + b^2*c^2*d^5 - 2*a*b*c*d^6 + a^2*d^7) + 2*(C*b*c^4*d*tan(f*x
 + e) - 2*B*b*c^3*d^2*tan(f*x + e) + B*a*c^2*d^3*tan(f*x + e) + 3*A*b*c^2*d
^3*tan(f*x + e) - C*b*c^2*d^3*tan(f*x + e) - 2*A*a*c*d^4*tan(f*x + e) + 2*C
*a*c*d^4*tan(f*x + e) - B*a*d^5*tan(f*x + e) + A*b*d^5*tan(f*x + e) + 2*C*b
*c^5 - C*a*c^4*d - 3*B*b*c^4*d + 2*B*a*c^3*d^2 + 4*A*b*c^3*d^2 - 3*A*a*c^2*
d^3 + C*a*c^2*d^3 - B*b*c^2*d^3 + 2*A*b*c*d^4 - A*a*d^5)/((b^2*c^6 - 2*a*b*
```

```
c^5*d + a^2*c^4*d^2 + 2*b^2*c^4*d^2 - 4*a*b*c^3*d^3 + 2*a^2*c^2*d^4 + b^2*c
^2*d^4 - 2*a*b*c*d^5 + a^2*d^6)*(d*tan(f*x + e) + c)))/f
```

Mupad [B] (verification not implemented)

Time = 65.17 (sec) , antiderivative size = 430, normalized size of antiderivative = 1.47

$$
\begin{aligned}
&\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^2}\,dx \\
&= \frac{\ln(\tan(e+f\,x)-\mathrm{i})\,(B-A\,1\mathrm{i}+C\,1\mathrm{i})}{2\,f\,(a\,c^2-a\,d^2-2\,b\,c\,d+b\,c^2\,1\mathrm{i}-b\,d^2\,1\mathrm{i}+a\,c\,d\,2\mathrm{i})} \\
&\quad-\frac{\ln(\tan(e+f\,x)+1\mathrm{i})\,(A\,1\mathrm{i}+B-C\,1\mathrm{i})}{2\,f\,(a\,d^2-a\,c^2+2\,b\,c\,d+b\,c^2\,1\mathrm{i}-b\,d^2\,1\mathrm{i}+a\,c\,d\,2\mathrm{i})} \\
&\quad+\frac{\ln(a+b\tan(e+f\,x))\,(C\,a^2\,b-B\,a\,b^2+A\,b^3)}{f\,(a^4\,d^2-2\,a^3\,b\,c\,d+a^2\,b^2\,c^2+a^2\,b^2\,d^2-2\,a\,b^3\,c\,d+b^4\,c^2)} \\
&\quad-\frac{\ln(c+d\tan(e+f\,x))\,(C\,b\,c^4-2\,B\,b\,c^3\,d+(3\,A\,b+B\,a-C\,b)\,c^2\,d^2+(2\,C\,a-2\,A\,a)\,c\,d^3+(A\,b-B\,a}{f\,(a^2\,c^4\,d^2+2\,a^2\,c^2\,d^4+a^2\,d^6-2\,a\,b\,c^5\,d-4\,a\,b\,c^3\,d^3-2\,a\,b\,c\,d^5+b^2\,c^6+2\,b^2\,c^4\,d^2+b^2\,c^2\,d^4)} \\
&\quad-\frac{C\,c^2-B\,c\,d+A\,d^2}{f\,(a\,d-b\,c)\,(c^2+d^2)\,(c+d\tan(e+f\,x))}
\end{aligned}
$$

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))*(c + d*ta
n(e + f*x))^2),x)
```

```
[Out] (log(tan(e + f*x) - 1i)*(B - A*1i + C*1i))/(2*f*(a*c^2 - a*d^2 + b*c^2*1i -
 b*d^2*1i + a*c*d*2i - 2*b*c*d)) - (log(tan(e + f*x) + 1i)*(A*1i + B - C*1i
))/(2*f*(a*d^2 - a*c^2 + b*c^2*1i - b*d^2*1i + a*c*d*2i + 2*b*c*d)) + (log(
a + b*tan(e + f*x))*(A*b^3 - B*a*b^2 + C*a^2*b))/(f*(a^4*d^2 + b^4*c^2 + a^
2*b^2*c^2 + a^2*b^2*d^2 - 2*a*b^3*c*d - 2*a^3*b*c*d)) - (log(c + d*tan(e +
f*x))*(d^4*(A*b - B*a) + c^2*d^2*(3*A*b + B*a - C*b) + C*b*c^4 - c*d^3*(2*A
*a - 2*C*a) - 2*B*b*c^3*d))/(f*(a^2*d^6 + b^2*c^6 + 2*a^2*c^2*d^4 + a^2*c^4
*d^2 + b^2*c^2*d^4 + 2*b^2*c^4*d^2 - 2*a*b*c*d^5 - 2*a*b*c^5*d - 4*a*b*c^3*
d^3)) - (A*d^2 + C*c^2 - B*c*d)/(f*(a*d - b*c)*(c^2 + d^2)*(c + d*tan(e + f
*x)))
```

3.82 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^2}\,dx$

Optimal result 841
Rubi [A] (verified) 842
Mathematica [A] (verified) 844
Maple [A] (verified) 845
Fricas [B] (verification not implemented) 846
Sympy [F(-2)] 848
Maxima [B] (verification not implemented) 848
Giac [B] (verification not implemented) 849
Mupad [B] (verification not implemented) 851

Optimal result

Integrand size = 45, antiderivative size = 509

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^2}\,dx =$$
$$-\frac{(a^2(c^2C-2Bcd-Cd^2-A(c^2-d^2))-b^2(c^2C-2Bcd-Cd^2-A(c^2-d^2))+2ab(2c(A-C)d-B(c}{(a^2+b^2)^2(c^2+d^2)^2}$$
$$+\frac{b(3a^3bBd-2a^4Cd+b^4(Bc-2Ad)-a^2b^2(Bc+4Ad)+ab^3(2Ac-2cC+Bd))\log(a\cos(e+fx)+b\,s}{(a^2+b^2)^2(bc-ad)^3f}$$
$$+\frac{d(b(2c^4C-3Bc^3d+4Ac^2d^2-Bcd^3+2Ad^4)-ad^2(2c(A-C)d-B(c^2-d^2)))\log(c\cos(e+fx)+d\,s}{(bc-ad)^3(c^2+d^2)^2f}$$
$$-\frac{d(b^2c(cC-Bd)-abB(c^2+d^2)+a^2(2c^2C-Bcd+Cd^2)+A(a^2d^2+b^2(c^2+2d^2)))}{(a^2+b^2)(bc-ad)^2(c^2+d^2)f(c+d\tan(e+fx))}$$
$$-\frac{Ab^2-a(bB-aC)}{(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))(c+d\tan(e+fx))}$$

```
[Out] -(a^2*(c^2*C-2*B*c*d-C*d^2-A*(c^2-d^2))-b^2*(c^2*C-2*B*c*d-C*d^2-A*(c^2-d^2
))+2*a*b*(2*c*(A-C)*d-B*(c^2-d^2)))*x/(a^2+b^2)^2/(c^2+d^2)^2+b*(3*a^3*b*B*
d-2*a^4*C*d+b^4*(-2*A*d+B*c)-a^2*b^2*(4*A*d+B*c)+a*b^3*(2*A*c+B*d-2*C*c))*l
n(a*cos(f*x+e)+b*sin(f*x+e))/(a^2+b^2)^2/(-a*d+b*c)^3/f+d*(b*(4*A*c^2*d^2+2
*A*d^4-3*B*c^3*d-B*c*d^3+2*C*c^4)-a*d^2*(2*c*(A-C)*d-B*(c^2-d^2)))*ln(c*cos
(f*x+e)+d*sin(f*x+e))/(-a*d+b*c)^3/(c^2+d^2)^2/f-d*(b^2*c*(-B*d+C*c)-a*b*B*
(c^2+d^2)+a^2*(-B*c*d+2*C*c^2+C*d^2)+A*(a^2*d^2+b^2*(c^2+2*d^2)))/(a^2+b^2)
/(-a*d+b*c)^2/(c^2+d^2)/f/(c+d*tan(f*x+e))+(-A*b^2+a*(B*b-C*a))/(a^2+b^2)/(
-a*d+b*c)/f/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))
```

Rubi [A] (verified)

Time = 2.29 (sec) , antiderivative size = 508, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 3, $\frac{\text{number of rules}}{\text{integrand size}} = 0.067$, Rules used = {3730, 3732, 3611}

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^2}\,dx$$
$$= -\frac{d(a^2Ad^2+a^2(-Bcd+2c^2C+Cd^2)-abB(c^2+d^2)+Ab^2(c^2+2d^2)+b^2c(cC-Bd))}{f\,(a^2+b^2)\,(c^2+d^2)\,(bc-ad)^2(c+d\tan(e+fx))}$$
$$-\frac{x(a^2(-A(c^2-d^2)-2Bcd+c^2C-Cd^2)+2ab(2cd(A-C)-B(c^2-d^2))-b^2(-A(c^2-d^2)-2Bcd+}{(a^2+b^2)^2\,(c^2+d^2)^2}$$
$$-\frac{Ab^2-a(bB-aC)}{f\,(a^2+b^2)\,(bc-ad)(a+b\tan(e+fx))(c+d\tan(e+fx))}$$
$$+\frac{b(-2a^4Cd+3a^3bBd-a^2b^2(4Ad+Bc)+ab^3(2Ac+Bd-2cC)+b^4(Bc-2Ad))\log(a\cos(e+fx)+b}{f\,(a^2+b^2)^2\,(bc-ad)^3}$$
$$+\frac{d(b(4Ac^2d^2+2Ad^4-3Bc^3d-Bcd^3+2c^4C)-ad^2(2cd(A-C)-B(c^2-d^2)))\log(c\cos(e+fx)+d\sin}{f\,(c^2+d^2)^2\,(bc-ad)^3}$$

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^2*(c + d*
Tan[e + f*x])^2),x]
```

```
[Out] -(((a^2*(c^2*C - 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - b^2*(c^2*C - 2*B*c*d -
C*d^2 - A*(c^2 - d^2)) + 2*a*b*(2*c*(A - C)*d - B*(c^2 - d^2)))*x)/((a^2 +
b^2)^2*(c^2 + d^2)^2)) + (b*(3*a^3*b*B*d - 2*a^4*C*d + b^4*(B*c - 2*A*d) -
a^2*b^2*(B*c + 4*A*d) + a*b^3*(2*A*c - 2*c*C + B*d))*Log[a*Cos[e + f*x] + b
*Sin[e + f*x]])/((a^2 + b^2)^2*(b*c - a*d)^3*f) + (d*(b*(2*c^4*C - 3*B*c^3*
d + 4*A*c^2*d^2 - B*c*d^3 + 2*A*d^4) - a*d^2*(2*c*(A - C)*d - B*(c^2 - d^2)
))*Log[c*Cos[e + f*x] + d*Sin[e + f*x]])/((b*c - a*d)^3*(c^2 + d^2)^2*f) -
(d*(a^2*A*d^2 + b^2*c*(c*C - B*d) - a*b*B*(c^2 + d^2) + A*b^2*(c^2 + 2*d^2)
 + a^2*(2*c^2*C - B*c*d + C*d^2)))/((a^2 + b^2)*(b*c - a*d)^2*(c^2 + d^2)*f
*(c + d*Tan[e + f*x])) - (A*b^2 - a*(b*B - a*C))/((a^2 + b^2)*(b*c - a*d)*f
*(a + b*Tan[e + f*x])*(c + d*Tan[e + f*x]))
```

Rule 3611

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c/(b*f))*Log[RemoveContent[a*Cos[e + f*x] + b*Si
n[e + f*x], x]], x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3730

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
```

```
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 - a*(b*B - a*C))*(a + b*Tan[e +
 f*x])^(m + 1)*((c + d*Tan[e + f*x])^(n + 1)/(f*(m + 1)*(b*c - a*d)*(a^2 +
b^2))), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3732

```
Int[((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^
2)/(((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])), x_Symbol] :> Simp[(a*(A*c - c*C + B*d) + b*(B*c - A*d + C*d))*(x/
((a^2 + b^2)*(c^2 + d^2))), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/((b*c - a*d)
*(a^2 + b^2)), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] - Dist
[(c^2*C - B*c*d + A*d^2)/((b*c - a*d)*(c^2 + d^2)), Int[(d - c*Tan[e + f*x]
)/(c + d*Tan[e + f*x]), x], x]) /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] &&
 NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0]
```

Rubi steps

integral

$$= -\frac{Ab^2 - a(bB - aC)}{(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))(c+d\tan(e+fx))} - \frac{\int \frac{2Ab^2d - aA(bc-ad) - (bB-aC)(bc+ad) + (Ab-aB-bC)(bc-ad)\tan(e+fx) + 2(Ab^2-a(bB-aC))d\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^2}\,dx}{(a^2+b^2)(bc-ad)}$$

$$= -\frac{d(a^2Ad^2 + b^2c(cC-Bd) - abB(c^2+d^2) + Ab^2(c^2+2d^2) + a^2(2c^2C - Bcd + Cd^2))}{(a^2+b^2)(bc-ad)^2(c^2+d^2)f(c+d\tan(e+fx))} - \frac{Ab^2 - a(bB-aC)}{(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))(c+d\tan(e+fx))} - \frac{\int \frac{-a^3d^2(Ac-cC+Bd) + 2a^2Abd(c^2+d^2) - b^3(Bc-2Ad)(c^2+d^2) + ab^2(c^3C + 2cCd^2 - Bd^3 - A(c^3+2cd^2)) - (bc-ad)^2(bcC - bBd - A(bc+ad) + a}{(a+b\tan(e+fx))(c+d\tan(e-}}{(a^2+b^2)(bc-ad)^2(c^2}$$

$$= -\frac{(a^2(c^2C - 2Bcd - Cd^2 - A(c^2 - d^2)) - b^2(c^2C - 2Bcd - Cd^2 - A(c^2 - d^2)) + 2ab(2c(A - C)d - B(c^2}{(a^2 + b^2)^2 (c^2 + d^2)^2}$$
$$- \frac{d(a^2Ad^2 + b^2c(cC - Bd) - abB(c^2 + d^2) + Ab^2(c^2 + 2d^2) + a^2(2c^2C - Bcd + Cd^2))}{(a^2 + b^2)(bc - ad)^2 (c^2 + d^2) f(c + d\tan(e + fx))}$$
$$- \frac{Ab^2 - a(bB - aC)}{(a^2 + b^2)(bc - ad) f(a + b\tan(e + fx))(c + d\tan(e + fx))}$$
$$+ \frac{(b(3a^3bBd - 2a^4Cd + b^4(Bc - 2Ad) - a^2b^2(Bc + 4Ad) + ab^3(2Ac - 2cC + Bd))) \int \frac{b - a\tan(e+fx)}{a + b\tan(e+fx)}\,dx}{(a^2 + b^2)^2 (bc - ad)^3}$$
$$+ \frac{(d(b(2c^4C - 3Bc^3d + 4Ac^2d^2 - Bcd^3 + 2Ad^4) - ad^2(2c(A - C)d - B(c^2 - d^2)))) \int \frac{d - c\tan(e+fx)}{c + d\tan(e+fx)}\,dx}{(bc - ad)^3 (c^2 + d^2)^2}$$

$$= -\frac{(a^2(c^2C - 2Bcd - Cd^2 - A(c^2 - d^2)) - b^2(c^2C - 2Bcd - Cd^2 - A(c^2 - d^2)) + 2ab(2c(A - C)d - B(c^2}{(a^2 + b^2)^2 (c^2 + d^2)^2}$$
$$+ \frac{b(3a^3bBd - 2a^4Cd + b^4(Bc - 2Ad) - a^2b^2(Bc + 4Ad) + ab^3(2Ac - 2cC + Bd)) \log(a\cos(e + fx) + b\,\mathrm{si}}{(a^2 + b^2)^2 (bc - ad)^3 f}$$
$$+ \frac{d(b(2c^4C - 3Bc^3d + 4Ac^2d^2 - Bcd^3 + 2Ad^4) - ad^2(2c(A - C)d - B(c^2 - d^2))) \log(c\cos(e + fx) + d\,\mathrm{si}}{(bc - ad)^3 (c^2 + d^2)^2 f}$$
$$- \frac{d(a^2Ad^2 + b^2c(cC - Bd) - abB(c^2 + d^2) + Ab^2(c^2 + 2d^2) + a^2(2c^2C - Bcd + Cd^2))}{(a^2 + b^2)(bc - ad)^2 (c^2 + d^2) f(c + d\tan(e + fx))}$$
$$- \frac{Ab^2 - a(bB - aC)}{(a^2 + b^2)(bc - ad) f(a + b\tan(e + fx))(c + d\tan(e + fx))}$$

Mathematica [A] (verified)

Time = 9.12 (sec) , antiderivative size = 984, normalized size of antiderivative = 1.93

$$\int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{(a + b\tan(e + fx))^2 (c + d\tan(e + fx))^2}\,dx$$
$$= -\frac{Ab^2 - a(bB - aC)}{(a^2 + b^2)(bc - ad) f(a + b\tan(e + fx))(c + d\tan(e + fx))}$$
$$- \frac{- \frac{b(bc-ad)^2 \left(2aAbc^2 - a^2Bc^2 + b^2Bc^2 - 2abc^2C + 2a^2Acd - 2Ab^2cd + 4abBcd - 2a^2cCd + 2b^2cCd - 2aAbd^2 + a^2Bd^2 - b^2Bd^2 + 2abCd^2 - \frac{\sqrt{-b^2}\left(a^2\left(c^2C - 2Bcd - Cd\right.\right.}{}\right.}{2(a^2+b^2)(c^2+d^2)}}{}$$

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^2*(
c + d*Tan[e + f*x])^2),x]
```

```
[Out] -((A*b^2 - a*(b*B - a*C))/((a^2 + b^2)*(b*c - a*d)*f*(a + b*Tan[e + f*x])*(
c + d*Tan[e + f*x]))) - (-(((b*(b*c - a*d)^2*(2*a*A*b*c^2 - a^2*B*c^2 + b^2
```

```
*B*c^2 - 2*a*b*c^2*C + 2*a^2*A*c*d - 2*A*b^2*c*d + 4*a*b*B*c*d - 2*a^2*c*C*
d + 2*b^2*c*C*d - 2*a*A*b*d^2 + a^2*B*d^2 - b^2*B*d^2 + 2*a*b*C*d^2 - (Sqrt
[-b^2]*(a^2*(c^2*C - 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - b^2*(c^2*C - 2*B*c*
d - C*d^2 - A*(c^2 - d^2)) + 2*a*b*(2*c*(A - C)*d - B*(c^2 - d^2))))/b)*Log
[Sqrt[-b^2] - b*Tan[e + f*x]])/(2*(a^2 + b^2)*(c^2 + d^2)) - (b^2*(c^2 + d^
2)*(3*a^3*b*B*d - 2*a^4*C*d + b^4*(B*c - 2*A*d) - a^2*b^2*(B*c + 4*A*d) + a
*b^3*(2*A*c - 2*c*C + B*d))*Log[a + b*Tan[e + f*x]])/((a^2 + b^2)*(b*c - a*
d)) + (b*(b*c - a*d)^2*(2*a*A*b*c^2 - a^2*B*c^2 + b^2*B*c^2 - 2*a*b*c^2*C +
 2*a^2*A*c*d - 2*A*b^2*c*d + 4*a*b*B*c*d - 2*a^2*c*C*d + 2*b^2*c*C*d - 2*a*
A*b*d^2 + a^2*B*d^2 - b^2*B*d^2 + 2*a*b*C*d^2 + (Sqrt[-b^2]*(a^2*(c^2*C - 2
*B*c*d - C*d^2 - A*(c^2 - d^2)) - b^2*(c^2*C - 2*B*c*d - C*d^2 - A*(c^2 - d
^2)) + 2*a*b*(2*c*(A - C)*d - B*(c^2 - d^2))))/b)*Log[Sqrt[-b^2] + b*Tan[e
+ f*x]])/(2*(a^2 + b^2)*(c^2 + d^2)) - (b*(a^2 + b^2)*d*(b*(2*c^4*C - 3*B*c
^3*d + 4*A*c^2*d^2 - B*c*d^3 + 2*A*d^4) - a*d^2*(2*c*(A - C)*d - B*(c^2 - d
^2)))*Log[c + d*Tan[e + f*x]])/((b*c - a*d)*(c^2 + d^2)))/(b*(-(b*c) + a*d)
*(c^2 + d^2)*f)) - (-(c*(-2*c*(A*b^2 - a*(b*B - a*C))*d + (A*b - a*B - b*C)
*d*(b*c - a*d))) + d^2*(2*A*b^2*d - a*A*(b*c - a*d) - (b*B - a*C)*(b*c + a*
d)))/((-(b*c) + a*d)*(c^2 + d^2)*f*(c + d*Tan[e + f*x])))/((a^2 + b^2)*(b*c
 - a*d))
```

Maple [A] (verified)

Time = 1.81 (sec) , antiderivative size = 577, normalized size of antiderivative = 1.13

method	result
derivativedivides	$\frac{\frac{b\left(4A\,a^2b^2d-2Aa\,b^3c+2A\,b^4d-3a^3bBd+B\,a^2b^2c-Ba\,b^3d-B\,b^4c+2a^4Cd+2Ca\,b^3c\right)\ln(a+b\tan(fx+e))}{(ad-bc)^3\left(a^2+b^2\right)^2}-\frac{\left(A\,b^2-Bab+C\,a^2\right)}{(ad-bc)^2\left(a^2+b^2\right)(a+b\tan}}{}$
default	$\frac{\frac{b\left(4A\,a^2b^2d-2Aa\,b^3c+2A\,b^4d-3a^3bBd+B\,a^2b^2c-Ba\,b^3d-B\,b^4c+2a^4Cd+2Ca\,b^3c\right)\ln(a+b\tan(fx+e))}{(ad-bc)^3\left(a^2+b^2\right)^2}-\frac{\left(A\,b^2-Bab+C\,a^2\right)}{(ad-bc)^2\left(a^2+b^2\right)(a+b\tan}}{}$
norman	Expression too large to display
risch	Expression too large to display
parallelrisch	Expression too large to display

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e))^2,x
,method=_RETURNVERBOSE)
```

```
[Out] 1/f*(b*(4*A*a^2*b^2*d-2*A*a*b^3*c+2*A*b^4*d-3*B*a^3*b*d+B*a^2*b^2*c-B*a*b^3
*d-B*b^4*c+2*C*a^4*d+2*C*a*b^3*c)/(a*d-b*c)^3/(a^2+b^2)^2*ln(a+b*tan(f*x+e)
)-(A*b^2-B*a*b+C*a^2)*b/(a*d-b*c)^2/(a^2+b^2)/(a+b*tan(f*x+e))+1/(a^2+b^2)^
2/(c^2+d^2)^2*(1/2*(-2*A*a^2*c*d-2*A*a*b*c^2+2*A*a*b*d^2+2*A*b^2*c*d+B*a^2*
c^2-B*a^2*d^2-4*B*a*b*c*d-B*b^2*c^2+B*b^2*d^2+2*C*a^2*c*d+2*C*a*b*c^2-2*C*a
*b*d^2-2*C*b^2*c*d)*ln(1+tan(f*x+e)^2)+(A*a^2*c^2-A*a^2*d^2-4*A*a*b*c*d-A*b
^2*c^2+A*b^2*d^2+2*B*a^2*c*d+2*B*a*b*c^2-2*B*a*b*d^2-2*B*b^2*c*d-C*a^2*c^2+
```

```
C*a^2*d^2+4*C*a*b*c*d+C*b^2*c^2-C*b^2*d^2)*arctan(tan(f*x+e)))+d*(2*A*a*c*d
^3-4*A*b*c^2*d^2-2*A*b*d^4-B*a*c^2*d^2+B*a*d^4+3*B*b*c^3*d+B*b*c*d^3-2*C*a*
c*d^3-2*C*b*c^4)/(a*d-b*c)^3/(c^2+d^2)^2*ln(c+d*tan(f*x+e))-(A*d^2-B*c*d+C*
c^2)*d/(a*d-b*c)^2/(c^2+d^2)/(c+d*tan(f*x+e)))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 4174 vs. $2(512) = 1024$.

Time = 3.54 (sec) , antiderivative size = 4174, normalized size of antiderivative = 8.20

$$\int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{(a + b\tan(e + fx))^2(c + d\tan(e + fx))^2}\,dx = \text{Too large to display}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e
))^2,x, algorithm="fricas")
```

```
[Out] -1/2*(2*(C*a^2*b^4 - B*a*b^5 + A*b^6)*c^6 - 2*(C*a^3*b^3 - B*a^2*b^4 + A*a*
b^5)*c^5*d + 4*(C*a^2*b^4 - B*a*b^5 + A*b^6)*c^4*d^2 + 2*(C*a^5*b + 2*B*a^2
*b^4 - (2*A - C)*a*b^5)*c^3*d^3 - 2*(C*a^6 + B*a^5*b + 2*C*a^4*b^2 + 2*B*a^
3*b^3 + 2*B*a*b^5 - A*b^6)*c^2*d^4 + 2*(B*a^6 + A*a^5*b + 2*B*a^4*b^2 + (2*
A - C)*a^3*b^3 + 2*B*a^2*b^4)*c*d^5 - 2*(A*a^6 + 2*A*a^4*b^2 + A*a^2*b^4)*d
^6 - 2*(((A - C)*a^3*b^3 + 2*B*a^2*b^4 - (A - C)*a*b^5)*c^6 - (3*(A - C)*a^
4*b^2 + 4*B*a^3*b^3 + (A - C)*a^2*b^4 + 2*B*a*b^5)*c^5*d + (3*(A - C)*a^5*b
 + 8*(A - C)*a^3*b^3 + 4*B*a^2*b^4 + (A - C)*a*b^5)*c^4*d^2 - ((A - C)*a^6
 - 4*B*a^5*b + 8*(A - C)*a^4*b^2 + 3*(A - C)*a^2*b^4)*c^3*d^3 - (2*B*a^6 - (
A - C)*a^5*b + 4*B*a^4*b^2 - 3*(A - C)*a^3*b^3)*c^2*d^4 + ((A - C)*a^6 + 2*
B*a^5*b - (A - C)*a^4*b^2)*c*d^5)*f*x - 2*((C*a^3*b^3 - B*a^2*b^4 + A*a*b^5
)*c^5*d + (B*a^3*b^3 - (A - 2*C)*a^2*b^4 + C*b^6)*c^4*d^2 - (C*a^5*b + B*a^
4*b^2 + 4*B*a^2*b^4 - (2*A - C)*a*b^5 + B*b^6)*c^3*d^3 + (B*a^5*b + (A - 2*
C)*a^4*b^2 + 4*B*a^3*b^3 + B*a*b^5 + A*b^6)*c^2*d^4 - (A*a^5*b + (2*A - C)*
a^3*b^3 + B*a^2*b^4)*c*d^5 - (C*a^4*b^2 - B*a^3*b^3 + A*a^2*b^4)*d^6 + (((A
 - C)*a^2*b^4 + 2*B*a*b^5 - (A - C)*b^6)*c^5*d - (3*(A - C)*a^3*b^3 + 4*B*a
^2*b^4 + (A - C)*a*b^5 + 2*B*b^6)*c^4*d^2 + (3*(A - C)*a^4*b^2 + 8*(A - C)*
a^2*b^4 + 4*B*a*b^5 + (A - C)*b^6)*c^3*d^3 - ((A - C)*a^5*b - 4*B*a^4*b^2 +
 8*(A - C)*a^3*b^3 + 3*(A - C)*a*b^5)*c^2*d^4 - (2*B*a^5*b - (A - C)*a^4*b^
2 + 4*B*a^3*b^3 - 3*(A - C)*a^2*b^4)*c*d^5 + ((A - C)*a^5*b + 2*B*a^4*b^2 -
 (A - C)*a^3*b^3)*d^6)*f*x)*tan(f*x + e)^2 + ((B*a^3*b^3 - 2*(A - C)*a^2*b^
4 - B*a*b^5)*c^6 + (2*C*a^5*b - 3*B*a^4*b^2 + 4*A*a^3*b^3 - B*a^2*b^4 + 2*A
*a*b^5)*c^5*d + 2*(B*a^3*b^3 - 2*(A - C)*a^2*b^4 - B*a*b^5)*c^4*d^2 + 2*(2*
C*a^5*b - 3*B*a^4*b^2 + 4*A*a^3*b^3 - B*a^2*b^4 + 2*A*a*b^5)*c^3*d^3 + (B*a
^3*b^3 - 2*(A - C)*a^2*b^4 - B*a*b^5)*c^2*d^4 + (2*C*a^5*b - 3*B*a^4*b^2 +
4*A*a^3*b^3 - B*a^2*b^4 + 2*A*a*b^5)*c*d^5 + ((B*a^2*b^4 - 2*(A - C)*a*b^5
- B*b^6)*c^5*d + (2*C*a^4*b^2 - 3*B*a^3*b^3 + 4*A*a^2*b^4 - B*a*b^5 + 2*A*b
^6)*c^4*d^2 + 2*(B*a^2*b^4 - 2*(A - C)*a*b^5 - B*b^6)*c^3*d^3 + 2*(2*C*a^4*
b^2 - 3*B*a^3*b^3 + 4*A*a^2*b^4 - B*a*b^5 + 2*A*b^6)*c^2*d^4 + (B*a^2*b^4 -
```

2*(A - C)*a*b^5 - B*b^6)*c*d^5 + (2*C*a^4*b^2 - 3*B*a^3*b^3 + 4*A*a^2*b^4
- B*a*b^5 + 2*A*b^6)*d^6)*tan(f*x + e)^2 + ((B*a^2*b^4 - 2*(A - C)*a*b^5 -
B*b^6)*c^6 + 2*(C*a^4*b^2 - B*a^3*b^3 + (A + C)*a^2*b^4 - B*a*b^5 + A*b^6)*
c^5*d + (2*C*a^5*b - 3*B*a^4*b^2 + 4*A*a^3*b^3 + B*a^2*b^4 - 2*(A - 2*C)*a*
b^5 - 2*B*b^6)*c^4*d^2 + 4*(C*a^4*b^2 - B*a^3*b^3 + (A + C)*a^2*b^4 - B*a*b
^5 + A*b^6)*c^3*d^3 + (4*C*a^5*b - 6*B*a^4*b^2 + 8*A*a^3*b^3 - B*a^2*b^4 +
2*(A + C)*a*b^5 - B*b^6)*c^2*d^4 + 2*(C*a^4*b^2 - B*a^3*b^3 + (A + C)*a^2*b
^4 - B*a*b^5 + A*b^6)*c*d^5 + (2*C*a^5*b - 3*B*a^4*b^2 + 4*A*a^3*b^3 - B*a^
2*b^4 + 2*A*a*b^5)*d^6)*tan(f*x + e))*log((b^2*tan(f*x + e)^2 + 2*a*b*tan(f
*x + e) + a^2)/(tan(f*x + e)^2 + 1)) - (2*(C*a^5*b + 2*C*a^3*b^3 + C*a*b^5)
*c^5*d - 3*(B*a^5*b + 2*B*a^3*b^3 + B*a*b^5)*c^4*d^2 + (B*a^6 + 4*A*a^5*b +
2*B*a^4*b^2 + 8*A*a^3*b^3 + B*a^2*b^4 + 4*A*a*b^5)*c^3*d^3 - (2*(A - C)*a^
6 + B*a^5*b + 4*(A - C)*a^4*b^2 + 2*B*a^3*b^3 + 2*(A - C)*a^2*b^4 + B*a*b^5
)*c^2*d^4 - (B*a^6 - 2*A*a^5*b + 2*B*a^4*b^2 - 4*A*a^3*b^3 + B*a^2*b^4 - 2*
A*a*b^5)*c*d^5 + (2*(C*a^4*b^2 + 2*C*a^2*b^4 + C*b^6)*c^4*d^2 - 3*(B*a^4*b^
2 + 2*B*a^2*b^4 + B*b^6)*c^3*d^3 + (B*a^5*b + 4*A*a^4*b^2 + 2*B*a^3*b^3 + 8
*A*a^2*b^4 + B*a*b^5 + 4*A*b^6)*c^2*d^4 - (2*(A - C)*a^5*b + B*a^4*b^2 + 4*
(A - C)*a^3*b^3 + 2*B*a^2*b^4 + 2*(A - C)*a*b^5 + B*b^6)*c*d^5 - (B*a^5*b -
2*A*a^4*b^2 + 2*B*a^3*b^3 - 4*A*a^2*b^4 + B*a*b^5 - 2*A*b^6)*d^6)*tan(f*x
+ e)^2 + (2*(C*a^4*b^2 + 2*C*a^2*b^4 + C*b^6)*c^5*d + (2*C*a^5*b - 3*B*a^4*
b^2 + 4*C*a^3*b^3 - 6*B*a^2*b^4 + 2*C*a*b^5 - 3*B*b^6)*c^4*d^2 - 2*(B*a^5*b
- 2*A*a^4*b^2 + 2*B*a^3*b^3 - 4*A*a^2*b^4 + B*a*b^5 - 2*A*b^6)*c^3*d^3 + (
B*a^6 + 2*(A + C)*a^5*b + B*a^4*b^2 + 4*(A + C)*a^3*b^3 - B*a^2*b^4 + 2*(A
+ C)*a*b^5 - B*b^6)*c^2*d^4 - 2*((A - C)*a^6 + B*a^5*b + (A - 2*C)*a^4*b^2
+ 2*B*a^3*b^3 - (A + C)*a^2*b^4 + B*a*b^5 - A*b^6)*c*d^5 - (B*a^6 - 2*A*a^5
*b + 2*B*a^4*b^2 - 4*A*a^3*b^3 + B*a^2*b^4 - 2*A*a*b^5)*d^6)*tan(f*x + e))*
log((d^2*tan(f*x + e)^2 + 2*c*d*tan(f*x + e) + c^2)/(tan(f*x + e)^2 + 1)) -
2*((C*a^3*b^3 - B*a^2*b^4 + A*a*b^5)*c^6 - (C*a^4*b^2 - B*a^3*b^3 + (A + C
)*a^2*b^4 - B*a*b^5 + A*b^6)*c^5*d + (C*a^5*b + 5*C*a^3*b^3 - 3*B*a^2*b^4 +
(3*A + C)*a*b^5)*c^4*d^2 - (C*a^6 + B*a^5*b + 5*C*a^4*b^2 + (2*A + 5*C)*a^
2*b^4 - B*a*b^5 + (2*A + C)*b^6)*c^3*d^3 + (B*a^6 + (A + C)*a^5*b + 3*B*a^4
*b^2 + (2*A + 5*C)*a^3*b^3 + (4*A + C)*a*b^5 + B*b^6)*c^2*d^4 - (A*a^6 + B*
a^5*b + (3*A + C)*a^4*b^2 + B*a^3*b^3 + (4*A + C)*a^2*b^4 + 2*A*b^6)*c*d^5
+ (A*a^5*b + (2*A + C)*a^3*b^3 - B*a^2*b^4 + 2*A*a*b^5)*d^6 + (((A - C)*a^2
*b^4 + 2*B*a*b^5 - (A - C)*b^6)*c^6 - 2*((A - C)*a^3*b^3 + B*a^2*b^4 + (A -
C)*a*b^5 + B*b^6)*c^5*d - (4*B*a^3*b^3 - 7*(A - C)*a^2*b^4 - 2*B*a*b^5 - (
A - C)*b^6)*c^4*d^2 + 2*((A - C)*a^5*b + 2*B*a^4*b^2 + 2*B*a^2*b^4 - (A - C
)*a*b^5)*c^3*d^3 - ((A - C)*a^6 - 2*B*a^5*b + 7*(A - C)*a^4*b^2 + 4*B*a^3*b
^3)*c^2*d^4 - 2*(B*a^6 - (A - C)*a^5*b + B*a^4*b^2 - (A - C)*a^3*b^3)*c*d^5
+ ((A - C)*a^6 + 2*B*a^5*b - (A - C)*a^4*b^2)*d^6)*f*x)*tan(f*x + e))/(((a
^4*b^4 + 2*a^2*b^6 + b^8)*c^7*d - 3*(a^5*b^3 + 2*a^3*b^5 + a*b^7)*c^6*d^2 +
(3*a^6*b^2 + 8*a^4*b^4 + 7*a^2*b^6 + 2*b^8)*c^5*d^3 - (a^7*b + 8*a^5*b^3 +
13*a^3*b^5 + 6*a*b^7)*c^4*d^4 + (6*a^6*b^2 + 13*a^4*b^4 + 8*a^2*b^6 + b^8)
*c^3*d^5 - (2*a^7*b + 7*a^5*b^3 + 8*a^3*b^5 + 3*a*b^7)*c^2*d^6 + 3*(a^6*b^2
+ 2*a^4*b^4 + a^2*b^6)*c*d^7 - (a^7*b + 2*a^5*b^3 + a^3*b^5)*d^8)*f*tan(f*

```
x + e)^2 + ((a^4*b^4 + 2*a^2*b^6 + b^8)*c^8 - 2*(a^5*b^3 + 2*a^3*b^5 + a*b^
7)*c^7*d + 2*(a^4*b^4 + 2*a^2*b^6 + b^8)*c^6*d^2 + 2*(a^7*b - 3*a^3*b^5 - 2
*a*b^7)*c^5*d^3 - (a^8 + 2*a^6*b^2 - 2*a^2*b^6 - b^8)*c^4*d^4 + 2*(2*a^7*b
+ 3*a^5*b^3 - a*b^7)*c^3*d^5 - 2*(a^8 + 2*a^6*b^2 + a^4*b^4)*c^2*d^6 + 2*(a
^7*b + 2*a^5*b^3 + a^3*b^5)*c*d^7 - (a^8 + 2*a^6*b^2 + a^4*b^4)*d^8)*f*tan(
f*x + e) + ((a^5*b^3 + 2*a^3*b^5 + a*b^7)*c^8 - 3*(a^6*b^2 + 2*a^4*b^4 + a^
2*b^6)*c^7*d + (3*a^7*b + 8*a^5*b^3 + 7*a^3*b^5 + 2*a*b^7)*c^6*d^2 - (a^8 +
 8*a^6*b^2 + 13*a^4*b^4 + 6*a^2*b^6)*c^5*d^3 + (6*a^7*b + 13*a^5*b^3 + 8*a^
3*b^5 + a*b^7)*c^4*d^4 - (2*a^8 + 7*a^6*b^2 + 8*a^4*b^4 + 3*a^2*b^6)*c^3*d^
5 + 3*(a^7*b + 2*a^5*b^3 + a^3*b^5)*c^2*d^6 - (a^8 + 2*a^6*b^2 + a^4*b^4)*c
*d^7)*f)
```

Sympy [F(-2)]

Exception generated.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^2}\,dx = \text{Exception raised: NotImplementedError}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e))**2/(c+d*tan(f*x
+e))**2,x)
```

```
[Out] Exception raised: NotImplementedError >> no valid subset found
```

Maxima [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1185 vs. $2(512) = 1024$.

Time = 0.41 (sec) , antiderivative size = 1185, normalized size of antiderivative = 2.33

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^2}\,dx = \text{Too large to display}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e
))^2,x, algorithm="maxima")
```

```
[Out] 1/2*(2*(((A - C)*a^2 + 2*B*a*b - (A - C)*b^2)*c^2 + 2*(B*a^2 - 2*(A - C)*a*
b - B*b^2)*c*d - ((A - C)*a^2 + 2*B*a*b - (A - C)*b^2)*d^2)*(f*x + e)/((a^4
 + 2*a^2*b^2 + b^4)*c^4 + 2*(a^4 + 2*a^2*b^2 + b^4)*c^2*d^2 + (a^4 + 2*a^2*
b^2 + b^4)*d^4) - 2*((B*a^2*b^3 - 2*(A - C)*a*b^4 - B*b^5)*c + (2*C*a^4*b -
 3*B*a^3*b^2 + 4*A*a^2*b^3 - B*a*b^4 + 2*A*b^5)*d)*log(b*tan(f*x + e) + a)/
((a^4*b^3 + 2*a^2*b^5 + b^7)*c^3 - 3*(a^5*b^2 + 2*a^3*b^4 + a*b^6)*c^2*d +
3*(a^6*b + 2*a^4*b^3 + a^2*b^5)*c*d^2 - (a^7 + 2*a^5*b^2 + a^3*b^4)*d^3) +
2*(2*C*b*c^4*d - 3*B*b*c^3*d^2 + (B*a + 4*A*b)*c^2*d^3 - (2*(A - C)*a + B*b
)*c*d^4 - (B*a - 2*A*b)*d^5)*log(d*tan(f*x + e) + c)/(b^3*c^7 - 3*a*b^2*c^6
```

```
*d + 3*a^2*b*c*d^6 - a^3*d^7 + (3*a^2*b + 2*b^3)*c^5*d^2 - (a^3 + 6*a*b^2)*
c^4*d^3 + (6*a^2*b + b^3)*c^3*d^4 - (2*a^3 + 3*a*b^2)*c^2*d^5) + ((B*a^2 -
2*(A - C)*a*b - B*b^2)*c^2 - 2*((A - C)*a^2 + 2*B*a*b - (A - C)*b^2)*c*d -
(B*a^2 - 2*(A - C)*a*b - B*b^2)*d^2)*log(tan(f*x + e)^2 + 1)/((a^4 + 2*a^2*
b^2 + b^4)*c^4 + 2*(a^4 + 2*a^2*b^2 + b^4)*c^2*d^2 + (a^4 + 2*a^2*b^2 + b^4
)*d^4) - 2*((C*a^2*b - B*a*b^2 + A*b^3)*c^3 + (C*a^3 + C*a*b^2)*c^2*d - (B*
a^3 - C*a^2*b + 2*B*a*b^2 - A*b^3)*c*d^2 + (A*a^3 + A*a*b^2)*d^3 + ((2*C*a^
2*b - B*a*b^2 + (A + C)*b^3)*c^2*d - (B*a^2*b + B*b^3)*c*d^2 + ((A + C)*a^2
*b - B*a*b^2 + 2*A*b^3)*d^3)*tan(f*x + e))/((a^3*b^2 + a*b^4)*c^5 - 2*(a^4*
b + a^2*b^3)*c^4*d + (a^5 + 2*a^3*b^2 + a*b^4)*c^3*d^2 - 2*(a^4*b + a^2*b^3
)*c^2*d^3 + (a^5 + a^3*b^2)*c*d^4 + ((a^2*b^3 + b^5)*c^4*d - 2*(a^3*b^2 + a
*b^4)*c^3*d^2 + (a^4*b + 2*a^2*b^3 + b^5)*c^2*d^3 - 2*(a^3*b^2 + a*b^4)*c*d
^4 + (a^4*b + a^2*b^3)*d^5)*tan(f*x + e)^2 + ((a^2*b^3 + b^5)*c^5 - (a^3*b^
2 + a*b^4)*c^4*d - (a^4*b - b^5)*c^3*d^2 + (a^5 - a*b^4)*c^2*d^3 - (a^4*b +
 a^2*b^3)*c*d^4 + (a^5 + a^3*b^2)*d^5)*tan(f*x + e)))/f
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 2823 vs. $2(512) = 1024$.

Time = 1.07 (sec) , antiderivative size = 2823, normalized size of antiderivative = 5.55

$$\int \frac{A + B\tan(e+fx) + C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^2}\,dx = \text{Too large to display}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e
))^2,x, algorithm="giac")
```

```
[Out] 1/2*(2*(A*a^2*c^2 - C*a^2*c^2 + 2*B*a*b*c^2 - A*b^2*c^2 + C*b^2*c^2 + 2*B*a
^2*c*d - 4*A*a*b*c*d + 4*C*a*b*c*d - 2*B*b^2*c*d - A*a^2*d^2 + C*a^2*d^2 -
2*B*a*b*d^2 + A*b^2*d^2 - C*b^2*d^2)*(f*x + e)/(a^4*c^4 + 2*a^2*b^2*c^4 + b
^4*c^4 + 2*a^4*c^2*d^2 + 4*a^2*b^2*c^2*d^2 + 2*b^4*c^2*d^2 + a^4*d^4 + 2*a^
2*b^2*d^4 + b^4*d^4) + (B*a^2*c^2 - 2*A*a*b*c^2 + 2*C*a*b*c^2 - B*b^2*c^2 -
 2*A*a^2*c*d + 2*C*a^2*c*d - 4*B*a*b*c*d + 2*A*b^2*c*d - 2*C*b^2*c*d - B*a^
2*d^2 + 2*A*a*b*d^2 - 2*C*a*b*d^2 + B*b^2*d^2)*log(tan(f*x + e)^2 + 1)/(a^4
*c^4 + 2*a^2*b^2*c^4 + b^4*c^4 + 2*a^4*c^2*d^2 + 4*a^2*b^2*c^2*d^2 + 2*b^4*
c^2*d^2 + a^4*d^4 + 2*a^2*b^2*d^4 + b^4*d^4) - 2*(B*a^2*b^4*c - 2*A*a*b^5*c
 + 2*C*a*b^5*c - B*b^6*c + 2*C*a^4*b^2*d - 3*B*a^3*b^3*d + 4*A*a^2*b^4*d -
B*a*b^5*d + 2*A*b^6*d)*log(abs(b*tan(f*x + e) + a))/(a^4*b^4*c^3 + 2*a^2*b^
6*c^3 + b^8*c^3 - 3*a^5*b^3*c^2*d - 6*a^3*b^5*c^2*d - 3*a*b^7*c^2*d + 3*a^6
*b^2*c*d^2 + 6*a^4*b^4*c*d^2 + 3*a^2*b^6*c*d^2 - a^7*b*d^3 - 2*a^5*b^3*d^3
- a^3*b^5*d^3) + 2*(2*C*b*c^4*d^2 - 3*B*b*c^3*d^3 + B*a*c^2*d^4 + 4*A*b*c^2
*d^4 - 2*A*a*c*d^5 + 2*C*a*c*d^5 - B*b*c*d^5 - B*a*d^6 + 2*A*b*d^6)*log(abs
(d*tan(f*x + e) + c))/(b^3*c^7*d - 3*a*b^2*c^6*d^2 + 3*a^2*b*c^5*d^3 + 2*b^
3*c^5*d^3 - a^3*c^4*d^4 - 6*a*b^2*c^4*d^4 + 6*a^2*b*c^3*d^5 + b^3*c^3*d^5 -
 2*a^3*c^2*d^6 - 3*a*b^2*c^2*d^6 + 3*a^2*b*c*d^7 - a^3*d^8) + (B*a^2*b^3*c^
```

4*d*tan(f*x + e)^2 - 2*A*a*b^4*c^4*d*tan(f*x + e)^2 + 2*C*a*b^4*c^4*d*tan(f*x + e)^2 - B*b^5*c^4*d*tan(f*x + e)^2 - 2*B*a^3*b^2*c^3*d^2*tan(f*x + e)^2 + 2*A*a^2*b^3*c^3*d^2*tan(f*x + e)^2 - 2*C*a^2*b^3*c^3*d^2*tan(f*x + e)^2 - 2*B*a*b^4*c^3*d^2*tan(f*x + e)^2 + 2*A*b^5*c^3*d^2*tan(f*x + e)^2 - 2*C*b^5*c^3*d^2*tan(f*x + e)^2 + B*a^4*b*c^2*d^3*tan(f*x + e)^2 + 2*A*a^3*b^2*c^2*d^3*tan(f*x + e)^2 - 2*C*a^3*b^2*c^2*d^3*tan(f*x + e)^2 + 6*B*a^2*b^3*c^2*d^3*tan(f*x + e)^2 - 2*A*a*b^4*c^2*d^3*tan(f*x + e)^2 + 2*C*a*b^4*c^2*d^3*tan(f*x + e)^2 + B*b^5*c^2*d^3*tan(f*x + e)^2 - 2*A*a^4*b*c*d^4*tan(f*x + e)^2 + 2*C*a^4*b*c*d^4*tan(f*x + e)^2 - 2*B*a^3*b^2*c*d^4*tan(f*x + e)^2 - 2*A*a^2*b^3*c*d^4*tan(f*x + e)^2 + 2*C*a^2*b^3*c*d^4*tan(f*x + e)^2 - 2*B*a*b^4*c*d^4*tan(f*x + e)^2 - B*a^4*b*d^5*tan(f*x + e)^2 + 2*A*a^3*b^2*d^5*tan(f*x + e)^2 - 2*C*a^3*b^2*d^5*tan(f*x + e)^2 + B*a^2*b^3*d^5*tan(f*x + e)^2 + B*a^2*b^3*c^5*tan(f*x + e) - 2*A*a*b^4*c^5*tan(f*x + e) + 2*C*a*b^4*c^5*tan(f*x + e) - B*b^5*c^5*tan(f*x + e) - 4*C*a^4*b*c^4*d*tan(f*x + e) + B*a^3*b^2*c^4*d*tan(f*x + e) - 2*A*a^2*b^3*c^4*d*tan(f*x + e) - 6*C*a^2*b^3*c^4*d*tan(f*x + e) - B*a*b^4*c^4*d*tan(f*x + e) - 4*C*b^5*c^4*d*tan(f*x + e) + B*a^4*b*c^3*d^2*tan(f*x + e) + 4*A*a^3*b^2*c^3*d^2*tan(f*x + e) - 4*C*a^3*b^2*c^3*d^2*tan(f*x + e) + 8*B*a^2*b^3*c^3*d^2*tan(f*x + e) + 3*B*b^5*c^3*d^2*tan(f*x + e) + B*a^5*c^2*d^3*tan(f*x + e) - 2*A*a^4*b*c^2*d^3*tan(f*x + e) - 6*C*a^4*b*c^2*d^3*tan(f*x + e) + 8*B*a^3*b^2*c^2*d^3*tan(f*x + e) - 12*A*a^2*b^3*c^2*d^3*tan(f*x + e) - 4*C*a^2*b^3*c^2*d^3*tan(f*x + e) + 3*B*a*b^4*c^2*d^3*tan(f*x + e) - 6*A*b^5*c^2*d^3*tan(f*x + e) - 2*C*b^5*c^2*d^3*tan(f*x + e) - 2*A*a^5*c*d^4*tan(f*x + e) + 2*C*a^5*c*d^4*tan(f*x + e) - B*a^4*b*c*d^4*tan(f*x + e) + 3*B*a^2*b^3*c*d^4*tan(f*x + e) + 2*B*b^5*c*d^4*tan(f*x + e) - B*a^5*d^5*tan(f*x + e) - 4*C*a^4*b*d^5*tan(f*x + e) + 3*B*a^3*b^2*d^5*tan(f*x + e) - 6*A*a^2*b^3*d^5*tan(f*x + e) - 2*C*a^2*b^3*d^5*tan(f*x + e) + 2*B*a*b^4*d^5*tan(f*x + e) - 4*A*b^5*d^5*tan(f*x + e) - 2*C*a^4*b*c^5 + 3*B*a^3*b^2*c^5 - 4*A*a^2*b^3*c^5 + B*a*b^4*c^5 - 2*A*b^5*c^5 - 2*C*a^5*c^4*d - 2*B*a^4*b*c^4*d + 2*A*a^3*b^2*c^4*d - 6*C*a^3*b^2*c^4*d - 2*B*a^2*b^3*c^4*d + 2*A*a*b^4*c^4*d - 4*C*a*b^4*c^4*d + 3*B*a^5*c^3*d^2 + 2*A*a^4*b*c^3*d^2 - 6*C*a^4*b*c^3*d^2 + 14*B*a^3*b^2*c^3*d^2 - 6*A*a^2*b^3*c^3*d^2 - 2*C*a^2*b^3*c^3*d^2 + 7*B*a*b^4*c^3*d^2 - 4*A*b^5*c^3*d^2 - 4*A*a^5*c^2*d^3 - 2*B*a^4*b*c^2*d^3 - 6*A*a^3*b^2*c^2*d^3 - 2*C*a^3*b^2*c^2*d^3 - 2*B*a^2*b^3*c^2*d^3 - 2*A*a*b^4*c^2*d^3 - 2*C*a*b^4*c^2*d^3 + B*a^5*c*d^4 + 2*A*a^4*b*c*d^4 - 4*C*a^4*b*c*d^4 + 7*B*a^3*b^2*c*d^4 - 2*A*a^2*b^3*c*d^4 - 2*C*a^2*b^3*c*d^4 + 4*B*a*b^4*c*d^4 - 2*A*b^5*c*d^4 - 2*A*a^5*d^5 - 4*A*a^3*b^2*d^5 - 2*A*a*b^4*d^5)/((a^4*b^2*c^6 + 2*a^2*b^4*c^6 + b^6*c^6 - 2*a^5*b*c^5*d - 4*a^3*b^3*c^5*d - 2*a*b^5*c^5*d + a^6*c^4*d^2 + 4*a^4*b^2*c^4*d^2 + 5*a^2*b^4*c^4*d^2 + 2*b^6*c^4*d^2 - 4*a^5*b*c^3*d^3 - 8*a^3*b^3*c^3*d^3 - 4*a*b^5*c^3*d^3 + 2*a^6*c^2*d^4 + 5*a^4*b^2*c^2*d^4 + 4*a^2*b^4*c^2*d^4 + b^6*c^2*d^4 - 2*a^5*b*c*d^5 - 4*a^3*b^3*c*d^5 - 2*a*b^5*c*d^5 + a^6*d^6 + 2*a^4*b^2*d^6 + a^2*b^4*d^6)*(b*d*tan(f*x + e)^2 + b*c*tan(f*x + e) + a*d*tan(f*x + e) + a*c)))/f

Mupad [B] (verification not implemented)

Time = 25.22 (sec) , antiderivative size = 73684, normalized size of antiderivative = 144.76

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^2}\,dx = \text{Too large to display}$$

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))^2*(c + d*
tan(e + f*x))^2),x)
```

```
[Out] (symsum(log((tan(e + f*x)*(4*A^3*a^3*b^4*d^7 + B^3*a^2*b^5*d^7 + 4*A^3*b^7*
c^3*d^4 + 2*C^3*a^5*b^2*d^7 + B^3*b^7*c^2*d^5 + 2*C^3*b^7*c^5*d^2 + 4*A^2*B
*b^7*d^7 - 4*B^3*a^2*b^5*c^2*d^5 - 3*B^3*a^2*b^5*c^4*d^3 + 10*B^3*a^3*b^4*c
^3*d^4 - 3*B^3*a^4*b^3*c^2*d^5 - 4*A*B^2*a*b^6*d^7 - 4*A*B^2*b^7*c*d^6 + 2*
B^3*a*b^6*c*d^6 - 6*A*B^2*a^3*b^4*d^7 + 8*A^2*B*a^2*b^5*d^7 - 3*A^2*B*a^4*b
^3*d^7 + 4*A*C^2*a^3*b^4*d^7 - 4*A*C^2*a^5*b^2*d^7 - 8*A^2*C*a^3*b^4*d^7 +
2*A^2*C*a^5*b^2*d^7 - 6*A*B^2*b^7*c^3*d^4 - 3*B*C^2*a^4*b^3*d^7 + 8*A^2*B*b
^7*c^2*d^5 - 3*A^2*B*b^7*c^4*d^3 + 4*A*C^2*b^7*c^3*d^4 - 4*A*C^2*b^7*c^5*d^
2 - 8*A^2*C*b^7*c^3*d^4 + 2*A^2*C*b^7*c^5*d^2 - 3*B*C^2*b^7*c^4*d^3 - 4*A^3
*a*b^6*c^2*d^5 - 4*A^3*a^2*b^5*c*d^6 + 6*B^3*a*b^6*c^3*d^4 + 6*B^3*a^3*b^4*
c*d^6 - 2*C^3*a*b^6*c^4*d^3 - 2*C^3*a^4*b^3*c*d^6 - 10*A*B^2*a^2*b^5*c^3*d^
4 - 10*A*B^2*a^3*b^4*c^2*d^5 + 18*A^2*B*a^2*b^5*c^2*d^5 + 2*B*C^2*a^2*b^5*c
^2*d^5 + 4*B*C^2*a^4*b^3*c^4*d^3 + 2*B^2*C*a^2*b^5*c^3*d^4 + 2*B^2*C*a^2*b^
5*c^5*d^2 + 2*B^2*C*a^3*b^4*c^2*d^5 - 6*B^2*C*a^3*b^4*c^4*d^3 - 6*B^2*C*a^4
*b^3*c^3*d^4 + 2*B^2*C*a^5*b^2*c^2*d^5 + 10*A*B*C*a^4*b^3*d^7 + 10*A*B*C*b^
7*c^4*d^3 - 8*A^2*B*a*b^6*c*d^6 - 2*A*B^2*a*b^6*c^2*d^5 + 6*A*B^2*a*b^6*c^4
*d^3 - 2*A*B^2*a^2*b^5*c*d^6 + 6*A*B^2*a^4*b^3*c*d^6 - 4*A^2*B*a*b^6*c^3*d^
4 - 4*A^2*B*a^3*b^4*c*d^6 - 4*A*C^2*a*b^6*c^2*d^5 + 4*A*C^2*a*b^6*c^4*d^3 -
 4*A*C^2*a^2*b^5*c*d^6 + 4*A*C^2*a^4*b^3*c*d^6 + 8*A^2*C*a*b^6*c^2*d^5 - 2*
A^2*C*a*b^6*c^4*d^3 + 8*A^2*C*a^2*b^5*c*d^6 - 2*A^2*C*a^4*b^3*c*d^6 + 4*B*C
^2*a*b^6*c^3*d^4 + 4*B*C^2*a*b^6*c^5*d^2 + 4*B*C^2*a^3*b^4*c*d^6 + 4*B*C^2*
a^5*b^2*c*d^6 - 4*B^2*C*a*b^6*c^2*d^5 - 10*B^2*C*a*b^6*c^4*d^3 - 4*B^2*C*a^
2*b^5*c*d^6 - 10*B^2*C*a^4*b^3*c*d^6 - 4*A*B*C*a^2*b^5*c^2*d^5 + 8*A*B*C*a^
2*b^5*c^4*d^3 + 8*A*B*C*a^4*b^3*c^2*d^5 + 8*A*B*C*a*b^6*c*d^6 - 4*A*B*C*a*b
^6*c^5*d^2 - 4*A*B*C*a^5*b^2*c*d^6))/(a^8*d^8 + b^8*c^8 + 2*a^2*b^6*c^8 + a
^4*b^4*c^8 + a^4*b^4*d^8 + 2*a^6*b^2*d^8 + 2*a^8*c^2*d^6 + a^8*c^4*d^4 + b^
8*c^4*d^4 + 2*b^8*c^6*d^2 - 4*a*b^7*c^3*d^5 - 8*a*b^7*c^5*d^3 - 4*a^3*b^5*c
*d^7 - 8*a^3*b^5*c^7*d - 8*a^5*b^3*c*d^7 - 4*a^5*b^3*c^7*d - 8*a^7*b*c^3*d^
5 - 4*a^7*b*c^5*d^3 + 6*a^2*b^6*c^2*d^6 + 14*a^2*b^6*c^4*d^4 + 10*a^2*b^6*c
^6*d^2 - 16*a^3*b^5*c^3*d^5 - 20*a^3*b^5*c^5*d^3 + 14*a^4*b^4*c^2*d^6 + 26*
a^4*b^4*c^4*d^4 + 14*a^4*b^4*c^6*d^2 - 20*a^5*b^3*c^3*d^5 - 16*a^5*b^3*c^5*
d^3 + 10*a^6*b^2*c^2*d^6 + 14*a^6*b^2*c^4*d^4 + 6*a^6*b^2*c^6*d^2 - 4*a*b^7
*c^7*d - 4*a^7*b*c*d^7) - (4*A^2*C*b^7*d^7 - 6*A^3*a^2*b^5*d^7 - B^3*a^3*b^
4*d^7 - 6*A^3*b^7*c^2*d^5 - B^3*b^7*c^3*d^4 - 4*A^3*b^7*d^7 - 8*A^3*a^2*b^5
*c^2*d^5 - 3*B^3*a^2*b^5*c^3*d^4 - 3*B^3*a^3*b^4*c^2*d^5 + 2*C^3*a^2*b^5*c^
```

4*d^3 + 2*C^3*a^4*b^3*c^2*d^5 + 4*C^3*a^4*b^3*c^4*d^3 + 4*A^2*B*a*b^6*d^7 + 4*A^2*B*b^7*c*d^6 + 4*A^3*a*b^6*c*d^6 + A*B^2*a^2*b^5*d^7 - 3*A*B^2*a^4*b^3*d^7 + 9*A^2*B*a^3*b^4*d^7 + 2*A*C^2*a^2*b^5*d^7 + 4*A*C^2*a^4*b^3*d^7 + 4*A^2*C*a^2*b^5*d^7 - 4*A^2*C*a^4*b^3*d^7 + A*B^2*b^7*c^2*d^5 - 3*A*B^2*b^7*c^4*d^3 - B*C^2*a^3*b^4*d^7 - 2*B*C^2*a^5*b^2*d^7 + 9*A^2*B*b^7*c^3*d^4 + B^2*C*a^2*b^5*d^7 + 3*B^2*C*a^4*b^3*d^7 + 2*A*C^2*b^7*c^2*d^5 + 4*A*C^2*b^7*c^4*d^3 + 4*A^2*C*b^7*c^2*d^5 - 4*A^2*C*b^7*c^4*d^3 - B*C^2*b^7*c^3*d^4 - 2*B*C^2*b^7*c^5*d^2 + B^2*C*b^7*c^2*d^5 + 3*B^2*C*b^7*c^4*d^3 + 2*A^3*a*b^6*c^3*d^4 + 2*A^3*a^3*b^4*c*d^6 + B^3*a*b^6*c^2*d^5 + 3*B^3*a*b^6*c^4*d^3 + B^3*a^2*b^5*c*d^6 + 3*B^3*a^4*b^3*c*d^6 + 2*C^3*a*b^6*c^3*d^4 + 2*C^3*a*b^6*c^5*d^2 + 2*C^3*a^3*b^4*c*d^6 + 2*C^3*a^5*b^2*c*d^6 - 4*A*B*C*a*b^6*d^7 - 4*A*B*C*b^7*c*d^6 + 14*A*B^2*a^2*b^5*c^2*d^5 + 3*A*B^2*a^2*b^5*c^4*d^3 - 10*A*B^2*a^3*b^4*c^3*d^4 + 3*A*B^2*a^4*b^3*c^2*d^5 + 7*A^2*B*a^2*b^5*c^3*d^4 + 7*A^2*B*a^3*b^4*c^2*d^5 + 8*A*C^2*a^2*b^5*c^2*d^5 + 4*A*C^2*a^2*b^5*c^4*d^3 + 4*A*C^2*a^4*b^3*c^2*d^5 - 4*A*C^2*a^4*b^3*c^4*d^3 - 6*A^2*C*a^2*b^5*c^4*d^3 - 6*A^2*C*a^4*b^3*c^2*d^5 - B*C^2*a^2*b^5*c^3*d^4 + 2*B*C^2*a^2*b^5*c^5*d^2 - B*C^2*a^3*b^4*c^2*d^5 - 6*B*C^2*a^3*b^4*c^4*d^3 - 6*B*C^2*a^4*b^3*c^3*d^4 + 2*B*C^2*a^5*b^2*c^2*d^5 - 6*B^2*C*a^2*b^5*c^2*d^5 - B^2*C*a^2*b^5*c^4*d^3 + 10*B^2*C*a^3*b^4*c^3*d^4 - B^2*C*a^4*b^3*c^2*d^5 - 8*A*B*C*a^3*b^4*d^7 + 2*A*B*C*a^5*b^2*d^7 - 8*A*B*C*b^7*c^3*d^4 + 2*A*B*C*b^7*c^5*d^2 - 6*A*B^2*a*b^6*c*d^6 + 4*A*C^2*a*b^6*c*d^6 - 8*A^2*C*a*b^6*c*d^6 + 2*B^2*C*a*b^6*c*d^6 - 8*A*B^2*a*b^6*c^3*d^4 - 8*A*B^2*a^3*b^4*c*d^6 - A^2*B*a*b^6*c^2*d^5 - 3*A^2*B*a*b^6*c^4*d^3 - A^2*B*a^2*b^5*c*d^6 - 3*A^2*B*a^4*b^3*c*d^6 - 2*A*C^2*a*b^6*c^3*d^4 - 4*A*C^2*a*b^6*c^5*d^2 - 2*A*C^2*a^3*b^4*c*d^6 - 4*A*C^2*a^5*b^2*c*d^6 - 2*A^2*C*a*b^6*c^3*d^4 + 2*A^2*C*a*b^6*c^5*d^2 - 2*A^2*C*a^3*b^4*c*d^6 + 2*A^2*C*a^5*b^2*c*d^6 - 3*B*C^2*a*b^6*c^2*d^5 - 5*B*C^2*a*b^6*c^4*d^3 - 3*B*C^2*a^2*b^5*c*d^6 - 5*B*C^2*a^4*b^3*c*d^6 + 4*B^2*C*a*b^6*c^3*d^4 - 2*B^2*C*a*b^6*c^5*d^2 + 4*B^2*C*a^3*b^4*c*d^6 - 2*B^2*C*a^5*b^2*c*d^6 - 6*A*B*C*a^2*b^5*c^3*d^4 - 2*A*B*C*a^2*b^5*c^5*d^2 - 6*A*B*C*a^3*b^4*c^2*d^5 + 6*A*B*C*a^3*b^4*c^4*d^3 + 6*A*B*C*a^4*b^3*c^3*d^4 - 2*A*B*C*a^5*b^2*c^2*d^5 + 4*A*B*C*a*b^6*c^2*d^5 + 8*A*B*C*a*b^6*c^4*d^3 + 4*A*B*C*a^2*b^5*c*d^6 + 8*A*B*C*a^4*b^3*c*d^6)/(a^8*d^8 + b^8*c^8 + 2*a^2*b^6*c^8 + a^4*b^4*c^8 + a^4*b^4*d^8 + 2*a^6*b^2*d^8 + 2*a^8*c^2*d^6 + a^8*c^4*d^4 + b^8*c^4*d^4 + 2*b^8*c^6*d^2 - 4*a*b^7*c^3*d^5 - 8*a*b^7*c^5*d^3 - 4*a^3*b^5*c*d^7 - 8*a^3*b^5*c^7*d - 8*a^5*b^3*c*d^7 - 4*a^5*b^3*c^7*d - 8*a^7*b*c^3*d^5 - 4*a^7*b*c^5*d^3 + 6*a^2*b^6*c^2*d^6 + 14*a^2*b^6*c^4*d^4 + 10*a^2*b^6*c^6*d^2 - 16*a^3*b^5*c^3*d^5 - 20*a^3*b^5*c^5*d^3 + 14*a^4*b^4*c^2*d^6 + 26*a^4*b^4*c^4*d^4 + 14*a^4*b^4*c^6*d^2 - 20*a^5*b^3*c^3*d^5 - 16*a^5*b^3*c^5*d^3 + 10*a^6*b^2*c^2*d^6 + 14*a^6*b^2*c^4*d^4 + 6*a^6*b^2*c^6*d^2 - 4*a*b^7*c^7*d - 4*a^7*b*c*d^7) - root(144*a^13*b*c^5*d^9*f^4 + 144*a^9*b^5*c*d^13*f^4 + 144*a^5*b^9*c^13*d*f^4 + 144*a*b^13*c^9*d^5*f^4 + 96*a^13*b*c^7*d^7*f^4 + 96*a^13*b*c^3*d^11*f^4 + 96*a^11*b^3*c*d^13*f^4 + 96*a^7*b^7*c^13*d*f^4 + 96*a^7*b^7*c*d^13*f^4 + 96*a^3*b^11*c^13*d*f^4 + 96*a*b^13*c^11*d^3*f^4 + 96*a*b^13*c^7*d^7*f^4 + 24*a^13*b*c^9*d^5*f^4 + 24*a^9*b^5*c^13*d*f^4 + 24*a^5*b^9*c*d^13*f^4 + 24*a*b^13*c^5*d^9*f^4 + 24*a^13*b*c*d^13*f^4 + 24*a

*b^13*c^13*d*f^4 + 3648*a^7*b^7*c^7*d^7*f^4 - 3188*a^8*b^6*c^6*d^8*f^4 - 31
88*a^6*b^8*c^8*d^6*f^4 - 2912*a^8*b^6*c^8*d^6*f^4 - 2912*a^6*b^8*c^6*d^8*f^
4 + 2592*a^9*b^5*c^7*d^7*f^4 + 2592*a^7*b^7*c^9*d^5*f^4 + 2592*a^7*b^7*c^5*
d^9*f^4 + 2592*a^5*b^9*c^7*d^7*f^4 + 2168*a^9*b^5*c^5*d^9*f^4 + 2168*a^5*b^
9*c^9*d^5*f^4 - 1776*a^10*b^4*c^6*d^8*f^4 - 1776*a^8*b^6*c^4*d^10*f^4 - 177
6*a^6*b^8*c^10*d^4*f^4 - 1776*a^4*b^10*c^8*d^6*f^4 + 1568*a^9*b^5*c^9*d^5*f
^4 + 1568*a^5*b^9*c^5*d^9*f^4 - 1344*a^10*b^4*c^8*d^6*f^4 - 1344*a^8*b^6*c^
10*d^4*f^4 - 1344*a^6*b^8*c^4*d^10*f^4 - 1344*a^4*b^10*c^6*d^8*f^4 - 1164*a
^10*b^4*c^4*d^10*f^4 - 1164*a^4*b^10*c^10*d^4*f^4 + 896*a^11*b^3*c^5*d^9*f^
4 + 896*a^9*b^5*c^3*d^11*f^4 + 896*a^5*b^9*c^11*d^3*f^4 + 896*a^3*b^11*c^9*
d^5*f^4 + 864*a^11*b^3*c^7*d^7*f^4 + 864*a^7*b^7*c^11*d^3*f^4 + 864*a^7*b^7
*c^3*d^11*f^4 + 864*a^3*b^11*c^7*d^7*f^4 - 480*a^10*b^4*c^10*d^4*f^4 - 480*
a^4*b^10*c^4*d^10*f^4 + 464*a^11*b^3*c^3*d^11*f^4 + 464*a^3*b^11*c^11*d^3*f
^4 - 424*a^12*b^2*c^6*d^8*f^4 - 424*a^8*b^6*c^2*d^12*f^4 - 424*a^6*b^8*c^12
*d^2*f^4 - 424*a^2*b^12*c^8*d^6*f^4 + 416*a^11*b^3*c^9*d^5*f^4 + 416*a^9*b^
5*c^11*d^3*f^4 + 416*a^5*b^9*c^3*d^11*f^4 + 416*a^3*b^11*c^5*d^9*f^4 - 336*
a^12*b^2*c^4*d^10*f^4 - 336*a^10*b^4*c^2*d^12*f^4 - 336*a^4*b^10*c^12*d^2*f
^4 - 336*a^2*b^12*c^10*d^4*f^4 - 256*a^12*b^2*c^8*d^6*f^4 - 256*a^8*b^6*c^1
2*d^2*f^4 - 256*a^6*b^8*c^2*d^12*f^4 - 256*a^2*b^12*c^6*d^8*f^4 - 124*a^12*
b^2*c^2*d^12*f^4 - 124*a^2*b^12*c^12*d^2*f^4 + 80*a^11*b^3*c^11*d^3*f^4 + 8
0*a^3*b^11*c^3*d^11*f^4 - 60*a^12*b^2*c^10*d^4*f^4 - 60*a^10*b^4*c^12*d^2*f
^4 - 60*a^4*b^10*c^2*d^12*f^4 - 60*a^2*b^12*c^4*d^10*f^4 - 24*b^14*c^10*d^4
*f^4 - 16*b^14*c^12*d^2*f^4 - 16*b^14*c^8*d^6*f^4 - 4*b^14*c^6*d^8*f^4 - 24
*a^14*c^4*d^10*f^4 - 16*a^14*c^6*d^8*f^4 - 16*a^14*c^2*d^12*f^4 - 4*a^14*c^
8*d^6*f^4 - 24*a^10*b^4*d^14*f^4 - 16*a^12*b^2*d^14*f^4 - 16*a^8*b^6*d^14*f
^4 - 4*a^6*b^8*d^14*f^4 - 24*a^4*b^10*c^14*f^4 - 16*a^6*b^8*c^14*f^4 - 16*a
^2*b^12*c^14*f^4 - 4*a^8*b^6*c^14*f^4 - 4*b^14*c^14*f^4 - 4*a^14*d^14*f^4 +
36*A*C*a^9*b*c*d^9*f^2 + 36*A*C*a*b^9*c^9*d*f^2 + 32*A*C*a*b^9*c*d^9*f^2 -
552*B*C*a^7*b^3*c^4*d^6*f^2 - 552*B*C*a^4*b^6*c^7*d^3*f^2 - 408*B*C*a^5*b^
5*c^4*d^6*f^2 - 408*B*C*a^4*b^6*c^5*d^5*f^2 + 360*B*C*a^6*b^4*c^3*d^7*f^2 +
360*B*C*a^3*b^7*c^6*d^4*f^2 - 248*B*C*a^7*b^3*c^2*d^8*f^2 - 248*B*C*a^2*b^
8*c^7*d^3*f^2 + 184*B*C*a^6*b^4*c^5*d^5*f^2 + 184*B*C*a^5*b^5*c^6*d^4*f^2 +
152*B*C*a^8*b^2*c^3*d^7*f^2 - 152*B*C*a^5*b^5*c^2*d^8*f^2 + 152*B*C*a^3*b^
7*c^8*d^2*f^2 - 152*B*C*a^2*b^8*c^5*d^5*f^2 - 104*B*C*a^7*b^3*c^6*d^4*f^2 -
104*B*C*a^6*b^4*c^7*d^3*f^2 + 64*B*C*a^8*b^2*c^5*d^5*f^2 + 64*B*C*a^5*b^5*
c^8*d^2*f^2 - 56*B*C*a^4*b^6*c^3*d^7*f^2 - 56*B*C*a^3*b^7*c^4*d^6*f^2 - 24*
B*C*a^8*b^2*c^7*d^3*f^2 - 24*B*C*a^7*b^3*c^8*d^2*f^2 - 24*B*C*a^3*b^7*c^2*d
^8*f^2 - 24*B*C*a^2*b^8*c^3*d^7*f^2 - 696*A*C*a^5*b^5*c^5*d^5*f^2 + 536*A*C
*a^6*b^4*c^6*d^4*f^2 + 536*A*C*a^6*b^4*c^4*d^6*f^2 + 536*A*C*a^4*b^6*c^6*d^
4*f^2 + 472*A*C*a^4*b^6*c^4*d^6*f^2 - 232*A*C*a^7*b^3*c^5*d^5*f^2 - 232*A*C
*a^5*b^5*c^7*d^3*f^2 + 216*A*C*a^3*b^7*c^3*d^7*f^2 + 168*A*C*a^7*b^3*c^3*d^
7*f^2 + 168*A*C*a^3*b^7*c^7*d^3*f^2 - 154*A*C*a^8*b^2*c^2*d^8*f^2 - 154*A*C
*a^2*b^8*c^8*d^2*f^2 + 62*A*C*a^8*b^2*c^6*d^4*f^2 + 62*A*C*a^6*b^4*c^8*d^2*
f^2 - 40*A*C*a^7*b^3*c^7*d^3*f^2 - 40*A*C*a^5*b^5*c^3*d^7*f^2 - 40*A*C*a^3*
b^7*c^5*d^5*f^2 + 32*A*C*a^6*b^4*c^2*d^8*f^2 + 32*A*C*a^2*b^8*c^6*d^4*f^2 -

32*A*C*a^2*b^8*c^2*d^8*f^2 + 30*A*C*a^4*b^6*c^2*d^8*f^2 + 30*A*C*a^2*b^8*c
^4*d^6*f^2 + 16*A*C*a^8*b^2*c^4*d^6*f^2 + 16*A*C*a^4*b^6*c^8*d^2*f^2 - 488*
A*B*a^6*b^4*c^3*d^7*f^2 - 488*A*B*a^3*b^7*c^6*d^4*f^2 + 440*A*B*a^7*b^3*c^4
*d^6*f^2 + 440*A*B*a^4*b^6*c^7*d^3*f^2 - 360*A*B*a^6*b^4*c^5*d^5*f^2 - 360*
A*B*a^5*b^5*c^6*d^4*f^2 - 192*A*B*a^8*b^2*c^3*d^7*f^2 - 192*A*B*a^3*b^7*c^8
*d^2*f^2 - 168*A*B*a^3*b^7*c^2*d^8*f^2 - 168*A*B*a^2*b^8*c^3*d^7*f^2 - 152*
A*B*a^4*b^6*c^3*d^7*f^2 - 152*A*B*a^3*b^7*c^4*d^6*f^2 - 120*A*B*a^8*b^2*c^5
*d^5*f^2 + 120*A*B*a^7*b^3*c^2*d^8*f^2 - 120*A*B*a^5*b^5*c^8*d^2*f^2 + 120*
A*B*a^5*b^5*c^4*d^6*f^2 - 120*A*B*a^5*b^5*c^2*d^8*f^2 + 120*A*B*a^4*b^6*c^5
*d^5*f^2 + 120*A*B*a^2*b^8*c^7*d^3*f^2 - 120*A*B*a^2*b^8*c^5*d^5*f^2 + 40*A
*B*a^7*b^3*c^6*d^4*f^2 + 40*A*B*a^6*b^4*c^7*d^3*f^2 - 72*B*C*a^9*b*c^4*d^6*
f^2 - 72*B*C*a^4*b^6*c^9*d*f^2 - 64*B*C*a^4*b^6*c*d^9*f^2 - 64*B*C*a*b^9*c^
4*d^6*f^2 - 32*B*C*a^8*b^2*c*d^9*f^2 - 32*B*C*a*b^9*c^8*d^2*f^2 - 16*B*C*a^
2*b^8*c*d^9*f^2 - 16*B*C*a*b^9*c^2*d^8*f^2 + 8*B*C*a^9*b*c^6*d^4*f^2 - 8*B*
C*a^9*b*c^2*d^8*f^2 + 8*B*C*a^6*b^4*c^9*d*f^2 - 8*B*C*a^2*b^8*c^9*d*f^2 + 1
04*A*C*a^7*b^3*c*d^9*f^2 + 104*A*C*a*b^9*c^7*d^3*f^2 + 96*A*C*a^3*b^7*c*d^9
*f^2 + 96*A*C*a*b^9*c^3*d^7*f^2 + 72*A*C*a^9*b*c^3*d^7*f^2 + 72*A*C*a^3*b^7
*c^9*d*f^2 + 68*A*C*a^5*b^5*c*d^9*f^2 + 68*A*C*a*b^9*c^5*d^5*f^2 - 28*A*C*a
^9*b*c^5*d^5*f^2 - 28*A*C*a^5*b^5*c^9*d*f^2 + 80*A*B*a^9*b*c^4*d^6*f^2 + 80
*A*B*a^4*b^6*c^9*d*f^2 + 24*A*B*a^8*b^2*c*d^9*f^2 - 24*A*B*a^6*b^4*c*d^9*f^
2 + 24*A*B*a^4*b^6*c*d^9*f^2 - 24*A*B*a^2*b^8*c*d^9*f^2 + 24*A*B*a*b^9*c^8*
d^2*f^2 - 24*A*B*a*b^9*c^6*d^4*f^2 + 24*A*B*a*b^9*c^4*d^6*f^2 - 24*A*B*a*b^
9*c^2*d^8*f^2 - 32*B*C*b^10*c^7*d^3*f^2 - 8*B*C*b^10*c^5*d^5*f^2 + 34*A*C*b
^10*c^6*d^4*f^2 + 16*B*C*a^10*c^3*d^7*f^2 + 16*A*C*b^10*c^4*d^6*f^2 - 12*A*
C*b^10*c^8*d^2*f^2 - 96*A*B*b^10*c^5*d^5*f^2 - 72*A*B*b^10*c^3*d^7*f^2 - 32
*B*C*a^7*b^3*d^10*f^2 - 28*A*C*a^10*c^2*d^8*f^2 - 24*A*B*b^10*c^7*d^3*f^2 -
8*B*C*a^5*b^5*d^10*f^2 + 2*A*C*a^10*c^4*d^6*f^2 + 34*A*C*a^6*b^4*d^10*f^2
+ 16*B*C*a^3*b^7*c^10*f^2 + 16*A*C*a^4*b^6*d^10*f^2 - 16*A*B*a^10*c^3*d^7*f
^2 - 12*A*C*a^8*b^2*d^10*f^2 - 96*A*B*a^5*b^5*d^10*f^2 - 72*A*B*a^3*b^7*d^1
0*f^2 - 28*A*C*a^2*b^8*c^10*f^2 - 24*A*B*a^7*b^3*d^10*f^2 + 2*A*C*a^4*b^6*c
^10*f^2 - 16*A*B*a^3*b^7*c^10*f^2 + 444*C^2*a^5*b^5*c^5*d^5*f^2 + 148*C^2*a
^7*b^3*c^5*d^5*f^2 + 148*C^2*a^5*b^5*c^7*d^3*f^2 + 148*C^2*a^5*b^5*c^3*d^7*
f^2 + 148*C^2*a^3*b^7*c^5*d^5*f^2 - 140*C^2*a^6*b^4*c^6*d^4*f^2 - 140*C^2*a
^6*b^4*c^4*d^6*f^2 - 140*C^2*a^4*b^6*c^6*d^4*f^2 - 140*C^2*a^4*b^6*c^4*d^6*
f^2 + 109*C^2*a^8*b^2*c^2*d^8*f^2 + 109*C^2*a^2*b^8*c^8*d^2*f^2 + 48*C^2*a^
8*b^2*c^4*d^6*f^2 + 48*C^2*a^6*b^4*c^2*d^8*f^2 + 48*C^2*a^4*b^6*c^8*d^2*f^2
+ 48*C^2*a^2*b^8*c^6*d^4*f^2 + 20*C^2*a^7*b^3*c^7*d^3*f^2 - 20*C^2*a^7*b^3
*c^3*d^7*f^2 - 20*C^2*a^3*b^7*c^7*d^3*f^2 + 20*C^2*a^3*b^7*c^3*d^7*f^2 + 17
*C^2*a^8*b^2*c^6*d^4*f^2 + 17*C^2*a^6*b^4*c^8*d^2*f^2 + 17*C^2*a^4*b^6*c^2*
d^8*f^2 + 17*C^2*a^2*b^8*c^4*d^6*f^2 + 16*C^2*a^8*b^2*c^8*d^2*f^2 + 16*C^2*
a^2*b^8*c^2*d^8*f^2 - 396*B^2*a^5*b^5*c^5*d^5*f^2 + 308*B^2*a^6*b^4*c^4*d^6
*f^2 + 308*B^2*a^4*b^6*c^6*d^4*f^2 + 300*B^2*a^4*b^6*c^4*d^6*f^2 + 284*B^2*
a^6*b^4*c^6*d^4*f^2 - 132*B^2*a^7*b^3*c^5*d^5*f^2 - 132*B^2*a^5*b^5*c^7*d^3
*f^2 - 84*B^2*a^5*b^5*c^3*d^7*f^2 - 84*B^2*a^3*b^7*c^5*d^5*f^2 + 61*B^2*a^4
*b^6*c^2*d^8*f^2 + 61*B^2*a^2*b^8*c^4*d^6*f^2 - 59*B^2*a^8*b^2*c^2*d^8*f^2

- 59*B^2*a^2*b^8*c^8*d^2*f^2 + 56*B^2*a^6*b^4*c^2*d^8*f^2 + 56*B^2*a^2*b^8*
c^6*d^4*f^2 + 52*B^2*a^7*b^3*c^3*d^7*f^2 + 52*B^2*a^3*b^7*c^7*d^3*f^2 + 44*
B^2*a^3*b^7*c^3*d^7*f^2 + 33*B^2*a^8*b^2*c^6*d^4*f^2 + 33*B^2*a^6*b^4*c^8*d
^2*f^2 + 20*B^2*a^8*b^2*c^4*d^6*f^2 - 20*B^2*a^7*b^3*c^7*d^3*f^2 + 20*B^2*a
^4*b^6*c^8*d^2*f^2 + 8*B^2*a^2*b^8*c^2*d^8*f^2 + 337*A^2*a^4*b^6*c^2*d^8*f^
2 + 337*A^2*a^2*b^8*c^4*d^6*f^2 + 272*A^2*a^2*b^8*c^2*d^8*f^2 + 252*A^2*a^5
*b^5*c^5*d^5*f^2 + 244*A^2*a^4*b^6*c^4*d^6*f^2 - 236*A^2*a^3*b^7*c^3*d^7*f^
2 + 176*A^2*a^6*b^4*c^2*d^8*f^2 + 176*A^2*a^2*b^8*c^6*d^4*f^2 - 148*A^2*a^7
*b^3*c^3*d^7*f^2 - 148*A^2*a^3*b^7*c^7*d^3*f^2 - 140*A^2*a^6*b^4*c^6*d^4*f^
2 + 109*A^2*a^8*b^2*c^2*d^8*f^2 + 109*A^2*a^2*b^8*c^8*d^2*f^2 - 108*A^2*a^5
*b^5*c^3*d^7*f^2 - 108*A^2*a^3*b^7*c^5*d^5*f^2 + 84*A^2*a^7*b^3*c^5*d^5*f^2
 + 84*A^2*a^5*b^5*c^7*d^3*f^2 + 32*A^2*a^8*b^2*c^4*d^6*f^2 + 32*A^2*a^4*b^6
*c^8*d^2*f^2 + 20*A^2*a^7*b^3*c^7*d^3*f^2 - 15*A^2*a^8*b^2*c^6*d^4*f^2 - 15
*A^2*a^6*b^4*c^8*d^2*f^2 - 12*A^2*a^6*b^4*c^4*d^6*f^2 - 12*A^2*a^4*b^6*c^6*
d^4*f^2 + 8*B*C*b^10*c^9*d*f^2 - 16*B*C*a^10*c*d^9*f^2 - 16*A*B*b^10*c^9*d*
f^2 - 16*A*B*b^10*c*d^9*f^2 + 8*B*C*a^9*b*d^10*f^2 - 16*B*C*a*b^9*c^10*f^2
+ 16*A*B*a^10*c*d^9*f^2 - 16*A*B*a^9*b*d^10*f^2 - 16*A*B*a*b^9*d^10*f^2 + 1
6*A*B*a*b^9*c^10*f^2 + 22*C^2*a^9*b*c^5*d^5*f^2 + 22*C^2*a^5*b^5*c^9*d*f^2
+ 22*C^2*a^5*b^5*c*d^9*f^2 + 22*C^2*a*b^9*c^5*d^5*f^2 - 20*C^2*a^9*b*c^3*d^
7*f^2 - 20*C^2*a^7*b^3*c*d^9*f^2 - 20*C^2*a^3*b^7*c^9*d*f^2 - 20*C^2*a*b^9*
c^7*d^3*f^2 + 36*B^2*a^7*b^3*c*d^9*f^2 + 36*B^2*a*b^9*c^7*d^3*f^2 + 28*B^2*
a^9*b*c^3*d^7*f^2 + 28*B^2*a^3*b^7*c^9*d*f^2 + 24*B^2*a^3*b^7*c*d^9*f^2 + 2
4*B^2*a*b^9*c^3*d^7*f^2 - 18*B^2*a^9*b*c^5*d^5*f^2 - 18*B^2*a^5*b^5*c^9*d*f
^2 + 6*B^2*a^5*b^5*c*d^9*f^2 + 6*B^2*a*b^9*c^5*d^5*f^2 - 96*A^2*a^3*b^7*c*d
^9*f^2 - 96*A^2*a*b^9*c^3*d^7*f^2 - 90*A^2*a^5*b^5*c*d^9*f^2 - 90*A^2*a*b^9
*c^5*d^5*f^2 - 84*A^2*a^7*b^3*c*d^9*f^2 - 84*A^2*a*b^9*c^7*d^3*f^2 - 52*A^2
*a^9*b*c^3*d^7*f^2 - 52*A^2*a^3*b^7*c^9*d*f^2 + 6*A^2*a^9*b*c^5*d^5*f^2 + 6
*A^2*a^5*b^5*c^9*d*f^2 - 10*C^2*a^9*b*c*d^9*f^2 - 10*C^2*a*b^9*c^9*d*f^2 +
14*B^2*a^9*b*c*d^9*f^2 + 14*B^2*a*b^9*c^9*d*f^2 + 8*B^2*a*b^9*c*d^9*f^2 - 3
2*A^2*a*b^9*c*d^9*f^2 - 26*A^2*a^9*b*c*d^9*f^2 - 26*A^2*a*b^9*c^9*d*f^2 + 2
*A*C*b^10*c^10*f^2 + 2*A*C*a^10*d^10*f^2 + 14*C^2*b^10*c^8*d^2*f^2 - C^2*b^
10*c^6*d^4*f^2 + 31*B^2*b^10*c^6*d^4*f^2 + 20*B^2*b^10*c^4*d^6*f^2 + 14*C^2
*a^10*c^2*d^8*f^2 + 4*B^2*b^10*c^2*d^8*f^2 + 2*B^2*b^10*c^8*d^2*f^2 - C^2*a
^10*c^4*d^6*f^2 + 80*A^2*b^10*c^4*d^6*f^2 + 64*A^2*b^10*c^2*d^8*f^2 + 31*A^
2*b^10*c^6*d^4*f^2 + 14*C^2*a^8*b^2*d^10*f^2 + 14*A^2*b^10*c^8*d^2*f^2 - 10
*B^2*a^10*c^2*d^8*f^2 + 3*B^2*a^10*c^4*d^6*f^2 - C^2*a^6*b^4*d^10*f^2 + 31*
B^2*a^6*b^4*d^10*f^2 + 20*B^2*a^4*b^6*d^10*f^2 + 14*C^2*a^2*b^8*c^10*f^2 +
14*A^2*a^10*c^2*d^8*f^2 + 4*B^2*a^2*b^8*d^10*f^2 + 2*B^2*a^8*b^2*d^10*f^2 -
 C^2*a^4*b^6*c^10*f^2 - A^2*a^10*c^4*d^6*f^2 + 80*A^2*a^4*b^6*d^10*f^2 + 64
*A^2*a^2*b^8*d^10*f^2 + 31*A^2*a^6*b^4*d^10*f^2 + 14*A^2*a^8*b^2*d^10*f^2 -
 10*B^2*a^2*b^8*c^10*f^2 + 3*B^2*a^4*b^6*c^10*f^2 + 14*A^2*a^2*b^8*c^10*f^2
 - A^2*a^4*b^6*c^10*f^2 - C^2*b^10*c^10*f^2 - C^2*a^10*d^10*f^2 + 16*A^2*b^
10*d^10*f^2 + 3*B^2*b^10*c^10*f^2 + 3*B^2*a^10*d^10*f^2 - A^2*b^10*c^10*f^2
 - A^2*a^10*d^10*f^2 - 96*A*B*C*a*b^7*c*d^7*f - 28*A*B*C*a^7*b*c*d^7*f - 28
*A*B*C*a*b^7*c^7*d*f + 484*A*B*C*a^4*b^4*c^4*d^4*f - 424*A*B*C*a^3*b^5*c^3*

d^5*f + 320*A*B*C*a^2*b^6*c^2*d^6*f - 176*A*B*C*a^6*b^2*c^2*d^6*f - 176*A*B
*C*a^2*b^6*c^6*d^2*f + 158*A*B*C*a^4*b^4*c^2*d^6*f + 158*A*B*C*a^2*b^6*c^4*
d^4*f - 136*A*B*C*a^5*b^3*c^5*d^3*f - 34*A*B*C*a^6*b^2*c^4*d^4*f - 34*A*B*C
*a^4*b^4*c^6*d^2*f + 28*A*B*C*a^5*b^3*c^3*d^5*f + 28*A*B*C*a^3*b^5*c^5*d^3*
f + 308*A*B*C*a^5*b^3*c*d^7*f + 308*A*B*C*a*b^7*c^5*d^3*f + 20*A*B*C*a^7*b*
c^3*d^5*f + 20*A*B*C*a^3*b^5*c^7*d*f + 30*B*C^2*a^7*b*c*d^7*f + 30*B*C^2*a*
b^7*c^7*d*f + 160*A^2*B*a*b^7*c*d^7*f - 2*A^2*B*a^7*b*c*d^7*f - 2*A^2*B*a*b
^7*c^7*d*f - 96*A*B*C*b^8*c^4*d^4*f + 34*A*B*C*b^8*c^6*d^2*f - 32*A*B*C*b^8
*c^2*d^6*f + 2*A*B*C*a^8*c^2*d^6*f - 96*A*B*C*a^4*b^4*d^8*f + 34*A*B*C*a^6*
b^2*d^8*f - 32*A*B*C*a^2*b^6*d^8*f + 2*A*B*C*a^2*b^6*c^8*f - 210*B*C^2*a^4*
b^4*c^4*d^4*f - 182*B^2*C*a^5*b^3*c^2*d^6*f - 182*B^2*C*a^2*b^6*c^5*d^3*f +
 180*B*C^2*a^5*b^3*c^5*d^3*f + 180*B*C^2*a^3*b^5*c^3*d^5*f - 166*B^2*C*a^5*
b^3*c^4*d^4*f - 166*B^2*C*a^4*b^4*c^5*d^3*f + 152*B*C^2*a^6*b^2*c^2*d^6*f +
 152*B*C^2*a^2*b^6*c^6*d^2*f - 112*B^2*C*a^3*b^5*c^2*d^6*f - 112*B^2*C*a^2*
b^6*c^3*d^5*f + 94*B^2*C*a^4*b^4*c^3*d^5*f + 94*B^2*C*a^3*b^5*c^4*d^4*f - 8
0*B*C^2*a^2*b^6*c^2*d^6*f + 66*B*C^2*a^5*b^3*c^3*d^5*f + 66*B*C^2*a^3*b^5*c
^5*d^3*f + 46*B^2*C*a^6*b^2*c^3*d^5*f + 46*B^2*C*a^3*b^5*c^6*d^2*f + 33*B*C
^2*a^6*b^2*c^4*d^4*f + 33*B*C^2*a^4*b^4*c^6*d^2*f + 24*B^2*C*a^6*b^2*c^5*d^
3*f + 24*B^2*C*a^5*b^3*c^6*d^2*f - 16*B*C^2*a^6*b^2*c^6*d^2*f - 15*B*C^2*a^
4*b^4*c^2*d^6*f - 15*B*C^2*a^2*b^6*c^4*d^4*f - 190*A^2*C*a^4*b^4*c^3*d^5*f
- 190*A^2*C*a^3*b^5*c^4*d^4*f + 182*A^2*C*a^5*b^3*c^2*d^6*f + 182*A^2*C*a^2
*b^6*c^5*d^3*f + 160*A^2*C*a^3*b^5*c^2*d^6*f + 160*A^2*C*a^2*b^6*c^3*d^5*f
- 150*A*C^2*a^5*b^3*c^2*d^6*f - 150*A*C^2*a^2*b^6*c^5*d^3*f - 126*A*C^2*a^5
*b^3*c^4*d^4*f - 126*A*C^2*a^4*b^4*c^5*d^3*f + 126*A*C^2*a^4*b^4*c^3*d^5*f
+ 126*A*C^2*a^3*b^5*c^4*d^4*f - 96*A*C^2*a^3*b^5*c^2*d^6*f - 96*A*C^2*a^2*b
^6*c^3*d^5*f + 94*A^2*C*a^5*b^3*c^4*d^4*f + 94*A^2*C*a^4*b^4*c^5*d^3*f + 54
*A*C^2*a^6*b^2*c^3*d^5*f + 54*A*C^2*a^3*b^5*c^6*d^2*f + 32*A*C^2*a^6*b^2*c^
5*d^3*f + 32*A*C^2*a^5*b^3*c^6*d^2*f - 22*A^2*C*a^6*b^2*c^3*d^5*f - 22*A^2*
C*a^3*b^5*c^6*d^2*f + 500*A^2*B*a^3*b^5*c^3*d^5*f - 290*A^2*B*a^4*b^4*c^4*d
^4*f - 256*A^2*B*a^2*b^6*c^2*d^6*f - 230*A*B^2*a^4*b^4*c^3*d^5*f - 230*A*B^
2*a^3*b^5*c^4*d^4*f + 142*A*B^2*a^5*b^3*c^2*d^6*f + 142*A*B^2*a^2*b^6*c^5*d
^3*f - 127*A^2*B*a^4*b^4*c^2*d^6*f - 127*A^2*B*a^2*b^6*c^4*d^4*f + 86*A*B^2
*a^5*b^3*c^4*d^4*f + 86*A*B^2*a^4*b^4*c^5*d^3*f + 80*A*B^2*a^3*b^5*c^2*d^6*
f + 80*A*B^2*a^2*b^6*c^3*d^5*f + 40*A^2*B*a^6*b^2*c^2*d^6*f + 40*A^2*B*a^2*
b^6*c^6*d^2*f + 34*A^2*B*a^5*b^3*c^3*d^5*f + 34*A^2*B*a^3*b^5*c^5*d^3*f - 3
0*A*B^2*a^6*b^2*c^3*d^5*f - 30*A*B^2*a^3*b^5*c^6*d^2*f + 20*A^2*B*a^5*b^3*c
^5*d^3*f - 15*A^2*B*a^6*b^2*c^4*d^4*f - 15*A^2*B*a^4*b^4*c^6*d^2*f - 98*B^2
*C*a^6*b^2*c*d^7*f - 98*B^2*C*a*b^7*c^6*d^2*f - 90*B*C^2*a^5*b^3*c*d^7*f -
90*B*C^2*a*b^7*c^5*d^3*f + 48*B^2*C*a^4*b^4*c*d^7*f + 48*B^2*C*a*b^7*c^4*d^
4*f + 40*B^2*C*a^2*b^6*c*d^7*f + 40*B^2*C*a*b^7*c^2*d^6*f - 32*B*C^2*a^3*b^
5*c*d^7*f - 32*B*C^2*a*b^7*c^3*d^5*f + 26*B^2*C*a^7*b*c^2*d^6*f + 26*B^2*C*
a^2*b^6*c^7*d*f - 26*B*C^2*a^7*b*c^3*d^5*f - 26*B*C^2*a^3*b^5*c^7*d*f - 8*B
^2*C*a^7*b*c^4*d^4*f - 8*B^2*C*a^4*b^4*c^7*d*f - 224*A^2*C*a^4*b^4*c*d^7*f
- 224*A^2*C*a*b^7*c^4*d^4*f - 96*A^2*C*a^2*b^6*c*d^7*f - 96*A^2*C*a*b^7*c^2
*d^6*f + 96*A*C^2*a^4*b^4*c*d^7*f + 96*A*C^2*a*b^7*c^4*d^4*f - 66*A*C^2*a^6

*b^2*c*d^7*f - 66*A*C^2*a*b^7*c^6*d^2*f + 64*A*C^2*a^2*b^6*c*d^7*f + 64*A*C
^2*a*b^7*c^2*d^6*f + 34*A^2*C*a^6*b^2*c*d^7*f + 34*A^2*C*a*b^7*c^6*d^2*f +
34*A*C^2*a^7*b*c^2*d^6*f + 34*A*C^2*a^2*b^6*c^7*d*f - 2*A^2*C*a^7*b*c^2*d^6
*f - 2*A^2*C*a^2*b^6*c^7*d*f - 208*A*B^2*a^4*b^4*c*d^7*f - 208*A*B^2*a*b^7*
c^4*d^4*f + 160*A^2*B*a^3*b^5*c*d^7*f + 160*A^2*B*a*b^7*c^3*d^5*f - 154*A^2
*B*a^5*b^3*c*d^7*f - 154*A^2*B*a*b^7*c^5*d^3*f - 112*A*B^2*a^2*b^6*c*d^7*f
- 112*A*B^2*a*b^7*c^2*d^6*f + 58*A*B^2*a^6*b^2*c*d^7*f + 58*A*B^2*a*b^7*c^6
*d^2*f - 10*A*B^2*a^7*b*c^2*d^6*f - 10*A*B^2*a^2*b^6*c^7*d*f + 6*A^2*B*a^7*
b*c^3*d^5*f + 6*A^2*B*a^3*b^5*c^7*d*f + 32*B^2*C*b^8*c^5*d^3*f - 17*B*C^2*b
^8*c^6*d^2*f + 8*B^2*C*b^8*c^3*d^5*f + 64*A^2*C*b^8*c^3*d^5*f - 32*A^2*C*b^
8*c^5*d^3*f + 32*A*C^2*b^8*c^5*d^3*f - B*C^2*a^8*c^2*d^6*f + 112*A^2*B*b^8*
c^4*d^4*f - 64*A*B^2*b^8*c^5*d^3*f + 32*B^2*C*a^5*b^3*d^8*f - 17*B*C^2*a^6*
b^2*d^8*f + 16*A^2*B*b^8*c^2*d^6*f + 16*A*B^2*b^8*c^3*d^5*f + 8*B^2*C*a^3*b
^5*d^8*f - A^2*B*b^8*c^6*d^2*f + 64*A^2*C*a^3*b^5*d^8*f - 32*A^2*C*a^5*b^3*
d^8*f + 32*A*C^2*a^5*b^3*d^8*f - A^2*B*a^8*c^2*d^6*f - B*C^2*a^2*b^6*c^8*f
+ 112*A^2*B*a^4*b^4*d^8*f - 64*A*B^2*a^5*b^3*d^8*f + 16*A^2*B*a^2*b^6*d^8*f
 + 16*A*B^2*a^3*b^5*d^8*f - A^2*B*a^6*b^2*d^8*f - A^2*B*a^2*b^6*c^8*f - 8*B
^3*a*b^7*c*d^7*f - 2*B^3*a^7*b*c*d^7*f - 2*B^3*a*b^7*c^7*d*f - 6*B^2*C*b^8*
c^7*d*f + 32*A^2*C*b^8*c*d^7*f + 6*A^2*C*b^8*c^7*d*f - 6*A*C^2*b^8*c^7*d*f
- 2*B^2*C*a^8*c*d^7*f + 16*A*B^2*b^8*c*d^7*f - 6*B^2*C*a^7*b*d^8*f - 6*A^2*
C*a^8*c*d^7*f + 6*A*C^2*a^8*c*d^7*f - 2*A*B^2*b^8*c^7*d*f + 32*A^2*C*a*b^7*
d^8*f + 6*A^2*C*a^7*b*d^8*f - 6*A*C^2*a^7*b*d^8*f - 2*B^2*C*a*b^7*c^8*f + 2
*A*B^2*a^8*c*d^7*f + 16*A*B^2*a*b^7*d^8*f - 6*A^2*C*a*b^7*c^8*f + 6*A*C^2*a
*b^7*c^8*f - 2*A*B^2*a^7*b*d^8*f + 2*A*B^2*a*b^7*c^8*f - 50*C^3*a^6*b^2*c^3
*d^5*f + 50*C^3*a^5*b^3*c^2*d^6*f - 50*C^3*a^3*b^5*c^6*d^2*f + 50*C^3*a^2*b
^6*c^5*d^3*f + 42*C^3*a^5*b^3*c^4*d^4*f + 42*C^3*a^4*b^4*c^5*d^3*f - 42*C^3
*a^4*b^4*c^3*d^5*f - 42*C^3*a^3*b^5*c^4*d^4*f - 32*C^3*a^6*b^2*c^5*d^3*f -
32*C^3*a^5*b^3*c^6*d^2*f + 32*C^3*a^3*b^5*c^2*d^6*f + 32*C^3*a^2*b^6*c^3*d^
5*f + 94*B^3*a^4*b^4*c^4*d^4*f + 48*B^3*a^2*b^6*c^2*d^6*f - 44*B^3*a^3*b^5*
c^3*d^5*f - 32*B^3*a^6*b^2*c^2*d^6*f - 32*B^3*a^2*b^6*c^6*d^2*f + 29*B^3*a^
4*b^4*c^2*d^6*f + 29*B^3*a^2*b^6*c^4*d^4*f - 20*B^3*a^5*b^3*c^5*d^3*f + 18*
B^3*a^5*b^3*c^3*d^5*f + 18*B^3*a^3*b^5*c^5*d^3*f - 3*B^3*a^6*b^2*c^4*d^4*f
- 3*B^3*a^4*b^4*c^6*d^2*f + 106*A^3*a^4*b^4*c^3*d^5*f + 106*A^3*a^3*b^5*c^4
*d^4*f - 96*A^3*a^3*b^5*c^2*d^6*f - 96*A^3*a^2*b^6*c^3*d^5*f - 82*A^3*a^5*b
^3*c^2*d^6*f - 82*A^3*a^2*b^6*c^5*d^3*f + 18*A^3*a^6*b^2*c^3*d^5*f + 18*A^3
*a^3*b^5*c^6*d^2*f - 10*A^3*a^5*b^3*c^4*d^4*f - 10*A^3*a^4*b^4*c^5*d^3*f -
22*C^3*a^7*b*c^2*d^6*f + 22*C^3*a^6*b^2*c*d^7*f - 22*C^3*a^2*b^6*c^7*d*f +
22*C^3*a*b^7*c^6*d^2*f - 2*A*B*C*b^8*c^8*f - 2*A*B*C*a^8*d^8*f + 62*B^3*a^5
*b^3*c*d^7*f + 62*B^3*a*b^7*c^5*d^3*f + 16*B^3*a^3*b^5*c*d^7*f + 16*B^3*a*b
^7*c^3*d^5*f + 6*B^3*a^7*b*c^3*d^5*f + 6*B^3*a^3*b^5*c^7*d*f + 128*A^3*a^4*
b^4*c*d^7*f + 128*A^3*a*b^7*c^4*d^4*f + 32*A^3*a^2*b^6*c*d^7*f + 32*A^3*a*b
^7*c^2*d^6*f - 10*A^3*a^7*b*c^2*d^6*f + 10*A^3*a^6*b^2*c*d^7*f - 10*A^3*a^2
*b^6*c^7*d*f + 10*A^3*a*b^7*c^6*d^2*f + 11*B^3*b^8*c^6*d^2*f - 8*B^3*b^8*c^
4*d^4*f - 4*B^3*b^8*c^2*d^6*f - 64*A^3*b^8*c^3*d^5*f - B^3*a^8*c^2*d^6*f +
11*B^3*a^6*b^2*d^8*f - 8*B^3*a^4*b^4*d^8*f - 4*B^3*a^2*b^6*d^8*f - 64*A^3*a

^3*b^5*d^8*f - B^3*a^2*b^6*c^8*f + 2*C^3*b^8*c^7*d*f - 2*C^3*a^8*c*d^7*f -
32*A^3*b^8*c*d^7*f + 2*C^3*a^7*b*d^8*f - 2*A^3*b^8*c^7*d*f - 2*C^3*a*b^7*c^
8*f + 2*A^3*a^8*c*d^7*f - 32*A^3*a*b^7*d^8*f - 2*A^3*a^7*b*d^8*f + 2*A^3*a*
b^7*c^8*f - 16*A^2*B*b^8*d^8*f + B*C^2*b^8*c^8*f + B*C^2*a^8*d^8*f + A^2*B*
b^8*c^8*f + A^2*B*a^8*d^8*f + B^3*b^8*c^8*f + B^3*a^8*d^8*f - 4*A*B^2*C*a^5
*b*c*d^5 - 4*A*B^2*C*a*b^5*c^5*d + 4*A*B^2*C*a*b^5*c*d^5 + 22*A^2*B*C*a^3*b
^3*c^2*d^4 + 22*A^2*B*C*a^2*b^4*c^3*d^3 - 20*A*B^2*C*a^3*b^3*c^3*d^3 + 14*A
*B^2*C*a^4*b^2*c^2*d^4 + 14*A*B^2*C*a^2*b^4*c^4*d^2 - 14*A*B*C^2*a^3*b^3*c^
2*d^4 - 14*A*B*C^2*a^2*b^4*c^3*d^3 + 12*A*B*C^2*a^4*b^2*c^3*d^3 + 12*A*B*C^
2*a^3*b^3*c^4*d^2 - 6*A^2*B*C*a^4*b^2*c^3*d^3 - 6*A^2*B*C*a^3*b^3*c^4*d^2 -
4*A*B^2*C*a^2*b^4*c^2*d^4 + 22*A*B*C^2*a^4*b^2*c*d^5 + 22*A*B*C^2*a*b^5*c^
4*d^2 - 20*A^2*B*C*a^4*b^2*c*d^5 - 20*A^2*B*C*a*b^5*c^4*d^2 + 10*A*B*C^2*a^
2*b^4*c*d^5 + 10*A*B*C^2*a*b^5*c^2*d^4 - 8*A^2*B*C*a^2*b^4*c*d^5 - 8*A^2*B*
C*a*b^5*c^2*d^4 + 4*A*B^2*C*a^3*b^3*c*d^5 + 4*A*B^2*C*a*b^5*c^3*d^3 - 4*A*B
*C^2*a^5*b*c^2*d^4 - 4*A*B*C^2*a^2*b^4*c^5*d + 2*A^2*B*C*a^5*b*c^2*d^4 + 2*
A^2*B*C*a^2*b^4*c^5*d - 8*B^3*C*a^4*b^2*c*d^5 - 8*B^3*C*a*b^5*c^4*d^2 - 8*B
*C^3*a^4*b^2*c*d^5 - 8*B*C^3*a*b^5*c^4*d^2 - 4*B^3*C*a^2*b^4*c*d^5 - 4*B^3*
C*a*b^5*c^2*d^4 + 4*B^2*C^2*a^5*b*c*d^5 + 4*B^2*C^2*a*b^5*c^5*d - 4*B*C^3*a
^2*b^4*c*d^5 - 4*B*C^3*a*b^5*c^2*d^4 + 2*B^3*C*a^5*b*c^2*d^4 + 2*B^3*C*a^2*
b^4*c^5*d + 2*B^2*C^2*a*b^5*c*d^5 + 2*B*C^3*a^5*b*c^2*d^4 + 2*B*C^3*a^2*b^4
*c^5*d + 24*A^3*C*a^3*b^3*c*d^5 + 24*A^3*C*a*b^5*c^3*d^3 - 24*A^2*C^2*a*b^5
*c*d^5 + 12*A^2*C^2*a^5*b*c*d^5 + 12*A^2*C^2*a*b^5*c^5*d + 8*A*C^3*a^3*b^3*
c*d^5 + 8*A*C^3*a*b^5*c^3*d^3 + 6*A^3*B*a^4*b^2*c*d^5 + 6*A^3*B*a*b^5*c^4*d
^2 - 6*A^2*B^2*a*b^5*c*d^5 + 6*A*B^3*a^4*b^2*c*d^5 + 6*A*B^3*a*b^5*c^4*d^2
+ 2*A^3*B*a^2*b^4*c*d^5 + 2*A^3*B*a*b^5*c^2*d^4 + 2*A*B^3*a^2*b^4*c*d^5 + 2
*A*B^3*a*b^5*c^2*d^4 + 20*A^2*B*C*b^6*c^3*d^3 - 10*A*B*C^2*b^6*c^3*d^3 - 2*
A*B^2*C*b^6*c^4*d^2 - 2*A*B^2*C*b^6*c^2*d^4 + 20*A^2*B*C*a^3*b^3*d^6 - 10*A
*B*C^2*a^3*b^3*d^6 - 2*A*B^2*C*a^4*b^2*d^6 - 2*A*B^2*C*a^2*b^4*d^6 + 10*B^2
*C^2*a^3*b^3*c^3*d^3 + 4*B^2*C^2*a^4*b^2*c^4*d^2 - 3*B^2*C^2*a^4*b^2*c^2*d^
4 - 3*B^2*C^2*a^2*b^4*c^4*d^2 + 2*B^2*C^2*a^2*b^4*c^2*d^4 + 40*A^2*C^2*a^2*
b^4*c^2*d^4 - 16*A^2*C^2*a^4*b^2*c^2*d^4 - 16*A^2*C^2*a^2*b^4*c^4*d^2 + 4*A
^2*C^2*a^4*b^2*c^4*d^2 + 18*A^2*B^2*a^2*b^4*c^2*d^4 + 10*A^2*B^2*a^3*b^3*c^
3*d^3 - 3*A^2*B^2*a^4*b^2*c^2*d^4 - 3*A^2*B^2*a^2*b^4*c^4*d^2 + 24*A^3*C*a*
b^5*c*d^5 - 12*A*C^3*a^5*b*c*d^5 - 12*A*C^3*a*b^5*c^5*d + 8*A*C^3*a*b^5*c*d
^5 - 4*A^3*C*a^5*b*c*d^5 - 4*A^3*C*a*b^5*c^5*d + 8*A^2*B*C*b^6*c*d^5 + 4*A*
B*C^2*b^6*c^5*d - 4*A*B*C^2*b^6*c*d^5 - 2*A^2*B*C*b^6*c^5*d + 8*A^2*B*C*a*b
^5*d^6 + 4*A*B*C^2*a^5*b*d^6 - 4*A*B*C^2*a*b^5*d^6 - 2*A^2*B*C*a^5*b*d^6 -
6*B^3*C*a^4*b^2*c^3*d^3 - 6*B^3*C*a^3*b^3*c^4*d^2 - 6*B*C^3*a^4*b^2*c^3*d^3
- 6*B*C^3*a^3*b^3*c^4*d^2 + 2*B^3*C*a^3*b^3*c^2*d^4 + 2*B^3*C*a^2*b^4*c^3*
d^3 + 2*B^2*C^2*a^3*b^3*c*d^5 + 2*B^2*C^2*a*b^5*c^3*d^3 + 2*B*C^3*a^3*b^3*c
^2*d^4 + 2*B*C^3*a^2*b^4*c^3*d^3 - 48*A^3*C*a^2*b^4*c^2*d^4 - 24*A^2*C^2*a^
3*b^3*c*d^5 - 24*A^2*C^2*a*b^5*c^3*d^3 - 16*A*C^3*a^2*b^4*c^2*d^4 + 8*A^3*C
*a^4*b^2*c^2*d^4 + 8*A^3*C*a^2*b^4*c^4*d^2 - 8*A*C^3*a^4*b^2*c^4*d^2 + 8*A*
C^3*a^4*b^2*c^2*d^4 + 8*A*C^3*a^2*b^4*c^4*d^2 - 10*A^3*B*a^3*b^3*c^2*d^4 -
10*A^3*B*a^2*b^4*c^3*d^3 - 10*A*B^3*a^3*b^3*c^2*d^4 - 10*A*B^3*a^2*b^4*c^3*

d^3 - 6*A^2*B^2*a^3*b^3*c*d^5 - 6*A^2*B^2*a*b^5*c^3*d^3 + 3*B^2*C^2*b^6*c^4
*d^2 - 8*A^2*C^2*b^6*c^4*d^2 + 8*A^2*C^2*b^6*c^2*d^4 + 9*A^2*B^2*b^6*c^2*d^
4 + 3*B^2*C^2*a^4*b^2*d^6 + 3*A^2*B^2*b^6*c^4*d^2 - 8*A^2*C^2*a^4*b^2*d^6 +
 8*A^2*C^2*a^2*b^4*d^6 + 9*A^2*B^2*a^2*b^4*d^6 + 3*A^2*B^2*a^4*b^2*d^6 + 2*
B^4*a^3*b^3*c*d^5 + 2*B^4*a*b^5*c^3*d^3 - 8*A^4*a^3*b^3*c*d^5 - 8*A^4*a*b^5
*c^3*d^3 - 16*A^3*C*b^6*c^2*d^4 + 4*A^3*C*b^6*c^4*d^2 + 4*A*C^3*b^6*c^4*d^2
 - 10*A^3*B*b^6*c^3*d^3 - 10*A*B^3*b^6*c^3*d^3 - 16*A^3*C*a^2*b^4*d^6 + 4*A
^3*C*a^4*b^2*d^6 + 4*A*C^3*a^4*b^2*d^6 - 10*A^3*B*a^3*b^3*d^6 - 10*A*B^3*a^
3*b^3*d^6 + 4*C^4*a^5*b*c*d^5 + 4*C^4*a*b^5*c^5*d + 2*B^4*a*b^5*c*d^5 - 8*A
^4*a*b^5*c*d^5 - 2*B^3*C*b^6*c^5*d - 2*B*C^3*b^6*c^5*d - 4*A^3*B*b^6*c*d^5
- 4*A*B^3*b^6*c*d^5 - 2*B^3*C*a^5*b*d^6 - 2*B*C^3*a^5*b*d^6 - 4*A^3*B*a*b^5
*d^6 - 4*A*B^3*a*b^5*d^6 + 4*C^4*a^4*b^2*c^4*d^2 + 4*C^4*a^2*b^4*c^2*d^4 +
10*B^4*a^3*b^3*c^3*d^3 - 3*B^4*a^4*b^2*c^2*d^4 - 3*B^4*a^2*b^4*c^4*d^2 - 2*
B^4*a^2*b^4*c^2*d^4 + 20*A^4*a^2*b^4*c^2*d^4 + B^2*C^2*b^6*c^2*d^4 + B^2*C^
2*a^2*b^4*d^6 - 8*A^3*C*b^6*d^6 + 3*B^4*b^6*c^4*d^2 + 8*A^4*b^6*c^2*d^4 + 3
*B^4*a^4*b^2*d^6 + 8*A^4*a^2*b^4*d^6 + 4*A^2*C^2*b^6*d^6 + 4*A^2*B^2*b^6*d^
6 + 4*A^4*b^6*d^6 + B^4*b^6*c^2*d^4 + B^4*a^2*b^4*d^6, f, k)*((16*A^2*a^3*b
^6*d^9 + A^2*a^5*b^4*d^9 + 2*A^2*a^7*b^2*d^9 + 8*B^2*a^3*b^6*d^9 + 9*B^2*a^
5*b^4*d^9 + 16*A^2*b^9*c^3*d^6 + A^2*b^9*c^5*d^4 + 2*A^2*b^9*c^7*d^2 + C^2*
a^5*b^4*d^9 + 2*C^2*a^7*b^2*d^9 + 8*B^2*b^9*c^3*d^6 + 9*B^2*b^9*c^5*d^4 + C
^2*b^9*c^5*d^4 + 2*C^2*b^9*c^7*d^2 + 16*A^2*a*b^8*d^9 + 16*A^2*b^9*c*d^8 -
14*A^2*a^2*b^7*c^3*d^6 - 4*A^2*a^2*b^7*c^5*d^4 + 6*A^2*a^2*b^7*c^7*d^2 - 14
*A^2*a^3*b^6*c^2*d^7 - 6*A^2*a^3*b^6*c^4*d^5 - 12*A^2*a^3*b^6*c^6*d^3 - 6*A
^2*a^4*b^5*c^3*d^6 + 7*A^2*a^4*b^5*c^5*d^4 - 4*A^2*a^5*b^4*c^2*d^7 + 7*A^2*
a^5*b^4*c^4*d^5 - 12*A^2*a^6*b^3*c^3*d^6 + 6*A^2*a^7*b^2*c^2*d^7 + 18*B^2*a
^2*b^7*c^3*d^6 + 2*B^2*a^2*b^7*c^5*d^4 - 4*B^2*a^2*b^7*c^7*d^2 + 18*B^2*a^3
*b^6*c^2*d^7 - 20*B^2*a^3*b^6*c^4*d^5 + 6*B^2*a^3*b^6*c^6*d^3 - 20*B^2*a^4*
b^5*c^3*d^6 - 19*B^2*a^4*b^5*c^5*d^4 + 2*B^2*a^5*b^4*c^2*d^7 - 19*B^2*a^5*b
^4*c^4*d^5 + 6*B^2*a^6*b^3*c^3*d^6 - 4*B^2*a^7*b^2*c^2*d^7 + 2*C^2*a^2*b^7*
c^3*d^6 + 12*C^2*a^2*b^7*c^5*d^4 + 6*C^2*a^2*b^7*c^7*d^2 + 2*C^2*a^3*b^6*c^
2*d^7 + 10*C^2*a^3*b^6*c^4*d^5 - 28*C^2*a^3*b^6*c^6*d^3 + 10*C^2*a^4*b^5*c^
3*d^6 + 7*C^2*a^4*b^5*c^5*d^4 + 12*C^2*a^5*b^4*c^2*d^7 + 7*C^2*a^5*b^4*c^4*
d^5 - 16*C^2*a^5*b^4*c^6*d^3 - 28*C^2*a^6*b^3*c^3*d^6 - 16*C^2*a^6*b^3*c^5*
d^4 + 6*C^2*a^7*b^2*c^2*d^7 - 24*A*B*a^2*b^7*d^9 - 24*A*B*a^4*b^5*d^9 + A*B
*a^6*b^3*d^9 + 16*A*C*a^3*b^6*d^9 + 14*A*C*a^5*b^4*d^9 - 4*A*C*a^7*b^2*d^9
- 24*A*B*b^9*c^2*d^7 - 24*A*B*b^9*c^4*d^5 + A*B*b^9*c^6*d^3 - 8*B*C*a^4*b^5
*d^9 - 9*B*C*a^6*b^3*d^9 + 16*A*C*b^9*c^3*d^6 + 14*A*C*b^9*c^5*d^4 - 4*A*C*
b^9*c^7*d^2 - 8*B*C*b^9*c^4*d^5 - 9*B*C*b^9*c^6*d^3 - A^2*a*b^8*c^8*d - A^2
*a^8*b*c*d^8 + B^2*a*b^8*c^8*d + B^2*a^8*b*c*d^8 - C^2*a*b^8*c^8*d - C^2*a^
8*b*c*d^8 - 3*A^2*a*b^8*c^4*d^5 - 8*A^2*a*b^8*c^6*d^3 - 3*A^2*a^4*b^5*c*d^8
 - 8*A^2*a^6*b^3*c*d^8 + 8*B^2*a*b^8*c^2*d^7 - 11*B^2*a*b^8*c^4*d^5 + 2*B^2
*a*b^8*c^6*d^3 + 8*B^2*a^2*b^7*c*d^8 - 11*B^2*a^4*b^5*c*d^8 + 2*B^2*a^6*b^3
*c*d^8 + 13*C^2*a*b^8*c^4*d^5 - 8*C^2*a*b^8*c^6*d^3 + 13*C^2*a^4*b^5*c*d^8
- 8*C^2*a^6*b^3*c*d^8 - A*B*a^8*b*d^9 - A*B*b^9*c^8*d + B*C*a^8*b*d^9 + B*C
*b^9*c^8*d - 16*A*B*a*b^8*c*d^8 + 2*A*C*a*b^8*c^8*d + 2*A*C*a^8*b*c*d^8 + 2

4*A*B*a*b^8*c^3*d^6 + 2*A*B*a*b^8*c^5*d^4 + 2*A*B*a*b^8*c^7*d^2 + A*B*a^2*b
^7*c^8*d + 24*A*B*a^3*b^6*c*d^8 + 2*A*B*a^5*b^4*c*d^8 + 2*A*B*a^7*b^2*c*d^8
 + A*B*a^8*b*c^2*d^7 + 16*A*C*a*b^8*c^2*d^7 - 26*A*C*a*b^8*c^4*d^5 + 16*A*C
*a^2*b^7*c*d^8 - 26*A*C*a^4*b^5*c*d^8 - 24*B*C*a*b^8*c^3*d^6 + 14*B*C*a*b^8
*c^5*d^4 - 2*B*C*a*b^8*c^7*d^2 - B*C*a^2*b^7*c^8*d - 24*B*C*a^3*b^6*c*d^8 +
 14*B*C*a^5*b^4*c*d^8 - 2*B*C*a^7*b^2*c*d^8 - B*C*a^8*b*c^2*d^7 - 64*A*B*a^
2*b^7*c^2*d^7 - 25*A*B*a^2*b^7*c^4*d^5 + 8*A*B*a^2*b^7*c^6*d^3 + 108*A*B*a^
3*b^6*c^3*d^6 + 6*A*B*a^3*b^6*c^5*d^4 - 6*A*B*a^3*b^6*c^7*d^2 - 25*A*B*a^4*
b^5*c^2*d^7 + 34*A*B*a^4*b^5*c^4*d^5 + 15*A*B*a^4*b^5*c^6*d^3 + 6*A*B*a^5*b
^4*c^3*d^6 - 20*A*B*a^5*b^4*c^5*d^4 + 8*A*B*a^6*b^3*c^2*d^7 + 15*A*B*a^6*b^
3*c^4*d^5 - 6*A*B*a^7*b^2*c^3*d^6 + 44*A*C*a^2*b^7*c^3*d^6 + 8*A*C*a^2*b^7*
c^5*d^4 - 12*A*C*a^2*b^7*c^7*d^2 + 44*A*C*a^3*b^6*c^2*d^7 - 36*A*C*a^3*b^6*
c^4*d^5 + 8*A*C*a^3*b^6*c^6*d^3 - 36*A*C*a^4*b^5*c^3*d^6 - 30*A*C*a^4*b^5*c
^5*d^4 + 8*A*C*a^5*b^4*c^2*d^7 - 30*A*C*a^5*b^4*c^4*d^5 + 8*A*C*a^6*b^3*c^3
*d^6 - 12*A*C*a^7*b^2*c^2*d^7 - 15*B*C*a^2*b^7*c^4*d^5 - 8*B*C*a^2*b^7*c^6*
d^3 - 44*B*C*a^3*b^6*c^3*d^6 + 58*B*C*a^3*b^6*c^5*d^4 + 6*B*C*a^3*b^6*c^7*d
^2 - 15*B*C*a^4*b^5*c^2*d^7 - 34*B*C*a^4*b^5*c^4*d^5 - 7*B*C*a^4*b^5*c^6*d^
3 + 58*B*C*a^5*b^4*c^3*d^6 + 68*B*C*a^5*b^4*c^5*d^4 - 8*B*C*a^6*b^3*c^2*d^7
 - 7*B*C*a^6*b^3*c^4*d^5 + 6*B*C*a^7*b^2*c^3*d^6)/(a^8*d^8 + b^8*c^8 + 2*a^
2*b^6*c^8 + a^4*b^4*c^8 + a^4*b^4*d^8 + 2*a^6*b^2*d^8 + 2*a^8*c^2*d^6 + a^8
*c^4*d^4 + b^8*c^4*d^4 + 2*b^8*c^6*d^2 - 4*a*b^7*c^3*d^5 - 8*a*b^7*c^5*d^3
- 4*a^3*b^5*c*d^7 - 8*a^3*b^5*c^7*d - 8*a^5*b^3*c*d^7 - 4*a^5*b^3*c^7*d - 8
*a^7*b*c^3*d^5 - 4*a^7*b*c^5*d^3 + 6*a^2*b^6*c^2*d^6 + 14*a^2*b^6*c^4*d^4 +
 10*a^2*b^6*c^6*d^2 - 16*a^3*b^5*c^3*d^5 - 20*a^3*b^5*c^5*d^3 + 14*a^4*b^4*
c^2*d^6 + 26*a^4*b^4*c^4*d^4 + 14*a^4*b^4*c^6*d^2 - 20*a^5*b^3*c^3*d^5 - 16
*a^5*b^3*c^5*d^3 + 10*a^6*b^2*c^2*d^6 + 14*a^6*b^2*c^4*d^4 + 6*a^6*b^2*c^6*
d^2 - 4*a*b^7*c^7*d - 4*a^7*b*c*d^7) + root(144*a^13*b*c^5*d^9*f^4 + 144*a^
9*b^5*c*d^13*f^4 + 144*a^5*b^9*c^13*d*f^4 + 144*a*b^13*c^9*d^5*f^4 + 96*a^1
3*b*c^7*d^7*f^4 + 96*a^13*b*c^3*d^11*f^4 + 96*a^11*b^3*c*d^13*f^4 + 96*a^7*
b^7*c^13*d*f^4 + 96*a^7*b^7*c*d^13*f^4 + 96*a^3*b^11*c^13*d*f^4 + 96*a*b^13
*c^11*d^3*f^4 + 96*a*b^13*c^7*d^7*f^4 + 24*a^13*b*c^9*d^5*f^4 + 24*a^9*b^5*
c^13*d*f^4 + 24*a^5*b^9*c*d^13*f^4 + 24*a*b^13*c^5*d^9*f^4 + 24*a^13*b*c*d^
13*f^4 + 24*a*b^13*c^13*d*f^4 + 3648*a^7*b^7*c^7*d^7*f^4 - 3188*a^8*b^6*c^6
*d^8*f^4 - 3188*a^6*b^8*c^8*d^6*f^4 - 2912*a^8*b^6*c^8*d^6*f^4 - 2912*a^6*b
^8*c^6*d^8*f^4 + 2592*a^9*b^5*c^7*d^7*f^4 + 2592*a^7*b^7*c^9*d^5*f^4 + 2592
*a^7*b^7*c^5*d^9*f^4 + 2592*a^5*b^9*c^7*d^7*f^4 + 2168*a^9*b^5*c^5*d^9*f^4
+ 2168*a^5*b^9*c^9*d^5*f^4 - 1776*a^10*b^4*c^6*d^8*f^4 - 1776*a^8*b^6*c^4*d
^10*f^4 - 1776*a^6*b^8*c^10*d^4*f^4 - 1776*a^4*b^10*c^8*d^6*f^4 + 1568*a^9*
b^5*c^9*d^5*f^4 + 1568*a^5*b^9*c^5*d^9*f^4 - 1344*a^10*b^4*c^8*d^6*f^4 - 13
44*a^8*b^6*c^10*d^4*f^4 - 1344*a^6*b^8*c^4*d^10*f^4 - 1344*a^4*b^10*c^6*d^8
*f^4 - 1164*a^10*b^4*c^4*d^10*f^4 - 1164*a^4*b^10*c^10*d^4*f^4 + 896*a^11*b
^3*c^5*d^9*f^4 + 896*a^9*b^5*c^3*d^11*f^4 + 896*a^5*b^9*c^11*d^3*f^4 + 896*
a^3*b^11*c^9*d^5*f^4 + 864*a^11*b^3*c^7*d^7*f^4 + 864*a^7*b^7*c^11*d^3*f^4
+ 864*a^7*b^7*c^3*d^11*f^4 + 864*a^3*b^11*c^7*d^7*f^4 - 480*a^10*b^4*c^10*d
^4*f^4 - 480*a^4*b^10*c^4*d^10*f^4 + 464*a^11*b^3*c^3*d^11*f^4 + 464*a^3*b^

11*c^11*d^3*f^4 - 424*a^12*b^2*c^6*d^8*f^4 - 424*a^8*b^6*c^2*d^12*f^4 - 424
*a^6*b^8*c^12*d^2*f^4 - 424*a^2*b^12*c^8*d^6*f^4 + 416*a^11*b^3*c^9*d^5*f^4
 + 416*a^9*b^5*c^11*d^3*f^4 + 416*a^5*b^9*c^3*d^11*f^4 + 416*a^3*b^11*c^5*d
^9*f^4 - 336*a^12*b^2*c^4*d^10*f^4 - 336*a^10*b^4*c^2*d^12*f^4 - 336*a^4*b^
10*c^12*d^2*f^4 - 336*a^2*b^12*c^10*d^4*f^4 - 256*a^12*b^2*c^8*d^6*f^4 - 25
6*a^8*b^6*c^12*d^2*f^4 - 256*a^6*b^8*c^2*d^12*f^4 - 256*a^2*b^12*c^6*d^8*f^
4 - 124*a^12*b^2*c^2*d^12*f^4 - 124*a^2*b^12*c^12*d^2*f^4 + 80*a^11*b^3*c^1
1*d^3*f^4 + 80*a^3*b^11*c^3*d^11*f^4 - 60*a^12*b^2*c^10*d^4*f^4 - 60*a^10*b
^4*c^12*d^2*f^4 - 60*a^4*b^10*c^2*d^12*f^4 - 60*a^2*b^12*c^4*d^10*f^4 - 24*
b^14*c^10*d^4*f^4 - 16*b^14*c^12*d^2*f^4 - 16*b^14*c^8*d^6*f^4 - 4*b^14*c^6
*d^8*f^4 - 24*a^14*c^4*d^10*f^4 - 16*a^14*c^6*d^8*f^4 - 16*a^14*c^2*d^12*f^
4 - 4*a^14*c^8*d^6*f^4 - 24*a^10*b^4*d^14*f^4 - 16*a^12*b^2*d^14*f^4 - 16*a
^8*b^6*d^14*f^4 - 4*a^6*b^8*d^14*f^4 - 24*a^4*b^10*c^14*f^4 - 16*a^6*b^8*c^
14*f^4 - 16*a^2*b^12*c^14*f^4 - 4*a^8*b^6*c^14*f^4 - 4*b^14*c^14*f^4 - 4*a^
14*d^14*f^4 + 36*A*C*a^9*b*c*d^9*f^2 + 36*A*C*a*b^9*c^9*d*f^2 + 32*A*C*a*b^
9*c*d^9*f^2 - 552*B*C*a^7*b^3*c^4*d^6*f^2 - 552*B*C*a^4*b^6*c^7*d^3*f^2 - 4
08*B*C*a^5*b^5*c^4*d^6*f^2 - 408*B*C*a^4*b^6*c^5*d^5*f^2 + 360*B*C*a^6*b^4*
c^3*d^7*f^2 + 360*B*C*a^3*b^7*c^6*d^4*f^2 - 248*B*C*a^7*b^3*c^2*d^8*f^2 - 2
48*B*C*a^2*b^8*c^7*d^3*f^2 + 184*B*C*a^6*b^4*c^5*d^5*f^2 + 184*B*C*a^5*b^5*
c^6*d^4*f^2 + 152*B*C*a^8*b^2*c^3*d^7*f^2 - 152*B*C*a^5*b^5*c^2*d^8*f^2 + 1
52*B*C*a^3*b^7*c^8*d^2*f^2 - 152*B*C*a^2*b^8*c^5*d^5*f^2 - 104*B*C*a^7*b^3*
c^6*d^4*f^2 - 104*B*C*a^6*b^4*c^7*d^3*f^2 + 64*B*C*a^8*b^2*c^5*d^5*f^2 + 64
*B*C*a^5*b^5*c^8*d^2*f^2 - 56*B*C*a^4*b^6*c^3*d^7*f^2 - 56*B*C*a^3*b^7*c^4*
d^6*f^2 - 24*B*C*a^8*b^2*c^7*d^3*f^2 - 24*B*C*a^7*b^3*c^8*d^2*f^2 - 24*B*C*
a^3*b^7*c^2*d^8*f^2 - 24*B*C*a^2*b^8*c^3*d^7*f^2 - 696*A*C*a^5*b^5*c^5*d^5*
f^2 + 536*A*C*a^6*b^4*c^6*d^4*f^2 + 536*A*C*a^6*b^4*c^4*d^6*f^2 + 536*A*C*a
^4*b^6*c^6*d^4*f^2 + 472*A*C*a^4*b^6*c^4*d^6*f^2 - 232*A*C*a^7*b^3*c^5*d^5*
f^2 - 232*A*C*a^5*b^5*c^7*d^3*f^2 + 216*A*C*a^3*b^7*c^3*d^7*f^2 + 168*A*C*a
^7*b^3*c^3*d^7*f^2 + 168*A*C*a^3*b^7*c^7*d^3*f^2 - 154*A*C*a^8*b^2*c^2*d^8*
f^2 - 154*A*C*a^2*b^8*c^8*d^2*f^2 + 62*A*C*a^8*b^2*c^6*d^4*f^2 + 62*A*C*a^6
*b^4*c^8*d^2*f^2 - 40*A*C*a^7*b^3*c^7*d^3*f^2 - 40*A*C*a^5*b^5*c^3*d^7*f^2
- 40*A*C*a^3*b^7*c^5*d^5*f^2 + 32*A*C*a^6*b^4*c^2*d^8*f^2 + 32*A*C*a^2*b^8*
c^6*d^4*f^2 - 32*A*C*a^2*b^8*c^2*d^8*f^2 + 30*A*C*a^4*b^6*c^2*d^8*f^2 + 30*
A*C*a^2*b^8*c^4*d^6*f^2 + 16*A*C*a^8*b^2*c^4*d^6*f^2 + 16*A*C*a^4*b^6*c^8*d
^2*f^2 - 488*A*B*a^6*b^4*c^3*d^7*f^2 - 488*A*B*a^3*b^7*c^6*d^4*f^2 + 440*A*
B*a^7*b^3*c^4*d^6*f^2 + 440*A*B*a^4*b^6*c^7*d^3*f^2 - 360*A*B*a^6*b^4*c^5*d
^5*f^2 - 360*A*B*a^5*b^5*c^6*d^4*f^2 - 192*A*B*a^8*b^2*c^3*d^7*f^2 - 192*A*
B*a^3*b^7*c^8*d^2*f^2 - 168*A*B*a^3*b^7*c^2*d^8*f^2 - 168*A*B*a^2*b^8*c^3*d
^7*f^2 - 152*A*B*a^4*b^6*c^3*d^7*f^2 - 152*A*B*a^3*b^7*c^4*d^6*f^2 - 120*A*
B*a^8*b^2*c^5*d^5*f^2 + 120*A*B*a^7*b^3*c^2*d^8*f^2 - 120*A*B*a^5*b^5*c^8*d
^2*f^2 + 120*A*B*a^5*b^5*c^4*d^6*f^2 - 120*A*B*a^5*b^5*c^2*d^8*f^2 + 120*A*
B*a^4*b^6*c^5*d^5*f^2 + 120*A*B*a^2*b^8*c^7*d^3*f^2 - 120*A*B*a^2*b^8*c^5*d
^5*f^2 + 40*A*B*a^7*b^3*c^6*d^4*f^2 + 40*A*B*a^6*b^4*c^7*d^3*f^2 - 72*B*C*a
^9*b*c^4*d^6*f^2 - 72*B*C*a^4*b^6*c^9*d*f^2 - 64*B*C*a^4*b^6*c*d^9*f^2 - 64
*B*C*a*b^9*c^4*d^6*f^2 - 32*B*C*a^8*b^2*c*d^9*f^2 - 32*B*C*a*b^9*c^8*d^2*f^

2 - 16*B*C*a^2*b^8*c*d^9*f^2 - 16*B*C*a*b^9*c^2*d^8*f^2 + 8*B*C*a^9*b*c^6*d
^4*f^2 - 8*B*C*a^9*b*c^2*d^8*f^2 + 8*B*C*a^6*b^4*c^9*d*f^2 - 8*B*C*a^2*b^8*
c^9*d*f^2 + 104*A*C*a^7*b^3*c*d^9*f^2 + 104*A*C*a*b^9*c^7*d^3*f^2 + 96*A*C*
a^3*b^7*c*d^9*f^2 + 96*A*C*a*b^9*c^3*d^7*f^2 + 72*A*C*a^9*b*c^3*d^7*f^2 + 7
2*A*C*a^3*b^7*c^9*d*f^2 + 68*A*C*a^5*b^5*c*d^9*f^2 + 68*A*C*a*b^9*c^5*d^5*f
^2 - 28*A*C*a^9*b*c^5*d^5*f^2 - 28*A*C*a^5*b^5*c^9*d*f^2 + 80*A*B*a^9*b*c^4
*d^6*f^2 + 80*A*B*a^4*b^6*c^9*d*f^2 + 24*A*B*a^8*b^2*c*d^9*f^2 - 24*A*B*a^6
*b^4*c*d^9*f^2 + 24*A*B*a^4*b^6*c*d^9*f^2 - 24*A*B*a^2*b^8*c*d^9*f^2 + 24*A
*B*a*b^9*c^8*d^2*f^2 - 24*A*B*a*b^9*c^6*d^4*f^2 + 24*A*B*a*b^9*c^4*d^6*f^2
- 24*A*B*a*b^9*c^2*d^8*f^2 - 32*B*C*b^10*c^7*d^3*f^2 - 8*B*C*b^10*c^5*d^5*f
^2 + 34*A*C*b^10*c^6*d^4*f^2 + 16*B*C*a^10*c^3*d^7*f^2 + 16*A*C*b^10*c^4*d^
6*f^2 - 12*A*C*b^10*c^8*d^2*f^2 - 96*A*B*b^10*c^5*d^5*f^2 - 72*A*B*b^10*c^3
*d^7*f^2 - 32*B*C*a^7*b^3*d^10*f^2 - 28*A*C*a^10*c^2*d^8*f^2 - 24*A*B*b^10*
c^7*d^3*f^2 - 8*B*C*a^5*b^5*d^10*f^2 + 2*A*C*a^10*c^4*d^6*f^2 + 34*A*C*a^6*
b^4*d^10*f^2 + 16*B*C*a^3*b^7*c^10*f^2 + 16*A*C*a^4*b^6*d^10*f^2 - 16*A*B*a
^10*c^3*d^7*f^2 - 12*A*C*a^8*b^2*d^10*f^2 - 96*A*B*a^5*b^5*d^10*f^2 - 72*A*
B*a^3*b^7*d^10*f^2 - 28*A*C*a^2*b^8*c^10*f^2 - 24*A*B*a^7*b^3*d^10*f^2 + 2*
A*C*a^4*b^6*c^10*f^2 - 16*A*B*a^3*b^7*c^10*f^2 + 444*C^2*a^5*b^5*c^5*d^5*f^
2 + 148*C^2*a^7*b^3*c^5*d^5*f^2 + 148*C^2*a^5*b^5*c^7*d^3*f^2 + 148*C^2*a^5
*b^5*c^3*d^7*f^2 + 148*C^2*a^3*b^7*c^5*d^5*f^2 - 140*C^2*a^6*b^4*c^6*d^4*f^
2 - 140*C^2*a^6*b^4*c^4*d^6*f^2 - 140*C^2*a^4*b^6*c^6*d^4*f^2 - 140*C^2*a^4
*b^6*c^4*d^6*f^2 + 109*C^2*a^8*b^2*c^2*d^8*f^2 + 109*C^2*a^2*b^8*c^8*d^2*f^
2 + 48*C^2*a^8*b^2*c^4*d^6*f^2 + 48*C^2*a^6*b^4*c^2*d^8*f^2 + 48*C^2*a^4*b^
6*c^8*d^2*f^2 + 48*C^2*a^2*b^8*c^6*d^4*f^2 + 20*C^2*a^7*b^3*c^7*d^3*f^2 - 2
0*C^2*a^7*b^3*c^3*d^7*f^2 - 20*C^2*a^3*b^7*c^7*d^3*f^2 + 20*C^2*a^3*b^7*c^3
*d^7*f^2 + 17*C^2*a^8*b^2*c^6*d^4*f^2 + 17*C^2*a^6*b^4*c^8*d^2*f^2 + 17*C^2
*a^4*b^6*c^2*d^8*f^2 + 17*C^2*a^2*b^8*c^4*d^6*f^2 + 16*C^2*a^8*b^2*c^8*d^2*
f^2 + 16*C^2*a^2*b^8*c^2*d^8*f^2 - 396*B^2*a^5*b^5*c^5*d^5*f^2 + 308*B^2*a^
6*b^4*c^4*d^6*f^2 + 308*B^2*a^4*b^6*c^6*d^4*f^2 + 300*B^2*a^4*b^6*c^4*d^6*f
^2 + 284*B^2*a^6*b^4*c^6*d^4*f^2 - 132*B^2*a^7*b^3*c^5*d^5*f^2 - 132*B^2*a^
5*b^5*c^7*d^3*f^2 - 84*B^2*a^5*b^5*c^3*d^7*f^2 - 84*B^2*a^3*b^7*c^5*d^5*f^2
 + 61*B^2*a^4*b^6*c^2*d^8*f^2 + 61*B^2*a^2*b^8*c^4*d^6*f^2 - 59*B^2*a^8*b^2
*c^2*d^8*f^2 - 59*B^2*a^2*b^8*c^8*d^2*f^2 + 56*B^2*a^6*b^4*c^2*d^8*f^2 + 56
*B^2*a^2*b^8*c^6*d^4*f^2 + 52*B^2*a^7*b^3*c^3*d^7*f^2 + 52*B^2*a^3*b^7*c^7*
d^3*f^2 + 44*B^2*a^3*b^7*c^3*d^7*f^2 + 33*B^2*a^8*b^2*c^6*d^4*f^2 + 33*B^2*
a^6*b^4*c^8*d^2*f^2 + 20*B^2*a^8*b^2*c^4*d^6*f^2 - 20*B^2*a^7*b^3*c^7*d^3*f
^2 + 20*B^2*a^4*b^6*c^8*d^2*f^2 + 8*B^2*a^2*b^8*c^2*d^8*f^2 + 337*A^2*a^4*b
^6*c^2*d^8*f^2 + 337*A^2*a^2*b^8*c^4*d^6*f^2 + 272*A^2*a^2*b^8*c^2*d^8*f^2
+ 252*A^2*a^5*b^5*c^5*d^5*f^2 + 244*A^2*a^4*b^6*c^4*d^6*f^2 - 236*A^2*a^3*b
^7*c^3*d^7*f^2 + 176*A^2*a^6*b^4*c^2*d^8*f^2 + 176*A^2*a^2*b^8*c^6*d^4*f^2
- 148*A^2*a^7*b^3*c^3*d^7*f^2 - 148*A^2*a^3*b^7*c^7*d^3*f^2 - 140*A^2*a^6*b
^4*c^6*d^4*f^2 + 109*A^2*a^8*b^2*c^2*d^8*f^2 + 109*A^2*a^2*b^8*c^8*d^2*f^2
- 108*A^2*a^5*b^5*c^3*d^7*f^2 - 108*A^2*a^3*b^7*c^5*d^5*f^2 + 84*A^2*a^7*b^
3*c^5*d^5*f^2 + 84*A^2*a^5*b^5*c^7*d^3*f^2 + 32*A^2*a^8*b^2*c^4*d^6*f^2 + 3
2*A^2*a^4*b^6*c^8*d^2*f^2 + 20*A^2*a^7*b^3*c^7*d^3*f^2 - 15*A^2*a^8*b^2*c^6

*d^4*f^2 - 15*A^2*a^6*b^4*c^8*d^2*f^2 - 12*A^2*a^6*b^4*c^4*d^6*f^2 - 12*A^2
*a^4*b^6*c^6*d^4*f^2 + 8*B*C*b^10*c^9*d*f^2 - 16*B*C*a^10*c*d^9*f^2 - 16*A*
B*b^10*c^9*d*f^2 - 16*A*B*b^10*c*d^9*f^2 + 8*B*C*a^9*b*d^10*f^2 - 16*B*C*a*
b^9*c^10*f^2 + 16*A*B*a^10*c*d^9*f^2 - 16*A*B*a^9*b*d^10*f^2 - 16*A*B*a*b^9
*d^10*f^2 + 16*A*B*a*b^9*c^10*f^2 + 22*C^2*a^9*b*c^5*d^5*f^2 + 22*C^2*a^5*b
^5*c^9*d*f^2 + 22*C^2*a^5*b^5*c*d^9*f^2 + 22*C^2*a*b^9*c^5*d^5*f^2 - 20*C^2
*a^9*b*c^3*d^7*f^2 - 20*C^2*a^7*b^3*c*d^9*f^2 - 20*C^2*a^3*b^7*c^9*d*f^2 -
20*C^2*a*b^9*c^7*d^3*f^2 + 36*B^2*a^7*b^3*c*d^9*f^2 + 36*B^2*a*b^9*c^7*d^3*
f^2 + 28*B^2*a^9*b*c^3*d^7*f^2 + 28*B^2*a^3*b^7*c^9*d*f^2 + 24*B^2*a^3*b^7*
c*d^9*f^2 + 24*B^2*a*b^9*c^3*d^7*f^2 - 18*B^2*a^9*b*c^5*d^5*f^2 - 18*B^2*a^
5*b^5*c^9*d*f^2 + 6*B^2*a^5*b^5*c*d^9*f^2 + 6*B^2*a*b^9*c^5*d^5*f^2 - 96*A^
2*a^3*b^7*c*d^9*f^2 - 96*A^2*a*b^9*c^3*d^7*f^2 - 90*A^2*a^5*b^5*c*d^9*f^2 -
90*A^2*a*b^9*c^5*d^5*f^2 - 84*A^2*a^7*b^3*c*d^9*f^2 - 84*A^2*a*b^9*c^7*d^3
*f^2 - 52*A^2*a^9*b*c^3*d^7*f^2 - 52*A^2*a^3*b^7*c^9*d*f^2 + 6*A^2*a^9*b*c^
5*d^5*f^2 + 6*A^2*a^5*b^5*c^9*d*f^2 - 10*C^2*a^9*b*c*d^9*f^2 - 10*C^2*a*b^9
*c^9*d*f^2 + 14*B^2*a^9*b*c*d^9*f^2 + 14*B^2*a*b^9*c^9*d*f^2 + 8*B^2*a*b^9*
c*d^9*f^2 - 32*A^2*a*b^9*c*d^9*f^2 - 26*A^2*a^9*b*c*d^9*f^2 - 26*A^2*a*b^9*
c^9*d*f^2 + 2*A*C*b^10*c^10*f^2 + 2*A*C*a^10*d^10*f^2 + 14*C^2*b^10*c^8*d^2
*f^2 - C^2*b^10*c^6*d^4*f^2 + 31*B^2*b^10*c^6*d^4*f^2 + 20*B^2*b^10*c^4*d^6
*f^2 + 14*C^2*a^10*c^2*d^8*f^2 + 4*B^2*b^10*c^2*d^8*f^2 + 2*B^2*b^10*c^8*d^
2*f^2 - C^2*a^10*c^4*d^6*f^2 + 80*A^2*b^10*c^4*d^6*f^2 + 64*A^2*b^10*c^2*d^
8*f^2 + 31*A^2*b^10*c^6*d^4*f^2 + 14*C^2*a^8*b^2*d^10*f^2 + 14*A^2*b^10*c^8
*d^2*f^2 - 10*B^2*a^10*c^2*d^8*f^2 + 3*B^2*a^10*c^4*d^6*f^2 - C^2*a^6*b^4*d
^10*f^2 + 31*B^2*a^6*b^4*d^10*f^2 + 20*B^2*a^4*b^6*d^10*f^2 + 14*C^2*a^2*b^
8*c^10*f^2 + 14*A^2*a^10*c^2*d^8*f^2 + 4*B^2*a^2*b^8*d^10*f^2 + 2*B^2*a^8*b
^2*d^10*f^2 - C^2*a^4*b^6*c^10*f^2 - A^2*a^10*c^4*d^6*f^2 + 80*A^2*a^4*b^6*
d^10*f^2 + 64*A^2*a^2*b^8*d^10*f^2 + 31*A^2*a^6*b^4*d^10*f^2 + 14*A^2*a^8*b
^2*d^10*f^2 - 10*B^2*a^2*b^8*c^10*f^2 + 3*B^2*a^4*b^6*c^10*f^2 + 14*A^2*a^2
*b^8*c^10*f^2 - A^2*a^4*b^6*c^10*f^2 - C^2*b^10*c^10*f^2 - C^2*a^10*d^10*f^
2 + 16*A^2*b^10*d^10*f^2 + 3*B^2*b^10*c^10*f^2 + 3*B^2*a^10*d^10*f^2 - A^2*
b^10*c^10*f^2 - A^2*a^10*d^10*f^2 - 96*A*B*C*a*b^7*c*d^7*f - 28*A*B*C*a^7*b
*c*d^7*f - 28*A*B*C*a*b^7*c^7*d*f + 484*A*B*C*a^4*b^4*c^4*d^4*f - 424*A*B*C
*a^3*b^5*c^3*d^5*f + 320*A*B*C*a^2*b^6*c^2*d^6*f - 176*A*B*C*a^6*b^2*c^2*d^
6*f - 176*A*B*C*a^2*b^6*c^6*d^2*f + 158*A*B*C*a^4*b^4*c^2*d^6*f + 158*A*B*C
*a^2*b^6*c^4*d^4*f - 136*A*B*C*a^5*b^3*c^5*d^3*f - 34*A*B*C*a^6*b^2*c^4*d^4
*f - 34*A*B*C*a^4*b^4*c^6*d^2*f + 28*A*B*C*a^5*b^3*c^3*d^5*f + 28*A*B*C*a^3
*b^5*c^5*d^3*f + 308*A*B*C*a^5*b^3*c*d^7*f + 308*A*B*C*a*b^7*c^5*d^3*f + 20
*A*B*C*a^7*b*c^3*d^5*f + 20*A*B*C*a^3*b^5*c^7*d*f + 30*B*C^2*a^7*b*c*d^7*f
+ 30*B*C^2*a*b^7*c^7*d*f + 160*A^2*B*a*b^7*c*d^7*f - 2*A^2*B*a^7*b*c*d^7*f
- 2*A^2*B*a*b^7*c^7*d*f - 96*A*B*C*b^8*c^4*d^4*f + 34*A*B*C*b^8*c^6*d^2*f -
32*A*B*C*b^8*c^2*d^6*f + 2*A*B*C*a^8*c^2*d^6*f - 96*A*B*C*a^4*b^4*d^8*f +
34*A*B*C*a^6*b^2*d^8*f - 32*A*B*C*a^2*b^6*d^8*f + 2*A*B*C*a^2*b^6*c^8*f - 2
10*B*C^2*a^4*b^4*c^4*d^4*f - 182*B^2*C*a^5*b^3*c^2*d^6*f - 182*B^2*C*a^2*b^
6*c^5*d^3*f + 180*B*C^2*a^5*b^3*c^5*d^3*f + 180*B*C^2*a^3*b^5*c^3*d^5*f - 1
66*B^2*C*a^5*b^3*c^4*d^4*f - 166*B^2*C*a^4*b^4*c^5*d^3*f + 152*B*C^2*a^6*b^

2*c^2*d^6*f + 152*B*C^2*a^2*b^6*c^6*d^2*f - 112*B^2*C*a^3*b^5*c^2*d^6*f - 1
12*B^2*C*a^2*b^6*c^3*d^5*f + 94*B^2*C*a^4*b^4*c^3*d^5*f + 94*B^2*C*a^3*b^5*
c^4*d^4*f - 80*B*C^2*a^2*b^6*c^2*d^6*f + 66*B*C^2*a^5*b^3*c^3*d^5*f + 66*B*
C^2*a^3*b^5*c^5*d^3*f + 46*B^2*C*a^6*b^2*c^3*d^5*f + 46*B^2*C*a^3*b^5*c^6*d
^2*f + 33*B*C^2*a^6*b^2*c^4*d^4*f + 33*B*C^2*a^4*b^4*c^6*d^2*f + 24*B^2*C*a
^6*b^2*c^5*d^3*f + 24*B^2*C*a^5*b^3*c^6*d^2*f - 16*B*C^2*a^6*b^2*c^6*d^2*f
- 15*B*C^2*a^4*b^4*c^2*d^6*f - 15*B*C^2*a^2*b^6*c^4*d^4*f - 190*A^2*C*a^4*b
^4*c^3*d^5*f - 190*A^2*C*a^3*b^5*c^4*d^4*f + 182*A^2*C*a^5*b^3*c^2*d^6*f +
182*A^2*C*a^2*b^6*c^5*d^3*f + 160*A^2*C*a^3*b^5*c^2*d^6*f + 160*A^2*C*a^2*b
^6*c^3*d^5*f - 150*A*C^2*a^5*b^3*c^2*d^6*f - 150*A*C^2*a^2*b^6*c^5*d^3*f -
126*A*C^2*a^5*b^3*c^4*d^4*f - 126*A*C^2*a^4*b^4*c^5*d^3*f + 126*A*C^2*a^4*b
^4*c^3*d^5*f + 126*A*C^2*a^3*b^5*c^4*d^4*f - 96*A*C^2*a^3*b^5*c^2*d^6*f - 9
6*A*C^2*a^2*b^6*c^3*d^5*f + 94*A^2*C*a^5*b^3*c^4*d^4*f + 94*A^2*C*a^4*b^4*c
^5*d^3*f + 54*A*C^2*a^6*b^2*c^3*d^5*f + 54*A*C^2*a^3*b^5*c^6*d^2*f + 32*A*C
^2*a^6*b^2*c^5*d^3*f + 32*A*C^2*a^5*b^3*c^6*d^2*f - 22*A^2*C*a^6*b^2*c^3*d^
5*f - 22*A^2*C*a^3*b^5*c^6*d^2*f + 500*A^2*B*a^3*b^5*c^3*d^5*f - 290*A^2*B*
a^4*b^4*c^4*d^4*f - 256*A^2*B*a^2*b^6*c^2*d^6*f - 230*A*B^2*a^4*b^4*c^3*d^5
*f - 230*A*B^2*a^3*b^5*c^4*d^4*f + 142*A*B^2*a^5*b^3*c^2*d^6*f + 142*A*B^2*
a^2*b^6*c^5*d^3*f - 127*A^2*B*a^4*b^4*c^2*d^6*f - 127*A^2*B*a^2*b^6*c^4*d^4
*f + 86*A*B^2*a^5*b^3*c^4*d^4*f + 86*A*B^2*a^4*b^4*c^5*d^3*f + 80*A*B^2*a^3
*b^5*c^2*d^6*f + 80*A*B^2*a^2*b^6*c^3*d^5*f + 40*A^2*B*a^6*b^2*c^2*d^6*f +
40*A^2*B*a^2*b^6*c^6*d^2*f + 34*A^2*B*a^5*b^3*c^3*d^5*f + 34*A^2*B*a^3*b^5*
c^5*d^3*f - 30*A*B^2*a^6*b^2*c^3*d^5*f - 30*A*B^2*a^3*b^5*c^6*d^2*f + 20*A^
2*B*a^5*b^3*c^5*d^3*f - 15*A^2*B*a^6*b^2*c^4*d^4*f - 15*A^2*B*a^4*b^4*c^6*d
^2*f - 98*B^2*C*a^6*b^2*c*d^7*f - 98*B^2*C*a*b^7*c^6*d^2*f - 90*B*C^2*a^5*b
^3*c*d^7*f - 90*B*C^2*a*b^7*c^5*d^3*f + 48*B^2*C*a^4*b^4*c*d^7*f + 48*B^2*C
*a*b^7*c^4*d^4*f + 40*B^2*C*a^2*b^6*c*d^7*f + 40*B^2*C*a*b^7*c^2*d^6*f - 32
*B*C^2*a^3*b^5*c*d^7*f - 32*B*C^2*a*b^7*c^3*d^5*f + 26*B^2*C*a^7*b*c^2*d^6*
f + 26*B^2*C*a^2*b^6*c^7*d*f - 26*B*C^2*a^7*b*c^3*d^5*f - 26*B*C^2*a^3*b^5*
c^7*d*f - 8*B^2*C*a^7*b*c^4*d^4*f - 8*B^2*C*a^4*b^4*c^7*d*f - 224*A^2*C*a^4
*b^4*c*d^7*f - 224*A^2*C*a*b^7*c^4*d^4*f - 96*A^2*C*a^2*b^6*c*d^7*f - 96*A^
2*C*a*b^7*c^2*d^6*f + 96*A*C^2*a^4*b^4*c*d^7*f + 96*A*C^2*a*b^7*c^4*d^4*f -
66*A*C^2*a^6*b^2*c*d^7*f - 66*A*C^2*a*b^7*c^6*d^2*f + 64*A*C^2*a^2*b^6*c*d
^7*f + 64*A*C^2*a*b^7*c^2*d^6*f + 34*A^2*C*a^6*b^2*c*d^7*f + 34*A^2*C*a*b^7
*c^6*d^2*f + 34*A*C^2*a^7*b*c^2*d^6*f + 34*A*C^2*a^2*b^6*c^7*d*f - 2*A^2*C*
a^7*b*c^2*d^6*f - 2*A^2*C*a^2*b^6*c^7*d*f - 208*A*B^2*a^4*b^4*c*d^7*f - 208
*A*B^2*a*b^7*c^4*d^4*f + 160*A^2*B*a^3*b^5*c*d^7*f + 160*A^2*B*a*b^7*c^3*d^
5*f - 154*A^2*B*a^5*b^3*c*d^7*f - 154*A^2*B*a*b^7*c^5*d^3*f - 112*A*B^2*a^2
*b^6*c*d^7*f - 112*A*B^2*a*b^7*c^2*d^6*f + 58*A*B^2*a^6*b^2*c*d^7*f + 58*A*
B^2*a*b^7*c^6*d^2*f - 10*A*B^2*a^7*b*c^2*d^6*f - 10*A*B^2*a^2*b^6*c^7*d*f +
6*A^2*B*a^7*b*c^3*d^5*f + 6*A^2*B*a^3*b^5*c^7*d*f + 32*B^2*C*b^8*c^5*d^3*f
- 17*B*C^2*b^8*c^6*d^2*f + 8*B^2*C*b^8*c^3*d^5*f + 64*A^2*C*b^8*c^3*d^5*f
- 32*A^2*C*b^8*c^5*d^3*f + 32*A*C^2*b^8*c^5*d^3*f - B*C^2*a^8*c^2*d^6*f + 1
12*A^2*B*b^8*c^4*d^4*f - 64*A*B^2*b^8*c^5*d^3*f + 32*B^2*C*a^5*b^3*d^8*f -
17*B*C^2*a^6*b^2*d^8*f + 16*A^2*B*b^8*c^2*d^6*f + 16*A*B^2*b^8*c^3*d^5*f +

8*B^2*C*a^3*b^5*d^8*f - A^2*B*b^8*c^6*d^2*f + 64*A^2*C*a^3*b^5*d^8*f - 32*A
^2*C*a^5*b^3*d^8*f + 32*A*C^2*a^5*b^3*d^8*f - A^2*B*a^8*c^2*d^6*f - B*C^2*a
^2*b^6*c^8*f + 112*A^2*B*a^4*b^4*d^8*f - 64*A*B^2*a^5*b^3*d^8*f + 16*A^2*B*
a^2*b^6*d^8*f + 16*A*B^2*a^3*b^5*d^8*f - A^2*B*a^6*b^2*d^8*f - A^2*B*a^2*b^
6*c^8*f - 8*B^3*a*b^7*c*d^7*f - 2*B^3*a^7*b*c*d^7*f - 2*B^3*a*b^7*c^7*d*f -
 6*B^2*C*b^8*c^7*d*f + 32*A^2*C*b^8*c*d^7*f + 6*A^2*C*b^8*c^7*d*f - 6*A*C^2
*b^8*c^7*d*f - 2*B^2*C*a^8*c*d^7*f + 16*A*B^2*b^8*c*d^7*f - 6*B^2*C*a^7*b*d
^8*f - 6*A^2*C*a^8*c*d^7*f + 6*A*C^2*a^8*c*d^7*f - 2*A*B^2*b^8*c^7*d*f + 32
*A^2*C*a*b^7*d^8*f + 6*A^2*C*a^7*b*d^8*f - 6*A*C^2*a^7*b*d^8*f - 2*B^2*C*a*
b^7*c^8*f + 2*A*B^2*a^8*c*d^7*f + 16*A*B^2*a*b^7*d^8*f - 6*A^2*C*a*b^7*c^8*
f + 6*A*C^2*a*b^7*c^8*f - 2*A*B^2*a^7*b*d^8*f + 2*A*B^2*a*b^7*c^8*f - 50*C^
3*a^6*b^2*c^3*d^5*f + 50*C^3*a^5*b^3*c^2*d^6*f - 50*C^3*a^3*b^5*c^6*d^2*f +
 50*C^3*a^2*b^6*c^5*d^3*f + 42*C^3*a^5*b^3*c^4*d^4*f + 42*C^3*a^4*b^4*c^5*d
^3*f - 42*C^3*a^4*b^4*c^3*d^5*f - 42*C^3*a^3*b^5*c^4*d^4*f - 32*C^3*a^6*b^2
*c^5*d^3*f - 32*C^3*a^5*b^3*c^6*d^2*f + 32*C^3*a^3*b^5*c^2*d^6*f + 32*C^3*a
^2*b^6*c^3*d^5*f + 94*B^3*a^4*b^4*c^4*d^4*f + 48*B^3*a^2*b^6*c^2*d^6*f - 44
*B^3*a^3*b^5*c^3*d^5*f - 32*B^3*a^6*b^2*c^2*d^6*f - 32*B^3*a^2*b^6*c^6*d^2*
f + 29*B^3*a^4*b^4*c^2*d^6*f + 29*B^3*a^2*b^6*c^4*d^4*f - 20*B^3*a^5*b^3*c^
5*d^3*f + 18*B^3*a^5*b^3*c^3*d^5*f + 18*B^3*a^3*b^5*c^5*d^3*f - 3*B^3*a^6*b
^2*c^4*d^4*f - 3*B^3*a^4*b^4*c^6*d^2*f + 106*A^3*a^4*b^4*c^3*d^5*f + 106*A^
3*a^3*b^5*c^4*d^4*f - 96*A^3*a^3*b^5*c^2*d^6*f - 96*A^3*a^2*b^6*c^3*d^5*f -
 82*A^3*a^5*b^3*c^2*d^6*f - 82*A^3*a^2*b^6*c^5*d^3*f + 18*A^3*a^6*b^2*c^3*d
^5*f + 18*A^3*a^3*b^5*c^6*d^2*f - 10*A^3*a^5*b^3*c^4*d^4*f - 10*A^3*a^4*b^4
*c^5*d^3*f - 22*C^3*a^7*b*c^2*d^6*f + 22*C^3*a^6*b^2*c*d^7*f - 22*C^3*a^2*b
^6*c^7*d*f + 22*C^3*a*b^7*c^6*d^2*f - 2*A*B*C*b^8*c^8*f - 2*A*B*C*a^8*d^8*f
 + 62*B^3*a^5*b^3*c*d^7*f + 62*B^3*a*b^7*c^5*d^3*f + 16*B^3*a^3*b^5*c*d^7*f
 + 16*B^3*a*b^7*c^3*d^5*f + 6*B^3*a^7*b*c^3*d^5*f + 6*B^3*a^3*b^5*c^7*d*f +
 128*A^3*a^4*b^4*c*d^7*f + 128*A^3*a*b^7*c^4*d^4*f + 32*A^3*a^2*b^6*c*d^7*f
 + 32*A^3*a*b^7*c^2*d^6*f - 10*A^3*a^7*b*c^2*d^6*f + 10*A^3*a^6*b^2*c*d^7*f
 - 10*A^3*a^2*b^6*c^7*d*f + 10*A^3*a*b^7*c^6*d^2*f + 11*B^3*b^8*c^6*d^2*f -
 8*B^3*b^8*c^4*d^4*f - 4*B^3*b^8*c^2*d^6*f - 64*A^3*b^8*c^3*d^5*f - B^3*a^8
*c^2*d^6*f + 11*B^3*a^6*b^2*d^8*f - 8*B^3*a^4*b^4*d^8*f - 4*B^3*a^2*b^6*d^8
*f - 64*A^3*a^3*b^5*d^8*f - B^3*a^2*b^6*c^8*f + 2*C^3*b^8*c^7*d*f - 2*C^3*a
^8*c*d^7*f - 32*A^3*b^8*c*d^7*f + 2*C^3*a^7*b*d^8*f - 2*A^3*b^8*c^7*d*f - 2
*C^3*a*b^7*c^8*f + 2*A^3*a^8*c*d^7*f - 32*A^3*a*b^7*d^8*f - 2*A^3*a^7*b*d^8
*f + 2*A^3*a*b^7*c^8*f - 16*A^2*B*b^8*d^8*f + B*C^2*b^8*c^8*f + B*C^2*a^8*d
^8*f + A^2*B*b^8*c^8*f + A^2*B*a^8*d^8*f + B^3*b^8*c^8*f + B^3*a^8*d^8*f -
4*A*B^2*C*a^5*b*c*d^5 - 4*A*B^2*C*a*b^5*c^5*d + 4*A*B^2*C*a*b^5*c*d^5 + 22*
A^2*B*C*a^3*b^3*c^2*d^4 + 22*A^2*B*C*a^2*b^4*c^3*d^3 - 20*A*B^2*C*a^3*b^3*c
^3*d^3 + 14*A*B^2*C*a^4*b^2*c^2*d^4 + 14*A*B^2*C*a^2*b^4*c^4*d^2 - 14*A*B*C
^2*a^3*b^3*c^2*d^4 - 14*A*B*C^2*a^2*b^4*c^3*d^3 + 12*A*B*C^2*a^4*b^2*c^3*d^
3 + 12*A*B*C^2*a^3*b^3*c^4*d^2 - 6*A^2*B*C*a^4*b^2*c^3*d^3 - 6*A^2*B*C*a^3*
b^3*c^4*d^2 - 4*A*B^2*C*a^2*b^4*c^2*d^4 + 22*A*B*C^2*a^4*b^2*c*d^5 + 22*A*B
*C^2*a*b^5*c^4*d^2 - 20*A^2*B*C*a^4*b^2*c*d^5 - 20*A^2*B*C*a*b^5*c^4*d^2 +
10*A*B*C^2*a^2*b^4*c*d^5 + 10*A*B*C^2*a*b^5*c^2*d^4 - 8*A^2*B*C*a^2*b^4*c*d

^5 - 8*A^2*B*C*a*b^5*c^2*d^4 + 4*A*B^2*C*a^3*b^3*c*d^5 + 4*A*B^2*C*a*b^5*c^
3*d^3 - 4*A*B*C^2*a^5*b*c^2*d^4 - 4*A*B*C^2*a^2*b^4*c^5*d + 2*A^2*B*C*a^5*b
*c^2*d^4 + 2*A^2*B*C*a^2*b^4*c^5*d - 8*B^3*C*a^4*b^2*c*d^5 - 8*B^3*C*a*b^5*
c^4*d^2 - 8*B*C^3*a^4*b^2*c*d^5 - 8*B*C^3*a*b^5*c^4*d^2 - 4*B^3*C*a^2*b^4*c
*d^5 - 4*B^3*C*a*b^5*c^2*d^4 + 4*B^2*C^2*a^5*b*c*d^5 + 4*B^2*C^2*a*b^5*c^5*
d - 4*B*C^3*a^2*b^4*c*d^5 - 4*B*C^3*a*b^5*c^2*d^4 + 2*B^3*C*a^5*b*c^2*d^4 +
2*B^3*C*a^2*b^4*c^5*d + 2*B^2*C^2*a*b^5*c*d^5 + 2*B*C^3*a^5*b*c^2*d^4 + 2*
B*C^3*a^2*b^4*c^5*d + 24*A^3*C*a^3*b^3*c*d^5 + 24*A^3*C*a*b^5*c^3*d^3 - 24*
A^2*C^2*a*b^5*c*d^5 + 12*A^2*C^2*a^5*b*c*d^5 + 12*A^2*C^2*a*b^5*c^5*d + 8*A
*C^3*a^3*b^3*c*d^5 + 8*A*C^3*a*b^5*c^3*d^3 + 6*A^3*B*a^4*b^2*c*d^5 + 6*A^3*
B*a*b^5*c^4*d^2 - 6*A^2*B^2*a*b^5*c*d^5 + 6*A*B^3*a^4*b^2*c*d^5 + 6*A*B^3*a
*b^5*c^4*d^2 + 2*A^3*B*a^2*b^4*c*d^5 + 2*A^3*B*a*b^5*c^2*d^4 + 2*A*B^3*a^2*
b^4*c*d^5 + 2*A*B^3*a*b^5*c^2*d^4 + 20*A^2*B*C*b^6*c^3*d^3 - 10*A*B*C^2*b^6
*c^3*d^3 - 2*A*B^2*C*b^6*c^4*d^2 - 2*A*B^2*C*b^6*c^2*d^4 + 20*A^2*B*C*a^3*b
^3*d^6 - 10*A*B*C^2*a^3*b^3*d^6 - 2*A*B^2*C*a^4*b^2*d^6 - 2*A*B^2*C*a^2*b^4
*d^6 + 10*B^2*C^2*a^3*b^3*c^3*d^3 + 4*B^2*C^2*a^4*b^2*c^4*d^2 - 3*B^2*C^2*a
^4*b^2*c^2*d^4 - 3*B^2*C^2*a^2*b^4*c^4*d^2 + 2*B^2*C^2*a^2*b^4*c^2*d^4 + 40
*A^2*C^2*a^2*b^4*c^2*d^4 - 16*A^2*C^2*a^4*b^2*c^2*d^4 - 16*A^2*C^2*a^2*b^4*
c^4*d^2 + 4*A^2*C^2*a^4*b^2*c^4*d^2 + 18*A^2*B^2*a^2*b^4*c^2*d^4 + 10*A^2*B
^2*a^3*b^3*c^3*d^3 - 3*A^2*B^2*a^4*b^2*c^2*d^4 - 3*A^2*B^2*a^2*b^4*c^4*d^2
+ 24*A^3*C*a*b^5*c*d^5 - 12*A*C^3*a^5*b*c*d^5 - 12*A*C^3*a*b^5*c^5*d + 8*A*
C^3*a*b^5*c*d^5 - 4*A^3*C*a^5*b*c*d^5 - 4*A^3*C*a*b^5*c^5*d + 8*A^2*B*C*b^6
*c*d^5 + 4*A*B*C^2*b^6*c^5*d - 4*A*B*C^2*b^6*c*d^5 - 2*A^2*B*C*b^6*c^5*d +
8*A^2*B*C*a*b^5*d^6 + 4*A*B*C^2*a^5*b*d^6 - 4*A*B*C^2*a*b^5*d^6 - 2*A^2*B*C
*a^5*b*d^6 - 6*B^3*C*a^4*b^2*c^3*d^3 - 6*B^3*C*a^3*b^3*c^4*d^2 - 6*B*C^3*a^
4*b^2*c^3*d^3 - 6*B*C^3*a^3*b^3*c^4*d^2 + 2*B^3*C*a^3*b^3*c^2*d^4 + 2*B^3*C
*a^2*b^4*c^3*d^3 + 2*B^2*C^2*a^3*b^3*c*d^5 + 2*B^2*C^2*a*b^5*c^3*d^3 + 2*B*
C^3*a^3*b^3*c^2*d^4 + 2*B*C^3*a^2*b^4*c^3*d^3 - 48*A^3*C*a^2*b^4*c^2*d^4 -
24*A^2*C^2*a^3*b^3*c*d^5 - 24*A^2*C^2*a*b^5*c^3*d^3 - 16*A*C^3*a^2*b^4*c^2*
d^4 + 8*A^3*C*a^4*b^2*c^2*d^4 + 8*A^3*C*a^2*b^4*c^4*d^2 - 8*A*C^3*a^4*b^2*c
^4*d^2 + 8*A*C^3*a^4*b^2*c^2*d^4 + 8*A*C^3*a^2*b^4*c^4*d^2 - 10*A^3*B*a^3*b
^3*c^2*d^4 - 10*A^3*B*a^2*b^4*c^3*d^3 - 10*A*B^3*a^3*b^3*c^2*d^4 - 10*A*B^3
*a^2*b^4*c^3*d^3 - 6*A^2*B^2*a^3*b^3*c*d^5 - 6*A^2*B^2*a*b^5*c^3*d^3 + 3*B^
2*C^2*b^6*c^4*d^2 - 8*A^2*C^2*b^6*c^4*d^2 + 8*A^2*C^2*b^6*c^2*d^4 + 9*A^2*B
^2*b^6*c^2*d^4 + 3*B^2*C^2*a^4*b^2*d^6 + 3*A^2*B^2*b^6*c^4*d^2 - 8*A^2*C^2*
a^4*b^2*d^6 + 8*A^2*C^2*a^2*b^4*d^6 + 9*A^2*B^2*a^2*b^4*d^6 + 3*A^2*B^2*a^4
*b^2*d^6 + 2*B^4*a^3*b^3*c*d^5 + 2*B^4*a*b^5*c^3*d^3 - 8*A^4*a^3*b^3*c*d^5
- 8*A^4*a*b^5*c^3*d^3 - 16*A^3*C*b^6*c^2*d^4 + 4*A^3*C*b^6*c^4*d^2 + 4*A*C^
3*b^6*c^4*d^2 - 10*A^3*B*b^6*c^3*d^3 - 10*A*B^3*b^6*c^3*d^3 - 16*A^3*C*a^2*
b^4*d^6 + 4*A^3*C*a^4*b^2*d^6 + 4*A*C^3*a^4*b^2*d^6 - 10*A^3*B*a^3*b^3*d^6
- 10*A*B^3*a^3*b^3*d^6 + 4*C^4*a^5*b*c*d^5 + 4*C^4*a*b^5*c^5*d + 2*B^4*a*b^
5*c*d^5 - 8*A^4*a*b^5*c*d^5 - 2*B^3*C*b^6*c^5*d - 2*B*C^3*b^6*c^5*d - 4*A^3
*B*b^6*c*d^5 - 4*A*B^3*b^6*c*d^5 - 2*B^3*C*a^5*b*d^6 - 2*B*C^3*a^5*b*d^6 -
4*A^3*B*a*b^5*d^6 - 4*A*B^3*a*b^5*d^6 + 4*C^4*a^4*b^2*c^4*d^2 + 4*C^4*a^2*b
^4*c^2*d^4 + 10*B^4*a^3*b^3*c^3*d^3 - 3*B^4*a^4*b^2*c^2*d^4 - 3*B^4*a^2*b^4

*c^4*d^2 - 2*B^4*a^2*b^4*c^2*d^4 + 20*A^4*a^2*b^4*c^2*d^4 + B^2*C^2*b^6*c^2
*d^4 + B^2*C^2*a^2*b^4*d^6 - 8*A^3*C*b^6*d^6 + 3*B^4*b^6*c^4*d^2 + 8*A^4*b^
6*c^2*d^4 + 3*B^4*a^4*b^2*d^6 + 8*A^4*a^2*b^4*d^6 + 4*A^2*C^2*b^6*d^6 + 4*A
^2*B^2*b^6*d^6 + 4*A^4*b^6*d^6 + B^4*b^6*c^2*d^4 + B^4*a^2*b^4*d^6, f, k)*(
root(144*a^13*b*c^5*d^9*f^4 + 144*a^9*b^5*c*d^13*f^4 + 144*a^5*b^9*c^13*d*f
^4 + 144*a*b^13*c^9*d^5*f^4 + 96*a^13*b*c^7*d^7*f^4 + 96*a^13*b*c^3*d^11*f^
4 + 96*a^11*b^3*c*d^13*f^4 + 96*a^7*b^7*c^13*d*f^4 + 96*a^7*b^7*c*d^13*f^4
+ 96*a^3*b^11*c^13*d*f^4 + 96*a*b^13*c^11*d^3*f^4 + 96*a*b^13*c^7*d^7*f^4 +
24*a^13*b*c^9*d^5*f^4 + 24*a^9*b^5*c^13*d*f^4 + 24*a^5*b^9*c*d^13*f^4 + 24
*a*b^13*c^5*d^9*f^4 + 24*a^13*b*c*d^13*f^4 + 24*a*b^13*c^13*d*f^4 + 3648*a^
7*b^7*c^7*d^7*f^4 - 3188*a^8*b^6*c^6*d^8*f^4 - 3188*a^6*b^8*c^8*d^6*f^4 - 2
912*a^8*b^6*c^8*d^6*f^4 - 2912*a^6*b^8*c^6*d^8*f^4 + 2592*a^9*b^5*c^7*d^7*f
^4 + 2592*a^7*b^7*c^9*d^5*f^4 + 2592*a^7*b^7*c^5*d^9*f^4 + 2592*a^5*b^9*c^7
*d^7*f^4 + 2168*a^9*b^5*c^5*d^9*f^4 + 2168*a^5*b^9*c^9*d^5*f^4 - 1776*a^10*
b^4*c^6*d^8*f^4 - 1776*a^8*b^6*c^4*d^10*f^4 - 1776*a^6*b^8*c^10*d^4*f^4 - 1
776*a^4*b^10*c^8*d^6*f^4 + 1568*a^9*b^5*c^9*d^5*f^4 + 1568*a^5*b^9*c^5*d^9*
f^4 - 1344*a^10*b^4*c^8*d^6*f^4 - 1344*a^8*b^6*c^10*d^4*f^4 - 1344*a^6*b^8*
c^4*d^10*f^4 - 1344*a^4*b^10*c^6*d^8*f^4 - 1164*a^10*b^4*c^4*d^10*f^4 - 116
4*a^4*b^10*c^10*d^4*f^4 + 896*a^11*b^3*c^5*d^9*f^4 + 896*a^9*b^5*c^3*d^11*f
^4 + 896*a^5*b^9*c^11*d^3*f^4 + 896*a^3*b^11*c^9*d^5*f^4 + 864*a^11*b^3*c^7
*d^7*f^4 + 864*a^7*b^7*c^11*d^3*f^4 + 864*a^7*b^7*c^3*d^11*f^4 + 864*a^3*b^
11*c^7*d^7*f^4 - 480*a^10*b^4*c^10*d^4*f^4 - 480*a^4*b^10*c^4*d^10*f^4 + 46
4*a^11*b^3*c^3*d^11*f^4 + 464*a^3*b^11*c^11*d^3*f^4 - 424*a^12*b^2*c^6*d^8*
f^4 - 424*a^8*b^6*c^2*d^12*f^4 - 424*a^6*b^8*c^12*d^2*f^4 - 424*a^2*b^12*c^
8*d^6*f^4 + 416*a^11*b^3*c^9*d^5*f^4 + 416*a^9*b^5*c^11*d^3*f^4 + 416*a^5*b
^9*c^3*d^11*f^4 + 416*a^3*b^11*c^5*d^9*f^4 - 336*a^12*b^2*c^4*d^10*f^4 - 33
6*a^10*b^4*c^2*d^12*f^4 - 336*a^4*b^10*c^12*d^2*f^4 - 336*a^2*b^12*c^10*d^4
*f^4 - 256*a^12*b^2*c^8*d^6*f^4 - 256*a^8*b^6*c^12*d^2*f^4 - 256*a^6*b^8*c^
2*d^12*f^4 - 256*a^2*b^12*c^6*d^8*f^4 - 124*a^12*b^2*c^2*d^12*f^4 - 124*a^2
*b^12*c^12*d^2*f^4 + 80*a^11*b^3*c^11*d^3*f^4 + 80*a^3*b^11*c^3*d^11*f^4 -
60*a^12*b^2*c^10*d^4*f^4 - 60*a^10*b^4*c^12*d^2*f^4 - 60*a^4*b^10*c^2*d^12*
f^4 - 60*a^2*b^12*c^4*d^10*f^4 - 24*b^14*c^10*d^4*f^4 - 16*b^14*c^12*d^2*f^
4 - 16*b^14*c^8*d^6*f^4 - 4*b^14*c^6*d^8*f^4 - 24*a^14*c^4*d^10*f^4 - 16*a^
14*c^6*d^8*f^4 - 16*a^14*c^2*d^12*f^4 - 4*a^14*c^8*d^6*f^4 - 24*a^10*b^4*d^
14*f^4 - 16*a^12*b^2*d^14*f^4 - 16*a^8*b^6*d^14*f^4 - 4*a^6*b^8*d^14*f^4 -
24*a^4*b^10*c^14*f^4 - 16*a^6*b^8*c^14*f^4 - 16*a^2*b^12*c^14*f^4 - 4*a^8*b
^6*c^14*f^4 - 4*b^14*c^14*f^4 - 4*a^14*d^14*f^4 + 36*A*C*a^9*b*c*d^9*f^2 +
36*A*C*a*b^9*c^9*d*f^2 + 32*A*C*a*b^9*c*d^9*f^2 - 552*B*C*a^7*b^3*c^4*d^6*f
^2 - 552*B*C*a^4*b^6*c^7*d^3*f^2 - 408*B*C*a^5*b^5*c^4*d^6*f^2 - 408*B*C*a^
4*b^6*c^5*d^5*f^2 + 360*B*C*a^6*b^4*c^3*d^7*f^2 + 360*B*C*a^3*b^7*c^6*d^4*f
^2 - 248*B*C*a^7*b^3*c^2*d^8*f^2 - 248*B*C*a^2*b^8*c^7*d^3*f^2 + 184*B*C*a^
6*b^4*c^5*d^5*f^2 + 184*B*C*a^5*b^5*c^6*d^4*f^2 + 152*B*C*a^8*b^2*c^3*d^7*f
^2 - 152*B*C*a^5*b^5*c^2*d^8*f^2 + 152*B*C*a^3*b^7*c^8*d^2*f^2 - 152*B*C*a^
2*b^8*c^5*d^5*f^2 - 104*B*C*a^7*b^3*c^6*d^4*f^2 - 104*B*C*a^6*b^4*c^7*d^3*f
^2 + 64*B*C*a^8*b^2*c^5*d^5*f^2 + 64*B*C*a^5*b^5*c^8*d^2*f^2 - 56*B*C*a^4*b

^6*c^3*d^7*f^2 - 56*B*C*a^3*b^7*c^4*d^6*f^2 - 24*B*C*a^8*b^2*c^7*d^3*f^2 - 24*B*C*a^7*b^3*c^8*d^2*f^2 - 24*B*C*a^3*b^7*c^2*d^8*f^2 - 24*B*C*a^2*b^8*c^3*d^7*f^2 - 696*A*C*a^5*b^5*c^5*d^5*f^2 + 536*A*C*a^6*b^4*c^6*d^4*f^2 + 536*A*C*a^6*b^4*c^4*d^6*f^2 + 536*A*C*a^4*b^6*c^6*d^4*f^2 + 472*A*C*a^4*b^6*c^4*d^6*f^2 - 232*A*C*a^7*b^3*c^5*d^5*f^2 - 232*A*C*a^5*b^5*c^7*d^3*f^2 + 216*A*C*a^3*b^7*c^3*d^7*f^2 + 168*A*C*a^7*b^3*c^3*d^7*f^2 + 168*A*C*a^3*b^7*c^7*d^3*f^2 - 154*A*C*a^8*b^2*c^2*d^8*f^2 - 154*A*C*a^2*b^8*c^8*d^2*f^2 + 62*A*C*a^8*b^2*c^6*d^4*f^2 + 62*A*C*a^6*b^4*c^8*d^2*f^2 - 40*A*C*a^7*b^3*c^7*d^3*f^2 - 40*A*C*a^5*b^5*c^3*d^7*f^2 - 40*A*C*a^3*b^7*c^5*d^5*f^2 + 32*A*C*a^6*b^4*c^2*d^8*f^2 + 32*A*C*a^2*b^8*c^6*d^4*f^2 - 32*A*C*a^2*b^8*c^2*d^8*f^2 + 30*A*C*a^4*b^6*c^2*d^8*f^2 + 30*A*C*a^2*b^8*c^4*d^6*f^2 + 16*A*C*a^8*b^2*c^4*d^6*f^2 + 16*A*C*a^4*b^6*c^8*d^2*f^2 - 488*A*B*a^6*b^4*c^3*d^7*f^2 - 488*A*B*a^3*b^7*c^6*d^4*f^2 + 440*A*B*a^7*b^3*c^4*d^6*f^2 + 440*A*B*a^4*b^6*c^7*d^3*f^2 - 360*A*B*a^6*b^4*c^5*d^5*f^2 - 360*A*B*a^5*b^5*c^6*d^4*f^2 - 192*A*B*a^8*b^2*c^3*d^7*f^2 - 192*A*B*a^3*b^7*c^8*d^2*f^2 - 168*A*B*a^3*b^7*c^2*d^8*f^2 - 168*A*B*a^2*b^8*c^3*d^7*f^2 - 152*A*B*a^4*b^6*c^3*d^7*f^2 - 152*A*B*a^3*b^7*c^4*d^6*f^2 - 120*A*B*a^8*b^2*c^5*d^5*f^2 + 120*A*B*a^7*b^3*c^2*d^8*f^2 - 120*A*B*a^5*b^5*c^8*d^2*f^2 + 120*A*B*a^5*b^5*c^4*d^6*f^2 - 120*A*B*a^5*b^5*c^2*d^8*f^2 + 120*A*B*a^4*b^6*c^5*d^5*f^2 + 120*A*B*a^2*b^8*c^7*d^3*f^2 - 120*A*B*a^2*b^8*c^5*d^5*f^2 + 40*A*B*a^7*b^3*c^6*d^4*f^2 + 40*A*B*a^6*b^4*c^7*d^3*f^2 - 72*B*C*a^9*b*c^4*d^6*f^2 - 72*B*C*a^4*b^6*c^9*d*f^2 - 64*B*C*a^4*b^6*c*d^9*f^2 - 64*B*C*a*b^9*c^4*d^6*f^2 - 32*B*C*a^8*b^2*c*d^9*f^2 - 32*B*C*a*b^9*c^8*d^2*f^2 - 16*B*C*a^2*b^8*c*d^9*f^2 - 16*B*C*a*b^9*c^2*d^8*f^2 + 8*B*C*a^9*b*c^6*d^4*f^2 - 8*B*C*a^9*b*c^2*d^8*f^2 + 8*B*C*a^6*b^4*c^9*d*f^2 - 8*B*C*a^2*b^8*c^9*d*f^2 + 104*A*C*a^7*b^3*c*d^9*f^2 + 104*A*C*a*b^9*c^7*d^3*f^2 + 96*A*C*a^3*b^7*c*d^9*f^2 + 96*A*C*a*b^9*c^3*d^7*f^2 + 72*A*C*a^9*b*c^3*d^7*f^2 + 72*A*C*a^3*b^7*c^9*d*f^2 + 68*A*C*a^5*b^5*c*d^9*f^2 + 68*A*C*a*b^9*c^5*d^5*f^2 - 28*A*C*a^9*b*c^5*d^5*f^2 - 28*A*C*a^5*b^5*c^9*d*f^2 + 80*A*B*a^9*b*c^4*d^6*f^2 + 80*A*B*a^4*b^6*c^9*d*f^2 + 24*A*B*a^8*b^2*c*d^9*f^2 - 24*A*B*a^6*b^4*c*d^9*f^2 + 24*A*B*a^4*b^6*c*d^9*f^2 - 24*A*B*a^2*b^8*c*d^9*f^2 + 24*A*B*a*b^9*c^8*d^2*f^2 - 24*A*B*a*b^9*c^6*d^4*f^2 + 24*A*B*a*b^9*c^4*d^6*f^2 - 24*A*B*a*b^9*c^2*d^8*f^2 - 32*B*C*b^10*c^7*d^3*f^2 - 8*B*C*b^10*c^5*d^5*f^2 + 34*A*C*b^10*c^6*d^4*f^2 + 16*B*C*a^10*c^3*d^7*f^2 + 16*A*C*b^10*c^4*d^6*f^2 - 12*A*C*b^10*c^8*d^2*f^2 - 96*A*B*b^10*c^5*d^5*f^2 - 72*A*B*b^10*c^3*d^7*f^2 - 32*B*C*a^7*b^3*d^10*f^2 - 28*A*C*a^10*c^2*d^8*f^2 - 24*A*B*b^10*c^7*d^3*f^2 - 8*B*C*a^5*b^5*d^10*f^2 + 2*A*C*a^10*c^4*d^6*f^2 + 34*A*C*a^6*b^4*d^10*f^2 + 16*B*C*a^3*b^7*c^10*f^2 + 16*A*C*a^4*b^6*d^10*f^2 - 16*A*B*a^10*c^3*d^7*f^2 - 12*A*C*a^8*b^2*d^10*f^2 - 96*A*B*a^5*b^5*d^10*f^2 - 72*A*B*a^3*b^7*d^10*f^2 - 28*A*C*a^2*b^8*c^10*f^2 - 24*A*B*a^7*b^3*d^10*f^2 + 2*A*C*a^4*b^6*c^10*f^2 - 16*A*B*a^3*b^7*c^10*f^2 + 444*C^2*a^5*b^5*c^5*d^5*f^2 + 148*C^2*a^7*b^3*c^5*d^5*f^2 + 148*C^2*a^5*b^5*c^7*d^3*f^2 + 148*C^2*a^5*b^5*c^3*d^7*f^2 + 148*C^2*a^3*b^7*c^5*d^5*f^2 - 140*C^2*a^6*b^4*c^6*d^4*f^2 - 140*C^2*a^6*b^4*c^4*d^6*f^2 - 140*C^2*a^4*b^6*c^6*d^4*f^2 - 140*C^2*a^4*b^6*c^4*d^6*f^2 + 109*C^2*a^8*b^2*c^2*d^8*f^2 + 109*C^2*a^2*b^8*c^8*d^2*f^2 + 48*C^2*a^8*b^2*c^4*d^6*f^2 + 48*C^2

*a^6*b^4*c^2*d^8*f^2 + 48*C^2*a^4*b^6*c^8*d^2*f^2 + 48*C^2*a^2*b^8*c^6*d^4*
f^2 + 20*C^2*a^7*b^3*c^7*d^3*f^2 - 20*C^2*a^7*b^3*c^3*d^7*f^2 - 20*C^2*a^3*
b^7*c^7*d^3*f^2 + 20*C^2*a^3*b^7*c^3*d^7*f^2 + 17*C^2*a^8*b^2*c^6*d^4*f^2 +
17*C^2*a^6*b^4*c^8*d^2*f^2 + 17*C^2*a^4*b^6*c^2*d^8*f^2 + 17*C^2*a^2*b^8*c
^4*d^6*f^2 + 16*C^2*a^8*b^2*c^8*d^2*f^2 + 16*C^2*a^2*b^8*c^2*d^8*f^2 - 396*
B^2*a^5*b^5*c^5*d^5*f^2 + 308*B^2*a^6*b^4*c^4*d^6*f^2 + 308*B^2*a^4*b^6*c^6
*d^4*f^2 + 300*B^2*a^4*b^6*c^4*d^6*f^2 + 284*B^2*a^6*b^4*c^6*d^4*f^2 - 132*
B^2*a^7*b^3*c^5*d^5*f^2 - 132*B^2*a^5*b^5*c^7*d^3*f^2 - 84*B^2*a^5*b^5*c^3*
d^7*f^2 - 84*B^2*a^3*b^7*c^5*d^5*f^2 + 61*B^2*a^4*b^6*c^2*d^8*f^2 + 61*B^2*
a^2*b^8*c^4*d^6*f^2 - 59*B^2*a^8*b^2*c^2*d^8*f^2 - 59*B^2*a^2*b^8*c^8*d^2*f
^2 + 56*B^2*a^6*b^4*c^2*d^8*f^2 + 56*B^2*a^2*b^8*c^6*d^4*f^2 + 52*B^2*a^7*b
^3*c^3*d^7*f^2 + 52*B^2*a^3*b^7*c^7*d^3*f^2 + 44*B^2*a^3*b^7*c^3*d^7*f^2 +
33*B^2*a^8*b^2*c^6*d^4*f^2 + 33*B^2*a^6*b^4*c^8*d^2*f^2 + 20*B^2*a^8*b^2*c^
4*d^6*f^2 - 20*B^2*a^7*b^3*c^7*d^3*f^2 + 20*B^2*a^4*b^6*c^8*d^2*f^2 + 8*B^2
*a^2*b^8*c^2*d^8*f^2 + 337*A^2*a^4*b^6*c^2*d^8*f^2 + 337*A^2*a^2*b^8*c^4*d^
6*f^2 + 272*A^2*a^2*b^8*c^2*d^8*f^2 + 252*A^2*a^5*b^5*c^5*d^5*f^2 + 244*A^2
*a^4*b^6*c^4*d^6*f^2 - 236*A^2*a^3*b^7*c^3*d^7*f^2 + 176*A^2*a^6*b^4*c^2*d^
8*f^2 + 176*A^2*a^2*b^8*c^6*d^4*f^2 - 148*A^2*a^7*b^3*c^3*d^7*f^2 - 148*A^2
*a^3*b^7*c^7*d^3*f^2 - 140*A^2*a^6*b^4*c^6*d^4*f^2 + 109*A^2*a^8*b^2*c^2*d^
8*f^2 + 109*A^2*a^2*b^8*c^8*d^2*f^2 - 108*A^2*a^5*b^5*c^3*d^7*f^2 - 108*A^2
*a^3*b^7*c^5*d^5*f^2 + 84*A^2*a^7*b^3*c^5*d^5*f^2 + 84*A^2*a^5*b^5*c^7*d^3*
f^2 + 32*A^2*a^8*b^2*c^4*d^6*f^2 + 32*A^2*a^4*b^6*c^8*d^2*f^2 + 20*A^2*a^7*
b^3*c^7*d^3*f^2 - 15*A^2*a^8*b^2*c^6*d^4*f^2 - 15*A^2*a^6*b^4*c^8*d^2*f^2 -
12*A^2*a^6*b^4*c^4*d^6*f^2 - 12*A^2*a^4*b^6*c^6*d^4*f^2 + 8*B*C*b^10*c^9*d
*f^2 - 16*B*C*a^10*c*d^9*f^2 - 16*A*B*b^10*c^9*d*f^2 - 16*A*B*b^10*c*d^9*f^
2 + 8*B*C*a^9*b*d^10*f^2 - 16*B*C*a*b^9*c^10*f^2 + 16*A*B*a^10*c*d^9*f^2 -
16*A*B*a^9*b*d^10*f^2 - 16*A*B*a*b^9*d^10*f^2 + 16*A*B*a*b^9*c^10*f^2 + 22*
C^2*a^9*b*c^5*d^5*f^2 + 22*C^2*a^5*b^5*c^9*d*f^2 + 22*C^2*a^5*b^5*c*d^9*f^2
+ 22*C^2*a*b^9*c^5*d^5*f^2 - 20*C^2*a^9*b*c^3*d^7*f^2 - 20*C^2*a^7*b^3*c*d
^9*f^2 - 20*C^2*a^3*b^7*c^9*d*f^2 - 20*C^2*a*b^9*c^7*d^3*f^2 + 36*B^2*a^7*b
^3*c*d^9*f^2 + 36*B^2*a*b^9*c^7*d^3*f^2 + 28*B^2*a^9*b*c^3*d^7*f^2 + 28*B^2
*a^3*b^7*c^9*d*f^2 + 24*B^2*a^3*b^7*c*d^9*f^2 + 24*B^2*a*b^9*c^3*d^7*f^2 -
18*B^2*a^9*b*c^5*d^5*f^2 - 18*B^2*a^5*b^5*c^9*d*f^2 + 6*B^2*a^5*b^5*c*d^9*f
^2 + 6*B^2*a*b^9*c^5*d^5*f^2 - 96*A^2*a^3*b^7*c*d^9*f^2 - 96*A^2*a*b^9*c^3*
d^7*f^2 - 90*A^2*a^5*b^5*c*d^9*f^2 - 90*A^2*a*b^9*c^5*d^5*f^2 - 84*A^2*a^7*
b^3*c*d^9*f^2 - 84*A^2*a*b^9*c^7*d^3*f^2 - 52*A^2*a^9*b*c^3*d^7*f^2 - 52*A^
2*a^3*b^7*c^9*d*f^2 + 6*A^2*a^9*b*c^5*d^5*f^2 + 6*A^2*a^5*b^5*c^9*d*f^2 - 1
0*C^2*a^9*b*c*d^9*f^2 - 10*C^2*a*b^9*c^9*d*f^2 + 14*B^2*a^9*b*c*d^9*f^2 + 1
4*B^2*a*b^9*c^9*d*f^2 + 8*B^2*a*b^9*c*d^9*f^2 - 32*A^2*a*b^9*c*d^9*f^2 - 26
*A^2*a^9*b*c*d^9*f^2 - 26*A^2*a*b^9*c^9*d*f^2 + 2*A*C*b^10*c^10*f^2 + 2*A*C
*a^10*d^10*f^2 + 14*C^2*b^10*c^8*d^2*f^2 - C^2*b^10*c^6*d^4*f^2 + 31*B^2*b^
10*c^6*d^4*f^2 + 20*B^2*b^10*c^4*d^6*f^2 + 14*C^2*a^10*c^2*d^8*f^2 + 4*B^2*
b^10*c^2*d^8*f^2 + 2*B^2*b^10*c^8*d^2*f^2 - C^2*a^10*c^4*d^6*f^2 + 80*A^2*b
^10*c^4*d^6*f^2 + 64*A^2*b^10*c^2*d^8*f^2 + 31*A^2*b^10*c^6*d^4*f^2 + 14*C^
2*a^8*b^2*d^10*f^2 + 14*A^2*b^10*c^8*d^2*f^2 - 10*B^2*a^10*c^2*d^8*f^2 + 3*

B^2*a^10*c^4*d^6*f^2 - C^2*a^6*b^4*d^10*f^2 + 31*B^2*a^6*b^4*d^10*f^2 + 20*
B^2*a^4*b^6*d^10*f^2 + 14*C^2*a^2*b^8*c^10*f^2 + 14*A^2*a^10*c^2*d^8*f^2 +
4*B^2*a^2*b^8*d^10*f^2 + 2*B^2*a^8*b^2*d^10*f^2 - C^2*a^4*b^6*c^10*f^2 - A^
2*a^10*c^4*d^6*f^2 + 80*A^2*a^4*b^6*d^10*f^2 + 64*A^2*a^2*b^8*d^10*f^2 + 31
*A^2*a^6*b^4*d^10*f^2 + 14*A^2*a^8*b^2*d^10*f^2 - 10*B^2*a^2*b^8*c^10*f^2 +
 3*B^2*a^4*b^6*c^10*f^2 + 14*A^2*a^2*b^8*c^10*f^2 - A^2*a^4*b^6*c^10*f^2 -
C^2*b^10*c^10*f^2 - C^2*a^10*d^10*f^2 + 16*A^2*b^10*d^10*f^2 + 3*B^2*b^10*c
^10*f^2 + 3*B^2*a^10*d^10*f^2 - A^2*b^10*c^10*f^2 - A^2*a^10*d^10*f^2 - 96*
A*B*C*a*b^7*c*d^7*f - 28*A*B*C*a^7*b*c*d^7*f - 28*A*B*C*a*b^7*c^7*d*f + 484
*A*B*C*a^4*b^4*c^4*d^4*f - 424*A*B*C*a^3*b^5*c^3*d^5*f + 320*A*B*C*a^2*b^6*
c^2*d^6*f - 176*A*B*C*a^6*b^2*c^2*d^6*f - 176*A*B*C*a^2*b^6*c^6*d^2*f + 158
*A*B*C*a^4*b^4*c^2*d^6*f + 158*A*B*C*a^2*b^6*c^4*d^4*f - 136*A*B*C*a^5*b^3*
c^5*d^3*f - 34*A*B*C*a^6*b^2*c^4*d^4*f - 34*A*B*C*a^4*b^4*c^6*d^2*f + 28*A*
B*C*a^5*b^3*c^3*d^5*f + 28*A*B*C*a^3*b^5*c^5*d^3*f + 308*A*B*C*a^5*b^3*c*d^
7*f + 308*A*B*C*a*b^7*c^5*d^3*f + 20*A*B*C*a^7*b*c^3*d^5*f + 20*A*B*C*a^3*b
^5*c^7*d*f + 30*B*C^2*a^7*b*c*d^7*f + 30*B*C^2*a*b^7*c^7*d*f + 160*A^2*B*a*
b^7*c*d^7*f - 2*A^2*B*a^7*b*c*d^7*f - 2*A^2*B*a*b^7*c^7*d*f - 96*A*B*C*b^8*
c^4*d^4*f + 34*A*B*C*b^8*c^6*d^2*f - 32*A*B*C*b^8*c^2*d^6*f + 2*A*B*C*a^8*c
^2*d^6*f - 96*A*B*C*a^4*b^4*d^8*f + 34*A*B*C*a^6*b^2*d^8*f - 32*A*B*C*a^2*b
^6*d^8*f + 2*A*B*C*a^2*b^6*c^8*f - 210*B*C^2*a^4*b^4*c^4*d^4*f - 182*B^2*C*
a^5*b^3*c^2*d^6*f - 182*B^2*C*a^2*b^6*c^5*d^3*f + 180*B*C^2*a^5*b^3*c^5*d^3
*f + 180*B*C^2*a^3*b^5*c^3*d^5*f - 166*B^2*C*a^5*b^3*c^4*d^4*f - 166*B^2*C*
a^4*b^4*c^5*d^3*f + 152*B*C^2*a^6*b^2*c^2*d^6*f + 152*B*C^2*a^2*b^6*c^6*d^2
*f - 112*B^2*C*a^3*b^5*c^2*d^6*f - 112*B^2*C*a^2*b^6*c^3*d^5*f + 94*B^2*C*a
^4*b^4*c^3*d^5*f + 94*B^2*C*a^3*b^5*c^4*d^4*f - 80*B*C^2*a^2*b^6*c^2*d^6*f
+ 66*B*C^2*a^5*b^3*c^3*d^5*f + 66*B*C^2*a^3*b^5*c^5*d^3*f + 46*B^2*C*a^6*b^
2*c^3*d^5*f + 46*B^2*C*a^3*b^5*c^6*d^2*f + 33*B*C^2*a^6*b^2*c^4*d^4*f + 33*
B*C^2*a^4*b^4*c^6*d^2*f + 24*B^2*C*a^6*b^2*c^5*d^3*f + 24*B^2*C*a^5*b^3*c^6
*d^2*f - 16*B*C^2*a^6*b^2*c^6*d^2*f - 15*B*C^2*a^4*b^4*c^2*d^6*f - 15*B*C^2
*a^2*b^6*c^4*d^4*f - 190*A^2*C*a^4*b^4*c^3*d^5*f - 190*A^2*C*a^3*b^5*c^4*d^
4*f + 182*A^2*C*a^5*b^3*c^2*d^6*f + 182*A^2*C*a^2*b^6*c^5*d^3*f + 160*A^2*C
*a^3*b^5*c^2*d^6*f + 160*A^2*C*a^2*b^6*c^3*d^5*f - 150*A*C^2*a^5*b^3*c^2*d^
6*f - 150*A*C^2*a^2*b^6*c^5*d^3*f - 126*A*C^2*a^5*b^3*c^4*d^4*f - 126*A*C^2
*a^4*b^4*c^5*d^3*f + 126*A*C^2*a^4*b^4*c^3*d^5*f + 126*A*C^2*a^3*b^5*c^4*d^
4*f - 96*A*C^2*a^3*b^5*c^2*d^6*f - 96*A*C^2*a^2*b^6*c^3*d^5*f + 94*A^2*C*a^
5*b^3*c^4*d^4*f + 94*A^2*C*a^4*b^4*c^5*d^3*f + 54*A*C^2*a^6*b^2*c^3*d^5*f +
 54*A*C^2*a^3*b^5*c^6*d^2*f + 32*A*C^2*a^6*b^2*c^5*d^3*f + 32*A*C^2*a^5*b^3
*c^6*d^2*f - 22*A^2*C*a^6*b^2*c^3*d^5*f - 22*A^2*C*a^3*b^5*c^6*d^2*f + 500*
A^2*B*a^3*b^5*c^3*d^5*f - 290*A^2*B*a^4*b^4*c^4*d^4*f - 256*A^2*B*a^2*b^6*c
^2*d^6*f - 230*A*B^2*a^4*b^4*c^3*d^5*f - 230*A*B^2*a^3*b^5*c^4*d^4*f + 142*
A*B^2*a^5*b^3*c^2*d^6*f + 142*A*B^2*a^2*b^6*c^5*d^3*f - 127*A^2*B*a^4*b^4*c
^2*d^6*f - 127*A^2*B*a^2*b^6*c^4*d^4*f + 86*A*B^2*a^5*b^3*c^4*d^4*f + 86*A*
B^2*a^4*b^4*c^5*d^3*f + 80*A*B^2*a^3*b^5*c^2*d^6*f + 80*A*B^2*a^2*b^6*c^3*d
^5*f + 40*A^2*B*a^6*b^2*c^2*d^6*f + 40*A^2*B*a^2*b^6*c^6*d^2*f + 34*A^2*B*a
^5*b^3*c^3*d^5*f + 34*A^2*B*a^3*b^5*c^5*d^3*f - 30*A*B^2*a^6*b^2*c^3*d^5*f

- 30*A*B^2*a^3*b^5*c^6*d^2*f + 20*A^2*B*a^5*b^3*c^5*d^3*f - 15*A^2*B*a^6*b^
2*c^4*d^4*f - 15*A^2*B*a^4*b^4*c^6*d^2*f - 98*B^2*C*a^6*b^2*c*d^7*f - 98*B^
2*C*a*b^7*c^6*d^2*f - 90*B*C^2*a^5*b^3*c*d^7*f - 90*B*C^2*a*b^7*c^5*d^3*f +
 48*B^2*C*a^4*b^4*c*d^7*f + 48*B^2*C*a*b^7*c^4*d^4*f + 40*B^2*C*a^2*b^6*c*d
^7*f + 40*B^2*C*a*b^7*c^2*d^6*f - 32*B*C^2*a^3*b^5*c*d^7*f - 32*B*C^2*a*b^7
*c^3*d^5*f + 26*B^2*C*a^7*b*c^2*d^6*f + 26*B^2*C*a^2*b^6*c^7*d*f - 26*B*C^2
*a^7*b*c^3*d^5*f - 26*B*C^2*a^3*b^5*c^7*d*f - 8*B^2*C*a^7*b*c^4*d^4*f - 8*B
^2*C*a^4*b^4*c^7*d*f - 224*A^2*C*a^4*b^4*c*d^7*f - 224*A^2*C*a*b^7*c^4*d^4*
f - 96*A^2*C*a^2*b^6*c*d^7*f - 96*A^2*C*a*b^7*c^2*d^6*f + 96*A*C^2*a^4*b^4*
c*d^7*f + 96*A*C^2*a*b^7*c^4*d^4*f - 66*A*C^2*a^6*b^2*c*d^7*f - 66*A*C^2*a*
b^7*c^6*d^2*f + 64*A*C^2*a^2*b^6*c*d^7*f + 64*A*C^2*a*b^7*c^2*d^6*f + 34*A^
2*C*a^6*b^2*c*d^7*f + 34*A^2*C*a*b^7*c^6*d^2*f + 34*A*C^2*a^7*b*c^2*d^6*f +
 34*A*C^2*a^2*b^6*c^7*d*f - 2*A^2*C*a^7*b*c^2*d^6*f - 2*A^2*C*a^2*b^6*c^7*d
*f - 208*A*B^2*a^4*b^4*c*d^7*f - 208*A*B^2*a*b^7*c^4*d^4*f + 160*A^2*B*a^3*
b^5*c*d^7*f + 160*A^2*B*a*b^7*c^3*d^5*f - 154*A^2*B*a^5*b^3*c*d^7*f - 154*A
^2*B*a*b^7*c^5*d^3*f - 112*A*B^2*a^2*b^6*c*d^7*f - 112*A*B^2*a*b^7*c^2*d^6*
f + 58*A*B^2*a^6*b^2*c*d^7*f + 58*A*B^2*a*b^7*c^6*d^2*f - 10*A*B^2*a^7*b*c^
2*d^6*f - 10*A*B^2*a^2*b^6*c^7*d*f + 6*A^2*B*a^7*b*c^3*d^5*f + 6*A^2*B*a^3*
b^5*c^7*d*f + 32*B^2*C*b^8*c^5*d^3*f - 17*B*C^2*b^8*c^6*d^2*f + 8*B^2*C*b^8
*c^3*d^5*f + 64*A^2*C*b^8*c^3*d^5*f - 32*A^2*C*b^8*c^5*d^3*f + 32*A*C^2*b^8
*c^5*d^3*f - B*C^2*a^8*c^2*d^6*f + 112*A^2*B*b^8*c^4*d^4*f - 64*A*B^2*b^8*c
^5*d^3*f + 32*B^2*C*a^5*b^3*d^8*f - 17*B*C^2*a^6*b^2*d^8*f + 16*A^2*B*b^8*c
^2*d^6*f + 16*A*B^2*b^8*c^3*d^5*f + 8*B^2*C*a^3*b^5*d^8*f - A^2*B*b^8*c^6*d
^2*f + 64*A^2*C*a^3*b^5*d^8*f - 32*A^2*C*a^5*b^3*d^8*f + 32*A*C^2*a^5*b^3*d
^8*f - A^2*B*a^8*c^2*d^6*f - B*C^2*a^2*b^6*c^8*f + 112*A^2*B*a^4*b^4*d^8*f
- 64*A*B^2*a^5*b^3*d^8*f + 16*A^2*B*a^2*b^6*d^8*f + 16*A*B^2*a^3*b^5*d^8*f
- A^2*B*a^6*b^2*d^8*f - A^2*B*a^2*b^6*c^8*f - 8*B^3*a*b^7*c*d^7*f - 2*B^3*a
^7*b*c*d^7*f - 2*B^3*a*b^7*c^7*d*f - 6*B^2*C*b^8*c^7*d*f + 32*A^2*C*b^8*c*d
^7*f + 6*A^2*C*b^8*c^7*d*f - 6*A*C^2*b^8*c^7*d*f - 2*B^2*C*a^8*c*d^7*f + 16
*A*B^2*b^8*c*d^7*f - 6*B^2*C*a^7*b*d^8*f - 6*A^2*C*a^8*c*d^7*f + 6*A*C^2*a^
8*c*d^7*f - 2*A*B^2*b^8*c^7*d*f + 32*A^2*C*a*b^7*d^8*f + 6*A^2*C*a^7*b*d^8*
f - 6*A*C^2*a^7*b*d^8*f - 2*B^2*C*a*b^7*c^8*f + 2*A*B^2*a^8*c*d^7*f + 16*A*
B^2*a*b^7*d^8*f - 6*A^2*C*a*b^7*c^8*f + 6*A*C^2*a*b^7*c^8*f - 2*A*B^2*a^7*b
*d^8*f + 2*A*B^2*a*b^7*c^8*f - 50*C^3*a^6*b^2*c^3*d^5*f + 50*C^3*a^5*b^3*c^
2*d^6*f - 50*C^3*a^3*b^5*c^6*d^2*f + 50*C^3*a^2*b^6*c^5*d^3*f + 42*C^3*a^5*
b^3*c^4*d^4*f + 42*C^3*a^4*b^4*c^5*d^3*f - 42*C^3*a^4*b^4*c^3*d^5*f - 42*C^
3*a^3*b^5*c^4*d^4*f - 32*C^3*a^6*b^2*c^5*d^3*f - 32*C^3*a^5*b^3*c^6*d^2*f +
 32*C^3*a^3*b^5*c^2*d^6*f + 32*C^3*a^2*b^6*c^3*d^5*f + 94*B^3*a^4*b^4*c^4*d
^4*f + 48*B^3*a^2*b^6*c^2*d^6*f - 44*B^3*a^3*b^5*c^3*d^5*f - 32*B^3*a^6*b^2
*c^2*d^6*f - 32*B^3*a^2*b^6*c^6*d^2*f + 29*B^3*a^4*b^4*c^2*d^6*f + 29*B^3*a
^2*b^6*c^4*d^4*f - 20*B^3*a^5*b^3*c^5*d^3*f + 18*B^3*a^5*b^3*c^3*d^5*f + 18
*B^3*a^3*b^5*c^5*d^3*f - 3*B^3*a^6*b^2*c^4*d^4*f - 3*B^3*a^4*b^4*c^6*d^2*f
+ 106*A^3*a^4*b^4*c^3*d^5*f + 106*A^3*a^3*b^5*c^4*d^4*f - 96*A^3*a^3*b^5*c^
2*d^6*f - 96*A^3*a^2*b^6*c^3*d^5*f - 82*A^3*a^5*b^3*c^2*d^6*f - 82*A^3*a^2*
b^6*c^5*d^3*f + 18*A^3*a^6*b^2*c^3*d^5*f + 18*A^3*a^3*b^5*c^6*d^2*f - 10*A^

3*a^5*b^3*c^4*d^4*f - 10*A^3*a^4*b^4*c^5*d^3*f - 22*C^3*a^7*b*c^2*d^6*f + 2
2*C^3*a^6*b^2*c*d^7*f - 22*C^3*a^2*b^6*c^7*d*f + 22*C^3*a*b^7*c^6*d^2*f - 2
*A*B*C*b^8*c^8*f - 2*A*B*C*a^8*d^8*f + 62*B^3*a^5*b^3*c*d^7*f + 62*B^3*a*b^
7*c^5*d^3*f + 16*B^3*a^3*b^5*c*d^7*f + 16*B^3*a*b^7*c^3*d^5*f + 6*B^3*a^7*b
*c^3*d^5*f + 6*B^3*a^3*b^5*c^7*d*f + 128*A^3*a^4*b^4*c*d^7*f + 128*A^3*a*b^
7*c^4*d^4*f + 32*A^3*a^2*b^6*c*d^7*f + 32*A^3*a*b^7*c^2*d^6*f - 10*A^3*a^7*
b*c^2*d^6*f + 10*A^3*a^6*b^2*c*d^7*f - 10*A^3*a^2*b^6*c^7*d*f + 10*A^3*a*b^
7*c^6*d^2*f + 11*B^3*b^8*c^6*d^2*f - 8*B^3*b^8*c^4*d^4*f - 4*B^3*b^8*c^2*d^
6*f - 64*A^3*b^8*c^3*d^5*f - B^3*a^8*c^2*d^6*f + 11*B^3*a^6*b^2*d^8*f - 8*B
^3*a^4*b^4*d^8*f - 4*B^3*a^2*b^6*d^8*f - 64*A^3*a^3*b^5*d^8*f - B^3*a^2*b^6
*c^8*f + 2*C^3*b^8*c^7*d*f - 2*C^3*a^8*c*d^7*f - 32*A^3*b^8*c*d^7*f + 2*C^3
*a^7*b*d^8*f - 2*A^3*b^8*c^7*d*f - 2*C^3*a*b^7*c^8*f + 2*A^3*a^8*c*d^7*f -
32*A^3*a*b^7*d^8*f - 2*A^3*a^7*b*d^8*f + 2*A^3*a*b^7*c^8*f - 16*A^2*B*b^8*d
^8*f + B*C^2*b^8*c^8*f + B*C^2*a^8*d^8*f + A^2*B*b^8*c^8*f + A^2*B*a^8*d^8*
f + B^3*b^8*c^8*f + B^3*a^8*d^8*f - 4*A*B^2*C*a^5*b*c*d^5 - 4*A*B^2*C*a*b^5
*c^5*d + 4*A*B^2*C*a*b^5*c*d^5 + 22*A^2*B*C*a^3*b^3*c^2*d^4 + 22*A^2*B*C*a^
2*b^4*c^3*d^3 - 20*A*B^2*C*a^3*b^3*c^3*d^3 + 14*A*B^2*C*a^4*b^2*c^2*d^4 + 1
4*A*B^2*C*a^2*b^4*c^4*d^2 - 14*A*B*C^2*a^3*b^3*c^2*d^4 - 14*A*B*C^2*a^2*b^4
*c^3*d^3 + 12*A*B*C^2*a^4*b^2*c^3*d^3 + 12*A*B*C^2*a^3*b^3*c^4*d^2 - 6*A^2*
B*C*a^4*b^2*c^3*d^3 - 6*A^2*B*C*a^3*b^3*c^4*d^2 - 4*A*B^2*C*a^2*b^4*c^2*d^4
 + 22*A*B*C^2*a^4*b^2*c*d^5 + 22*A*B*C^2*a*b^5*c^4*d^2 - 20*A^2*B*C*a^4*b^2
*c*d^5 - 20*A^2*B*C*a*b^5*c^4*d^2 + 10*A*B*C^2*a^2*b^4*c*d^5 + 10*A*B*C^2*a
*b^5*c^2*d^4 - 8*A^2*B*C*a^2*b^4*c*d^5 - 8*A^2*B*C*a*b^5*c^2*d^4 + 4*A*B^2*
C*a^3*b^3*c*d^5 + 4*A*B^2*C*a*b^5*c^3*d^3 - 4*A*B*C^2*a^5*b*c^2*d^4 - 4*A*B
*C^2*a^2*b^4*c^5*d + 2*A^2*B*C*a^5*b*c^2*d^4 + 2*A^2*B*C*a^2*b^4*c^5*d - 8*
B^3*C*a^4*b^2*c*d^5 - 8*B^3*C*a*b^5*c^4*d^2 - 8*B*C^3*a^4*b^2*c*d^5 - 8*B*C
^3*a*b^5*c^4*d^2 - 4*B^3*C*a^2*b^4*c*d^5 - 4*B^3*C*a*b^5*c^2*d^4 + 4*B^2*C^
2*a^5*b*c*d^5 + 4*B^2*C^2*a*b^5*c^5*d - 4*B*C^3*a^2*b^4*c*d^5 - 4*B*C^3*a*b
^5*c^2*d^4 + 2*B^3*C*a^5*b*c^2*d^4 + 2*B^3*C*a^2*b^4*c^5*d + 2*B^2*C^2*a*b^
5*c*d^5 + 2*B*C^3*a^5*b*c^2*d^4 + 2*B*C^3*a^2*b^4*c^5*d + 24*A^3*C*a^3*b^3*
c*d^5 + 24*A^3*C*a*b^5*c^3*d^3 - 24*A^2*C^2*a*b^5*c*d^5 + 12*A^2*C^2*a^5*b*
c*d^5 + 12*A^2*C^2*a*b^5*c^5*d + 8*A*C^3*a^3*b^3*c*d^5 + 8*A*C^3*a*b^5*c^3*
d^3 + 6*A^3*B*a^4*b^2*c*d^5 + 6*A^3*B*a*b^5*c^4*d^2 - 6*A^2*B^2*a*b^5*c*d^5
 + 6*A*B^3*a^4*b^2*c*d^5 + 6*A*B^3*a*b^5*c^4*d^2 + 2*A^3*B*a^2*b^4*c*d^5 +
2*A^3*B*a*b^5*c^2*d^4 + 2*A*B^3*a^2*b^4*c*d^5 + 2*A*B^3*a*b^5*c^2*d^4 + 20*
A^2*B*C*b^6*c^3*d^3 - 10*A*B*C^2*b^6*c^3*d^3 - 2*A*B^2*C*b^6*c^4*d^2 - 2*A*
B^2*C*b^6*c^2*d^4 + 20*A^2*B*C*a^3*b^3*d^6 - 10*A*B*C^2*a^3*b^3*d^6 - 2*A*B
^2*C*a^4*b^2*d^6 - 2*A*B^2*C*a^2*b^4*d^6 + 10*B^2*C^2*a^3*b^3*c^3*d^3 + 4*B
^2*C^2*a^4*b^2*c^4*d^2 - 3*B^2*C^2*a^4*b^2*c^2*d^4 - 3*B^2*C^2*a^2*b^4*c^4*
d^2 + 2*B^2*C^2*a^2*b^4*c^2*d^4 + 40*A^2*C^2*a^2*b^4*c^2*d^4 - 16*A^2*C^2*a
^4*b^2*c^2*d^4 - 16*A^2*C^2*a^2*b^4*c^4*d^2 + 4*A^2*C^2*a^4*b^2*c^4*d^2 + 1
8*A^2*B^2*a^2*b^4*c^2*d^4 + 10*A^2*B^2*a^3*b^3*c^3*d^3 - 3*A^2*B^2*a^4*b^2*
c^2*d^4 - 3*A^2*B^2*a^2*b^4*c^4*d^2 + 24*A^3*C*a*b^5*c*d^5 - 12*A*C^3*a^5*b
*c*d^5 - 12*A*C^3*a*b^5*c^5*d + 8*A*C^3*a*b^5*c*d^5 - 4*A^3*C*a^5*b*c*d^5 -
 4*A^3*C*a*b^5*c^5*d + 8*A^2*B*C*b^6*c*d^5 + 4*A*B*C^2*b^6*c^5*d - 4*A*B*C^

2*b^6*c*d^5 - 2*A^2*B*C*b^6*c^5*d + 8*A^2*B*C*a*b^5*d^6 + 4*A*B*C^2*a^5*b*d
^6 - 4*A*B*C^2*a*b^5*d^6 - 2*A^2*B*C*a^5*b*d^6 - 6*B^3*C*a^4*b^2*c^3*d^3 -
6*B^3*C*a^3*b^3*c^4*d^2 - 6*B*C^3*a^4*b^2*c^3*d^3 - 6*B*C^3*a^3*b^3*c^4*d^2
 + 2*B^3*C*a^3*b^3*c^2*d^4 + 2*B^3*C*a^2*b^4*c^3*d^3 + 2*B^2*C^2*a^3*b^3*c*
d^5 + 2*B^2*C^2*a*b^5*c^3*d^3 + 2*B*C^3*a^3*b^3*c^2*d^4 + 2*B*C^3*a^2*b^4*c
^3*d^3 - 48*A^3*C*a^2*b^4*c^2*d^4 - 24*A^2*C^2*a^3*b^3*c*d^5 - 24*A^2*C^2*a
*b^5*c^3*d^3 - 16*A*C^3*a^2*b^4*c^2*d^4 + 8*A^3*C*a^4*b^2*c^2*d^4 + 8*A^3*C
*a^2*b^4*c^4*d^2 - 8*A*C^3*a^4*b^2*c^4*d^2 + 8*A*C^3*a^4*b^2*c^2*d^4 + 8*A*
C^3*a^2*b^4*c^4*d^2 - 10*A^3*B*a^3*b^3*c^2*d^4 - 10*A^3*B*a^2*b^4*c^3*d^3 -
 10*A*B^3*a^3*b^3*c^2*d^4 - 10*A*B^3*a^2*b^4*c^3*d^3 - 6*A^2*B^2*a^3*b^3*c*
d^5 - 6*A^2*B^2*a*b^5*c^3*d^3 + 3*B^2*C^2*b^6*c^4*d^2 - 8*A^2*C^2*b^6*c^4*d
^2 + 8*A^2*C^2*b^6*c^2*d^4 + 9*A^2*B^2*b^6*c^2*d^4 + 3*B^2*C^2*a^4*b^2*d^6
+ 3*A^2*B^2*b^6*c^4*d^2 - 8*A^2*C^2*a^4*b^2*d^6 + 8*A^2*C^2*a^2*b^4*d^6 + 9
*A^2*B^2*a^2*b^4*d^6 + 3*A^2*B^2*a^4*b^2*d^6 + 2*B^4*a^3*b^3*c*d^5 + 2*B^4*
a*b^5*c^3*d^3 - 8*A^4*a^3*b^3*c*d^5 - 8*A^4*a*b^5*c^3*d^3 - 16*A^3*C*b^6*c^
2*d^4 + 4*A^3*C*b^6*c^4*d^2 + 4*A*C^3*b^6*c^4*d^2 - 10*A^3*B*b^6*c^3*d^3 -
10*A*B^3*b^6*c^3*d^3 - 16*A^3*C*a^2*b^4*d^6 + 4*A^3*C*a^4*b^2*d^6 + 4*A*C^3
*a^4*b^2*d^6 - 10*A^3*B*a^3*b^3*d^6 - 10*A*B^3*a^3*b^3*d^6 + 4*C^4*a^5*b*c*
d^5 + 4*C^4*a*b^5*c^5*d + 2*B^4*a*b^5*c*d^5 - 8*A^4*a*b^5*c*d^5 - 2*B^3*C*b
^6*c^5*d - 2*B*C^3*b^6*c^5*d - 4*A^3*B*b^6*c*d^5 - 4*A*B^3*b^6*c*d^5 - 2*B^
3*C*a^5*b*d^6 - 2*B*C^3*a^5*b*d^6 - 4*A^3*B*a*b^5*d^6 - 4*A*B^3*a*b^5*d^6 +
 4*C^4*a^4*b^2*c^4*d^2 + 4*C^4*a^2*b^4*c^2*d^4 + 10*B^4*a^3*b^3*c^3*d^3 - 3
*B^4*a^4*b^2*c^2*d^4 - 3*B^4*a^2*b^4*c^4*d^2 - 2*B^4*a^2*b^4*c^2*d^4 + 20*A
^4*a^2*b^4*c^2*d^4 + B^2*C^2*b^6*c^2*d^4 + B^2*C^2*a^2*b^4*d^6 - 8*A^3*C*b^
6*d^6 + 3*B^4*b^6*c^4*d^2 + 8*A^4*b^6*c^2*d^4 + 3*B^4*a^4*b^2*d^6 + 8*A^4*a
^2*b^4*d^6 + 4*A^2*C^2*b^6*d^6 + 4*A^2*B^2*b^6*d^6 + 4*A^4*b^6*d^6 + B^4*b^
6*c^2*d^4 + B^4*a^2*b^4*d^6, f, k)*((4*a^5*b^8*d^13 + 4*a^7*b^6*d^13 - 4*a^
9*b^4*d^13 - 4*a^11*b^2*d^13 + 4*b^13*c^5*d^8 + 4*b^13*c^7*d^6 - 4*b^13*c^9
*d^4 - 4*b^13*c^11*d^2 - 12*a*b^12*c^4*d^9 + 4*a*b^12*c^6*d^7 + 52*a*b^12*c
^8*d^5 + 44*a*b^12*c^10*d^3 + 16*a^3*b^10*c^12*d - 12*a^4*b^9*c*d^12 + 8*a^
5*b^8*c^12*d + 4*a^6*b^7*c*d^12 + 52*a^8*b^5*c*d^12 + 44*a^10*b^3*c*d^12 +
16*a^12*b*c^3*d^10 + 8*a^12*b*c^5*d^8 + 8*a^2*b^11*c^3*d^10 - 36*a^2*b^11*c
^5*d^8 - 140*a^2*b^11*c^7*d^6 - 140*a^2*b^11*c^9*d^4 - 44*a^2*b^11*c^11*d^2
 + 8*a^3*b^10*c^2*d^11 + 28*a^3*b^10*c^4*d^9 + 148*a^3*b^10*c^6*d^7 + 260*a
^3*b^10*c^8*d^5 + 148*a^3*b^10*c^10*d^3 + 28*a^4*b^9*c^3*d^10 - 56*a^4*b^9*
c^5*d^8 - 320*a^4*b^9*c^7*d^6 - 300*a^4*b^9*c^9*d^4 - 76*a^4*b^9*c^11*d^2 -
 36*a^5*b^8*c^2*d^11 - 56*a^5*b^8*c^4*d^9 + 160*a^5*b^8*c^6*d^7 + 332*a^5*b
^8*c^8*d^5 + 164*a^5*b^8*c^10*d^3 + 148*a^6*b^7*c^3*d^10 + 160*a^6*b^7*c^5*
d^8 - 144*a^6*b^7*c^7*d^6 - 196*a^6*b^7*c^9*d^4 - 36*a^6*b^7*c^11*d^2 - 140
*a^7*b^6*c^2*d^11 - 320*a^7*b^6*c^4*d^9 - 144*a^7*b^6*c^6*d^7 + 92*a^7*b^6*
c^8*d^5 + 60*a^7*b^6*c^10*d^3 + 260*a^8*b^5*c^3*d^10 + 332*a^8*b^5*c^5*d^8
+ 92*a^8*b^5*c^7*d^6 - 32*a^8*b^5*c^9*d^4 - 140*a^9*b^4*c^2*d^11 - 300*a^9*
b^4*c^4*d^9 - 196*a^9*b^4*c^6*d^7 - 32*a^9*b^4*c^8*d^5 + 148*a^10*b^3*c^3*d
^10 + 164*a^10*b^3*c^5*d^8 + 60*a^10*b^3*c^7*d^6 - 44*a^11*b^2*c^2*d^11 - 7
6*a^11*b^2*c^4*d^9 - 36*a^11*b^2*c^6*d^7 + 8*a*b^12*c^12*d + 8*a^12*b*c*d^1

2)/(a^8*d^8 + b^8*c^8 + 2*a^2*b^6*c^8 + a^4*b^4*c^8 + a^4*b^4*d^8 + 2*a^6*b
^2*d^8 + 2*a^8*c^2*d^6 + a^8*c^4*d^4 + b^8*c^4*d^4 + 2*b^8*c^6*d^2 - 4*a*b^
7*c^3*d^5 - 8*a*b^7*c^5*d^3 - 4*a^3*b^5*c*d^7 - 8*a^3*b^5*c^7*d - 8*a^5*b^3
*c*d^7 - 4*a^5*b^3*c^7*d - 8*a^7*b*c^3*d^5 - 4*a^7*b*c^5*d^3 + 6*a^2*b^6*c^
2*d^6 + 14*a^2*b^6*c^4*d^4 + 10*a^2*b^6*c^6*d^2 - 16*a^3*b^5*c^3*d^5 - 20*a
^3*b^5*c^5*d^3 + 14*a^4*b^4*c^2*d^6 + 26*a^4*b^4*c^4*d^4 + 14*a^4*b^4*c^6*d
^2 - 20*a^5*b^3*c^3*d^5 - 16*a^5*b^3*c^5*d^3 + 10*a^6*b^2*c^2*d^6 + 14*a^6*
b^2*c^4*d^4 + 6*a^6*b^2*c^6*d^2 - 4*a*b^7*c^7*d - 4*a^7*b*c*d^7) + (tan(e +
f*x)*(6*a^12*b*d^13 + 6*b^13*c^12*d + 8*a^4*b^9*d^13 + 22*a^6*b^7*d^13 + 2
6*a^8*b^5*d^13 + 18*a^10*b^3*d^13 + 8*b^13*c^4*d^9 + 22*b^13*c^6*d^7 + 26*b
^13*c^8*d^5 + 18*b^13*c^10*d^3 - 32*a*b^12*c^3*d^10 - 84*a*b^12*c^5*d^8 - 9
2*a*b^12*c^7*d^6 - 60*a*b^12*c^9*d^4 - 20*a*b^12*c^11*d^2 + 10*a^2*b^11*c^1
2*d - 32*a^3*b^10*c*d^12 + 2*a^4*b^9*c^12*d - 84*a^5*b^8*c*d^12 - 2*a^6*b^7
*c^12*d - 92*a^7*b^6*c*d^12 - 60*a^9*b^4*c*d^12 - 20*a^11*b^2*c*d^12 + 10*a
^12*b*c^2*d^11 + 2*a^12*b*c^4*d^9 - 2*a^12*b*c^6*d^7 + 48*a^2*b^11*c^2*d^11
+ 138*a^2*b^11*c^4*d^9 + 152*a^2*b^11*c^6*d^7 + 92*a^2*b^11*c^8*d^5 + 40*a
^2*b^11*c^10*d^3 - 152*a^3*b^10*c^3*d^10 - 196*a^3*b^10*c^5*d^8 - 92*a^3*b^
10*c^7*d^6 - 44*a^3*b^10*c^9*d^4 - 28*a^3*b^10*c^11*d^2 + 138*a^4*b^9*c^2*d
^11 + 220*a^4*b^9*c^4*d^9 + 50*a^4*b^9*c^6*d^7 - 46*a^4*b^9*c^8*d^5 - 4*a^4
*b^9*c^10*d^3 - 196*a^5*b^8*c^3*d^10 - 16*a^5*b^8*c^5*d^8 + 224*a^5*b^8*c^7
*d^6 + 132*a^5*b^8*c^9*d^4 + 4*a^5*b^8*c^11*d^2 + 152*a^6*b^7*c^2*d^11 + 50
*a^6*b^7*c^4*d^9 - 320*a^6*b^7*c^6*d^7 - 294*a^6*b^7*c^8*d^5 - 56*a^6*b^7*c
^10*d^3 - 92*a^7*b^6*c^3*d^10 + 224*a^7*b^6*c^5*d^8 + 368*a^7*b^6*c^7*d^6 +
156*a^7*b^6*c^9*d^4 + 12*a^7*b^6*c^11*d^2 + 92*a^8*b^5*c^2*d^11 - 46*a^8*b
^5*c^4*d^9 - 294*a^8*b^5*c^6*d^7 - 212*a^8*b^5*c^8*d^5 - 30*a^8*b^5*c^10*d^
3 - 44*a^9*b^4*c^3*d^10 + 132*a^9*b^4*c^5*d^8 + 156*a^9*b^4*c^7*d^6 + 40*a^
9*b^4*c^9*d^4 + 40*a^10*b^3*c^2*d^11 - 4*a^10*b^3*c^4*d^9 - 56*a^10*b^3*c^6
*d^7 - 30*a^10*b^3*c^8*d^5 - 28*a^11*b^2*c^3*d^10 + 4*a^11*b^2*c^5*d^8 + 12
*a^11*b^2*c^7*d^6))/(a^8*d^8 + b^8*c^8 + 2*a^2*b^6*c^8 + a^4*b^4*c^8 + a^4*
b^4*d^8 + 2*a^6*b^2*d^8 + 2*a^8*c^2*d^6 + a^8*c^4*d^4 + b^8*c^4*d^4 + 2*b^8
*c^6*d^2 - 4*a*b^7*c^3*d^5 - 8*a*b^7*c^5*d^3 - 4*a^3*b^5*c*d^7 - 8*a^3*b^5*
c^7*d - 8*a^5*b^3*c*d^7 - 4*a^5*b^3*c^7*d - 8*a^7*b*c^3*d^5 - 4*a^7*b*c^5*d
^3 + 6*a^2*b^6*c^2*d^6 + 14*a^2*b^6*c^4*d^4 + 10*a^2*b^6*c^6*d^2 - 16*a^3*b
^5*c^3*d^5 - 20*a^3*b^5*c^5*d^3 + 14*a^4*b^4*c^2*d^6 + 26*a^4*b^4*c^4*d^4 +
14*a^4*b^4*c^6*d^2 - 20*a^5*b^3*c^3*d^5 - 16*a^5*b^3*c^5*d^3 + 10*a^6*b^2*
c^2*d^6 + 14*a^6*b^2*c^4*d^4 + 6*a^6*b^2*c^6*d^2 - 4*a*b^7*c^7*d - 4*a^7*b*
c*d^7)) - (C*a^10*b*d^11 - A*b^11*c^10*d - A*a^10*b*d^11 + C*b^11*c^10*d -
8*A*a^2*b^9*d^11 - 16*A*a^4*b^7*d^11 - A*a^6*b^5*d^11 + 6*A*a^8*b^3*d^11 +
4*B*a^3*b^8*d^11 + 12*B*a^5*b^6*d^11 + 4*B*a^7*b^4*d^11 - 4*B*a^9*b^2*d^11
- 8*A*b^11*c^2*d^9 - 16*A*b^11*c^4*d^7 - A*b^11*c^6*d^5 + 6*A*b^11*c^8*d^3
- 7*C*a^6*b^5*d^11 - 6*C*a^8*b^3*d^11 + 4*B*b^11*c^3*d^8 + 12*B*b^11*c^5*d^
6 + 4*B*b^11*c^7*d^4 - 4*B*b^11*c^9*d^2 - 7*C*b^11*c^6*d^5 - 6*C*b^11*c^8*d
^3 + 56*A*a*b^10*c^3*d^8 + 54*A*a*b^10*c^5*d^6 + 12*A*a*b^10*c^7*d^4 - 2*A*
a*b^10*c^9*d^2 - 2*A*a^2*b^9*c^10*d + 56*A*a^3*b^8*c*d^10 - A*a^4*b^7*c^10*
d + 54*A*a^5*b^6*c*d^10 + 12*A*a^7*b^4*c*d^10 - 2*A*a^9*b^2*c*d^10 - 2*A*a^

10*b*c^2*d^9 - A*a^10*b*c^4*d^7 - 4*B*a*b^10*c^2*d^9 - 32*B*a*b^10*c^4*d^7 - 44*B*a*b^10*c^6*d^5 - 16*B*a*b^10*c^8*d^3 - 4*B*a^2*b^9*c*d^10 - 32*B*a^4*b^7*c*d^10 - 44*B*a^6*b^5*c*d^10 - 16*B*a^8*b^3*c*d^10 - 8*C*a*b^10*c^3*d^8 + 2*C*a*b^10*c^5*d^6 + 20*C*a*b^10*c^7*d^4 + 10*C*a*b^10*c^9*d^2 + 2*C*a^2*b^9*c^10*d - 8*C*a^3*b^8*c*d^10 + C*a^4*b^7*c^10*d + 2*C*a^5*b^6*c*d^10 + 20*C*a^7*b^4*c*d^10 + 10*C*a^9*b^2*c*d^10 + 2*C*a^10*b*c^2*d^9 + C*a^10*b*c^4*d^7 - 80*A*a^2*b^9*c^2*d^9 - 159*A*a^2*b^9*c^4*d^7 - 80*A*a^2*b^9*c^6*d^5 + 5*A*a^2*b^9*c^8*d^3 + 212*A*a^3*b^8*c^3*d^8 + 228*A*a^3*b^8*c^5*d^6 + 76*A*a^3*b^8*c^7*d^4 + 4*A*a^3*b^8*c^9*d^2 - 159*A*a^4*b^7*c^2*d^9 - 332*A*a^4*b^7*c^4*d^7 - 204*A*a^4*b^7*c^6*d^5 - 16*A*a^4*b^7*c^8*d^3 + 228*A*a^5*b^6*c^3*d^8 + 252*A*a^5*b^6*c^5*d^6 + 84*A*a^5*b^6*c^7*d^4 + 6*A*a^5*b^6*c^9*d^2 - 80*A*a^6*b^5*c^2*d^9 - 204*A*a^6*b^5*c^4*d^7 - 140*A*a^6*b^5*c^6*d^5 - 15*A*a^6*b^5*c^8*d^3 + 76*A*a^7*b^4*c^3*d^8 + 84*A*a^7*b^4*c^5*d^6 + 20*A*a^7*b^4*c^7*d^4 + 5*A*a^8*b^3*c^2*d^9 - 16*A*a^8*b^3*c^4*d^7 - 15*A*a^8*b^3*c^6*d^5 + 4*A*a^9*b^2*c^3*d^8 + 6*A*a^9*b^2*c^5*d^6 + 20*B*a^2*b^9*c^3*d^8 + 84*B*a^2*b^9*c^5*d^6 + 60*B*a^2*b^9*c^7*d^4 + 20*B*a^3*b^8*c^2*d^9 - 44*B*a^3*b^8*c^4*d^7 - 100*B*a^3*b^8*c^6*d^5 - 40*B*a^3*b^8*c^8*d^3 - 44*B*a^4*b^7*c^3*d^8 + 60*B*a^4*b^7*c^5*d^6 + 76*B*a^4*b^7*c^7*d^4 + 4*B*a^4*b^7*c^9*d^2 + 84*B*a^5*b^6*c^2*d^9 + 60*B*a^5*b^6*c^4*d^7 - 36*B*a^5*b^6*c^6*d^5 - 24*B*a^5*b^6*c^8*d^3 - 100*B*a^6*b^5*c^3*d^8 - 36*B*a^6*b^5*c^5*d^6 + 20*B*a^6*b^5*c^7*d^4 + 60*B*a^7*b^4*c^2*d^9 + 76*B*a^7*b^4*c^4*d^7 + 20*B*a^7*b^4*c^6*d^5 - 40*B*a^8*b^3*c^3*d^8 - 24*B*a^8*b^3*c^5*d^6 + 4*B*a^9*b^2*c^4*d^7 + 16*C*a^2*b^9*c^2*d^9 + 47*C*a^2*b^9*c^4*d^7 + 16*C*a^2*b^9*c^6*d^5 - 13*C*a^2*b^9*c^8*d^3 - 84*C*a^3*b^8*c^3*d^8 - 100*C*a^3*b^8*c^5*d^6 - 12*C*a^3*b^8*c^7*d^4 + 12*C*a^3*b^8*c^9*d^2 + 47*C*a^4*b^7*c^2*d^9 + 140*C*a^4*b^7*c^4*d^7 + 92*C*a^4*b^7*c^6*d^5 - 100*C*a^5*b^6*c^3*d^8 - 156*C*a^5*b^6*c^5*d^6 - 52*C*a^5*b^6*c^7*d^4 + 2*C*a^5*b^6*c^9*d^2 + 16*C*a^6*b^5*c^2*d^9 + 92*C*a^6*b^5*c^4*d^7 + 76*C*a^6*b^5*c^6*d^5 + 7*C*a^6*b^5*c^8*d^3 - 12*C*a^7*b^4*c^3*d^8 - 52*C*a^7*b^4*c^5*d^6 - 20*C*a^7*b^4*c^7*d^4 - 13*C*a^8*b^3*c^2*d^9 + 7*C*a^8*b^3*c^6*d^5 + 12*C*a^9*b^2*c^3*d^8 + 2*C*a^9*b^2*c^5*d^6 + 16*A*a*b^10*c*d^10)/(a^8*d^8 + b^8*c^8 + 2*a^2*b^6*c^8 + a^4*b^4*c^8 + a^4*b^4*d^8 + 2*a^6*b^2*d^8 + 2*a^8*c^2*d^6 + a^8*c^4*d^4 + b^8*c^4*d^4 + 2*b^8*c^6*d^2 - 4*a*b^7*c^3*d^5 - 8*a*b^7*c^5*d^3 - 4*a^3*b^5*c*d^7 - 8*a^3*b^5*c^7*d - 8*a^5*b^3*c*d^7 - 4*a^5*b^3*c^7*d - 8*a^7*b*c^3*d^5 - 4*a^7*b*c^5*d^3 + 6*a^2*b^6*c^2*d^6 + 14*a^2*b^6*c^4*d^4 + 10*a^2*b^6*c^6*d^2 - 16*a^3*b^5*c^3*d^5 - 20*a^3*b^5*c^5*d^3 + 14*a^4*b^4*c^2*d^6 + 26*a^4*b^4*c^4*d^4 + 14*a^4*b^4*c^6*d^2 - 20*a^5*b^3*c^3*d^5 - 16*a^5*b^3*c^5*d^3 + 10*a^6*b^2*c^2*d^6 + 14*a^6*b^2*c^4*d^4 + 6*a^6*b^2*c^6*d^2 - 4*a*b^7*c^7*d - 4*a^7*b*c*d^7) + (tan(e + f*x)*(3*B*a^10*b*d^11 + 3*B*b^11*c^10*d - 16*A*a^3*b^8*d^11 - 48*A*a^5*b^6*d^11 - 36*A*a^7*b^4*d^11 - 4*A*a^9*b^2*d^11 + 4*B*a^4*b^7*d^11 + 23*B*a^6*b^5*d^11 + 22*B*a^8*b^3*d^11 - 16*A*b^11*c^3*d^8 - 48*A*b^11*c^5*d^6 - 36*A*b^11*c^7*d^4 - 4*A*b^11*c^9*d^2 + 8*C*a^5*b^6*d^11 + 4*C*a^7*b^4*d^11 - 4*C*a^9*b^2*d^11 + 4*B*b^11*c^4*d^7 + 23*B*b^11*c^6*d^5 + 22*B*b^11*c^8*d^3 + 8*C*b^11*c^5*d^6 + 4*C*b^11*c^7*d^4 - 4*C*b^11*c^9*d^2 + 16*A*a*b^10*c^2*d^9 + 80*A*a*b^10*c^4*d^7 + 100*A*a*b^10*c^6*d^5 +

40*A*a*b^10*c^8*d^3 + 16*A*a^2*b^9*c*d^10 + 4*A*a^3*b^8*c^10*d + 80*A*a^4*b
^7*c*d^10 + 100*A*a^6*b^5*c*d^10 + 40*A*a^8*b^3*c*d^10 + 4*A*a^10*b*c^3*d^8
 + 16*B*a*b^10*c^3*d^8 + 6*B*a*b^10*c^5*d^6 - 20*B*a*b^10*c^7*d^4 - 10*B*a*
b^10*c^9*d^2 + 2*B*a^2*b^9*c^10*d + 16*B*a^3*b^8*c*d^10 - B*a^4*b^7*c^10*d
+ 6*B*a^5*b^6*c*d^10 - 20*B*a^7*b^4*c*d^10 - 10*B*a^9*b^2*c*d^10 + 2*B*a^10
*b*c^2*d^9 - B*a^10*b*c^4*d^7 - 40*C*a*b^10*c^4*d^7 - 68*C*a*b^10*c^6*d^5 -
 32*C*a*b^10*c^8*d^3 - 4*C*a^3*b^8*c^10*d - 40*C*a^4*b^7*c*d^10 - 68*C*a^6*
b^5*c*d^10 - 32*C*a^8*b^3*c*d^10 - 4*C*a^10*b*c^3*d^8 - 32*A*a^2*b^9*c^3*d^
8 - 180*A*a^2*b^9*c^5*d^6 - 156*A*a^2*b^9*c^7*d^4 - 24*A*a^2*b^9*c^9*d^2 -
32*A*a^3*b^8*c^2*d^9 + 116*A*a^3*b^8*c^4*d^7 + 204*A*a^3*b^8*c^6*d^5 + 76*A
*a^3*b^8*c^8*d^3 + 116*A*a^4*b^7*c^3*d^8 - 84*A*a^4*b^7*c^5*d^6 - 140*A*a^4
*b^7*c^7*d^4 - 20*A*a^4*b^7*c^9*d^2 - 180*A*a^5*b^6*c^2*d^9 - 84*A*a^5*b^6*
c^4*d^7 + 84*A*a^5*b^6*c^6*d^5 + 36*A*a^5*b^6*c^8*d^3 + 204*A*a^6*b^5*c^3*d
^8 + 84*A*a^6*b^5*c^5*d^6 - 20*A*a^6*b^5*c^7*d^4 - 156*A*a^7*b^4*c^2*d^9 -
140*A*a^7*b^4*c^4*d^7 - 20*A*a^7*b^4*c^6*d^5 + 76*A*a^8*b^3*c^3*d^8 + 36*A*
a^8*b^3*c^5*d^6 - 24*A*a^9*b^2*c^2*d^9 - 20*A*a^9*b^2*c^4*d^7 - 40*B*a^2*b^
9*c^2*d^9 - 103*B*a^2*b^9*c^4*d^7 - 40*B*a^2*b^9*c^6*d^5 + 25*B*a^2*b^9*c^8
*d^3 + 148*B*a^3*b^8*c^3*d^8 + 180*B*a^3*b^8*c^5*d^6 + 44*B*a^3*b^8*c^7*d^4
 - 4*B*a^3*b^8*c^9*d^2 - 103*B*a^4*b^7*c^2*d^9 - 284*B*a^4*b^7*c^4*d^7 - 18
8*B*a^4*b^7*c^6*d^5 - 12*B*a^4*b^7*c^8*d^3 + 180*B*a^5*b^6*c^3*d^8 + 252*B*
a^5*b^6*c^5*d^6 + 84*B*a^5*b^6*c^7*d^4 + 6*B*a^5*b^6*c^9*d^2 - 40*B*a^6*b^5
*c^2*d^9 - 188*B*a^6*b^5*c^4*d^7 - 140*B*a^6*b^5*c^6*d^5 - 15*B*a^6*b^5*c^8
*d^3 + 44*B*a^7*b^4*c^3*d^8 + 84*B*a^7*b^4*c^5*d^6 + 20*B*a^7*b^4*c^7*d^4 +
 25*B*a^8*b^3*c^2*d^9 - 12*B*a^8*b^3*c^4*d^7 - 15*B*a^8*b^3*c^6*d^5 - 4*B*a
^9*b^2*c^3*d^8 + 6*B*a^9*b^2*c^5*d^6 + 32*C*a^2*b^9*c^3*d^8 + 116*C*a^2*b^9
*c^5*d^6 + 92*C*a^2*b^9*c^7*d^4 + 8*C*a^2*b^9*c^9*d^2 + 32*C*a^3*b^8*c^2*d^
9 - 52*C*a^3*b^8*c^4*d^7 - 140*C*a^3*b^8*c^6*d^5 - 60*C*a^3*b^8*c^8*d^3 - 5
2*C*a^4*b^7*c^3*d^8 + 84*C*a^4*b^7*c^5*d^6 + 108*C*a^4*b^7*c^7*d^4 + 12*C*a
^4*b^7*c^9*d^2 + 116*C*a^5*b^6*c^2*d^9 + 84*C*a^5*b^6*c^4*d^7 - 52*C*a^5*b^
6*c^6*d^5 - 28*C*a^5*b^6*c^8*d^3 - 140*C*a^6*b^5*c^3*d^8 - 52*C*a^6*b^5*c^5
*d^6 + 20*C*a^6*b^5*c^7*d^4 + 92*C*a^7*b^4*c^2*d^9 + 108*C*a^7*b^4*c^4*d^7
+ 20*C*a^7*b^4*c^6*d^5 - 60*C*a^8*b^3*c^3*d^8 - 28*C*a^8*b^3*c^5*d^6 + 8*C*
a^9*b^2*c^2*d^9 + 12*C*a^9*b^2*c^4*d^7 + 4*A*a*b^10*c^10*d + 4*A*a^10*b*c*d
^10 - 4*C*a*b^10*c^10*d - 4*C*a^10*b*c*d^10))/(a^8*d^8 + b^8*c^8 + 2*a^2*b^
6*c^8 + a^4*b^4*c^8 + a^4*b^4*d^8 + 2*a^6*b^2*d^8 + 2*a^8*c^2*d^6 + a^8*c^4
*d^4 + b^8*c^4*d^4 + 2*b^8*c^6*d^2 - 4*a*b^7*c^3*d^5 - 8*a*b^7*c^5*d^3 - 4*
a^3*b^5*c*d^7 - 8*a^3*b^5*c^7*d - 8*a^5*b^3*c*d^7 - 4*a^5*b^3*c^7*d - 8*a^7
*b*c^3*d^5 - 4*a^7*b*c^5*d^3 + 6*a^2*b^6*c^2*d^6 + 14*a^2*b^6*c^4*d^4 + 10*
a^2*b^6*c^6*d^2 - 16*a^3*b^5*c^3*d^5 - 20*a^3*b^5*c^5*d^3 + 14*a^4*b^4*c^2*
d^6 + 26*a^4*b^4*c^4*d^4 + 14*a^4*b^4*c^6*d^2 - 20*a^5*b^3*c^3*d^5 - 16*a^5
*b^3*c^5*d^3 + 10*a^6*b^2*c^2*d^6 + 14*a^6*b^2*c^4*d^4 + 6*a^6*b^2*c^6*d^2
- 4*a*b^7*c^7*d - 4*a^7*b*c*d^7)) + (tan(e + f*x)*(8*A^2*b^9*d^9 + 8*A^2*a^
2*b^7*d^9 + 18*A^2*a^4*b^5*d^9 + 2*A^2*a^6*b^3*d^9 + 2*B^2*a^2*b^7*d^9 - 6*
B^2*a^4*b^5*d^9 + 9*B^2*a^6*b^3*d^9 + 8*A^2*b^9*c^2*d^7 + 18*A^2*b^9*c^4*d^
5 + 2*A^2*b^9*c^6*d^3 + 2*C^2*a^4*b^5*d^9 - 14*C^2*a^6*b^3*d^9 + 2*B^2*b^9*

c^2*d^7 - 6*B^2*b^9*c^4*d^5 + 9*B^2*b^9*c^6*d^3 + 2*C^2*b^9*c^4*d^5 - 14*C^
2*b^9*c^6*d^3 + A^2*a^8*b*d^9 + A^2*b^9*c^8*d + C^2*a^8*b*d^9 + C^2*b^9*c^8
*d + 28*A^2*a^2*b^7*c^2*d^7 + 54*A^2*a^2*b^7*c^4*d^5 + 6*A^2*a^2*b^7*c^6*d^
3 - 96*A^2*a^3*b^6*c^3*d^6 - 20*A^2*a^3*b^6*c^5*d^4 + 54*A^2*a^4*b^5*c^2*d^
7 + 42*A^2*a^4*b^5*c^4*d^5 - 20*A^2*a^5*b^4*c^3*d^6 + 6*A^2*a^6*b^3*c^2*d^7
 - 20*B^2*a^2*b^7*c^2*d^7 - 37*B^2*a^2*b^7*c^4*d^5 + 14*B^2*a^2*b^7*c^6*d^3
 - 4*B^2*a^3*b^6*c^3*d^6 - 14*B^2*a^3*b^6*c^5*d^4 - 6*B^2*a^3*b^6*c^7*d^2 -
 37*B^2*a^4*b^5*c^2*d^7 - 28*B^2*a^4*b^5*c^4*d^5 + 9*B^2*a^4*b^5*c^6*d^3 -
14*B^2*a^5*b^4*c^3*d^6 + 14*B^2*a^6*b^3*c^2*d^7 + 9*B^2*a^6*b^3*c^4*d^5 - 6
*B^2*a^7*b^2*c^3*d^6 + 20*C^2*a^2*b^7*c^2*d^7 + 22*C^2*a^2*b^7*c^4*d^5 - 26
*C^2*a^2*b^7*c^6*d^3 - 48*C^2*a^3*b^6*c^3*d^6 - 28*C^2*a^3*b^6*c^5*d^4 + 8*
C^2*a^3*b^6*c^7*d^2 + 22*C^2*a^4*b^5*c^2*d^7 + 18*C^2*a^4*b^5*c^4*d^5 - 8*C
^2*a^4*b^5*c^6*d^3 - 28*C^2*a^5*b^4*c^3*d^6 - 32*C^2*a^5*b^4*c^5*d^4 - 26*C
^2*a^6*b^3*c^2*d^7 - 8*C^2*a^6*b^3*c^4*d^5 + 8*C^2*a^6*b^3*c^6*d^3 + 8*C^2*
a^7*b^2*c^3*d^6 + 4*A*B*a^3*b^6*d^9 - 20*A*B*a^5*b^4*d^9 + 2*A*B*a^7*b^2*d^
9 - 28*A*C*a^4*b^5*d^9 + 4*A*C*a^6*b^3*d^9 + 4*A*B*b^9*c^3*d^6 - 20*A*B*b^9
*c^5*d^4 + 2*A*B*b^9*c^7*d^2 + 28*B*C*a^5*b^4*d^9 - 6*B*C*a^7*b^2*d^9 - 28*
A*C*b^9*c^4*d^5 + 4*A*C*b^9*c^6*d^3 + 28*B*C*b^9*c^5*d^4 - 6*B*C*b^9*c^7*d^
2 - 48*A^2*a*b^8*c*d^8 + 4*B^2*a*b^8*c*d^8 - 72*A^2*a*b^8*c^3*d^6 - 24*A^2*
a*b^8*c^5*d^4 - 4*A^2*a*b^8*c^7*d^2 - 72*A^2*a^3*b^6*c*d^8 - 24*A^2*a^5*b^4
*c*d^8 - 4*A^2*a^7*b^2*c*d^8 - 10*B^2*a*b^8*c^5*d^4 - 2*B^2*a*b^8*c^7*d^2 +
 B^2*a^2*b^7*c^8*d - 10*B^2*a^5*b^4*c*d^8 - 2*B^2*a^7*b^2*c*d^8 + B^2*a^8*b
*c^2*d^7 - 8*C^2*a*b^8*c^3*d^6 + 4*C^2*a*b^8*c^7*d^2 - 8*C^2*a^3*b^6*c*d^8
+ 4*C^2*a^7*b^2*c*d^8 - 8*A*B*a*b^8*d^9 - 2*A*C*a^8*b*d^9 - 8*A*B*b^9*c*d^8
 - 2*A*C*b^9*c^8*d - 2*A*B*a*b^8*c^8*d - 2*A*B*a^8*b*c*d^8 + 16*A*C*a*b^8*c
*d^8 + 2*B*C*a*b^8*c^8*d + 2*B*C*a^8*b*c*d^8 + 28*A*B*a*b^8*c^2*d^7 + 48*A*
B*a*b^8*c^4*d^5 + 2*A*B*a*b^8*c^6*d^3 + 28*A*B*a^2*b^7*c*d^8 + 48*A*B*a^4*b
^5*c*d^8 + 2*A*B*a^6*b^3*c*d^8 + 16*A*C*a*b^8*c^3*d^6 - 8*A*C*a*b^8*c^5*d^4
 + 16*A*C*a^3*b^6*c*d^8 - 8*A*C*a^5*b^4*c*d^8 - 8*B*C*a*b^8*c^2*d^7 - 24*B*
C*a*b^8*c^4*d^5 - 6*B*C*a*b^8*c^6*d^3 - 8*B*C*a^2*b^7*c*d^8 - 24*B*C*a^4*b^
5*c*d^8 - 6*B*C*a^6*b^3*c*d^8 + 52*A*B*a^2*b^7*c^3*d^6 - 22*A*B*a^2*b^7*c^5
*d^4 + 10*A*B*a^2*b^7*c^7*d^2 + 52*A*B*a^3*b^6*c^2*d^7 + 50*A*B*a^3*b^6*c^4
*d^5 - 6*A*B*a^3*b^6*c^6*d^3 + 50*A*B*a^4*b^5*c^3*d^6 - 10*A*B*a^4*b^5*c^5*
d^4 - 22*A*B*a^5*b^4*c^2*d^7 - 10*A*B*a^5*b^4*c^4*d^5 - 6*A*B*a^6*b^3*c^3*d
^6 + 10*A*B*a^7*b^2*c^2*d^7 - 40*A*C*a^2*b^7*c^2*d^7 - 84*A*C*a^2*b^7*c^4*d
^5 + 12*A*C*a^2*b^7*c^6*d^3 + 16*A*C*a^3*b^6*c^3*d^6 - 16*A*C*a^3*b^6*c^5*d
^4 - 8*A*C*a^3*b^6*c^7*d^2 - 84*A*C*a^4*b^5*c^2*d^7 - 52*A*C*a^4*b^5*c^4*d^
5 + 16*A*C*a^4*b^5*c^6*d^3 - 16*A*C*a^5*b^4*c^3*d^6 + 12*A*C*a^6*b^3*c^2*d^
7 + 16*A*C*a^6*b^3*c^4*d^5 - 8*A*C*a^7*b^2*c^3*d^6 + 28*B*C*a^2*b^7*c^3*d^6
 + 82*B*C*a^2*b^7*c^5*d^4 - 10*B*C*a^2*b^7*c^7*d^2 + 28*B*C*a^3*b^6*c^2*d^7
 + 10*B*C*a^3*b^6*c^4*d^5 - 10*B*C*a^3*b^6*c^6*d^3 + 10*B*C*a^4*b^5*c^3*d^6
 + 50*B*C*a^4*b^5*c^5*d^4 + 4*B*C*a^4*b^5*c^7*d^2 + 82*B*C*a^5*b^4*c^2*d^7
+ 50*B*C*a^5*b^4*c^4*d^5 - 12*B*C*a^5*b^4*c^6*d^3 - 10*B*C*a^6*b^3*c^3*d^6
- 12*B*C*a^6*b^3*c^5*d^4 - 10*B*C*a^7*b^2*c^2*d^7 + 4*B*C*a^7*b^2*c^4*d^5))
/(a^8*d^8 + b^8*c^8 + 2*a^2*b^6*c^8 + a^4*b^4*c^8 + a^4*b^4*d^8 + 2*a^6*b^2

*d^8 + 2*a^8*c^2*d^6 + a^8*c^4*d^4 + b^8*c^4*d^4 + 2*b^8*c^6*d^2 - 4*a*b^7*c^3*d^5 - 8*a*b^7*c^5*d^3 - 4*a^3*b^5*c*d^7 - 8*a^3*b^5*c^7*d - 8*a^5*b^3*c*d^7 - 4*a^5*b^3*c^7*d - 8*a^7*b*c^3*d^5 - 4*a^7*b*c^5*d^3 + 6*a^2*b^6*c^2*d^6 + 14*a^2*b^6*c^4*d^4 + 10*a^2*b^6*c^6*d^2 - 16*a^3*b^5*c^3*d^5 - 20*a^3*b^5*c^5*d^3 + 14*a^4*b^4*c^2*d^6 + 26*a^4*b^4*c^4*d^4 + 14*a^4*b^4*c^6*d^2 - 20*a^5*b^3*c^3*d^5 - 16*a^5*b^3*c^5*d^3 + 10*a^6*b^2*c^2*d^6 + 14*a^6*b^2*c^4*d^4 + 6*a^6*b^2*c^6*d^2 - 4*a*b^7*c^7*d - 4*a^7*b*c*d^7))*root(144*a^13*b*c^5*d^9*f^4 + 144*a^9*b^5*c*d^13*f^4 + 144*a^5*b^9*c^13*d*f^4 + 144*a*b^13*c^9*d^5*f^4 + 96*a^13*b*c^7*d^7*f^4 + 96*a^13*b*c^3*d^11*f^4 + 96*a^11*b^3*c*d^13*f^4 + 96*a^7*b^7*c^13*d*f^4 + 96*a^7*b^7*c*d^13*f^4 + 96*a^3*b^11*c^13*d*f^4 + 96*a*b^13*c^11*d^3*f^4 + 96*a*b^13*c^7*d^7*f^4 + 24*a^13*b*c^9*d^5*f^4 + 24*a^9*b^5*c^13*d*f^4 + 24*a^5*b^9*c*d^13*f^4 + 24*a*b^13*c^5*d^9*f^4 + 24*a^13*b*c*d^13*f^4 + 24*a*b^13*c^13*d*f^4 + 3648*a^7*b^7*c^7*d^7*f^4 - 3188*a^8*b^6*c^6*d^8*f^4 - 3188*a^6*b^8*c^8*d^6*f^4 - 2912*a^8*b^6*c^8*d^6*f^4 - 2912*a^6*b^8*c^6*d^8*f^4 + 2592*a^9*b^5*c^7*d^7*f^4 + 2592*a^7*b^7*c^9*d^5*f^4 + 2592*a^7*b^7*c^5*d^9*f^4 + 2592*a^5*b^9*c^7*d^7*f^4 + 2168*a^9*b^5*c^5*d^9*f^4 + 2168*a^5*b^9*c^9*d^5*f^4 - 1776*a^10*b^4*c^6*d^8*f^4 - 1776*a^8*b^6*c^4*d^10*f^4 - 1776*a^6*b^8*c^10*d^4*f^4 - 1776*a^4*b^10*c^8*d^6*f^4 + 1568*a^9*b^5*c^9*d^5*f^4 + 1568*a^5*b^9*c^5*d^9*f^4 - 1344*a^10*b^4*c^8*d^6*f^4 - 1344*a^8*b^6*c^10*d^4*f^4 - 1344*a^6*b^8*c^4*d^10*f^4 - 1344*a^4*b^10*c^6*d^8*f^4 - 1164*a^10*b^4*c^4*d^10*f^4 - 1164*a^4*b^10*c^10*d^4*f^4 + 896*a^11*b^3*c^5*d^9*f^4 + 896*a^9*b^5*c^3*d^11*f^4 + 896*a^5*b^9*c^11*d^3*f^4 + 896*a^3*b^11*c^9*d^5*f^4 + 864*a^11*b^3*c^7*d^7*f^4 + 864*a^7*b^7*c^11*d^3*f^4 + 864*a^7*b^7*c^3*d^11*f^4 + 864*a^3*b^11*c^7*d^7*f^4 - 480*a^10*b^4*c^10*d^4*f^4 - 480*a^4*b^10*c^4*d^10*f^4 + 464*a^11*b^3*c^3*d^11*f^4 + 464*a^3*b^11*c^11*d^3*f^4 - 424*a^12*b^2*c^6*d^8*f^4 - 424*a^8*b^6*c^2*d^12*f^4 - 424*a^6*b^8*c^12*d^2*f^4 - 424*a^2*b^12*c^8*d^6*f^4 + 416*a^11*b^3*c^9*d^5*f^4 + 416*a^9*b^5*c^11*d^3*f^4 + 416*a^5*b^9*c^3*d^11*f^4 + 416*a^3*b^11*c^5*d^9*f^4 - 336*a^12*b^2*c^4*d^10*f^4 - 336*a^10*b^4*c^2*d^12*f^4 - 336*a^4*b^10*c^12*d^2*f^4 - 336*a^2*b^12*c^10*d^4*f^4 - 256*a^12*b^2*c^8*d^6*f^4 - 256*a^8*b^6*c^12*d^2*f^4 - 256*a^6*b^8*c^2*d^12*f^4 - 256*a^2*b^12*c^6*d^8*f^4 - 124*a^12*b^2*c^2*d^12*f^4 - 124*a^2*b^12*c^12*d^2*f^4 + 80*a^11*b^3*c^11*d^3*f^4 + 80*a^3*b^11*c^3*d^11*f^4 - 60*a^12*b^2*c^10*d^4*f^4 - 60*a^10*b^4*c^12*d^2*f^4 - 60*a^4*b^10*c^2*d^12*f^4 - 60*a^2*b^12*c^4*d^10*f^4 - 24*b^14*c^10*d^4*f^4 - 16*b^14*c^12*d^2*f^4 - 16*b^14*c^8*d^6*f^4 - 4*b^14*c^6*d^8*f^4 - 24*a^14*c^4*d^10*f^4 - 16*a^14*c^6*d^8*f^4 - 16*a^14*c^2*d^12*f^4 - 4*a^14*c^8*d^6*f^4 - 24*a^10*b^4*d^14*f^4 - 16*a^12*b^2*d^14*f^4 - 16*a^8*b^6*d^14*f^4 - 4*a^6*b^8*d^14*f^4 - 24*a^4*b^10*c^14*f^4 - 16*a^6*b^8*c^14*f^4 - 16*a^2*b^12*c^14*f^4 - 4*a^8*b^6*c^14*f^4 - 4*b^14*c^14*f^4 - 4*a^14*d^14*f^4 + 36*A*C*a^9*b*c*d^9*f^2 + 36*A*C*a*b^9*c^9*d*f^2 + 32*A*C*a*b^9*c*d^9*f^2 - 552*B*C*a^7*b^3*c^4*d^6*f^2 - 552*B*C*a^4*b^6*c^7*d^3*f^2 - 408*B*C*a^5*b^5*c^4*d^6*f^2 - 408*B*C*a^4*b^6*c^5*d^5*f^2 + 360*B*C*a^6*b^4*c^3*d^7*f^2 + 360*B*C*a^3*b^7*c^6*d^4*f^2 - 248*B*C*a^7*b^3*c^2*d^8*f^2 - 248*B*C*a^2*b^8*c^7*d^3*f^2 + 184*B*C*a^6*b^4*c^5*d^5*f^2 + 184*B*C*a^5*b^5*c^6*d^4*f^2 + 152*B*C*a^8*b^2*c^3*d^7*f^2 - 152*B

*C*a^5*b^5*c^2*d^8*f^2 + 152*B*C*a^3*b^7*c^8*d^2*f^2 - 152*B*C*a^2*b^8*c^5*
d^5*f^2 - 104*B*C*a^7*b^3*c^6*d^4*f^2 - 104*B*C*a^6*b^4*c^7*d^3*f^2 + 64*B*
C*a^8*b^2*c^5*d^5*f^2 + 64*B*C*a^5*b^5*c^8*d^2*f^2 - 56*B*C*a^4*b^6*c^3*d^7
*f^2 - 56*B*C*a^3*b^7*c^4*d^6*f^2 - 24*B*C*a^8*b^2*c^7*d^3*f^2 - 24*B*C*a^7
*b^3*c^8*d^2*f^2 - 24*B*C*a^3*b^7*c^2*d^8*f^2 - 24*B*C*a^2*b^8*c^3*d^7*f^2
- 696*A*C*a^5*b^5*c^5*d^5*f^2 + 536*A*C*a^6*b^4*c^6*d^4*f^2 + 536*A*C*a^6*b
^4*c^4*d^6*f^2 + 536*A*C*a^4*b^6*c^6*d^4*f^2 + 472*A*C*a^4*b^6*c^4*d^6*f^2
- 232*A*C*a^7*b^3*c^5*d^5*f^2 - 232*A*C*a^5*b^5*c^7*d^3*f^2 + 216*A*C*a^3*b
^7*c^3*d^7*f^2 + 168*A*C*a^7*b^3*c^3*d^7*f^2 + 168*A*C*a^3*b^7*c^7*d^3*f^2
- 154*A*C*a^8*b^2*c^2*d^8*f^2 - 154*A*C*a^2*b^8*c^8*d^2*f^2 + 62*A*C*a^8*b^
2*c^6*d^4*f^2 + 62*A*C*a^6*b^4*c^8*d^2*f^2 - 40*A*C*a^7*b^3*c^7*d^3*f^2 - 4
0*A*C*a^5*b^5*c^3*d^7*f^2 - 40*A*C*a^3*b^7*c^5*d^5*f^2 + 32*A*C*a^6*b^4*c^2
*d^8*f^2 + 32*A*C*a^2*b^8*c^6*d^4*f^2 - 32*A*C*a^2*b^8*c^2*d^8*f^2 + 30*A*C
*a^4*b^6*c^2*d^8*f^2 + 30*A*C*a^2*b^8*c^4*d^6*f^2 + 16*A*C*a^8*b^2*c^4*d^6*
f^2 + 16*A*C*a^4*b^6*c^8*d^2*f^2 - 488*A*B*a^6*b^4*c^3*d^7*f^2 - 488*A*B*a^
3*b^7*c^6*d^4*f^2 + 440*A*B*a^7*b^3*c^4*d^6*f^2 + 440*A*B*a^4*b^6*c^7*d^3*f
^2 - 360*A*B*a^6*b^4*c^5*d^5*f^2 - 360*A*B*a^5*b^5*c^6*d^4*f^2 - 192*A*B*a^
8*b^2*c^3*d^7*f^2 - 192*A*B*a^3*b^7*c^8*d^2*f^2 - 168*A*B*a^3*b^7*c^2*d^8*f
^2 - 168*A*B*a^2*b^8*c^3*d^7*f^2 - 152*A*B*a^4*b^6*c^3*d^7*f^2 - 152*A*B*a^
3*b^7*c^4*d^6*f^2 - 120*A*B*a^8*b^2*c^5*d^5*f^2 + 120*A*B*a^7*b^3*c^2*d^8*f
^2 - 120*A*B*a^5*b^5*c^8*d^2*f^2 + 120*A*B*a^5*b^5*c^4*d^6*f^2 - 120*A*B*a^
5*b^5*c^2*d^8*f^2 + 120*A*B*a^4*b^6*c^5*d^5*f^2 + 120*A*B*a^2*b^8*c^7*d^3*f
^2 - 120*A*B*a^2*b^8*c^5*d^5*f^2 + 40*A*B*a^7*b^3*c^6*d^4*f^2 + 40*A*B*a^6*
b^4*c^7*d^3*f^2 - 72*B*C*a^9*b*c^4*d^6*f^2 - 72*B*C*a^4*b^6*c^9*d*f^2 - 64*
B*C*a^4*b^6*c*d^9*f^2 - 64*B*C*a*b^9*c^4*d^6*f^2 - 32*B*C*a^8*b^2*c*d^9*f^2
 - 32*B*C*a*b^9*c^8*d^2*f^2 - 16*B*C*a^2*b^8*c*d^9*f^2 - 16*B*C*a*b^9*c^2*d
^8*f^2 + 8*B*C*a^9*b*c^6*d^4*f^2 - 8*B*C*a^9*b*c^2*d^8*f^2 + 8*B*C*a^6*b^4*
c^9*d*f^2 - 8*B*C*a^2*b^8*c^9*d*f^2 + 104*A*C*a^7*b^3*c*d^9*f^2 + 104*A*C*a
*b^9*c^7*d^3*f^2 + 96*A*C*a^3*b^7*c*d^9*f^2 + 96*A*C*a*b^9*c^3*d^7*f^2 + 72
*A*C*a^9*b*c^3*d^7*f^2 + 72*A*C*a^3*b^7*c^9*d*f^2 + 68*A*C*a^5*b^5*c*d^9*f^
2 + 68*A*C*a*b^9*c^5*d^5*f^2 - 28*A*C*a^9*b*c^5*d^5*f^2 - 28*A*C*a^5*b^5*c^
9*d*f^2 + 80*A*B*a^9*b*c^4*d^6*f^2 + 80*A*B*a^4*b^6*c^9*d*f^2 + 24*A*B*a^8*
b^2*c*d^9*f^2 - 24*A*B*a^6*b^4*c*d^9*f^2 + 24*A*B*a^4*b^6*c*d^9*f^2 - 24*A*
B*a^2*b^8*c*d^9*f^2 + 24*A*B*a*b^9*c^8*d^2*f^2 - 24*A*B*a*b^9*c^6*d^4*f^2 +
 24*A*B*a*b^9*c^4*d^6*f^2 - 24*A*B*a*b^9*c^2*d^8*f^2 - 32*B*C*b^10*c^7*d^3*
f^2 - 8*B*C*b^10*c^5*d^5*f^2 + 34*A*C*b^10*c^6*d^4*f^2 + 16*B*C*a^10*c^3*d^
7*f^2 + 16*A*C*b^10*c^4*d^6*f^2 - 12*A*C*b^10*c^8*d^2*f^2 - 96*A*B*b^10*c^5
*d^5*f^2 - 72*A*B*b^10*c^3*d^7*f^2 - 32*B*C*a^7*b^3*d^10*f^2 - 28*A*C*a^10*
c^2*d^8*f^2 - 24*A*B*b^10*c^7*d^3*f^2 - 8*B*C*a^5*b^5*d^10*f^2 + 2*A*C*a^10
*c^4*d^6*f^2 + 34*A*C*a^6*b^4*d^10*f^2 + 16*B*C*a^3*b^7*c^10*f^2 + 16*A*C*a
^4*b^6*d^10*f^2 - 16*A*B*a^10*c^3*d^7*f^2 - 12*A*C*a^8*b^2*d^10*f^2 - 96*A*
B*a^5*b^5*d^10*f^2 - 72*A*B*a^3*b^7*d^10*f^2 - 28*A*C*a^2*b^8*c^10*f^2 - 24
*A*B*a^7*b^3*d^10*f^2 + 2*A*C*a^4*b^6*c^10*f^2 - 16*A*B*a^3*b^7*c^10*f^2 +
444*C^2*a^5*b^5*c^5*d^5*f^2 + 148*C^2*a^7*b^3*c^5*d^5*f^2 + 148*C^2*a^5*b^5
*c^7*d^3*f^2 + 148*C^2*a^5*b^5*c^3*d^7*f^2 + 148*C^2*a^3*b^7*c^5*d^5*f^2 -

140*C^2*a^6*b^4*c^6*d^4*f^2 - 140*C^2*a^6*b^4*c^4*d^6*f^2 - 140*C^2*a^4*b^6
*c^6*d^4*f^2 - 140*C^2*a^4*b^6*c^4*d^6*f^2 + 109*C^2*a^8*b^2*c^2*d^8*f^2 +
109*C^2*a^2*b^8*c^8*d^2*f^2 + 48*C^2*a^8*b^2*c^4*d^6*f^2 + 48*C^2*a^6*b^4*c
^2*d^8*f^2 + 48*C^2*a^4*b^6*c^8*d^2*f^2 + 48*C^2*a^2*b^8*c^6*d^4*f^2 + 20*C
^2*a^7*b^3*c^7*d^3*f^2 - 20*C^2*a^7*b^3*c^3*d^7*f^2 - 20*C^2*a^3*b^7*c^7*d^
3*f^2 + 20*C^2*a^3*b^7*c^3*d^7*f^2 + 17*C^2*a^8*b^2*c^6*d^4*f^2 + 17*C^2*a^
6*b^4*c^8*d^2*f^2 + 17*C^2*a^4*b^6*c^2*d^8*f^2 + 17*C^2*a^2*b^8*c^4*d^6*f^2
 + 16*C^2*a^8*b^2*c^8*d^2*f^2 + 16*C^2*a^2*b^8*c^2*d^8*f^2 - 396*B^2*a^5*b^
5*c^5*d^5*f^2 + 308*B^2*a^6*b^4*c^4*d^6*f^2 + 308*B^2*a^4*b^6*c^6*d^4*f^2 +
 300*B^2*a^4*b^6*c^4*d^6*f^2 + 284*B^2*a^6*b^4*c^6*d^4*f^2 - 132*B^2*a^7*b^
3*c^5*d^5*f^2 - 132*B^2*a^5*b^5*c^7*d^3*f^2 - 84*B^2*a^5*b^5*c^3*d^7*f^2 -
84*B^2*a^3*b^7*c^5*d^5*f^2 + 61*B^2*a^4*b^6*c^2*d^8*f^2 + 61*B^2*a^2*b^8*c^
4*d^6*f^2 - 59*B^2*a^8*b^2*c^2*d^8*f^2 - 59*B^2*a^2*b^8*c^8*d^2*f^2 + 56*B^
2*a^6*b^4*c^2*d^8*f^2 + 56*B^2*a^2*b^8*c^6*d^4*f^2 + 52*B^2*a^7*b^3*c^3*d^7
*f^2 + 52*B^2*a^3*b^7*c^7*d^3*f^2 + 44*B^2*a^3*b^7*c^3*d^7*f^2 + 33*B^2*a^8
*b^2*c^6*d^4*f^2 + 33*B^2*a^6*b^4*c^8*d^2*f^2 + 20*B^2*a^8*b^2*c^4*d^6*f^2
- 20*B^2*a^7*b^3*c^7*d^3*f^2 + 20*B^2*a^4*b^6*c^8*d^2*f^2 + 8*B^2*a^2*b^8*c
^2*d^8*f^2 + 337*A^2*a^4*b^6*c^2*d^8*f^2 + 337*A^2*a^2*b^8*c^4*d^6*f^2 + 27
2*A^2*a^2*b^8*c^2*d^8*f^2 + 252*A^2*a^5*b^5*c^5*d^5*f^2 + 244*A^2*a^4*b^6*c
^4*d^6*f^2 - 236*A^2*a^3*b^7*c^3*d^7*f^2 + 176*A^2*a^6*b^4*c^2*d^8*f^2 + 17
6*A^2*a^2*b^8*c^6*d^4*f^2 - 148*A^2*a^7*b^3*c^3*d^7*f^2 - 148*A^2*a^3*b^7*c
^7*d^3*f^2 - 140*A^2*a^6*b^4*c^6*d^4*f^2 + 109*A^2*a^8*b^2*c^2*d^8*f^2 + 10
9*A^2*a^2*b^8*c^8*d^2*f^2 - 108*A^2*a^5*b^5*c^3*d^7*f^2 - 108*A^2*a^3*b^7*c
^5*d^5*f^2 + 84*A^2*a^7*b^3*c^5*d^5*f^2 + 84*A^2*a^5*b^5*c^7*d^3*f^2 + 32*A
^2*a^8*b^2*c^4*d^6*f^2 + 32*A^2*a^4*b^6*c^8*d^2*f^2 + 20*A^2*a^7*b^3*c^7*d^
3*f^2 - 15*A^2*a^8*b^2*c^6*d^4*f^2 - 15*A^2*a^6*b^4*c^8*d^2*f^2 - 12*A^2*a^
6*b^4*c^4*d^6*f^2 - 12*A^2*a^4*b^6*c^6*d^4*f^2 + 8*B*C*b^10*c^9*d*f^2 - 16*
B*C*a^10*c*d^9*f^2 - 16*A*B*b^10*c^9*d*f^2 - 16*A*B*b^10*c*d^9*f^2 + 8*B*C*
a^9*b*d^10*f^2 - 16*B*C*a*b^9*c^10*f^2 + 16*A*B*a^10*c*d^9*f^2 - 16*A*B*a^9
*b*d^10*f^2 - 16*A*B*a*b^9*d^10*f^2 + 16*A*B*a*b^9*c^10*f^2 + 22*C^2*a^9*b*
c^5*d^5*f^2 + 22*C^2*a^5*b^5*c^9*d*f^2 + 22*C^2*a^5*b^5*c*d^9*f^2 + 22*C^2*
a*b^9*c^5*d^5*f^2 - 20*C^2*a^9*b*c^3*d^7*f^2 - 20*C^2*a^7*b^3*c*d^9*f^2 - 2
0*C^2*a^3*b^7*c^9*d*f^2 - 20*C^2*a*b^9*c^7*d^3*f^2 + 36*B^2*a^7*b^3*c*d^9*f
^2 + 36*B^2*a*b^9*c^7*d^3*f^2 + 28*B^2*a^9*b*c^3*d^7*f^2 + 28*B^2*a^3*b^7*c
^9*d*f^2 + 24*B^2*a^3*b^7*c*d^9*f^2 + 24*B^2*a*b^9*c^3*d^7*f^2 - 18*B^2*a^9
*b*c^5*d^5*f^2 - 18*B^2*a^5*b^5*c^9*d*f^2 + 6*B^2*a^5*b^5*c*d^9*f^2 + 6*B^2
*a*b^9*c^5*d^5*f^2 - 96*A^2*a^3*b^7*c*d^9*f^2 - 96*A^2*a*b^9*c^3*d^7*f^2 -
90*A^2*a^5*b^5*c*d^9*f^2 - 90*A^2*a*b^9*c^5*d^5*f^2 - 84*A^2*a^7*b^3*c*d^9*
f^2 - 84*A^2*a*b^9*c^7*d^3*f^2 - 52*A^2*a^9*b*c^3*d^7*f^2 - 52*A^2*a^3*b^7*
c^9*d*f^2 + 6*A^2*a^9*b*c^5*d^5*f^2 + 6*A^2*a^5*b^5*c^9*d*f^2 - 10*C^2*a^9*
b*c*d^9*f^2 - 10*C^2*a*b^9*c^9*d*f^2 + 14*B^2*a^9*b*c*d^9*f^2 + 14*B^2*a*b^
9*c^9*d*f^2 + 8*B^2*a*b^9*c*d^9*f^2 - 32*A^2*a*b^9*c*d^9*f^2 - 26*A^2*a^9*b
*c*d^9*f^2 - 26*A^2*a*b^9*c^9*d*f^2 + 2*A*C*b^10*c^10*f^2 + 2*A*C*a^10*d^10
*f^2 + 14*C^2*b^10*c^8*d^2*f^2 - C^2*b^10*c^6*d^4*f^2 + 31*B^2*b^10*c^6*d^4
*f^2 + 20*B^2*b^10*c^4*d^6*f^2 + 14*C^2*a^10*c^2*d^8*f^2 + 4*B^2*b^10*c^2*d

^8*f^2 + 2*B^2*b^10*c^8*d^2*f^2 - C^2*a^10*c^4*d^6*f^2 + 80*A^2*b^10*c^4*d^
6*f^2 + 64*A^2*b^10*c^2*d^8*f^2 + 31*A^2*b^10*c^6*d^4*f^2 + 14*C^2*a^8*b^2*
d^10*f^2 + 14*A^2*b^10*c^8*d^2*f^2 - 10*B^2*a^10*c^2*d^8*f^2 + 3*B^2*a^10*c
^4*d^6*f^2 - C^2*a^6*b^4*d^10*f^2 + 31*B^2*a^6*b^4*d^10*f^2 + 20*B^2*a^4*b^
6*d^10*f^2 + 14*C^2*a^2*b^8*c^10*f^2 + 14*A^2*a^10*c^2*d^8*f^2 + 4*B^2*a^2*
b^8*d^10*f^2 + 2*B^2*a^8*b^2*d^10*f^2 - C^2*a^4*b^6*c^10*f^2 - A^2*a^10*c^4
*d^6*f^2 + 80*A^2*a^4*b^6*d^10*f^2 + 64*A^2*a^2*b^8*d^10*f^2 + 31*A^2*a^6*b
^4*d^10*f^2 + 14*A^2*a^8*b^2*d^10*f^2 - 10*B^2*a^2*b^8*c^10*f^2 + 3*B^2*a^4
*b^6*c^10*f^2 + 14*A^2*a^2*b^8*c^10*f^2 - A^2*a^4*b^6*c^10*f^2 - C^2*b^10*c
^10*f^2 - C^2*a^10*d^10*f^2 + 16*A^2*b^10*d^10*f^2 + 3*B^2*b^10*c^10*f^2 +
3*B^2*a^10*d^10*f^2 - A^2*b^10*c^10*f^2 - A^2*a^10*d^10*f^2 - 96*A*B*C*a*b^
7*c*d^7*f - 28*A*B*C*a^7*b*c*d^7*f - 28*A*B*C*a*b^7*c^7*d*f + 484*A*B*C*a^4
*b^4*c^4*d^4*f - 424*A*B*C*a^3*b^5*c^3*d^5*f + 320*A*B*C*a^2*b^6*c^2*d^6*f
- 176*A*B*C*a^6*b^2*c^2*d^6*f - 176*A*B*C*a^2*b^6*c^6*d^2*f + 158*A*B*C*a^4
*b^4*c^2*d^6*f + 158*A*B*C*a^2*b^6*c^4*d^4*f - 136*A*B*C*a^5*b^3*c^5*d^3*f
- 34*A*B*C*a^6*b^2*c^4*d^4*f - 34*A*B*C*a^4*b^4*c^6*d^2*f + 28*A*B*C*a^5*b^
3*c^3*d^5*f + 28*A*B*C*a^3*b^5*c^5*d^3*f + 308*A*B*C*a^5*b^3*c*d^7*f + 308*
A*B*C*a*b^7*c^5*d^3*f + 20*A*B*C*a^7*b*c^3*d^5*f + 20*A*B*C*a^3*b^5*c^7*d*f
 + 30*B*C^2*a^7*b*c*d^7*f + 30*B*C^2*a*b^7*c^7*d*f + 160*A^2*B*a*b^7*c*d^7*
f - 2*A^2*B*a^7*b*c*d^7*f - 2*A^2*B*a*b^7*c^7*d*f - 96*A*B*C*b^8*c^4*d^4*f
+ 34*A*B*C*b^8*c^6*d^2*f - 32*A*B*C*b^8*c^2*d^6*f + 2*A*B*C*a^8*c^2*d^6*f -
 96*A*B*C*a^4*b^4*d^8*f + 34*A*B*C*a^6*b^2*d^8*f - 32*A*B*C*a^2*b^6*d^8*f +
 2*A*B*C*a^2*b^6*c^8*f - 210*B*C^2*a^4*b^4*c^4*d^4*f - 182*B^2*C*a^5*b^3*c^
2*d^6*f - 182*B^2*C*a^2*b^6*c^5*d^3*f + 180*B*C^2*a^5*b^3*c^5*d^3*f + 180*B
*C^2*a^3*b^5*c^3*d^5*f - 166*B^2*C*a^5*b^3*c^4*d^4*f - 166*B^2*C*a^4*b^4*c^
5*d^3*f + 152*B*C^2*a^6*b^2*c^2*d^6*f + 152*B*C^2*a^2*b^6*c^6*d^2*f - 112*B
^2*C*a^3*b^5*c^2*d^6*f - 112*B^2*C*a^2*b^6*c^3*d^5*f + 94*B^2*C*a^4*b^4*c^3
*d^5*f + 94*B^2*C*a^3*b^5*c^4*d^4*f - 80*B*C^2*a^2*b^6*c^2*d^6*f + 66*B*C^2
*a^5*b^3*c^3*d^5*f + 66*B*C^2*a^3*b^5*c^5*d^3*f + 46*B^2*C*a^6*b^2*c^3*d^5*
f + 46*B^2*C*a^3*b^5*c^6*d^2*f + 33*B*C^2*a^6*b^2*c^4*d^4*f + 33*B*C^2*a^4*
b^4*c^6*d^2*f + 24*B^2*C*a^6*b^2*c^5*d^3*f + 24*B^2*C*a^5*b^3*c^6*d^2*f - 1
6*B*C^2*a^6*b^2*c^6*d^2*f - 15*B*C^2*a^4*b^4*c^2*d^6*f - 15*B*C^2*a^2*b^6*c
^4*d^4*f - 190*A^2*C*a^4*b^4*c^3*d^5*f - 190*A^2*C*a^3*b^5*c^4*d^4*f + 182*
A^2*C*a^5*b^3*c^2*d^6*f + 182*A^2*C*a^2*b^6*c^5*d^3*f + 160*A^2*C*a^3*b^5*c
^2*d^6*f + 160*A^2*C*a^2*b^6*c^3*d^5*f - 150*A*C^2*a^5*b^3*c^2*d^6*f - 150*
A*C^2*a^2*b^6*c^5*d^3*f - 126*A*C^2*a^5*b^3*c^4*d^4*f - 126*A*C^2*a^4*b^4*c
^5*d^3*f + 126*A*C^2*a^4*b^4*c^3*d^5*f + 126*A*C^2*a^3*b^5*c^4*d^4*f - 96*A
*C^2*a^3*b^5*c^2*d^6*f - 96*A*C^2*a^2*b^6*c^3*d^5*f + 94*A^2*C*a^5*b^3*c^4*
d^4*f + 94*A^2*C*a^4*b^4*c^5*d^3*f + 54*A*C^2*a^6*b^2*c^3*d^5*f + 54*A*C^2*
a^3*b^5*c^6*d^2*f + 32*A*C^2*a^6*b^2*c^5*d^3*f + 32*A*C^2*a^5*b^3*c^6*d^2*f
 - 22*A^2*C*a^6*b^2*c^3*d^5*f - 22*A^2*C*a^3*b^5*c^6*d^2*f + 500*A^2*B*a^3*
b^5*c^3*d^5*f - 290*A^2*B*a^4*b^4*c^4*d^4*f - 256*A^2*B*a^2*b^6*c^2*d^6*f -
 230*A*B^2*a^4*b^4*c^3*d^5*f - 230*A*B^2*a^3*b^5*c^4*d^4*f + 142*A*B^2*a^5*
b^3*c^2*d^6*f + 142*A*B^2*a^2*b^6*c^5*d^3*f - 127*A^2*B*a^4*b^4*c^2*d^6*f -
 127*A^2*B*a^2*b^6*c^4*d^4*f + 86*A*B^2*a^5*b^3*c^4*d^4*f + 86*A*B^2*a^4*b^

4*c^5*d^3*f + 80*A*B^2*a^3*b^5*c^2*d^6*f + 80*A*B^2*a^2*b^6*c^3*d^5*f + 40*
A^2*B*a^6*b^2*c^2*d^6*f + 40*A^2*B*a^2*b^6*c^6*d^2*f + 34*A^2*B*a^5*b^3*c^3
*d^5*f + 34*A^2*B*a^3*b^5*c^5*d^3*f - 30*A*B^2*a^6*b^2*c^3*d^5*f - 30*A*B^2
*a^3*b^5*c^6*d^2*f + 20*A^2*B*a^5*b^3*c^5*d^3*f - 15*A^2*B*a^6*b^2*c^4*d^4*
f - 15*A^2*B*a^4*b^4*c^6*d^2*f - 98*B^2*C*a^6*b^2*c*d^7*f - 98*B^2*C*a*b^7*
c^6*d^2*f - 90*B*C^2*a^5*b^3*c*d^7*f - 90*B*C^2*a*b^7*c^5*d^3*f + 48*B^2*C*
a^4*b^4*c*d^7*f + 48*B^2*C*a*b^7*c^4*d^4*f + 40*B^2*C*a^2*b^6*c*d^7*f + 40*
B^2*C*a*b^7*c^2*d^6*f - 32*B*C^2*a^3*b^5*c*d^7*f - 32*B*C^2*a*b^7*c^3*d^5*f
 + 26*B^2*C*a^7*b*c^2*d^6*f + 26*B^2*C*a^2*b^6*c^7*d*f - 26*B*C^2*a^7*b*c^3
*d^5*f - 26*B*C^2*a^3*b^5*c^7*d*f - 8*B^2*C*a^7*b*c^4*d^4*f - 8*B^2*C*a^4*b
^4*c^7*d*f - 224*A^2*C*a^4*b^4*c*d^7*f - 224*A^2*C*a*b^7*c^4*d^4*f - 96*A^2
*C*a^2*b^6*c*d^7*f - 96*A^2*C*a*b^7*c^2*d^6*f + 96*A*C^2*a^4*b^4*c*d^7*f +
96*A*C^2*a*b^7*c^4*d^4*f - 66*A*C^2*a^6*b^2*c*d^7*f - 66*A*C^2*a*b^7*c^6*d^
2*f + 64*A*C^2*a^2*b^6*c*d^7*f + 64*A*C^2*a*b^7*c^2*d^6*f + 34*A^2*C*a^6*b^
2*c*d^7*f + 34*A^2*C*a*b^7*c^6*d^2*f + 34*A*C^2*a^7*b*c^2*d^6*f + 34*A*C^2*
a^2*b^6*c^7*d*f - 2*A^2*C*a^7*b*c^2*d^6*f - 2*A^2*C*a^2*b^6*c^7*d*f - 208*A
*B^2*a^4*b^4*c*d^7*f - 208*A*B^2*a*b^7*c^4*d^4*f + 160*A^2*B*a^3*b^5*c*d^7*
f + 160*A^2*B*a*b^7*c^3*d^5*f - 154*A^2*B*a^5*b^3*c*d^7*f - 154*A^2*B*a*b^7
*c^5*d^3*f - 112*A*B^2*a^2*b^6*c*d^7*f - 112*A*B^2*a*b^7*c^2*d^6*f + 58*A*B
^2*a^6*b^2*c*d^7*f + 58*A*B^2*a*b^7*c^6*d^2*f - 10*A*B^2*a^7*b*c^2*d^6*f -
10*A*B^2*a^2*b^6*c^7*d*f + 6*A^2*B*a^7*b*c^3*d^5*f + 6*A^2*B*a^3*b^5*c^7*d*
f + 32*B^2*C*b^8*c^5*d^3*f - 17*B*C^2*b^8*c^6*d^2*f + 8*B^2*C*b^8*c^3*d^5*f
 + 64*A^2*C*b^8*c^3*d^5*f - 32*A^2*C*b^8*c^5*d^3*f + 32*A*C^2*b^8*c^5*d^3*f
 - B*C^2*a^8*c^2*d^6*f + 112*A^2*B*b^8*c^4*d^4*f - 64*A*B^2*b^8*c^5*d^3*f +
 32*B^2*C*a^5*b^3*d^8*f - 17*B*C^2*a^6*b^2*d^8*f + 16*A^2*B*b^8*c^2*d^6*f +
 16*A*B^2*b^8*c^3*d^5*f + 8*B^2*C*a^3*b^5*d^8*f - A^2*B*b^8*c^6*d^2*f + 64*
A^2*C*a^3*b^5*d^8*f - 32*A^2*C*a^5*b^3*d^8*f + 32*A*C^2*a^5*b^3*d^8*f - A^2
*B*a^8*c^2*d^6*f - B*C^2*a^2*b^6*c^8*f + 112*A^2*B*a^4*b^4*d^8*f - 64*A*B^2
*a^5*b^3*d^8*f + 16*A^2*B*a^2*b^6*d^8*f + 16*A*B^2*a^3*b^5*d^8*f - A^2*B*a^
6*b^2*d^8*f - A^2*B*a^2*b^6*c^8*f - 8*B^3*a*b^7*c*d^7*f - 2*B^3*a^7*b*c*d^7
*f - 2*B^3*a*b^7*c^7*d*f - 6*B^2*C*b^8*c^7*d*f + 32*A^2*C*b^8*c*d^7*f + 6*A
^2*C*b^8*c^7*d*f - 6*A*C^2*b^8*c^7*d*f - 2*B^2*C*a^8*c*d^7*f + 16*A*B^2*b^8
*c*d^7*f - 6*B^2*C*a^7*b*d^8*f - 6*A^2*C*a^8*c*d^7*f + 6*A*C^2*a^8*c*d^7*f
- 2*A*B^2*b^8*c^7*d*f + 32*A^2*C*a*b^7*d^8*f + 6*A^2*C*a^7*b*d^8*f - 6*A*C^
2*a^7*b*d^8*f - 2*B^2*C*a*b^7*c^8*f + 2*A*B^2*a^8*c*d^7*f + 16*A*B^2*a*b^7*
d^8*f - 6*A^2*C*a*b^7*c^8*f + 6*A*C^2*a*b^7*c^8*f - 2*A*B^2*a^7*b*d^8*f + 2
*A*B^2*a*b^7*c^8*f - 50*C^3*a^6*b^2*c^3*d^5*f + 50*C^3*a^5*b^3*c^2*d^6*f -
50*C^3*a^3*b^5*c^6*d^2*f + 50*C^3*a^2*b^6*c^5*d^3*f + 42*C^3*a^5*b^3*c^4*d^
4*f + 42*C^3*a^4*b^4*c^5*d^3*f - 42*C^3*a^4*b^4*c^3*d^5*f - 42*C^3*a^3*b^5*
c^4*d^4*f - 32*C^3*a^6*b^2*c^5*d^3*f - 32*C^3*a^5*b^3*c^6*d^2*f + 32*C^3*a^
3*b^5*c^2*d^6*f + 32*C^3*a^2*b^6*c^3*d^5*f + 94*B^3*a^4*b^4*c^4*d^4*f + 48*
B^3*a^2*b^6*c^2*d^6*f - 44*B^3*a^3*b^5*c^3*d^5*f - 32*B^3*a^6*b^2*c^2*d^6*f
 - 32*B^3*a^2*b^6*c^6*d^2*f + 29*B^3*a^4*b^4*c^2*d^6*f + 29*B^3*a^2*b^6*c^4
*d^4*f - 20*B^3*a^5*b^3*c^5*d^3*f + 18*B^3*a^5*b^3*c^3*d^5*f + 18*B^3*a^3*b
^5*c^5*d^3*f - 3*B^3*a^6*b^2*c^4*d^4*f - 3*B^3*a^4*b^4*c^6*d^2*f + 106*A^3*

a^4*b^4*c^3*d^5*f + 106*A^3*a^3*b^5*c^4*d^4*f - 96*A^3*a^3*b^5*c^2*d^6*f -
96*A^3*a^2*b^6*c^3*d^5*f - 82*A^3*a^5*b^3*c^2*d^6*f - 82*A^3*a^2*b^6*c^5*d^
3*f + 18*A^3*a^6*b^2*c^3*d^5*f + 18*A^3*a^3*b^5*c^6*d^2*f - 10*A^3*a^5*b^3*
c^4*d^4*f - 10*A^3*a^4*b^4*c^5*d^3*f - 22*C^3*a^7*b*c^2*d^6*f + 22*C^3*a^6*
b^2*c*d^7*f - 22*C^3*a^2*b^6*c^7*d*f + 22*C^3*a*b^7*c^6*d^2*f - 2*A*B*C*b^8
*c^8*f - 2*A*B*C*a^8*d^8*f + 62*B^3*a^5*b^3*c*d^7*f + 62*B^3*a*b^7*c^5*d^3*
f + 16*B^3*a^3*b^5*c*d^7*f + 16*B^3*a*b^7*c^3*d^5*f + 6*B^3*a^7*b*c^3*d^5*f
 + 6*B^3*a^3*b^5*c^7*d*f + 128*A^3*a^4*b^4*c*d^7*f + 128*A^3*a*b^7*c^4*d^4*
f + 32*A^3*a^2*b^6*c*d^7*f + 32*A^3*a*b^7*c^2*d^6*f - 10*A^3*a^7*b*c^2*d^6*
f + 10*A^3*a^6*b^2*c*d^7*f - 10*A^3*a^2*b^6*c^7*d*f + 10*A^3*a*b^7*c^6*d^2*
f + 11*B^3*b^8*c^6*d^2*f - 8*B^3*b^8*c^4*d^4*f - 4*B^3*b^8*c^2*d^6*f - 64*A
^3*b^8*c^3*d^5*f - B^3*a^8*c^2*d^6*f + 11*B^3*a^6*b^2*d^8*f - 8*B^3*a^4*b^4
*d^8*f - 4*B^3*a^2*b^6*d^8*f - 64*A^3*a^3*b^5*d^8*f - B^3*a^2*b^6*c^8*f + 2
*C^3*b^8*c^7*d*f - 2*C^3*a^8*c*d^7*f - 32*A^3*b^8*c*d^7*f + 2*C^3*a^7*b*d^8
*f - 2*A^3*b^8*c^7*d*f - 2*C^3*a*b^7*c^8*f + 2*A^3*a^8*c*d^7*f - 32*A^3*a*b
^7*d^8*f - 2*A^3*a^7*b*d^8*f + 2*A^3*a*b^7*c^8*f - 16*A^2*B*b^8*d^8*f + B*C
^2*b^8*c^8*f + B*C^2*a^8*d^8*f + A^2*B*b^8*c^8*f + A^2*B*a^8*d^8*f + B^3*b^
8*c^8*f + B^3*a^8*d^8*f - 4*A*B^2*C*a^5*b*c*d^5 - 4*A*B^2*C*a*b^5*c^5*d + 4
*A*B^2*C*a*b^5*c*d^5 + 22*A^2*B*C*a^3*b^3*c^2*d^4 + 22*A^2*B*C*a^2*b^4*c^3*
d^3 - 20*A*B^2*C*a^3*b^3*c^3*d^3 + 14*A*B^2*C*a^4*b^2*c^2*d^4 + 14*A*B^2*C*
a^2*b^4*c^4*d^2 - 14*A*B*C^2*a^3*b^3*c^2*d^4 - 14*A*B*C^2*a^2*b^4*c^3*d^3 +
 12*A*B*C^2*a^4*b^2*c^3*d^3 + 12*A*B*C^2*a^3*b^3*c^4*d^2 - 6*A^2*B*C*a^4*b^
2*c^3*d^3 - 6*A^2*B*C*a^3*b^3*c^4*d^2 - 4*A*B^2*C*a^2*b^4*c^2*d^4 + 22*A*B*
C^2*a^4*b^2*c*d^5 + 22*A*B*C^2*a*b^5*c^4*d^2 - 20*A^2*B*C*a^4*b^2*c*d^5 - 2
0*A^2*B*C*a*b^5*c^4*d^2 + 10*A*B*C^2*a^2*b^4*c*d^5 + 10*A*B*C^2*a*b^5*c^2*d
^4 - 8*A^2*B*C*a^2*b^4*c*d^5 - 8*A^2*B*C*a*b^5*c^2*d^4 + 4*A*B^2*C*a^3*b^3*
c*d^5 + 4*A*B^2*C*a*b^5*c^3*d^3 - 4*A*B*C^2*a^5*b*c^2*d^4 - 4*A*B*C^2*a^2*b
^4*c^5*d + 2*A^2*B*C*a^5*b*c^2*d^4 + 2*A^2*B*C*a^2*b^4*c^5*d - 8*B^3*C*a^4*
b^2*c*d^5 - 8*B^3*C*a*b^5*c^4*d^2 - 8*B*C^3*a^4*b^2*c*d^5 - 8*B*C^3*a*b^5*c
^4*d^2 - 4*B^3*C*a^2*b^4*c*d^5 - 4*B^3*C*a*b^5*c^2*d^4 + 4*B^2*C^2*a^5*b*c*
d^5 + 4*B^2*C^2*a*b^5*c^5*d - 4*B*C^3*a^2*b^4*c*d^5 - 4*B*C^3*a*b^5*c^2*d^4
 + 2*B^3*C*a^5*b*c^2*d^4 + 2*B^3*C*a^2*b^4*c^5*d + 2*B^2*C^2*a*b^5*c*d^5 +
2*B*C^3*a^5*b*c^2*d^4 + 2*B*C^3*a^2*b^4*c^5*d + 24*A^3*C*a^3*b^3*c*d^5 + 24
*A^3*C*a*b^5*c^3*d^3 - 24*A^2*C^2*a*b^5*c*d^5 + 12*A^2*C^2*a^5*b*c*d^5 + 12
*A^2*C^2*a*b^5*c^5*d + 8*A*C^3*a^3*b^3*c*d^5 + 8*A*C^3*a*b^5*c^3*d^3 + 6*A^
3*B*a^4*b^2*c*d^5 + 6*A^3*B*a*b^5*c^4*d^2 - 6*A^2*B^2*a*b^5*c*d^5 + 6*A*B^3
*a^4*b^2*c*d^5 + 6*A*B^3*a*b^5*c^4*d^2 + 2*A^3*B*a^2*b^4*c*d^5 + 2*A^3*B*a*
b^5*c^2*d^4 + 2*A*B^3*a^2*b^4*c*d^5 + 2*A*B^3*a*b^5*c^2*d^4 + 20*A^2*B*C*b^
6*c^3*d^3 - 10*A*B*C^2*b^6*c^3*d^3 - 2*A*B^2*C*b^6*c^4*d^2 - 2*A*B^2*C*b^6*
c^2*d^4 + 20*A^2*B*C*a^3*b^3*d^6 - 10*A*B*C^2*a^3*b^3*d^6 - 2*A*B^2*C*a^4*b
^2*d^6 - 2*A*B^2*C*a^2*b^4*d^6 + 10*B^2*C^2*a^3*b^3*c^3*d^3 + 4*B^2*C^2*a^4
*b^2*c^4*d^2 - 3*B^2*C^2*a^4*b^2*c^2*d^4 - 3*B^2*C^2*a^2*b^4*c^4*d^2 + 2*B^
2*C^2*a^2*b^4*c^2*d^4 + 40*A^2*C^2*a^2*b^4*c^2*d^4 - 16*A^2*C^2*a^4*b^2*c^2
*d^4 - 16*A^2*C^2*a^2*b^4*c^4*d^2 + 4*A^2*C^2*a^4*b^2*c^4*d^2 + 18*A^2*B^2*
a^2*b^4*c^2*d^4 + 10*A^2*B^2*a^3*b^3*c^3*d^3 - 3*A^2*B^2*a^4*b^2*c^2*d^4 -

```
3*A^2*B^2*a^2*b^4*c^4*d^2 + 24*A^3*C*a*b^5*c*d^5 - 12*A*C^3*a^5*b*c*d^5 - 1
2*A*C^3*a*b^5*c^5*d + 8*A*C^3*a*b^5*c*d^5 - 4*A^3*C*a^5*b*c*d^5 - 4*A^3*C*a
*b^5*c^5*d + 8*A^2*B*C*b^6*c*d^5 + 4*A*B*C^2*b^6*c^5*d - 4*A*B*C^2*b^6*c*d^
5 - 2*A^2*B*C*b^6*c^5*d + 8*A^2*B*C*a*b^5*d^6 + 4*A*B*C^2*a^5*b*d^6 - 4*A*B
*C^2*a*b^5*d^6 - 2*A^2*B*C*a^5*b*d^6 - 6*B^3*C*a^4*b^2*c^3*d^3 - 6*B^3*C*a^
3*b^3*c^4*d^2 - 6*B*C^3*a^4*b^2*c^3*d^3 - 6*B*C^3*a^3*b^3*c^4*d^2 + 2*B^3*C
*a^3*b^3*c^2*d^4 + 2*B^3*C*a^2*b^4*c^3*d^3 + 2*B^2*C^2*a^3*b^3*c*d^5 + 2*B^
2*C^2*a*b^5*c^3*d^3 + 2*B*C^3*a^3*b^3*c^2*d^4 + 2*B*C^3*a^2*b^4*c^3*d^3 - 4
8*A^3*C*a^2*b^4*c^2*d^4 - 24*A^2*C^2*a^3*b^3*c*d^5 - 24*A^2*C^2*a*b^5*c^3*d
^3 - 16*A*C^3*a^2*b^4*c^2*d^4 + 8*A^3*C*a^4*b^2*c^2*d^4 + 8*A^3*C*a^2*b^4*c
^4*d^2 - 8*A*C^3*a^4*b^2*c^4*d^2 + 8*A*C^3*a^4*b^2*c^2*d^4 + 8*A*C^3*a^2*b^
4*c^4*d^2 - 10*A^3*B*a^3*b^3*c^2*d^4 - 10*A^3*B*a^2*b^4*c^3*d^3 - 10*A*B^3*
a^3*b^3*c^2*d^4 - 10*A*B^3*a^2*b^4*c^3*d^3 - 6*A^2*B^2*a^3*b^3*c*d^5 - 6*A^
2*B^2*a*b^5*c^3*d^3 + 3*B^2*C^2*b^6*c^4*d^2 - 8*A^2*C^2*b^6*c^4*d^2 + 8*A^2
*C^2*b^6*c^2*d^4 + 9*A^2*B^2*b^6*c^2*d^4 + 3*B^2*C^2*a^4*b^2*d^6 + 3*A^2*B^
2*b^6*c^4*d^2 - 8*A^2*C^2*a^4*b^2*d^6 + 8*A^2*C^2*a^2*b^4*d^6 + 9*A^2*B^2*a
^2*b^4*d^6 + 3*A^2*B^2*a^4*b^2*d^6 + 2*B^4*a^3*b^3*c*d^5 + 2*B^4*a*b^5*c^3*
d^3 - 8*A^4*a^3*b^3*c*d^5 - 8*A^4*a*b^5*c^3*d^3 - 16*A^3*C*b^6*c^2*d^4 + 4*
A^3*C*b^6*c^4*d^2 + 4*A*C^3*b^6*c^4*d^2 - 10*A^3*B*b^6*c^3*d^3 - 10*A*B^3*b
^6*c^3*d^3 - 16*A^3*C*a^2*b^4*d^6 + 4*A^3*C*a^4*b^2*d^6 + 4*A*C^3*a^4*b^2*d
^6 - 10*A^3*B*a^3*b^3*d^6 - 10*A*B^3*a^3*b^3*d^6 + 4*C^4*a^5*b*c*d^5 + 4*C^
4*a*b^5*c^5*d + 2*B^4*a*b^5*c*d^5 - 8*A^4*a*b^5*c*d^5 - 2*B^3*C*b^6*c^5*d -
 2*B*C^3*b^6*c^5*d - 4*A^3*B*b^6*c*d^5 - 4*A*B^3*b^6*c*d^5 - 2*B^3*C*a^5*b*
d^6 - 2*B*C^3*a^5*b*d^6 - 4*A^3*B*a*b^5*d^6 - 4*A*B^3*a*b^5*d^6 + 4*C^4*a^4
*b^2*c^4*d^2 + 4*C^4*a^2*b^4*c^2*d^4 + 10*B^4*a^3*b^3*c^3*d^3 - 3*B^4*a^4*b
^2*c^2*d^4 - 3*B^4*a^2*b^4*c^4*d^2 - 2*B^4*a^2*b^4*c^2*d^4 + 20*A^4*a^2*b^4
*c^2*d^4 + B^2*C^2*b^6*c^2*d^4 + B^2*C^2*a^2*b^4*d^6 - 8*A^3*C*b^6*d^6 + 3*
B^4*b^6*c^4*d^2 + 8*A^4*b^6*c^2*d^4 + 3*B^4*a^4*b^2*d^6 + 8*A^4*a^2*b^4*d^6
 + 4*A^2*C^2*b^6*d^6 + 4*A^2*B^2*b^6*d^6 + 4*A^4*b^6*d^6 + B^4*b^6*c^2*d^4
+ B^4*a^2*b^4*d^6, f, k), k, 1, 4) - ((A*a^3*d^3 + A*b^3*c^3 + A*a*b^2*d^3
- B*a*b^2*c^3 + C*a^2*b*c^3 + A*b^3*c*d^2 - B*a^3*c*d^2 + C*a^3*c^2*d - 2*B
*a*b^2*c*d^2 + C*a*b^2*c^2*d + C*a^2*b*c*d^2)/((a^2*d^2 + b^2*c^2 - 2*a*b*c
*d)*(a^2*c^2 + a^2*d^2 + b^2*c^2 + b^2*d^2)) + (tan(e + f*x)*(2*A*b^3*d^3 +
 A*a^2*b*d^3 - B*a*b^2*d^3 + A*b^3*c^2*d + C*a^2*b*d^3 - B*b^3*c*d^2 + C*b^
3*c^2*d - B*a*b^2*c^2*d - B*a^2*b*c*d^2 + 2*C*a^2*b*c^2*d))/((a^2*d^2 + b^2
*c^2 - 2*a*b*c*d)*(a^2*c^2 + a^2*d^2 + b^2*c^2 + b^2*d^2)))/(a*c + tan(e +
f*x)*(a*d + b*c) + b*d*tan(e + f*x)^2))/f
```

3.83 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^3(c+d\tan(e+fx))^2}\,dx$

Optimal result 885
Rubi [A] (verified) 886
Mathematica [B] (verified) 889
Maple [A] (verified) 890
Fricas [B] (verification not implemented) 891
Sympy [F(-1)] 891
Maxima [B] (verification not implemented) 892
Giac [B] (verification not implemented) 893
Mupad [B] (verification not implemented) 895

Optimal result

Integrand size = 45, antiderivative size = 841

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^3(c+d\tan(e+fx))^2}\,dx =$$

$$-\frac{(a^3(c^2C-2Bcd-Cd^2-A(c^2-d^2))-3ab^2(c^2C-2Bcd-Cd^2-A(c^2-d^2))+3a^2b(2c(A-C)d-}{(a^2+b^2)^3(c^2+d^2)^2}$$

$$-\frac{b(6a^5bBd^2-3a^6Cd^2-a^4b^2d(4Bc+(10A-C)d)-b^6(c(cC-2Bd)-A(c^2-3d^2))+ab^5(2c(A-C)c}{}$$

$$-\frac{d^2(b(3c^4C-4Bc^3d+c^2(5A+C)d^2-2Bcd^3+3Ad^4)-ad^2(2c(A-C)d-B(c^2-d^2)))\log(c\cos(e+}{(bc-ad)^4(c^2+d^2)^2f}$$

$$-\frac{d(3a^3bBd(c^2+d^2)+ab^3(2Ac-2cC+Bd)(c^2+d^2)-a^4d(3c^2C-Bcd+(A+2C)d^2)-a^2b^2(Bc^3+}{(a^2+b^2)^2(bc-ad)^3(c^2+d^2)f(c+d}$$

$$-\frac{Ab^2-a(bB-aC)}{2(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))^2(c+d\tan(e+fx))}$$

$$-\frac{5a^3bBd-3a^4Cd+b^4(2Bc-3Ad)+ab^3(4Ac-4cC+Bd)-a^2b^2(2Bc+(7A-C)d)}{2(a^2+b^2)^2(bc-ad)^2f(a+b\tan(e+fx))(c+d\tan(e+fx))}$$

```
[Out] -(a^3*(c^2*C-2*B*c*d-C*d^2-A*(c^2-d^2))-3*a*b^2*(c^2*C-2*B*c*d-C*d^2-A*(c^2
-d^2))+3*a^2*b*(2*c*(A-C)*d-B*(c^2-d^2))-b^3*(2*c*(A-C)*d-B*(c^2-d^2)))*x/(
a^2+b^2)^3/(c^2+d^2)^2-b*(6*a^5*b*B*d^2-3*a^6*C*d^2-a^4*b^2*d*(4*B*c+(10*A-
C)*d)-b^6*(c*(-2*B*d+C*c)-A*(c^2-3*d^2))+a*b^5*(2*c*(A-C)*d-B*(3*c^2-d^2))+
3*a^2*b^4*(c*(2*B*d+C*c)-A*(c^2+3*d^2))+a^3*b^3*(10*c*(A-C)*d+B*(c^2+3*d^2)
))*ln(a*cos(f*x+e)+b*sin(f*x+e))/(a^2+b^2)^3/(-a*d+b*c)^4/f-d^2*(b*(3*c^4*C
-4*B*c^3*d+c^2*(5*A+C)*d^2-2*B*c*d^3+3*A*d^4)-a*d^2*(2*c*(A-C)*d-B*(c^2-d^2
)))*ln(c*cos(f*x+e)+d*sin(f*x+e))/(-a*d+b*c)^4/(c^2+d^2)^2/f-d*(3*a^3*b*B*d
*(c^2+d^2)+a*b^3*(2*A*c+B*d-2*C*c)*(c^2+d^2)-a^4*d*(3*c^2*C-B*c*d+(A+2*C)*d
^2)-a^2*b^2*(4*A*c^2*d+6*A*d^3+B*c^3-B*c*d^2+2*C*c^2*d)-b^4*(d*(2*A*c^2+3*A
```

```
*d^2+C*c^2)-B*(c^3+2*c*d^2)))/(a^2+b^2)^2/(-a*d+b*c)^3/(c^2+d^2)/f/(c+d*tan
(f*x+e))+1/2*(-A*b^2+a*(B*b-C*a))/(a^2+b^2)/(-a*d+b*c)/f/(a+b*tan(f*x+e))^2
/(c+d*tan(f*x+e))+1/2*(-5*a^3*b*B*d+3*a^4*C*d-b^4*(-3*A*d+2*B*c)-a*b^3*(4*A
*c+B*d-4*C*c)+a^2*b^2*(2*B*c+(7*A-C)*d))/(a^2+b^2)^2/(-a*d+b*c)^2/f/(a+b*ta
n(f*x+e))/(c+d*tan(f*x+e))
```

Rubi [A] (verified)

Time = 4.43 (sec) , antiderivative size = 841, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 3, $\frac{\text{number of rules}}{\text{integrand size}} = 0.067$, Rules used = {3730, 3732, 3611}

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^3(c+d\tan(e+fx))^2}\,dx =$$

$$-\frac{(b(3Cc^4-4Bdc^3+(5A+C)d^2c^2-2Bd^3c+3Ad^4)-ad^2(2c(A-C)d-B(c^2-d^2)))\log(c\cos(e+fx)}{(bc-ad)^4\left(c^2+d^2\right)^2f}$$

$$-\frac{(-d(3Cc^2-Bdc+(A+2C)d^2)\,a^4+3bBd(c^2+d^2)\,a^3-b^2(Bc^3+4Adc^2+2Cdc^2-Bd^2c+6Ad^3)\,a^2+}{\left(a^2+b^2\right)^2(bc-ad)^3\left(c^2+d^2\right)f(c+d}$$

$$-\frac{((Cc^2-2Bdc-Cd^2-A(c^2-d^2))\,a^3+3b(2c(A-C)d-B(c^2-d^2))\,a^2-3b^2(Cc^2-2Bdc-Cd^2-A}{\left(a^2+b^2\right)^3\left(c^2+d^2\right)^2}$$

$$-\frac{b(-3Cd^2a^6+6bBd^2a^5-b^2d(4Bc+(10A-C)d)a^4+b^3(10c(A-C)d+B(c^2+3d^2))\,a^3+3b^4(c(cC+2}{}$$

$$-\frac{-3Cda^4+5bBda^3-b^2(2Bc+(7A-C)d)a^2+b^3(4Ac-4Cc+Bd)a+b^4(2Bc-3Ad)}{2\left(a^2+b^2\right)^2(bc-ad)^2f(a+b\tan(e+fx))(c+d\tan(e+fx))}$$

$$-\frac{Ab^2-a(bB-aC)}{2\left(a^2+b^2\right)(bc-ad)f(a+b\tan(e+fx))^2(c+d\tan(e+fx))}$$

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^3*(c + d*
Tan[e + f*x])^2),x]
```

```
[Out] -(((a^3*(c^2*C - 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - 3*a*b^2*(c^2*C - 2*B*c*
d - C*d^2 - A*(c^2 - d^2)) + 3*a^2*b*(2*c*(A - C)*d - B*(c^2 - d^2)) - b^3*
(2*c*(A - C)*d - B*(c^2 - d^2)))*x)/((a^2 + b^2)^3*(c^2 + d^2)^2)) - (b*(6*
a^5*b*B*d^2 - 3*a^6*C*d^2 - a^4*b^2*d*(4*B*c + (10*A - C)*d) - b^6*(c*(c*C
 - 2*B*d) - A*(c^2 - 3*d^2)) + a*b^5*(2*c*(A - C)*d - B*(3*c^2 - d^2)) + 3*a
^2*b^4*(c*(c*C + 2*B*d) - A*(c^2 + 3*d^2)) + a^3*b^3*(10*c*(A - C)*d + B*(c
^2 + 3*d^2)))*Log[a*Cos[e + f*x] + b*Sin[e + f*x]])/((a^2 + b^2)^3*(b*c - a
*d)^4*f) - (d^2*(b*(3*c^4*C - 4*B*c^3*d + c^2*(5*A + C)*d^2 - 2*B*c*d^3 + 3
*A*d^4) - a*d^2*(2*c*(A - C)*d - B*(c^2 - d^2)))*Log[c*Cos[e + f*x] + d*Sin
[e + f*x]])/((b*c - a*d)^4*(c^2 + d^2)^2*f) - (d*(3*a^3*b*B*d*(c^2 + d^2) +
 a*b^3*(2*A*c - 2*c*C + B*d)*(c^2 + d^2) - a^4*d*(3*c^2*C - B*c*d + (A + 2*
C)*d^2) - a^2*b^2*(B*c^3 + 4*A*c^2*d + 2*c^2*C*d - B*c*d^2 + 6*A*d^3) - b^4
*(d*(2*A*c^2 + c^2*C + 3*A*d^2) - B*(c^3 + 2*c*d^2))))/((a^2 + b^2)^2*(b*c
```

```
- a*d)^3*(c^2 + d^2)*f*(c + d*Tan[e + f*x])) - (A*b^2 - a*(b*B - a*C))/(2*(
a^2 + b^2)*(b*c - a*d)*f*(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])) - (5*
a^3*b*B*d - 3*a^4*C*d + b^4*(2*B*c - 3*A*d) + a*b^3*(4*A*c - 4*c*C + B*d) -
 a^2*b^2*(2*B*c + (7*A - C)*d))/(2*(a^2 + b^2)^2*(b*c - a*d)^2*f*(a + b*Tan
[e + f*x])*(c + d*Tan[e + f*x]))
```

Rule 3611

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c/(b*f))*Log[RemoveContent[a*Cos[e + f*x] + b*Si
n[e + f*x], x]], x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3730

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 - a*(b*B - a*C))*(a + b*Tan[e +
 f*x])^(m + 1)*((c + d*Tan[e + f*x])^(n + 1)/(f*(m + 1)*(b*c - a*d)*(a^2 +
b^2))), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3732

```
Int[((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^
2)/(((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])), x_Symbol] :> Simp[(a*(A*c - c*C + B*d) + b*(B*c - A*d + C*d))*(x/
((a^2 + b^2)*(c^2 + d^2))), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/((b*c - a*d)
*(a^2 + b^2)), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] - Dist
[(c^2*C - B*c*d + A*d^2)/((b*c - a*d)*(c^2 + d^2)), Int[(d - c*Tan[e + f*x]
)/(c + d*Tan[e + f*x]), x], x]) /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] &&
 NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0]
```

Rubi steps

$$\text{integral} = -\frac{Ab^2 - a(bB - aC)}{2(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))^2(c+d\tan(e+fx))} - \frac{\int \frac{3Ab^2d - 2aA(bc-ad) - (bB-aC)(2bc+ad) + 2(Ab-aB-bC)(bc-ad)\tan(e+fx) + 3(Ab^2-a(bB-aC))d\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^2}\,dx}{2(a^2+b^2)(bc-ad)}$$

$$= -\frac{Ab^2 - a(bB - aC)}{2(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))^2(c+d\tan(e+fx))}$$

$$-\frac{5a^3bBd - 3a^4Cd + b^4(2Bc-3Ad) + ab^3(4Ac-4cC+Bd) - a^2b^2(2Bc+(7A-C)d)}{2(a^2+b^2)^2(bc-ad)^2f(a+b\tan(e+fx))(c+d\tan(e+fx))}$$

$$+\frac{\int \frac{(bc+ad)(3a(Ab^2-a(bB-aC))d-2b(Ab-aB-bC)(bc-ad))-(abc-a^2d-2b^2d)(3Ab^2d-2aA(bc-ad)-(bB-aC)(2bc+ad))+2(a^2B}{(a+b\tan(e+f}}{2(a^2+b}$$

$$=$$

$$-\frac{d(3a^3bBd(c^2+d^2)+ab^3(2Ac-2cC+Bd)(c^2+d^2)-a^4d(3c^2C-Bcd+(A+2C)d^2)-a^2b^2(}{(a^2+b^2)^2(bc-ad)^3(c^2+d^2)f}$$

$$-\frac{Ab^2 - a(bB - aC)}{2(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))^2(c+d\tan(e+fx))}$$

$$-\frac{5a^3bBd - 3a^4Cd + b^4(2Bc-3Ad) + ab^3(4Ac-4cC+Bd) - a^2b^2(2Bc+(7A-C)d)}{2(a^2+b^2)^2(bc-ad)^2f(a+b\tan(e+fx))(c+d\tan(e+fx))}$$

$$+\frac{\int \frac{-2(a^5d^3(Ac-cC+Bd)-3a^4Abd^2(c^2+d^2)+b^5(c^2+d^2)(Ac^2-c^2C+2Bcd-3Ad^2)+a^3b^2d(3Ac^3-3c^3C+5Acd^2-5cCd^2+2Bd^3)+a}{}}{}$$

$$=$$

$$-\frac{(a^3(c^2C-2Bcd-Cd^2-A(c^2-d^2))-3ab^2(c^2C-2Bcd-Cd^2-A(c^2-d^2))+3a^2b(2c(A-}{(a^2+b^2)^3(c^2+d^2)^2}$$

$$-\frac{d(3a^3bBd(c^2+d^2)+ab^3(2Ac-2cC+Bd)(c^2+d^2)-a^4d(3c^2C-Bcd+(A+2C)d^2)-a^2b^2(}{(a^2+b^2)^2(bc-ad)^3(c^2+d^2)f}$$

$$-\frac{Ab^2 - a(bB - aC)}{2(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))^2(c+d\tan(e+fx))}$$

$$-\frac{5a^3bBd - 3a^4Cd + b^4(2Bc-3Ad) + ab^3(4Ac-4cC+Bd) - a^2b^2(2Bc+(7A-C)d)}{2(a^2+b^2)^2(bc-ad)^2f(a+b\tan(e+fx))(c+d\tan(e+fx))}$$

$$-\frac{(d^2(b(3c^4C-4Bc^3d+c^2(5A+C)d^2-2Bcd^3+3Ad^4)-ad^2(2c(A-C)d-B(c^2-d^2)))) \int \frac{d-c}{c+d}}{(bc-ad)^4(c^2+d^2)^2}$$

$$-\frac{(b(6a^5bBd^2-3a^6Cd^2-a^4b^2d(4Bc+(10A-C)d)-b^6(c(cC-2Bd)-A(c^2-3d^2))+ab^5(2c(}{}$$

$$= -\frac{(a^3(c^2C - 2Bcd - Cd^2 - A(c^2 - d^2)) - 3ab^2(c^2C - 2Bcd - Cd^2 - A(c^2 - d^2)) + 3a^2b(2c(A - }{(a^2 + b^2)^3 (c^2 + d^2)^2}$$
$$- \frac{b(6a^5bBd^2 - 3a^6Cd^2 - a^4b^2d(4Bc + (10A - C)d) - b^6(c(cC - 2Bd) - A(c^2 - 3d^2)) + ab^5(2c(}{}$$
$$- \frac{d^2(b(3c^4C - 4Bc^3d + c^2(5A + C)d^2 - 2Bcd^3 + 3Ad^4) - ad^2(2c(A - C)d - B(c^2 - d^2))) \log(c}{(bc - ad)^4 (c^2 + d^2)^2 f}$$
$$- \frac{d(3a^3bBd(c^2 + d^2) + ab^3(2Ac - 2cC + Bd)(c^2 + d^2) - a^4d(3c^2C - Bcd + (A + 2C)d^2) - a^2b^2}{(a^2 + b^2)^2 (bc - ad)^3 (c^2 + d^2)}$$
$$- \frac{Ab^2 - a(bB - aC)}{2(a^2 + b^2)(bc - ad)f(a + b\tan(e + fx))^2(c + d\tan(e + fx))}$$
$$- \frac{5a^3bBd - 3a^4Cd + b^4(2Bc - 3Ad) + ab^3(4Ac - 4cC + Bd) - a^2b^2(2Bc + (7A - C)d)}{2(a^2 + b^2)^2 (bc - ad)^2 f(a + b\tan(e + fx))(c + d\tan(e + fx))}$$

Mathematica [B] (verified)

Leaf count is larger than twice the leaf count of optimal. 1758 vs. $2(841) = 1682$.

Time = 8.88 (sec) , antiderivative size = 1758, normalized size of antiderivative = 2.09

$$\int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{(a + b\tan(e + fx))^3(c + d\tan(e + fx))^2} dx$$
$$= -\frac{Ab^2 - a(bB - aC)}{2(a^2 + b^2)(bc - ad)f(a + b\tan(e + fx))^2(c + d\tan(e + fx))}$$
$$- \frac{-\frac{-a(-3a(Ab^2 - a(bB - aC))d + 2b(Ab - aB - bC)(bc - ad)) + b^2(3Ab^2d - 2aA(bc - ad) - (bB - aC)(2bc + ad))}{(a^2 + b^2)(bc - ad)f(a + b\tan(e + fx))(c + d\tan(e + fx))} - \frac{-\frac{(bc - ad)^3(-b^2(-3a^2Abc^2 + A}{}}{}}{}$$

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^3*(
c + d*Tan[e + f*x])^2),x]
```

```
[Out] -1/2*(A*b^2 - a*(b*B - a*C))/((a^2 + b^2)*(b*c - a*d)*f*(a + b*Tan[e + f*x]
)^2*(c + d*Tan[e + f*x])) - (-((-(a*(-3*a*(A*b^2 - a*(b*B - a*C))*d + 2*b*(
A*b - a*B - b*C)*(b*c - a*d))) + b^2*(3*A*b^2*d - 2*a*A*(b*c - a*d) - (b*B
- a*C)*(2*b*c + a*d)))/((a^2 + b^2)*(b*c - a*d)*f*(a + b*Tan[e + f*x])*(c +
 d*Tan[e + f*x]))) - (-((-(((b*c - a*d)^3*(-(b^2*(-3*a^2*A*b*c^2 + A*b^3*c^
2 + a^3*B*c^2 - 3*a*b^2*B*c^2 + 3*a^2*b*c^2*C - b^3*c^2*C - 2*a^3*A*c*d + 6
*a*A*b^2*c*d - 6*a^2*b*B*c*d + 2*b^3*B*c*d + 2*a^3*c*C*d - 6*a*b^2*c*C*d +
3*a^2*A*b*d^2 - A*b^3*d^2 - a^3*B*d^2 + 3*a*b^2*B*d^2 - 3*a^2*b*C*d^2 + b^3
*C*d^2)) + Sqrt[-b^2]*(a^3*A*b*c^2 - 3*a*A*b^3*c^2 + 3*a^2*b^2*B*c^2 - b^4*
B*c^2 - a^3*b*c^2*C + 3*a*b^3*c^2*C - 6*a^2*A*b^2*c*d + 2*A*b^4*c*d + 2*a^3
*b*B*c*d - 6*a*b^3*B*c*d + 6*a^2*b^2*c*C*d - 2*b^4*c*C*d - a^3*A*b*d^2 + 3*
```

```
a*A*b^3*d^2 - 3*a^2*b^2*B*d^2 + b^4*B*d^2 + a^3*b*C*d^2 - 3*a*b^3*C*d^2))*L
og[Sqrt[-b^2] - b*Tan[e + f*x]])/(b*(a^2 + b^2)*(c^2 + d^2))) - (2*b^2*(c^2
 + d^2)*(6*a^5*b*B*d^2 - 3*a^6*C*d^2 - a^4*b^2*d*(4*B*c + (10*A - C)*d) - b
^6*(c*(c*C - 2*B*d) - A*(c^2 - 3*d^2)) + a*b^5*(2*c*(A - C)*d - B*(3*c^2 -
d^2)) + 3*a^2*b^4*(c*(c*C + 2*B*d) - A*(c^2 + 3*d^2)) + a^3*b^3*(10*c*(A -
C)*d + B*(c^2 + 3*d^2)))*Log[a + b*Tan[e + f*x]])/((a^2 + b^2)*(b*c - a*d))
 + ((b*c - a*d)^3*(b^2*(-3*a^2*A*b*c^2 + A*b^3*c^2 + a^3*B*c^2 - 3*a*b^2*B*
c^2 + 3*a^2*b*c^2*C - b^3*c^2*C - 2*a^3*A*c*d + 6*a*A*b^2*c*d - 6*a^2*b*B*c
*d + 2*b^3*B*c*d + 2*a^3*c*C*d - 6*a*b^2*c*C*d + 3*a^2*A*b*d^2 - A*b^3*d^2
- a^3*B*d^2 + 3*a*b^2*B*d^2 - 3*a^2*b*C*d^2 + b^3*C*d^2) + Sqrt[-b^2]*(a^3*
A*b*c^2 - 3*a*A*b^3*c^2 + 3*a^2*b^2*B*c^2 - b^4*B*c^2 - a^3*b*c^2*C + 3*a*b
^3*c^2*C - 6*a^2*A*b^2*c*d + 2*A*b^4*c*d + 2*a^3*b*B*c*d - 6*a*b^3*B*c*d +
6*a^2*b^2*c*C*d - 2*b^4*c*C*d - a^3*A*b*d^2 + 3*a*A*b^3*d^2 - 3*a^2*b^2*B*d
^2 + b^4*B*d^2 + a^3*b*C*d^2 - 3*a*b^3*C*d^2))*Log[Sqrt[-b^2] + b*Tan[e + f
*x]])/(b*(a^2 + b^2)*(c^2 + d^2)) - (2*b*(a^2 + b^2)^2*d^2*(b*(3*c^4*C - 4*
B*c^3*d + c^2*(5*A + C)*d^2 - 2*B*c*d^3 + 3*A*d^4) - a*d^2*(2*c*(A - C)*d -
 B*(c^2 - d^2)))*Log[c + d*Tan[e + f*x]])/((b*c - a*d)*(c^2 + d^2)))/(b*(-(
b*c) + a*d)*(c^2 + d^2)*f)) - (d^2*((-(b*c) - a*d)*(-3*a*(A*b^2 - a*(b*B -
a*C))*d + 2*b*(A*b - a*B - b*C)*(b*c - a*d)) + (2*b^2*d - a*(b*c - a*d))*(3
*A*b^2*d - 2*a*A*(b*c - a*d) - (b*B - a*C)*(2*b*c + a*d))) - c*(d*(b*c - a*
d)*(-3*b*(A*b^2 - a*(b*B - a*C))*d - 2*a*(A*b - a*B - b*C)*(b*c - a*d) + b*
(3*A*b^2*d - 2*a*A*(b*c - a*d) - (b*B - a*C)*(2*b*c + a*d))) - 2*c*d*(-(a*(
-3*a*(A*b^2 - a*(b*B - a*C))*d + 2*b*(A*b - a*B - b*C)*(b*c - a*d))) + b^2*
(3*A*b^2*d - 2*a*A*(b*c - a*d) - (b*B - a*C)*(2*b*c + a*d)))))/((-(b*c) + a
*d)*(c^2 + d^2)*f*(c + d*Tan[e + f*x])))/((a^2 + b^2)*(b*c - a*d)))/(2*(a^2
 + b^2)*(b*c - a*d))
```

Maple [A] (verified)

Time = 6.32 (sec) , antiderivative size = 951, normalized size of antiderivative = 1.13

method	result
derivativedivides	$\frac{-\frac{b\left(4A\,a^{2}b^{2}d-2Aa\,b^{3}c+2A\,b^{4}d-3a^{3}bBd+B\,a^{2}b^{2}c-Ba\,b^{3}d-B\,b^{4}c+2a^{4}Cd+2Ca\,b^{3}c\right)}{(ad-bc)^{3}\left(a^{2}+b^{2}\right)^{2}(a+b\tan(fx+e))}+\frac{b\left(10A\,a^{4}b^{2}d^{2}-10A\,a^{3}b^{3}cd+3A\,a^{2}b^{4}c^{2}+9A\,a\right.}{}}{}$
default	$\frac{-\frac{b\left(4A\,a^{2}b^{2}d-2Aa\,b^{3}c+2A\,b^{4}d-3a^{3}bBd+B\,a^{2}b^{2}c-Ba\,b^{3}d-B\,b^{4}c+2a^{4}Cd+2Ca\,b^{3}c\right)}{(ad-bc)^{3}\left(a^{2}+b^{2}\right)^{2}(a+b\tan(fx+e))}+\frac{b\left(10A\,a^{4}b^{2}d^{2}-10A\,a^{3}b^{3}cd+3A\,a^{2}b^{4}c^{2}+9A\,a\right.}{}}{}$
norman	Expression too large to display
risch	Expression too large to display
parallelrisch	Expression too large to display

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^3/(c+d*tan(f*x+e))^2,x
,method=_RETURNVERBOSE)
```

```
[Out] 1/f*(-b*(4*A*a^2*b^2*d-2*A*a*b^3*c+2*A*b^4*d-3*B*a^3*b*d+B*a^2*b^2*c-B*a*b^
3*d-B*b^4*c+2*C*a^4*d+2*C*a*b^3*c)/(a*d-b*c)^3/(a^2+b^2)^2/(a+b*tan(f*x+e))
+b*(10*A*a^4*b^2*d^2-10*A*a^3*b^3*c*d+3*A*a^2*b^4*c^2+9*A*a^2*b^4*d^2-2*A*a
*b^5*c*d-A*b^6*c^2+3*A*b^6*d^2-6*B*a^5*b*d^2+4*B*a^4*b^2*c*d-B*a^3*b^3*c^2-
3*B*a^3*b^3*d^2-6*B*a^2*b^4*c*d+3*B*a*b^5*c^2-B*a*b^5*d^2-2*B*b^6*c*d+3*C*a
^6*d^2-C*a^4*b^2*d^2+10*C*a^3*b^3*c*d-3*C*a^2*b^4*c^2+2*C*a*b^5*c*d+C*b^6*c
^2)/(a*d-b*c)^4/(a^2+b^2)^3*ln(a+b*tan(f*x+e))-1/2*(A*b^2-B*a*b+C*a^2)*b/(a
*d-b*c)^2/(a^2+b^2)/(a+b*tan(f*x+e))^2+1/(a^2+b^2)^3/(c^2+d^2)^2*(1/2*(-2*A
*a^3*c*d-3*A*a^2*b*c^2+3*A*a^2*b*d^2+6*A*a*b^2*c*d+A*b^3*c^2-A*b^3*d^2+B*a^
3*c^2-B*a^3*d^2-6*B*a^2*b*c*d-3*B*a*b^2*c^2+3*B*a*b^2*d^2+2*B*b^3*c*d+2*C*a
^3*c*d+3*C*a^2*b*c^2-3*C*a^2*b*d^2-6*C*a*b^2*c*d-C*b^3*c^2+C*b^3*d^2)*ln(1+
tan(f*x+e)^2)+(A*a^3*c^2-A*a^3*d^2-6*A*a^2*b*c*d-3*A*a*b^2*c^2+3*A*a*b^2*d^
2+2*A*b^3*c*d+2*B*a^3*c*d+3*B*a^2*b*c^2-3*B*a^2*b*d^2-6*B*a*b^2*c*d-B*b^3*c
^2+B*b^3*d^2-C*a^3*c^2+C*a^3*d^2+6*C*a^2*b*c*d+3*C*a*b^2*c^2-3*C*a*b^2*d^2-
2*C*b^3*c*d)*arctan(tan(f*x+e)))+d^2*(2*A*a*c*d^3-5*A*b*c^2*d^2-3*A*b*d^4-B
*a*c^2*d^2+B*a*d^4+4*B*b*c^3*d+2*B*b*c*d^3-2*C*a*c*d^3-3*C*b*c^4-C*b*c^2*d^
2)/(a*d-b*c)^4/(c^2+d^2)^2*ln(c+d*tan(f*x+e))-(A*d^2-B*c*d+C*c^2)*d^2/(a*d-
b*c)^3/(c^2+d^2)/(c+d*tan(f*x+e)))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 9594 vs. $2(835) = 1670$.

Time = 10.63 (sec) , antiderivative size = 9594, normalized size of antiderivative = 11.41

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^3(c+d\tan(e+fx))^2}\,dx = \text{Too large to display}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^3/(c+d*tan(f*x+e
))^2,x, algorithm="fricas")
```

```
[Out] Too large to include
```

Sympy [F(-1)]

Timed out.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^3(c+d\tan(e+fx))^2}\,dx = \text{Timed out}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e))**3/(c+d*tan(f*x
+e))**2,x)
```

```
[Out] Timed out
```

Maxima [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 2519 vs. $2(835) = 1670$.

Time = 0.48 (sec) , antiderivative size = 2519, normalized size of antiderivative = 3.00

$$\int \frac{A + B\tan(e+fx) + C\tan^2(e+fx)}{(a+b\tan(e+fx))^3(c+d\tan(e+fx))^2}\,dx = \text{Too large to display}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^3/(c+d*tan(f*x+e
))^2,x, algorithm="maxima")
```

```
[Out] 1/2*(2*(((A - C)*a^3 + 3*B*a^2*b - 3*(A - C)*a*b^2 - B*b^3)*c^2 + 2*(B*a^3
- 3*(A - C)*a^2*b - 3*B*a*b^2 + (A - C)*b^3)*c*d - ((A - C)*a^3 + 3*B*a^2*b
 - 3*(A - C)*a*b^2 - B*b^3)*d^2)*(f*x + e)/((a^6 + 3*a^4*b^2 + 3*a^2*b^4 +
b^6)*c^4 + 2*(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6)*c^2*d^2 + (a^6 + 3*a^4*b^2
 + 3*a^2*b^4 + b^6)*d^4) - 2*((B*a^3*b^4 - 3*(A - C)*a^2*b^5 - 3*B*a*b^6 +
(A - C)*b^7)*c^2 - 2*(2*B*a^4*b^3 - 5*(A - C)*a^3*b^4 - 3*B*a^2*b^5 - (A -
C)*a*b^6 - B*b^7)*c*d - (3*C*a^6*b - 6*B*a^5*b^2 + (10*A - C)*a^4*b^3 - 3*B
*a^3*b^4 + 9*A*a^2*b^5 - B*a*b^6 + 3*A*b^7)*d^2)*log(b*tan(f*x + e) + a)/((
a^6*b^4 + 3*a^4*b^6 + 3*a^2*b^8 + b^10)*c^4 - 4*(a^7*b^3 + 3*a^5*b^5 + 3*a^
3*b^7 + a*b^9)*c^3*d + 6*(a^8*b^2 + 3*a^6*b^4 + 3*a^4*b^6 + a^2*b^8)*c^2*d^
2 - 4*(a^9*b + 3*a^7*b^3 + 3*a^5*b^5 + a^3*b^7)*c*d^3 + (a^10 + 3*a^8*b^2 +
 3*a^6*b^4 + a^4*b^6)*d^4) - 2*(3*C*b*c^4*d^2 - 4*B*b*c^3*d^3 + (B*a + (5*A
 + C)*b)*c^2*d^4 - 2*((A - C)*a + B*b)*c*d^5 - (B*a - 3*A*b)*d^6)*log(d*tan
(f*x + e) + c)/(b^4*c^8 - 4*a*b^3*c^7*d - 4*a^3*b*c*d^7 + a^4*d^8 + 2*(3*a^
2*b^2 + b^4)*c^6*d^2 - 4*(a^3*b + 2*a*b^3)*c^5*d^3 + (a^4 + 12*a^2*b^2 + b^
4)*c^4*d^4 - 4*(2*a^3*b + a*b^3)*c^3*d^5 + 2*(a^4 + 3*a^2*b^2)*c^2*d^6) + (
(B*a^3 - 3*(A - C)*a^2*b - 3*B*a*b^2 + (A - C)*b^3)*c^2 - 2*((A - C)*a^3 +
3*B*a^2*b - 3*(A - C)*a*b^2 - B*b^3)*c*d - (B*a^3 - 3*(A - C)*a^2*b - 3*B*a
*b^2 + (A - C)*b^3)*d^2)*log(tan(f*x + e)^2 + 1)/((a^6 + 3*a^4*b^2 + 3*a^2*
b^4 + b^6)*c^4 + 2*(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6)*c^2*d^2 + (a^6 + 3*a
^4*b^2 + 3*a^2*b^4 + b^6)*d^4) - ((C*a^4*b^2 - 3*B*a^3*b^3 + (5*A - 3*C)*a^
2*b^4 + B*a*b^5 + A*b^6)*c^4 - (5*C*a^5*b - 7*B*a^4*b^2 + (9*A + C)*a^3*b^3
 - 3*B*a^2*b^4 + 5*A*a*b^5)*c^3*d - (2*C*a^6 + 3*C*a^4*b^2 + 3*B*a^3*b^3 -
5*(A - C)*a^2*b^4 - B*a*b^5 - A*b^6)*c^2*d^2 + (2*B*a^6 - 5*C*a^5*b + 11*B*
a^4*b^2 - (9*A + C)*a^3*b^3 + 5*B*a^2*b^4 - 5*A*a*b^5)*c*d^3 - 2*(A*a^6 + 2
*A*a^4*b^2 + A*a^2*b^4)*d^4 - 2*((B*a^2*b^4 - 2*(A - C)*a*b^5 - B*b^6)*c^3*
d + (3*C*a^4*b^2 - 3*B*a^3*b^3 + 2*(2*A + C)*a^2*b^4 - B*a*b^5 + (2*A + C)*
b^6)*c^2*d^2 - (B*a^4*b^2 + B*a^2*b^4 + 2*(A - C)*a*b^5 + 2*B*b^6)*c*d^3 +
((A + 2*C)*a^4*b^2 - 3*B*a^3*b^3 + 6*A*a^2*b^4 - B*a*b^5 + 3*A*b^6)*d^4)*ta
n(f*x + e)^2 - (2*(B*a^2*b^4 - 2*(A - C)*a*b^5 - B*b^6)*c^4 + 3*(C*a^4*b^2
- B*a^3*b^3 + (A + C)*a^2*b^4 - B*a*b^5 + A*b^6)*c^3*d + (9*C*a^5*b - 7*B*a
^4*b^2 + 9*(A + C)*a^3*b^3 - B*a^2*b^4 + (A + 8*C)*a*b^5 - 2*B*b^6)*c^2*d^2
 - (4*B*a^5*b - 3*C*a^4*b^2 + 11*B*a^3*b^3 - 3*(A + C)*a^2*b^4 + 7*B*a*b^5
- 3*A*b^6)*c*d^3 + ((4*A + 5*C)*a^5*b - 7*B*a^4*b^2 + (17*A + C)*a^3*b^3 -
```

```
3*B*a^2*b^4 + 9*A*a*b^5)*d^4)*tan(f*x + e))/((a^6*b^3 + 2*a^4*b^5 + a^2*b^7
)*c^6 - 3*(a^7*b^2 + 2*a^5*b^4 + a^3*b^6)*c^5*d + (3*a^8*b + 7*a^6*b^3 + 5*
a^4*b^5 + a^2*b^7)*c^4*d^2 - (a^9 + 5*a^7*b^2 + 7*a^5*b^4 + 3*a^3*b^6)*c^3*
d^3 + 3*(a^8*b + 2*a^6*b^3 + a^4*b^5)*c^2*d^4 - (a^9 + 2*a^7*b^2 + a^5*b^4)
*c*d^5 + ((a^4*b^5 + 2*a^2*b^7 + b^9)*c^5*d - 3*(a^5*b^4 + 2*a^3*b^6 + a*b^
8)*c^4*d^2 + (3*a^6*b^3 + 7*a^4*b^5 + 5*a^2*b^7 + b^9)*c^3*d^3 - (a^7*b^2 +
 5*a^5*b^4 + 7*a^3*b^6 + 3*a*b^8)*c^2*d^4 + 3*(a^6*b^3 + 2*a^4*b^5 + a^2*b^
7)*c*d^5 - (a^7*b^2 + 2*a^5*b^4 + a^3*b^6)*d^6)*tan(f*x + e)^3 + ((a^4*b^5
+ 2*a^2*b^7 + b^9)*c^6 - (a^5*b^4 + 2*a^3*b^6 + a*b^8)*c^5*d - (3*a^6*b^3 +
 5*a^4*b^5 + a^2*b^7 - b^9)*c^4*d^2 + (5*a^7*b^2 + 9*a^5*b^4 + 3*a^3*b^6 -
a*b^8)*c^3*d^3 - (2*a^8*b + 7*a^6*b^3 + 8*a^4*b^5 + 3*a^2*b^7)*c^2*d^4 + 5*
(a^7*b^2 + 2*a^5*b^4 + a^3*b^6)*c*d^5 - 2*(a^8*b + 2*a^6*b^3 + a^4*b^5)*d^6
)*tan(f*x + e)^2 + (2*(a^5*b^4 + 2*a^3*b^6 + a*b^8)*c^6 - 5*(a^6*b^3 + 2*a^
4*b^5 + a^2*b^7)*c^5*d + (3*a^7*b^2 + 8*a^5*b^4 + 7*a^3*b^6 + 2*a*b^8)*c^4*
d^2 + (a^8*b - 3*a^6*b^3 - 9*a^4*b^5 - 5*a^2*b^7)*c^3*d^3 - (a^9 - a^7*b^2
- 5*a^5*b^4 - 3*a^3*b^6)*c^2*d^4 + (a^8*b + 2*a^6*b^3 + a^4*b^5)*c*d^5 - (a
^9 + 2*a^7*b^2 + a^5*b^4)*d^6)*tan(f*x + e)))/f
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 3115 vs. 2(835) = 1670.

Time = 1.10 (sec) , antiderivative size = 3115, normalized size of antiderivative = 3.70

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^3(c+d\tan(e+fx))^2}\,dx = \text{Too large to display}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^3/(c+d*tan(f*x+e
))^2,x, algorithm="giac")
```

```
[Out] 1/2*(2*(A*a^3*c^2 - C*a^3*c^2 + 3*B*a^2*b*c^2 - 3*A*a*b^2*c^2 + 3*C*a*b^2*c
^2 - B*b^3*c^2 + 2*B*a^3*c*d - 6*A*a^2*b*c*d + 6*C*a^2*b*c*d - 6*B*a*b^2*c*
d + 2*A*b^3*c*d - 2*C*b^3*c*d - A*a^3*d^2 + C*a^3*d^2 - 3*B*a^2*b*d^2 + 3*A
*a*b^2*d^2 - 3*C*a*b^2*d^2 + B*b^3*d^2)*(f*x + e)/(a^6*c^4 + 3*a^4*b^2*c^4
+ 3*a^2*b^4*c^4 + b^6*c^4 + 2*a^6*c^2*d^2 + 6*a^4*b^2*c^2*d^2 + 6*a^2*b^4*c
^2*d^2 + 2*b^6*c^2*d^2 + a^6*d^4 + 3*a^4*b^2*d^4 + 3*a^2*b^4*d^4 + b^6*d^4)
 + (B*a^3*c^2 - 3*A*a^2*b*c^2 + 3*C*a^2*b*c^2 - 3*B*a*b^2*c^2 + A*b^3*c^2 -
 C*b^3*c^2 - 2*A*a^3*c*d + 2*C*a^3*c*d - 6*B*a^2*b*c*d + 6*A*a*b^2*c*d - 6*
C*a*b^2*c*d + 2*B*b^3*c*d - B*a^3*d^2 + 3*A*a^2*b*d^2 - 3*C*a^2*b*d^2 + 3*B
*a*b^2*d^2 - A*b^3*d^2 + C*b^3*d^2)*log(tan(f*x + e)^2 + 1)/(a^6*c^4 + 3*a^
4*b^2*c^4 + 3*a^2*b^4*c^4 + b^6*c^4 + 2*a^6*c^2*d^2 + 6*a^4*b^2*c^2*d^2 + 6
*a^2*b^4*c^2*d^2 + 2*b^6*c^2*d^2 + a^6*d^4 + 3*a^4*b^2*d^4 + 3*a^2*b^4*d^4
+ b^6*d^4) - 2*(B*a^3*b^5*c^2 - 3*A*a^2*b^6*c^2 + 3*C*a^2*b^6*c^2 - 3*B*a*b
^7*c^2 + A*b^8*c^2 - C*b^8*c^2 - 4*B*a^4*b^4*c*d + 10*A*a^3*b^5*c*d - 10*C*
a^3*b^5*c*d + 6*B*a^2*b^6*c*d + 2*A*a*b^7*c*d - 2*C*a*b^7*c*d + 2*B*b^8*c*d
 - 3*C*a^6*b^2*d^2 + 6*B*a^5*b^3*d^2 - 10*A*a^4*b^4*d^2 + C*a^4*b^4*d^2 + 3
```

*B*a^3*b^5*d^2 - 9*A*a^2*b^6*d^2 + B*a*b^7*d^2 - 3*A*b^8*d^2)*log(abs(b*tan
(f*x + e) + a))/(a^6*b^5*c^4 + 3*a^4*b^7*c^4 + 3*a^2*b^9*c^4 + b^11*c^4 - 4
*a^7*b^4*c^3*d - 12*a^5*b^6*c^3*d - 12*a^3*b^8*c^3*d - 4*a*b^10*c^3*d + 6*a
^8*b^3*c^2*d^2 + 18*a^6*b^5*c^2*d^2 + 18*a^4*b^7*c^2*d^2 + 6*a^2*b^9*c^2*d^
2 - 4*a^9*b^2*c*d^3 - 12*a^7*b^4*c*d^3 - 12*a^5*b^6*c*d^3 - 4*a^3*b^8*c*d^3
 + a^10*b*d^4 + 3*a^8*b^3*d^4 + 3*a^6*b^5*d^4 + a^4*b^7*d^4) - 2*(3*C*b*c^4
*d^3 - 4*B*b*c^3*d^4 + B*a*c^2*d^5 + 5*A*b*c^2*d^5 + C*b*c^2*d^5 - 2*A*a*c*
d^6 + 2*C*a*c*d^6 - 2*B*b*c*d^6 - B*a*d^7 + 3*A*b*d^7)*log(abs(d*tan(f*x +
e) + c))/(b^4*c^8*d - 4*a*b^3*c^7*d^2 + 6*a^2*b^2*c^6*d^3 + 2*b^4*c^6*d^3 -
 4*a^3*b*c^5*d^4 - 8*a*b^3*c^5*d^4 + a^4*c^4*d^5 + 12*a^2*b^2*c^4*d^5 + b^4
*c^4*d^5 - 8*a^3*b*c^3*d^6 - 4*a*b^3*c^3*d^6 + 2*a^4*c^2*d^7 + 6*a^2*b^2*c^
2*d^7 - 4*a^3*b*c*d^8 + a^4*d^9) + 2*(3*C*b*c^4*d^3*tan(f*x + e) - 4*B*b*c^
3*d^4*tan(f*x + e) + B*a*c^2*d^5*tan(f*x + e) + 5*A*b*c^2*d^5*tan(f*x + e)
+ C*b*c^2*d^5*tan(f*x + e) - 2*A*a*c*d^6*tan(f*x + e) + 2*C*a*c*d^6*tan(f*x
 + e) - 2*B*b*c*d^6*tan(f*x + e) - B*a*d^7*tan(f*x + e) + 3*A*b*d^7*tan(f*x
 + e) + 4*C*b*c^5*d^2 - C*a*c^4*d^3 - 5*B*b*c^4*d^3 + 2*B*a*c^3*d^4 + 6*A*b
*c^3*d^4 + 2*C*b*c^3*d^4 - 3*A*a*c^2*d^5 + C*a*c^2*d^5 - 3*B*b*c^2*d^5 + 4*
A*b*c*d^6 - A*a*d^7)/((b^4*c^8 - 4*a*b^3*c^7*d + 6*a^2*b^2*c^6*d^2 + 2*b^4*
c^6*d^2 - 4*a^3*b*c^5*d^3 - 8*a*b^3*c^5*d^3 + a^4*c^4*d^4 + 12*a^2*b^2*c^4*
d^4 + b^4*c^4*d^4 - 8*a^3*b*c^3*d^5 - 4*a*b^3*c^3*d^5 + 2*a^4*c^2*d^6 + 6*a
^2*b^2*c^2*d^6 - 4*a^3*b*c*d^7 + a^4*d^8)*(d*tan(f*x + e) + c)) + (3*B*a^3*
b^6*c^2*tan(f*x + e)^2 - 9*A*a^2*b^7*c^2*tan(f*x + e)^2 + 9*C*a^2*b^7*c^2*t
an(f*x + e)^2 - 9*B*a*b^8*c^2*tan(f*x + e)^2 + 3*A*b^9*c^2*tan(f*x + e)^2 -
 3*C*b^9*c^2*tan(f*x + e)^2 - 12*B*a^4*b^5*c*d*tan(f*x + e)^2 + 30*A*a^3*b^
6*c*d*tan(f*x + e)^2 - 30*C*a^3*b^6*c*d*tan(f*x + e)^2 + 18*B*a^2*b^7*c*d*t
an(f*x + e)^2 + 6*A*a*b^8*c*d*tan(f*x + e)^2 - 6*C*a*b^8*c*d*tan(f*x + e)^2
 + 6*B*b^9*c*d*tan(f*x + e)^2 - 9*C*a^6*b^3*d^2*tan(f*x + e)^2 + 18*B*a^5*b
^4*d^2*tan(f*x + e)^2 - 30*A*a^4*b^5*d^2*tan(f*x + e)^2 + 3*C*a^4*b^5*d^2*t
an(f*x + e)^2 + 9*B*a^3*b^6*d^2*tan(f*x + e)^2 - 27*A*a^2*b^7*d^2*tan(f*x +
 e)^2 + 3*B*a*b^8*d^2*tan(f*x + e)^2 - 9*A*b^9*d^2*tan(f*x + e)^2 + 8*B*a^4
*b^5*c^2*tan(f*x + e) - 22*A*a^3*b^6*c^2*tan(f*x + e) + 22*C*a^3*b^6*c^2*ta
n(f*x + e) - 18*B*a^2*b^7*c^2*tan(f*x + e) + 2*A*a*b^8*c^2*tan(f*x + e) - 2
*C*a*b^8*c^2*tan(f*x + e) - 2*B*b^9*c^2*tan(f*x + e) + 4*C*a^6*b^3*c*d*tan(
f*x + e) - 32*B*a^5*b^4*c*d*tan(f*x + e) + 72*A*a^4*b^5*c*d*tan(f*x + e) -
60*C*a^4*b^5*c*d*tan(f*x + e) + 28*B*a^3*b^6*c*d*tan(f*x + e) + 28*A*a^2*b^
7*c*d*tan(f*x + e) - 16*C*a^2*b^7*c*d*tan(f*x + e) + 12*B*a*b^8*c*d*tan(f*x
 + e) + 4*A*b^9*c*d*tan(f*x + e) - 22*C*a^7*b^2*d^2*tan(f*x + e) + 42*B*a^6
*b^3*d^2*tan(f*x + e) - 68*A*a^5*b^4*d^2*tan(f*x + e) + 2*C*a^5*b^4*d^2*tan
(f*x + e) + 26*B*a^4*b^5*d^2*tan(f*x + e) - 66*A*a^3*b^6*d^2*tan(f*x + e) +
 8*B*a^2*b^7*d^2*tan(f*x + e) - 22*A*a*b^8*d^2*tan(f*x + e) - C*a^6*b^3*c^2
 + 6*B*a^5*b^4*c^2 - 14*A*a^4*b^5*c^2 + 11*C*a^4*b^5*c^2 - 7*B*a^3*b^6*c^2
- 3*A*a^2*b^7*c^2 - B*a*b^8*c^2 - A*b^9*c^2 + 6*C*a^7*b^2*c*d - 22*B*a^6*b^
3*c*d + 44*A*a^5*b^4*c*d - 26*C*a^5*b^4*c*d + 6*B*a^4*b^5*c*d + 26*A*a^3*b^
6*c*d - 8*C*a^3*b^6*c*d + 4*B*a^2*b^7*c*d + 6*A*a*b^8*c*d - 14*C*a^8*b*d^2
+ 25*B*a^7*b^2*d^2 - 39*A*a^6*b^3*d^2 - 3*C*a^6*b^3*d^2 + 19*B*a^5*b^4*d^2

```
- 41*A*a^4*b^5*d^2 - C*a^4*b^5*d^2 + 6*B*a^3*b^6*d^2 - 14*A*a^2*b^7*d^2)/((
a^6*b^4*c^4 + 3*a^4*b^6*c^4 + 3*a^2*b^8*c^4 + b^10*c^4 - 4*a^7*b^3*c^3*d -
12*a^5*b^5*c^3*d - 12*a^3*b^7*c^3*d - 4*a*b^9*c^3*d + 6*a^8*b^2*c^2*d^2 + 1
8*a^6*b^4*c^2*d^2 + 18*a^4*b^6*c^2*d^2 + 6*a^2*b^8*c^2*d^2 - 4*a^9*b*c*d^3
- 12*a^7*b^3*c*d^3 - 12*a^5*b^5*c*d^3 - 4*a^3*b^7*c*d^3 + a^10*d^4 + 3*a^8*
b^2*d^4 + 3*a^6*b^4*d^4 + a^4*b^6*d^4)*(b*tan(f*x + e) + a)^2))/f
```

Mupad [B] (verification not implemented)

Time = 44.53 (sec) , antiderivative size = 128667, normalized size of antiderivative = 152.99

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^3(c+d\tan(e+fx))^2}\,dx = \text{Too large to display}$$

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))^3*(c + d*
tan(e + f*x))^2),x)
```

```
[Out] (symsum(log((24*A^3*a^3*b^7*d^9 + 27*A^3*a^5*b^5*d^9 + B^3*a^2*b^8*d^9 + 4*
B^3*a^4*b^6*d^9 + 7*B^3*a^6*b^4*d^9 + 3*A^3*b^10*c^3*d^6 - A^3*b^10*c^5*d^4
 + 4*B^3*b^10*c^2*d^7 + 6*B^3*b^10*c^4*d^5 + C^3*b^10*c^5*d^4 + 9*A^2*B*b^1
0*d^9 + 9*A^3*a*b^9*d^9 + 16*A^3*a^2*b^8*c^3*d^6 + 3*A^3*a^2*b^8*c^5*d^4 +
26*A^3*a^3*b^7*c^2*d^7 - 6*A^3*a^3*b^7*c^4*d^5 - 11*A^3*a^4*b^6*c^3*d^6 + 3
1*A^3*a^5*b^5*c^2*d^7 + 5*B^3*a^2*b^8*c^2*d^7 + 6*B^3*a^2*b^8*c^4*d^5 + 28*
B^3*a^3*b^7*c^3*d^6 + 7*B^3*a^3*b^7*c^5*d^4 - 14*B^3*a^4*b^6*c^2*d^7 - 20*B
^3*a^4*b^6*c^4*d^5 + 19*B^3*a^5*b^5*c^3*d^6 + 9*B^3*a^6*b^4*c^2*d^7 - 7*C^3
*a^2*b^8*c^3*d^6 - 3*C^3*a^2*b^8*c^5*d^4 + C^3*a^3*b^7*c^2*d^7 + 15*C^3*a^3
*b^7*c^4*d^5 + 6*C^3*a^3*b^7*c^6*d^3 - 28*C^3*a^4*b^6*c^3*d^6 - 24*C^3*a^4*
b^6*c^5*d^4 - 4*C^3*a^5*b^5*c^2*d^7 + 3*C^3*a^6*b^4*c^3*d^6 - 9*C^3*a^7*b^3
*c^2*d^7 - 9*C^3*a^7*b^3*c^4*d^5 - 6*A*B^2*a*b^9*d^9 - 9*A^2*C*a*b^9*d^9 -
12*A*B^2*b^10*c*d^8 + 4*B^3*a*b^9*c*d^8 - 20*A*B^2*a^3*b^7*d^9 - 28*A*B^2*a
^5*b^5*d^9 + 6*A*B^2*a^7*b^3*d^9 + 21*A^2*B*a^2*b^8*d^9 + 13*A^2*B*a^4*b^6*
d^9 - 27*A^2*B*a^6*b^4*d^9 - 3*A*C^2*a^3*b^7*d^9 - 9*A*C^2*a^7*b^3*d^9 - 21
*A^2*C*a^3*b^7*d^9 - 27*A^2*C*a^5*b^5*d^9 + 9*A^2*C*a^7*b^3*d^9 - 17*A*B^2*
b^10*c^3*d^6 + 3*A*B^2*b^10*c^5*d^4 + B*C^2*a^4*b^6*d^9 + 3*B*C^2*a^8*b^2*d
^9 + 12*A^2*B*b^10*c^2*d^7 - 7*A^2*B*b^10*c^4*d^5 - B^2*C*a^3*b^7*d^9 - 2*B
^2*C*a^5*b^5*d^9 - 9*B^2*C*a^7*b^3*d^9 + 3*A*C^2*b^10*c^3*d^6 - 3*A*C^2*b^1
0*c^5*d^4 - 6*A^2*C*b^10*c^3*d^6 + 3*A^2*C*b^10*c^5*d^4 - B*C^2*b^10*c^4*d^
5 + 3*B*C^2*b^10*c^6*d^3 - 4*B^2*C*b^10*c^3*d^6 - 9*B^2*C*b^10*c^5*d^4 + 3*
A^3*a*b^9*c^2*d^7 - 10*A^3*a*b^9*c^4*d^5 - 3*A^3*a^2*b^8*c*d^8 - 31*A^3*a^4
*b^6*c*d^8 - 8*A^3*a^6*b^4*c*d^8 + B^3*a*b^9*c^3*d^6 - 5*B^3*a*b^9*c^5*d^4
+ 11*B^3*a^3*b^7*c*d^8 + 5*B^3*a^5*b^5*c*d^8 - 6*B^3*a^7*b^3*c*d^8 - 2*C^3*
a*b^9*c^4*d^5 - 6*C^3*a*b^9*c^6*d^3 - 2*C^3*a^4*b^6*c*d^8 - C^3*a^6*b^4*c*d
^8 - 3*C^3*a^8*b^2*c*d^8 - 60*A*B^2*a^2*b^8*c^3*d^6 - 21*A*B^2*a^2*b^8*c^5*
d^4 - 4*A*B^2*a^3*b^7*c^2*d^7 + 44*A*B^2*a^3*b^7*c^4*d^5 + 25*A*B^2*a^4*b^6
*c^3*d^6 + 4*A*B^2*a^4*b^6*c^5*d^4 - 77*A*B^2*a^5*b^5*c^2*d^7 - 17*A*B^2*a^
```

5*b^5*c^4*d^5 + 28*A*B^2*a^6*b^4*c^3*d^6 - 6*A*B^2*a^7*b^3*c^2*d^7 + 71*A^2
*B*a^2*b^8*c^2*d^7 + 16*A^2*B*a^2*b^8*c^4*d^5 - 116*A^2*B*a^3*b^7*c^3*d^6 -
9*A^2*B*a^3*b^7*c^5*d^4 + 86*A^2*B*a^4*b^6*c^2*d^7 + 35*A^2*B*a^4*b^6*c^4*
d^5 - 37*A^2*B*a^5*b^5*c^3*d^6 - 13*A^2*B*a^6*b^4*c^2*d^7 + 30*A*C^2*a^2*b^
8*c^3*d^6 + 9*A*C^2*a^2*b^8*c^5*d^4 - 30*A*C^2*a^3*b^7*c^2*d^7 - 63*A*C^2*a
^3*b^7*c^4*d^5 - 12*A*C^2*a^3*b^7*c^6*d^3 + 45*A*C^2*a^4*b^6*c^3*d^6 + 48*A
*C^2*a^4*b^6*c^5*d^4 - 15*A*C^2*a^5*b^5*c^2*d^7 - 27*A*C^2*a^5*b^5*c^4*d^5
- 6*A*C^2*a^6*b^4*c^3*d^6 + 9*A*C^2*a^7*b^3*c^4*d^5 - 39*A^2*C*a^2*b^8*c^3*
d^6 - 9*A^2*C*a^2*b^8*c^5*d^4 + 3*A^2*C*a^3*b^7*c^2*d^7 + 54*A^2*C*a^3*b^7*
c^4*d^5 + 6*A^2*C*a^3*b^7*c^6*d^3 - 6*A^2*C*a^4*b^6*c^3*d^6 - 24*A^2*C*a^4*
b^6*c^5*d^4 - 12*A^2*C*a^5*b^5*c^2*d^7 + 27*A^2*C*a^5*b^5*c^4*d^5 + 3*A^2*C
*a^6*b^4*c^3*d^6 + 9*A^2*C*a^7*b^3*c^2*d^7 + 11*B*C^2*a^2*b^8*c^2*d^7 - 17*
B*C^2*a^2*b^8*c^4*d^5 - 18*B*C^2*a^2*b^8*c^6*d^3 + 16*B*C^2*a^3*b^7*c^3*d^6
+ 39*B*C^2*a^3*b^7*c^5*d^4 + 47*B*C^2*a^4*b^6*c^2*d^7 + 47*B*C^2*a^4*b^6*c
^4*d^5 + 3*B*C^2*a^4*b^6*c^6*d^3 - 25*B*C^2*a^5*b^5*c^3*d^6 - 12*B*C^2*a^5*
b^5*c^5*d^4 + 17*B*C^2*a^6*b^4*c^2*d^7 + 27*B*C^2*a^6*b^4*c^4*d^5 + 12*B*C^
2*a^7*b^3*c^3*d^6 - 3*B*C^2*a^8*b^2*c^2*d^7 + 9*B^2*C*a^2*b^8*c^3*d^6 + 9*B
^2*C*a^2*b^8*c^5*d^4 - 35*B^2*C*a^3*b^7*c^2*d^7 - 68*B^2*C*a^3*b^7*c^4*d^5
- 6*B^2*C*a^3*b^7*c^6*d^3 - 16*B^2*C*a^4*b^6*c^3*d^6 + 14*B^2*C*a^4*b^6*c^5
*d^4 + 26*B^2*C*a^5*b^5*c^2*d^7 - 4*B^2*C*a^5*b^5*c^4*d^5 - 37*B^2*C*a^6*b^
4*c^3*d^6 + 3*B^2*C*a^7*b^3*c^2*d^7 + 6*A*B*C*a^2*b^8*d^9 + 13*A*B*C*a^4*b^
6*d^9 + 36*A*B*C*a^6*b^4*d^9 - 3*A*B*C*a^8*b^2*d^9 + 6*A*B*C*b^10*c^2*d^7 +
17*A*B*C*b^10*c^4*d^5 - 3*A*B*C*b^10*c^6*d^3 - 24*A^2*B*a*b^9*c*d^8 + 11*A
*B^2*a*b^9*c^2*d^7 + 25*A*B^2*a*b^9*c^4*d^5 - 19*A*B^2*a^2*b^8*c*d^8 + 37*A
*B^2*a^4*b^6*c*d^8 + 32*A*B^2*a^6*b^4*c*d^8 - 23*A^2*B*a*b^9*c^3*d^6 + 11*A
^2*B*a*b^9*c^5*d^4 - 81*A^2*B*a^3*b^7*c*d^8 - 15*A^2*B*a^5*b^5*c*d^8 + 6*A^
2*B*a^7*b^3*c*d^8 - 15*A*C^2*a*b^9*c^2*d^7 - 15*A*C^2*a*b^9*c^4*d^5 + 12*A*
C^2*a*b^9*c^6*d^3 - 3*A*C^2*a^2*b^8*c*d^8 - 27*A*C^2*a^4*b^6*c*d^8 - 6*A*C^
2*a^6*b^4*c*d^8 + 6*A*C^2*a^8*b^2*c*d^8 + 12*A^2*C*a*b^9*c^2*d^7 + 27*A^2*C
*a*b^9*c^4*d^5 - 6*A^2*C*a*b^9*c^6*d^3 + 6*A^2*C*a^2*b^8*c*d^8 + 60*A^2*C*a
^4*b^6*c*d^8 + 15*A^2*C*a^6*b^4*c*d^8 - 3*A^2*C*a^8*b^2*c*d^8 + 13*B*C^2*a*
b^9*c^3*d^6 + 23*B*C^2*a*b^9*c^5*d^4 + 3*B*C^2*a^3*b^7*c*d^8 + 9*B*C^2*a^5*
b^5*c*d^8 + 18*B*C^2*a^7*b^3*c*d^8 - 14*B^2*C*a*b^9*c^2*d^7 - 16*B^2*C*a*b^
9*c^4*d^5 + 6*B^2*C*a*b^9*c^6*d^3 - 8*B^2*C*a^2*b^8*c*d^8 - 28*B^2*C*a^4*b^
6*c*d^8 - 29*B^2*C*a^6*b^4*c*d^8 + 3*B^2*C*a^8*b^2*c*d^8 - 28*A*B*C*a^2*b^8
*c^2*d^7 + 28*A*B*C*a^2*b^8*c^4*d^5 + 18*A*B*C*a^2*b^8*c^6*d^3 + 100*A*B*C*
a^3*b^7*c^3*d^6 - 30*A*B*C*a^3*b^7*c^5*d^4 - 79*A*B*C*a^4*b^6*c^2*d^7 - 55*
A*B*C*a^4*b^6*c^4*d^5 - 3*A*B*C*a^4*b^6*c^6*d^3 + 62*A*B*C*a^5*b^5*c^3*d^6
+ 12*A*B*C*a^5*b^5*c^5*d^4 + 14*A*B*C*a^6*b^4*c^2*d^7 - 18*A*B*C*a^6*b^4*c^
4*d^5 - 12*A*B*C*a^7*b^3*c^3*d^6 + 3*A*B*C*a^8*b^2*c^2*d^7 + 24*A*B*C*a*b^9
*c*d^8 + 10*A*B*C*a*b^9*c^3*d^6 - 34*A*B*C*a*b^9*c^5*d^4 + 78*A*B*C*a^3*b^7
*c*d^8 + 6*A*B*C*a^5*b^5*c*d^8 - 24*A*B*C*a^7*b^3*c*d^8)/(a^14*d^10 + b^14*
c^10 + 4*a^2*b^12*c^10 + 6*a^4*b^10*c^10 + 4*a^6*b^8*c^10 + a^8*b^6*c^10 +
a^6*b^8*d^10 + 4*a^8*b^6*d^10 + 6*a^10*b^4*d^10 + 4*a^12*b^2*d^10 + 2*a^14*
c^2*d^8 + a^14*c^4*d^6 + b^14*c^6*d^4 + 2*b^14*c^8*d^2 - 6*a*b^13*c^5*d^5 -

12*a*b^13*c^7*d^3 - 24*a^3*b^11*c^9*d - 6*a^5*b^9*c*d^9 - 36*a^5*b^9*c^9*d
- 24*a^7*b^7*c*d^9 - 24*a^7*b^7*c^9*d - 36*a^9*b^5*c*d^9 - 6*a^9*b^5*c^9*d
- 24*a^11*b^3*c*d^9 - 12*a^13*b*c^3*d^7 - 6*a^13*b*c^5*d^5 + 15*a^2*b^12*c
^4*d^6 + 34*a^2*b^12*c^6*d^4 + 23*a^2*b^12*c^8*d^2 - 20*a^3*b^11*c^3*d^7 -
64*a^3*b^11*c^5*d^5 - 68*a^3*b^11*c^7*d^3 + 15*a^4*b^10*c^2*d^8 + 90*a^4*b^
10*c^4*d^6 + 141*a^4*b^10*c^6*d^4 + 72*a^4*b^10*c^8*d^2 - 92*a^5*b^9*c^3*d^
7 - 202*a^5*b^9*c^5*d^5 - 152*a^5*b^9*c^7*d^3 + 62*a^6*b^8*c^2*d^8 + 211*a^
6*b^8*c^4*d^6 + 244*a^6*b^8*c^6*d^4 + 98*a^6*b^8*c^8*d^2 - 168*a^7*b^7*c^3*
d^7 - 288*a^7*b^7*c^5*d^5 - 168*a^7*b^7*c^7*d^3 + 98*a^8*b^6*c^2*d^8 + 244*
a^8*b^6*c^4*d^6 + 211*a^8*b^6*c^6*d^4 + 62*a^8*b^6*c^8*d^2 - 152*a^9*b^5*c^
3*d^7 - 202*a^9*b^5*c^5*d^5 - 92*a^9*b^5*c^7*d^3 + 72*a^10*b^4*c^2*d^8 + 14
1*a^10*b^4*c^4*d^6 + 90*a^10*b^4*c^6*d^4 + 15*a^10*b^4*c^8*d^2 - 68*a^11*b^
3*c^3*d^7 - 64*a^11*b^3*c^5*d^5 - 20*a^11*b^3*c^7*d^3 + 23*a^12*b^2*c^2*d^8
+ 34*a^12*b^2*c^4*d^6 + 15*a^12*b^2*c^6*d^4 - 6*a*b^13*c^9*d - 6*a^13*b*c*
d^9) - root(640*a^13*b^7*c*d^15*f^4 + 640*a^7*b^13*c^15*d*f^4 + 480*a^15*b^
5*c*d^15*f^4 + 480*a^11*b^9*c*d^15*f^4 + 480*a^9*b^11*c^15*d*f^4 + 480*a^5*
b^15*c^15*d*f^4 + 192*a^19*b*c^5*d^11*f^4 + 192*a^17*b^3*c*d^15*f^4 + 192*a
^11*b^9*c^15*d*f^4 + 192*a^9*b^11*c*d^15*f^4 + 192*a^3*b^17*c^15*d*f^4 + 19
2*a*b^19*c^11*d^5*f^4 + 128*a^19*b*c^7*d^9*f^4 + 128*a^19*b*c^3*d^13*f^4 +
128*a*b^19*c^13*d^3*f^4 + 128*a*b^19*c^9*d^7*f^4 + 32*a^19*b*c^9*d^7*f^4 +
32*a^13*b^7*c^15*d*f^4 + 32*a^7*b^13*c*d^15*f^4 + 32*a*b^19*c^7*d^9*f^4 + 3
2*a^19*b*c*d^15*f^4 + 32*a*b^19*c^15*d*f^4 - 47088*a^10*b^10*c^8*d^8*f^4 +
42432*a^11*b^9*c^7*d^9*f^4 + 42432*a^9*b^11*c^9*d^7*f^4 + 39328*a^11*b^9*c^
9*d^7*f^4 + 39328*a^9*b^11*c^7*d^9*f^4 - 36912*a^12*b^8*c^8*d^8*f^4 - 36912
*a^8*b^12*c^8*d^8*f^4 - 34256*a^10*b^10*c^10*d^6*f^4 - 34256*a^10*b^10*c^6*
d^10*f^4 - 31152*a^12*b^8*c^6*d^10*f^4 - 31152*a^8*b^12*c^10*d^6*f^4 + 2812
8*a^13*b^7*c^7*d^9*f^4 + 28128*a^7*b^13*c^9*d^7*f^4 + 24160*a^11*b^9*c^5*d^
11*f^4 + 24160*a^9*b^11*c^11*d^5*f^4 - 23088*a^12*b^8*c^10*d^6*f^4 - 23088*
a^8*b^12*c^6*d^10*f^4 + 22272*a^13*b^7*c^9*d^7*f^4 + 22272*a^7*b^13*c^7*d^9
*f^4 + 19072*a^11*b^9*c^11*d^5*f^4 + 19072*a^9*b^11*c^5*d^11*f^4 + 18624*a^
13*b^7*c^5*d^11*f^4 + 18624*a^7*b^13*c^11*d^5*f^4 - 17328*a^14*b^6*c^8*d^8*
f^4 - 17328*a^6*b^14*c^8*d^8*f^4 - 17232*a^14*b^6*c^6*d^10*f^4 - 17232*a^6*
b^14*c^10*d^6*f^4 - 13520*a^12*b^8*c^4*d^12*f^4 - 13520*a^8*b^12*c^12*d^4*f
^4 - 12464*a^10*b^10*c^12*d^4*f^4 - 12464*a^10*b^10*c^4*d^12*f^4 + 10880*a^
15*b^5*c^7*d^9*f^4 + 10880*a^5*b^15*c^9*d^7*f^4 - 9072*a^14*b^6*c^10*d^6*f^
4 - 9072*a^6*b^14*c^6*d^10*f^4 + 8928*a^13*b^7*c^11*d^5*f^4 + 8928*a^7*b^13
*c^5*d^11*f^4 - 8880*a^14*b^6*c^4*d^12*f^4 - 8880*a^6*b^14*c^12*d^4*f^4 + 8
480*a^15*b^5*c^5*d^11*f^4 + 8480*a^5*b^15*c^11*d^5*f^4 + 7200*a^15*b^5*c^9*
d^7*f^4 + 7200*a^5*b^15*c^7*d^9*f^4 - 6912*a^12*b^8*c^12*d^4*f^4 - 6912*a^8
*b^12*c^4*d^12*f^4 + 6400*a^11*b^9*c^3*d^13*f^4 + 6400*a^9*b^11*c^13*d^3*f^
4 + 5920*a^13*b^7*c^3*d^13*f^4 + 5920*a^7*b^13*c^13*d^3*f^4 - 5392*a^16*b^4
*c^6*d^10*f^4 - 5392*a^4*b^16*c^10*d^6*f^4 - 4428*a^16*b^4*c^8*d^8*f^4 - 44
28*a^4*b^16*c^8*d^8*f^4 + 4128*a^11*b^9*c^13*d^3*f^4 + 4128*a^9*b^11*c^3*d^
13*f^4 - 3328*a^16*b^4*c^4*d^12*f^4 - 3328*a^4*b^16*c^12*d^4*f^4 + 3264*a^1
5*b^5*c^3*d^13*f^4 + 3264*a^5*b^15*c^13*d^3*f^4 - 2480*a^12*b^8*c^2*d^14*f^

4 - 2480*a^8*b^12*c^14*d^2*f^4 + 2240*a^15*b^5*c^11*d^5*f^4 + 2240*a^5*b^15
*c^5*d^11*f^4 - 2128*a^14*b^6*c^12*d^4*f^4 - 2128*a^6*b^14*c^4*d^12*f^4 + 2
112*a^17*b^3*c^7*d^9*f^4 + 2112*a^3*b^17*c^9*d^7*f^4 + 2048*a^17*b^3*c^5*d^
11*f^4 + 2048*a^3*b^17*c^11*d^5*f^4 - 2000*a^14*b^6*c^2*d^14*f^4 - 2000*a^6
*b^14*c^14*d^2*f^4 - 1792*a^16*b^4*c^10*d^6*f^4 - 1792*a^4*b^16*c^6*d^10*f^
4 - 1776*a^10*b^10*c^14*d^2*f^4 - 1776*a^10*b^10*c^2*d^14*f^4 + 1472*a^13*b
^7*c^13*d^3*f^4 + 1472*a^7*b^13*c^3*d^13*f^4 + 1088*a^17*b^3*c^9*d^7*f^4 +
1088*a^3*b^17*c^7*d^9*f^4 + 992*a^17*b^3*c^3*d^13*f^4 + 992*a^3*b^17*c^13*d
^3*f^4 - 912*a^16*b^4*c^2*d^14*f^4 - 912*a^4*b^16*c^14*d^2*f^4 - 768*a^18*b
^2*c^6*d^10*f^4 - 768*a^2*b^18*c^10*d^6*f^4 - 688*a^12*b^8*c^14*d^2*f^4 - 6
88*a^8*b^12*c^2*d^14*f^4 - 592*a^18*b^2*c^4*d^12*f^4 - 592*a^2*b^18*c^12*d^
4*f^4 - 472*a^18*b^2*c^8*d^8*f^4 - 472*a^2*b^18*c^8*d^8*f^4 - 280*a^16*b^4*
c^12*d^4*f^4 - 280*a^4*b^16*c^4*d^12*f^4 + 224*a^17*b^3*c^11*d^5*f^4 + 224*
a^15*b^5*c^13*d^3*f^4 + 224*a^5*b^15*c^3*d^13*f^4 + 224*a^3*b^17*c^5*d^11*f
^4 - 208*a^18*b^2*c^2*d^14*f^4 - 208*a^2*b^18*c^14*d^2*f^4 - 112*a^18*b^2*c
^10*d^6*f^4 - 112*a^14*b^6*c^14*d^2*f^4 - 112*a^6*b^14*c^2*d^14*f^4 - 112*a
^2*b^18*c^6*d^10*f^4 - 24*b^20*c^12*d^4*f^4 - 16*b^20*c^14*d^2*f^4 - 16*b^2
0*c^10*d^6*f^4 - 4*b^20*c^8*d^8*f^4 - 24*a^20*c^4*d^12*f^4 - 16*a^20*c^6*d^
10*f^4 - 16*a^20*c^2*d^14*f^4 - 4*a^20*c^8*d^8*f^4 - 80*a^14*b^6*d^16*f^4 -
 60*a^16*b^4*d^16*f^4 - 60*a^12*b^8*d^16*f^4 - 24*a^18*b^2*d^16*f^4 - 24*a^
10*b^10*d^16*f^4 - 4*a^8*b^12*d^16*f^4 - 80*a^6*b^14*c^16*f^4 - 60*a^8*b^12
*c^16*f^4 - 60*a^4*b^16*c^16*f^4 - 24*a^10*b^10*c^16*f^4 - 24*a^2*b^18*c^16
*f^4 - 4*a^12*b^8*c^16*f^4 - 4*b^20*c^16*f^4 - 4*a^20*d^16*f^4 + 56*A*C*a^1
3*b*c*d^11*f^2 - 48*A*C*a*b^13*c^11*d*f^2 + 48*A*C*a*b^13*c*d^11*f^2 + 5904
*B*C*a^7*b^7*c^6*d^6*f^2 - 5016*B*C*a^8*b^6*c^5*d^7*f^2 - 4608*B*C*a^6*b^8*
c^7*d^5*f^2 - 4512*B*C*a^6*b^8*c^5*d^7*f^2 - 4384*B*C*a^8*b^6*c^7*d^5*f^2 +
 3056*B*C*a^7*b^7*c^8*d^4*f^2 + 2256*B*C*a^7*b^7*c^4*d^8*f^2 - 1824*B*C*a^8
*b^6*c^3*d^9*f^2 + 1632*B*C*a^4*b^10*c^9*d^3*f^2 - 1400*B*C*a^3*b^11*c^8*d^
4*f^2 - 1320*B*C*a^11*b^3*c^4*d^8*f^2 - 1248*B*C*a^6*b^8*c^3*d^9*f^2 + 1152
*B*C*a^10*b^4*c^3*d^9*f^2 - 1072*B*C*a^6*b^8*c^9*d^3*f^2 + 1068*B*C*a^9*b^5
*c^6*d^6*f^2 - 1004*B*C*a^5*b^9*c^4*d^8*f^2 - 968*B*C*a^3*b^11*c^6*d^6*f^2
- 864*B*C*a^5*b^9*c^8*d^4*f^2 - 828*B*C*a^9*b^5*c^4*d^8*f^2 - 792*B*C*a^11*
b^3*c^2*d^10*f^2 - 792*B*C*a^3*b^11*c^4*d^8*f^2 - 776*B*C*a^8*b^6*c^9*d^3*f
^2 + 688*B*C*a^4*b^10*c^7*d^5*f^2 - 672*B*C*a^3*b^11*c^10*d^2*f^2 - 592*B*C
*a^9*b^5*c^2*d^10*f^2 + 544*B*C*a^7*b^7*c^10*d^2*f^2 - 492*B*C*a^5*b^9*c^2*
d^10*f^2 + 480*B*C*a^10*b^4*c^5*d^7*f^2 - 392*B*C*a^5*b^9*c^10*d^2*f^2 + 33
2*B*C*a^9*b^5*c^8*d^4*f^2 - 328*B*C*a^11*b^3*c^6*d^6*f^2 + 320*B*C*a^2*b^12
*c^9*d^3*f^2 + 272*B*C*a^12*b^2*c^3*d^9*f^2 - 248*B*C*a^4*b^10*c^5*d^7*f^2
- 248*B*C*a^3*b^11*c^2*d^10*f^2 - 208*B*C*a^10*b^4*c^7*d^5*f^2 - 192*B*C*a^
2*b^12*c^5*d^7*f^2 + 144*B*C*a^7*b^7*c^2*d^10*f^2 - 96*B*C*a^4*b^10*c^3*d^9
*f^2 + 88*B*C*a^12*b^2*c^5*d^7*f^2 - 72*B*C*a^11*b^3*c^8*d^4*f^2 - 48*B*C*a
^12*b^2*c^7*d^5*f^2 + 48*B*C*a^10*b^4*c^9*d^3*f^2 - 48*B*C*a^2*b^12*c^7*d^5
*f^2 - 48*B*C*a^2*b^12*c^3*d^9*f^2 - 12*B*C*a^9*b^5*c^10*d^2*f^2 + 4*B*C*a^
5*b^9*c^6*d^6*f^2 + 5824*A*C*a^5*b^9*c^7*d^5*f^2 - 4378*A*C*a^6*b^8*c^8*d^4
*f^2 + 4296*A*C*a^5*b^9*c^5*d^7*f^2 - 3912*A*C*a^6*b^8*c^6*d^6*f^2 - 3672*A

*C*a^9*b^5*c^5*d^7*f^2 + 3594*A*C*a^8*b^6*c^4*d^8*f^2 + 3236*A*C*a^8*b^6*c^
6*d^6*f^2 + 2816*A*C*a^5*b^9*c^9*d^3*f^2 + 2624*A*C*a^5*b^9*c^3*d^9*f^2 + 2
432*A*C*a^7*b^7*c^7*d^5*f^2 - 2366*A*C*a^4*b^10*c^8*d^4*f^2 + 2298*A*C*a^10
*b^4*c^4*d^8*f^2 + 1872*A*C*a^7*b^7*c^3*d^9*f^2 + 1848*A*C*a^10*b^4*c^6*d^6
*f^2 - 1644*A*C*a^4*b^10*c^6*d^6*f^2 - 1488*A*C*a^9*b^5*c^7*d^5*f^2 - 1408*
A*C*a^9*b^5*c^3*d^9*f^2 - 1308*A*C*a^6*b^8*c^4*d^8*f^2 + 1248*A*C*a^7*b^7*c
^5*d^7*f^2 - 1012*A*C*a^6*b^8*c^10*d^2*f^2 + 1008*A*C*a^3*b^11*c^7*d^5*f^2
+ 992*A*C*a^3*b^11*c^5*d^7*f^2 + 928*A*C*a^3*b^11*c^3*d^9*f^2 + 848*A*C*a^7
*b^7*c^9*d^3*f^2 + 636*A*C*a^8*b^6*c^2*d^10*f^2 - 628*A*C*a^4*b^10*c^10*d^2
*f^2 - 600*A*C*a^6*b^8*c^2*d^10*f^2 - 576*A*C*a^11*b^3*c^5*d^7*f^2 + 572*A*
C*a^10*b^4*c^2*d^10*f^2 + 464*A*C*a^8*b^6*c^8*d^4*f^2 - 304*A*C*a^4*b^10*c^
4*d^8*f^2 + 304*A*C*a^2*b^12*c^6*d^6*f^2 + 296*A*C*a^2*b^12*c^4*d^8*f^2 + 2
60*A*C*a^10*b^4*c^8*d^4*f^2 - 232*A*C*a^12*b^2*c^2*d^10*f^2 - 232*A*C*a^9*b
^5*c^9*d^3*f^2 + 228*A*C*a^2*b^12*c^10*d^2*f^2 - 188*A*C*a^4*b^10*c^2*d^10*
f^2 + 144*A*C*a^11*b^3*c^3*d^9*f^2 + 116*A*C*a^12*b^2*c^6*d^6*f^2 - 112*A*C
*a^11*b^3*c^7*d^5*f^2 + 112*A*C*a^3*b^11*c^9*d^3*f^2 + 92*A*C*a^8*b^6*c^10*
d^2*f^2 + 74*A*C*a^12*b^2*c^4*d^8*f^2 + 62*A*C*a^2*b^12*c^8*d^4*f^2 + 40*A*
C*a^2*b^12*c^2*d^10*f^2 - 7008*A*B*a^7*b^7*c^6*d^6*f^2 - 4032*A*B*a^7*b^7*c
^4*d^8*f^2 + 3952*A*B*a^8*b^6*c^7*d^5*f^2 + 3648*A*B*a^8*b^6*c^5*d^7*f^2 -
3392*A*B*a^7*b^7*c^8*d^4*f^2 + 3264*A*B*a^6*b^8*c^7*d^5*f^2 - 2992*A*B*a^4*
b^10*c^5*d^7*f^2 - 2368*A*B*a^4*b^10*c^7*d^5*f^2 - 2304*A*B*a^4*b^10*c^3*d^
9*f^2 - 1968*A*B*a^9*b^5*c^6*d^6*f^2 - 1872*A*B*a^4*b^10*c^9*d^3*f^2 - 1728
*A*B*a^7*b^7*c^2*d^10*f^2 + 1712*A*B*a^3*b^11*c^8*d^4*f^2 - 1536*A*B*a^10*b
^4*c^3*d^9*f^2 + 1536*A*B*a^6*b^8*c^5*d^7*f^2 - 1392*A*B*a^2*b^12*c^5*d^7*f
^2 + 1328*A*B*a^3*b^11*c^6*d^6*f^2 - 1104*A*B*a^2*b^12*c^3*d^9*f^2 - 1056*A
*B*a^6*b^8*c^3*d^9*f^2 + 976*A*B*a^6*b^8*c^9*d^3*f^2 + 960*A*B*a^11*b^3*c^4
*d^8*f^2 + 936*A*B*a^5*b^9*c^8*d^4*f^2 - 912*A*B*a^10*b^4*c^5*d^7*f^2 + 848
*A*B*a^8*b^6*c^9*d^3*f^2 + 816*A*B*a^3*b^11*c^4*d^8*f^2 - 816*A*B*a^2*b^12*
c^7*d^5*f^2 + 768*A*B*a^3*b^11*c^10*d^2*f^2 + 672*A*B*a^8*b^6*c^3*d^9*f^2 -
632*A*B*a^9*b^5*c^8*d^4*f^2 - 608*A*B*a^9*b^5*c^2*d^10*f^2 - 552*A*B*a^9*b
^5*c^4*d^8*f^2 - 544*A*B*a^7*b^7*c^10*d^2*f^2 - 480*A*B*a^5*b^9*c^2*d^10*f^
2 + 464*A*B*a^5*b^9*c^10*d^2*f^2 - 464*A*B*a^2*b^12*c^9*d^3*f^2 + 432*A*B*a
^11*b^3*c^2*d^10*f^2 - 368*A*B*a^12*b^2*c^3*d^9*f^2 - 256*A*B*a^5*b^9*c^6*d
^6*f^2 - 208*A*B*a^12*b^2*c^5*d^7*f^2 + 176*A*B*a^5*b^9*c^4*d^8*f^2 + 112*A
*B*a^11*b^3*c^6*d^6*f^2 + 112*A*B*a^10*b^4*c^7*d^5*f^2 - 16*A*B*a^3*b^11*c^
2*d^10*f^2 - 576*B*C*a^8*b^6*c*d^11*f^2 + 400*B*C*a^4*b^10*c^11*d*f^2 - 288
*B*C*a^6*b^8*c*d^11*f^2 - 176*B*C*a^6*b^8*c^11*d*f^2 + 128*B*C*a^10*b^4*c*d
^11*f^2 - 108*B*C*a*b^13*c^4*d^8*f^2 - 104*B*C*a^4*b^10*c*d^11*f^2 - 92*B*C
*a^13*b*c^4*d^8*f^2 - 60*B*C*a*b^13*c^8*d^4*f^2 - 60*B*C*a*b^13*c^6*d^6*f^2
+ 48*B*C*a^2*b^12*c^11*d*f^2 - 40*B*C*a*b^13*c^2*d^10*f^2 - 28*B*C*a^13*b*
c^2*d^10*f^2 - 24*B*C*a^12*b^2*c*d^11*f^2 + 20*B*C*a*b^13*c^10*d^2*f^2 - 16
*B*C*a^2*b^12*c*d^11*f^2 + 12*B*C*a^13*b*c^6*d^6*f^2 + 912*A*C*a^7*b^7*c*d^
11*f^2 + 808*A*C*a^5*b^9*c*d^11*f^2 + 432*A*C*a^5*b^9*c^11*d*f^2 + 336*A*C*
a^3*b^11*c*d^11*f^2 + 224*A*C*a^11*b^3*c*d^11*f^2 - 112*A*C*a^3*b^11*c^11*d
*f^2 + 112*A*C*a*b^13*c^3*d^9*f^2 - 88*A*C*a*b^13*c^9*d^3*f^2 + 80*A*C*a^13

*b*c^3*d^9*f^2 + 56*A*C*a*b^13*c^5*d^7*f^2 + 48*A*C*a^9*b^5*c*d^11*f^2 - 40
*A*C*a^13*b*c^5*d^7*f^2 - 16*A*C*a^7*b^7*c^11*d*f^2 + 16*A*C*a*b^13*c^7*d^5
*f^2 - 496*A*B*a^4*b^10*c*d^11*f^2 - 400*A*B*a^4*b^10*c^11*d*f^2 + 288*A*B*
a^8*b^6*c*d^11*f^2 - 288*A*B*a^6*b^8*c*d^11*f^2 - 272*A*B*a^2*b^12*c*d^11*f
^2 + 240*A*B*a*b^13*c^6*d^6*f^2 - 224*A*B*a^10*b^4*c*d^11*f^2 + 192*A*B*a*b
^13*c^8*d^4*f^2 + 192*A*B*a*b^13*c^4*d^8*f^2 + 176*A*B*a^6*b^8*c^11*d*f^2 +
104*A*B*a^13*b*c^4*d^8*f^2 - 48*A*B*a^2*b^12*c^11*d*f^2 + 16*A*B*a^13*b*c^
2*d^10*f^2 + 16*A*B*a*b^13*c^10*d^2*f^2 + 16*A*B*a*b^13*c^2*d^10*f^2 - 96*B
*C*b^14*c^7*d^5*f^2 - 72*B*C*b^14*c^5*d^7*f^2 - 24*B*C*b^14*c^9*d^3*f^2 - 1
6*B*C*b^14*c^3*d^9*f^2 + 116*A*C*b^14*c^6*d^6*f^2 + 100*A*C*b^14*c^4*d^8*f^
2 + 24*A*C*b^14*c^2*d^10*f^2 + 22*A*C*b^14*c^8*d^4*f^2 + 16*B*C*a^14*c^3*d^
9*f^2 + 8*A*C*b^14*c^10*d^2*f^2 - 192*A*B*b^14*c^5*d^7*f^2 - 176*A*B*b^14*c
^3*d^9*f^2 - 112*B*C*a^11*b^3*d^12*f^2 - 48*A*B*b^14*c^7*d^5*f^2 - 28*A*C*a
^14*c^2*d^10*f^2 + 4*B*C*a^5*b^9*d^12*f^2 + 2*A*C*a^14*c^4*d^8*f^2 + 150*A*
C*a^10*b^4*d^12*f^2 - 80*B*C*a^3*b^11*c^12*f^2 + 66*A*C*a^8*b^6*d^12*f^2 -
30*A*C*a^12*b^2*d^12*f^2 + 24*B*C*a^5*b^9*c^12*f^2 - 16*A*B*a^14*c^3*d^9*f^
2 - 12*A*C*a^4*b^10*d^12*f^2 - 576*A*B*a^7*b^7*d^12*f^2 - 432*A*B*a^9*b^5*d
^12*f^2 - 400*A*B*a^5*b^9*d^12*f^2 - 144*A*B*a^3*b^11*d^12*f^2 - 66*A*C*a^4
*b^10*c^12*f^2 + 54*A*C*a^2*b^12*c^12*f^2 - 32*A*B*a^11*b^3*d^12*f^2 + 2*A*
C*a^6*b^8*c^12*f^2 + 80*A*B*a^3*b^11*c^12*f^2 - 24*A*B*a^5*b^9*c^12*f^2 + 2
508*C^2*a^6*b^8*c^6*d^6*f^2 + 2376*C^2*a^9*b^5*c^5*d^7*f^2 + 2357*C^2*a^6*b
^8*c^8*d^4*f^2 - 2048*C^2*a^5*b^9*c^7*d^5*f^2 + 1304*C^2*a^9*b^5*c^3*d^9*f^
2 + 1303*C^2*a^4*b^10*c^8*d^4*f^2 + 1212*C^2*a^4*b^10*c^6*d^6*f^2 - 1203*C^
2*a^8*b^6*c^4*d^8*f^2 - 1192*C^2*a^5*b^9*c^9*d^3*f^2 + 1062*C^2*a^6*b^8*c^4
*d^8*f^2 + 984*C^2*a^9*b^5*c^7*d^5*f^2 - 952*C^2*a^8*b^6*c^6*d^6*f^2 + 768*
C^2*a^7*b^7*c^5*d^7*f^2 - 681*C^2*a^10*b^4*c^4*d^8*f^2 - 672*C^2*a^5*b^9*c^
5*d^7*f^2 - 480*C^2*a^10*b^4*c^6*d^6*f^2 + 458*C^2*a^6*b^8*c^10*d^2*f^2 - 4
48*C^2*a^7*b^7*c^7*d^5*f^2 + 422*C^2*a^4*b^10*c^4*d^8*f^2 + 372*C^2*a^6*b^8
*c^2*d^10*f^2 + 360*C^2*a^11*b^3*c^5*d^7*f^2 + 312*C^2*a^7*b^7*c^3*d^9*f^2
+ 278*C^2*a^4*b^10*c^10*d^2*f^2 - 232*C^2*a^7*b^7*c^9*d^3*f^2 + 194*C^2*a^1
2*b^2*c^2*d^10*f^2 + 176*C^2*a^9*b^5*c^9*d^3*f^2 + 152*C^2*a^3*b^11*c^5*d^7
*f^2 + 124*C^2*a^4*b^10*c^2*d^10*f^2 - 120*C^2*a^3*b^11*c^7*d^5*f^2 - 114*C
^2*a^2*b^12*c^10*d^2*f^2 - 102*C^2*a^8*b^6*c^2*d^10*f^2 + 101*C^2*a^12*b^2*
c^4*d^8*f^2 + 100*C^2*a^2*b^12*c^6*d^6*f^2 - 88*C^2*a^5*b^9*c^3*d^9*f^2 + 7
7*C^2*a^2*b^12*c^8*d^4*f^2 + 72*C^2*a^11*b^3*c^3*d^9*f^2 - 64*C^2*a^8*b^6*c
^10*d^2*f^2 + 64*C^2*a^3*b^11*c^3*d^9*f^2 - 58*C^2*a^10*b^4*c^2*d^10*f^2 +
56*C^2*a^12*b^2*c^6*d^6*f^2 + 56*C^2*a^11*b^3*c^7*d^5*f^2 + 40*C^2*a^3*b^11
*c^9*d^3*f^2 + 36*C^2*a^12*b^2*c^8*d^4*f^2 + 32*C^2*a^2*b^12*c^4*d^8*f^2 +
26*C^2*a^10*b^4*c^8*d^4*f^2 + 16*C^2*a^2*b^12*c^2*d^10*f^2 + 2*C^2*a^8*b^6*
c^8*d^4*f^2 + 2277*B^2*a^8*b^6*c^4*d^8*f^2 + 2144*B^2*a^5*b^9*c^7*d^5*f^2 -
2112*B^2*a^9*b^5*c^5*d^7*f^2 + 2028*B^2*a^8*b^6*c^6*d^6*f^2 - 1671*B^2*a^6
*b^8*c^8*d^4*f^2 + 1275*B^2*a^10*b^4*c^4*d^8*f^2 + 1176*B^2*a^5*b^9*c^5*d^7
*f^2 + 1096*B^2*a^5*b^9*c^9*d^3*f^2 - 1044*B^2*a^6*b^8*c^6*d^6*f^2 + 984*B^
2*a^10*b^4*c^6*d^6*f^2 - 968*B^2*a^9*b^5*c^3*d^9*f^2 - 888*B^2*a^9*b^5*c^7*
d^5*f^2 + 672*B^2*a^7*b^7*c^7*d^5*f^2 + 664*B^2*a^5*b^9*c^3*d^9*f^2 - 649*B

^2*a^4*b^10*c^8*d^4*f^2 + 618*B^2*a^8*b^6*c^2*d^10*f^2 + 514*B^2*a^4*b^10*c
^4*d^8*f^2 + 460*B^2*a^2*b^12*c^6*d^6*f^2 + 422*B^2*a^8*b^6*c^8*d^4*f^2 + 4
06*B^2*a^10*b^4*c^2*d^10*f^2 - 382*B^2*a^6*b^8*c^10*d^2*f^2 + 368*B^2*a^2*b
^12*c^4*d^8*f^2 - 312*B^2*a^11*b^3*c^5*d^7*f^2 + 312*B^2*a^7*b^7*c^3*d^9*f^
2 + 248*B^2*a^7*b^7*c^9*d^3*f^2 + 245*B^2*a^2*b^12*c^8*d^4*f^2 - 192*B^2*a^
7*b^7*c^5*d^7*f^2 - 184*B^2*a^3*b^11*c^9*d^3*f^2 + 182*B^2*a^2*b^12*c^10*d^
2*f^2 + 176*B^2*a^3*b^11*c^3*d^9*f^2 + 174*B^2*a^6*b^8*c^4*d^8*f^2 - 170*B^
2*a^4*b^10*c^10*d^2*f^2 - 152*B^2*a^9*b^5*c^9*d^3*f^2 + 152*B^2*a^4*b^10*c^
2*d^10*f^2 + 142*B^2*a^10*b^4*c^8*d^4*f^2 - 90*B^2*a^12*b^2*c^2*d^10*f^2 +
88*B^2*a^2*b^12*c^2*d^10*f^2 + 84*B^2*a^8*b^6*c^10*d^2*f^2 + 84*B^2*a^6*b^8
*c^2*d^10*f^2 + 60*B^2*a^12*b^2*c^6*d^6*f^2 - 56*B^2*a^11*b^3*c^7*d^5*f^2 +
53*B^2*a^12*b^2*c^4*d^8*f^2 + 24*B^2*a^11*b^3*c^3*d^9*f^2 + 24*B^2*a^4*b^1
0*c^6*d^6*f^2 + 24*B^2*a^3*b^11*c^7*d^5*f^2 - 8*B^2*a^3*b^11*c^5*d^7*f^2 +
4566*A^2*a^6*b^8*c^4*d^8*f^2 + 4284*A^2*a^6*b^8*c^6*d^6*f^2 - 3776*A^2*a^5*
b^9*c^7*d^5*f^2 - 3624*A^2*a^5*b^9*c^5*d^7*f^2 + 3122*A^2*a^4*b^10*c^4*d^8*
f^2 + 3108*A^2*a^6*b^8*c^2*d^10*f^2 + 2741*A^2*a^6*b^8*c^8*d^4*f^2 + 2592*A
^2*a^4*b^10*c^6*d^6*f^2 - 2536*A^2*a^5*b^9*c^3*d^9*f^2 + 2224*A^2*a^4*b^10*
c^2*d^10*f^2 - 2184*A^2*a^7*b^7*c^3*d^9*f^2 - 2016*A^2*a^7*b^7*c^5*d^7*f^2
- 1984*A^2*a^7*b^7*c^7*d^5*f^2 + 1626*A^2*a^8*b^6*c^2*d^10*f^2 - 1624*A^2*a
^5*b^9*c^9*d^3*f^2 + 1603*A^2*a^4*b^10*c^8*d^4*f^2 + 1296*A^2*a^9*b^5*c^5*d
^7*f^2 - 1144*A^2*a^3*b^11*c^5*d^7*f^2 - 992*A^2*a^3*b^11*c^3*d^9*f^2 + 968
*A^2*a^2*b^12*c^4*d^8*f^2 - 888*A^2*a^3*b^11*c^7*d^5*f^2 + 849*A^2*a^8*b^6*
c^4*d^8*f^2 + 808*A^2*a^2*b^12*c^2*d^10*f^2 - 616*A^2*a^7*b^7*c^9*d^3*f^2 +
554*A^2*a^6*b^8*c^10*d^2*f^2 - 504*A^2*a^10*b^4*c^6*d^6*f^2 + 504*A^2*a^9*
b^5*c^7*d^5*f^2 + 460*A^2*a^2*b^12*c^6*d^6*f^2 + 350*A^2*a^10*b^4*c^2*d^10*
f^2 + 350*A^2*a^4*b^10*c^10*d^2*f^2 - 321*A^2*a^10*b^4*c^4*d^8*f^2 + 216*A^
2*a^11*b^3*c^5*d^7*f^2 - 216*A^2*a^11*b^3*c^3*d^9*f^2 + 182*A^2*a^12*b^2*c^
2*d^10*f^2 - 152*A^2*a^3*b^11*c^9*d^3*f^2 - 124*A^2*a^8*b^6*c^6*d^6*f^2 - 1
14*A^2*a^2*b^12*c^10*d^2*f^2 + 104*A^2*a^9*b^5*c^3*d^9*f^2 + 77*A^2*a^2*b^1
2*c^8*d^4*f^2 + 74*A^2*a^8*b^6*c^8*d^4*f^2 - 70*A^2*a^10*b^4*c^8*d^4*f^2 +
56*A^2*a^11*b^3*c^7*d^5*f^2 + 56*A^2*a^9*b^5*c^9*d^3*f^2 + 41*A^2*a^12*b^2*
c^4*d^8*f^2 - 28*A^2*a^12*b^2*c^6*d^6*f^2 - 28*A^2*a^8*b^6*c^10*d^2*f^2 - 1
6*B*C*b^14*c^11*d*f^2 - 16*B*C*a^14*c*d^11*f^2 - 48*A*B*b^14*c*d^11*f^2 + 1
6*A*B*b^14*c^11*d*f^2 + 12*B*C*a^13*b*d^12*f^2 + 24*B*C*a*b^13*c^12*f^2 + 1
6*A*B*a^14*c*d^11*f^2 - 24*A*B*a^13*b*d^12*f^2 - 24*A*B*a*b^13*d^12*f^2 - 2
4*A*B*a*b^13*c^12*f^2 + 216*C^2*a^9*b^5*c*d^11*f^2 - 216*C^2*a^5*b^9*c^11*d
*f^2 + 56*C^2*a^3*b^11*c^11*d*f^2 + 56*C^2*a*b^13*c^9*d^3*f^2 + 56*C^2*a*b^
13*c^5*d^7*f^2 - 40*C^2*a^11*b^3*c*d^11*f^2 + 40*C^2*a*b^13*c^7*d^5*f^2 + 3
2*C^2*a^13*b*c^5*d^7*f^2 - 24*C^2*a^7*b^7*c*d^11*f^2 - 16*C^2*a^13*b*c^3*d^
9*f^2 + 16*C^2*a*b^13*c^3*d^9*f^2 + 8*C^2*a^7*b^7*c^11*d*f^2 - 8*C^2*a^5*b^
9*c*d^11*f^2 + 264*B^2*a^7*b^7*c*d^11*f^2 + 224*B^2*a^5*b^9*c*d^11*f^2 + 16
8*B^2*a^5*b^9*c^11*d*f^2 - 112*B^2*a*b^13*c^9*d^3*f^2 - 104*B^2*a^3*b^11*c^
11*d*f^2 - 104*B^2*a*b^13*c^7*d^5*f^2 + 96*B^2*a^3*b^11*c*d^11*f^2 + 88*B^2
*a^11*b^3*c*d^11*f^2 - 72*B^2*a^9*b^5*c*d^11*f^2 - 64*B^2*a*b^13*c^5*d^7*f^
2 + 32*B^2*a^13*b*c^3*d^9*f^2 - 24*B^2*a^13*b*c^5*d^7*f^2 - 24*B^2*a^7*b^7*

c^11*d*f^2 + 16*B^2*a*b^13*c^3*d^9*f^2 - 888*A^2*a^7*b^7*c*d^11*f^2 - 800*A
^2*a^5*b^9*c*d^11*f^2 - 336*A^2*a^3*b^11*c*d^11*f^2 - 264*A^2*a^9*b^5*c*d^1
1*f^2 - 216*A^2*a^5*b^9*c^11*d*f^2 - 184*A^2*a^11*b^3*c*d^11*f^2 - 128*A^2*
a*b^13*c^3*d^9*f^2 - 112*A^2*a*b^13*c^5*d^7*f^2 - 64*A^2*a^13*b*c^3*d^9*f^2
 + 56*A^2*a^3*b^11*c^11*d*f^2 - 56*A^2*a*b^13*c^7*d^5*f^2 + 32*A^2*a*b^13*c
^9*d^3*f^2 + 8*A^2*a^13*b*c^5*d^7*f^2 + 8*A^2*a^7*b^7*c^11*d*f^2 + 24*C^2*a
*b^13*c^11*d*f^2 - 16*C^2*a^13*b*c*d^11*f^2 - 40*B^2*a*b^13*c^11*d*f^2 + 24
*B^2*a^13*b*c*d^11*f^2 + 16*B^2*a*b^13*c*d^11*f^2 - 48*A^2*a*b^13*c*d^11*f^
2 - 40*A^2*a^13*b*c*d^11*f^2 + 24*A^2*a*b^13*c^11*d*f^2 - 6*A*C*b^14*c^12*f
^2 + 2*A*C*a^14*d^12*f^2 + 31*C^2*b^14*c^8*d^4*f^2 + 20*C^2*b^14*c^6*d^6*f^
2 + 4*C^2*b^14*c^4*d^8*f^2 + 2*C^2*b^14*c^10*d^2*f^2 + 80*B^2*b^14*c^6*d^6*
f^2 + 64*B^2*b^14*c^4*d^8*f^2 + 31*B^2*b^14*c^8*d^4*f^2 + 16*B^2*b^14*c^2*d
^10*f^2 + 14*C^2*a^14*c^2*d^10*f^2 + 14*B^2*b^14*c^10*d^2*f^2 - C^2*a^14*c^
4*d^8*f^2 + 120*A^2*b^14*c^2*d^10*f^2 + 112*A^2*b^14*c^4*d^8*f^2 + 33*C^2*a
^12*b^2*d^12*f^2 - 27*C^2*a^10*b^4*d^12*f^2 - 17*A^2*b^14*c^8*d^4*f^2 - 10*
B^2*a^14*c^2*d^10*f^2 - 10*A^2*b^14*c^10*d^2*f^2 + 8*A^2*b^14*c^6*d^6*f^2 +
 3*C^2*a^8*b^6*d^12*f^2 + 3*B^2*a^14*c^4*d^8*f^2 + 117*B^2*a^10*b^4*d^12*f^
2 + 111*B^2*a^8*b^6*d^12*f^2 + 72*B^2*a^6*b^8*d^12*f^2 + 33*C^2*a^4*b^10*c^
12*f^2 - 27*C^2*a^2*b^12*c^12*f^2 + 24*B^2*a^4*b^10*d^12*f^2 + 14*A^2*a^14*
c^2*d^10*f^2 + 4*B^2*a^2*b^12*d^12*f^2 - 3*B^2*a^12*b^2*d^12*f^2 - C^2*a^6*
b^8*c^12*f^2 - A^2*a^14*c^4*d^8*f^2 + 720*A^2*a^6*b^8*d^12*f^2 + 552*A^2*a^
4*b^10*d^12*f^2 + 471*A^2*a^8*b^6*d^12*f^2 + 216*A^2*a^2*b^12*d^12*f^2 + 93
*A^2*a^10*b^4*d^12*f^2 + 33*B^2*a^2*b^12*c^12*f^2 + 33*A^2*a^12*b^2*d^12*f^
2 - 27*B^2*a^4*b^10*c^12*f^2 + 3*B^2*a^6*b^8*c^12*f^2 + 33*A^2*a^4*b^10*c^1
2*f^2 - 27*A^2*a^2*b^12*c^12*f^2 - A^2*a^6*b^8*c^12*f^2 + 3*C^2*b^14*c^12*f
^2 - C^2*a^14*d^12*f^2 + 36*A^2*b^14*d^12*f^2 + 3*B^2*a^14*d^12*f^2 - B^2*b
^14*c^12*f^2 + 3*A^2*b^14*c^12*f^2 - A^2*a^14*d^12*f^2 - 44*A*B*C*a^10*b*c*
d^9*f + 3816*A*B*C*a^4*b^7*c^5*d^5*f + 2920*A*B*C*a^5*b^6*c^2*d^8*f - 2736*
A*B*C*a^6*b^5*c^3*d^7*f - 2672*A*B*C*a^3*b^8*c^4*d^6*f + 1996*A*B*C*a^7*b^4
*c^4*d^6*f - 1412*A*B*C*a^5*b^6*c^6*d^4*f + 1120*A*B*C*a^2*b^9*c^3*d^7*f +
1080*A*B*C*a^7*b^4*c^2*d^8*f + 1040*A*B*C*a^2*b^9*c^5*d^5*f + 684*A*B*C*a^5
*b^6*c^4*d^6*f + 592*A*B*C*a^4*b^7*c^3*d^7*f - 560*A*B*C*a^2*b^9*c^7*d^3*f
- 448*A*B*C*a^3*b^8*c^2*d^8*f - 400*A*B*C*a^8*b^3*c^5*d^5*f - 398*A*B*C*a^9
*b^2*c^2*d^8*f - 312*A*B*C*a^3*b^8*c^6*d^4*f + 166*A*B*C*a^3*b^8*c^8*d^2*f
+ 136*A*B*C*a^6*b^5*c^5*d^5*f + 128*A*B*C*a^6*b^5*c^7*d^3*f - 100*A*B*C*a^7
*b^4*c^6*d^4*f - 64*A*B*C*a^9*b^2*c^4*d^6*f + 64*A*B*C*a^4*b^7*c^7*d^3*f -
32*A*B*C*a^8*b^3*c^3*d^7*f - 16*A*B*C*a^5*b^6*c^8*d^2*f - 1312*A*B*C*a^4*b^
7*c*d^9*f + 996*A*B*C*a^8*b^3*c*d^9*f + 728*A*B*C*a*b^10*c^6*d^4*f - 624*A*
B*C*a^6*b^5*c*d^9*f - 584*A*B*C*a*b^10*c^2*d^8*f - 512*A*B*C*a*b^10*c^4*d^6
*f - 320*A*B*C*a^2*b^9*c*d^9*f - 98*A*B*C*a*b^10*c^8*d^2*f + 36*A*B*C*a^2*b
^9*c^9*d*f + 32*A*B*C*a^10*b*c^3*d^7*f - 16*A*B*C*a^4*b^7*c^9*d*f + 46*B*C^
2*a^10*b*c*d^9*f - 16*B^2*C*a*b^10*c*d^9*f - 2*B^2*C*a*b^10*c^9*d*f + 312*A
^2*C*a*b^10*c*d^9*f - 48*A*C^2*a*b^10*c*d^9*f - 6*A^2*C*a*b^10*c^9*d*f + 6*
A*C^2*a*b^10*c^9*d*f + 208*A*B^2*a*b^10*c*d^9*f - 2*A^2*B*a^10*b*c*d^9*f +
2*A*B^2*a*b^10*c^9*d*f - 224*A*B*C*b^11*c^5*d^5*f + 80*A*B*C*b^11*c^7*d^3*f

 - 32*A*B*C*b^11*c^3*d^7*f + 2*A*B*C*a^11*c^2*d^8*f - 480*A*B*C*a^7*b^4*d^1
0*f + 78*A*B*C*a^9*b^2*d^10*f - 64*A*B*C*a^5*b^6*d^10*f + 2*A*B*C*a^3*b^8*c
^10*f - 1692*B*C^2*a^4*b^7*c^5*d^5*f - 1500*B^2*C*a^5*b^6*c^5*d^5*f - 1464*
B^2*C*a^5*b^6*c^3*d^7*f + 1426*B*C^2*a^5*b^6*c^6*d^4*f - 1158*B^2*C*a^4*b^7
*c^6*d^4*f + 1152*B*C^2*a^6*b^5*c^3*d^7*f + 1026*B^2*C*a^6*b^5*c^4*d^6*f -
974*B*C^2*a^7*b^4*c^4*d^6*f + 960*B^2*C*a^3*b^8*c^5*d^5*f - 884*B*C^2*a^5*b
^6*c^2*d^8*f - 764*B^2*C*a^7*b^4*c^5*d^5*f + 752*B^2*C*a^4*b^7*c^2*d^8*f -
752*B*C^2*a^4*b^7*c^3*d^7*f + 738*B^2*C*a^4*b^7*c^4*d^6*f - 688*B^2*C*a^2*b
^9*c^6*d^4*f - 675*B^2*C*a^8*b^3*c^2*d^8*f + 560*B*C^2*a^8*b^3*c^5*d^5*f +
496*B*C^2*a^3*b^8*c^4*d^6*f + 496*B*C^2*a^2*b^9*c^7*d^3*f - 468*B*C^2*a^7*b
^4*c^2*d^8*f + 456*B^2*C*a^3*b^8*c^7*d^3*f - 452*B^2*C*a^8*b^3*c^4*d^6*f -
416*B*C^2*a^2*b^9*c^3*d^7*f + 378*B*C^2*a^5*b^6*c^4*d^6*f + 376*B*C^2*a^8*b
^3*c^3*d^7*f - 360*B^2*C*a^6*b^5*c^2*d^8*f + 355*B*C^2*a^9*b^2*c^2*d^8*f +
346*B^2*C*a^6*b^5*c^6*d^4*f - 320*B^2*C*a^2*b^9*c^4*d^6*f + 268*B^2*C*a^2*b
^9*c^2*d^8*f + 216*B^2*C*a^7*b^4*c^3*d^7*f - 203*B*C^2*a^3*b^8*c^8*d^2*f -
184*B*C^2*a^6*b^5*c^7*d^3*f + 170*B*C^2*a^7*b^4*c^6*d^4*f + 160*B^2*C*a^5*b
^6*c^7*d^3*f - 160*B*C^2*a^2*b^9*c^5*d^5*f - 140*B^2*C*a^4*b^7*c^8*d^2*f -
136*B*C^2*a^3*b^8*c^2*d^8*f + 112*B^2*C*a^9*b^2*c^3*d^7*f + 91*B^2*C*a^2*b^
9*c^8*d^2*f + 88*B*C^2*a^4*b^7*c^7*d^3*f + 72*B^2*C*a^8*b^3*c^6*d^4*f - 64*
B^2*C*a^3*b^8*c^3*d^7*f - 60*B*C^2*a^3*b^8*c^6*d^4*f + 56*B*C^2*a^9*b^2*c^4
*d^6*f + 52*B*C^2*a^6*b^5*c^5*d^5*f + 48*B^2*C*a^9*b^2*c^5*d^5*f - 48*B^2*C
*a^7*b^4*c^7*d^3*f + 44*B*C^2*a^5*b^6*c^8*d^2*f - 36*B*C^2*a^9*b^2*c^6*d^4*
f + 12*B^2*C*a^6*b^5*c^8*d^2*f - 2958*A^2*C*a^4*b^7*c^4*d^6*f - 1932*A^2*C*
a^4*b^7*c^2*d^8*f + 1848*A^2*C*a^5*b^6*c^3*d^7*f + 1728*A^2*C*a^3*b^8*c^3*d
^7*f + 1524*A^2*C*a^5*b^6*c^5*d^5*f + 1374*A*C^2*a^4*b^7*c^4*d^6*f - 1272*A
*C^2*a^5*b^6*c^3*d^7*f - 1236*A*C^2*a^5*b^6*c^5*d^5*f + 1116*A*C^2*a^4*b^7*
c^2*d^8*f - 1110*A^2*C*a^6*b^5*c^4*d^6*f + 1038*A*C^2*a^6*b^5*c^4*d^6*f - 7
68*A^2*C*a^2*b^9*c^2*d^8*f - 696*A^2*C*a^7*b^4*c^3*d^7*f - 666*A*C^2*a^4*b^
7*c^6*d^4*f + 564*A^2*C*a^6*b^5*c^2*d^8*f - 564*A*C^2*a^7*b^4*c^5*d^5*f - 5
55*A*C^2*a^8*b^3*c^2*d^8*f + 519*A^2*C*a^8*b^3*c^2*d^8*f - 480*A*C^2*a^3*b^
8*c^3*d^7*f + 456*A*C^2*a^3*b^8*c^5*d^5*f - 420*A*C^2*a^2*b^9*c^6*d^4*f + 4
08*A*C^2*a^7*b^4*c^3*d^7*f + 408*A*C^2*a^2*b^9*c^2*d^8*f + 348*A^2*C*a^2*b^
9*c^6*d^4*f - 348*A*C^2*a^6*b^5*c^2*d^8*f + 342*A*C^2*a^6*b^5*c^6*d^4*f - 3
36*A*C^2*a^8*b^3*c^4*d^6*f + 324*A^2*C*a^7*b^4*c^5*d^5*f - 312*A^2*C*a^2*b^
9*c^4*d^6*f + 264*A^2*C*a^8*b^3*c^4*d^6*f + 240*A*C^2*a^5*b^6*c^7*d^3*f + 1
95*A*C^2*a^2*b^9*c^8*d^2*f - 174*A^2*C*a^6*b^5*c^6*d^4*f + 144*A*C^2*a^9*b^
2*c^3*d^7*f - 123*A^2*C*a^2*b^9*c^8*d^2*f + 120*A*C^2*a^3*b^8*c^7*d^3*f + 1
08*A*C^2*a^8*b^3*c^6*d^4*f - 102*A^2*C*a^4*b^7*c^6*d^4*f - 96*A^2*C*a^4*b^7
*c^8*d^2*f + 72*A^2*C*a^3*b^8*c^7*d^3*f + 72*A*C^2*a^9*b^2*c^5*d^5*f - 48*A
^2*C*a^9*b^2*c^3*d^7*f + 48*A^2*C*a^5*b^6*c^7*d^3*f - 48*A*C^2*a^2*b^9*c^4*
d^6*f - 24*A^2*C*a^3*b^8*c^5*d^5*f - 12*A*C^2*a^4*b^7*c^8*d^2*f + 2736*A^2*
B*a^6*b^5*c^3*d^7*f + 2464*A^2*B*a^3*b^8*c^4*d^6*f - 2298*A*B^2*a^4*b^7*c^4
*d^6*f - 2252*A^2*B*a^5*b^6*c^2*d^8*f - 1692*A^2*B*a^4*b^7*c^5*d^5*f - 1592
*A*B^2*a^4*b^7*c^2*d^8*f - 1338*A*B^2*a^6*b^5*c^4*d^6*f + 1320*A*B^2*a^5*b^
6*c^3*d^7*f + 1212*A*B^2*a^5*b^6*c^5*d^5*f - 1056*A*B^2*a^3*b^8*c^5*d^5*f +

1024*A^2*B*a^4*b^7*c^3*d^7*f - 1022*A^2*B*a^7*b^4*c^4*d^6*f - 880*A^2*B*a^2*b^9*c^5*d^5*f - 846*A^2*B*a^5*b^6*c^4*d^6*f - 840*A*B^2*a^7*b^4*c^3*d^7*f + 760*A*B^2*a^2*b^9*c^6*d^4*f - 704*A^2*B*a^2*b^9*c^3*d^7*f + 688*A*B^2*a^3*b^8*c^3*d^7*f + 660*A^2*B*a^3*b^8*c^6*d^4*f - 612*A^2*B*a^7*b^4*c^2*d^8*f + 462*A*B^2*a^4*b^7*c^6*d^4*f + 459*A*B^2*a^8*b^3*c^2*d^8*f - 412*A*B^2*a^2*b^9*c^2*d^8*f - 408*A*B^2*a^3*b^8*c^7*d^3*f + 388*A^2*B*a^6*b^5*c^5*d^5*f + 296*A^2*B*a^3*b^8*c^2*d^8*f + 288*A*B^2*a^6*b^5*c^2*d^8*f + 284*A*B^2*a^7*b^4*c^5*d^5*f + 236*A*B^2*a^8*b^3*c^4*d^6*f - 226*A*B^2*a^6*b^5*c^6*d^4*f + 212*A*B^2*a^2*b^9*c^4*d^6*f + 202*A^2*B*a^5*b^6*c^6*d^4*f - 152*A^2*B*a^4*b^7*c^7*d^3*f + 88*A^2*B*a^8*b^3*c^3*d^7*f + 79*A^2*B*a^9*b^2*c^2*d^8*f - 70*A^2*B*a^7*b^4*c^6*d^4*f + 68*A*B^2*a^4*b^7*c^8*d^2*f + 64*A^2*B*a^2*b^9*c^7*d^3*f - 64*A*B^2*a^9*b^2*c^3*d^7*f + 56*A^2*B*a^8*b^3*c^5*d^5*f + 56*A^2*B*a^6*b^5*c^7*d^3*f + 37*A^2*B*a^3*b^8*c^8*d^2*f - 28*A^2*B*a^9*b^2*c^4*d^6*f - 28*A^2*B*a^5*b^6*c^8*d^2*f + 17*A*B^2*a^2*b^9*c^8*d^2*f - 16*A*B^2*a^5*b^6*c^7*d^3*f + 48*A*B*C*b^11*c*d^9*f + 4*A*B*C*b^11*c^9*d*f + 24*A*B*C*a*b^10*d^10*f - 6*A*B*C*a*b^10*c^10*f + 432*B^2*C*a^7*b^4*c*d^9*f - 376*B*C^2*a*b^10*c^6*d^4*f - 354*B*C^2*a^8*b^3*c*d^9*f + 352*B^2*C*a*b^10*c^5*d^5*f + 320*B^2*C*a^5*b^6*c*d^9*f + 256*B^2*C*a*b^10*c^3*d^7*f - 232*B^2*C*a*b^10*c^7*d^3*f - 210*B^2*C*a^9*b^2*c*d^9*f - 152*B*C^2*a*b^10*c^4*d^6*f + 85*B*C^2*a*b^10*c^8*d^2*f + 72*B^2*C*a^3*b^8*c*d^9*f - 48*B*C^2*a^6*b^5*c*d^9*f - 40*B*C^2*a^10*b*c^3*d^7*f + 40*B*C^2*a*b^10*c^2*d^8*f + 37*B^2*C*a^10*b*c^2*d^8*f + 22*B^2*C*a^3*b^8*c^9*d*f - 18*B*C^2*a^2*b^9*c^9*d*f + 16*B*C^2*a^2*b^9*c*d^9*f - 12*B^2*C*a^10*b*c^4*d^6*f + 8*B*C^2*a^4*b^7*c^9*d*f + 8*B*C^2*a^4*b^7*c*d^9*f - 984*A^2*C*a^7*b^4*c*d^9*f + 672*A^2*C*a^3*b^8*c*d^9*f + 552*A*C^2*a^7*b^4*c*d^9*f - 504*A^2*C*a*b^10*c^5*d^5*f - 408*A^2*C*a^5*b^6*c*d^9*f + 408*A*C^2*a^5*b^6*c*d^9*f + 336*A*C^2*a*b^10*c^5*d^5*f - 216*A*C^2*a*b^10*c^7*d^3*f + 192*A*C^2*a*b^10*c^3*d^7*f - 162*A*C^2*a^9*b^2*c*d^9*f + 120*A^2*C*a*b^10*c^7*d^3*f + 96*A^2*C*a*b^10*c^3*d^7*f + 90*A^2*C*a^9*b^2*c*d^9*f + 66*A^2*C*a^3*b^8*c^9*d*f - 66*A*C^2*a^3*b^8*c^9*d*f + 57*A*C^2*a^10*b*c^2*d^8*f - 48*A*C^2*a^3*b^8*c*d^9*f - 9*A^2*C*a^10*b*c^2*d^8*f + 1736*A^2*B*a^4*b^7*c*d^9*f + 1248*A^2*B*a^6*b^5*c*d^9*f - 1008*A*B^2*a^7*b^4*c*d^9*f + 772*A^2*B*a*b^10*c^4*d^6*f - 688*A*B^2*a*b^10*c^5*d^5*f - 608*A*B^2*a^5*b^6*c*d^9*f + 436*A^2*B*a*b^10*c^2*d^8*f - 426*A^2*B*a^8*b^3*c*d^9*f + 312*A*B^2*a^3*b^8*c*d^9*f + 304*A^2*B*a^2*b^9*c*d^9*f - 244*A^2*B*a*b^10*c^6*d^4*f - 160*A*B^2*a*b^10*c^3*d^7*f + 114*A*B^2*a^9*b^2*c*d^9*f + 88*A*B^2*a*b^10*c^7*d^3*f - 22*A*B^2*a^3*b^8*c^9*d*f - 18*A^2*B*a^2*b^9*c^9*d*f + 13*A^2*B*a*b^10*c^8*d^2*f - 13*A*B^2*a^10*b*c^2*d^8*f + 8*A^2*B*a^10*b*c^3*d^7*f + 8*A^2*B*a^4*b^7*c^9*d*f + 112*B^2*C*b^11*c^6*d^4*f - 64*B*C^2*b^11*c^7*d^3*f + 16*B^2*C*b^11*c^4*d^6*f - 16*B^2*C*b^11*c^2*d^8*f + 16*B*C^2*b^11*c^5*d^5*f + 16*B*C^2*b^11*c^3*d^7*f - B^2*C*b^11*c^8*d^2*f + 96*A^2*C*b^11*c^4*d^6*f - 84*A^2*C*b^11*c^6*d^4*f + 72*A*C^2*b^11*c^6*d^4*f - 24*A*C^2*b^11*c^4*d^6*f - 24*A*C^2*b^11*c^2*d^8*f - 21*A*C^2*b^11*c^8*d^2*f + 12*A^2*C*b^11*c^2*d^8*f + 9*A^2*C*b^11*c^8*d^2*f - B*C^2*a^11*c^2*d^8*f + 176*A*B^2*b^11*c^4*d^6*f + 136*A^2*B*b^11*c^5*d^5*f - 128*A^2*B*b^11*c^3*d^7*f + 112*A*B^2*b^11*c^2*d^8*f + 111*B^2*C*a^8*b^3*d^10*f - 64*A*B^2*b^11 *c^6*d^4*f - 39*B*C^2*a^9*b^2*d^10*f + 24*B*C^2*a^7*b^4*d^10*f - 16*A^2*B*b^11*c^7*d^3*f - 4*B^2*C*a^2*b^9*d^10*f - 4*B*C^2*a^5*b^6*d^10*f + 432*A^2*C*a^6*b^5*d^10*f + 192*A^2*C*a^4*b^7*d^10*f - 111*A^2*C*a^8*b^3*d^10*f + 111*A*C^2*a^8*b^3*d^10*f - 72*A*C^2*a^6*b^5*d^10*f + 12*A*C^2*a^4*b^7*d^10*f - 3*B^2*C*a^2*b^9*c^10*f - A^2*B*a^11*c^2*d^8*f - B*C^2*a^3*b^8*c^10*f + 456*A^2*B*a^7*b^4*d^10*f - 288*A^2*B*a^3*b^8*d^10*f + 252*A*B^2*a^6*b^5*d^10*f + 192*A*B^2*a^4*b^7*d^10*f - 183*A*B^2*a^8*b^3*d^10*f - 148*A^2*B*a^5*b^6*d^10*f + 76*A*B^2*a^2*b^9*d^10*f - 9*A^2*C*a^2*b^9*c^10*f + 9*A*C^2*a^2*b^9*c^10*f - 3*A^2*B*a^9*b^2*d^10*f + 3*A*B^2*a^2*b^9*c^10*f - A^2*B*a^3*b^8*c^10*f - 2*C^3*a*b^10*c^9*d*f - 2*B^3*a^10*b*c*d^9*f - 264*A^3*a*b^10*c*d^9*f + 2*A^3*a*b^10*c^9*d*f - 2*B*C^2*b^11*c^9*d*f - 2*B^2*C*a^11*c*d^9*f - 120*A^2*B*b^11*c*d^9*f - 9*B^2*C*a^10*b*d^10*f - 6*A^2*C*a^11*c*d^9*f + 6*A*C^2*a^11*c*d^9*f - 2*A^2*B*b^11*c^9*d*f + 9*A^2*C*a^10*b*d^10*f - 9*A*C^2*a^10*b*d^10*f + 3*B*C^2*a*b^10*c^10*f + 2*A*B^2*a^11*c*d^9*f - 132*A^2*B*a*b^10*d^10*f - 3*A*B^2*a^10*b*d^10*f + 3*A^2*B*a*b^10*c^10*f + 520*C^3*a^5*b^6*c^3*d^7*f + 460*C^3*a^5*b^6*c^5*d^5*f - 418*C^3*a^6*b^5*c^4*d^6*f + 406*C^3*a^4*b^7*c^6*d^4*f + 268*C^3*a^7*b^4*c^5*d^5*f - 266*C^3*a^6*b^5*c^6*d^4*f + 233*C^3*a^8*b^3*c^2*d^8*f - 176*C^3*a^5*b^6*c^7*d^3*f + 164*C^3*a^2*b^9*c^6*d^4*f + 140*C^3*a^6*b^5*c^2*d^8*f + 136*C^3*a^2*b^9*c^4*d^6*f - 128*C^3*a^9*b^2*c^3*d^7*f + 128*C^3*a^3*b^8*c^3*d^7*f - 108*C^3*a^8*b^3*c^6*d^4*f - 104*C^3*a^3*b^8*c^7*d^3*f - 104*C^3*a^3*b^8*c^5*d^5*f + 100*C^3*a^8*b^3*c^4*d^6*f - 89*C^3*a^2*b^9*c^8*d^2*f - 72*C^3*a^9*b^2*c^5*d^5*f - 40*C^3*a^7*b^4*c^3*d^7*f + 40*C^3*a^4*b^7*c^8*d^2*f - 28*C^3*a^4*b^7*c^2*d^8*f - 16*C^3*a^2*b^9*c^2*d^8*f - 2*C^3*a^4*b^7*c^4*d^6*f + 828*B^3*a^4*b^7*c^5*d^5*f + 408*B^3*a^5*b^6*c^2*d^8*f + 390*B^3*a^7*b^4*c^4*d^6*f - 372*B^3*a^3*b^8*c^4*d^6*f - 336*B^3*a^6*b^5*c^3*d^7*f - 314*B^3*a^5*b^6*c^6*d^4*f + 288*B^3*a^4*b^7*c^3*d^7*f + 216*B^3*a^7*b^4*c^2*d^8*f - 176*B^3*a^2*b^9*c^7*d^3*f + 128*B^3*a^2*b^9*c^3*d^7*f + 108*B^3*a^6*b^5*c^5*d^5*f + 88*B^3*a^4*b^7*c^7*d^3*f + 72*B^3*a^2*b^9*c^5*d^5*f - 68*B^3*a^3*b^8*c^2*d^8*f - 65*B^3*a^9*b^2*c^2*d^8*f - 56*B^3*a^8*b^3*c^5*d^5*f + 40*B^3*a^6*b^5*c^7*d^3*f + 37*B^3*a^3*b^8*c^8*d^2*f + 30*B^3*a^5*b^6*c^4*d^6*f - 28*B^3*a^5*b^6*c^8*d^2*f + 24*B^3*a^8*b^3*c^3*d^7*f - 4*B^3*a^9*b^2*c^4*d^6*f - 2*B^3*a^7*b^4*c^6*d^4*f + 1586*A^3*a^4*b^7*c^4*d^6*f - 1376*A^3*a^3*b^8*c^3*d^7*f - 1096*A^3*a^5*b^6*c^3*d^7*f + 844*A^3*a^4*b^7*c^2*d^8*f - 748*A^3*a^5*b^6*c^5*d^5*f + 490*A^3*a^6*b^5*c^4*d^6*f + 376*A^3*a^2*b^9*c^2*d^8*f + 362*A^3*a^4*b^7*c^6*d^4*f - 356*A^3*a^6*b^5*c^2*d^8*f + 328*A^3*a^7*b^4*c^3*d^7*f - 328*A^3*a^3*b^8*c^5*d^5*f + 224*A^3*a^2*b^9*c^4*d^6*f - 197*A^3*a^8*b^3*c^2*d^8*f - 112*A^3*a^5*b^6*c^7*d^3*f + 98*A^3*a^6*b^5*c^6*d^4*f - 92*A^3*a^2*b^9*c^6*d^4*f - 88*A^3*a^3*b^8*c^7*d^3*f + 68*A^3*a^4*b^7*c^8*d^2*f + 32*A^3*a^9*b^2*c^3*d^7*f - 28*A^3*a^8*b^3*c^4*d^6*f - 28*A^3*a^7*b^4*c^5*d^5*f + 17*A^3*a^2*b^9*c^8*d^2*f + 104*C^3*a*b^10*c^7*d^3*f + 54*C^3*a^9*b^2*c*d^9*f - 40*C^3*a^7*b^4*c*d^9*f - 35*C^3*a^10*b*c^2*d^8*f + 22*C^3*a^3*b^8*c^9*d*f + 16*C^3*a*b^10*c^5*d^5*f - 16*C^3*a*b^10*c^3*d^7*f + 8*C^3*a^5*b^6*c*d^9*f - 2*A*B*C*a^11*d^10*f + 198*B^3*a^8*b^3*c*d^9*f + 192*B^3*a*b^10*c^6*d^4*f - 128*B^3*a^4*b^7*c*d^9*f - 80*B^3*a*b^10*c^2*d^8*f - 56*B^3*a^2*b^9*c*d^9*f - 24*B

^3*a^6*b^5*c*d^9*f - 18*B^3*a^2*b^9*c^9*d*f - 16*B^3*a*b^10*c^4*d^6*f + 13*
B^3*a*b^10*c^8*d^2*f + 8*B^3*a^10*b*c^3*d^7*f + 8*B^3*a^4*b^7*c^9*d*f - 624
*A^3*a^3*b^8*c*d^9*f + 472*A^3*a^7*b^4*c*d^9*f - 272*A^3*a*b^10*c^3*d^7*f +
152*A^3*a*b^10*c^5*d^5*f - 22*A^3*a^3*b^8*c^9*d*f + 18*A^3*a^9*b^2*c*d^9*f
- 13*A^3*a^10*b*c^2*d^8*f - 8*A^3*a^5*b^6*c*d^9*f - 8*A^3*a*b^10*c^7*d^3*f
+ A*B^2*b^11*c^8*d^2*f + 11*C^3*b^11*c^8*d^2*f - 8*C^3*b^11*c^6*d^4*f - 4*
C^3*b^11*c^4*d^6*f - 64*B^3*b^11*c^5*d^5*f - 32*B^3*b^11*c^3*d^7*f - 68*A^3
*b^11*c^4*d^6*f + 20*A^3*b^11*c^6*d^4*f + 12*A^3*b^11*c^2*d^8*f - C^3*a^8*b
^3*d^10*f - B^3*a^11*c^2*d^8*f - 60*B^3*a^7*b^4*d^10*f - 32*B^3*a^5*b^6*d^1
0*f + 21*B^3*a^9*b^2*d^10*f - 12*B^3*a^3*b^8*d^10*f - 3*C^3*a^2*b^9*c^10*f
- 360*A^3*a^6*b^5*d^10*f - 204*A^3*a^4*b^7*d^10*f - B^3*a^3*b^8*c^10*f + 3*
A^3*a^2*b^9*c^10*f - 2*C^3*a^11*c*d^9*f - 2*B^3*b^11*c^9*d*f + 3*C^3*a^10*b
*d^10*f + 2*A^3*a^11*c*d^9*f + 3*B^3*a*b^10*c^10*f - 3*A^3*a^10*b*d^10*f -
36*A^2*C*b^11*d^10*f + 3*A^2*C*b^11*c^10*f - 3*A*C^2*b^11*c^10*f - A*B^2*b^
11*c^10*f + 36*A^3*b^11*d^10*f - A^3*b^11*c^10*f + A^3*b^11*c^8*d^2*f + A^3
*a^8*b^3*d^10*f + B^2*C*b^11*c^10*f + B*C^2*a^11*d^10*f + A^2*B*a^11*d^10*f
+ C^3*b^11*c^10*f + B^3*a^11*d^10*f - 6*A*B^2*C*a^7*b*c*d^7 + 4*A*B^2*C*a*
b^7*c*d^7 + 168*A^2*B*C*a^2*b^6*c^3*d^5 + 144*A*B*C^2*a^3*b^5*c^4*d^4 - 129
*A^2*B*C*a^3*b^5*c^4*d^4 - 96*A*B*C^2*a^2*b^6*c^3*d^5 + 84*A*B*C^2*a^3*b^5*
c^2*d^6 + 72*A^2*B*C*a^4*b^4*c^3*d^5 - 72*A^2*B*C*a^3*b^5*c^2*d^6 + 64*A*B^
2*C*a^4*b^4*c^4*d^4 - 60*A*B*C^2*a^4*b^4*c^3*d^5 + 57*A^2*B*C*a^5*b^3*c^2*d
^6 - 56*A*B^2*C*a^5*b^3*c^3*d^5 - 39*A*B^2*C*a^2*b^6*c^4*d^4 - 38*A*B^2*C*a
^3*b^5*c^5*d^3 + 36*A*B^2*C*a^3*b^5*c^3*d^5 + 36*A*B*C^2*a^5*b^3*c^4*d^4 -
30*A*B*C^2*a^5*b^3*c^2*d^6 + 27*A*B^2*C*a^6*b^2*c^2*d^6 - 24*A*B^2*C*a^2*b^
6*c^2*d^6 + 24*A*B*C^2*a^6*b^2*c^3*d^5 - 24*A*B*C^2*a^4*b^4*c^5*d^3 - 18*A^
2*B*C*a^5*b^3*c^4*d^4 + 18*A^2*B*C*a^2*b^6*c^5*d^3 - 15*A*B^2*C*a^4*b^4*c^2
*d^6 - 12*A^2*B*C*a^6*b^2*c^3*d^5 + 12*A^2*B*C*a^4*b^4*c^5*d^3 + 9*A*B^2*C*
a^2*b^6*c^6*d^2 + 6*A*B*C^2*a^3*b^5*c^6*d^2 - 3*A^2*B*C*a^3*b^5*c^6*d^2 + 6
0*A^2*B*C*a^2*b^6*c*d^7 - 51*A^2*B*C*a*b^7*c^4*d^4 + 48*A*B*C^2*a^6*b^2*c*d
^7 - 42*A^2*B*C*a^6*b^2*c*d^7 - 42*A^2*B*C*a*b^7*c^2*d^6 + 36*A*B*C^2*a^4*b
^4*c*d^7 + 36*A*B*C^2*a*b^7*c^4*d^4 + 36*A*B*C^2*a*b^7*c^2*d^6 - 30*A^2*B*C
*a^4*b^4*c*d^7 + 24*A*B^2*C*a^3*b^5*c*d^7 - 24*A*B*C^2*a^2*b^6*c*d^7 + 18*A
*B^2*C*a*b^7*c^5*d^3 - 18*A*B*C^2*a*b^7*c^6*d^2 + 12*A*B^2*C*a*b^7*c^3*d^5
+ 9*A^2*B*C*a*b^7*c^6*d^2 + 6*A*B^2*C*a^5*b^3*c*d^7 - 6*A*B*C^2*a^7*b*c^2*d
^6 + 3*A^2*B*C*a^7*b*c^2*d^6 - 18*B^3*C*a^6*b^2*c*d^7 - 18*B*C^3*a^6*b^2*c*
d^7 - 14*B^3*C*a^4*b^4*c*d^7 - 14*B*C^3*a^4*b^4*c*d^7 - 10*B^3*C*a*b^7*c^2*
d^6 - 10*B*C^3*a*b^7*c^2*d^6 + 9*B^3*C*a*b^7*c^6*d^2 + 9*B*C^3*a*b^7*c^6*d^
2 - 7*B^3*C*a*b^7*c^4*d^4 - 7*B*C^3*a*b^7*c^4*d^4 + 6*B^2*C^2*a^7*b*c*d^7 -
4*B^3*C*a^2*b^6*c*d^7 + 4*B^2*C^2*a*b^7*c*d^7 - 4*B*C^3*a^2*b^6*c*d^7 + 3*
B^3*C*a^7*b*c^2*d^6 + 3*B*C^3*a^7*b*c^2*d^6 + 144*A^3*C*a^3*b^5*c*d^7 + 62*
A^3*C*a^5*b^3*c*d^7 + 48*A*C^3*a^3*b^5*c*d^7 - 36*A^2*C^2*a*b^7*c*d^7 + 26*
A*C^3*a^5*b^3*c*d^7 + 20*A^3*C*a*b^7*c^3*d^5 + 18*A^2*C^2*a^7*b*c*d^7 - 18*
A*C^3*a*b^7*c^5*d^3 - 6*A^3*C*a*b^7*c^5*d^3 - 4*A*C^3*a*b^7*c^3*d^5 - 32*A^
3*B*a^2*b^6*c*d^7 - 32*A*B^3*a^2*b^6*c*d^7 + 22*A^3*B*a*b^7*c^4*d^4 + 22*A*
B^3*a*b^7*c^4*d^4 + 16*A^3*B*a*b^7*c^2*d^6 + 16*A*B^3*a*b^7*c^2*d^6 + 12*A^

3*B*a^6*b^2*c*d^7 + 12*A*B^3*a^6*b^2*c*d^7 + 8*A^3*B*a^4*b^4*c*d^7 - 8*A^2*
B^2*a*b^7*c*d^7 + 8*A*B^3*a^4*b^4*c*d^7 + 36*A^2*B*C*b^8*c^3*d^5 + 24*A*B*C
^2*b^8*c^5*d^3 - 18*A^2*B*C*b^8*c^5*d^3 - 12*A*B*C^2*b^8*c^3*d^5 - 3*A*B^2*
C*b^8*c^6*d^2 - 3*A*B^2*C*b^8*c^4*d^4 - 2*A*B^2*C*b^8*c^2*d^6 + 57*A^2*B*C*
a^5*b^3*d^8 + 36*A^2*B*C*a^3*b^5*d^8 - 30*A*B*C^2*a^5*b^3*d^8 - 18*A*B*C^2*
a^3*b^5*d^8 - 9*A*B^2*C*a^4*b^4*d^8 - 3*A*B^2*C*a^6*b^2*d^8 - 2*A*B^2*C*a^2
*b^6*d^8 + 34*B^2*C^2*a^3*b^5*c^5*d^3 + 28*B^2*C^2*a^5*b^3*c^3*d^5 + 24*B^2
*C^2*a^2*b^6*c^4*d^4 - 20*B^2*C^2*a^4*b^4*c^4*d^4 + 12*B^2*C^2*a^3*b^5*c^3*
d^5 + 12*B^2*C^2*a^2*b^6*c^2*d^6 + 9*B^2*C^2*a^6*b^2*c^4*d^4 + 9*B^2*C^2*a^
4*b^4*c^2*d^6 - 9*B^2*C^2*a^2*b^6*c^6*d^2 - 3*B^2*C^2*a^6*b^2*c^2*d^6 + 159
*A^2*C^2*a^4*b^4*c^2*d^6 - 156*A^2*C^2*a^3*b^5*c^3*d^5 + 90*A^2*C^2*a^3*b^5
*c^5*d^3 + 78*A^2*C^2*a^2*b^6*c^2*d^6 - 63*A^2*C^2*a^4*b^4*c^4*d^4 - 27*A^2
*C^2*a^6*b^2*c^2*d^6 - 27*A^2*C^2*a^2*b^6*c^6*d^2 - 18*A^2*C^2*a^2*b^6*c^4*
d^4 + 9*A^2*C^2*a^6*b^2*c^4*d^4 + 66*A^2*B^2*a^2*b^6*c^2*d^6 + 60*A^2*B^2*a
^4*b^4*c^2*d^6 - 48*A^2*B^2*a^3*b^5*c^3*d^5 + 42*A^2*B^2*a^2*b^6*c^4*d^4 +
28*A^2*B^2*a^5*b^3*c^3*d^5 - 17*A^2*B^2*a^4*b^4*c^4*d^4 - 6*A^2*B^2*a^6*b^2
*c^2*d^6 + 4*A^2*B^2*a^3*b^5*c^5*d^3 + 36*A^3*C*a*b^7*c*d^7 - 18*A*C^3*a^7*
b*c*d^7 + 12*A*C^3*a*b^7*c*d^7 - 6*A^3*C*a^7*b*c*d^7 + 24*A^2*B*C*b^8*c*d^7
 - 12*A*B*C^2*b^8*c*d^7 + 12*A^2*B*C*a*b^7*d^8 + 6*A*B*C^2*a^7*b*d^8 - 6*A*
B*C^2*a*b^7*d^8 - 3*A^2*B*C*a^7*b*d^8 - 53*B^3*C*a^3*b^5*c^4*d^4 - 53*B*C^3
*a^3*b^5*c^4*d^4 - 32*B^3*C*a^3*b^5*c^2*d^6 - 32*B*C^3*a^3*b^5*c^2*d^6 - 18
*B^3*C*a^5*b^3*c^4*d^4 - 18*B*C^3*a^5*b^3*c^4*d^4 + 16*B^3*C*a^4*b^4*c^3*d^
5 + 16*B*C^3*a^4*b^4*c^3*d^5 - 12*B^3*C*a^6*b^2*c^3*d^5 + 12*B^3*C*a^4*b^4*
c^5*d^3 + 12*B^2*C^2*a^3*b^5*c*d^7 - 12*B*C^3*a^6*b^2*c^3*d^5 + 12*B*C^3*a^
4*b^4*c^5*d^3 + 8*B^3*C*a^2*b^6*c^3*d^5 + 8*B*C^3*a^2*b^6*c^3*d^5 - 6*B^3*C
*a^2*b^6*c^5*d^3 + 6*B^2*C^2*a^5*b^3*c*d^7 - 6*B^2*C^2*a*b^7*c^5*d^3 - 6*B*
C^3*a^2*b^6*c^5*d^3 - 3*B^3*C*a^3*b^5*c^6*d^2 - 3*B*C^3*a^3*b^5*c^6*d^2 - 1
75*A^3*C*a^4*b^4*c^2*d^6 + 164*A^3*C*a^3*b^5*c^3*d^5 - 144*A^2*C^2*a^3*b^5*
c*d^7 - 124*A^3*C*a^2*b^6*c^2*d^6 - 90*A*C^3*a^3*b^5*c^5*d^3 - 73*A*C^3*a^4
*b^4*c^2*d^6 - 66*A^2*C^2*a^5*b^3*c*d^7 + 44*A*C^3*a^3*b^5*c^3*d^5 + 36*A*C
^3*a^4*b^4*c^4*d^4 + 30*A^3*C*a^4*b^4*c^4*d^4 - 30*A^3*C*a^3*b^5*c^5*d^3 +
27*A*C^3*a^2*b^6*c^6*d^2 + 21*A*C^3*a^2*b^6*c^4*d^4 + 18*A^2*C^2*a*b^7*c^5*
d^3 - 18*A*C^3*a^6*b^2*c^4*d^4 - 16*A*C^3*a^2*b^6*c^2*d^6 + 15*A^3*C*a^6*b^
2*c^2*d^6 - 15*A^3*C*a^2*b^6*c^4*d^4 - 12*A^2*C^2*a*b^7*c^3*d^5 + 9*A^3*C*a
^2*b^6*c^6*d^2 + 9*A*C^3*a^6*b^2*c^2*d^6 - 80*A^3*B*a^2*b^6*c^3*d^5 - 80*A*
B^3*a^2*b^6*c^3*d^5 + 38*A^3*B*a^3*b^5*c^4*d^4 + 38*A*B^3*a^3*b^5*c^4*d^4 -
 36*A^2*B^2*a^3*b^5*c*d^7 - 28*A^3*B*a^5*b^3*c^2*d^6 - 28*A^3*B*a^4*b^4*c^3
*d^5 - 28*A*B^3*a^5*b^3*c^2*d^6 - 28*A*B^3*a^4*b^4*c^3*d^5 + 20*A^3*B*a^3*b
^5*c^2*d^6 + 20*A*B^3*a^3*b^5*c^2*d^6 - 12*A^3*B*a^2*b^6*c^5*d^3 - 12*A^2*B
^2*a^5*b^3*c*d^7 - 12*A^2*B^2*a*b^7*c^5*d^3 - 12*A^2*B^2*a*b^7*c^3*d^5 - 12
*A*B^3*a^2*b^6*c^5*d^3 + 9*B^2*C^2*b^8*c^4*d^4 + 4*B^2*C^2*b^8*c^2*d^6 + 3*
B^2*C^2*b^8*c^6*d^2 - 30*A^2*C^2*b^8*c^4*d^4 + 9*A^2*C^2*b^8*c^6*d^2 + 16*A
^2*B^2*b^8*c^2*d^6 + 6*B^2*C^2*a^6*b^2*d^8 + 3*B^2*C^2*a^4*b^4*d^8 + 3*A^2*
B^2*b^8*c^4*d^4 + 36*A^2*C^2*a^4*b^4*d^8 + 27*A^2*C^2*a^2*b^6*d^8 - 18*A^2*
C^2*a^6*b^2*d^8 + 33*A^2*B^2*a^4*b^4*d^8 + 28*A^2*B^2*a^2*b^6*d^8 + 6*A^2*B

^2*a^6*b^2*d^8 + 6*C^4*a*b^7*c^5*d^3 + 4*C^4*a*b^7*c^3*d^5 - 2*C^4*a^5*b^3*
c*d^7 + 12*B^4*a^3*b^5*c*d^7 - 12*B^4*a*b^7*c^5*d^3 + 8*B^4*a^5*b^3*c*d^7 -
4*B^4*a*b^7*c^3*d^5 - 48*A^4*a^3*b^5*c*d^7 - 20*A^4*a^5*b^3*c*d^7 - 8*A^4*
a*b^7*c^3*d^5 - 10*B^3*C*b^8*c^5*d^3 - 10*B*C^3*b^8*c^5*d^3 - 4*B^3*C*b^8*c
^3*d^5 - 4*B*C^3*b^8*c^3*d^5 + 23*A^3*C*b^8*c^4*d^4 - 18*A^3*C*b^8*c^2*d^6
+ 11*A*C^3*b^8*c^4*d^4 - 9*A*C^3*b^8*c^6*d^2 + 6*A*C^3*b^8*c^2*d^6 - 3*A^3*
C*b^8*c^6*d^2 - 20*A^3*B*b^8*c^3*d^5 - 20*A*B^3*b^8*c^3*d^5 + 4*A^3*B*b^8*c
^5*d^3 + 4*A*B^3*b^8*c^5*d^3 - 63*A^3*C*a^4*b^4*d^8 - 54*A^3*C*a^2*b^6*d^8
+ 9*A^3*C*a^6*b^2*d^8 + 9*A*C^3*a^6*b^2*d^8 - 3*A*C^3*a^4*b^4*d^8 - 28*A^3*
B*a^5*b^3*d^8 - 28*A*B^3*a^5*b^3*d^8 - 18*A^3*B*a^3*b^5*d^8 - 18*A*B^3*a^3*
b^5*d^8 + B^3*C*a^5*b^3*c^2*d^6 + B*C^3*a^5*b^3*c^2*d^6 + 6*C^4*a^7*b*c*d^7
+ 4*B^4*a*b^7*c*d^7 - 12*A^4*a*b^7*c*d^7 - 12*A^3*B*b^8*c*d^7 - 12*A*B^3*b
^8*c*d^7 - 3*B^3*C*a^7*b*d^8 - 3*B*C^3*a^7*b*d^8 - 6*A^3*B*a*b^7*d^8 - 6*A*
B^3*a*b^7*d^8 + 30*C^4*a^3*b^5*c^5*d^3 + 19*C^4*a^4*b^4*c^2*d^6 + 9*C^4*a^6
*b^2*c^4*d^4 - 9*C^4*a^2*b^6*c^6*d^2 + 4*C^4*a^3*b^5*c^3*d^5 + 4*C^4*a^2*b^
6*c^2*d^6 + 3*C^4*a^6*b^2*c^2*d^6 - 3*C^4*a^4*b^4*c^4*d^4 - 3*C^4*a^2*b^6*c
^4*d^4 + 28*B^4*a^5*b^3*c^3*d^5 + 27*B^4*a^2*b^6*c^4*d^4 - 17*B^4*a^4*b^4*c
^4*d^4 - 10*B^4*a^4*b^4*c^2*d^6 + 8*B^4*a^3*b^5*c^3*d^5 + 8*B^4*a^2*b^6*c^2
*d^6 - 6*B^4*a^6*b^2*c^2*d^6 + 4*B^4*a^3*b^5*c^5*d^3 + 70*A^4*a^4*b^4*c^2*d
^6 + 58*A^4*a^2*b^6*c^2*d^6 - 56*A^4*a^3*b^5*c^3*d^5 + 15*A^4*a^2*b^6*c^4*d
^4 + B^2*C^2*a^2*b^6*d^8 - 18*A^3*C*b^8*d^8 + B^3*C*a^5*b^3*d^8 + B*C^3*a^5
*b^3*d^8 + 3*C^4*b^8*c^6*d^2 + 8*B^4*b^8*c^4*d^4 + 4*B^4*b^8*c^2*d^6 + 12*A
^4*b^8*c^2*d^6 - 5*A^4*b^8*c^4*d^4 + 6*B^4*a^6*b^2*d^8 + 3*B^4*a^4*b^4*d^8
+ 30*A^4*a^4*b^4*d^8 + 27*A^4*a^2*b^6*d^8 + 9*A^2*C^2*b^8*d^8 + 9*A^2*B^2*b
^8*d^8 + 9*A^4*b^8*d^8 + C^4*b^8*c^4*d^4 + B^4*a^2*b^6*d^8, f, k)*(root(640
*a^13*b^7*c*d^15*f^4 + 640*a^7*b^13*c^15*d*f^4 + 480*a^15*b^5*c*d^15*f^4 +
480*a^11*b^9*c*d^15*f^4 + 480*a^9*b^11*c^15*d*f^4 + 480*a^5*b^15*c^15*d*f^4
+ 192*a^19*b*c^5*d^11*f^4 + 192*a^17*b^3*c*d^15*f^4 + 192*a^11*b^9*c^15*d*
f^4 + 192*a^9*b^11*c*d^15*f^4 + 192*a^3*b^17*c^15*d*f^4 + 192*a*b^19*c^11*d
^5*f^4 + 128*a^19*b*c^7*d^9*f^4 + 128*a^19*b*c^3*d^13*f^4 + 128*a*b^19*c^13
*d^3*f^4 + 128*a*b^19*c^9*d^7*f^4 + 32*a^19*b*c^9*d^7*f^4 + 32*a^13*b^7*c^1
5*d*f^4 + 32*a^7*b^13*c*d^15*f^4 + 32*a*b^19*c^7*d^9*f^4 + 32*a^19*b*c*d^15
*f^4 + 32*a*b^19*c^15*d*f^4 - 47088*a^10*b^10*c^8*d^8*f^4 + 42432*a^11*b^9*
c^7*d^9*f^4 + 42432*a^9*b^11*c^9*d^7*f^4 + 39328*a^11*b^9*c^9*d^7*f^4 + 393
28*a^9*b^11*c^7*d^9*f^4 - 36912*a^12*b^8*c^8*d^8*f^4 - 36912*a^8*b^12*c^8*d
^8*f^4 - 34256*a^10*b^10*c^10*d^6*f^4 - 34256*a^10*b^10*c^6*d^10*f^4 - 3115
2*a^12*b^8*c^6*d^10*f^4 - 31152*a^8*b^12*c^10*d^6*f^4 + 28128*a^13*b^7*c^7*
d^9*f^4 + 28128*a^7*b^13*c^9*d^7*f^4 + 24160*a^11*b^9*c^5*d^11*f^4 + 24160*
a^9*b^11*c^11*d^5*f^4 - 23088*a^12*b^8*c^10*d^6*f^4 - 23088*a^8*b^12*c^6*d^
10*f^4 + 22272*a^13*b^7*c^9*d^7*f^4 + 22272*a^7*b^13*c^7*d^9*f^4 + 19072*a^
11*b^9*c^11*d^5*f^4 + 19072*a^9*b^11*c^5*d^11*f^4 + 18624*a^13*b^7*c^5*d^11
*f^4 + 18624*a^7*b^13*c^11*d^5*f^4 - 17328*a^14*b^6*c^8*d^8*f^4 - 17328*a^6
*b^14*c^8*d^8*f^4 - 17232*a^14*b^6*c^6*d^10*f^4 - 17232*a^6*b^14*c^10*d^6*f
^4 - 13520*a^12*b^8*c^4*d^12*f^4 - 13520*a^8*b^12*c^12*d^4*f^4 - 12464*a^10
*b^10*c^12*d^4*f^4 - 12464*a^10*b^10*c^4*d^12*f^4 + 10880*a^15*b^5*c^7*d^9*

f^4 + 10880*a^5*b^15*c^9*d^7*f^4 - 9072*a^14*b^6*c^10*d^6*f^4 - 9072*a^6*b^
14*c^6*d^10*f^4 + 8928*a^13*b^7*c^11*d^5*f^4 + 8928*a^7*b^13*c^5*d^11*f^4 -
8880*a^14*b^6*c^4*d^12*f^4 - 8880*a^6*b^14*c^12*d^4*f^4 + 8480*a^15*b^5*c^
5*d^11*f^4 + 8480*a^5*b^15*c^11*d^5*f^4 + 7200*a^15*b^5*c^9*d^7*f^4 + 7200*
a^5*b^15*c^7*d^9*f^4 - 6912*a^12*b^8*c^12*d^4*f^4 - 6912*a^8*b^12*c^4*d^12*
f^4 + 6400*a^11*b^9*c^3*d^13*f^4 + 6400*a^9*b^11*c^13*d^3*f^4 + 5920*a^13*b
^7*c^3*d^13*f^4 + 5920*a^7*b^13*c^13*d^3*f^4 - 5392*a^16*b^4*c^6*d^10*f^4 -
5392*a^4*b^16*c^10*d^6*f^4 - 4428*a^16*b^4*c^8*d^8*f^4 - 4428*a^4*b^16*c^8
*d^8*f^4 + 4128*a^11*b^9*c^13*d^3*f^4 + 4128*a^9*b^11*c^3*d^13*f^4 - 3328*a
^16*b^4*c^4*d^12*f^4 - 3328*a^4*b^16*c^12*d^4*f^4 + 3264*a^15*b^5*c^3*d^13*
f^4 + 3264*a^5*b^15*c^13*d^3*f^4 - 2480*a^12*b^8*c^2*d^14*f^4 - 2480*a^8*b^
12*c^14*d^2*f^4 + 2240*a^15*b^5*c^11*d^5*f^4 + 2240*a^5*b^15*c^5*d^11*f^4 -
2128*a^14*b^6*c^12*d^4*f^4 - 2128*a^6*b^14*c^4*d^12*f^4 + 2112*a^17*b^3*c^
7*d^9*f^4 + 2112*a^3*b^17*c^9*d^7*f^4 + 2048*a^17*b^3*c^5*d^11*f^4 + 2048*a
^3*b^17*c^11*d^5*f^4 - 2000*a^14*b^6*c^2*d^14*f^4 - 2000*a^6*b^14*c^14*d^2*
f^4 - 1792*a^16*b^4*c^10*d^6*f^4 - 1792*a^4*b^16*c^6*d^10*f^4 - 1776*a^10*b
^10*c^14*d^2*f^4 - 1776*a^10*b^10*c^2*d^14*f^4 + 1472*a^13*b^7*c^13*d^3*f^4
+ 1472*a^7*b^13*c^3*d^13*f^4 + 1088*a^17*b^3*c^9*d^7*f^4 + 1088*a^3*b^17*c
^7*d^9*f^4 + 992*a^17*b^3*c^3*d^13*f^4 + 992*a^3*b^17*c^13*d^3*f^4 - 912*a^
16*b^4*c^2*d^14*f^4 - 912*a^4*b^16*c^14*d^2*f^4 - 768*a^18*b^2*c^6*d^10*f^4
- 768*a^2*b^18*c^10*d^6*f^4 - 688*a^12*b^8*c^14*d^2*f^4 - 688*a^8*b^12*c^2
*d^14*f^4 - 592*a^18*b^2*c^4*d^12*f^4 - 592*a^2*b^18*c^12*d^4*f^4 - 472*a^1
8*b^2*c^8*d^8*f^4 - 472*a^2*b^18*c^8*d^8*f^4 - 280*a^16*b^4*c^12*d^4*f^4 -
280*a^4*b^16*c^4*d^12*f^4 + 224*a^17*b^3*c^11*d^5*f^4 + 224*a^15*b^5*c^13*d
^3*f^4 + 224*a^5*b^15*c^3*d^13*f^4 + 224*a^3*b^17*c^5*d^11*f^4 - 208*a^18*b
^2*c^2*d^14*f^4 - 208*a^2*b^18*c^14*d^2*f^4 - 112*a^18*b^2*c^10*d^6*f^4 - 1
12*a^14*b^6*c^14*d^2*f^4 - 112*a^6*b^14*c^2*d^14*f^4 - 112*a^2*b^18*c^6*d^1
0*f^4 - 24*b^20*c^12*d^4*f^4 - 16*b^20*c^14*d^2*f^4 - 16*b^20*c^10*d^6*f^4
- 4*b^20*c^8*d^8*f^4 - 24*a^20*c^4*d^12*f^4 - 16*a^20*c^6*d^10*f^4 - 16*a^2
0*c^2*d^14*f^4 - 4*a^20*c^8*d^8*f^4 - 80*a^14*b^6*d^16*f^4 - 60*a^16*b^4*d^
16*f^4 - 60*a^12*b^8*d^16*f^4 - 24*a^18*b^2*d^16*f^4 - 24*a^10*b^10*d^16*f^
4 - 4*a^8*b^12*d^16*f^4 - 80*a^6*b^14*c^16*f^4 - 60*a^8*b^12*c^16*f^4 - 60*
a^4*b^16*c^16*f^4 - 24*a^10*b^10*c^16*f^4 - 24*a^2*b^18*c^16*f^4 - 4*a^12*b
^8*c^16*f^4 - 4*b^20*c^16*f^4 - 4*a^20*d^16*f^4 + 56*A*C*a^13*b*c*d^11*f^2
- 48*A*C*a*b^13*c^11*d*f^2 + 48*A*C*a*b^13*c*d^11*f^2 + 5904*B*C*a^7*b^7*c^
6*d^6*f^2 - 5016*B*C*a^8*b^6*c^5*d^7*f^2 - 4608*B*C*a^6*b^8*c^7*d^5*f^2 - 4
512*B*C*a^6*b^8*c^5*d^7*f^2 - 4384*B*C*a^8*b^6*c^7*d^5*f^2 + 3056*B*C*a^7*b
^7*c^8*d^4*f^2 + 2256*B*C*a^7*b^7*c^4*d^8*f^2 - 1824*B*C*a^8*b^6*c^3*d^9*f^
2 + 1632*B*C*a^4*b^10*c^9*d^3*f^2 - 1400*B*C*a^3*b^11*c^8*d^4*f^2 - 1320*B*
C*a^11*b^3*c^4*d^8*f^2 - 1248*B*C*a^6*b^8*c^3*d^9*f^2 + 1152*B*C*a^10*b^4*c
^3*d^9*f^2 - 1072*B*C*a^6*b^8*c^9*d^3*f^2 + 1068*B*C*a^9*b^5*c^6*d^6*f^2 -
1004*B*C*a^5*b^9*c^4*d^8*f^2 - 968*B*C*a^3*b^11*c^6*d^6*f^2 - 864*B*C*a^5*b
^9*c^8*d^4*f^2 - 828*B*C*a^9*b^5*c^4*d^8*f^2 - 792*B*C*a^11*b^3*c^2*d^10*f^
2 - 792*B*C*a^3*b^11*c^4*d^8*f^2 - 776*B*C*a^8*b^6*c^9*d^3*f^2 + 688*B*C*a^
4*b^10*c^7*d^5*f^2 - 672*B*C*a^3*b^11*c^10*d^2*f^2 - 592*B*C*a^9*b^5*c^2*d^

10*f^2 + 544*B*C*a^7*b^7*c^10*d^2*f^2 - 492*B*C*a^5*b^9*c^2*d^10*f^2 + 480*
B*C*a^10*b^4*c^5*d^7*f^2 - 392*B*C*a^5*b^9*c^10*d^2*f^2 + 332*B*C*a^9*b^5*c
^8*d^4*f^2 - 328*B*C*a^11*b^3*c^6*d^6*f^2 + 320*B*C*a^2*b^12*c^9*d^3*f^2 +
272*B*C*a^12*b^2*c^3*d^9*f^2 - 248*B*C*a^4*b^10*c^5*d^7*f^2 - 248*B*C*a^3*b
^11*c^2*d^10*f^2 - 208*B*C*a^10*b^4*c^7*d^5*f^2 - 192*B*C*a^2*b^12*c^5*d^7*
f^2 + 144*B*C*a^7*b^7*c^2*d^10*f^2 - 96*B*C*a^4*b^10*c^3*d^9*f^2 + 88*B*C*a
^12*b^2*c^5*d^7*f^2 - 72*B*C*a^11*b^3*c^8*d^4*f^2 - 48*B*C*a^12*b^2*c^7*d^5
*f^2 + 48*B*C*a^10*b^4*c^9*d^3*f^2 - 48*B*C*a^2*b^12*c^7*d^5*f^2 - 48*B*C*a
^2*b^12*c^3*d^9*f^2 - 12*B*C*a^9*b^5*c^10*d^2*f^2 + 4*B*C*a^5*b^9*c^6*d^6*f
^2 + 5824*A*C*a^5*b^9*c^7*d^5*f^2 - 4378*A*C*a^6*b^8*c^8*d^4*f^2 + 4296*A*C
*a^5*b^9*c^5*d^7*f^2 - 3912*A*C*a^6*b^8*c^6*d^6*f^2 - 3672*A*C*a^9*b^5*c^5*
d^7*f^2 + 3594*A*C*a^8*b^6*c^4*d^8*f^2 + 3236*A*C*a^8*b^6*c^6*d^6*f^2 + 281
6*A*C*a^5*b^9*c^9*d^3*f^2 + 2624*A*C*a^5*b^9*c^3*d^9*f^2 + 2432*A*C*a^7*b^7
*c^7*d^5*f^2 - 2366*A*C*a^4*b^10*c^8*d^4*f^2 + 2298*A*C*a^10*b^4*c^4*d^8*f^
2 + 1872*A*C*a^7*b^7*c^3*d^9*f^2 + 1848*A*C*a^10*b^4*c^6*d^6*f^2 - 1644*A*C
*a^4*b^10*c^6*d^6*f^2 - 1488*A*C*a^9*b^5*c^7*d^5*f^2 - 1408*A*C*a^9*b^5*c^3
*d^9*f^2 - 1308*A*C*a^6*b^8*c^4*d^8*f^2 + 1248*A*C*a^7*b^7*c^5*d^7*f^2 - 10
12*A*C*a^6*b^8*c^10*d^2*f^2 + 1008*A*C*a^3*b^11*c^7*d^5*f^2 + 992*A*C*a^3*b
^11*c^5*d^7*f^2 + 928*A*C*a^3*b^11*c^3*d^9*f^2 + 848*A*C*a^7*b^7*c^9*d^3*f^
2 + 636*A*C*a^8*b^6*c^2*d^10*f^2 - 628*A*C*a^4*b^10*c^10*d^2*f^2 - 600*A*C*
a^6*b^8*c^2*d^10*f^2 - 576*A*C*a^11*b^3*c^5*d^7*f^2 + 572*A*C*a^10*b^4*c^2*
d^10*f^2 + 464*A*C*a^8*b^6*c^8*d^4*f^2 - 304*A*C*a^4*b^10*c^4*d^8*f^2 + 304
*A*C*a^2*b^12*c^6*d^6*f^2 + 296*A*C*a^2*b^12*c^4*d^8*f^2 + 260*A*C*a^10*b^4
*c^8*d^4*f^2 - 232*A*C*a^12*b^2*c^2*d^10*f^2 - 232*A*C*a^9*b^5*c^9*d^3*f^2
+ 228*A*C*a^2*b^12*c^10*d^2*f^2 - 188*A*C*a^4*b^10*c^2*d^10*f^2 + 144*A*C*a
^11*b^3*c^3*d^9*f^2 + 116*A*C*a^12*b^2*c^6*d^6*f^2 - 112*A*C*a^11*b^3*c^7*d
^5*f^2 + 112*A*C*a^3*b^11*c^9*d^3*f^2 + 92*A*C*a^8*b^6*c^10*d^2*f^2 + 74*A*
C*a^12*b^2*c^4*d^8*f^2 + 62*A*C*a^2*b^12*c^8*d^4*f^2 + 40*A*C*a^2*b^12*c^2*
d^10*f^2 - 7008*A*B*a^7*b^7*c^6*d^6*f^2 - 4032*A*B*a^7*b^7*c^4*d^8*f^2 + 39
52*A*B*a^8*b^6*c^7*d^5*f^2 + 3648*A*B*a^8*b^6*c^5*d^7*f^2 - 3392*A*B*a^7*b^
7*c^8*d^4*f^2 + 3264*A*B*a^6*b^8*c^7*d^5*f^2 - 2992*A*B*a^4*b^10*c^5*d^7*f^
2 - 2368*A*B*a^4*b^10*c^7*d^5*f^2 - 2304*A*B*a^4*b^10*c^3*d^9*f^2 - 1968*A*
B*a^9*b^5*c^6*d^6*f^2 - 1872*A*B*a^4*b^10*c^9*d^3*f^2 - 1728*A*B*a^7*b^7*c^
2*d^10*f^2 + 1712*A*B*a^3*b^11*c^8*d^4*f^2 - 1536*A*B*a^10*b^4*c^3*d^9*f^2
+ 1536*A*B*a^6*b^8*c^5*d^7*f^2 - 1392*A*B*a^2*b^12*c^5*d^7*f^2 + 1328*A*B*a
^3*b^11*c^6*d^6*f^2 - 1104*A*B*a^2*b^12*c^3*d^9*f^2 - 1056*A*B*a^6*b^8*c^3*
d^9*f^2 + 976*A*B*a^6*b^8*c^9*d^3*f^2 + 960*A*B*a^11*b^3*c^4*d^8*f^2 + 936*
A*B*a^5*b^9*c^8*d^4*f^2 - 912*A*B*a^10*b^4*c^5*d^7*f^2 + 848*A*B*a^8*b^6*c^
9*d^3*f^2 + 816*A*B*a^3*b^11*c^4*d^8*f^2 - 816*A*B*a^2*b^12*c^7*d^5*f^2 + 7
68*A*B*a^3*b^11*c^10*d^2*f^2 + 672*A*B*a^8*b^6*c^3*d^9*f^2 - 632*A*B*a^9*b^
5*c^8*d^4*f^2 - 608*A*B*a^9*b^5*c^2*d^10*f^2 - 552*A*B*a^9*b^5*c^4*d^8*f^2
- 544*A*B*a^7*b^7*c^10*d^2*f^2 - 480*A*B*a^5*b^9*c^2*d^10*f^2 + 464*A*B*a^5
*b^9*c^10*d^2*f^2 - 464*A*B*a^2*b^12*c^9*d^3*f^2 + 432*A*B*a^11*b^3*c^2*d^1
0*f^2 - 368*A*B*a^12*b^2*c^3*d^9*f^2 - 256*A*B*a^5*b^9*c^6*d^6*f^2 - 208*A*
B*a^12*b^2*c^5*d^7*f^2 + 176*A*B*a^5*b^9*c^4*d^8*f^2 + 112*A*B*a^11*b^3*c^6

*d^6*f^2 + 112*A*B*a^10*b^4*c^7*d^5*f^2 - 16*A*B*a^3*b^11*c^2*d^10*f^2 - 57
6*B*C*a^8*b^6*c*d^11*f^2 + 400*B*C*a^4*b^10*c^11*d*f^2 - 288*B*C*a^6*b^8*c*
d^11*f^2 - 176*B*C*a^6*b^8*c^11*d*f^2 + 128*B*C*a^10*b^4*c*d^11*f^2 - 108*B
*C*a*b^13*c^4*d^8*f^2 - 104*B*C*a^4*b^10*c*d^11*f^2 - 92*B*C*a^13*b*c^4*d^8
*f^2 - 60*B*C*a*b^13*c^8*d^4*f^2 - 60*B*C*a*b^13*c^6*d^6*f^2 + 48*B*C*a^2*b
^12*c^11*d*f^2 - 40*B*C*a*b^13*c^2*d^10*f^2 - 28*B*C*a^13*b*c^2*d^10*f^2 -
24*B*C*a^12*b^2*c*d^11*f^2 + 20*B*C*a*b^13*c^10*d^2*f^2 - 16*B*C*a^2*b^12*c
*d^11*f^2 + 12*B*C*a^13*b*c^6*d^6*f^2 + 912*A*C*a^7*b^7*c*d^11*f^2 + 808*A*
C*a^5*b^9*c*d^11*f^2 + 432*A*C*a^5*b^9*c^11*d*f^2 + 336*A*C*a^3*b^11*c*d^11
*f^2 + 224*A*C*a^11*b^3*c*d^11*f^2 - 112*A*C*a^3*b^11*c^11*d*f^2 + 112*A*C*
a*b^13*c^3*d^9*f^2 - 88*A*C*a*b^13*c^9*d^3*f^2 + 80*A*C*a^13*b*c^3*d^9*f^2
+ 56*A*C*a*b^13*c^5*d^7*f^2 + 48*A*C*a^9*b^5*c*d^11*f^2 - 40*A*C*a^13*b*c^5
*d^7*f^2 - 16*A*C*a^7*b^7*c^11*d*f^2 + 16*A*C*a*b^13*c^7*d^5*f^2 - 496*A*B*
a^4*b^10*c*d^11*f^2 - 400*A*B*a^4*b^10*c^11*d*f^2 + 288*A*B*a^8*b^6*c*d^11*
f^2 - 288*A*B*a^6*b^8*c*d^11*f^2 - 272*A*B*a^2*b^12*c*d^11*f^2 + 240*A*B*a*
b^13*c^6*d^6*f^2 - 224*A*B*a^10*b^4*c*d^11*f^2 + 192*A*B*a*b^13*c^8*d^4*f^2
 + 192*A*B*a*b^13*c^4*d^8*f^2 + 176*A*B*a^6*b^8*c^11*d*f^2 + 104*A*B*a^13*b
*c^4*d^8*f^2 - 48*A*B*a^2*b^12*c^11*d*f^2 + 16*A*B*a^13*b*c^2*d^10*f^2 + 16
*A*B*a*b^13*c^10*d^2*f^2 + 16*A*B*a*b^13*c^2*d^10*f^2 - 96*B*C*b^14*c^7*d^5
*f^2 - 72*B*C*b^14*c^5*d^7*f^2 - 24*B*C*b^14*c^9*d^3*f^2 - 16*B*C*b^14*c^3*
d^9*f^2 + 116*A*C*b^14*c^6*d^6*f^2 + 100*A*C*b^14*c^4*d^8*f^2 + 24*A*C*b^14
*c^2*d^10*f^2 + 22*A*C*b^14*c^8*d^4*f^2 + 16*B*C*a^14*c^3*d^9*f^2 + 8*A*C*b
^14*c^10*d^2*f^2 - 192*A*B*b^14*c^5*d^7*f^2 - 176*A*B*b^14*c^3*d^9*f^2 - 11
2*B*C*a^11*b^3*d^12*f^2 - 48*A*B*b^14*c^7*d^5*f^2 - 28*A*C*a^14*c^2*d^10*f^
2 + 4*B*C*a^5*b^9*d^12*f^2 + 2*A*C*a^14*c^4*d^8*f^2 + 150*A*C*a^10*b^4*d^12
*f^2 - 80*B*C*a^3*b^11*c^12*f^2 + 66*A*C*a^8*b^6*d^12*f^2 - 30*A*C*a^12*b^2
*d^12*f^2 + 24*B*C*a^5*b^9*c^12*f^2 - 16*A*B*a^14*c^3*d^9*f^2 - 12*A*C*a^4*
b^10*d^12*f^2 - 576*A*B*a^7*b^7*d^12*f^2 - 432*A*B*a^9*b^5*d^12*f^2 - 400*A
*B*a^5*b^9*d^12*f^2 - 144*A*B*a^3*b^11*d^12*f^2 - 66*A*C*a^4*b^10*c^12*f^2
+ 54*A*C*a^2*b^12*c^12*f^2 - 32*A*B*a^11*b^3*d^12*f^2 + 2*A*C*a^6*b^8*c^12*
f^2 + 80*A*B*a^3*b^11*c^12*f^2 - 24*A*B*a^5*b^9*c^12*f^2 + 2508*C^2*a^6*b^8
*c^6*d^6*f^2 + 2376*C^2*a^9*b^5*c^5*d^7*f^2 + 2357*C^2*a^6*b^8*c^8*d^4*f^2
- 2048*C^2*a^5*b^9*c^7*d^5*f^2 + 1304*C^2*a^9*b^5*c^3*d^9*f^2 + 1303*C^2*a^
4*b^10*c^8*d^4*f^2 + 1212*C^2*a^4*b^10*c^6*d^6*f^2 - 1203*C^2*a^8*b^6*c^4*d
^8*f^2 - 1192*C^2*a^5*b^9*c^9*d^3*f^2 + 1062*C^2*a^6*b^8*c^4*d^8*f^2 + 984*
C^2*a^9*b^5*c^7*d^5*f^2 - 952*C^2*a^8*b^6*c^6*d^6*f^2 + 768*C^2*a^7*b^7*c^5
*d^7*f^2 - 681*C^2*a^10*b^4*c^4*d^8*f^2 - 672*C^2*a^5*b^9*c^5*d^7*f^2 - 480
*C^2*a^10*b^4*c^6*d^6*f^2 + 458*C^2*a^6*b^8*c^10*d^2*f^2 - 448*C^2*a^7*b^7*
c^7*d^5*f^2 + 422*C^2*a^4*b^10*c^4*d^8*f^2 + 372*C^2*a^6*b^8*c^2*d^10*f^2 +
 360*C^2*a^11*b^3*c^5*d^7*f^2 + 312*C^2*a^7*b^7*c^3*d^9*f^2 + 278*C^2*a^4*b
^10*c^10*d^2*f^2 - 232*C^2*a^7*b^7*c^9*d^3*f^2 + 194*C^2*a^12*b^2*c^2*d^10*
f^2 + 176*C^2*a^9*b^5*c^9*d^3*f^2 + 152*C^2*a^3*b^11*c^5*d^7*f^2 + 124*C^2*
a^4*b^10*c^2*d^10*f^2 - 120*C^2*a^3*b^11*c^7*d^5*f^2 - 114*C^2*a^2*b^12*c^1
0*d^2*f^2 - 102*C^2*a^8*b^6*c^2*d^10*f^2 + 101*C^2*a^12*b^2*c^4*d^8*f^2 + 1
00*C^2*a^2*b^12*c^6*d^6*f^2 - 88*C^2*a^5*b^9*c^3*d^9*f^2 + 77*C^2*a^2*b^12*

c^8*d^4*f^2 + 72*C^2*a^11*b^3*c^3*d^9*f^2 - 64*C^2*a^8*b^6*c^10*d^2*f^2 + 6
4*C^2*a^3*b^11*c^3*d^9*f^2 - 58*C^2*a^10*b^4*c^2*d^10*f^2 + 56*C^2*a^12*b^2
*c^6*d^6*f^2 + 56*C^2*a^11*b^3*c^7*d^5*f^2 + 40*C^2*a^3*b^11*c^9*d^3*f^2 +
36*C^2*a^12*b^2*c^8*d^4*f^2 + 32*C^2*a^2*b^12*c^4*d^8*f^2 + 26*C^2*a^10*b^4
*c^8*d^4*f^2 + 16*C^2*a^2*b^12*c^2*d^10*f^2 + 2*C^2*a^8*b^6*c^8*d^4*f^2 + 2
277*B^2*a^8*b^6*c^4*d^8*f^2 + 2144*B^2*a^5*b^9*c^7*d^5*f^2 - 2112*B^2*a^9*b
^5*c^5*d^7*f^2 + 2028*B^2*a^8*b^6*c^6*d^6*f^2 - 1671*B^2*a^6*b^8*c^8*d^4*f^
2 + 1275*B^2*a^10*b^4*c^4*d^8*f^2 + 1176*B^2*a^5*b^9*c^5*d^7*f^2 + 1096*B^2
*a^5*b^9*c^9*d^3*f^2 - 1044*B^2*a^6*b^8*c^6*d^6*f^2 + 984*B^2*a^10*b^4*c^6*
d^6*f^2 - 968*B^2*a^9*b^5*c^3*d^9*f^2 - 888*B^2*a^9*b^5*c^7*d^5*f^2 + 672*B
^2*a^7*b^7*c^7*d^5*f^2 + 664*B^2*a^5*b^9*c^3*d^9*f^2 - 649*B^2*a^4*b^10*c^8
*d^4*f^2 + 618*B^2*a^8*b^6*c^2*d^10*f^2 + 514*B^2*a^4*b^10*c^4*d^8*f^2 + 46
0*B^2*a^2*b^12*c^6*d^6*f^2 + 422*B^2*a^8*b^6*c^8*d^4*f^2 + 406*B^2*a^10*b^4
*c^2*d^10*f^2 - 382*B^2*a^6*b^8*c^10*d^2*f^2 + 368*B^2*a^2*b^12*c^4*d^8*f^2
 - 312*B^2*a^11*b^3*c^5*d^7*f^2 + 312*B^2*a^7*b^7*c^3*d^9*f^2 + 248*B^2*a^7
*b^7*c^9*d^3*f^2 + 245*B^2*a^2*b^12*c^8*d^4*f^2 - 192*B^2*a^7*b^7*c^5*d^7*f
^2 - 184*B^2*a^3*b^11*c^9*d^3*f^2 + 182*B^2*a^2*b^12*c^10*d^2*f^2 + 176*B^2
*a^3*b^11*c^3*d^9*f^2 + 174*B^2*a^6*b^8*c^4*d^8*f^2 - 170*B^2*a^4*b^10*c^10
*d^2*f^2 - 152*B^2*a^9*b^5*c^9*d^3*f^2 + 152*B^2*a^4*b^10*c^2*d^10*f^2 + 14
2*B^2*a^10*b^4*c^8*d^4*f^2 - 90*B^2*a^12*b^2*c^2*d^10*f^2 + 88*B^2*a^2*b^12
*c^2*d^10*f^2 + 84*B^2*a^8*b^6*c^10*d^2*f^2 + 84*B^2*a^6*b^8*c^2*d^10*f^2 +
 60*B^2*a^12*b^2*c^6*d^6*f^2 - 56*B^2*a^11*b^3*c^7*d^5*f^2 + 53*B^2*a^12*b^
2*c^4*d^8*f^2 + 24*B^2*a^11*b^3*c^3*d^9*f^2 + 24*B^2*a^4*b^10*c^6*d^6*f^2 +
 24*B^2*a^3*b^11*c^7*d^5*f^2 - 8*B^2*a^3*b^11*c^5*d^7*f^2 + 4566*A^2*a^6*b^
8*c^4*d^8*f^2 + 4284*A^2*a^6*b^8*c^6*d^6*f^2 - 3776*A^2*a^5*b^9*c^7*d^5*f^2
 - 3624*A^2*a^5*b^9*c^5*d^7*f^2 + 3122*A^2*a^4*b^10*c^4*d^8*f^2 + 3108*A^2*
a^6*b^8*c^2*d^10*f^2 + 2741*A^2*a^6*b^8*c^8*d^4*f^2 + 2592*A^2*a^4*b^10*c^6
*d^6*f^2 - 2536*A^2*a^5*b^9*c^3*d^9*f^2 + 2224*A^2*a^4*b^10*c^2*d^10*f^2 -
2184*A^2*a^7*b^7*c^3*d^9*f^2 - 2016*A^2*a^7*b^7*c^5*d^7*f^2 - 1984*A^2*a^7*
b^7*c^7*d^5*f^2 + 1626*A^2*a^8*b^6*c^2*d^10*f^2 - 1624*A^2*a^5*b^9*c^9*d^3*
f^2 + 1603*A^2*a^4*b^10*c^8*d^4*f^2 + 1296*A^2*a^9*b^5*c^5*d^7*f^2 - 1144*A
^2*a^3*b^11*c^5*d^7*f^2 - 992*A^2*a^3*b^11*c^3*d^9*f^2 + 968*A^2*a^2*b^12*c
^4*d^8*f^2 - 888*A^2*a^3*b^11*c^7*d^5*f^2 + 849*A^2*a^8*b^6*c^4*d^8*f^2 + 8
08*A^2*a^2*b^12*c^2*d^10*f^2 - 616*A^2*a^7*b^7*c^9*d^3*f^2 + 554*A^2*a^6*b^
8*c^10*d^2*f^2 - 504*A^2*a^10*b^4*c^6*d^6*f^2 + 504*A^2*a^9*b^5*c^7*d^5*f^2
 + 460*A^2*a^2*b^12*c^6*d^6*f^2 + 350*A^2*a^10*b^4*c^2*d^10*f^2 + 350*A^2*a
^4*b^10*c^10*d^2*f^2 - 321*A^2*a^10*b^4*c^4*d^8*f^2 + 216*A^2*a^11*b^3*c^5*
d^7*f^2 - 216*A^2*a^11*b^3*c^3*d^9*f^2 + 182*A^2*a^12*b^2*c^2*d^10*f^2 - 15
2*A^2*a^3*b^11*c^9*d^3*f^2 - 124*A^2*a^8*b^6*c^6*d^6*f^2 - 114*A^2*a^2*b^12
*c^10*d^2*f^2 + 104*A^2*a^9*b^5*c^3*d^9*f^2 + 77*A^2*a^2*b^12*c^8*d^4*f^2 +
 74*A^2*a^8*b^6*c^8*d^4*f^2 - 70*A^2*a^10*b^4*c^8*d^4*f^2 + 56*A^2*a^11*b^3
*c^7*d^5*f^2 + 56*A^2*a^9*b^5*c^9*d^3*f^2 + 41*A^2*a^12*b^2*c^4*d^8*f^2 - 2
8*A^2*a^12*b^2*c^6*d^6*f^2 - 28*A^2*a^8*b^6*c^10*d^2*f^2 - 16*B*C*b^14*c^11
*d*f^2 - 16*B*C*a^14*c*d^11*f^2 - 48*A*B*b^14*c*d^11*f^2 + 16*A*B*b^14*c^11
*d*f^2 + 12*B*C*a^13*b*d^12*f^2 + 24*B*C*a*b^13*c^12*f^2 + 16*A*B*a^14*c*d^

11*f^2 - 24*A*B*a^13*b*d^12*f^2 - 24*A*B*a*b^13*d^12*f^2 - 24*A*B*a*b^13*c^
12*f^2 + 216*C^2*a^9*b^5*c*d^11*f^2 - 216*C^2*a^5*b^9*c^11*d*f^2 + 56*C^2*a
^3*b^11*c^11*d*f^2 + 56*C^2*a*b^13*c^9*d^3*f^2 + 56*C^2*a*b^13*c^5*d^7*f^2
- 40*C^2*a^11*b^3*c*d^11*f^2 + 40*C^2*a*b^13*c^7*d^5*f^2 + 32*C^2*a^13*b*c^
5*d^7*f^2 - 24*C^2*a^7*b^7*c*d^11*f^2 - 16*C^2*a^13*b*c^3*d^9*f^2 + 16*C^2*
a*b^13*c^3*d^9*f^2 + 8*C^2*a^7*b^7*c^11*d*f^2 - 8*C^2*a^5*b^9*c*d^11*f^2 +
264*B^2*a^7*b^7*c*d^11*f^2 + 224*B^2*a^5*b^9*c*d^11*f^2 + 168*B^2*a^5*b^9*c
^11*d*f^2 - 112*B^2*a*b^13*c^9*d^3*f^2 - 104*B^2*a^3*b^11*c^11*d*f^2 - 104*
B^2*a*b^13*c^7*d^5*f^2 + 96*B^2*a^3*b^11*c*d^11*f^2 + 88*B^2*a^11*b^3*c*d^1
1*f^2 - 72*B^2*a^9*b^5*c*d^11*f^2 - 64*B^2*a*b^13*c^5*d^7*f^2 + 32*B^2*a^13
*b*c^3*d^9*f^2 - 24*B^2*a^13*b*c^5*d^7*f^2 - 24*B^2*a^7*b^7*c^11*d*f^2 + 16
*B^2*a*b^13*c^3*d^9*f^2 - 888*A^2*a^7*b^7*c*d^11*f^2 - 800*A^2*a^5*b^9*c*d^
11*f^2 - 336*A^2*a^3*b^11*c*d^11*f^2 - 264*A^2*a^9*b^5*c*d^11*f^2 - 216*A^2
*a^5*b^9*c^11*d*f^2 - 184*A^2*a^11*b^3*c*d^11*f^2 - 128*A^2*a*b^13*c^3*d^9*
f^2 - 112*A^2*a*b^13*c^5*d^7*f^2 - 64*A^2*a^13*b*c^3*d^9*f^2 + 56*A^2*a^3*b
^11*c^11*d*f^2 - 56*A^2*a*b^13*c^7*d^5*f^2 + 32*A^2*a*b^13*c^9*d^3*f^2 + 8*
A^2*a^13*b*c^5*d^7*f^2 + 8*A^2*a^7*b^7*c^11*d*f^2 + 24*C^2*a*b^13*c^11*d*f^
2 - 16*C^2*a^13*b*c*d^11*f^2 - 40*B^2*a*b^13*c^11*d*f^2 + 24*B^2*a^13*b*c*d
^11*f^2 + 16*B^2*a*b^13*c*d^11*f^2 - 48*A^2*a*b^13*c*d^11*f^2 - 40*A^2*a^13
*b*c*d^11*f^2 + 24*A^2*a*b^13*c^11*d*f^2 - 6*A*C*b^14*c^12*f^2 + 2*A*C*a^14
*d^12*f^2 + 31*C^2*b^14*c^8*d^4*f^2 + 20*C^2*b^14*c^6*d^6*f^2 + 4*C^2*b^14*
c^4*d^8*f^2 + 2*C^2*b^14*c^10*d^2*f^2 + 80*B^2*b^14*c^6*d^6*f^2 + 64*B^2*b^
14*c^4*d^8*f^2 + 31*B^2*b^14*c^8*d^4*f^2 + 16*B^2*b^14*c^2*d^10*f^2 + 14*C^
2*a^14*c^2*d^10*f^2 + 14*B^2*b^14*c^10*d^2*f^2 - C^2*a^14*c^4*d^8*f^2 + 120
*A^2*b^14*c^2*d^10*f^2 + 112*A^2*b^14*c^4*d^8*f^2 + 33*C^2*a^12*b^2*d^12*f^
2 - 27*C^2*a^10*b^4*d^12*f^2 - 17*A^2*b^14*c^8*d^4*f^2 - 10*B^2*a^14*c^2*d^
10*f^2 - 10*A^2*b^14*c^10*d^2*f^2 + 8*A^2*b^14*c^6*d^6*f^2 + 3*C^2*a^8*b^6*
d^12*f^2 + 3*B^2*a^14*c^4*d^8*f^2 + 117*B^2*a^10*b^4*d^12*f^2 + 111*B^2*a^8
*b^6*d^12*f^2 + 72*B^2*a^6*b^8*d^12*f^2 + 33*C^2*a^4*b^10*c^12*f^2 - 27*C^2
*a^2*b^12*c^12*f^2 + 24*B^2*a^4*b^10*d^12*f^2 + 14*A^2*a^14*c^2*d^10*f^2 +
4*B^2*a^2*b^12*d^12*f^2 - 3*B^2*a^12*b^2*d^12*f^2 - C^2*a^6*b^8*c^12*f^2 -
A^2*a^14*c^4*d^8*f^2 + 720*A^2*a^6*b^8*d^12*f^2 + 552*A^2*a^4*b^10*d^12*f^2
+ 471*A^2*a^8*b^6*d^12*f^2 + 216*A^2*a^2*b^12*d^12*f^2 + 93*A^2*a^10*b^4*d
^12*f^2 + 33*B^2*a^2*b^12*c^12*f^2 + 33*A^2*a^12*b^2*d^12*f^2 - 27*B^2*a^4*
b^10*c^12*f^2 + 3*B^2*a^6*b^8*c^12*f^2 + 33*A^2*a^4*b^10*c^12*f^2 - 27*A^2*
a^2*b^12*c^12*f^2 - A^2*a^6*b^8*c^12*f^2 + 3*C^2*b^14*c^12*f^2 - C^2*a^14*d
^12*f^2 + 36*A^2*b^14*d^12*f^2 + 3*B^2*a^14*d^12*f^2 - B^2*b^14*c^12*f^2 +
3*A^2*b^14*c^12*f^2 - A^2*a^14*d^12*f^2 - 44*A*B*C*a^10*b*c*d^9*f + 3816*A*
B*C*a^4*b^7*c^5*d^5*f + 2920*A*B*C*a^5*b^6*c^2*d^8*f - 2736*A*B*C*a^6*b^5*c
^3*d^7*f - 2672*A*B*C*a^3*b^8*c^4*d^6*f + 1996*A*B*C*a^7*b^4*c^4*d^6*f - 14
12*A*B*C*a^5*b^6*c^6*d^4*f + 1120*A*B*C*a^2*b^9*c^3*d^7*f + 1080*A*B*C*a^7*
b^4*c^2*d^8*f + 1040*A*B*C*a^2*b^9*c^5*d^5*f + 684*A*B*C*a^5*b^6*c^4*d^6*f
+ 592*A*B*C*a^4*b^7*c^3*d^7*f - 560*A*B*C*a^2*b^9*c^7*d^3*f - 448*A*B*C*a^3
*b^8*c^2*d^8*f - 400*A*B*C*a^8*b^3*c^5*d^5*f - 398*A*B*C*a^9*b^2*c^2*d^8*f
- 312*A*B*C*a^3*b^8*c^6*d^4*f + 166*A*B*C*a^3*b^8*c^8*d^2*f + 136*A*B*C*a^6

*b^5*c^5*d^5*f + 128*A*B*C*a^6*b^5*c^7*d^3*f - 100*A*B*C*a^7*b^4*c^6*d^4*f
- 64*A*B*C*a^9*b^2*c^4*d^6*f + 64*A*B*C*a^4*b^7*c^7*d^3*f - 32*A*B*C*a^8*b^
3*c^3*d^7*f - 16*A*B*C*a^5*b^6*c^8*d^2*f - 1312*A*B*C*a^4*b^7*c*d^9*f + 996
*A*B*C*a^8*b^3*c*d^9*f + 728*A*B*C*a*b^10*c^6*d^4*f - 624*A*B*C*a^6*b^5*c*d
^9*f - 584*A*B*C*a*b^10*c^2*d^8*f - 512*A*B*C*a*b^10*c^4*d^6*f - 320*A*B*C*
a^2*b^9*c*d^9*f - 98*A*B*C*a*b^10*c^8*d^2*f + 36*A*B*C*a^2*b^9*c^9*d*f + 32
*A*B*C*a^10*b*c^3*d^7*f - 16*A*B*C*a^4*b^7*c^9*d*f + 46*B*C^2*a^10*b*c*d^9*
f - 16*B^2*C*a*b^10*c*d^9*f - 2*B^2*C*a*b^10*c^9*d*f + 312*A^2*C*a*b^10*c*d
^9*f - 48*A*C^2*a*b^10*c*d^9*f - 6*A^2*C*a*b^10*c^9*d*f + 6*A*C^2*a*b^10*c^
9*d*f + 208*A*B^2*a*b^10*c*d^9*f - 2*A^2*B*a^10*b*c*d^9*f + 2*A*B^2*a*b^10*
c^9*d*f - 224*A*B*C*b^11*c^5*d^5*f + 80*A*B*C*b^11*c^7*d^3*f - 32*A*B*C*b^1
1*c^3*d^7*f + 2*A*B*C*a^11*c^2*d^8*f - 480*A*B*C*a^7*b^4*d^10*f + 78*A*B*C*
a^9*b^2*d^10*f - 64*A*B*C*a^5*b^6*d^10*f + 2*A*B*C*a^3*b^8*c^10*f - 1692*B*
C^2*a^4*b^7*c^5*d^5*f - 1500*B^2*C*a^5*b^6*c^5*d^5*f - 1464*B^2*C*a^5*b^6*c
^3*d^7*f + 1426*B*C^2*a^5*b^6*c^6*d^4*f - 1158*B^2*C*a^4*b^7*c^6*d^4*f + 11
52*B*C^2*a^6*b^5*c^3*d^7*f + 1026*B^2*C*a^6*b^5*c^4*d^6*f - 974*B*C^2*a^7*b
^4*c^4*d^6*f + 960*B^2*C*a^3*b^8*c^5*d^5*f - 884*B*C^2*a^5*b^6*c^2*d^8*f -
764*B^2*C*a^7*b^4*c^5*d^5*f + 752*B^2*C*a^4*b^7*c^2*d^8*f - 752*B*C^2*a^4*b
^7*c^3*d^7*f + 738*B^2*C*a^4*b^7*c^4*d^6*f - 688*B^2*C*a^2*b^9*c^6*d^4*f -
675*B^2*C*a^8*b^3*c^2*d^8*f + 560*B*C^2*a^8*b^3*c^5*d^5*f + 496*B*C^2*a^3*b
^8*c^4*d^6*f + 496*B*C^2*a^2*b^9*c^7*d^3*f - 468*B*C^2*a^7*b^4*c^2*d^8*f +
456*B^2*C*a^3*b^8*c^7*d^3*f - 452*B^2*C*a^8*b^3*c^4*d^6*f - 416*B*C^2*a^2*b
^9*c^3*d^7*f + 378*B*C^2*a^5*b^6*c^4*d^6*f + 376*B*C^2*a^8*b^3*c^3*d^7*f -
360*B^2*C*a^6*b^5*c^2*d^8*f + 355*B*C^2*a^9*b^2*c^2*d^8*f + 346*B^2*C*a^6*b
^5*c^6*d^4*f - 320*B^2*C*a^2*b^9*c^4*d^6*f + 268*B^2*C*a^2*b^9*c^2*d^8*f +
216*B^2*C*a^7*b^4*c^3*d^7*f - 203*B*C^2*a^3*b^8*c^8*d^2*f - 184*B*C^2*a^6*b
^5*c^7*d^3*f + 170*B*C^2*a^7*b^4*c^6*d^4*f + 160*B^2*C*a^5*b^6*c^7*d^3*f -
160*B*C^2*a^2*b^9*c^5*d^5*f - 140*B^2*C*a^4*b^7*c^8*d^2*f - 136*B*C^2*a^3*b
^8*c^2*d^8*f + 112*B^2*C*a^9*b^2*c^3*d^7*f + 91*B^2*C*a^2*b^9*c^8*d^2*f + 8
8*B*C^2*a^4*b^7*c^7*d^3*f + 72*B^2*C*a^8*b^3*c^6*d^4*f - 64*B^2*C*a^3*b^8*c
^3*d^7*f - 60*B*C^2*a^3*b^8*c^6*d^4*f + 56*B*C^2*a^9*b^2*c^4*d^6*f + 52*B*C
^2*a^6*b^5*c^5*d^5*f + 48*B^2*C*a^9*b^2*c^5*d^5*f - 48*B^2*C*a^7*b^4*c^7*d^
3*f + 44*B*C^2*a^5*b^6*c^8*d^2*f - 36*B*C^2*a^9*b^2*c^6*d^4*f + 12*B^2*C*a^
6*b^5*c^8*d^2*f - 2958*A^2*C*a^4*b^7*c^4*d^6*f - 1932*A^2*C*a^4*b^7*c^2*d^8
*f + 1848*A^2*C*a^5*b^6*c^3*d^7*f + 1728*A^2*C*a^3*b^8*c^3*d^7*f + 1524*A^2
*C*a^5*b^6*c^5*d^5*f + 1374*A*C^2*a^4*b^7*c^4*d^6*f - 1272*A*C^2*a^5*b^6*c^
3*d^7*f - 1236*A*C^2*a^5*b^6*c^5*d^5*f + 1116*A*C^2*a^4*b^7*c^2*d^8*f - 111
0*A^2*C*a^6*b^5*c^4*d^6*f + 1038*A*C^2*a^6*b^5*c^4*d^6*f - 768*A^2*C*a^2*b^
9*c^2*d^8*f - 696*A^2*C*a^7*b^4*c^3*d^7*f - 666*A*C^2*a^4*b^7*c^6*d^4*f + 5
64*A^2*C*a^6*b^5*c^2*d^8*f - 564*A*C^2*a^7*b^4*c^5*d^5*f - 555*A*C^2*a^8*b^
3*c^2*d^8*f + 519*A^2*C*a^8*b^3*c^2*d^8*f - 480*A*C^2*a^3*b^8*c^3*d^7*f + 4
56*A*C^2*a^3*b^8*c^5*d^5*f - 420*A*C^2*a^2*b^9*c^6*d^4*f + 408*A*C^2*a^7*b^
4*c^3*d^7*f + 408*A*C^2*a^2*b^9*c^2*d^8*f + 348*A^2*C*a^2*b^9*c^6*d^4*f - 3
48*A*C^2*a^6*b^5*c^2*d^8*f + 342*A*C^2*a^6*b^5*c^6*d^4*f - 336*A*C^2*a^8*b^
3*c^4*d^6*f + 324*A^2*C*a^7*b^4*c^5*d^5*f - 312*A^2*C*a^2*b^9*c^4*d^6*f + 2

64*A^2*C*a^8*b^3*c^4*d^6*f + 240*A*C^2*a^5*b^6*c^7*d^3*f + 195*A*C^2*a^2*b^
9*c^8*d^2*f - 174*A^2*C*a^6*b^5*c^6*d^4*f + 144*A*C^2*a^9*b^2*c^3*d^7*f - 1
23*A^2*C*a^2*b^9*c^8*d^2*f + 120*A*C^2*a^3*b^8*c^7*d^3*f + 108*A*C^2*a^8*b^
3*c^6*d^4*f - 102*A^2*C*a^4*b^7*c^6*d^4*f - 96*A^2*C*a^4*b^7*c^8*d^2*f + 72
*A^2*C*a^3*b^8*c^7*d^3*f + 72*A*C^2*a^9*b^2*c^5*d^5*f - 48*A^2*C*a^9*b^2*c^
3*d^7*f + 48*A^2*C*a^5*b^6*c^7*d^3*f - 48*A*C^2*a^2*b^9*c^4*d^6*f - 24*A^2*
C*a^3*b^8*c^5*d^5*f - 12*A*C^2*a^4*b^7*c^8*d^2*f + 2736*A^2*B*a^6*b^5*c^3*d
^7*f + 2464*A^2*B*a^3*b^8*c^4*d^6*f - 2298*A*B^2*a^4*b^7*c^4*d^6*f - 2252*A
^2*B*a^5*b^6*c^2*d^8*f - 1692*A^2*B*a^4*b^7*c^5*d^5*f - 1592*A*B^2*a^4*b^7*
c^2*d^8*f - 1338*A*B^2*a^6*b^5*c^4*d^6*f + 1320*A*B^2*a^5*b^6*c^3*d^7*f + 1
212*A*B^2*a^5*b^6*c^5*d^5*f - 1056*A*B^2*a^3*b^8*c^5*d^5*f + 1024*A^2*B*a^4
*b^7*c^3*d^7*f - 1022*A^2*B*a^7*b^4*c^4*d^6*f - 880*A^2*B*a^2*b^9*c^5*d^5*f
 - 846*A^2*B*a^5*b^6*c^4*d^6*f - 840*A*B^2*a^7*b^4*c^3*d^7*f + 760*A*B^2*a^
2*b^9*c^6*d^4*f - 704*A^2*B*a^2*b^9*c^3*d^7*f + 688*A*B^2*a^3*b^8*c^3*d^7*f
 + 660*A^2*B*a^3*b^8*c^6*d^4*f - 612*A^2*B*a^7*b^4*c^2*d^8*f + 462*A*B^2*a^
4*b^7*c^6*d^4*f + 459*A*B^2*a^8*b^3*c^2*d^8*f - 412*A*B^2*a^2*b^9*c^2*d^8*f
 - 408*A*B^2*a^3*b^8*c^7*d^3*f + 388*A^2*B*a^6*b^5*c^5*d^5*f + 296*A^2*B*a^
3*b^8*c^2*d^8*f + 288*A*B^2*a^6*b^5*c^2*d^8*f + 284*A*B^2*a^7*b^4*c^5*d^5*f
 + 236*A*B^2*a^8*b^3*c^4*d^6*f - 226*A*B^2*a^6*b^5*c^6*d^4*f + 212*A*B^2*a^
2*b^9*c^4*d^6*f + 202*A^2*B*a^5*b^6*c^6*d^4*f - 152*A^2*B*a^4*b^7*c^7*d^3*f
 + 88*A^2*B*a^8*b^3*c^3*d^7*f + 79*A^2*B*a^9*b^2*c^2*d^8*f - 70*A^2*B*a^7*b
^4*c^6*d^4*f + 68*A*B^2*a^4*b^7*c^8*d^2*f + 64*A^2*B*a^2*b^9*c^7*d^3*f - 64
*A*B^2*a^9*b^2*c^3*d^7*f + 56*A^2*B*a^8*b^3*c^5*d^5*f + 56*A^2*B*a^6*b^5*c^
7*d^3*f + 37*A^2*B*a^3*b^8*c^8*d^2*f - 28*A^2*B*a^9*b^2*c^4*d^6*f - 28*A^2*
B*a^5*b^6*c^8*d^2*f + 17*A*B^2*a^2*b^9*c^8*d^2*f - 16*A*B^2*a^5*b^6*c^7*d^3
*f + 48*A*B*C*b^11*c*d^9*f + 4*A*B*C*b^11*c^9*d*f + 24*A*B*C*a*b^10*d^10*f
- 6*A*B*C*a*b^10*c^10*f + 432*B^2*C*a^7*b^4*c*d^9*f - 376*B*C^2*a*b^10*c^6*
d^4*f - 354*B*C^2*a^8*b^3*c*d^9*f + 352*B^2*C*a*b^10*c^5*d^5*f + 320*B^2*C*
a^5*b^6*c*d^9*f + 256*B^2*C*a*b^10*c^3*d^7*f - 232*B^2*C*a*b^10*c^7*d^3*f -
 210*B^2*C*a^9*b^2*c*d^9*f - 152*B*C^2*a*b^10*c^4*d^6*f + 85*B*C^2*a*b^10*c
^8*d^2*f + 72*B^2*C*a^3*b^8*c*d^9*f - 48*B*C^2*a^6*b^5*c*d^9*f - 40*B*C^2*a
^10*b*c^3*d^7*f + 40*B*C^2*a*b^10*c^2*d^8*f + 37*B^2*C*a^10*b*c^2*d^8*f + 2
2*B^2*C*a^3*b^8*c^9*d*f - 18*B*C^2*a^2*b^9*c^9*d*f + 16*B*C^2*a^2*b^9*c*d^9
*f - 12*B^2*C*a^10*b*c^4*d^6*f + 8*B*C^2*a^4*b^7*c^9*d*f + 8*B*C^2*a^4*b^7*
c*d^9*f - 984*A^2*C*a^7*b^4*c*d^9*f + 672*A^2*C*a^3*b^8*c*d^9*f + 552*A*C^2
*a^7*b^4*c*d^9*f - 504*A^2*C*a*b^10*c^5*d^5*f - 408*A^2*C*a^5*b^6*c*d^9*f +
 408*A*C^2*a^5*b^6*c*d^9*f + 336*A*C^2*a*b^10*c^5*d^5*f - 216*A*C^2*a*b^10*
c^7*d^3*f + 192*A*C^2*a*b^10*c^3*d^7*f - 162*A*C^2*a^9*b^2*c*d^9*f + 120*A^
2*C*a*b^10*c^7*d^3*f + 96*A^2*C*a*b^10*c^3*d^7*f + 90*A^2*C*a^9*b^2*c*d^9*f
 + 66*A^2*C*a^3*b^8*c^9*d*f - 66*A*C^2*a^3*b^8*c^9*d*f + 57*A*C^2*a^10*b*c^
2*d^8*f - 48*A*C^2*a^3*b^8*c*d^9*f - 9*A^2*C*a^10*b*c^2*d^8*f + 1736*A^2*B*
a^4*b^7*c*d^9*f + 1248*A^2*B*a^6*b^5*c*d^9*f - 1008*A*B^2*a^7*b^4*c*d^9*f +
 772*A^2*B*a*b^10*c^4*d^6*f - 688*A*B^2*a*b^10*c^5*d^5*f - 608*A*B^2*a^5*b^
6*c*d^9*f + 436*A^2*B*a*b^10*c^2*d^8*f - 426*A^2*B*a^8*b^3*c*d^9*f + 312*A*
B^2*a^3*b^8*c*d^9*f + 304*A^2*B*a^2*b^9*c*d^9*f - 244*A^2*B*a*b^10*c^6*d^4*

f - 160*A*B^2*a*b^10*c^3*d^7*f + 114*A*B^2*a^9*b^2*c*d^9*f + 88*A*B^2*a*b^1
0*c^7*d^3*f - 22*A*B^2*a^3*b^8*c^9*d*f - 18*A^2*B*a^2*b^9*c^9*d*f + 13*A^2*
B*a*b^10*c^8*d^2*f - 13*A*B^2*a^10*b*c^2*d^8*f + 8*A^2*B*a^10*b*c^3*d^7*f +
 8*A^2*B*a^4*b^7*c^9*d*f + 112*B^2*C*b^11*c^6*d^4*f - 64*B*C^2*b^11*c^7*d^3
*f + 16*B^2*C*b^11*c^4*d^6*f - 16*B^2*C*b^11*c^2*d^8*f + 16*B*C^2*b^11*c^5*
d^5*f + 16*B*C^2*b^11*c^3*d^7*f - B^2*C*b^11*c^8*d^2*f + 96*A^2*C*b^11*c^4*
d^6*f - 84*A^2*C*b^11*c^6*d^4*f + 72*A*C^2*b^11*c^6*d^4*f - 24*A*C^2*b^11*c
^4*d^6*f - 24*A*C^2*b^11*c^2*d^8*f - 21*A*C^2*b^11*c^8*d^2*f + 12*A^2*C*b^1
1*c^2*d^8*f + 9*A^2*C*b^11*c^8*d^2*f - B*C^2*a^11*c^2*d^8*f + 176*A*B^2*b^1
1*c^4*d^6*f + 136*A^2*B*b^11*c^5*d^5*f - 128*A^2*B*b^11*c^3*d^7*f + 112*A*B
^2*b^11*c^2*d^8*f + 111*B^2*C*a^8*b^3*d^10*f - 64*A*B^2*b^11*c^6*d^4*f - 39
*B*C^2*a^9*b^2*d^10*f + 24*B*C^2*a^7*b^4*d^10*f - 16*A^2*B*b^11*c^7*d^3*f -
 4*B^2*C*a^2*b^9*d^10*f - 4*B*C^2*a^5*b^6*d^10*f + 432*A^2*C*a^6*b^5*d^10*f
 + 192*A^2*C*a^4*b^7*d^10*f - 111*A^2*C*a^8*b^3*d^10*f + 111*A*C^2*a^8*b^3*
d^10*f - 72*A*C^2*a^6*b^5*d^10*f + 12*A*C^2*a^4*b^7*d^10*f - 3*B^2*C*a^2*b^
9*c^10*f - A^2*B*a^11*c^2*d^8*f - B*C^2*a^3*b^8*c^10*f + 456*A^2*B*a^7*b^4*
d^10*f - 288*A^2*B*a^3*b^8*d^10*f + 252*A*B^2*a^6*b^5*d^10*f + 192*A*B^2*a^
4*b^7*d^10*f - 183*A*B^2*a^8*b^3*d^10*f - 148*A^2*B*a^5*b^6*d^10*f + 76*A*B
^2*a^2*b^9*d^10*f - 9*A^2*C*a^2*b^9*c^10*f + 9*A*C^2*a^2*b^9*c^10*f - 3*A^2
*B*a^9*b^2*d^10*f + 3*A*B^2*a^2*b^9*c^10*f - A^2*B*a^3*b^8*c^10*f - 2*C^3*a
*b^10*c^9*d*f - 2*B^3*a^10*b*c*d^9*f - 264*A^3*a*b^10*c*d^9*f + 2*A^3*a*b^1
0*c^9*d*f - 2*B*C^2*b^11*c^9*d*f - 2*B^2*C*a^11*c*d^9*f - 120*A^2*B*b^11*c*
d^9*f - 9*B^2*C*a^10*b*d^10*f - 6*A^2*C*a^11*c*d^9*f + 6*A*C^2*a^11*c*d^9*f
 - 2*A^2*B*b^11*c^9*d*f + 9*A^2*C*a^10*b*d^10*f - 9*A*C^2*a^10*b*d^10*f + 3
*B*C^2*a*b^10*c^10*f + 2*A*B^2*a^11*c*d^9*f - 132*A^2*B*a*b^10*d^10*f - 3*A
*B^2*a^10*b*d^10*f + 3*A^2*B*a*b^10*c^10*f + 520*C^3*a^5*b^6*c^3*d^7*f + 46
0*C^3*a^5*b^6*c^5*d^5*f - 418*C^3*a^6*b^5*c^4*d^6*f + 406*C^3*a^4*b^7*c^6*d
^4*f + 268*C^3*a^7*b^4*c^5*d^5*f - 266*C^3*a^6*b^5*c^6*d^4*f + 233*C^3*a^8*
b^3*c^2*d^8*f - 176*C^3*a^5*b^6*c^7*d^3*f + 164*C^3*a^2*b^9*c^6*d^4*f + 140
*C^3*a^6*b^5*c^2*d^8*f + 136*C^3*a^2*b^9*c^4*d^6*f - 128*C^3*a^9*b^2*c^3*d^
7*f + 128*C^3*a^3*b^8*c^3*d^7*f - 108*C^3*a^8*b^3*c^6*d^4*f - 104*C^3*a^3*b
^8*c^7*d^3*f - 104*C^3*a^3*b^8*c^5*d^5*f + 100*C^3*a^8*b^3*c^4*d^6*f - 89*C
^3*a^2*b^9*c^8*d^2*f - 72*C^3*a^9*b^2*c^5*d^5*f - 40*C^3*a^7*b^4*c^3*d^7*f
+ 40*C^3*a^4*b^7*c^8*d^2*f - 28*C^3*a^4*b^7*c^2*d^8*f - 16*C^3*a^2*b^9*c^2*
d^8*f - 2*C^3*a^4*b^7*c^4*d^6*f + 828*B^3*a^4*b^7*c^5*d^5*f + 408*B^3*a^5*b
^6*c^2*d^8*f + 390*B^3*a^7*b^4*c^4*d^6*f - 372*B^3*a^3*b^8*c^4*d^6*f - 336*
B^3*a^6*b^5*c^3*d^7*f - 314*B^3*a^5*b^6*c^6*d^4*f + 288*B^3*a^4*b^7*c^3*d^7
*f + 216*B^3*a^7*b^4*c^2*d^8*f - 176*B^3*a^2*b^9*c^7*d^3*f + 128*B^3*a^2*b^
9*c^3*d^7*f + 108*B^3*a^6*b^5*c^5*d^5*f + 88*B^3*a^4*b^7*c^7*d^3*f + 72*B^3
*a^2*b^9*c^5*d^5*f - 68*B^3*a^3*b^8*c^2*d^8*f - 65*B^3*a^9*b^2*c^2*d^8*f -
56*B^3*a^8*b^3*c^5*d^5*f + 40*B^3*a^6*b^5*c^7*d^3*f + 37*B^3*a^3*b^8*c^8*d^
2*f + 30*B^3*a^5*b^6*c^4*d^6*f - 28*B^3*a^5*b^6*c^8*d^2*f + 24*B^3*a^8*b^3*
c^3*d^7*f - 4*B^3*a^9*b^2*c^4*d^6*f - 2*B^3*a^7*b^4*c^6*d^4*f + 1586*A^3*a^
4*b^7*c^4*d^6*f - 1376*A^3*a^3*b^8*c^3*d^7*f - 1096*A^3*a^5*b^6*c^3*d^7*f +
 844*A^3*a^4*b^7*c^2*d^8*f - 748*A^3*a^5*b^6*c^5*d^5*f + 490*A^3*a^6*b^5*c^

4*d^6*f + 376*A^3*a^2*b^9*c^2*d^8*f + 362*A^3*a^4*b^7*c^6*d^4*f - 356*A^3*a
^6*b^5*c^2*d^8*f + 328*A^3*a^7*b^4*c^3*d^7*f - 328*A^3*a^3*b^8*c^5*d^5*f +
224*A^3*a^2*b^9*c^4*d^6*f - 197*A^3*a^8*b^3*c^2*d^8*f - 112*A^3*a^5*b^6*c^7
*d^3*f + 98*A^3*a^6*b^5*c^6*d^4*f - 92*A^3*a^2*b^9*c^6*d^4*f - 88*A^3*a^3*b
^8*c^7*d^3*f + 68*A^3*a^4*b^7*c^8*d^2*f + 32*A^3*a^9*b^2*c^3*d^7*f - 28*A^3
*a^8*b^3*c^4*d^6*f - 28*A^3*a^7*b^4*c^5*d^5*f + 17*A^3*a^2*b^9*c^8*d^2*f +
104*C^3*a*b^10*c^7*d^3*f + 54*C^3*a^9*b^2*c*d^9*f - 40*C^3*a^7*b^4*c*d^9*f
- 35*C^3*a^10*b*c^2*d^8*f + 22*C^3*a^3*b^8*c^9*d*f + 16*C^3*a*b^10*c^5*d^5*
f - 16*C^3*a*b^10*c^3*d^7*f + 8*C^3*a^5*b^6*c*d^9*f - 2*A*B*C*a^11*d^10*f +
198*B^3*a^8*b^3*c*d^9*f + 192*B^3*a*b^10*c^6*d^4*f - 128*B^3*a^4*b^7*c*d^9
*f - 80*B^3*a*b^10*c^2*d^8*f - 56*B^3*a^2*b^9*c*d^9*f - 24*B^3*a^6*b^5*c*d^
9*f - 18*B^3*a^2*b^9*c^9*d*f - 16*B^3*a*b^10*c^4*d^6*f + 13*B^3*a*b^10*c^8*
d^2*f + 8*B^3*a^10*b*c^3*d^7*f + 8*B^3*a^4*b^7*c^9*d*f - 624*A^3*a^3*b^8*c*
d^9*f + 472*A^3*a^7*b^4*c*d^9*f - 272*A^3*a*b^10*c^3*d^7*f + 152*A^3*a*b^10
*c^5*d^5*f - 22*A^3*a^3*b^8*c^9*d*f + 18*A^3*a^9*b^2*c*d^9*f - 13*A^3*a^10*
b*c^2*d^8*f - 8*A^3*a^5*b^6*c*d^9*f - 8*A^3*a*b^10*c^7*d^3*f + A*B^2*b^11*c
^8*d^2*f + 11*C^3*b^11*c^8*d^2*f - 8*C^3*b^11*c^6*d^4*f - 4*C^3*b^11*c^4*d^
6*f - 64*B^3*b^11*c^5*d^5*f - 32*B^3*b^11*c^3*d^7*f - 68*A^3*b^11*c^4*d^6*f
+ 20*A^3*b^11*c^6*d^4*f + 12*A^3*b^11*c^2*d^8*f - C^3*a^8*b^3*d^10*f - B^3
*a^11*c^2*d^8*f - 60*B^3*a^7*b^4*d^10*f - 32*B^3*a^5*b^6*d^10*f + 21*B^3*a^
9*b^2*d^10*f - 12*B^3*a^3*b^8*d^10*f - 3*C^3*a^2*b^9*c^10*f - 360*A^3*a^6*b
^5*d^10*f - 204*A^3*a^4*b^7*d^10*f - B^3*a^3*b^8*c^10*f + 3*A^3*a^2*b^9*c^1
0*f - 2*C^3*a^11*c*d^9*f - 2*B^3*b^11*c^9*d*f + 3*C^3*a^10*b*d^10*f + 2*A^3
*a^11*c*d^9*f + 3*B^3*a*b^10*c^10*f - 3*A^3*a^10*b*d^10*f - 36*A^2*C*b^11*d
^10*f + 3*A^2*C*b^11*c^10*f - 3*A*C^2*b^11*c^10*f - A*B^2*b^11*c^10*f + 36*
A^3*b^11*d^10*f - A^3*b^11*c^10*f + A^3*b^11*c^8*d^2*f + A^3*a^8*b^3*d^10*f
+ B^2*C*b^11*c^10*f + B*C^2*a^11*d^10*f + A^2*B*a^11*d^10*f + C^3*b^11*c^1
0*f + B^3*a^11*d^10*f - 6*A*B^2*C*a^7*b*c*d^7 + 4*A*B^2*C*a*b^7*c*d^7 + 168
*A^2*B*C*a^2*b^6*c^3*d^5 + 144*A*B*C^2*a^3*b^5*c^4*d^4 - 129*A^2*B*C*a^3*b^
5*c^4*d^4 - 96*A*B*C^2*a^2*b^6*c^3*d^5 + 84*A*B*C^2*a^3*b^5*c^2*d^6 + 72*A^
2*B*C*a^4*b^4*c^3*d^5 - 72*A^2*B*C*a^3*b^5*c^2*d^6 + 64*A*B^2*C*a^4*b^4*c^4
*d^4 - 60*A*B*C^2*a^4*b^4*c^3*d^5 + 57*A^2*B*C*a^5*b^3*c^2*d^6 - 56*A*B^2*C
*a^5*b^3*c^3*d^5 - 39*A*B^2*C*a^2*b^6*c^4*d^4 - 38*A*B^2*C*a^3*b^5*c^5*d^3
+ 36*A*B^2*C*a^3*b^5*c^3*d^5 + 36*A*B*C^2*a^5*b^3*c^4*d^4 - 30*A*B*C^2*a^5*
b^3*c^2*d^6 + 27*A*B^2*C*a^6*b^2*c^2*d^6 - 24*A*B^2*C*a^2*b^6*c^2*d^6 + 24*
A*B*C^2*a^6*b^2*c^3*d^5 - 24*A*B*C^2*a^4*b^4*c^5*d^3 - 18*A^2*B*C*a^5*b^3*c
^4*d^4 + 18*A^2*B*C*a^2*b^6*c^5*d^3 - 15*A*B^2*C*a^4*b^4*c^2*d^6 - 12*A^2*B
*C*a^6*b^2*c^3*d^5 + 12*A^2*B*C*a^4*b^4*c^5*d^3 + 9*A*B^2*C*a^2*b^6*c^6*d^2
+ 6*A*B*C^2*a^3*b^5*c^6*d^2 - 3*A^2*B*C*a^3*b^5*c^6*d^2 + 60*A^2*B*C*a^2*b
^6*c*d^7 - 51*A^2*B*C*a*b^7*c^4*d^4 + 48*A*B*C^2*a^6*b^2*c*d^7 - 42*A^2*B*C
*a^6*b^2*c*d^7 - 42*A^2*B*C*a*b^7*c^2*d^6 + 36*A*B*C^2*a^4*b^4*c*d^7 + 36*A
*B*C^2*a*b^7*c^4*d^4 + 36*A*B*C^2*a*b^7*c^2*d^6 - 30*A^2*B*C*a^4*b^4*c*d^7
+ 24*A*B^2*C*a^3*b^5*c*d^7 - 24*A*B*C^2*a^2*b^6*c*d^7 + 18*A*B^2*C*a*b^7*c^
5*d^3 - 18*A*B*C^2*a*b^7*c^6*d^2 + 12*A*B^2*C*a*b^7*c^3*d^5 + 9*A^2*B*C*a*b
^7*c^6*d^2 + 6*A*B^2*C*a^5*b^3*c*d^7 - 6*A*B*C^2*a^7*b*c^2*d^6 + 3*A^2*B*C*

a^7*b*c^2*d^6 - 18*B^3*C*a^6*b^2*c*d^7 - 18*B*C^3*a^6*b^2*c*d^7 - 14*B^3*C*
a^4*b^4*c*d^7 - 14*B*C^3*a^4*b^4*c*d^7 - 10*B^3*C*a*b^7*c^2*d^6 - 10*B*C^3*
a*b^7*c^2*d^6 + 9*B^3*C*a*b^7*c^6*d^2 + 9*B*C^3*a*b^7*c^6*d^2 - 7*B^3*C*a*b
^7*c^4*d^4 - 7*B*C^3*a*b^7*c^4*d^4 + 6*B^2*C^2*a^7*b*c*d^7 - 4*B^3*C*a^2*b^
6*c*d^7 + 4*B^2*C^2*a*b^7*c*d^7 - 4*B*C^3*a^2*b^6*c*d^7 + 3*B^3*C*a^7*b*c^2
*d^6 + 3*B*C^3*a^7*b*c^2*d^6 + 144*A^3*C*a^3*b^5*c*d^7 + 62*A^3*C*a^5*b^3*c
*d^7 + 48*A*C^3*a^3*b^5*c*d^7 - 36*A^2*C^2*a*b^7*c*d^7 + 26*A*C^3*a^5*b^3*c
*d^7 + 20*A^3*C*a*b^7*c^3*d^5 + 18*A^2*C^2*a^7*b*c*d^7 - 18*A*C^3*a*b^7*c^5
*d^3 - 6*A^3*C*a*b^7*c^5*d^3 - 4*A*C^3*a*b^7*c^3*d^5 - 32*A^3*B*a^2*b^6*c*d
^7 - 32*A*B^3*a^2*b^6*c*d^7 + 22*A^3*B*a*b^7*c^4*d^4 + 22*A*B^3*a*b^7*c^4*d
^4 + 16*A^3*B*a*b^7*c^2*d^6 + 16*A*B^3*a*b^7*c^2*d^6 + 12*A^3*B*a^6*b^2*c*d
^7 + 12*A*B^3*a^6*b^2*c*d^7 + 8*A^3*B*a^4*b^4*c*d^7 - 8*A^2*B^2*a*b^7*c*d^7
+ 8*A*B^3*a^4*b^4*c*d^7 + 36*A^2*B*C*b^8*c^3*d^5 + 24*A*B*C^2*b^8*c^5*d^3
- 18*A^2*B*C*b^8*c^5*d^3 - 12*A*B*C^2*b^8*c^3*d^5 - 3*A*B^2*C*b^8*c^6*d^2 -
3*A*B^2*C*b^8*c^4*d^4 - 2*A*B^2*C*b^8*c^2*d^6 + 57*A^2*B*C*a^5*b^3*d^8 + 3
6*A^2*B*C*a^3*b^5*d^8 - 30*A*B*C^2*a^5*b^3*d^8 - 18*A*B*C^2*a^3*b^5*d^8 - 9
*A*B^2*C*a^4*b^4*d^8 - 3*A*B^2*C*a^6*b^2*d^8 - 2*A*B^2*C*a^2*b^6*d^8 + 34*B
^2*C^2*a^3*b^5*c^5*d^3 + 28*B^2*C^2*a^5*b^3*c^3*d^5 + 24*B^2*C^2*a^2*b^6*c^
4*d^4 - 20*B^2*C^2*a^4*b^4*c^4*d^4 + 12*B^2*C^2*a^3*b^5*c^3*d^5 + 12*B^2*C^
2*a^2*b^6*c^2*d^6 + 9*B^2*C^2*a^6*b^2*c^4*d^4 + 9*B^2*C^2*a^4*b^4*c^2*d^6 -
9*B^2*C^2*a^2*b^6*c^6*d^2 - 3*B^2*C^2*a^6*b^2*c^2*d^6 + 159*A^2*C^2*a^4*b^
4*c^2*d^6 - 156*A^2*C^2*a^3*b^5*c^3*d^5 + 90*A^2*C^2*a^3*b^5*c^5*d^3 + 78*A
^2*C^2*a^2*b^6*c^2*d^6 - 63*A^2*C^2*a^4*b^4*c^4*d^4 - 27*A^2*C^2*a^6*b^2*c^
2*d^6 - 27*A^2*C^2*a^2*b^6*c^6*d^2 - 18*A^2*C^2*a^2*b^6*c^4*d^4 + 9*A^2*C^2
*a^6*b^2*c^4*d^4 + 66*A^2*B^2*a^2*b^6*c^2*d^6 + 60*A^2*B^2*a^4*b^4*c^2*d^6
- 48*A^2*B^2*a^3*b^5*c^3*d^5 + 42*A^2*B^2*a^2*b^6*c^4*d^4 + 28*A^2*B^2*a^5*
b^3*c^3*d^5 - 17*A^2*B^2*a^4*b^4*c^4*d^4 - 6*A^2*B^2*a^6*b^2*c^2*d^6 + 4*A^
2*B^2*a^3*b^5*c^5*d^3 + 36*A^3*C*a*b^7*c*d^7 - 18*A*C^3*a^7*b*c*d^7 + 12*A*
C^3*a*b^7*c*d^7 - 6*A^3*C*a^7*b*c*d^7 + 24*A^2*B*C*b^8*c*d^7 - 12*A*B*C^2*b
^8*c*d^7 + 12*A^2*B*C*a*b^7*d^8 + 6*A*B*C^2*a^7*b*d^8 - 6*A*B*C^2*a*b^7*d^8
- 3*A^2*B*C*a^7*b*d^8 - 53*B^3*C*a^3*b^5*c^4*d^4 - 53*B*C^3*a^3*b^5*c^4*d^
4 - 32*B^3*C*a^3*b^5*c^2*d^6 - 32*B*C^3*a^3*b^5*c^2*d^6 - 18*B^3*C*a^5*b^3*
c^4*d^4 - 18*B*C^3*a^5*b^3*c^4*d^4 + 16*B^3*C*a^4*b^4*c^3*d^5 + 16*B*C^3*a^
4*b^4*c^3*d^5 - 12*B^3*C*a^6*b^2*c^3*d^5 + 12*B^3*C*a^4*b^4*c^5*d^3 + 12*B^
2*C^2*a^3*b^5*c*d^7 - 12*B*C^3*a^6*b^2*c^3*d^5 + 12*B*C^3*a^4*b^4*c^5*d^3 +
8*B^3*C*a^2*b^6*c^3*d^5 + 8*B*C^3*a^2*b^6*c^3*d^5 - 6*B^3*C*a^2*b^6*c^5*d^
3 + 6*B^2*C^2*a^5*b^3*c*d^7 - 6*B^2*C^2*a*b^7*c^5*d^3 - 6*B*C^3*a^2*b^6*c^5
*d^3 - 3*B^3*C*a^3*b^5*c^6*d^2 - 3*B*C^3*a^3*b^5*c^6*d^2 - 175*A^3*C*a^4*b^
4*c^2*d^6 + 164*A^3*C*a^3*b^5*c^3*d^5 - 144*A^2*C^2*a^3*b^5*c*d^7 - 124*A^3
*C*a^2*b^6*c^2*d^6 - 90*A*C^3*a^3*b^5*c^5*d^3 - 73*A*C^3*a^4*b^4*c^2*d^6 -
66*A^2*C^2*a^5*b^3*c*d^7 + 44*A*C^3*a^3*b^5*c^3*d^5 + 36*A*C^3*a^4*b^4*c^4*
d^4 + 30*A^3*C*a^4*b^4*c^4*d^4 - 30*A^3*C*a^3*b^5*c^5*d^3 + 27*A*C^3*a^2*b^
6*c^6*d^2 + 21*A*C^3*a^2*b^6*c^4*d^4 + 18*A^2*C^2*a*b^7*c^5*d^3 - 18*A*C^3*
a^6*b^2*c^4*d^4 - 16*A*C^3*a^2*b^6*c^2*d^6 + 15*A^3*C*a^6*b^2*c^2*d^6 - 15*
A^3*C*a^2*b^6*c^4*d^4 - 12*A^2*C^2*a*b^7*c^3*d^5 + 9*A^3*C*a^2*b^6*c^6*d^2

+ 9*A*C^3*a^6*b^2*c^2*d^6 - 80*A^3*B*a^2*b^6*c^3*d^5 - 80*A*B^3*a^2*b^6*c^3
*d^5 + 38*A^3*B*a^3*b^5*c^4*d^4 + 38*A*B^3*a^3*b^5*c^4*d^4 - 36*A^2*B^2*a^3
*b^5*c*d^7 - 28*A^3*B*a^5*b^3*c^2*d^6 - 28*A^3*B*a^4*b^4*c^3*d^5 - 28*A*B^3
*a^5*b^3*c^2*d^6 - 28*A*B^3*a^4*b^4*c^3*d^5 + 20*A^3*B*a^3*b^5*c^2*d^6 + 20
*A*B^3*a^3*b^5*c^2*d^6 - 12*A^3*B*a^2*b^6*c^5*d^3 - 12*A^2*B^2*a^5*b^3*c*d^
7 - 12*A^2*B^2*a*b^7*c^5*d^3 - 12*A^2*B^2*a*b^7*c^3*d^5 - 12*A*B^3*a^2*b^6*
c^5*d^3 + 9*B^2*C^2*b^8*c^4*d^4 + 4*B^2*C^2*b^8*c^2*d^6 + 3*B^2*C^2*b^8*c^6
*d^2 - 30*A^2*C^2*b^8*c^4*d^4 + 9*A^2*C^2*b^8*c^6*d^2 + 16*A^2*B^2*b^8*c^2*
d^6 + 6*B^2*C^2*a^6*b^2*d^8 + 3*B^2*C^2*a^4*b^4*d^8 + 3*A^2*B^2*b^8*c^4*d^4
 + 36*A^2*C^2*a^4*b^4*d^8 + 27*A^2*C^2*a^2*b^6*d^8 - 18*A^2*C^2*a^6*b^2*d^8
 + 33*A^2*B^2*a^4*b^4*d^8 + 28*A^2*B^2*a^2*b^6*d^8 + 6*A^2*B^2*a^6*b^2*d^8
+ 6*C^4*a*b^7*c^5*d^3 + 4*C^4*a*b^7*c^3*d^5 - 2*C^4*a^5*b^3*c*d^7 + 12*B^4*
a^3*b^5*c*d^7 - 12*B^4*a*b^7*c^5*d^3 + 8*B^4*a^5*b^3*c*d^7 - 4*B^4*a*b^7*c^
3*d^5 - 48*A^4*a^3*b^5*c*d^7 - 20*A^4*a^5*b^3*c*d^7 - 8*A^4*a*b^7*c^3*d^5 -
 10*B^3*C*b^8*c^5*d^3 - 10*B*C^3*b^8*c^5*d^3 - 4*B^3*C*b^8*c^3*d^5 - 4*B*C^
3*b^8*c^3*d^5 + 23*A^3*C*b^8*c^4*d^4 - 18*A^3*C*b^8*c^2*d^6 + 11*A*C^3*b^8*
c^4*d^4 - 9*A*C^3*b^8*c^6*d^2 + 6*A*C^3*b^8*c^2*d^6 - 3*A^3*C*b^8*c^6*d^2 -
 20*A^3*B*b^8*c^3*d^5 - 20*A*B^3*b^8*c^3*d^5 + 4*A^3*B*b^8*c^5*d^3 + 4*A*B^
3*b^8*c^5*d^3 - 63*A^3*C*a^4*b^4*d^8 - 54*A^3*C*a^2*b^6*d^8 + 9*A^3*C*a^6*b
^2*d^8 + 9*A*C^3*a^6*b^2*d^8 - 3*A*C^3*a^4*b^4*d^8 - 28*A^3*B*a^5*b^3*d^8 -
 28*A*B^3*a^5*b^3*d^8 - 18*A^3*B*a^3*b^5*d^8 - 18*A*B^3*a^3*b^5*d^8 + B^3*C
*a^5*b^3*c^2*d^6 + B*C^3*a^5*b^3*c^2*d^6 + 6*C^4*a^7*b*c*d^7 + 4*B^4*a*b^7*
c*d^7 - 12*A^4*a*b^7*c*d^7 - 12*A^3*B*b^8*c*d^7 - 12*A*B^3*b^8*c*d^7 - 3*B^
3*C*a^7*b*d^8 - 3*B*C^3*a^7*b*d^8 - 6*A^3*B*a*b^7*d^8 - 6*A*B^3*a*b^7*d^8 +
 30*C^4*a^3*b^5*c^5*d^3 + 19*C^4*a^4*b^4*c^2*d^6 + 9*C^4*a^6*b^2*c^4*d^4 -
9*C^4*a^2*b^6*c^6*d^2 + 4*C^4*a^3*b^5*c^3*d^5 + 4*C^4*a^2*b^6*c^2*d^6 + 3*C
^4*a^6*b^2*c^2*d^6 - 3*C^4*a^4*b^4*c^4*d^4 - 3*C^4*a^2*b^6*c^4*d^4 + 28*B^4
*a^5*b^3*c^3*d^5 + 27*B^4*a^2*b^6*c^4*d^4 - 17*B^4*a^4*b^4*c^4*d^4 - 10*B^4
*a^4*b^4*c^2*d^6 + 8*B^4*a^3*b^5*c^3*d^5 + 8*B^4*a^2*b^6*c^2*d^6 - 6*B^4*a^
6*b^2*c^2*d^6 + 4*B^4*a^3*b^5*c^5*d^3 + 70*A^4*a^4*b^4*c^2*d^6 + 58*A^4*a^2
*b^6*c^2*d^6 - 56*A^4*a^3*b^5*c^3*d^5 + 15*A^4*a^2*b^6*c^4*d^4 + B^2*C^2*a^
2*b^6*d^8 - 18*A^3*C*b^8*d^8 + B^3*C*a^5*b^3*d^8 + B*C^3*a^5*b^3*d^8 + 3*C^
4*b^8*c^6*d^2 + 8*B^4*b^8*c^4*d^4 + 4*B^4*b^8*c^2*d^6 + 12*A^4*b^8*c^2*d^6
- 5*A^4*b^8*c^4*d^4 + 6*B^4*a^6*b^2*d^8 + 3*B^4*a^4*b^4*d^8 + 30*A^4*a^4*b^
4*d^8 + 27*A^4*a^2*b^6*d^8 + 9*A^2*C^2*b^8*d^8 + 9*A^2*B^2*b^8*d^8 + 9*A^4*
b^8*d^8 + C^4*b^8*c^4*d^4 + B^4*a^2*b^6*d^8, f, k)*(root(640*a^13*b^7*c*d^1
5*f^4 + 640*a^7*b^13*c^15*d*f^4 + 480*a^15*b^5*c*d^15*f^4 + 480*a^11*b^9*c*
d^15*f^4 + 480*a^9*b^11*c^15*d*f^4 + 480*a^5*b^15*c^15*d*f^4 + 192*a^19*b*c
^5*d^11*f^4 + 192*a^17*b^3*c*d^15*f^4 + 192*a^11*b^9*c^15*d*f^4 + 192*a^9*b
^11*c*d^15*f^4 + 192*a^3*b^17*c^15*d*f^4 + 192*a*b^19*c^11*d^5*f^4 + 128*a^
19*b*c^7*d^9*f^4 + 128*a^19*b*c^3*d^13*f^4 + 128*a*b^19*c^13*d^3*f^4 + 128*
a*b^19*c^9*d^7*f^4 + 32*a^19*b*c^9*d^7*f^4 + 32*a^13*b^7*c^15*d*f^4 + 32*a^
7*b^13*c*d^15*f^4 + 32*a*b^19*c^7*d^9*f^4 + 32*a^19*b*c*d^15*f^4 + 32*a*b^1
9*c^15*d*f^4 - 47088*a^10*b^10*c^8*d^8*f^4 + 42432*a^11*b^9*c^7*d^9*f^4 + 4
2432*a^9*b^11*c^9*d^7*f^4 + 39328*a^11*b^9*c^9*d^7*f^4 + 39328*a^9*b^11*c^7

*d^9*f^4 - 36912*a^12*b^8*c^8*d^8*f^4 - 36912*a^8*b^12*c^8*d^8*f^4 - 34256*
a^10*b^10*c^10*d^6*f^4 - 34256*a^10*b^10*c^6*d^10*f^4 - 31152*a^12*b^8*c^6*
d^10*f^4 - 31152*a^8*b^12*c^10*d^6*f^4 + 28128*a^13*b^7*c^7*d^9*f^4 + 28128
*a^7*b^13*c^9*d^7*f^4 + 24160*a^11*b^9*c^5*d^11*f^4 + 24160*a^9*b^11*c^11*d
^5*f^4 - 23088*a^12*b^8*c^10*d^6*f^4 - 23088*a^8*b^12*c^6*d^10*f^4 + 22272*
a^13*b^7*c^9*d^7*f^4 + 22272*a^7*b^13*c^7*d^9*f^4 + 19072*a^11*b^9*c^11*d^5
*f^4 + 19072*a^9*b^11*c^5*d^11*f^4 + 18624*a^13*b^7*c^5*d^11*f^4 + 18624*a^
7*b^13*c^11*d^5*f^4 - 17328*a^14*b^6*c^8*d^8*f^4 - 17328*a^6*b^14*c^8*d^8*f
^4 - 17232*a^14*b^6*c^6*d^10*f^4 - 17232*a^6*b^14*c^10*d^6*f^4 - 13520*a^12
*b^8*c^4*d^12*f^4 - 13520*a^8*b^12*c^12*d^4*f^4 - 12464*a^10*b^10*c^12*d^4*
f^4 - 12464*a^10*b^10*c^4*d^12*f^4 + 10880*a^15*b^5*c^7*d^9*f^4 + 10880*a^5
*b^15*c^9*d^7*f^4 - 9072*a^14*b^6*c^10*d^6*f^4 - 9072*a^6*b^14*c^6*d^10*f^4
 + 8928*a^13*b^7*c^11*d^5*f^4 + 8928*a^7*b^13*c^5*d^11*f^4 - 8880*a^14*b^6*
c^4*d^12*f^4 - 8880*a^6*b^14*c^12*d^4*f^4 + 8480*a^15*b^5*c^5*d^11*f^4 + 84
80*a^5*b^15*c^11*d^5*f^4 + 7200*a^15*b^5*c^9*d^7*f^4 + 7200*a^5*b^15*c^7*d^
9*f^4 - 6912*a^12*b^8*c^12*d^4*f^4 - 6912*a^8*b^12*c^4*d^12*f^4 + 6400*a^11
*b^9*c^3*d^13*f^4 + 6400*a^9*b^11*c^13*d^3*f^4 + 5920*a^13*b^7*c^3*d^13*f^4
 + 5920*a^7*b^13*c^13*d^3*f^4 - 5392*a^16*b^4*c^6*d^10*f^4 - 5392*a^4*b^16*
c^10*d^6*f^4 - 4428*a^16*b^4*c^8*d^8*f^4 - 4428*a^4*b^16*c^8*d^8*f^4 + 4128
*a^11*b^9*c^13*d^3*f^4 + 4128*a^9*b^11*c^3*d^13*f^4 - 3328*a^16*b^4*c^4*d^1
2*f^4 - 3328*a^4*b^16*c^12*d^4*f^4 + 3264*a^15*b^5*c^3*d^13*f^4 + 3264*a^5*
b^15*c^13*d^3*f^4 - 2480*a^12*b^8*c^2*d^14*f^4 - 2480*a^8*b^12*c^14*d^2*f^4
 + 2240*a^15*b^5*c^11*d^5*f^4 + 2240*a^5*b^15*c^5*d^11*f^4 - 2128*a^14*b^6*
c^12*d^4*f^4 - 2128*a^6*b^14*c^4*d^12*f^4 + 2112*a^17*b^3*c^7*d^9*f^4 + 211
2*a^3*b^17*c^9*d^7*f^4 + 2048*a^17*b^3*c^5*d^11*f^4 + 2048*a^3*b^17*c^11*d^
5*f^4 - 2000*a^14*b^6*c^2*d^14*f^4 - 2000*a^6*b^14*c^14*d^2*f^4 - 1792*a^16
*b^4*c^10*d^6*f^4 - 1792*a^4*b^16*c^6*d^10*f^4 - 1776*a^10*b^10*c^14*d^2*f^
4 - 1776*a^10*b^10*c^2*d^14*f^4 + 1472*a^13*b^7*c^13*d^3*f^4 + 1472*a^7*b^1
3*c^3*d^13*f^4 + 1088*a^17*b^3*c^9*d^7*f^4 + 1088*a^3*b^17*c^7*d^9*f^4 + 99
2*a^17*b^3*c^3*d^13*f^4 + 992*a^3*b^17*c^13*d^3*f^4 - 912*a^16*b^4*c^2*d^14
*f^4 - 912*a^4*b^16*c^14*d^2*f^4 - 768*a^18*b^2*c^6*d^10*f^4 - 768*a^2*b^18
*c^10*d^6*f^4 - 688*a^12*b^8*c^14*d^2*f^4 - 688*a^8*b^12*c^2*d^14*f^4 - 592
*a^18*b^2*c^4*d^12*f^4 - 592*a^2*b^18*c^12*d^4*f^4 - 472*a^18*b^2*c^8*d^8*f
^4 - 472*a^2*b^18*c^8*d^8*f^4 - 280*a^16*b^4*c^12*d^4*f^4 - 280*a^4*b^16*c^
4*d^12*f^4 + 224*a^17*b^3*c^11*d^5*f^4 + 224*a^15*b^5*c^13*d^3*f^4 + 224*a^
5*b^15*c^3*d^13*f^4 + 224*a^3*b^17*c^5*d^11*f^4 - 208*a^18*b^2*c^2*d^14*f^4
 - 208*a^2*b^18*c^14*d^2*f^4 - 112*a^18*b^2*c^10*d^6*f^4 - 112*a^14*b^6*c^1
4*d^2*f^4 - 112*a^6*b^14*c^2*d^14*f^4 - 112*a^2*b^18*c^6*d^10*f^4 - 24*b^20
*c^12*d^4*f^4 - 16*b^20*c^14*d^2*f^4 - 16*b^20*c^10*d^6*f^4 - 4*b^20*c^8*d^
8*f^4 - 24*a^20*c^4*d^12*f^4 - 16*a^20*c^6*d^10*f^4 - 16*a^20*c^2*d^14*f^4
 - 4*a^20*c^8*d^8*f^4 - 80*a^14*b^6*d^16*f^4 - 60*a^16*b^4*d^16*f^4 - 60*a^1
2*b^8*d^16*f^4 - 24*a^18*b^2*d^16*f^4 - 24*a^10*b^10*d^16*f^4 - 4*a^8*b^12*
d^16*f^4 - 80*a^6*b^14*c^16*f^4 - 60*a^8*b^12*c^16*f^4 - 60*a^4*b^16*c^16*f
^4 - 24*a^10*b^10*c^16*f^4 - 24*a^2*b^18*c^16*f^4 - 4*a^12*b^8*c^16*f^4 - 4
*b^20*c^16*f^4 - 4*a^20*d^16*f^4 + 56*A*C*a^13*b*c*d^11*f^2 - 48*A*C*a*b^13

*c^11*d*f^2 + 48*A*C*a*b^13*c*d^11*f^2 + 5904*B*C*a^7*b^7*c^6*d^6*f^2 - 501
6*B*C*a^8*b^6*c^5*d^7*f^2 - 4608*B*C*a^6*b^8*c^7*d^5*f^2 - 4512*B*C*a^6*b^8
*c^5*d^7*f^2 - 4384*B*C*a^8*b^6*c^7*d^5*f^2 + 3056*B*C*a^7*b^7*c^8*d^4*f^2
+ 2256*B*C*a^7*b^7*c^4*d^8*f^2 - 1824*B*C*a^8*b^6*c^3*d^9*f^2 + 1632*B*C*a^
4*b^10*c^9*d^3*f^2 - 1400*B*C*a^3*b^11*c^8*d^4*f^2 - 1320*B*C*a^11*b^3*c^4*
d^8*f^2 - 1248*B*C*a^6*b^8*c^3*d^9*f^2 + 1152*B*C*a^10*b^4*c^3*d^9*f^2 - 10
72*B*C*a^6*b^8*c^9*d^3*f^2 + 1068*B*C*a^9*b^5*c^6*d^6*f^2 - 1004*B*C*a^5*b^
9*c^4*d^8*f^2 - 968*B*C*a^3*b^11*c^6*d^6*f^2 - 864*B*C*a^5*b^9*c^8*d^4*f^2
- 828*B*C*a^9*b^5*c^4*d^8*f^2 - 792*B*C*a^11*b^3*c^2*d^10*f^2 - 792*B*C*a^3
*b^11*c^4*d^8*f^2 - 776*B*C*a^8*b^6*c^9*d^3*f^2 + 688*B*C*a^4*b^10*c^7*d^5*
f^2 - 672*B*C*a^3*b^11*c^10*d^2*f^2 - 592*B*C*a^9*b^5*c^2*d^10*f^2 + 544*B*
C*a^7*b^7*c^10*d^2*f^2 - 492*B*C*a^5*b^9*c^2*d^10*f^2 + 480*B*C*a^10*b^4*c^
5*d^7*f^2 - 392*B*C*a^5*b^9*c^10*d^2*f^2 + 332*B*C*a^9*b^5*c^8*d^4*f^2 - 32
8*B*C*a^11*b^3*c^6*d^6*f^2 + 320*B*C*a^2*b^12*c^9*d^3*f^2 + 272*B*C*a^12*b^
2*c^3*d^9*f^2 - 248*B*C*a^4*b^10*c^5*d^7*f^2 - 248*B*C*a^3*b^11*c^2*d^10*f^
2 - 208*B*C*a^10*b^4*c^7*d^5*f^2 - 192*B*C*a^2*b^12*c^5*d^7*f^2 + 144*B*C*a
^7*b^7*c^2*d^10*f^2 - 96*B*C*a^4*b^10*c^3*d^9*f^2 + 88*B*C*a^12*b^2*c^5*d^7
*f^2 - 72*B*C*a^11*b^3*c^8*d^4*f^2 - 48*B*C*a^12*b^2*c^7*d^5*f^2 + 48*B*C*a
^10*b^4*c^9*d^3*f^2 - 48*B*C*a^2*b^12*c^7*d^5*f^2 - 48*B*C*a^2*b^12*c^3*d^9
*f^2 - 12*B*C*a^9*b^5*c^10*d^2*f^2 + 4*B*C*a^5*b^9*c^6*d^6*f^2 + 5824*A*C*a
^5*b^9*c^7*d^5*f^2 - 4378*A*C*a^6*b^8*c^8*d^4*f^2 + 4296*A*C*a^5*b^9*c^5*d^
7*f^2 - 3912*A*C*a^6*b^8*c^6*d^6*f^2 - 3672*A*C*a^9*b^5*c^5*d^7*f^2 + 3594*
A*C*a^8*b^6*c^4*d^8*f^2 + 3236*A*C*a^8*b^6*c^6*d^6*f^2 + 2816*A*C*a^5*b^9*c
^9*d^3*f^2 + 2624*A*C*a^5*b^9*c^3*d^9*f^2 + 2432*A*C*a^7*b^7*c^7*d^5*f^2 -
2366*A*C*a^4*b^10*c^8*d^4*f^2 + 2298*A*C*a^10*b^4*c^4*d^8*f^2 + 1872*A*C*a^
7*b^7*c^3*d^9*f^2 + 1848*A*C*a^10*b^4*c^6*d^6*f^2 - 1644*A*C*a^4*b^10*c^6*d
^6*f^2 - 1488*A*C*a^9*b^5*c^7*d^5*f^2 - 1408*A*C*a^9*b^5*c^3*d^9*f^2 - 1308
*A*C*a^6*b^8*c^4*d^8*f^2 + 1248*A*C*a^7*b^7*c^5*d^7*f^2 - 1012*A*C*a^6*b^8*
c^10*d^2*f^2 + 1008*A*C*a^3*b^11*c^7*d^5*f^2 + 992*A*C*a^3*b^11*c^5*d^7*f^2
+ 928*A*C*a^3*b^11*c^3*d^9*f^2 + 848*A*C*a^7*b^7*c^9*d^3*f^2 + 636*A*C*a^8
*b^6*c^2*d^10*f^2 - 628*A*C*a^4*b^10*c^10*d^2*f^2 - 600*A*C*a^6*b^8*c^2*d^1
0*f^2 - 576*A*C*a^11*b^3*c^5*d^7*f^2 + 572*A*C*a^10*b^4*c^2*d^10*f^2 + 464*
A*C*a^8*b^6*c^8*d^4*f^2 - 304*A*C*a^4*b^10*c^4*d^8*f^2 + 304*A*C*a^2*b^12*c
^6*d^6*f^2 + 296*A*C*a^2*b^12*c^4*d^8*f^2 + 260*A*C*a^10*b^4*c^8*d^4*f^2 -
232*A*C*a^12*b^2*c^2*d^10*f^2 - 232*A*C*a^9*b^5*c^9*d^3*f^2 + 228*A*C*a^2*b
^12*c^10*d^2*f^2 - 188*A*C*a^4*b^10*c^2*d^10*f^2 + 144*A*C*a^11*b^3*c^3*d^9
*f^2 + 116*A*C*a^12*b^2*c^6*d^6*f^2 - 112*A*C*a^11*b^3*c^7*d^5*f^2 + 112*A*
C*a^3*b^11*c^9*d^3*f^2 + 92*A*C*a^8*b^6*c^10*d^2*f^2 + 74*A*C*a^12*b^2*c^4*
d^8*f^2 + 62*A*C*a^2*b^12*c^8*d^4*f^2 + 40*A*C*a^2*b^12*c^2*d^10*f^2 - 7008
*A*B*a^7*b^7*c^6*d^6*f^2 - 4032*A*B*a^7*b^7*c^4*d^8*f^2 + 3952*A*B*a^8*b^6*
c^7*d^5*f^2 + 3648*A*B*a^8*b^6*c^5*d^7*f^2 - 3392*A*B*a^7*b^7*c^8*d^4*f^2 +
3264*A*B*a^6*b^8*c^7*d^5*f^2 - 2992*A*B*a^4*b^10*c^5*d^7*f^2 - 2368*A*B*a^
4*b^10*c^7*d^5*f^2 - 2304*A*B*a^4*b^10*c^3*d^9*f^2 - 1968*A*B*a^9*b^5*c^6*d
^6*f^2 - 1872*A*B*a^4*b^10*c^9*d^3*f^2 - 1728*A*B*a^7*b^7*c^2*d^10*f^2 + 17
12*A*B*a^3*b^11*c^8*d^4*f^2 - 1536*A*B*a^10*b^4*c^3*d^9*f^2 + 1536*A*B*a^6*

b^8*c^5*d^7*f^2 - 1392*A*B*a^2*b^12*c^5*d^7*f^2 + 1328*A*B*a^3*b^11*c^6*d^6
*f^2 - 1104*A*B*a^2*b^12*c^3*d^9*f^2 - 1056*A*B*a^6*b^8*c^3*d^9*f^2 + 976*A
*B*a^6*b^8*c^9*d^3*f^2 + 960*A*B*a^11*b^3*c^4*d^8*f^2 + 936*A*B*a^5*b^9*c^8
*d^4*f^2 - 912*A*B*a^10*b^4*c^5*d^7*f^2 + 848*A*B*a^8*b^6*c^9*d^3*f^2 + 816
*A*B*a^3*b^11*c^4*d^8*f^2 - 816*A*B*a^2*b^12*c^7*d^5*f^2 + 768*A*B*a^3*b^11
*c^10*d^2*f^2 + 672*A*B*a^8*b^6*c^3*d^9*f^2 - 632*A*B*a^9*b^5*c^8*d^4*f^2 -
608*A*B*a^9*b^5*c^2*d^10*f^2 - 552*A*B*a^9*b^5*c^4*d^8*f^2 - 544*A*B*a^7*b
^7*c^10*d^2*f^2 - 480*A*B*a^5*b^9*c^2*d^10*f^2 + 464*A*B*a^5*b^9*c^10*d^2*f
^2 - 464*A*B*a^2*b^12*c^9*d^3*f^2 + 432*A*B*a^11*b^3*c^2*d^10*f^2 - 368*A*B
*a^12*b^2*c^3*d^9*f^2 - 256*A*B*a^5*b^9*c^6*d^6*f^2 - 208*A*B*a^12*b^2*c^5*
d^7*f^2 + 176*A*B*a^5*b^9*c^4*d^8*f^2 + 112*A*B*a^11*b^3*c^6*d^6*f^2 + 112*
A*B*a^10*b^4*c^7*d^5*f^2 - 16*A*B*a^3*b^11*c^2*d^10*f^2 - 576*B*C*a^8*b^6*c
*d^11*f^2 + 400*B*C*a^4*b^10*c^11*d*f^2 - 288*B*C*a^6*b^8*c*d^11*f^2 - 176*
B*C*a^6*b^8*c^11*d*f^2 + 128*B*C*a^10*b^4*c*d^11*f^2 - 108*B*C*a*b^13*c^4*d
^8*f^2 - 104*B*C*a^4*b^10*c*d^11*f^2 - 92*B*C*a^13*b*c^4*d^8*f^2 - 60*B*C*a
*b^13*c^8*d^4*f^2 - 60*B*C*a*b^13*c^6*d^6*f^2 + 48*B*C*a^2*b^12*c^11*d*f^2
- 40*B*C*a*b^13*c^2*d^10*f^2 - 28*B*C*a^13*b*c^2*d^10*f^2 - 24*B*C*a^12*b^2
*c*d^11*f^2 + 20*B*C*a*b^13*c^10*d^2*f^2 - 16*B*C*a^2*b^12*c*d^11*f^2 + 12*
B*C*a^13*b*c^6*d^6*f^2 + 912*A*C*a^7*b^7*c*d^11*f^2 + 808*A*C*a^5*b^9*c*d^1
1*f^2 + 432*A*C*a^5*b^9*c^11*d*f^2 + 336*A*C*a^3*b^11*c*d^11*f^2 + 224*A*C*
a^11*b^3*c*d^11*f^2 - 112*A*C*a^3*b^11*c^11*d*f^2 + 112*A*C*a*b^13*c^3*d^9*
f^2 - 88*A*C*a*b^13*c^9*d^3*f^2 + 80*A*C*a^13*b*c^3*d^9*f^2 + 56*A*C*a*b^13
*c^5*d^7*f^2 + 48*A*C*a^9*b^5*c*d^11*f^2 - 40*A*C*a^13*b*c^5*d^7*f^2 - 16*A
*C*a^7*b^7*c^11*d*f^2 + 16*A*C*a*b^13*c^7*d^5*f^2 - 496*A*B*a^4*b^10*c*d^11
*f^2 - 400*A*B*a^4*b^10*c^11*d*f^2 + 288*A*B*a^8*b^6*c*d^11*f^2 - 288*A*B*a
^6*b^8*c*d^11*f^2 - 272*A*B*a^2*b^12*c*d^11*f^2 + 240*A*B*a*b^13*c^6*d^6*f^
2 - 224*A*B*a^10*b^4*c*d^11*f^2 + 192*A*B*a*b^13*c^8*d^4*f^2 + 192*A*B*a*b^
13*c^4*d^8*f^2 + 176*A*B*a^6*b^8*c^11*d*f^2 + 104*A*B*a^13*b*c^4*d^8*f^2 -
48*A*B*a^2*b^12*c^11*d*f^2 + 16*A*B*a^13*b*c^2*d^10*f^2 + 16*A*B*a*b^13*c^1
0*d^2*f^2 + 16*A*B*a*b^13*c^2*d^10*f^2 - 96*B*C*b^14*c^7*d^5*f^2 - 72*B*C*b
^14*c^5*d^7*f^2 - 24*B*C*b^14*c^9*d^3*f^2 - 16*B*C*b^14*c^3*d^9*f^2 + 116*A
*C*b^14*c^6*d^6*f^2 + 100*A*C*b^14*c^4*d^8*f^2 + 24*A*C*b^14*c^2*d^10*f^2 +
22*A*C*b^14*c^8*d^4*f^2 + 16*B*C*a^14*c^3*d^9*f^2 + 8*A*C*b^14*c^10*d^2*f^
2 - 192*A*B*b^14*c^5*d^7*f^2 - 176*A*B*b^14*c^3*d^9*f^2 - 112*B*C*a^11*b^3*
d^12*f^2 - 48*A*B*b^14*c^7*d^5*f^2 - 28*A*C*a^14*c^2*d^10*f^2 + 4*B*C*a^5*b
^9*d^12*f^2 + 2*A*C*a^14*c^4*d^8*f^2 + 150*A*C*a^10*b^4*d^12*f^2 - 80*B*C*a
^3*b^11*c^12*f^2 + 66*A*C*a^8*b^6*d^12*f^2 - 30*A*C*a^12*b^2*d^12*f^2 + 24*
B*C*a^5*b^9*c^12*f^2 - 16*A*B*a^14*c^3*d^9*f^2 - 12*A*C*a^4*b^10*d^12*f^2 -
576*A*B*a^7*b^7*d^12*f^2 - 432*A*B*a^9*b^5*d^12*f^2 - 400*A*B*a^5*b^9*d^12
*f^2 - 144*A*B*a^3*b^11*d^12*f^2 - 66*A*C*a^4*b^10*c^12*f^2 + 54*A*C*a^2*b^
12*c^12*f^2 - 32*A*B*a^11*b^3*d^12*f^2 + 2*A*C*a^6*b^8*c^12*f^2 + 80*A*B*a^
3*b^11*c^12*f^2 - 24*A*B*a^5*b^9*c^12*f^2 + 2508*C^2*a^6*b^8*c^6*d^6*f^2 +
2376*C^2*a^9*b^5*c^5*d^7*f^2 + 2357*C^2*a^6*b^8*c^8*d^4*f^2 - 2048*C^2*a^5*
b^9*c^7*d^5*f^2 + 1304*C^2*a^9*b^5*c^3*d^9*f^2 + 1303*C^2*a^4*b^10*c^8*d^4*
f^2 + 1212*C^2*a^4*b^10*c^6*d^6*f^2 - 1203*C^2*a^8*b^6*c^4*d^8*f^2 - 1192*C

^2*a^5*b^9*c^9*d^3*f^2 + 1062*C^2*a^6*b^8*c^4*d^8*f^2 + 984*C^2*a^9*b^5*c^7
*d^5*f^2 - 952*C^2*a^8*b^6*c^6*d^6*f^2 + 768*C^2*a^7*b^7*c^5*d^7*f^2 - 681*
C^2*a^10*b^4*c^4*d^8*f^2 - 672*C^2*a^5*b^9*c^5*d^7*f^2 - 480*C^2*a^10*b^4*c
^6*d^6*f^2 + 458*C^2*a^6*b^8*c^10*d^2*f^2 - 448*C^2*a^7*b^7*c^7*d^5*f^2 + 4
22*C^2*a^4*b^10*c^4*d^8*f^2 + 372*C^2*a^6*b^8*c^2*d^10*f^2 + 360*C^2*a^11*b
^3*c^5*d^7*f^2 + 312*C^2*a^7*b^7*c^3*d^9*f^2 + 278*C^2*a^4*b^10*c^10*d^2*f^
2 - 232*C^2*a^7*b^7*c^9*d^3*f^2 + 194*C^2*a^12*b^2*c^2*d^10*f^2 + 176*C^2*a
^9*b^5*c^9*d^3*f^2 + 152*C^2*a^3*b^11*c^5*d^7*f^2 + 124*C^2*a^4*b^10*c^2*d^
10*f^2 - 120*C^2*a^3*b^11*c^7*d^5*f^2 - 114*C^2*a^2*b^12*c^10*d^2*f^2 - 102
*C^2*a^8*b^6*c^2*d^10*f^2 + 101*C^2*a^12*b^2*c^4*d^8*f^2 + 100*C^2*a^2*b^12
*c^6*d^6*f^2 - 88*C^2*a^5*b^9*c^3*d^9*f^2 + 77*C^2*a^2*b^12*c^8*d^4*f^2 + 7
2*C^2*a^11*b^3*c^3*d^9*f^2 - 64*C^2*a^8*b^6*c^10*d^2*f^2 + 64*C^2*a^3*b^11*
c^3*d^9*f^2 - 58*C^2*a^10*b^4*c^2*d^10*f^2 + 56*C^2*a^12*b^2*c^6*d^6*f^2 +
56*C^2*a^11*b^3*c^7*d^5*f^2 + 40*C^2*a^3*b^11*c^9*d^3*f^2 + 36*C^2*a^12*b^2
*c^8*d^4*f^2 + 32*C^2*a^2*b^12*c^4*d^8*f^2 + 26*C^2*a^10*b^4*c^8*d^4*f^2 +
16*C^2*a^2*b^12*c^2*d^10*f^2 + 2*C^2*a^8*b^6*c^8*d^4*f^2 + 2277*B^2*a^8*b^6
*c^4*d^8*f^2 + 2144*B^2*a^5*b^9*c^7*d^5*f^2 - 2112*B^2*a^9*b^5*c^5*d^7*f^2
+ 2028*B^2*a^8*b^6*c^6*d^6*f^2 - 1671*B^2*a^6*b^8*c^8*d^4*f^2 + 1275*B^2*a^
10*b^4*c^4*d^8*f^2 + 1176*B^2*a^5*b^9*c^5*d^7*f^2 + 1096*B^2*a^5*b^9*c^9*d^
3*f^2 - 1044*B^2*a^6*b^8*c^6*d^6*f^2 + 984*B^2*a^10*b^4*c^6*d^6*f^2 - 968*B
^2*a^9*b^5*c^3*d^9*f^2 - 888*B^2*a^9*b^5*c^7*d^5*f^2 + 672*B^2*a^7*b^7*c^7*
d^5*f^2 + 664*B^2*a^5*b^9*c^3*d^9*f^2 - 649*B^2*a^4*b^10*c^8*d^4*f^2 + 618*
B^2*a^8*b^6*c^2*d^10*f^2 + 514*B^2*a^4*b^10*c^4*d^8*f^2 + 460*B^2*a^2*b^12*
c^6*d^6*f^2 + 422*B^2*a^8*b^6*c^8*d^4*f^2 + 406*B^2*a^10*b^4*c^2*d^10*f^2 -
382*B^2*a^6*b^8*c^10*d^2*f^2 + 368*B^2*a^2*b^12*c^4*d^8*f^2 - 312*B^2*a^11
*b^3*c^5*d^7*f^2 + 312*B^2*a^7*b^7*c^3*d^9*f^2 + 248*B^2*a^7*b^7*c^9*d^3*f^
2 + 245*B^2*a^2*b^12*c^8*d^4*f^2 - 192*B^2*a^7*b^7*c^5*d^7*f^2 - 184*B^2*a^
3*b^11*c^9*d^3*f^2 + 182*B^2*a^2*b^12*c^10*d^2*f^2 + 176*B^2*a^3*b^11*c^3*d
^9*f^2 + 174*B^2*a^6*b^8*c^4*d^8*f^2 - 170*B^2*a^4*b^10*c^10*d^2*f^2 - 152*
B^2*a^9*b^5*c^9*d^3*f^2 + 152*B^2*a^4*b^10*c^2*d^10*f^2 + 142*B^2*a^10*b^4*
c^8*d^4*f^2 - 90*B^2*a^12*b^2*c^2*d^10*f^2 + 88*B^2*a^2*b^12*c^2*d^10*f^2 +
84*B^2*a^8*b^6*c^10*d^2*f^2 + 84*B^2*a^6*b^8*c^2*d^10*f^2 + 60*B^2*a^12*b^
2*c^6*d^6*f^2 - 56*B^2*a^11*b^3*c^7*d^5*f^2 + 53*B^2*a^12*b^2*c^4*d^8*f^2 +
24*B^2*a^11*b^3*c^3*d^9*f^2 + 24*B^2*a^4*b^10*c^6*d^6*f^2 + 24*B^2*a^3*b^1
1*c^7*d^5*f^2 - 8*B^2*a^3*b^11*c^5*d^7*f^2 + 4566*A^2*a^6*b^8*c^4*d^8*f^2 +
4284*A^2*a^6*b^8*c^6*d^6*f^2 - 3776*A^2*a^5*b^9*c^7*d^5*f^2 - 3624*A^2*a^5
*b^9*c^5*d^7*f^2 + 3122*A^2*a^4*b^10*c^4*d^8*f^2 + 3108*A^2*a^6*b^8*c^2*d^1
0*f^2 + 2741*A^2*a^6*b^8*c^8*d^4*f^2 + 2592*A^2*a^4*b^10*c^6*d^6*f^2 - 2536
*A^2*a^5*b^9*c^3*d^9*f^2 + 2224*A^2*a^4*b^10*c^2*d^10*f^2 - 2184*A^2*a^7*b^
7*c^3*d^9*f^2 - 2016*A^2*a^7*b^7*c^5*d^7*f^2 - 1984*A^2*a^7*b^7*c^7*d^5*f^2
+ 1626*A^2*a^8*b^6*c^2*d^10*f^2 - 1624*A^2*a^5*b^9*c^9*d^3*f^2 + 1603*A^2*
a^4*b^10*c^8*d^4*f^2 + 1296*A^2*a^9*b^5*c^5*d^7*f^2 - 1144*A^2*a^3*b^11*c^5
*d^7*f^2 - 992*A^2*a^3*b^11*c^3*d^9*f^2 + 968*A^2*a^2*b^12*c^4*d^8*f^2 - 88
8*A^2*a^3*b^11*c^7*d^5*f^2 + 849*A^2*a^8*b^6*c^4*d^8*f^2 + 808*A^2*a^2*b^12
*c^2*d^10*f^2 - 616*A^2*a^7*b^7*c^9*d^3*f^2 + 554*A^2*a^6*b^8*c^10*d^2*f^2

- 504*A^2*a^10*b^4*c^6*d^6*f^2 + 504*A^2*a^9*b^5*c^7*d^5*f^2 + 460*A^2*a^2*
b^12*c^6*d^6*f^2 + 350*A^2*a^10*b^4*c^2*d^10*f^2 + 350*A^2*a^4*b^10*c^10*d^
2*f^2 - 321*A^2*a^10*b^4*c^4*d^8*f^2 + 216*A^2*a^11*b^3*c^5*d^7*f^2 - 216*A
^2*a^11*b^3*c^3*d^9*f^2 + 182*A^2*a^12*b^2*c^2*d^10*f^2 - 152*A^2*a^3*b^11*
c^9*d^3*f^2 - 124*A^2*a^8*b^6*c^6*d^6*f^2 - 114*A^2*a^2*b^12*c^10*d^2*f^2 +
104*A^2*a^9*b^5*c^3*d^9*f^2 + 77*A^2*a^2*b^12*c^8*d^4*f^2 + 74*A^2*a^8*b^6
*c^8*d^4*f^2 - 70*A^2*a^10*b^4*c^8*d^4*f^2 + 56*A^2*a^11*b^3*c^7*d^5*f^2 +
56*A^2*a^9*b^5*c^9*d^3*f^2 + 41*A^2*a^12*b^2*c^4*d^8*f^2 - 28*A^2*a^12*b^2*
c^6*d^6*f^2 - 28*A^2*a^8*b^6*c^10*d^2*f^2 - 16*B*C*b^14*c^11*d*f^2 - 16*B*C
*a^14*c*d^11*f^2 - 48*A*B*b^14*c*d^11*f^2 + 16*A*B*b^14*c^11*d*f^2 + 12*B*C
*a^13*b*d^12*f^2 + 24*B*C*a*b^13*c^12*f^2 + 16*A*B*a^14*c*d^11*f^2 - 24*A*B
*a^13*b*d^12*f^2 - 24*A*B*a*b^13*d^12*f^2 - 24*A*B*a*b^13*c^12*f^2 + 216*C^
2*a^9*b^5*c*d^11*f^2 - 216*C^2*a^5*b^9*c^11*d*f^2 + 56*C^2*a^3*b^11*c^11*d*
f^2 + 56*C^2*a*b^13*c^9*d^3*f^2 + 56*C^2*a*b^13*c^5*d^7*f^2 - 40*C^2*a^11*b
^3*c*d^11*f^2 + 40*C^2*a*b^13*c^7*d^5*f^2 + 32*C^2*a^13*b*c^5*d^7*f^2 - 24*
C^2*a^7*b^7*c*d^11*f^2 - 16*C^2*a^13*b*c^3*d^9*f^2 + 16*C^2*a*b^13*c^3*d^9*
f^2 + 8*C^2*a^7*b^7*c^11*d*f^2 - 8*C^2*a^5*b^9*c*d^11*f^2 + 264*B^2*a^7*b^7
*c*d^11*f^2 + 224*B^2*a^5*b^9*c*d^11*f^2 + 168*B^2*a^5*b^9*c^11*d*f^2 - 112
*B^2*a*b^13*c^9*d^3*f^2 - 104*B^2*a^3*b^11*c^11*d*f^2 - 104*B^2*a*b^13*c^7*
d^5*f^2 + 96*B^2*a^3*b^11*c*d^11*f^2 + 88*B^2*a^11*b^3*c*d^11*f^2 - 72*B^2*
a^9*b^5*c*d^11*f^2 - 64*B^2*a*b^13*c^5*d^7*f^2 + 32*B^2*a^13*b*c^3*d^9*f^2
- 24*B^2*a^13*b*c^5*d^7*f^2 - 24*B^2*a^7*b^7*c^11*d*f^2 + 16*B^2*a*b^13*c^3
*d^9*f^2 - 888*A^2*a^7*b^7*c*d^11*f^2 - 800*A^2*a^5*b^9*c*d^11*f^2 - 336*A^
2*a^3*b^11*c*d^11*f^2 - 264*A^2*a^9*b^5*c*d^11*f^2 - 216*A^2*a^5*b^9*c^11*d
*f^2 - 184*A^2*a^11*b^3*c*d^11*f^2 - 128*A^2*a*b^13*c^3*d^9*f^2 - 112*A^2*a
*b^13*c^5*d^7*f^2 - 64*A^2*a^13*b*c^3*d^9*f^2 + 56*A^2*a^3*b^11*c^11*d*f^2
- 56*A^2*a*b^13*c^7*d^5*f^2 + 32*A^2*a*b^13*c^9*d^3*f^2 + 8*A^2*a^13*b*c^5*
d^7*f^2 + 8*A^2*a^7*b^7*c^11*d*f^2 + 24*C^2*a*b^13*c^11*d*f^2 - 16*C^2*a^13
*b*c*d^11*f^2 - 40*B^2*a*b^13*c^11*d*f^2 + 24*B^2*a^13*b*c*d^11*f^2 + 16*B^
2*a*b^13*c*d^11*f^2 - 48*A^2*a*b^13*c*d^11*f^2 - 40*A^2*a^13*b*c*d^11*f^2 +
24*A^2*a*b^13*c^11*d*f^2 - 6*A*C*b^14*c^12*f^2 + 2*A*C*a^14*d^12*f^2 + 31*
C^2*b^14*c^8*d^4*f^2 + 20*C^2*b^14*c^6*d^6*f^2 + 4*C^2*b^14*c^4*d^8*f^2 + 2
*C^2*b^14*c^10*d^2*f^2 + 80*B^2*b^14*c^6*d^6*f^2 + 64*B^2*b^14*c^4*d^8*f^2
+ 31*B^2*b^14*c^8*d^4*f^2 + 16*B^2*b^14*c^2*d^10*f^2 + 14*C^2*a^14*c^2*d^10
*f^2 + 14*B^2*b^14*c^10*d^2*f^2 - C^2*a^14*c^4*d^8*f^2 + 120*A^2*b^14*c^2*d
^10*f^2 + 112*A^2*b^14*c^4*d^8*f^2 + 33*C^2*a^12*b^2*d^12*f^2 - 27*C^2*a^10
*b^4*d^12*f^2 - 17*A^2*b^14*c^8*d^4*f^2 - 10*B^2*a^14*c^2*d^10*f^2 - 10*A^2
*b^14*c^10*d^2*f^2 + 8*A^2*b^14*c^6*d^6*f^2 + 3*C^2*a^8*b^6*d^12*f^2 + 3*B^
2*a^14*c^4*d^8*f^2 + 117*B^2*a^10*b^4*d^12*f^2 + 111*B^2*a^8*b^6*d^12*f^2 +
72*B^2*a^6*b^8*d^12*f^2 + 33*C^2*a^4*b^10*c^12*f^2 - 27*C^2*a^2*b^12*c^12*
f^2 + 24*B^2*a^4*b^10*d^12*f^2 + 14*A^2*a^14*c^2*d^10*f^2 + 4*B^2*a^2*b^12*
d^12*f^2 - 3*B^2*a^12*b^2*d^12*f^2 - C^2*a^6*b^8*c^12*f^2 - A^2*a^14*c^4*d^
8*f^2 + 720*A^2*a^6*b^8*d^12*f^2 + 552*A^2*a^4*b^10*d^12*f^2 + 471*A^2*a^8*
b^6*d^12*f^2 + 216*A^2*a^2*b^12*d^12*f^2 + 93*A^2*a^10*b^4*d^12*f^2 + 33*B^
2*a^2*b^12*c^12*f^2 + 33*A^2*a^12*b^2*d^12*f^2 - 27*B^2*a^4*b^10*c^12*f^2 +

3*B^2*a^6*b^8*c^12*f^2 + 33*A^2*a^4*b^10*c^12*f^2 - 27*A^2*a^2*b^12*c^12*f
^2 - A^2*a^6*b^8*c^12*f^2 + 3*C^2*b^14*c^12*f^2 - C^2*a^14*d^12*f^2 + 36*A^
2*b^14*d^12*f^2 + 3*B^2*a^14*d^12*f^2 - B^2*b^14*c^12*f^2 + 3*A^2*b^14*c^12
*f^2 - A^2*a^14*d^12*f^2 - 44*A*B*C*a^10*b*c*d^9*f + 3816*A*B*C*a^4*b^7*c^5
*d^5*f + 2920*A*B*C*a^5*b^6*c^2*d^8*f - 2736*A*B*C*a^6*b^5*c^3*d^7*f - 2672
*A*B*C*a^3*b^8*c^4*d^6*f + 1996*A*B*C*a^7*b^4*c^4*d^6*f - 1412*A*B*C*a^5*b^
6*c^6*d^4*f + 1120*A*B*C*a^2*b^9*c^3*d^7*f + 1080*A*B*C*a^7*b^4*c^2*d^8*f +
 1040*A*B*C*a^2*b^9*c^5*d^5*f + 684*A*B*C*a^5*b^6*c^4*d^6*f + 592*A*B*C*a^4
*b^7*c^3*d^7*f - 560*A*B*C*a^2*b^9*c^7*d^3*f - 448*A*B*C*a^3*b^8*c^2*d^8*f
- 400*A*B*C*a^8*b^3*c^5*d^5*f - 398*A*B*C*a^9*b^2*c^2*d^8*f - 312*A*B*C*a^3
*b^8*c^6*d^4*f + 166*A*B*C*a^3*b^8*c^8*d^2*f + 136*A*B*C*a^6*b^5*c^5*d^5*f
+ 128*A*B*C*a^6*b^5*c^7*d^3*f - 100*A*B*C*a^7*b^4*c^6*d^4*f - 64*A*B*C*a^9*
b^2*c^4*d^6*f + 64*A*B*C*a^4*b^7*c^7*d^3*f - 32*A*B*C*a^8*b^3*c^3*d^7*f - 1
6*A*B*C*a^5*b^6*c^8*d^2*f - 1312*A*B*C*a^4*b^7*c*d^9*f + 996*A*B*C*a^8*b^3*
c*d^9*f + 728*A*B*C*a*b^10*c^6*d^4*f - 624*A*B*C*a^6*b^5*c*d^9*f - 584*A*B*
C*a*b^10*c^2*d^8*f - 512*A*B*C*a*b^10*c^4*d^6*f - 320*A*B*C*a^2*b^9*c*d^9*f
 - 98*A*B*C*a*b^10*c^8*d^2*f + 36*A*B*C*a^2*b^9*c^9*d*f + 32*A*B*C*a^10*b*c
^3*d^7*f - 16*A*B*C*a^4*b^7*c^9*d*f + 46*B*C^2*a^10*b*c*d^9*f - 16*B^2*C*a*
b^10*c*d^9*f - 2*B^2*C*a*b^10*c^9*d*f + 312*A^2*C*a*b^10*c*d^9*f - 48*A*C^2
*a*b^10*c*d^9*f - 6*A^2*C*a*b^10*c^9*d*f + 6*A*C^2*a*b^10*c^9*d*f + 208*A*B
^2*a*b^10*c*d^9*f - 2*A^2*B*a^10*b*c*d^9*f + 2*A*B^2*a*b^10*c^9*d*f - 224*A
*B*C*b^11*c^5*d^5*f + 80*A*B*C*b^11*c^7*d^3*f - 32*A*B*C*b^11*c^3*d^7*f + 2
*A*B*C*a^11*c^2*d^8*f - 480*A*B*C*a^7*b^4*d^10*f + 78*A*B*C*a^9*b^2*d^10*f
- 64*A*B*C*a^5*b^6*d^10*f + 2*A*B*C*a^3*b^8*c^10*f - 1692*B*C^2*a^4*b^7*c^5
*d^5*f - 1500*B^2*C*a^5*b^6*c^5*d^5*f - 1464*B^2*C*a^5*b^6*c^3*d^7*f + 1426
*B*C^2*a^5*b^6*c^6*d^4*f - 1158*B^2*C*a^4*b^7*c^6*d^4*f + 1152*B*C^2*a^6*b^
5*c^3*d^7*f + 1026*B^2*C*a^6*b^5*c^4*d^6*f - 974*B*C^2*a^7*b^4*c^4*d^6*f +
960*B^2*C*a^3*b^8*c^5*d^5*f - 884*B*C^2*a^5*b^6*c^2*d^8*f - 764*B^2*C*a^7*b
^4*c^5*d^5*f + 752*B^2*C*a^4*b^7*c^2*d^8*f - 752*B*C^2*a^4*b^7*c^3*d^7*f +
738*B^2*C*a^4*b^7*c^4*d^6*f - 688*B^2*C*a^2*b^9*c^6*d^4*f - 675*B^2*C*a^8*b
^3*c^2*d^8*f + 560*B*C^2*a^8*b^3*c^5*d^5*f + 496*B*C^2*a^3*b^8*c^4*d^6*f +
496*B*C^2*a^2*b^9*c^7*d^3*f - 468*B*C^2*a^7*b^4*c^2*d^8*f + 456*B^2*C*a^3*b
^8*c^7*d^3*f - 452*B^2*C*a^8*b^3*c^4*d^6*f - 416*B*C^2*a^2*b^9*c^3*d^7*f +
378*B*C^2*a^5*b^6*c^4*d^6*f + 376*B*C^2*a^8*b^3*c^3*d^7*f - 360*B^2*C*a^6*b
^5*c^2*d^8*f + 355*B*C^2*a^9*b^2*c^2*d^8*f + 346*B^2*C*a^6*b^5*c^6*d^4*f -
320*B^2*C*a^2*b^9*c^4*d^6*f + 268*B^2*C*a^2*b^9*c^2*d^8*f + 216*B^2*C*a^7*b
^4*c^3*d^7*f - 203*B*C^2*a^3*b^8*c^8*d^2*f - 184*B*C^2*a^6*b^5*c^7*d^3*f +
170*B*C^2*a^7*b^4*c^6*d^4*f + 160*B^2*C*a^5*b^6*c^7*d^3*f - 160*B*C^2*a^2*b
^9*c^5*d^5*f - 140*B^2*C*a^4*b^7*c^8*d^2*f - 136*B*C^2*a^3*b^8*c^2*d^8*f +
112*B^2*C*a^9*b^2*c^3*d^7*f + 91*B^2*C*a^2*b^9*c^8*d^2*f + 88*B*C^2*a^4*b^7
*c^7*d^3*f + 72*B^2*C*a^8*b^3*c^6*d^4*f - 64*B^2*C*a^3*b^8*c^3*d^7*f - 60*B
*C^2*a^3*b^8*c^6*d^4*f + 56*B*C^2*a^9*b^2*c^4*d^6*f + 52*B*C^2*a^6*b^5*c^5*
d^5*f + 48*B^2*C*a^9*b^2*c^5*d^5*f - 48*B^2*C*a^7*b^4*c^7*d^3*f + 44*B*C^2*
a^5*b^6*c^8*d^2*f - 36*B*C^2*a^9*b^2*c^6*d^4*f + 12*B^2*C*a^6*b^5*c^8*d^2*f
 - 2958*A^2*C*a^4*b^7*c^4*d^6*f - 1932*A^2*C*a^4*b^7*c^2*d^8*f + 1848*A^2*C

*a^5*b^6*c^3*d^7*f + 1728*A^2*C*a^3*b^8*c^3*d^7*f + 1524*A^2*C*a^5*b^6*c^5*
d^5*f + 1374*A*C^2*a^4*b^7*c^4*d^6*f - 1272*A*C^2*a^5*b^6*c^3*d^7*f - 1236*
A*C^2*a^5*b^6*c^5*d^5*f + 1116*A*C^2*a^4*b^7*c^2*d^8*f - 1110*A^2*C*a^6*b^5
*c^4*d^6*f + 1038*A*C^2*a^6*b^5*c^4*d^6*f - 768*A^2*C*a^2*b^9*c^2*d^8*f - 6
96*A^2*C*a^7*b^4*c^3*d^7*f - 666*A*C^2*a^4*b^7*c^6*d^4*f + 564*A^2*C*a^6*b^
5*c^2*d^8*f - 564*A*C^2*a^7*b^4*c^5*d^5*f - 555*A*C^2*a^8*b^3*c^2*d^8*f + 5
19*A^2*C*a^8*b^3*c^2*d^8*f - 480*A*C^2*a^3*b^8*c^3*d^7*f + 456*A*C^2*a^3*b^
8*c^5*d^5*f - 420*A*C^2*a^2*b^9*c^6*d^4*f + 408*A*C^2*a^7*b^4*c^3*d^7*f + 4
08*A*C^2*a^2*b^9*c^2*d^8*f + 348*A^2*C*a^2*b^9*c^6*d^4*f - 348*A*C^2*a^6*b^
5*c^2*d^8*f + 342*A*C^2*a^6*b^5*c^6*d^4*f - 336*A*C^2*a^8*b^3*c^4*d^6*f + 3
24*A^2*C*a^7*b^4*c^5*d^5*f - 312*A^2*C*a^2*b^9*c^4*d^6*f + 264*A^2*C*a^8*b^
3*c^4*d^6*f + 240*A*C^2*a^5*b^6*c^7*d^3*f + 195*A*C^2*a^2*b^9*c^8*d^2*f - 1
74*A^2*C*a^6*b^5*c^6*d^4*f + 144*A*C^2*a^9*b^2*c^3*d^7*f - 123*A^2*C*a^2*b^
9*c^8*d^2*f + 120*A*C^2*a^3*b^8*c^7*d^3*f + 108*A*C^2*a^8*b^3*c^6*d^4*f - 1
02*A^2*C*a^4*b^7*c^6*d^4*f - 96*A^2*C*a^4*b^7*c^8*d^2*f + 72*A^2*C*a^3*b^8*
c^7*d^3*f + 72*A*C^2*a^9*b^2*c^5*d^5*f - 48*A^2*C*a^9*b^2*c^3*d^7*f + 48*A^
2*C*a^5*b^6*c^7*d^3*f - 48*A*C^2*a^2*b^9*c^4*d^6*f - 24*A^2*C*a^3*b^8*c^5*d
^5*f - 12*A*C^2*a^4*b^7*c^8*d^2*f + 2736*A^2*B*a^6*b^5*c^3*d^7*f + 2464*A^2
*B*a^3*b^8*c^4*d^6*f - 2298*A*B^2*a^4*b^7*c^4*d^6*f - 2252*A^2*B*a^5*b^6*c^
2*d^8*f - 1692*A^2*B*a^4*b^7*c^5*d^5*f - 1592*A*B^2*a^4*b^7*c^2*d^8*f - 133
8*A*B^2*a^6*b^5*c^4*d^6*f + 1320*A*B^2*a^5*b^6*c^3*d^7*f + 1212*A*B^2*a^5*b
^6*c^5*d^5*f - 1056*A*B^2*a^3*b^8*c^5*d^5*f + 1024*A^2*B*a^4*b^7*c^3*d^7*f
- 1022*A^2*B*a^7*b^4*c^4*d^6*f - 880*A^2*B*a^2*b^9*c^5*d^5*f - 846*A^2*B*a^
5*b^6*c^4*d^6*f - 840*A*B^2*a^7*b^4*c^3*d^7*f + 760*A*B^2*a^2*b^9*c^6*d^4*f
 - 704*A^2*B*a^2*b^9*c^3*d^7*f + 688*A*B^2*a^3*b^8*c^3*d^7*f + 660*A^2*B*a^
3*b^8*c^6*d^4*f - 612*A^2*B*a^7*b^4*c^2*d^8*f + 462*A*B^2*a^4*b^7*c^6*d^4*f
 + 459*A*B^2*a^8*b^3*c^2*d^8*f - 412*A*B^2*a^2*b^9*c^2*d^8*f - 408*A*B^2*a^
3*b^8*c^7*d^3*f + 388*A^2*B*a^6*b^5*c^5*d^5*f + 296*A^2*B*a^3*b^8*c^2*d^8*f
 + 288*A*B^2*a^6*b^5*c^2*d^8*f + 284*A*B^2*a^7*b^4*c^5*d^5*f + 236*A*B^2*a^
8*b^3*c^4*d^6*f - 226*A*B^2*a^6*b^5*c^6*d^4*f + 212*A*B^2*a^2*b^9*c^4*d^6*f
 + 202*A^2*B*a^5*b^6*c^6*d^4*f - 152*A^2*B*a^4*b^7*c^7*d^3*f + 88*A^2*B*a^8
*b^3*c^3*d^7*f + 79*A^2*B*a^9*b^2*c^2*d^8*f - 70*A^2*B*a^7*b^4*c^6*d^4*f +
68*A*B^2*a^4*b^7*c^8*d^2*f + 64*A^2*B*a^2*b^9*c^7*d^3*f - 64*A*B^2*a^9*b^2*
c^3*d^7*f + 56*A^2*B*a^8*b^3*c^5*d^5*f + 56*A^2*B*a^6*b^5*c^7*d^3*f + 37*A^
2*B*a^3*b^8*c^8*d^2*f - 28*A^2*B*a^9*b^2*c^4*d^6*f - 28*A^2*B*a^5*b^6*c^8*d
^2*f + 17*A*B^2*a^2*b^9*c^8*d^2*f - 16*A*B^2*a^5*b^6*c^7*d^3*f + 48*A*B*C*b
^11*c*d^9*f + 4*A*B*C*b^11*c^9*d*f + 24*A*B*C*a*b^10*d^10*f - 6*A*B*C*a*b^1
0*c^10*f + 432*B^2*C*a^7*b^4*c*d^9*f - 376*B*C^2*a*b^10*c^6*d^4*f - 354*B*C
^2*a^8*b^3*c*d^9*f + 352*B^2*C*a*b^10*c^5*d^5*f + 320*B^2*C*a^5*b^6*c*d^9*f
 + 256*B^2*C*a*b^10*c^3*d^7*f - 232*B^2*C*a*b^10*c^7*d^3*f - 210*B^2*C*a^9*
b^2*c*d^9*f - 152*B*C^2*a*b^10*c^4*d^6*f + 85*B*C^2*a*b^10*c^8*d^2*f + 72*B
^2*C*a^3*b^8*c*d^9*f - 48*B*C^2*a^6*b^5*c*d^9*f - 40*B*C^2*a^10*b*c^3*d^7*f
 + 40*B*C^2*a*b^10*c^2*d^8*f + 37*B^2*C*a^10*b*c^2*d^8*f + 22*B^2*C*a^3*b^8
*c^9*d*f - 18*B*C^2*a^2*b^9*c^9*d*f + 16*B*C^2*a^2*b^9*c*d^9*f - 12*B^2*C*a
^10*b*c^4*d^6*f + 8*B*C^2*a^4*b^7*c^9*d*f + 8*B*C^2*a^4*b^7*c*d^9*f - 984*A

^2*C*a^7*b^4*c*d^9*f + 672*A^2*C*a^3*b^8*c*d^9*f + 552*A*C^2*a^7*b^4*c*d^9*
f - 504*A^2*C*a*b^10*c^5*d^5*f - 408*A^2*C*a^5*b^6*c*d^9*f + 408*A*C^2*a^5*
b^6*c*d^9*f + 336*A*C^2*a*b^10*c^5*d^5*f - 216*A*C^2*a*b^10*c^7*d^3*f + 192
*A*C^2*a*b^10*c^3*d^7*f - 162*A*C^2*a^9*b^2*c*d^9*f + 120*A^2*C*a*b^10*c^7*
d^3*f + 96*A^2*C*a*b^10*c^3*d^7*f + 90*A^2*C*a^9*b^2*c*d^9*f + 66*A^2*C*a^3
*b^8*c^9*d*f - 66*A*C^2*a^3*b^8*c^9*d*f + 57*A*C^2*a^10*b*c^2*d^8*f - 48*A*
C^2*a^3*b^8*c*d^9*f - 9*A^2*C*a^10*b*c^2*d^8*f + 1736*A^2*B*a^4*b^7*c*d^9*f
 + 1248*A^2*B*a^6*b^5*c*d^9*f - 1008*A*B^2*a^7*b^4*c*d^9*f + 772*A^2*B*a*b^
10*c^4*d^6*f - 688*A*B^2*a*b^10*c^5*d^5*f - 608*A*B^2*a^5*b^6*c*d^9*f + 436
*A^2*B*a*b^10*c^2*d^8*f - 426*A^2*B*a^8*b^3*c*d^9*f + 312*A*B^2*a^3*b^8*c*d
^9*f + 304*A^2*B*a^2*b^9*c*d^9*f - 244*A^2*B*a*b^10*c^6*d^4*f - 160*A*B^2*a
*b^10*c^3*d^7*f + 114*A*B^2*a^9*b^2*c*d^9*f + 88*A*B^2*a*b^10*c^7*d^3*f - 2
2*A*B^2*a^3*b^8*c^9*d*f - 18*A^2*B*a^2*b^9*c^9*d*f + 13*A^2*B*a*b^10*c^8*d^
2*f - 13*A*B^2*a^10*b*c^2*d^8*f + 8*A^2*B*a^10*b*c^3*d^7*f + 8*A^2*B*a^4*b^
7*c^9*d*f + 112*B^2*C*b^11*c^6*d^4*f - 64*B*C^2*b^11*c^7*d^3*f + 16*B^2*C*b
^11*c^4*d^6*f - 16*B^2*C*b^11*c^2*d^8*f + 16*B*C^2*b^11*c^5*d^5*f + 16*B*C^
2*b^11*c^3*d^7*f - B^2*C*b^11*c^8*d^2*f + 96*A^2*C*b^11*c^4*d^6*f - 84*A^2*
C*b^11*c^6*d^4*f + 72*A*C^2*b^11*c^6*d^4*f - 24*A*C^2*b^11*c^4*d^6*f - 24*A
*C^2*b^11*c^2*d^8*f - 21*A*C^2*b^11*c^8*d^2*f + 12*A^2*C*b^11*c^2*d^8*f + 9
*A^2*C*b^11*c^8*d^2*f - B*C^2*a^11*c^2*d^8*f + 176*A*B^2*b^11*c^4*d^6*f + 1
36*A^2*B*b^11*c^5*d^5*f - 128*A^2*B*b^11*c^3*d^7*f + 112*A*B^2*b^11*c^2*d^8
*f + 111*B^2*C*a^8*b^3*d^10*f - 64*A*B^2*b^11*c^6*d^4*f - 39*B*C^2*a^9*b^2*
d^10*f + 24*B*C^2*a^7*b^4*d^10*f - 16*A^2*B*b^11*c^7*d^3*f - 4*B^2*C*a^2*b^
9*d^10*f - 4*B*C^2*a^5*b^6*d^10*f + 432*A^2*C*a^6*b^5*d^10*f + 192*A^2*C*a^
4*b^7*d^10*f - 111*A^2*C*a^8*b^3*d^10*f + 111*A*C^2*a^8*b^3*d^10*f - 72*A*C
^2*a^6*b^5*d^10*f + 12*A*C^2*a^4*b^7*d^10*f - 3*B^2*C*a^2*b^9*c^10*f - A^2*
B*a^11*c^2*d^8*f - B*C^2*a^3*b^8*c^10*f + 456*A^2*B*a^7*b^4*d^10*f - 288*A^
2*B*a^3*b^8*d^10*f + 252*A*B^2*a^6*b^5*d^10*f + 192*A*B^2*a^4*b^7*d^10*f -
183*A*B^2*a^8*b^3*d^10*f - 148*A^2*B*a^5*b^6*d^10*f + 76*A*B^2*a^2*b^9*d^10
*f - 9*A^2*C*a^2*b^9*c^10*f + 9*A*C^2*a^2*b^9*c^10*f - 3*A^2*B*a^9*b^2*d^10
*f + 3*A*B^2*a^2*b^9*c^10*f - A^2*B*a^3*b^8*c^10*f - 2*C^3*a*b^10*c^9*d*f -
 2*B^3*a^10*b*c*d^9*f - 264*A^3*a*b^10*c*d^9*f + 2*A^3*a*b^10*c^9*d*f - 2*B
*C^2*b^11*c^9*d*f - 2*B^2*C*a^11*c*d^9*f - 120*A^2*B*b^11*c*d^9*f - 9*B^2*C
*a^10*b*d^10*f - 6*A^2*C*a^11*c*d^9*f + 6*A*C^2*a^11*c*d^9*f - 2*A^2*B*b^11
*c^9*d*f + 9*A^2*C*a^10*b*d^10*f - 9*A*C^2*a^10*b*d^10*f + 3*B*C^2*a*b^10*c
^10*f + 2*A*B^2*a^11*c*d^9*f - 132*A^2*B*a*b^10*d^10*f - 3*A*B^2*a^10*b*d^1
0*f + 3*A^2*B*a*b^10*c^10*f + 520*C^3*a^5*b^6*c^3*d^7*f + 460*C^3*a^5*b^6*c
^5*d^5*f - 418*C^3*a^6*b^5*c^4*d^6*f + 406*C^3*a^4*b^7*c^6*d^4*f + 268*C^3*
a^7*b^4*c^5*d^5*f - 266*C^3*a^6*b^5*c^6*d^4*f + 233*C^3*a^8*b^3*c^2*d^8*f -
 176*C^3*a^5*b^6*c^7*d^3*f + 164*C^3*a^2*b^9*c^6*d^4*f + 140*C^3*a^6*b^5*c^
2*d^8*f + 136*C^3*a^2*b^9*c^4*d^6*f - 128*C^3*a^9*b^2*c^3*d^7*f + 128*C^3*a
^3*b^8*c^3*d^7*f - 108*C^3*a^8*b^3*c^6*d^4*f - 104*C^3*a^3*b^8*c^7*d^3*f -
104*C^3*a^3*b^8*c^5*d^5*f + 100*C^3*a^8*b^3*c^4*d^6*f - 89*C^3*a^2*b^9*c^8*
d^2*f - 72*C^3*a^9*b^2*c^5*d^5*f - 40*C^3*a^7*b^4*c^3*d^7*f + 40*C^3*a^4*b^
7*c^8*d^2*f - 28*C^3*a^4*b^7*c^2*d^8*f - 16*C^3*a^2*b^9*c^2*d^8*f - 2*C^3*a

^4*b^7*c^4*d^6*f + 828*B^3*a^4*b^7*c^5*d^5*f + 408*B^3*a^5*b^6*c^2*d^8*f +
390*B^3*a^7*b^4*c^4*d^6*f - 372*B^3*a^3*b^8*c^4*d^6*f - 336*B^3*a^6*b^5*c^3
*d^7*f - 314*B^3*a^5*b^6*c^6*d^4*f + 288*B^3*a^4*b^7*c^3*d^7*f + 216*B^3*a^
7*b^4*c^2*d^8*f - 176*B^3*a^2*b^9*c^7*d^3*f + 128*B^3*a^2*b^9*c^3*d^7*f + 1
08*B^3*a^6*b^5*c^5*d^5*f + 88*B^3*a^4*b^7*c^7*d^3*f + 72*B^3*a^2*b^9*c^5*d^
5*f - 68*B^3*a^3*b^8*c^2*d^8*f - 65*B^3*a^9*b^2*c^2*d^8*f - 56*B^3*a^8*b^3*
c^5*d^5*f + 40*B^3*a^6*b^5*c^7*d^3*f + 37*B^3*a^3*b^8*c^8*d^2*f + 30*B^3*a^
5*b^6*c^4*d^6*f - 28*B^3*a^5*b^6*c^8*d^2*f + 24*B^3*a^8*b^3*c^3*d^7*f - 4*B
^3*a^9*b^2*c^4*d^6*f - 2*B^3*a^7*b^4*c^6*d^4*f + 1586*A^3*a^4*b^7*c^4*d^6*f
 - 1376*A^3*a^3*b^8*c^3*d^7*f - 1096*A^3*a^5*b^6*c^3*d^7*f + 844*A^3*a^4*b^
7*c^2*d^8*f - 748*A^3*a^5*b^6*c^5*d^5*f + 490*A^3*a^6*b^5*c^4*d^6*f + 376*A
^3*a^2*b^9*c^2*d^8*f + 362*A^3*a^4*b^7*c^6*d^4*f - 356*A^3*a^6*b^5*c^2*d^8*
f + 328*A^3*a^7*b^4*c^3*d^7*f - 328*A^3*a^3*b^8*c^5*d^5*f + 224*A^3*a^2*b^9
*c^4*d^6*f - 197*A^3*a^8*b^3*c^2*d^8*f - 112*A^3*a^5*b^6*c^7*d^3*f + 98*A^3
*a^6*b^5*c^6*d^4*f - 92*A^3*a^2*b^9*c^6*d^4*f - 88*A^3*a^3*b^8*c^7*d^3*f +
68*A^3*a^4*b^7*c^8*d^2*f + 32*A^3*a^9*b^2*c^3*d^7*f - 28*A^3*a^8*b^3*c^4*d^
6*f - 28*A^3*a^7*b^4*c^5*d^5*f + 17*A^3*a^2*b^9*c^8*d^2*f + 104*C^3*a*b^10*
c^7*d^3*f + 54*C^3*a^9*b^2*c*d^9*f - 40*C^3*a^7*b^4*c*d^9*f - 35*C^3*a^10*b
*c^2*d^8*f + 22*C^3*a^3*b^8*c^9*d*f + 16*C^3*a*b^10*c^5*d^5*f - 16*C^3*a*b^
10*c^3*d^7*f + 8*C^3*a^5*b^6*c*d^9*f - 2*A*B*C*a^11*d^10*f + 198*B^3*a^8*b^
3*c*d^9*f + 192*B^3*a*b^10*c^6*d^4*f - 128*B^3*a^4*b^7*c*d^9*f - 80*B^3*a*b
^10*c^2*d^8*f - 56*B^3*a^2*b^9*c*d^9*f - 24*B^3*a^6*b^5*c*d^9*f - 18*B^3*a^
2*b^9*c^9*d*f - 16*B^3*a*b^10*c^4*d^6*f + 13*B^3*a*b^10*c^8*d^2*f + 8*B^3*a
^10*b*c^3*d^7*f + 8*B^3*a^4*b^7*c^9*d*f - 624*A^3*a^3*b^8*c*d^9*f + 472*A^3
*a^7*b^4*c*d^9*f - 272*A^3*a*b^10*c^3*d^7*f + 152*A^3*a*b^10*c^5*d^5*f - 22
*A^3*a^3*b^8*c^9*d*f + 18*A^3*a^9*b^2*c*d^9*f - 13*A^3*a^10*b*c^2*d^8*f - 8
*A^3*a^5*b^6*c*d^9*f - 8*A^3*a*b^10*c^7*d^3*f + A*B^2*b^11*c^8*d^2*f + 11*C
^3*b^11*c^8*d^2*f - 8*C^3*b^11*c^6*d^4*f - 4*C^3*b^11*c^4*d^6*f - 64*B^3*b^
11*c^5*d^5*f - 32*B^3*b^11*c^3*d^7*f - 68*A^3*b^11*c^4*d^6*f + 20*A^3*b^11*
c^6*d^4*f + 12*A^3*b^11*c^2*d^8*f - C^3*a^8*b^3*d^10*f - B^3*a^11*c^2*d^8*f
 - 60*B^3*a^7*b^4*d^10*f - 32*B^3*a^5*b^6*d^10*f + 21*B^3*a^9*b^2*d^10*f -
12*B^3*a^3*b^8*d^10*f - 3*C^3*a^2*b^9*c^10*f - 360*A^3*a^6*b^5*d^10*f - 204
*A^3*a^4*b^7*d^10*f - B^3*a^3*b^8*c^10*f + 3*A^3*a^2*b^9*c^10*f - 2*C^3*a^1
1*c*d^9*f - 2*B^3*b^11*c^9*d*f + 3*C^3*a^10*b*d^10*f + 2*A^3*a^11*c*d^9*f +
 3*B^3*a*b^10*c^10*f - 3*A^3*a^10*b*d^10*f - 36*A^2*C*b^11*d^10*f + 3*A^2*C
*b^11*c^10*f - 3*A*C^2*b^11*c^10*f - A*B^2*b^11*c^10*f + 36*A^3*b^11*d^10*f
 - A^3*b^11*c^10*f + A^3*b^11*c^8*d^2*f + A^3*a^8*b^3*d^10*f + B^2*C*b^11*c
^10*f + B*C^2*a^11*d^10*f + A^2*B*a^11*d^10*f + C^3*b^11*c^10*f + B^3*a^11*
d^10*f - 6*A*B^2*C*a^7*b*c*d^7 + 4*A*B^2*C*a*b^7*c*d^7 + 168*A^2*B*C*a^2*b^
6*c^3*d^5 + 144*A*B*C^2*a^3*b^5*c^4*d^4 - 129*A^2*B*C*a^3*b^5*c^4*d^4 - 96*
A*B*C^2*a^2*b^6*c^3*d^5 + 84*A*B*C^2*a^3*b^5*c^2*d^6 + 72*A^2*B*C*a^4*b^4*c
^3*d^5 - 72*A^2*B*C*a^3*b^5*c^2*d^6 + 64*A*B^2*C*a^4*b^4*c^4*d^4 - 60*A*B*C
^2*a^4*b^4*c^3*d^5 + 57*A^2*B*C*a^5*b^3*c^2*d^6 - 56*A*B^2*C*a^5*b^3*c^3*d^
5 - 39*A*B^2*C*a^2*b^6*c^4*d^4 - 38*A*B^2*C*a^3*b^5*c^5*d^3 + 36*A*B^2*C*a^
3*b^5*c^3*d^5 + 36*A*B*C^2*a^5*b^3*c^4*d^4 - 30*A*B*C^2*a^5*b^3*c^2*d^6 + 2

7*A*B^2*C*a^6*b^2*c^2*d^6 - 24*A*B^2*C*a^2*b^6*c^2*d^6 + 24*A*B*C^2*a^6*b^2
*c^3*d^5 - 24*A*B*C^2*a^4*b^4*c^5*d^3 - 18*A^2*B*C*a^5*b^3*c^4*d^4 + 18*A^2
*B*C*a^2*b^6*c^5*d^3 - 15*A*B^2*C*a^4*b^4*c^2*d^6 - 12*A^2*B*C*a^6*b^2*c^3*
d^5 + 12*A^2*B*C*a^4*b^4*c^5*d^3 + 9*A*B^2*C*a^2*b^6*c^6*d^2 + 6*A*B*C^2*a^
3*b^5*c^6*d^2 - 3*A^2*B*C*a^3*b^5*c^6*d^2 + 60*A^2*B*C*a^2*b^6*c*d^7 - 51*A
^2*B*C*a*b^7*c^4*d^4 + 48*A*B*C^2*a^6*b^2*c*d^7 - 42*A^2*B*C*a^6*b^2*c*d^7
 - 42*A^2*B*C*a*b^7*c^2*d^6 + 36*A*B*C^2*a^4*b^4*c*d^7 + 36*A*B*C^2*a*b^7*c^
4*d^4 + 36*A*B*C^2*a*b^7*c^2*d^6 - 30*A^2*B*C*a^4*b^4*c*d^7 + 24*A*B^2*C*a^
3*b^5*c*d^7 - 24*A*B*C^2*a^2*b^6*c*d^7 + 18*A*B^2*C*a*b^7*c^5*d^3 - 18*A*B*
C^2*a*b^7*c^6*d^2 + 12*A*B^2*C*a*b^7*c^3*d^5 + 9*A^2*B*C*a*b^7*c^6*d^2 + 6*
A*B^2*C*a^5*b^3*c*d^7 - 6*A*B*C^2*a^7*b*c^2*d^6 + 3*A^2*B*C*a^7*b*c^2*d^6 -
 18*B^3*C*a^6*b^2*c*d^7 - 18*B*C^3*a^6*b^2*c*d^7 - 14*B^3*C*a^4*b^4*c*d^7 -
 14*B*C^3*a^4*b^4*c*d^7 - 10*B^3*C*a*b^7*c^2*d^6 - 10*B*C^3*a*b^7*c^2*d^6 +
 9*B^3*C*a*b^7*c^6*d^2 + 9*B*C^3*a*b^7*c^6*d^2 - 7*B^3*C*a*b^7*c^4*d^4 - 7*
B*C^3*a*b^7*c^4*d^4 + 6*B^2*C^2*a^7*b*c*d^7 - 4*B^3*C*a^2*b^6*c*d^7 + 4*B^2
*C^2*a*b^7*c*d^7 - 4*B*C^3*a^2*b^6*c*d^7 + 3*B^3*C*a^7*b*c^2*d^6 + 3*B*C^3*
a^7*b*c^2*d^6 + 144*A^3*C*a^3*b^5*c*d^7 + 62*A^3*C*a^5*b^3*c*d^7 + 48*A*C^3
*a^3*b^5*c*d^7 - 36*A^2*C^2*a*b^7*c*d^7 + 26*A*C^3*a^5*b^3*c*d^7 + 20*A^3*C
*a*b^7*c^3*d^5 + 18*A^2*C^2*a^7*b*c*d^7 - 18*A*C^3*a*b^7*c^5*d^3 - 6*A^3*C*
a*b^7*c^5*d^3 - 4*A*C^3*a*b^7*c^3*d^5 - 32*A^3*B*a^2*b^6*c*d^7 - 32*A*B^3*a
^2*b^6*c*d^7 + 22*A^3*B*a*b^7*c^4*d^4 + 22*A*B^3*a*b^7*c^4*d^4 + 16*A^3*B*a
*b^7*c^2*d^6 + 16*A*B^3*a*b^7*c^2*d^6 + 12*A^3*B*a^6*b^2*c*d^7 + 12*A*B^3*a
^6*b^2*c*d^7 + 8*A^3*B*a^4*b^4*c*d^7 - 8*A^2*B^2*a*b^7*c*d^7 + 8*A*B^3*a^4*
b^4*c*d^7 + 36*A^2*B*C*b^8*c^3*d^5 + 24*A*B*C^2*b^8*c^5*d^3 - 18*A^2*B*C*b^
8*c^5*d^3 - 12*A*B*C^2*b^8*c^3*d^5 - 3*A*B^2*C*b^8*c^6*d^2 - 3*A*B^2*C*b^8*
c^4*d^4 - 2*A*B^2*C*b^8*c^2*d^6 + 57*A^2*B*C*a^5*b^3*d^8 + 36*A^2*B*C*a^3*b
^5*d^8 - 30*A*B*C^2*a^5*b^3*d^8 - 18*A*B*C^2*a^3*b^5*d^8 - 9*A*B^2*C*a^4*b^
4*d^8 - 3*A*B^2*C*a^6*b^2*d^8 - 2*A*B^2*C*a^2*b^6*d^8 + 34*B^2*C^2*a^3*b^5*
c^5*d^3 + 28*B^2*C^2*a^5*b^3*c^3*d^5 + 24*B^2*C^2*a^2*b^6*c^4*d^4 - 20*B^2*
C^2*a^4*b^4*c^4*d^4 + 12*B^2*C^2*a^3*b^5*c^3*d^5 + 12*B^2*C^2*a^2*b^6*c^2*d
^6 + 9*B^2*C^2*a^6*b^2*c^4*d^4 + 9*B^2*C^2*a^4*b^4*c^2*d^6 - 9*B^2*C^2*a^2*
b^6*c^6*d^2 - 3*B^2*C^2*a^6*b^2*c^2*d^6 + 159*A^2*C^2*a^4*b^4*c^2*d^6 - 156
*A^2*C^2*a^3*b^5*c^3*d^5 + 90*A^2*C^2*a^3*b^5*c^5*d^3 + 78*A^2*C^2*a^2*b^6*
c^2*d^6 - 63*A^2*C^2*a^4*b^4*c^4*d^4 - 27*A^2*C^2*a^6*b^2*c^2*d^6 - 27*A^2*
C^2*a^2*b^6*c^6*d^2 - 18*A^2*C^2*a^2*b^6*c^4*d^4 + 9*A^2*C^2*a^6*b^2*c^4*d^
4 + 66*A^2*B^2*a^2*b^6*c^2*d^6 + 60*A^2*B^2*a^4*b^4*c^2*d^6 - 48*A^2*B^2*a^
3*b^5*c^3*d^5 + 42*A^2*B^2*a^2*b^6*c^4*d^4 + 28*A^2*B^2*a^5*b^3*c^3*d^5 - 1
7*A^2*B^2*a^4*b^4*c^4*d^4 - 6*A^2*B^2*a^6*b^2*c^2*d^6 + 4*A^2*B^2*a^3*b^5*c
^5*d^3 + 36*A^3*C*a*b^7*c*d^7 - 18*A*C^3*a^7*b*c*d^7 + 12*A*C^3*a*b^7*c*d^7
 - 6*A^3*C*a^7*b*c*d^7 + 24*A^2*B*C*b^8*c*d^7 - 12*A*B*C^2*b^8*c*d^7 + 12*A
^2*B*C*a*b^7*d^8 + 6*A*B*C^2*a^7*b*d^8 - 6*A*B*C^2*a*b^7*d^8 - 3*A^2*B*C*a^
7*b*d^8 - 53*B^3*C*a^3*b^5*c^4*d^4 - 53*B*C^3*a^3*b^5*c^4*d^4 - 32*B^3*C*a^
3*b^5*c^2*d^6 - 32*B*C^3*a^3*b^5*c^2*d^6 - 18*B^3*C*a^5*b^3*c^4*d^4 - 18*B*
C^3*a^5*b^3*c^4*d^4 + 16*B^3*C*a^4*b^4*c^3*d^5 + 16*B*C^3*a^4*b^4*c^3*d^5 -
 12*B^3*C*a^6*b^2*c^3*d^5 + 12*B^3*C*a^4*b^4*c^5*d^3 + 12*B^2*C^2*a^3*b^5*c

*d^7 - 12*B*C^3*a^6*b^2*c^3*d^5 + 12*B*C^3*a^4*b^4*c^5*d^3 + 8*B^3*C*a^2*b^6*c^3*d^5 + 8*B*C^3*a^2*b^6*c^3*d^5 - 6*B^3*C*a^2*b^6*c^5*d^3 + 6*B^2*C^2*a^5*b^3*c*d^7 - 6*B^2*C^2*a*b^7*c^5*d^3 - 6*B*C^3*a^2*b^6*c^5*d^3 - 3*B^3*C*a^3*b^5*c^6*d^2 - 3*B*C^3*a^3*b^5*c^6*d^2 - 175*A^3*C*a^4*b^4*c^2*d^6 + 164*A^3*C*a^3*b^5*c^3*d^5 - 144*A^2*C^2*a^3*b^5*c*d^7 - 124*A^3*C*a^2*b^6*c^2*d^6 - 90*A*C^3*a^3*b^5*c^5*d^3 - 73*A*C^3*a^4*b^4*c^2*d^6 - 66*A^2*C^2*a^5*b^3*c*d^7 + 44*A*C^3*a^3*b^5*c^3*d^5 + 36*A*C^3*a^4*b^4*c^4*d^4 + 30*A^3*C*a^4*b^4*c^4*d^4 - 30*A^3*C*a^3*b^5*c^5*d^3 + 27*A*C^3*a^2*b^6*c^6*d^2 + 21*A*C^3*a^2*b^6*c^4*d^4 + 18*A^2*C^2*a*b^7*c^5*d^3 - 18*A*C^3*a^6*b^2*c^4*d^4 - 16*A*C^3*a^2*b^6*c^2*d^6 + 15*A^3*C*a^6*b^2*c^2*d^6 - 15*A^3*C*a^2*b^6*c^4*d^4 - 12*A^2*C^2*a*b^7*c^3*d^5 + 9*A^3*C*a^2*b^6*c^6*d^2 + 9*A*C^3*a^6*b^2*c^2*d^6 - 80*A^3*B*a^2*b^6*c^3*d^5 - 80*A*B^3*a^2*b^6*c^3*d^5 + 38*A^3*B*a^3*b^5*c^4*d^4 + 38*A*B^3*a^3*b^5*c^4*d^4 - 36*A^2*B^2*a^3*b^5*c*d^7 - 28*A^3*B*a^5*b^3*c^2*d^6 - 28*A^3*B*a^4*b^4*c^3*d^5 - 28*A*B^3*a^5*b^3*c^2*d^6 - 28*A*B^3*a^4*b^4*c^3*d^5 + 20*A^3*B*a^3*b^5*c^2*d^6 + 20*A*B^3*a^3*b^5*c^2*d^6 - 12*A^3*B*a^2*b^6*c^5*d^3 - 12*A^2*B^2*a^5*b^3*c*d^7 - 12*A^2*B^2*a*b^7*c^5*d^3 - 12*A^2*B^2*a*b^7*c^3*d^5 - 12*A*B^3*a^2*b^6*c^5*d^3 + 9*B^2*C^2*b^8*c^4*d^4 + 4*B^2*C^2*b^8*c^2*d^6 + 3*B^2*C^2*b^8*c^6*d^2 - 30*A^2*C^2*b^8*c^4*d^4 + 9*A^2*C^2*b^8*c^6*d^2 + 16*A^2*B^2*b^8*c^2*d^6 + 6*B^2*C^2*a^6*b^2*d^8 + 3*B^2*C^2*a^4*b^4*d^8 + 3*A^2*B^2*b^8*c^4*d^4 + 36*A^2*C^2*a^4*b^4*d^8 + 27*A^2*C^2*a^2*b^6*d^8 - 18*A^2*C^2*a^6*b^2*d^8 + 33*A^2*B^2*a^4*b^4*d^8 + 28*A^2*B^2*a^2*b^6*d^8 + 6*A^2*B^2*a^6*b^2*d^8 + 6*C^4*a*b^7*c^5*d^3 + 4*C^4*a*b^7*c^3*d^5 - 2*C^4*a^5*b^3*c*d^7 + 12*B^4*a^3*b^5*c*d^7 - 12*B^4*a*b^7*c^5*d^3 + 8*B^4*a^5*b^3*c*d^7 - 4*B^4*a*b^7*c^3*d^5 - 48*A^4*a^3*b^5*c*d^7 - 20*A^4*a^5*b^3*c*d^7 - 8*A^4*a*b^7*c^3*d^5 - 10*B^3*C*b^8*c^5*d^3 - 10*B*C^3*b^8*c^5*d^3 - 4*B^3*C*b^8*c^3*d^5 - 4*B*C^3*b^8*c^3*d^5 + 23*A^3*C*b^8*c^4*d^4 - 18*A^3*C*b^8*c^2*d^6 + 11*A*C^3*b^8*c^4*d^4 - 9*A*C^3*b^8*c^6*d^2 + 6*A*C^3*b^8*c^2*d^6 - 3*A^3*C*b^8*c^6*d^2 - 20*A^3*B*b^8*c^3*d^5 - 20*A*B^3*b^8*c^3*d^5 + 4*A^3*B*b^8*c^5*d^3 + 4*A*B^3*b^8*c^5*d^3 - 63*A^3*C*a^4*b^4*d^8 - 54*A^3*C*a^2*b^6*d^8 + 9*A^3*C*a^6*b^2*d^8 + 9*A*C^3*a^6*b^2*d^8 - 3*A*C^3*a^4*b^4*d^8 - 28*A^3*B*a^5*b^3*d^8 - 28*A*B^3*a^5*b^3*d^8 - 18*A^3*B*a^3*b^5*d^8 - 18*A*B^3*a^3*b^5*d^8 + B^3*C*a^5*b^3*c^2*d^6 + B*C^3*a^5*b^3*c^2*d^6 + 6*C^4*a^7*b*c*d^7 + 4*B^4*a*b^7*c*d^7 - 12*A^4*a*b^7*c*d^7 - 12*A^3*B*b^8*c*d^7 - 12*A*B^3*b^8*c*d^7 - 3*B^3*C*a^7*b*d^8 - 3*B*C^3*a^7*b*d^8 - 6*A^3*B*a*b^7*d^8 - 6*A*B^3*a*b^7*d^8 + 30*C^4*a^3*b^5*c^5*d^3 + 19*C^4*a^4*b^4*c^2*d^6 + 9*C^4*a^6*b^2*c^4*d^4 - 9*C^4*a^2*b^6*c^6*d^2 + 4*C^4*a^3*b^5*c^3*d^5 + 4*C^4*a^2*b^6*c^2*d^6 + 3*C^4*a^6*b^2*c^2*d^6 - 3*C^4*a^4*b^4*c^4*d^4 - 3*C^4*a^2*b^6*c^4*d^4 + 28*B^4*a^5*b^3*c^3*d^5 + 27*B^4*a^2*b^6*c^4*d^4 - 17*B^4*a^4*b^4*c^4*d^4 - 10*B^4*a^4*b^4*c^2*d^6 + 8*B^4*a^3*b^5*c^3*d^5 + 8*B^4*a^2*b^6*c^2*d^6 - 6*B^4*a^6*b^2*c^2*d^6 + 4*B^4*a^3*b^5*c^5*d^3 + 70*A^4*a^4*b^4*c^2*d^6 + 58*A^4*a^2*b^6*c^2*d^6 - 56*A^4*a^3*b^5*c^3*d^5 + 15*A^4*a^2*b^6*c^4*d^4 + B^2*C^2*a^2*b^6*d^8 - 18*A^3*C*b^8*d^8 + B^3*C*a^5*b^3*d^8 + B*C^3*a^5*b^3*d^8 + 3*C^4*b^8*c^6*d^2 + 8*B^4*b^8*c^4*d^4 + 4*B^4*b^8*c^2*d^6 + 12*A^4*b^8*c^2*d^6 - 5*A^4*b^8*c^4*d^4 + 6*B^4*a^6*b^2*d^8 + 3*B^4*a^4*b^4*d^8 + 30*A^4*a^4*b^4*d^8 + 27*A^4*

a^2*b^6*d^8 + 9*A^2*C^2*b^8*d^8 + 9*A^2*B^2*b^8*d^8 + 9*A^4*b^8*d^8 + C^4*b
^8*c^4*d^4 + B^4*a^2*b^6*d^8, f, k)*((4*a^7*b^12*d^15 + 12*a^9*b^10*d^15 +
8*a^11*b^8*d^15 - 8*a^13*b^6*d^15 - 12*a^15*b^4*d^15 - 4*a^17*b^2*d^15 + 4*
b^19*c^7*d^8 + 4*b^19*c^9*d^6 - 4*b^19*c^11*d^4 - 4*b^19*c^13*d^2 - 20*a*b^
18*c^6*d^9 - 4*a*b^18*c^8*d^7 + 60*a*b^18*c^10*d^5 + 52*a*b^18*c^12*d^3 + 3
2*a^3*b^16*c^14*d + 48*a^5*b^14*c^14*d - 20*a^6*b^13*c*d^14 + 32*a^7*b^12*c
^14*d - 44*a^8*b^11*c*d^14 + 8*a^9*b^10*c^14*d + 32*a^10*b^9*c*d^14 + 168*a
^12*b^7*c*d^14 + 172*a^14*b^5*c*d^14 + 68*a^16*b^3*c*d^14 + 16*a^18*b*c^3*d
^12 + 8*a^18*b*c^5*d^10 + 36*a^2*b^17*c^5*d^10 - 32*a^2*b^17*c^7*d^8 - 240*
a^2*b^17*c^9*d^6 - 240*a^2*b^17*c^11*d^4 - 68*a^2*b^17*c^13*d^2 - 20*a^3*b^
16*c^4*d^11 + 64*a^3*b^16*c^6*d^9 + 472*a^3*b^16*c^8*d^7 + 704*a^3*b^16*c^1
0*d^5 + 348*a^3*b^16*c^12*d^3 - 20*a^4*b^15*c^3*d^12 + 8*a^4*b^15*c^5*d^10
- 568*a^4*b^15*c^7*d^8 - 1472*a^4*b^15*c^9*d^6 - 1108*a^4*b^15*c^11*d^4 - 2
32*a^4*b^15*c^13*d^2 + 36*a^5*b^14*c^2*d^13 - 104*a^5*b^14*c^4*d^11 + 392*a
^5*b^14*c^6*d^9 + 2016*a^5*b^14*c^8*d^7 + 2308*a^5*b^14*c^10*d^5 + 872*a^5*
b^14*c^12*d^3 + 64*a^6*b^13*c^3*d^12 + 112*a^6*b^13*c^5*d^10 - 1504*a^6*b^1
3*c^7*d^8 - 3316*a^6*b^13*c^9*d^6 - 2112*a^6*b^13*c^11*d^4 - 328*a^6*b^13*c
^13*d^2 + 32*a^7*b^12*c^2*d^13 - 640*a^7*b^12*c^4*d^11 + 32*a^7*b^12*c^6*d^
9 + 3076*a^7*b^12*c^8*d^7 + 3392*a^7*b^12*c^10*d^5 + 1048*a^7*b^12*c^12*d^3
 + 668*a^8*b^11*c^3*d^12 + 1404*a^8*b^11*c^5*d^10 - 976*a^8*b^11*c^7*d^8 -
3484*a^8*b^11*c^9*d^6 - 2028*a^8*b^11*c^11*d^4 - 212*a^8*b^11*c^13*d^2 - 29
2*a^9*b^10*c^2*d^13 - 2028*a^9*b^10*c^4*d^11 - 1864*a^9*b^10*c^6*d^9 + 1724
*a^9*b^10*c^8*d^7 + 2468*a^9*b^10*c^10*d^5 + 612*a^9*b^10*c^12*d^3 + 1648*a
^10*b^9*c^3*d^12 + 3404*a^10*b^9*c^5*d^10 + 1120*a^10*b^9*c^7*d^8 - 1592*a^
10*b^9*c^9*d^6 - 976*a^10*b^9*c^11*d^4 - 52*a^10*b^9*c^13*d^2 - 768*a^11*b^
8*c^2*d^13 - 3092*a^11*b^8*c^4*d^11 - 3296*a^11*b^8*c^6*d^9 - 288*a^11*b^8*
c^8*d^7 + 832*a^11*b^8*c^10*d^5 + 140*a^11*b^8*c^12*d^3 + 1892*a^12*b^7*c^3
*d^12 + 3552*a^12*b^7*c^5*d^10 + 1912*a^12*b^7*c^7*d^8 - 104*a^12*b^7*c^9*d
^6 - 188*a^12*b^7*c^11*d^4 - 772*a^13*b^6*c^2*d^13 - 2368*a^13*b^6*c^4*d^11
 - 2360*a^13*b^6*c^6*d^9 - 664*a^13*b^6*c^8*d^7 + 92*a^13*b^6*c^10*d^5 + 10
88*a^14*b^5*c^3*d^12 + 1752*a^14*b^5*c^5*d^10 + 928*a^14*b^5*c^7*d^8 + 92*a
^14*b^5*c^9*d^6 - 352*a^15*b^4*c^2*d^13 - 856*a^15*b^4*c^4*d^11 - 704*a^15*
b^4*c^6*d^9 - 188*a^15*b^4*c^8*d^7 + 276*a^16*b^3*c^3*d^12 + 348*a^16*b^3*c
^5*d^10 + 140*a^16*b^3*c^7*d^8 - 60*a^17*b^2*c^2*d^13 - 108*a^17*b^2*c^4*d^
11 - 52*a^17*b^2*c^6*d^9 + 8*a*b^18*c^14*d + 8*a^18*b*c*d^14)/(a^14*d^10 +
b^14*c^10 + 4*a^2*b^12*c^10 + 6*a^4*b^10*c^10 + 4*a^6*b^8*c^10 + a^8*b^6*c^
10 + a^6*b^8*d^10 + 4*a^8*b^6*d^10 + 6*a^10*b^4*d^10 + 4*a^12*b^2*d^10 + 2*
a^14*c^2*d^8 + a^14*c^4*d^6 + b^14*c^6*d^4 + 2*b^14*c^8*d^2 - 6*a*b^13*c^5*
d^5 - 12*a*b^13*c^7*d^3 - 24*a^3*b^11*c^9*d - 6*a^5*b^9*c*d^9 - 36*a^5*b^9*
c^9*d - 24*a^7*b^7*c*d^9 - 24*a^7*b^7*c^9*d - 36*a^9*b^5*c*d^9 - 6*a^9*b^5*
c^9*d - 24*a^11*b^3*c*d^9 - 12*a^13*b*c^3*d^7 - 6*a^13*b*c^5*d^5 + 15*a^2*b
^12*c^4*d^6 + 34*a^2*b^12*c^6*d^4 + 23*a^2*b^12*c^8*d^2 - 20*a^3*b^11*c^3*d
^7 - 64*a^3*b^11*c^5*d^5 - 68*a^3*b^11*c^7*d^3 + 15*a^4*b^10*c^2*d^8 + 90*a
^4*b^10*c^4*d^6 + 141*a^4*b^10*c^6*d^4 + 72*a^4*b^10*c^8*d^2 - 92*a^5*b^9*c
^3*d^7 - 202*a^5*b^9*c^5*d^5 - 152*a^5*b^9*c^7*d^3 + 62*a^6*b^8*c^2*d^8 + 2

11*a^6*b^8*c^4*d^6 + 244*a^6*b^8*c^6*d^4 + 98*a^6*b^8*c^8*d^2 - 168*a^7*b^7
*c^3*d^7 - 288*a^7*b^7*c^5*d^5 - 168*a^7*b^7*c^7*d^3 + 98*a^8*b^6*c^2*d^8 +
244*a^8*b^6*c^4*d^6 + 211*a^8*b^6*c^6*d^4 + 62*a^8*b^6*c^8*d^2 - 152*a^9*b
^5*c^3*d^7 - 202*a^9*b^5*c^5*d^5 - 92*a^9*b^5*c^7*d^3 + 72*a^10*b^4*c^2*d^8
+ 141*a^10*b^4*c^4*d^6 + 90*a^10*b^4*c^6*d^4 + 15*a^10*b^4*c^8*d^2 - 68*a^
11*b^3*c^3*d^7 - 64*a^11*b^3*c^5*d^5 - 20*a^11*b^3*c^7*d^3 + 23*a^12*b^2*c^
2*d^8 + 34*a^12*b^2*c^4*d^6 + 15*a^12*b^2*c^6*d^4 - 6*a*b^13*c^9*d - 6*a^13
*b*c*d^9) + (tan(e + f*x)*(6*a^18*b*d^15 + 6*b^19*c^14*d + 8*a^6*b^13*d^15
+ 38*a^8*b^11*d^15 + 78*a^10*b^9*d^15 + 92*a^12*b^7*d^15 + 68*a^14*b^5*d^15
+ 30*a^16*b^3*d^15 + 8*b^19*c^6*d^9 + 22*b^19*c^8*d^7 + 26*b^19*c^10*d^5 +
18*b^19*c^12*d^3 - 48*a*b^18*c^5*d^10 - 128*a*b^18*c^7*d^8 - 144*a*b^18*c^
9*d^6 - 96*a*b^18*c^11*d^4 - 32*a*b^18*c^13*d^2 + 22*a^2*b^17*c^14*d + 28*a
^4*b^15*c^14*d - 48*a^5*b^14*c*d^14 + 12*a^6*b^13*c^14*d - 224*a^7*b^12*c*d
^14 - 2*a^8*b^11*c^14*d - 448*a^9*b^10*c*d^14 - 2*a^10*b^9*c^14*d - 512*a^1
1*b^8*c*d^14 - 368*a^13*b^6*c*d^14 - 160*a^15*b^4*c*d^14 - 32*a^17*b^2*c*d^
14 + 10*a^18*b*c^2*d^13 + 2*a^18*b*c^4*d^11 - 2*a^18*b*c^6*d^9 + 120*a^2*b^
17*c^4*d^11 + 344*a^2*b^17*c^6*d^9 + 406*a^2*b^17*c^8*d^7 + 282*a^2*b^17*c^
10*d^5 + 122*a^2*b^17*c^12*d^3 - 160*a^3*b^16*c^3*d^12 - 608*a^3*b^16*c^5*d
^10 - 848*a^3*b^16*c^7*d^8 - 624*a^3*b^16*c^9*d^6 - 336*a^3*b^16*c^11*d^4 -
112*a^3*b^16*c^13*d^2 + 120*a^4*b^15*c^2*d^13 + 820*a^4*b^15*c^4*d^11 + 14
28*a^4*b^15*c^6*d^9 + 1072*a^4*b^15*c^8*d^7 + 568*a^4*b^15*c^10*d^5 + 252*a
^4*b^15*c^12*d^3 - 832*a^5*b^14*c^3*d^12 - 1904*a^5*b^14*c^5*d^10 - 1520*a^
5*b^14*c^7*d^8 - 544*a^5*b^14*c^9*d^6 - 272*a^5*b^14*c^11*d^4 - 128*a^5*b^1
4*c^13*d^2 + 568*a^6*b^13*c^2*d^13 + 2044*a^6*b^13*c^4*d^11 + 1988*a^6*b^13
*c^6*d^9 + 200*a^6*b^13*c^8*d^7 - 168*a^6*b^13*c^10*d^5 + 148*a^6*b^13*c^12
*d^3 - 1776*a^7*b^12*c^3*d^12 - 2384*a^7*b^12*c^5*d^10 + 80*a^7*b^12*c^7*d^
8 + 1296*a^7*b^12*c^9*d^6 + 352*a^7*b^12*c^11*d^4 - 32*a^7*b^12*c^13*d^2 +
1138*a^8*b^11*c^2*d^13 + 2434*a^8*b^11*c^4*d^11 + 214*a^8*b^11*c^6*d^9 - 26
26*a^8*b^11*c^8*d^7 - 1622*a^8*b^11*c^10*d^5 - 118*a^8*b^11*c^12*d^3 - 2032
*a^9*b^10*c^3*d^12 - 976*a^9*b^10*c^5*d^10 + 3056*a^9*b^10*c^7*d^8 + 3184*a
^9*b^10*c^9*d^6 + 768*a^9*b^10*c^11*d^4 + 32*a^9*b^10*c^13*d^2 + 1282*a^10*
b^9*c^2*d^13 + 1498*a^10*b^9*c^4*d^11 - 2058*a^10*b^9*c^6*d^9 - 4042*a^10*b
^9*c^8*d^7 - 1862*a^10*b^9*c^10*d^5 - 174*a^10*b^9*c^12*d^3 - 1408*a^11*b^8
*c^3*d^12 + 448*a^11*b^8*c^5*d^10 + 3536*a^11*b^8*c^7*d^8 + 2672*a^11*b^8*c
^9*d^6 + 496*a^11*b^8*c^11*d^4 + 16*a^11*b^8*c^13*d^2 + 908*a^12*b^7*c^2*d^
13 + 552*a^12*b^7*c^4*d^11 - 2000*a^12*b^7*c^6*d^9 - 2540*a^12*b^7*c^8*d^7
- 860*a^12*b^7*c^10*d^5 - 56*a^12*b^7*c^12*d^3 - 672*a^13*b^6*c^3*d^12 + 49
6*a^13*b^6*c^5*d^10 + 1648*a^13*b^6*c^7*d^8 + 960*a^13*b^6*c^9*d^6 + 112*a^
13*b^6*c^11*d^4 + 412*a^14*b^5*c^2*d^13 + 208*a^14*b^5*c^4*d^11 - 688*a^14*
b^5*c^6*d^9 - 692*a^14*b^5*c^8*d^7 - 140*a^14*b^5*c^10*d^5 - 240*a^15*b^4*c
^3*d^12 + 112*a^15*b^4*c^5*d^10 + 304*a^15*b^4*c^7*d^8 + 112*a^15*b^4*c^9*d
^6 + 106*a^16*b^3*c^2*d^13 + 66*a^16*b^3*c^4*d^11 - 66*a^16*b^3*c^6*d^9 - 5
6*a^16*b^3*c^8*d^7 - 48*a^17*b^2*c^3*d^12 + 16*a^17*b^2*c^7*d^8))/(a^14*d^1
0 + b^14*c^10 + 4*a^2*b^12*c^10 + 6*a^4*b^10*c^10 + 4*a^6*b^8*c^10 + a^8*b^
6*c^10 + a^6*b^8*d^10 + 4*a^8*b^6*d^10 + 6*a^10*b^4*d^10 + 4*a^12*b^2*d^10

+ 2*a^14*c^2*d^8 + a^14*c^4*d^6 + b^14*c^6*d^4 + 2*b^14*c^8*d^2 - 6*a*b^13*
c^5*d^5 - 12*a*b^13*c^7*d^3 - 24*a^3*b^11*c^9*d - 6*a^5*b^9*c*d^9 - 36*a^5*
b^9*c^9*d - 24*a^7*b^7*c*d^9 - 24*a^7*b^7*c^9*d - 36*a^9*b^5*c*d^9 - 6*a^9*
b^5*c^9*d - 24*a^11*b^3*c*d^9 - 12*a^13*b*c^3*d^7 - 6*a^13*b*c^5*d^5 + 15*a
^2*b^12*c^4*d^6 + 34*a^2*b^12*c^6*d^4 + 23*a^2*b^12*c^8*d^2 - 20*a^3*b^11*c
^3*d^7 - 64*a^3*b^11*c^5*d^5 - 68*a^3*b^11*c^7*d^3 + 15*a^4*b^10*c^2*d^8 +
90*a^4*b^10*c^4*d^6 + 141*a^4*b^10*c^6*d^4 + 72*a^4*b^10*c^8*d^2 - 92*a^5*b
^9*c^3*d^7 - 202*a^5*b^9*c^5*d^5 - 152*a^5*b^9*c^7*d^3 + 62*a^6*b^8*c^2*d^8
 + 211*a^6*b^8*c^4*d^6 + 244*a^6*b^8*c^6*d^4 + 98*a^6*b^8*c^8*d^2 - 168*a^7
*b^7*c^3*d^7 - 288*a^7*b^7*c^5*d^5 - 168*a^7*b^7*c^7*d^3 + 98*a^8*b^6*c^2*d
^8 + 244*a^8*b^6*c^4*d^6 + 211*a^8*b^6*c^6*d^4 + 62*a^8*b^6*c^8*d^2 - 152*a
^9*b^5*c^3*d^7 - 202*a^9*b^5*c^5*d^5 - 92*a^9*b^5*c^7*d^3 + 72*a^10*b^4*c^2
*d^8 + 141*a^10*b^4*c^4*d^6 + 90*a^10*b^4*c^6*d^4 + 15*a^10*b^4*c^8*d^2 - 6
8*a^11*b^3*c^3*d^7 - 64*a^11*b^3*c^5*d^5 - 20*a^11*b^3*c^7*d^3 + 23*a^12*b^
2*c^2*d^8 + 34*a^12*b^2*c^4*d^6 + 15*a^12*b^2*c^6*d^4 - 6*a*b^13*c^9*d - 6*
a^13*b*c*d^9)) - (C*a^15*b*d^13 - A*a^15*b*d^13 - B*b^16*c^12*d - 12*A*a^3*
b^13*d^13 - 48*A*a^5*b^11*d^13 - 76*A*a^7*b^9*d^13 - 45*A*a^9*b^7*d^13 + 5*
A*a^11*b^5*d^13 + 9*A*a^13*b^3*d^13 + 4*B*a^4*b^12*d^13 + 16*B*a^6*b^10*d^1
3 + 35*B*a^8*b^8*d^13 + 33*B*a^10*b^6*d^13 + 5*B*a^12*b^4*d^13 - 5*B*a^14*b
^2*d^13 + 12*A*b^16*c^3*d^10 + 20*A*b^16*c^5*d^8 - 4*A*b^16*c^9*d^4 + 4*A*b
^16*c^11*d^2 + 4*C*a^7*b^9*d^13 - 3*C*a^9*b^7*d^13 - 17*C*a^11*b^5*d^13 - 9
*C*a^13*b^3*d^13 - 8*B*b^16*c^4*d^9 - 16*B*b^16*c^6*d^7 - B*b^16*c^8*d^5 +
6*B*b^16*c^10*d^3 + 4*C*b^16*c^5*d^8 + 12*C*b^16*c^7*d^6 + 4*C*b^16*c^9*d^4
 - 4*C*b^16*c^11*d^2 - 36*A*a*b^15*c^2*d^11 - 92*A*a*b^15*c^4*d^9 - 56*A*a*
b^15*c^6*d^7 + 3*A*a*b^15*c^8*d^5 + 2*A*a*b^15*c^10*d^3 + 36*A*a^2*b^14*c*d
^12 - 3*A*a^3*b^13*c^12*d + 176*A*a^4*b^12*c*d^12 - 3*A*a^5*b^11*c^12*d + 3
80*A*a^6*b^10*c*d^12 - A*a^7*b^9*c^12*d + 396*A*a^8*b^8*c*d^12 + 176*A*a^10
*b^6*c*d^12 + 20*A*a^12*b^4*c*d^12 - 2*A*a^15*b*c^2*d^11 - A*a^15*b*c^4*d^9
 + 20*B*a*b^15*c^3*d^10 + 68*B*a*b^15*c^5*d^8 + 56*B*a*b^15*c^7*d^6 + 4*B*a
*b^15*c^9*d^4 - 4*B*a*b^15*c^11*d^2 - 3*B*a^2*b^14*c^12*d - 4*B*a^3*b^13*c*
d^12 - 3*B*a^4*b^12*c^12*d - 24*B*a^5*b^11*c*d^12 - B*a^6*b^10*c^12*d - 116
*B*a^7*b^9*c*d^12 - 196*B*a^9*b^7*c*d^12 - 120*B*a^11*b^5*c*d^12 - 20*B*a^1
3*b^3*c*d^12 - 4*C*a*b^15*c^4*d^9 - 40*C*a*b^15*c^6*d^7 - 51*C*a*b^15*c^8*d
^5 - 14*C*a*b^15*c^10*d^3 + 3*C*a^3*b^13*c^12*d - 8*C*a^4*b^12*c*d^12 + 3*C
*a^5*b^11*c^12*d - 56*C*a^6*b^10*c*d^12 + C*a^7*b^9*c^12*d - 60*C*a^8*b^8*c
*d^12 + 28*C*a^10*b^6*c*d^12 + 52*C*a^12*b^4*c*d^12 + 12*C*a^14*b^2*c*d^12
+ 2*C*a^15*b*c^2*d^11 + C*a^15*b*c^4*d^9 + 204*A*a^2*b^14*c^3*d^10 + 264*A*
a^2*b^14*c^5*d^8 + 24*A*a^2*b^14*c^7*d^6 - 68*A*a^2*b^14*c^9*d^4 + 4*A*a^2*
b^14*c^11*d^2 - 260*A*a^3*b^13*c^2*d^11 - 608*A*a^3*b^13*c^4*d^9 - 356*A*a^
3*b^13*c^6*d^7 + 33*A*a^3*b^13*c^8*d^5 + 26*A*a^3*b^13*c^10*d^3 + 876*A*a^4
*b^12*c^3*d^10 + 1180*A*a^4*b^12*c^5*d^8 + 368*A*a^4*b^12*c^7*d^6 - 108*A*a
^4*b^12*c^9*d^4 + 4*A*a^4*b^12*c^11*d^2 - 780*A*a^5*b^11*c^2*d^11 - 1866*A*
a^5*b^11*c^4*d^9 - 1320*A*a^5*b^11*c^6*d^7 - 165*A*a^5*b^11*c^8*d^5 + 18*A*
a^5*b^11*c^10*d^3 + 1812*A*a^6*b^10*c^3*d^10 + 2528*A*a^6*b^10*c^5*d^8 + 11
12*A*a^6*b^10*c^7*d^6 + 28*A*a^6*b^10*c^9*d^4 + 12*A*a^6*b^10*c^11*d^2 - 11

44*A*a^7*b^9*c^2*d^11 - 2802*A*a^7*b^9*c^4*d^9 - 2188*A*a^7*b^9*c^6*d^7 - 4
87*A*a^7*b^9*c^8*d^5 - 34*A*a^7*b^9*c^10*d^3 + 1872*A*a^8*b^8*c^3*d^10 + 26
28*A*a^8*b^8*c^5*d^8 + 1272*A*a^8*b^8*c^7*d^6 + 128*A*a^8*b^8*c^9*d^4 + 8*A
*a^8*b^8*c^11*d^2 - 798*A*a^9*b^7*c^2*d^11 - 2007*A*a^9*b^7*c^4*d^9 - 1588*
A*a^9*b^7*c^6*d^7 - 362*A*a^9*b^7*c^8*d^5 - 28*A*a^9*b^7*c^10*d^3 + 872*A*a
^10*b^6*c^3*d^10 + 1200*A*a^10*b^6*c^5*d^8 + 560*A*a^10*b^6*c^7*d^6 + 56*A*
a^10*b^6*c^9*d^4 - 202*A*a^11*b^5*c^2*d^11 - 585*A*a^11*b^5*c^4*d^9 - 448*A
*a^11*b^5*c^6*d^7 - 70*A*a^11*b^5*c^8*d^5 + 136*A*a^12*b^4*c^3*d^10 + 172*A
*a^12*b^4*c^5*d^8 + 56*A*a^12*b^4*c^7*d^6 + 6*A*a^13*b^3*c^2*d^11 - 31*A*a^
13*b^3*c^4*d^9 - 28*A*a^13*b^3*c^6*d^7 + 8*A*a^14*b^2*c^3*d^10 + 8*A*a^14*b
^2*c^5*d^8 - 12*B*a^2*b^14*c^2*d^11 - 132*B*a^2*b^14*c^4*d^9 - 244*B*a^2*b^
14*c^6*d^7 - 103*B*a^2*b^14*c^8*d^5 + 18*B*a^2*b^14*c^10*d^3 + 132*B*a^3*b^
13*c^3*d^10 + 496*B*a^3*b^13*c^5*d^8 + 488*B*a^3*b^13*c^7*d^6 + 132*B*a^3*b
^13*c^9*d^4 + 4*B*a^3*b^13*c^11*d^2 - 44*B*a^4*b^12*c^2*d^11 - 558*B*a^4*b^
12*c^4*d^9 - 1064*B*a^4*b^12*c^6*d^7 - 581*B*a^4*b^12*c^8*d^5 - 30*B*a^4*b^
12*c^10*d^3 + 284*B*a^5*b^11*c^3*d^10 + 1196*B*a^5*b^11*c^5*d^8 + 1224*B*a^
5*b^11*c^7*d^6 + 356*B*a^5*b^11*c^9*d^4 + 20*B*a^5*b^11*c^11*d^2 + 48*B*a^6
*b^10*c^2*d^11 - 694*B*a^6*b^10*c^4*d^9 - 1596*B*a^6*b^10*c^6*d^7 - 959*B*a
^6*b^10*c^8*d^5 - 90*B*a^6*b^10*c^10*d^3 + 28*B*a^7*b^9*c^3*d^10 + 1032*B*a
^7*b^9*c^5*d^8 + 1208*B*a^7*b^9*c^7*d^6 + 332*B*a^7*b^9*c^9*d^4 + 12*B*a^7*
b^9*c^11*d^2 + 302*B*a^8*b^8*c^2*d^11 - 27*B*a^8*b^8*c^4*d^9 - 828*B*a^8*b^
8*c^6*d^7 - 582*B*a^8*b^8*c^8*d^5 - 48*B*a^8*b^8*c^10*d^3 - 424*B*a^9*b^7*c
^3*d^10 + 84*B*a^9*b^7*c^5*d^8 + 416*B*a^9*b^7*c^7*d^6 + 104*B*a^9*b^7*c^9*
d^4 + 342*B*a^10*b^6*c^2*d^11 + 411*B*a^10*b^6*c^4*d^9 - 102*B*a^10*b^6*c^8
*d^5 - 336*B*a^11*b^5*c^3*d^10 - 216*B*a^11*b^5*c^5*d^8 + 118*B*a^12*b^4*c^
2*d^11 + 181*B*a^12*b^4*c^4*d^9 + 68*B*a^12*b^4*c^6*d^7 - 56*B*a^13*b^3*c^3
*d^10 - 36*B*a^13*b^3*c^5*d^8 - 2*B*a^14*b^2*c^2*d^11 + 3*B*a^14*b^2*c^4*d^
9 - 12*C*a^2*b^14*c^3*d^10 + 36*C*a^2*b^14*c^5*d^8 + 144*C*a^2*b^14*c^7*d^6
 + 92*C*a^2*b^14*c^9*d^4 - 4*C*a^2*b^14*c^11*d^2 + 20*C*a^3*b^13*c^2*d^11 +
 56*C*a^3*b^13*c^4*d^9 - 124*C*a^3*b^13*c^6*d^7 - 237*C*a^3*b^13*c^8*d^5 -
74*C*a^3*b^13*c^10*d^3 - 168*C*a^4*b^12*c^3*d^10 - 172*C*a^4*b^12*c^5*d^8 +
 196*C*a^4*b^12*c^7*d^6 + 204*C*a^4*b^12*c^9*d^4 - 4*C*a^4*b^12*c^11*d^2 +
156*C*a^5*b^11*c^2*d^11 + 570*C*a^5*b^11*c^4*d^9 + 336*C*a^5*b^11*c^6*d^7 -
 171*C*a^5*b^11*c^8*d^5 - 90*C*a^5*b^11*c^10*d^3 - 636*C*a^6*b^10*c^3*d^10
- 1004*C*a^6*b^10*c^5*d^8 - 296*C*a^6*b^10*c^7*d^6 + 116*C*a^6*b^10*c^9*d^4
 - 12*C*a^6*b^10*c^11*d^2 + 328*C*a^7*b^9*c^2*d^11 + 1218*C*a^7*b^9*c^4*d^9
 + 1132*C*a^7*b^9*c^6*d^7 + 223*C*a^7*b^9*c^8*d^5 - 14*C*a^7*b^9*c^10*d^3 -
 828*C*a^8*b^8*c^3*d^10 - 1452*C*a^8*b^8*c^5*d^8 - 708*C*a^8*b^8*c^7*d^6 -
32*C*a^8*b^8*c^9*d^4 - 8*C*a^8*b^8*c^11*d^2 + 234*C*a^9*b^7*c^2*d^11 + 951*
C*a^9*b^7*c^4*d^9 + 964*C*a^9*b^7*c^6*d^7 + 266*C*a^9*b^7*c^8*d^5 + 16*C*a^
9*b^7*c^10*d^3 - 344*C*a^10*b^6*c^3*d^10 - 732*C*a^10*b^6*c^5*d^8 - 392*C*a
^10*b^6*c^7*d^6 - 32*C*a^10*b^6*c^9*d^4 + 10*C*a^11*b^5*c^2*d^11 + 225*C*a^
11*b^5*c^4*d^9 + 256*C*a^11*b^5*c^6*d^7 + 58*C*a^11*b^5*c^8*d^5 + 20*C*a^12
*b^4*c^3*d^10 - 76*C*a^12*b^4*c^5*d^8 - 44*C*a^12*b^4*c^7*d^6 - 30*C*a^13*b
^3*c^2*d^11 - 17*C*a^13*b^3*c^4*d^9 + 4*C*a^13*b^3*c^6*d^7 + 16*C*a^14*b^2*

c^3*d^10 + 4*C*a^14*b^2*c^5*d^8 - A*a*b^15*c^12*d + C*a*b^15*c^12*d)/(a^14*
d^10 + b^14*c^10 + 4*a^2*b^12*c^10 + 6*a^4*b^10*c^10 + 4*a^6*b^8*c^10 + a^8
*b^6*c^10 + a^6*b^8*d^10 + 4*a^8*b^6*d^10 + 6*a^10*b^4*d^10 + 4*a^12*b^2*d^
10 + 2*a^14*c^2*d^8 + a^14*c^4*d^6 + b^14*c^6*d^4 + 2*b^14*c^8*d^2 - 6*a*b^
13*c^5*d^5 - 12*a*b^13*c^7*d^3 - 24*a^3*b^11*c^9*d - 6*a^5*b^9*c*d^9 - 36*a
^5*b^9*c^9*d - 24*a^7*b^7*c*d^9 - 24*a^7*b^7*c^9*d - 36*a^9*b^5*c*d^9 - 6*a
^9*b^5*c^9*d - 24*a^11*b^3*c*d^9 - 12*a^13*b*c^3*d^7 - 6*a^13*b*c^5*d^5 + 1
5*a^2*b^12*c^4*d^6 + 34*a^2*b^12*c^6*d^4 + 23*a^2*b^12*c^8*d^2 - 20*a^3*b^1
1*c^3*d^7 - 64*a^3*b^11*c^5*d^5 - 68*a^3*b^11*c^7*d^3 + 15*a^4*b^10*c^2*d^8
 + 90*a^4*b^10*c^4*d^6 + 141*a^4*b^10*c^6*d^4 + 72*a^4*b^10*c^8*d^2 - 92*a^
5*b^9*c^3*d^7 - 202*a^5*b^9*c^5*d^5 - 152*a^5*b^9*c^7*d^3 + 62*a^6*b^8*c^2*
d^8 + 211*a^6*b^8*c^4*d^6 + 244*a^6*b^8*c^6*d^4 + 98*a^6*b^8*c^8*d^2 - 168*
a^7*b^7*c^3*d^7 - 288*a^7*b^7*c^5*d^5 - 168*a^7*b^7*c^7*d^3 + 98*a^8*b^6*c^
2*d^8 + 244*a^8*b^6*c^4*d^6 + 211*a^8*b^6*c^6*d^4 + 62*a^8*b^6*c^8*d^2 - 15
2*a^9*b^5*c^3*d^7 - 202*a^9*b^5*c^5*d^5 - 92*a^9*b^5*c^7*d^3 + 72*a^10*b^4*
c^2*d^8 + 141*a^10*b^4*c^4*d^6 + 90*a^10*b^4*c^6*d^4 + 15*a^10*b^4*c^8*d^2
- 68*a^11*b^3*c^3*d^7 - 64*a^11*b^3*c^5*d^5 - 20*a^11*b^3*c^7*d^3 + 23*a^12
*b^2*c^2*d^8 + 34*a^12*b^2*c^4*d^6 + 15*a^12*b^2*c^6*d^4 - 6*a*b^13*c^9*d -
 6*a^13*b*c*d^9) + (tan(e + f*x)*(3*B*a^15*b*d^13 - 3*A*b^16*c^12*d + 3*C*b
^16*c^12*d - 24*A*a^4*b^12*d^13 - 104*A*a^6*b^10*d^13 - 199*A*a^8*b^8*d^13
- 189*A*a^10*b^6*d^13 - 77*A*a^12*b^4*d^13 - 7*A*a^14*b^2*d^13 + 8*B*a^5*b^
11*d^13 + 24*B*a^7*b^9*d^13 + 51*B*a^9*b^7*d^13 + 65*B*a^11*b^5*d^13 + 33*B
*a^13*b^3*d^13 + 24*A*b^16*c^4*d^9 + 56*A*b^16*c^6*d^7 + 25*A*b^16*c^8*d^5
- 10*A*b^16*c^10*d^3 - 4*C*a^6*b^10*d^13 + 7*C*a^8*b^8*d^13 + 21*C*a^10*b^6
*d^13 + 5*C*a^12*b^4*d^13 - 5*C*a^14*b^2*d^13 - 16*B*b^16*c^5*d^8 - 48*B*b^
16*c^7*d^6 - 36*B*b^16*c^9*d^4 - 4*B*b^16*c^11*d^2 + 4*C*b^16*c^6*d^7 + 23*
C*b^16*c^8*d^5 + 22*C*b^16*c^10*d^3 - 48*A*a*b^15*c^3*d^10 - 144*A*a*b^15*c
^5*d^8 - 104*A*a*b^15*c^7*d^6 + 4*A*a*b^15*c^9*d^4 + 12*A*a*b^15*c^11*d^2 -
 A*a^2*b^14*c^12*d + 48*A*a^3*b^13*c*d^12 + 7*A*a^4*b^12*c^12*d + 208*A*a^5
*b^11*c*d^12 + 5*A*a^6*b^10*c^12*d + 472*A*a^7*b^9*c*d^12 + 572*A*a^9*b^7*c
*d^12 + 324*A*a^11*b^5*c*d^12 + 68*A*a^13*b^3*c*d^12 + 4*A*a^15*b*c^3*d^10
+ 24*B*a*b^15*c^4*d^9 + 120*B*a*b^15*c^6*d^7 + 147*B*a*b^15*c^8*d^5 + 58*B*
a*b^15*c^10*d^3 + 13*B*a^3*b^13*c^12*d + 5*B*a^5*b^11*c^12*d + 64*B*a^6*b^1
0*c*d^12 - B*a^7*b^9*c^12*d + 100*B*a^8*b^8*c*d^12 - 4*B*a^10*b^6*c*d^12 -
52*B*a^12*b^4*c*d^12 - 12*B*a^14*b^2*c*d^12 + 2*B*a^15*b*c^2*d^11 - B*a^15*
b*c^4*d^9 + 24*C*a*b^15*c^5*d^8 + 8*C*a*b^15*c^7*d^6 - 28*C*a*b^15*c^9*d^4
- 12*C*a*b^15*c^11*d^2 + C*a^2*b^14*c^12*d - 7*C*a^4*b^12*c^12*d + 8*C*a^5*
b^11*c*d^12 - 5*C*a^6*b^10*c^12*d - 88*C*a^7*b^9*c*d^12 - 236*C*a^9*b^7*c*d
^12 - 180*C*a^11*b^5*c*d^12 - 44*C*a^13*b^3*c*d^12 - 4*C*a^15*b*c^3*d^10 +
200*A*a^2*b^14*c^4*d^9 + 468*A*a^2*b^14*c^6*d^7 + 283*A*a^2*b^14*c^8*d^5 +
14*A*a^2*b^14*c^10*d^3 - 192*A*a^3*b^13*c^3*d^10 - 936*A*a^3*b^13*c^5*d^8 -
 952*A*a^3*b^13*c^7*d^6 - 268*A*a^3*b^13*c^9*d^4 - 12*A*a^3*b^13*c^11*d^2 -
 24*A*a^4*b^12*c^2*d^11 + 790*A*a^4*b^12*c^4*d^9 + 1768*A*a^4*b^12*c^6*d^7
+ 1137*A*a^4*b^12*c^8*d^5 + 166*A*a^4*b^12*c^10*d^3 - 200*A*a^5*b^11*c^3*d^
10 - 2016*A*a^5*b^11*c^5*d^8 - 2264*A*a^5*b^11*c^7*d^6 - 716*A*a^5*b^11*c^9

*d^4 - 60*A*a^5*b^11*c^11*d^2 - 316*A*a^6*b^10*c^2*d^11 + 906*A*a^6*b^10*c^
4*d^9 + 2524*A*a^6*b^10*c^6*d^7 + 1651*A*a^6*b^10*c^8*d^5 + 250*A*a^6*b^10*
c^10*d^3 + 472*A*a^7*b^9*c^3*d^10 - 1512*A*a^7*b^9*c^5*d^8 - 2088*A*a^7*b^9
*c^7*d^6 - 612*A*a^7*b^9*c^9*d^4 - 36*A*a^7*b^9*c^11*d^2 - 838*A*a^8*b^8*c^
2*d^11 - 177*A*a^8*b^8*c^4*d^9 + 1252*A*a^8*b^8*c^6*d^7 + 898*A*a^8*b^8*c^8
*d^5 + 108*A*a^8*b^8*c^10*d^3 + 1148*A*a^9*b^7*c^3*d^10 + 72*A*a^9*b^7*c^5*
d^8 - 672*A*a^9*b^7*c^7*d^6 - 168*A*a^9*b^7*c^9*d^4 - 858*A*a^10*b^6*c^2*d^
11 - 795*A*a^10*b^6*c^4*d^9 + 126*A*a^10*b^6*c^8*d^5 + 756*A*a^11*b^5*c^3*d
^10 + 432*A*a^11*b^5*c^5*d^8 - 346*A*a^12*b^4*c^2*d^11 - 353*A*a^12*b^4*c^4
*d^9 - 84*A*a^12*b^4*c^6*d^7 + 140*A*a^13*b^3*c^3*d^10 + 72*A*a^13*b^3*c^5*
d^8 - 34*A*a^14*b^2*c^2*d^11 - 27*A*a^14*b^2*c^4*d^9 + 16*B*a^2*b^14*c^3*d^
10 - 128*B*a^2*b^14*c^5*d^8 - 408*B*a^2*b^14*c^7*d^6 - 316*B*a^2*b^14*c^9*d
^4 - 52*B*a^2*b^14*c^11*d^2 - 32*B*a^3*b^13*c^2*d^11 + 8*B*a^3*b^13*c^4*d^9
 + 460*B*a^3*b^13*c^6*d^7 + 617*B*a^3*b^13*c^8*d^5 + 210*B*a^3*b^13*c^10*d^
3 + 240*B*a^4*b^12*c^3*d^10 + 144*B*a^4*b^12*c^5*d^8 - 576*B*a^4*b^12*c^7*d
^6 - 564*B*a^4*b^12*c^9*d^4 - 84*B*a^4*b^12*c^11*d^2 - 280*B*a^5*b^11*c^2*d
^11 - 814*B*a^5*b^11*c^4*d^9 - 152*B*a^5*b^11*c^6*d^7 + 587*B*a^5*b^11*c^8*
d^5 + 218*B*a^5*b^11*c^10*d^3 + 968*B*a^6*b^10*c^3*d^10 + 1472*B*a^6*b^10*c
^5*d^8 + 328*B*a^6*b^10*c^7*d^6 - 268*B*a^6*b^10*c^9*d^4 - 28*B*a^6*b^10*c^
11*d^2 - 612*B*a^7*b^9*c^2*d^11 - 2034*B*a^7*b^9*c^4*d^9 - 1596*B*a^7*b^9*c
^6*d^7 - 159*B*a^7*b^9*c^8*d^5 + 38*B*a^7*b^9*c^10*d^3 + 1348*B*a^8*b^8*c^3
*d^10 + 2232*B*a^8*b^8*c^5*d^8 + 1048*B*a^8*b^8*c^7*d^6 + 72*B*a^8*b^8*c^9*
d^4 + 8*B*a^8*b^8*c^11*d^2 - 474*B*a^9*b^7*c^2*d^11 - 1731*B*a^9*b^7*c^4*d^
9 - 1524*B*a^9*b^7*c^6*d^7 - 346*B*a^9*b^7*c^8*d^5 - 28*B*a^9*b^7*c^10*d^3
 + 668*B*a^10*b^6*c^3*d^10 + 1176*B*a^10*b^6*c^5*d^8 + 560*B*a^10*b^6*c^7*d^
6 + 56*B*a^10*b^6*c^9*d^4 - 70*B*a^11*b^5*c^2*d^11 - 513*B*a^11*b^5*c^4*d^9
 - 448*B*a^11*b^5*c^6*d^7 - 70*B*a^11*b^5*c^8*d^5 + 60*B*a^12*b^4*c^3*d^10
 + 168*B*a^12*b^4*c^5*d^8 + 56*B*a^12*b^4*c^7*d^6 + 42*B*a^13*b^3*c^2*d^11 -
 19*B*a^13*b^3*c^4*d^9 - 28*B*a^13*b^3*c^6*d^7 - 4*B*a^14*b^2*c^3*d^10 + 8*
B*a^14*b^2*c^5*d^8 - 92*C*a^2*b^14*c^4*d^9 - 204*C*a^2*b^14*c^6*d^7 - 79*C*
a^2*b^14*c^8*d^5 + 34*C*a^2*b^14*c^10*d^3 + 96*C*a^3*b^13*c^3*d^10 + 504*C*
a^3*b^13*c^5*d^8 + 568*C*a^3*b^13*c^7*d^6 + 172*C*a^3*b^13*c^9*d^4 + 12*C*a
^3*b^13*c^11*d^2 - 36*C*a^4*b^12*c^2*d^11 - 646*C*a^4*b^12*c^4*d^9 - 1324*C
*a^4*b^12*c^6*d^7 - 801*C*a^4*b^12*c^8*d^5 - 94*C*a^4*b^12*c^10*d^3 + 344*C
*a^5*b^11*c^3*d^10 + 1512*C*a^5*b^11*c^5*d^8 + 1688*C*a^5*b^11*c^7*d^6 + 57
2*C*a^5*b^11*c^9*d^4 + 60*C*a^5*b^11*c^11*d^2 + 52*C*a^6*b^10*c^2*d^11 - 94
2*C*a^6*b^10*c^4*d^9 - 2188*C*a^6*b^10*c^6*d^7 - 1387*C*a^6*b^10*c^8*d^5 -
202*C*a^6*b^10*c^10*d^3 + 104*C*a^7*b^9*c^3*d^10 + 1416*C*a^7*b^9*c^5*d^8 +
 1704*C*a^7*b^9*c^7*d^6 + 516*C*a^7*b^9*c^9*d^4 + 36*C*a^7*b^9*c^11*d^2 + 3
82*C*a^8*b^8*c^2*d^11 - 87*C*a^8*b^8*c^4*d^9 - 1168*C*a^8*b^8*c^6*d^7 - 802
*C*a^8*b^8*c^8*d^5 - 96*C*a^8*b^8*c^10*d^3 - 524*C*a^9*b^7*c^3*d^10 + 144*C
*a^9*b^7*c^5*d^8 + 576*C*a^9*b^7*c^7*d^6 + 144*C*a^9*b^7*c^9*d^4 + 474*C*a^
10*b^6*c^2*d^11 + 543*C*a^10*b^6*c^4*d^9 - 24*C*a^10*b^6*c^6*d^7 - 114*C*a^
10*b^6*c^8*d^5 - 468*C*a^11*b^5*c^3*d^10 - 288*C*a^11*b^5*c^5*d^8 + 190*C*a
^12*b^4*c^2*d^11 + 257*C*a^12*b^4*c^4*d^9 + 72*C*a^12*b^4*c^6*d^7 - 92*C*a^

13*b^3*c^3*d^10 - 48*C*a^13*b^3*c^5*d^8 + 10*C*a^14*b^2*c^2*d^11 + 15*C*a^1
4*b^2*c^4*d^9 + 4*A*a^15*b*c*d^12 + 7*B*a*b^15*c^12*d - 4*C*a^15*b*c*d^12))
/(a^14*d^10 + b^14*c^10 + 4*a^2*b^12*c^10 + 6*a^4*b^10*c^10 + 4*a^6*b^8*c^1
0 + a^8*b^6*c^10 + a^6*b^8*d^10 + 4*a^8*b^6*d^10 + 6*a^10*b^4*d^10 + 4*a^12
*b^2*d^10 + 2*a^14*c^2*d^8 + a^14*c^4*d^6 + b^14*c^6*d^4 + 2*b^14*c^8*d^2 -
6*a*b^13*c^5*d^5 - 12*a*b^13*c^7*d^3 - 24*a^3*b^11*c^9*d - 6*a^5*b^9*c*d^9
- 36*a^5*b^9*c^9*d - 24*a^7*b^7*c*d^9 - 24*a^7*b^7*c^9*d - 36*a^9*b^5*c*d^
9 - 6*a^9*b^5*c^9*d - 24*a^11*b^3*c*d^9 - 12*a^13*b*c^3*d^7 - 6*a^13*b*c^5*
d^5 + 15*a^2*b^12*c^4*d^6 + 34*a^2*b^12*c^6*d^4 + 23*a^2*b^12*c^8*d^2 - 20*
a^3*b^11*c^3*d^7 - 64*a^3*b^11*c^5*d^5 - 68*a^3*b^11*c^7*d^3 + 15*a^4*b^10*
c^2*d^8 + 90*a^4*b^10*c^4*d^6 + 141*a^4*b^10*c^6*d^4 + 72*a^4*b^10*c^8*d^2
- 92*a^5*b^9*c^3*d^7 - 202*a^5*b^9*c^5*d^5 - 152*a^5*b^9*c^7*d^3 + 62*a^6*b
^8*c^2*d^8 + 211*a^6*b^8*c^4*d^6 + 244*a^6*b^8*c^6*d^4 + 98*a^6*b^8*c^8*d^2
- 168*a^7*b^7*c^3*d^7 - 288*a^7*b^7*c^5*d^5 - 168*a^7*b^7*c^7*d^3 + 98*a^8
*b^6*c^2*d^8 + 244*a^8*b^6*c^4*d^6 + 211*a^8*b^6*c^6*d^4 + 62*a^8*b^6*c^8*d
^2 - 152*a^9*b^5*c^3*d^7 - 202*a^9*b^5*c^5*d^5 - 92*a^9*b^5*c^7*d^3 + 72*a^
10*b^4*c^2*d^8 + 141*a^10*b^4*c^4*d^6 + 90*a^10*b^4*c^6*d^4 + 15*a^10*b^4*c
^8*d^2 - 68*a^11*b^3*c^3*d^7 - 64*a^11*b^3*c^5*d^5 - 20*a^11*b^3*c^7*d^3 +
23*a^12*b^2*c^2*d^8 + 34*a^12*b^2*c^4*d^6 + 15*a^12*b^2*c^6*d^4 - 6*a*b^13*
c^9*d - 6*a^13*b*c*d^9)) + (156*A^2*a^3*b^10*d^11 + 204*A^2*a^5*b^8*d^11 +
85*A^2*a^7*b^6*d^11 + 3*A^2*a^11*b^2*d^11 + 4*B^2*a^3*b^10*d^11 + 28*B^2*a^
5*b^8*d^11 + 45*B^2*a^7*b^6*d^11 + 24*B^2*a^9*b^4*d^11 - B^2*a^11*b^2*d^11
+ 36*A^2*b^13*c^3*d^8 - 4*A^2*b^13*c^5*d^6 - 3*A^2*b^13*c^7*d^4 + C^2*a^7*b
^6*d^11 + 3*C^2*a^11*b^2*d^11 + 16*B^2*b^13*c^3*d^8 + 16*B^2*b^13*c^5*d^6 +
B^2*b^13*c^7*d^4 + 2*B^2*b^13*c^9*d^2 + 8*C^2*b^13*c^5*d^6 + 9*C^2*b^13*c^
7*d^4 + 36*A^2*a*b^12*d^11 + 36*A^2*b^13*c*d^10 + 8*A^2*a^2*b^11*c^3*d^8 +
17*A^2*a^2*b^11*c^5*d^6 + 23*A^2*a^2*b^11*c^7*d^4 - 8*A^2*a^2*b^11*c^9*d^2
+ 168*A^2*a^3*b^10*c^2*d^9 + 87*A^2*a^3*b^10*c^4*d^7 + A^2*a^3*b^10*c^6*d^5
- 417*A^2*a^4*b^9*c^3*d^8 - 205*A^2*a^4*b^9*c^5*d^6 + 23*A^2*a^4*b^9*c^7*d
^4 + 16*A^2*a^4*b^9*c^9*d^2 + 393*A^2*a^5*b^8*c^2*d^9 + 359*A^2*a^5*b^8*c^4
*d^7 + 13*A^2*a^5*b^8*c^6*d^5 - 53*A^2*a^5*b^8*c^8*d^3 - 411*A^2*a^6*b^7*c^
3*d^8 - 13*A^2*a^6*b^7*c^5*d^6 + 93*A^2*a^6*b^7*c^7*d^4 + 43*A^2*a^7*b^6*c^
2*d^9 - 75*A^2*a^7*b^6*c^4*d^7 - 89*A^2*a^7*b^6*c^6*d^5 - 7*A^2*a^8*b^5*c^3
*d^8 + 37*A^2*a^8*b^5*c^5*d^6 + 5*A^2*a^9*b^4*c^2*d^9 + 9*A^2*a^9*b^4*c^4*d
^7 - 17*A^2*a^10*b^3*c^3*d^8 + 7*A^2*a^11*b^2*c^2*d^9 + 36*B^2*a^2*b^11*c^3
*d^8 - 11*B^2*a^2*b^11*c^5*d^6 - 13*B^2*a^2*b^11*c^7*d^4 + 12*B^2*a^2*b^11*
c^9*d^2 + 48*B^2*a^3*b^10*c^2*d^9 - 49*B^2*a^3*b^10*c^4*d^7 - 39*B^2*a^3*b^
10*c^6*d^5 - 20*B^2*a^3*b^10*c^8*d^3 + 163*B^2*a^4*b^9*c^3*d^8 + 91*B^2*a^4
*b^9*c^5*d^6 + 3*B^2*a^4*b^9*c^7*d^4 - 14*B^2*a^4*b^9*c^9*d^2 - 47*B^2*a^5*
b^8*c^2*d^9 - 209*B^2*a^5*b^8*c^4*d^7 + 13*B^2*a^5*b^8*c^6*d^5 + 43*B^2*a^5
*b^8*c^8*d^3 - 31*B^2*a^6*b^7*c^3*d^8 - 185*B^2*a^6*b^7*c^5*d^6 - 79*B^2*a^
6*b^7*c^7*d^4 + 131*B^2*a^7*b^6*c^2*d^9 + 149*B^2*a^7*b^6*c^4*d^7 + 119*B^2
*a^7*b^6*c^6*d^5 - 199*B^2*a^8*b^5*c^3*d^8 - 127*B^2*a^8*b^5*c^5*d^6 + 9*B^
2*a^9*b^4*c^2*d^9 - 19*B^2*a^9*b^4*c^4*d^7 + 7*B^2*a^10*b^3*c^3*d^8 - 5*B^2
*a^11*b^2*c^2*d^9 + 20*C^2*a^2*b^11*c^3*d^8 + 41*C^2*a^2*b^11*c^5*d^6 + 11*

C^2*a^2*b^11*c^7*d^4 - 8*C^2*a^2*b^11*c^9*d^2 + 36*C^2*a^3*b^10*c^2*d^9 + 9
9*C^2*a^3*b^10*c^4*d^7 - 11*C^2*a^3*b^10*c^6*d^5 - 69*C^2*a^4*b^9*c^3*d^8 -
97*C^2*a^4*b^9*c^5*d^6 - 37*C^2*a^4*b^9*c^7*d^4 + 16*C^2*a^4*b^9*c^9*d^2 +
141*C^2*a^5*b^8*c^2*d^9 + 179*C^2*a^5*b^8*c^4*d^7 - 119*C^2*a^5*b^8*c^6*d^
5 - 53*C^2*a^5*b^8*c^8*d^3 + 57*C^2*a^6*b^7*c^3*d^8 + 143*C^2*a^6*b^7*c^5*d
^6 + 57*C^2*a^6*b^7*c^7*d^4 - 65*C^2*a^7*b^6*c^2*d^9 - 231*C^2*a^7*b^6*c^4*
d^7 - 221*C^2*a^7*b^6*c^6*d^5 + 113*C^2*a^8*b^5*c^3*d^8 + 61*C^2*a^8*b^5*c^
5*d^6 + 17*C^2*a^9*b^4*c^2*d^9 - 15*C^2*a^9*b^4*c^4*d^7 - 36*C^2*a^9*b^4*c^
6*d^5 - 65*C^2*a^10*b^3*c^3*d^8 - 36*C^2*a^10*b^3*c^5*d^6 + 7*C^2*a^11*b^2*
c^2*d^9 - 24*A*B*a^2*b^11*d^11 - 136*A*B*a^4*b^9*d^11 - 200*A*B*a^6*b^7*d^1
1 - 89*A*B*a^8*b^5*d^11 + 6*A*B*a^10*b^3*d^11 - 12*A*C*a^3*b^10*d^11 + 12*A
*C*a^5*b^8*d^11 + 58*A*C*a^7*b^6*d^11 + 36*A*C*a^9*b^4*d^11 - 6*A*C*a^11*b^
2*d^11 - 48*A*B*b^13*c^2*d^9 - 48*A*B*b^13*c^4*d^7 - A*B*b^13*c^8*d^3 + 4*B
*C*a^4*b^9*d^11 - 4*B*C*a^6*b^7*d^11 - 19*B*C*a^8*b^5*d^11 - 18*B*C*a^10*b^
3*d^11 + 36*A*C*b^13*c^3*d^8 + 32*A*C*b^13*c^5*d^6 - 6*A*C*b^13*c^7*d^4 - 2
4*B*C*b^13*c^4*d^7 - 24*B*C*b^13*c^6*d^5 + B*C*b^13*c^8*d^3 + 2*A^2*a*b^12*
c^10*d - A^2*a^12*b*c*d^10 - 2*B^2*a*b^12*c^10*d + B^2*a^12*b*c*d^10 + 2*C^
2*a*b^12*c^10*d - C^2*a^12*b*c*d^10 - 44*A^2*a*b^12*c^4*d^7 - 29*A^2*a*b^12
*c^6*d^5 + A^2*a*b^12*c^8*d^3 + 24*A^2*a^2*b^11*c*d^10 - 2*A^2*a^3*b^10*c^1
0*d - 188*A^2*a^4*b^9*c*d^10 - 277*A^2*a^6*b^7*c*d^10 - 27*A^2*a^8*b^5*c*d^
10 - 15*A^2*a^10*b^3*c*d^10 + 32*B^2*a*b^12*c^2*d^9 + 16*B^2*a*b^12*c^4*d^7
- 5*B^2*a*b^12*c^6*d^5 - 11*B^2*a*b^12*c^8*d^3 + 20*B^2*a^2*b^11*c*d^10 +
2*B^2*a^3*b^10*c^10*d + 72*B^2*a^4*b^9*c*d^10 + 47*B^2*a^6*b^7*c*d^10 - 89*
B^2*a^8*b^5*c*d^10 + 5*B^2*a^10*b^3*c*d^10 + 16*C^2*a*b^12*c^4*d^7 - 5*C^2*
a*b^12*c^6*d^5 + C^2*a*b^12*c^8*d^3 - 2*C^2*a^3*b^10*c^10*d - 8*C^2*a^4*b^9
*c*d^10 - C^2*a^6*b^7*c*d^10 + 69*C^2*a^8*b^5*c*d^10 - 27*C^2*a^10*b^3*c*d^
10 - A*B*a^12*b*d^11 + A*B*b^13*c^10*d + B*C*a^12*b*d^11 - B*C*b^13*c^10*d
- 72*A*B*a*b^12*c*d^10 - 4*A*C*a*b^12*c^10*d + 2*A*C*a^12*b*c*d^10 - 24*A*B
*a*b^12*c^3*d^8 + 40*A*B*a*b^12*c^5*d^6 + 32*A*B*a*b^12*c^7*d^4 - 6*A*B*a^2
*b^11*c^10*d - 160*A*B*a^3*b^10*c*d^10 + A*B*a^4*b^9*c^10*d + 56*A*B*a^5*b^
8*c*d^10 + 312*A*B*a^7*b^6*c*d^10 - 8*A*B*a^9*b^4*c*d^10 + A*B*a^12*b*c^2*d
^9 + 36*A*C*a*b^12*c^2*d^9 - 8*A*C*a*b^12*c^4*d^7 - 2*A*C*a*b^12*c^6*d^5 -
2*A*C*a*b^12*c^8*d^3 + 84*A*C*a^2*b^11*c*d^10 + 4*A*C*a^3*b^10*c^10*d + 268
*A*C*a^4*b^9*c*d^10 + 206*A*C*a^6*b^7*c*d^10 - 150*A*C*a^8*b^5*c*d^10 + 6*A
*C*a^10*b^3*c*d^10 - 36*B*C*a*b^12*c^3*d^8 + 8*B*C*a*b^12*c^5*d^6 + 4*B*C*a
*b^12*c^7*d^4 + 6*B*C*a^2*b^11*c^10*d - 20*B*C*a^3*b^10*c*d^10 - B*C*a^4*b^
9*c^10*d - 116*B*C*a^5*b^8*c*d^10 - 180*B*C*a^7*b^6*c*d^10 + 92*B*C*a^9*b^4
*c*d^10 - B*C*a^12*b*c^2*d^9 - 64*A*B*a^2*b^11*c^2*d^9 + 40*A*B*a^2*b^11*c^
4*d^7 + 52*A*B*a^2*b^11*c^6*d^5 - 30*A*B*a^2*b^11*c^8*d^3 - 112*A*B*a^3*b^1
0*c^3*d^8 - 104*A*B*a^3*b^10*c^5*d^6 + 40*A*B*a^3*b^10*c^7*d^4 + 40*A*B*a^3
*b^10*c^9*d^2 - 112*A*B*a^4*b^9*c^2*d^9 + 114*A*B*a^4*b^9*c^4*d^7 - 50*A*B*
a^4*b^9*c^6*d^5 - 105*A*B*a^4*b^9*c^8*d^3 + 480*A*B*a^5*b^8*c^3*d^8 + 368*A
*B*a^5*b^8*c^5*d^6 + 144*A*B*a^5*b^8*c^7*d^4 - 8*A*B*a^5*b^8*c^9*d^2 - 508*
A*B*a^6*b^7*c^2*d^9 - 456*A*B*a^6*b^7*c^4*d^7 - 176*A*B*a^6*b^7*c^6*d^5 + 2
8*A*B*a^6*b^7*c^8*d^3 + 584*A*B*a^7*b^6*c^3*d^8 + 104*A*B*a^7*b^6*c^5*d^6 -

56*A*B*a^7*b^6*c^7*d^4 - 23*A*B*a^8*b^5*c^2*d^9 + 170*A*B*a^8*b^5*c^4*d^7 + 70*A*B*a^8*b^5*c^6*d^5 - 56*A*B*a^9*b^4*c^3*d^8 - 56*A*B*a^9*b^4*c^5*d^6 + 30*A*B*a^10*b^3*c^2*d^9 + 28*A*B*a^10*b^3*c^4*d^7 - 8*A*B*a^11*b^2*c^3*d^8 + 188*A*C*a^2*b^11*c^3*d^8 + 50*A*C*a^2*b^11*c^5*d^6 - 34*A*C*a^2*b^11*c^7*d^4 + 16*A*C*a^2*b^11*c^9*d^2 - 60*A*C*a^3*b^10*c^2*d^9 - 330*A*C*a^3*b^10*c^4*d^7 - 134*A*C*a^3*b^10*c^6*d^5 + 630*A*C*a^4*b^9*c^3*d^8 + 374*A*C*a^4*b^9*c^5*d^6 + 14*A*C*a^4*b^9*c^7*d^4 - 32*A*C*a^4*b^9*c^9*d^2 - 318*A*C*a^5*b^8*c^2*d^9 - 754*A*C*a^5*b^8*c^4*d^7 - 110*A*C*a^5*b^8*c^6*d^5 + 106*A*C*a^5*b^8*c^8*d^3 + 210*A*C*a^6*b^7*c^3*d^8 - 202*A*C*a^6*b^7*c^5*d^6 - 150*A*C*a^6*b^7*c^7*d^4 + 166*A*C*a^7*b^6*c^2*d^9 + 162*A*C*a^7*b^6*c^4*d^7 + 166*A*C*a^7*b^6*c^6*d^5 - 322*A*C*a^8*b^5*c^3*d^8 - 206*A*C*a^8*b^5*c^5*d^6 + 14*A*C*a^9*b^4*c^2*d^9 - 30*A*C*a^9*b^4*c^4*d^7 + 10*A*C*a^10*b^3*c^3*d^8 - 14*A*C*a^11*b^2*c^2*d^9 - 68*B*C*a^2*b^11*c^2*d^9 - 160*B*C*a^2*b^11*c^4*d^7 - 64*B*C*a^2*b^11*c^6*d^5 + 30*B*C*a^2*b^11*c^8*d^3 + 4*B*C*a^3*b^10*c^3*d^8 + 236*B*C*a^3*b^10*c^5*d^6 + 20*B*C*a^3*b^10*c^7*d^4 - 40*B*C*a^3*b^10*c^9*d^2 - 140*B*C*a^4*b^9*c^2*d^9 - 174*B*C*a^4*b^9*c^4*d^7 + 110*B*C*a^4*b^9*c^6*d^5 + 105*B*C*a^4*b^9*c^8*d^3 - 300*B*C*a^5*b^8*c^3*d^8 - 116*B*C*a^5*b^8*c^5*d^6 - 132*B*C*a^5*b^8*c^7*d^4 + 8*B*C*a^5*b^8*c^9*d^2 + 208*B*C*a^6*b^7*c^2*d^9 + 420*B*C*a^6*b^7*c^4*d^7 + 236*B*C*a^6*b^7*c^6*d^5 - 28*B*C*a^6*b^7*c^8*d^3 - 140*B*C*a^7*b^6*c^3*d^8 + 196*B*C*a^7*b^6*c^5*d^6 + 44*B*C*a^7*b^6*c^7*d^4 - 109*B*C*a^8*b^5*c^2*d^9 - 182*B*C*a^8*b^5*c^4*d^7 - 58*B*C*a^8*b^5*c^6*d^5 + 272*B*C*a^9*b^4*c^3*d^8 + 188*B*C*a^9*b^4*c^5*d^6 - 30*B*C*a^10*b^3*c^2*d^9 - 16*B*C*a^10*b^3*c^4*d^7 + 8*B*C*a^11*b^2*c^3*d^8)/(a^14*d^10 + b^14*c^10 + 4*a^2*b^12*c^10 + 6*a^4*b^10*c^10 + 4*a^6*b^8*c^10 + a^8*b^6*c^10 + a^6*b^8*d^10 + 4*a^8*b^6*d^10 + 6*a^10*b^4*d^10 + 4*a^12*b^2*d^10 + 2*a^14*c^2*d^8 + a^14*c^4*d^6 + b^14*c^6*d^4 + 2*b^14*c^8*d^2 - 6*a*b^13*c^5*d^5 - 12*a*b^13*c^7*d^3 - 24*a^3*b^11*c^9*d - 6*a^5*b^9*c*d^9 - 36*a^5*b^9*c^9*d - 24*a^7*b^7*c*d^9 - 24*a^7*b^7*c^9*d - 36*a^9*b^5*c*d^9 - 6*a^9*b^5*c^9*d - 24*a^11*b^3*c*d^9 - 12*a^13*b*c^3*d^7 - 6*a^13*b*c^5*d^5 + 15*a^2*b^12*c^4*d^6 + 34*a^2*b^12*c^6*d^4 + 23*a^2*b^12*c^8*d^2 - 20*a^3*b^11*c^3*d^7 - 64*a^3*b^11*c^5*d^5 - 68*a^3*b^11*c^7*d^3 + 15*a^4*b^10*c^2*d^8 + 90*a^4*b^10*c^4*d^6 + 141*a^4*b^10*c^6*d^4 + 72*a^4*b^10*c^8*d^2 - 92*a^5*b^9*c^3*d^7 - 202*a^5*b^9*c^5*d^5 - 152*a^5*b^9*c^7*d^3 + 62*a^6*b^8*c^2*d^8 + 211*a^6*b^8*c^4*d^6 + 244*a^6*b^8*c^6*d^4 + 98*a^6*b^8*c^8*d^2 - 168*a^7*b^7*c^3*d^7 - 288*a^7*b^7*c^5*d^5 - 168*a^7*b^7*c^7*d^3 + 98*a^8*b^6*c^2*d^8 + 244*a^8*b^6*c^4*d^6 + 211*a^8*b^6*c^6*d^4 + 62*a^8*b^6*c^8*d^2 - 152*a^9*b^5*c^3*d^7 - 202*a^9*b^5*c^5*d^5 - 92*a^9*b^5*c^7*d^3 + 72*a^10*b^4*c^2*d^8 + 141*a^10*b^4*c^4*d^6 + 90*a^10*b^4*c^6*d^4 + 15*a^10*b^4*c^8*d^2 - 68*a^11*b^3*c^3*d^7 - 64*a^11*b^3*c^5*d^5 - 20*a^11*b^3*c^7*d^3 + 23*a^12*b^2*c^2*d^8 + 34*a^12*b^2*c^4*d^6 + 15*a^12*b^2*c^6*d^4 - 6*a*b^13*c^9*d - 6*a^13*b*c*d^9) - (tan(e + f*x)*(20*A^2*a^6*b^7*d^11 - 54*A^2*a^2*b^11*d^11 - 18*A^2*a^4*b^9*d^11 - 18*A^2*b^13*d^11 - 65*A^2*a^8*b^5*d^11 - 2*B^2*a^2*b^11*d^11 - 6*B^2*a^4*b^9*d^11 + 12*B^2*a^6*b^7*d^11 + 66*B^2*a^8*b^5*d^11 - 18*B^2*a^10*b^3*d^11 - 6*A^2*b^13*c^2*d^9 + 10*A^2*b^13*c^4*d^7 + 12*A^2*b^13*c^6*d^5 - 3*A^2*b^13*c^8*d^3 + 2*C^2*a^6*b^7*d^11 - 29*C^

2*a^8*b^5*d^11 + 36*C^2*a^10*b^3*d^11 - 8*B^2*b^13*c^2*d^9 - 8*B^2*b^13*c^4
*d^7 - 18*B^2*b^13*c^6*d^5 - 2*B^2*b^13*c^8*d^3 - 2*C^2*b^13*c^4*d^7 + 6*C^
2*b^13*c^6*d^5 - 9*C^2*b^13*c^8*d^3 - A^2*a^12*b*d^11 - C^2*a^12*b*d^11 - B
^2*b^13*c^10*d - 158*A^2*a^2*b^11*c^2*d^9 - 232*A^2*a^2*b^11*c^4*d^7 - 96*A
^2*a^2*b^11*c^6*d^5 - 34*A^2*a^2*b^11*c^8*d^3 + 504*A^2*a^3*b^10*c^3*d^8 +
248*A^2*a^3*b^10*c^5*d^6 + 120*A^2*a^3*b^10*c^7*d^4 + 28*A^2*a^3*b^10*c^9*d
^2 - 224*A^2*a^4*b^9*c^2*d^9 - 446*A^2*a^4*b^9*c^4*d^7 - 244*A^2*a^4*b^9*c^
6*d^5 - 83*A^2*a^4*b^9*c^8*d^3 + 580*A^2*a^5*b^8*c^3*d^8 + 332*A^2*a^5*b^8*
c^5*d^6 + 132*A^2*a^5*b^8*c^7*d^4 - 252*A^2*a^6*b^7*c^2*d^9 - 452*A^2*a^6*b
^7*c^4*d^7 - 144*A^2*a^6*b^7*c^6*d^5 + 464*A^2*a^7*b^6*c^3*d^8 + 152*A^2*a^
7*b^6*c^5*d^6 - 194*A^2*a^8*b^5*c^2*d^9 - 128*A^2*a^8*b^5*c^4*d^7 + 28*A^2*
a^9*b^4*c^3*d^8 - 2*A^2*a^10*b^3*c^2*d^9 + 18*B^2*a^2*b^11*c^2*d^9 + 4*B^2*
a^2*b^11*c^4*d^7 - 84*B^2*a^2*b^11*c^6*d^5 - 4*B^2*a^2*b^11*c^8*d^3 + 128*B
^2*a^3*b^10*c^3*d^8 + 208*B^2*a^3*b^10*c^5*d^6 + 40*B^2*a^3*b^10*c^7*d^4 -
12*B^2*a^3*b^10*c^9*d^2 + 36*B^2*a^4*b^9*c^2*d^9 - 36*B^2*a^4*b^9*c^4*d^7 -
134*B^2*a^4*b^9*c^6*d^5 + 22*B^2*a^4*b^9*c^8*d^3 + 180*B^2*a^5*b^8*c^3*d^8
+ 148*B^2*a^5*b^8*c^5*d^6 + 20*B^2*a^5*b^8*c^7*d^4 + 8*B^2*a^5*b^8*c^9*d^2
+ 208*B^2*a^6*b^7*c^2*d^9 + 164*B^2*a^6*b^7*c^4*d^7 - 96*B^2*a^6*b^7*c^6*d
^5 - 28*B^2*a^6*b^7*c^8*d^3 - 96*B^2*a^7*b^6*c^3*d^8 + 16*B^2*a^7*b^6*c^5*d
^6 + 48*B^2*a^7*b^6*c^7*d^4 + 179*B^2*a^8*b^5*c^2*d^9 + 76*B^2*a^8*b^5*c^4*
d^7 - 36*B^2*a^8*b^5*c^6*d^5 + 36*B^2*a^9*b^4*c^3*d^8 - 32*B^2*a^10*b^3*c^2
*d^9 - 16*B^2*a^10*b^3*c^4*d^7 + 8*B^2*a^11*b^2*c^3*d^8 - 8*C^2*a^2*b^11*c^
2*d^9 + 44*C^2*a^2*b^11*c^4*d^7 + 90*C^2*a^2*b^11*c^6*d^5 - 28*C^2*a^2*b^11
*c^8*d^3 - 4*C^2*a^3*b^10*c^5*d^6 + 36*C^2*a^3*b^10*c^7*d^4 + 28*C^2*a^3*b^
10*c^9*d^2 + 16*C^2*a^4*b^9*c^2*d^9 + 178*C^2*a^4*b^9*c^4*d^7 + 188*C^2*a^4
*b^9*c^6*d^5 - 53*C^2*a^4*b^9*c^8*d^3 + 64*C^2*a^5*b^8*c^3*d^8 + 80*C^2*a^5
*b^8*c^5*d^6 - 132*C^2*a^6*b^7*c^2*d^9 - 68*C^2*a^6*b^7*c^4*d^7 + 120*C^2*a
^6*b^7*c^6*d^5 + 18*C^2*a^6*b^7*c^8*d^3 + 356*C^2*a^7*b^6*c^3*d^8 + 164*C^2
*a^7*b^6*c^5*d^6 - 60*C^2*a^7*b^6*c^7*d^4 - 104*C^2*a^8*b^5*c^2*d^9 - 68*C^
2*a^8*b^5*c^4*d^7 + 6*C^2*a^8*b^5*c^6*d^5 + 64*C^2*a^9*b^4*c^3*d^8 + 72*C^2
*a^9*b^4*c^5*d^6 + 64*C^2*a^10*b^3*c^2*d^9 + 12*C^2*a^10*b^3*c^4*d^7 - 18*C
^2*a^10*b^3*c^6*d^5 - 12*C^2*a^11*b^2*c^3*d^8 + 36*A*B*a^3*b^10*d^11 - 36*A
*B*a^5*b^8*d^11 - 132*A*B*a^7*b^6*d^11 + 60*A*B*a^9*b^4*d^11 - 4*A*B*a^11*b
^2*d^11 - 18*A*C*a^4*b^9*d^11 + 14*A*C*a^6*b^7*d^11 + 148*A*C*a^8*b^5*d^11
- 18*A*C*a^10*b^3*d^11 + 16*A*B*b^13*c^3*d^8 + 16*A*B*b^13*c^5*d^6 - 8*A*B*
b^13*c^7*d^4 + 2*A*B*b^13*c^9*d^2 + 6*B*C*a^5*b^8*d^11 + 18*B*C*a^7*b^6*d^1
1 - 114*B*C*a^9*b^4*d^11 + 10*B*C*a^11*b^2*d^11 - 12*A*C*b^13*c^2*d^9 + 10*
A*C*b^13*c^4*d^7 + 12*A*C*b^13*c^8*d^3 + 8*B*C*b^13*c^3*d^8 - 4*B*C*b^13*c^
5*d^6 + 20*B*C*b^13*c^7*d^4 - 2*B*C*b^13*c^9*d^2 + 96*A^2*a*b^12*c*d^10 - 8
*B^2*a*b^12*c*d^10 + 136*A^2*a*b^12*c^3*d^8 + 52*A^2*a*b^12*c^5*d^6 + 20*A^
2*a*b^12*c^7*d^4 + 4*A^2*a*b^12*c^9*d^2 - 4*A^2*a^2*b^11*c^10*d + 336*A^2*a
^3*b^10*c*d^10 + 372*A^2*a^5*b^8*c*d^10 + 320*A^2*a^7*b^6*c*d^10 + 40*A^2*a
^9*b^4*c*d^10 + 4*A^2*a^11*b^2*c*d^10 + 48*B^2*a*b^12*c^3*d^8 + 92*B^2*a*b^
12*c^5*d^6 + 36*B^2*a*b^12*c^7*d^4 + 4*B^2*a*b^12*c^9*d^2 + 2*B^2*a^2*b^11*
c^10*d - 16*B^2*a^3*b^10*c*d^10 - B^2*a^4*b^9*c^10*d + 52*B^2*a^5*b^8*c*d^1

0 - 72*B^2*a^7*b^6*c*d^10 + 24*B^2*a^9*b^4*c*d^10 + 4*B^2*a^11*b^2*c*d^10 -
B^2*a^12*b*c^2*d^9 - 8*C^2*a*b^12*c^3*d^8 - 8*C^2*a*b^12*c^5*d^6 + 8*C^2*a
*b^12*c^7*d^4 + 4*C^2*a*b^12*c^9*d^2 - 4*C^2*a^2*b^11*c^10*d - 24*C^2*a^5*b
^8*c*d^10 + 140*C^2*a^7*b^6*c*d^10 + 4*C^2*a^9*b^4*c*d^10 - 8*C^2*a^11*b^2*
c*d^10 + 12*A*B*a*b^12*d^11 + 2*A*C*a^12*b*d^11 + 24*A*B*b^13*c*d^10 - 4*A*
B*a*b^12*c^10*d + 2*A*B*a^12*b*c*d^10 - 24*A*C*a*b^12*c*d^10 + 4*B*C*a*b^12
*c^10*d - 2*B*C*a^12*b*c*d^10 - 140*A*B*a*b^12*c^2*d^9 - 220*A*B*a*b^12*c^4
*d^7 - 68*A*B*a*b^12*c^6*d^5 - 12*A*B*a*b^12*c^8*d^3 + 16*A*B*a^2*b^11*c*d^
10 + 4*A*B*a^3*b^10*c^10*d - 136*A*B*a^4*b^9*c*d^10 + 8*A*B*a^6*b^7*c*d^10
- 174*A*B*a^8*b^5*c*d^10 - 4*A*B*a^10*b^3*c*d^10 + 16*A*C*a*b^12*c^3*d^8 +
28*A*C*a*b^12*c^5*d^6 - 28*A*C*a*b^12*c^7*d^4 - 8*A*C*a*b^12*c^9*d^2 + 8*A*
C*a^2*b^11*c^10*d - 48*A*C*a^3*b^10*c*d^10 + 84*A*C*a^5*b^8*c*d^10 - 172*A*
C*a^7*b^6*c*d^10 + 28*A*C*a^9*b^4*c*d^10 + 4*A*C*a^11*b^2*c*d^10 + 20*B*C*a
*b^12*c^2*d^9 - 14*B*C*a*b^12*c^4*d^7 - 52*B*C*a*b^12*c^6*d^5 - 6*B*C*a*b^1
2*c^8*d^3 + 8*B*C*a^2*b^11*c*d^10 - 4*B*C*a^3*b^10*c^10*d + 28*B*C*a^4*b^9*
c*d^10 - 188*B*C*a^6*b^7*c*d^10 + 114*B*C*a^8*b^5*c*d^10 + 16*B*C*a^10*b^3*
c*d^10 + 64*A*B*a^2*b^11*c^3*d^8 + 184*A*B*a^2*b^11*c^5*d^6 + 32*A*B*a^2*b^
11*c^7*d^4 + 20*A*B*a^2*b^11*c^9*d^2 - 300*A*B*a^3*b^10*c^2*d^9 - 420*A*B*a
^3*b^10*c^4*d^7 - 84*A*B*a^3*b^10*c^6*d^5 - 20*A*B*a^3*b^10*c^8*d^3 + 8*A*B
*a^4*b^9*c^3*d^8 + 292*A*B*a^4*b^9*c^5*d^6 - 40*A*B*a^4*b^9*c^7*d^4 - 30*A*
B*a^4*b^9*c^9*d^2 - 580*A*B*a^5*b^8*c^2*d^9 - 596*A*B*a^5*b^8*c^4*d^7 + 60*
A*B*a^5*b^8*c^6*d^5 + 96*A*B*a^5*b^8*c^8*d^3 + 208*A*B*a^6*b^7*c^3*d^8 + 12
8*A*B*a^6*b^7*c^5*d^6 - 144*A*B*a^6*b^7*c^7*d^4 - 340*A*B*a^7*b^6*c^2*d^9 -
100*A*B*a^7*b^6*c^4*d^7 + 92*A*B*a^7*b^6*c^6*d^5 - 200*A*B*a^8*b^5*c^3*d^8
- 28*A*B*a^8*b^5*c^5*d^6 + 92*A*B*a^9*b^4*c^2*d^9 + 56*A*B*a^9*b^4*c^4*d^7
- 12*A*B*a^11*b^2*c^2*d^9 + 112*A*C*a^2*b^11*c^2*d^9 + 242*A*C*a^2*b^11*c^
4*d^7 + 60*A*C*a^2*b^11*c^6*d^5 + 62*A*C*a^2*b^11*c^8*d^3 + 72*A*C*a^3*b^10
*c^3*d^8 + 44*A*C*a^3*b^10*c^5*d^6 - 156*A*C*a^3*b^10*c^7*d^4 - 56*A*C*a^3*
b^10*c^9*d^2 + 172*A*C*a^4*b^9*c^2*d^9 + 304*A*C*a^4*b^9*c^4*d^7 + 92*A*C*a
^4*b^9*c^6*d^5 + 136*A*C*a^4*b^9*c^8*d^3 + 220*A*C*a^5*b^8*c^3*d^8 + 20*A*C
*a^5*b^8*c^5*d^6 - 132*A*C*a^5*b^8*c^7*d^4 + 420*A*C*a^6*b^7*c^2*d^9 + 484*
A*C*a^6*b^7*c^4*d^7 - 12*A*C*a^6*b^7*c^6*d^5 - 18*A*C*a^6*b^7*c^8*d^3 - 244
*A*C*a^7*b^6*c^3*d^8 - 28*A*C*a^7*b^6*c^5*d^6 + 60*A*C*a^7*b^6*c^7*d^4 + 35
2*A*C*a^8*b^5*c^2*d^9 + 142*A*C*a^8*b^5*c^4*d^7 - 60*A*C*a^8*b^5*c^6*d^5 +
52*A*C*a^9*b^4*c^3*d^8 - 44*A*C*a^10*b^3*c^2*d^9 - 30*A*C*a^10*b^3*c^4*d^7
+ 12*A*C*a^11*b^2*c^3*d^8 - 88*B*C*a^2*b^11*c^3*d^8 - 172*B*C*a^2*b^11*c^5*
d^6 + 28*B*C*a^2*b^11*c^7*d^4 - 20*B*C*a^2*b^11*c^9*d^2 - 66*B*C*a^3*b^10*c
^4*d^7 - 96*B*C*a^3*b^10*c^6*d^5 - 10*B*C*a^3*b^10*c^8*d^3 - 332*B*C*a^4*b^
9*c^3*d^8 - 448*B*C*a^4*b^9*c^5*d^6 + 100*B*C*a^4*b^9*c^7*d^4 + 30*B*C*a^4*
b^9*c^9*d^2 + 160*B*C*a^5*b^8*c^2*d^9 + 248*B*C*a^5*b^8*c^4*d^7 - 24*B*C*a^
5*b^8*c^6*d^5 - 102*B*C*a^5*b^8*c^8*d^3 - 652*B*C*a^6*b^7*c^3*d^8 - 404*B*C
*a^6*b^7*c^5*d^6 + 132*B*C*a^6*b^7*c^7*d^4 - 80*B*C*a^7*b^6*c^2*d^9 - 80*B*
C*a^7*b^6*c^4*d^7 + 40*B*C*a^7*b^6*c^6*d^5 + 6*B*C*a^7*b^6*c^8*d^3 + 68*B*C
*a^8*b^5*c^3*d^8 - 68*B*C*a^8*b^5*c^5*d^6 - 24*B*C*a^8*b^5*c^7*d^4 - 272*B*
C*a^9*b^4*c^2*d^9 - 146*B*C*a^9*b^4*c^4*d^7 + 36*B*C*a^9*b^4*c^6*d^5 + 36*B

*C*a^10*b^3*c^3*d^8 + 24*B*C*a^10*b^3*c^5*d^6 + 12*B*C*a^11*b^2*c^2*d^9 - 6 *B*C*a^11*b^2*c^4*d^7))/(a^14*d^10 + b^14*c^10 + 4*a^2*b^12*c^10 + 6*a^4*b^ 10*c^10 + 4*a^6*b^8*c^10 + a^8*b^6*c^10 + a^6*b^8*d^10 + 4*a^8*b^6*d^10 + 6 *a^10*b^4*d^10 + 4*a^12*b^2*d^10 + 2*a^14*c^2*d^8 + a^14*c^4*d^6 + b^14*c^6 *d^4 + 2*b^14*c^8*d^2 - 6*a*b^13*c^5*d^5 - 12*a*b^13*c^7*d^3 - 24*a^3*b^11* c^9*d - 6*a^5*b^9*c*d^9 - 36*a^5*b^9*c^9*d - 24*a^7*b^7*c*d^9 - 24*a^7*b^7* c^9*d - 36*a^9*b^5*c*d^9 - 6*a^9*b^5*c^9*d - 24*a^11*b^3*c*d^9 - 12*a^13*b* c^3*d^7 - 6*a^13*b*c^5*d^5 + 15*a^2*b^12*c^4*d^6 + 34*a^2*b^12*c^6*d^4 + 23 *a^2*b^12*c^8*d^2 - 20*a^3*b^11*c^3*d^7 - 64*a^3*b^11*c^5*d^5 - 68*a^3*b^11 *c^7*d^3 + 15*a^4*b^10*c^2*d^8 + 90*a^4*b^10*c^4*d^6 + 141*a^4*b^10*c^6*d^4 + 72*a^4*b^10*c^8*d^2 - 92*a^5*b^9*c^3*d^7 - 202*a^5*b^9*c^5*d^5 - 152*a^5 *b^9*c^7*d^3 + 62*a^6*b^8*c^2*d^8 + 211*a^6*b^8*c^4*d^6 + 244*a^6*b^8*c^6*d ^4 + 98*a^6*b^8*c^8*d^2 - 168*a^7*b^7*c^3*d^7 - 288*a^7*b^7*c^5*d^5 - 168*a ^7*b^7*c^7*d^3 + 98*a^8*b^6*c^2*d^8 + 244*a^8*b^6*c^4*d^6 + 211*a^8*b^6*c^6 *d^4 + 62*a^8*b^6*c^8*d^2 - 152*a^9*b^5*c^3*d^7 - 202*a^9*b^5*c^5*d^5 - 92* a^9*b^5*c^7*d^3 + 72*a^10*b^4*c^2*d^8 + 141*a^10*b^4*c^4*d^6 + 90*a^10*b^4* c^6*d^4 + 15*a^10*b^4*c^8*d^2 - 68*a^11*b^3*c^3*d^7 - 64*a^11*b^3*c^5*d^5 - 20*a^11*b^3*c^7*d^3 + 23*a^12*b^2*c^2*d^8 + 34*a^12*b^2*c^4*d^6 + 15*a^12* b^2*c^6*d^4 - 6*a*b^13*c^9*d - 6*a^13*b*c*d^9)) + (tan(e + f*x)*(10*A^3*a^6 *b^4*d^9 - 27*A^3*a^2*b^8*d^9 - 24*A^3*a^4*b^6*d^9 - 9*A^3*b^10*d^9 + B^3*a ^3*b^7*d^9 + B^3*a^5*b^5*d^9 - 12*A^3*b^10*c^2*d^7 - A^3*b^10*c^4*d^5 - C^3 *a^6*b^4*d^9 + 3*C^3*a^8*b^2*d^9 + 4*B^3*b^10*c^5*d^4 + C^3*b^10*c^4*d^5 + 9*A^2*C*b^10*d^9 - 58*A^3*a^2*b^8*c^2*d^7 - 17*A^3*a^2*b^8*c^4*d^5 + 52*A^3 *a^3*b^7*c^3*d^6 - 46*A^3*a^4*b^6*c^2*d^7 - 8*B^3*a^2*b^8*c^3*d^6 - 8*B^3*a ^2*b^8*c^5*d^4 + 16*B^3*a^3*b^7*c^2*d^7 + 17*B^3*a^3*b^7*c^4*d^5 + 20*B^3*a ^4*b^6*c^3*d^6 + 4*B^3*a^4*b^6*c^5*d^4 - 26*B^3*a^5*b^5*c^2*d^7 - 17*B^3*a^ 5*b^5*c^4*d^5 + 28*B^3*a^6*b^4*c^3*d^6 - 6*B^3*a^7*b^3*c^2*d^7 + 4*C^3*a^2* b^8*c^2*d^7 - 10*C^3*a^2*b^8*c^4*d^5 - 12*C^3*a^2*b^8*c^6*d^3 + 20*C^3*a^3* b^7*c^3*d^6 + 36*C^3*a^3*b^7*c^5*d^4 - 2*C^3*a^4*b^6*c^2*d^7 - 6*C^3*a^4*b^ 6*c^4*d^5 + 6*C^3*a^6*b^4*c^2*d^7 + 9*C^3*a^6*b^4*c^4*d^5 + 15*A^2*B*a*b^9* d^9 + 12*A^2*B*b^10*c*d^8 + 12*A^3*a*b^9*c*d^8 - 7*A*B^2*a^2*b^8*d^9 - 15*A *B^2*a^4*b^6*d^9 - 24*A*B^2*a^6*b^4*d^9 + 45*A^2*B*a^3*b^7*d^9 + 56*A^2*B*a ^5*b^5*d^9 - 6*A^2*B*a^7*b^3*d^9 + 3*A*C^2*a^4*b^6*d^9 + 21*A*C^2*a^6*b^4*d ^9 - 6*A*C^2*a^8*b^2*d^9 + 27*A^2*C*a^2*b^8*d^9 + 21*A^2*C*a^4*b^6*d^9 - 30 *A^2*C*a^6*b^4*d^9 + 3*A^2*C*a^8*b^2*d^9 - 4*A*B^2*b^10*c^2*d^7 - 14*A*B^2* b^10*c^4*d^5 - B*C^2*a^5*b^5*d^9 - 9*B*C^2*a^7*b^3*d^9 + 20*A^2*B*b^10*c^3* d^6 + B^2*C*a^2*b^8*d^9 + 3*B^2*C*a^4*b^6*d^9 + 6*B^2*C*a^6*b^4*d^9 + 6*A*C ^2*b^10*c^2*d^7 + 6*A*C^2*b^10*c^4*d^5 + 6*A^2*C*b^10*c^2*d^7 - 6*A^2*C*b^1 0*c^4*d^5 - 4*B*C^2*b^10*c^3*d^6 - 6*B*C^2*b^10*c^5*d^4 + 4*B^2*C*b^10*c^2* d^7 + 8*B^2*C*b^10*c^4*d^5 - 3*B^2*C*b^10*c^6*d^3 + 20*A^3*a*b^9*c^3*d^6 + 36*A^3*a^3*b^7*c*d^8 - 8*A^3*a^5*b^5*c*d^8 + 4*B^3*a*b^9*c^2*d^7 + 2*B^3*a* b^9*c^4*d^5 + 4*B^3*a^2*b^8*c*d^8 + 12*B^3*a^4*b^6*c*d^8 + 24*B^3*a^6*b^4*c *d^8 + 4*C^3*a*b^9*c^3*d^6 + 12*C^3*a*b^9*c^5*d^4 + 8*C^3*a^5*b^5*c*d^8 - 6 *A*B*C*a*b^9*d^9 - 12*A*B*C*b^10*c*d^8 + 8*A*B^2*a^2*b^8*c^2*d^7 - 7*A*B^2* a^2*b^8*c^4*d^5 - 92*A*B^2*a^3*b^7*c^3*d^6 - 16*A*B^2*a^3*b^7*c^5*d^4 + 54*

A*B^2*a^4*b^6*c^2*d^7 + 55*A*B^2*a^4*b^6*c^4*d^5 - 56*A*B^2*a^5*b^5*c^3*d^6 - 22*A*B^2*a^6*b^4*c^2*d^7 + 68*A^2*B*a^2*b^8*c^3*d^6 + 16*A^2*B*a^2*b^8*c^5*d^4 + 46*A^2*B*a^3*b^7*c^2*d^7 - 33*A^2*B*a^3*b^7*c^4*d^5 - 16*A^2*B*a^4*b^6*c^3*d^6 + 82*A^2*B*a^5*b^5*c^2*d^7 - 12*A*C^2*a^2*b^8*c^2*d^7 + 30*A*C^2*a^2*b^8*c^4*d^5 + 24*A*C^2*a^2*b^8*c^6*d^3 + 12*A*C^2*a^3*b^7*c^3*d^6 - 72*A*C^2*a^3*b^7*c^5*d^4 + 12*A*C^2*a^4*b^6*c^2*d^7 + 39*A*C^2*a^4*b^6*c^4*d^5 + 6*A*C^2*a^6*b^4*c^2*d^7 - 9*A*C^2*a^6*b^4*c^4*d^5 + 66*A^2*C*a^2*b^8*c^2*d^7 - 3*A^2*C*a^2*b^8*c^4*d^5 - 12*A^2*C*a^2*b^8*c^6*d^3 - 84*A^2*C*a^3*b^7*c^3*d^6 + 36*A^2*C*a^3*b^7*c^5*d^4 + 36*A^2*C*a^4*b^6*c^2*d^7 - 33*A^2*C*a^4*b^6*c^4*d^5 - 12*A^2*C*a^6*b^4*c^2*d^7 + 8*B*C^2*a^2*b^8*c^3*d^6 + 4*B*C^2*a^2*b^8*c^5*d^4 - 20*B*C^2*a^3*b^7*c^2*d^7 - 66*B*C^2*a^3*b^7*c^4*d^5 - 12*B*C^2*a^3*b^7*c^6*d^3 + 32*B*C^2*a^4*b^6*c^3*d^6 + 42*B*C^2*a^4*b^6*c^5*d^4 + 4*B*C^2*a^5*b^5*c^2*d^7 - 21*B*C^2*a^5*b^5*c^4*d^5 - 12*B*C^2*a^6*b^4*c^3*d^6 + 6*B*C^2*a^7*b^3*c^2*d^7 + 9*B*C^2*a^7*b^3*c^4*d^5 - 2*B^2*C*a^2*b^8*c^2*d^7 + 13*B^2*C*a^2*b^8*c^4*d^5 + 6*B^2*C*a^2*b^8*c^6*d^3 + 32*B^2*C*a^3*b^7*c^3*d^6 + 4*B^2*C*a^3*b^7*c^5*d^4 - 63*B^2*C*a^4*b^6*c^2*d^7 - 73*B^2*C*a^4*b^6*c^4*d^5 - 3*B^2*C*a^4*b^6*c^6*d^3 + 44*B^2*C*a^5*b^5*c^3*d^6 + 12*B^2*C*a^5*b^5*c^5*d^4 - 2*B^2*C*a^6*b^4*c^2*d^7 - 18*B^2*C*a^6*b^4*c^4*d^5 - 12*B^2*C*a^7*b^3*c^3*d^6 + 3*B^2*C*a^8*b^2*c^2*d^7 - 18*A*B*C*a^3*b^7*d^9 - 28*A*B*C*a^5*b^5*d^9 + 24*A*B*C*a^7*b^3*d^9 - 16*A*B*C*b^10*c^3*d^6 + 6*A*B*C*b^10*c^5*d^4 - 16*A*B^2*a*b^9*c*d^8 + 12*A*C^2*a*b^9*c*d^8 - 24*A^2*C*a*b^9*c*d^8 + 4*B^2*C*a*b^9*c*d^8 - 4*A*B^2*a*b^9*c^3*d^6 + 16*A*B^2*a*b^9*c^5*d^4 - 56*A*B^2*a^3*b^7*c*d^8 - 28*A*B^2*a^5*b^5*c*d^8 + 12*A*B^2*a^7*b^3*c*d^8 - 4*A^2*B*a*b^9*c^2*d^7 - 33*A^2*B*a*b^9*c^4*d^5 + 20*A^2*B*a^2*b^8*c*d^8 - 56*A^2*B*a^4*b^6*c*d^8 - 16*A^2*B*a^6*b^4*c*d^8 + 12*A*C^2*a*b^9*c^3*d^6 - 24*A*C^2*a*b^9*c^5*d^4 + 36*A*C^2*a^3*b^7*c*d^8 - 24*A*C^2*a^5*b^5*c*d^8 - 36*A^2*C*a*b^9*c^3*d^6 + 12*A^2*C*a*b^9*c^5*d^4 - 72*A^2*C*a^3*b^7*c*d^8 + 24*A^2*C*a^5*b^5*c*d^8 - 10*B*C^2*a*b^9*c^2*d^7 - 12*B*C^2*a*b^9*c^4*d^5 + 12*B*C^2*a*b^9*c^6*d^3 - 4*B*C^2*a^2*b^8*c*d^8 - 14*B*C^2*a^4*b^6*c*d^8 - 4*B*C^2*a^6*b^4*c*d^8 + 6*B*C^2*a^8*b^2*c*d^8 - 8*B^2*C*a*b^9*c^3*d^6 - 16*B^2*C*a*b^9*c^5*d^4 + 8*B^2*C*a^3*b^7*c*d^8 + 4*B^2*C*a^5*b^5*c*d^8 - 24*B^2*C*a^7*b^3*c*d^8 - 76*A*B*C*a^2*b^8*c^3*d^6 - 20*A*B*C*a^2*b^8*c^5*d^4 + 28*A*B*C*a^3*b^7*c^2*d^7 + 126*A*B*C*a^3*b^7*c^4*d^5 + 12*A*B*C*a^3*b^7*c^6*d^3 - 16*A*B*C*a^4*b^6*c^3*d^6 - 42*A*B*C*a^4*b^6*c^5*d^4 - 32*A*B*C*a^5*b^5*c^2*d^7 + 48*A*B*C*a^5*b^5*c^4*d^5 + 12*A*B*C*a^6*b^4*c^3*d^6 + 12*A*B*C*a^7*b^3*c^2*d^7 + 32*A*B*C*a*b^9*c^2*d^7 + 54*A*B*C*a*b^9*c^4*d^5 - 12*A*B*C*a*b^9*c^6*d^3 - 16*A*B*C*a^2*b^8*c*d^8 + 70*A*B*C*a^4*b^6*c*d^8 + 20*A*B*C*a^6*b^4*c*d^8 - 6*A*B*C*a^8*b^2*c*d^8))/(a^14*d^10 + b^14*c^10 + 4*a^2*b^12*c^10 + 6*a^4*b^10*c^10 + 4*a^6*b^8*c^10 + a^8*b^6*c^10 + a^6*b^8*d^10 + 4*a^8*b^6*d^10 + 6*a^10*b^4*d^10 + 4*a^12*b^2*d^10 + 2*a^14*c^2*d^8 + a^14*c^4*d^6 + b^14*c^6*d^4 + 2*b^14*c^8*d^2 - 6*a*b^13*c^5*d^5 - 12*a*b^13*c^7*d^3 - 24*a^3*b^11*c^9*d - 6*a^5*b^9*c*d^9 - 36*a^5*b^9*c^9*d - 24*a^7*b^7*c*d^9 - 24*a^7*b^7*c^9*d - 36*a^9*b^5*c*d^9 - 6*a^9*b^5*c^9*d - 24*a^11*b^3*c*d^9 - 12*a^13*b*c^3*d^7 - 6*a^13*b*c^5*d^5 + 15*a^2*b^12*c^4*d^6 + 34*a^2*b^12*c^6*d^4 + 23*a^2*b^12*c^8*d^2 - 20*a^3*b^11*c^3*d^7

- 64*a^3*b^11*c^5*d^5 - 68*a^3*b^11*c^7*d^3 + 15*a^4*b^10*c^2*d^8 + 90*a^4
*b^10*c^4*d^6 + 141*a^4*b^10*c^6*d^4 + 72*a^4*b^10*c^8*d^2 - 92*a^5*b^9*c^3
*d^7 - 202*a^5*b^9*c^5*d^5 - 152*a^5*b^9*c^7*d^3 + 62*a^6*b^8*c^2*d^8 + 211
*a^6*b^8*c^4*d^6 + 244*a^6*b^8*c^6*d^4 + 98*a^6*b^8*c^8*d^2 - 168*a^7*b^7*c
^3*d^7 - 288*a^7*b^7*c^5*d^5 - 168*a^7*b^7*c^7*d^3 + 98*a^8*b^6*c^2*d^8 + 2
44*a^8*b^6*c^4*d^6 + 211*a^8*b^6*c^6*d^4 + 62*a^8*b^6*c^8*d^2 - 152*a^9*b^5
*c^3*d^7 - 202*a^9*b^5*c^5*d^5 - 92*a^9*b^5*c^7*d^3 + 72*a^10*b^4*c^2*d^8 +
141*a^10*b^4*c^4*d^6 + 90*a^10*b^4*c^6*d^4 + 15*a^10*b^4*c^8*d^2 - 68*a^11
*b^3*c^3*d^7 - 64*a^11*b^3*c^5*d^5 - 20*a^11*b^3*c^7*d^3 + 23*a^12*b^2*c^2*
d^8 + 34*a^12*b^2*c^4*d^6 + 15*a^12*b^2*c^6*d^4 - 6*a*b^13*c^9*d - 6*a^13*b
*c*d^9))*root(640*a^13*b^7*c*d^15*f^4 + 640*a^7*b^13*c^15*d*f^4 + 480*a^15*
b^5*c*d^15*f^4 + 480*a^11*b^9*c*d^15*f^4 + 480*a^9*b^11*c^15*d*f^4 + 480*a^
5*b^15*c^15*d*f^4 + 192*a^19*b*c^5*d^11*f^4 + 192*a^17*b^3*c*d^15*f^4 + 192
*a^11*b^9*c^15*d*f^4 + 192*a^9*b^11*c*d^15*f^4 + 192*a^3*b^17*c^15*d*f^4 +
192*a*b^19*c^11*d^5*f^4 + 128*a^19*b*c^7*d^9*f^4 + 128*a^19*b*c^3*d^13*f^4
+ 128*a*b^19*c^13*d^3*f^4 + 128*a*b^19*c^9*d^7*f^4 + 32*a^19*b*c^9*d^7*f^4
+ 32*a^13*b^7*c^15*d*f^4 + 32*a^7*b^13*c*d^15*f^4 + 32*a*b^19*c^7*d^9*f^4 +
32*a^19*b*c*d^15*f^4 + 32*a*b^19*c^15*d*f^4 - 47088*a^10*b^10*c^8*d^8*f^4
+ 42432*a^11*b^9*c^7*d^9*f^4 + 42432*a^9*b^11*c^9*d^7*f^4 + 39328*a^11*b^9*
c^9*d^7*f^4 + 39328*a^9*b^11*c^7*d^9*f^4 - 36912*a^12*b^8*c^8*d^8*f^4 - 369
12*a^8*b^12*c^8*d^8*f^4 - 34256*a^10*b^10*c^10*d^6*f^4 - 34256*a^10*b^10*c^
6*d^10*f^4 - 31152*a^12*b^8*c^6*d^10*f^4 - 31152*a^8*b^12*c^10*d^6*f^4 + 28
128*a^13*b^7*c^7*d^9*f^4 + 28128*a^7*b^13*c^9*d^7*f^4 + 24160*a^11*b^9*c^5*
d^11*f^4 + 24160*a^9*b^11*c^11*d^5*f^4 - 23088*a^12*b^8*c^10*d^6*f^4 - 2308
8*a^8*b^12*c^6*d^10*f^4 + 22272*a^13*b^7*c^9*d^7*f^4 + 22272*a^7*b^13*c^7*d
^9*f^4 + 19072*a^11*b^9*c^11*d^5*f^4 + 19072*a^9*b^11*c^5*d^11*f^4 + 18624*
a^13*b^7*c^5*d^11*f^4 + 18624*a^7*b^13*c^11*d^5*f^4 - 17328*a^14*b^6*c^8*d^
8*f^4 - 17328*a^6*b^14*c^8*d^8*f^4 - 17232*a^14*b^6*c^6*d^10*f^4 - 17232*a^
6*b^14*c^10*d^6*f^4 - 13520*a^12*b^8*c^4*d^12*f^4 - 13520*a^8*b^12*c^12*d^4
*f^4 - 12464*a^10*b^10*c^12*d^4*f^4 - 12464*a^10*b^10*c^4*d^12*f^4 + 10880*
a^15*b^5*c^7*d^9*f^4 + 10880*a^5*b^15*c^9*d^7*f^4 - 9072*a^14*b^6*c^10*d^6*
f^4 - 9072*a^6*b^14*c^6*d^10*f^4 + 8928*a^13*b^7*c^11*d^5*f^4 + 8928*a^7*b^
13*c^5*d^11*f^4 - 8880*a^14*b^6*c^4*d^12*f^4 - 8880*a^6*b^14*c^12*d^4*f^4 +
8480*a^15*b^5*c^5*d^11*f^4 + 8480*a^5*b^15*c^11*d^5*f^4 + 7200*a^15*b^5*c^
9*d^7*f^4 + 7200*a^5*b^15*c^7*d^9*f^4 - 6912*a^12*b^8*c^12*d^4*f^4 - 6912*a
^8*b^12*c^4*d^12*f^4 + 6400*a^11*b^9*c^3*d^13*f^4 + 6400*a^9*b^11*c^13*d^3*
f^4 + 5920*a^13*b^7*c^3*d^13*f^4 + 5920*a^7*b^13*c^13*d^3*f^4 - 5392*a^16*b
^4*c^6*d^10*f^4 - 5392*a^4*b^16*c^10*d^6*f^4 - 4428*a^16*b^4*c^8*d^8*f^4 -
4428*a^4*b^16*c^8*d^8*f^4 + 4128*a^11*b^9*c^13*d^3*f^4 + 4128*a^9*b^11*c^3*
d^13*f^4 - 3328*a^16*b^4*c^4*d^12*f^4 - 3328*a^4*b^16*c^12*d^4*f^4 + 3264*a
^15*b^5*c^3*d^13*f^4 + 3264*a^5*b^15*c^13*d^3*f^4 - 2480*a^12*b^8*c^2*d^14*
f^4 - 2480*a^8*b^12*c^14*d^2*f^4 + 2240*a^15*b^5*c^11*d^5*f^4 + 2240*a^5*b^
15*c^5*d^11*f^4 - 2128*a^14*b^6*c^12*d^4*f^4 - 2128*a^6*b^14*c^4*d^12*f^4 +
2112*a^17*b^3*c^7*d^9*f^4 + 2112*a^3*b^17*c^9*d^7*f^4 + 2048*a^17*b^3*c^5*
d^11*f^4 + 2048*a^3*b^17*c^11*d^5*f^4 - 2000*a^14*b^6*c^2*d^14*f^4 - 2000*a

^6*b^14*c^14*d^2*f^4 - 1792*a^16*b^4*c^10*d^6*f^4 - 1792*a^4*b^16*c^6*d^10*
f^4 - 1776*a^10*b^10*c^14*d^2*f^4 - 1776*a^10*b^10*c^2*d^14*f^4 + 1472*a^13
*b^7*c^13*d^3*f^4 + 1472*a^7*b^13*c^3*d^13*f^4 + 1088*a^17*b^3*c^9*d^7*f^4
+ 1088*a^3*b^17*c^7*d^9*f^4 + 992*a^17*b^3*c^3*d^13*f^4 + 992*a^3*b^17*c^13
*d^3*f^4 - 912*a^16*b^4*c^2*d^14*f^4 - 912*a^4*b^16*c^14*d^2*f^4 - 768*a^18
*b^2*c^6*d^10*f^4 - 768*a^2*b^18*c^10*d^6*f^4 - 688*a^12*b^8*c^14*d^2*f^4 -
688*a^8*b^12*c^2*d^14*f^4 - 592*a^18*b^2*c^4*d^12*f^4 - 592*a^2*b^18*c^12*
d^4*f^4 - 472*a^18*b^2*c^8*d^8*f^4 - 472*a^2*b^18*c^8*d^8*f^4 - 280*a^16*b^
4*c^12*d^4*f^4 - 280*a^4*b^16*c^4*d^12*f^4 + 224*a^17*b^3*c^11*d^5*f^4 + 22
4*a^15*b^5*c^13*d^3*f^4 + 224*a^5*b^15*c^3*d^13*f^4 + 224*a^3*b^17*c^5*d^11
*f^4 - 208*a^18*b^2*c^2*d^14*f^4 - 208*a^2*b^18*c^14*d^2*f^4 - 112*a^18*b^2
*c^10*d^6*f^4 - 112*a^14*b^6*c^14*d^2*f^4 - 112*a^6*b^14*c^2*d^14*f^4 - 112
*a^2*b^18*c^6*d^10*f^4 - 24*b^20*c^12*d^4*f^4 - 16*b^20*c^14*d^2*f^4 - 16*b
^20*c^10*d^6*f^4 - 4*b^20*c^8*d^8*f^4 - 24*a^20*c^4*d^12*f^4 - 16*a^20*c^6*
d^10*f^4 - 16*a^20*c^2*d^14*f^4 - 4*a^20*c^8*d^8*f^4 - 80*a^14*b^6*d^16*f^4
- 60*a^16*b^4*d^16*f^4 - 60*a^12*b^8*d^16*f^4 - 24*a^18*b^2*d^16*f^4 - 24*
a^10*b^10*d^16*f^4 - 4*a^8*b^12*d^16*f^4 - 80*a^6*b^14*c^16*f^4 - 60*a^8*b^
12*c^16*f^4 - 60*a^4*b^16*c^16*f^4 - 24*a^10*b^10*c^16*f^4 - 24*a^2*b^18*c^
16*f^4 - 4*a^12*b^8*c^16*f^4 - 4*b^20*c^16*f^4 - 4*a^20*d^16*f^4 + 56*A*C*a
^13*b*c*d^11*f^2 - 48*A*C*a*b^13*c^11*d*f^2 + 48*A*C*a*b^13*c*d^11*f^2 + 59
04*B*C*a^7*b^7*c^6*d^6*f^2 - 5016*B*C*a^8*b^6*c^5*d^7*f^2 - 4608*B*C*a^6*b^
8*c^7*d^5*f^2 - 4512*B*C*a^6*b^8*c^5*d^7*f^2 - 4384*B*C*a^8*b^6*c^7*d^5*f^2
+ 3056*B*C*a^7*b^7*c^8*d^4*f^2 + 2256*B*C*a^7*b^7*c^4*d^8*f^2 - 1824*B*C*a
^8*b^6*c^3*d^9*f^2 + 1632*B*C*a^4*b^10*c^9*d^3*f^2 - 1400*B*C*a^3*b^11*c^8*
d^4*f^2 - 1320*B*C*a^11*b^3*c^4*d^8*f^2 - 1248*B*C*a^6*b^8*c^3*d^9*f^2 + 11
52*B*C*a^10*b^4*c^3*d^9*f^2 - 1072*B*C*a^6*b^8*c^9*d^3*f^2 + 1068*B*C*a^9*b
^5*c^6*d^6*f^2 - 1004*B*C*a^5*b^9*c^4*d^8*f^2 - 968*B*C*a^3*b^11*c^6*d^6*f^
2 - 864*B*C*a^5*b^9*c^8*d^4*f^2 - 828*B*C*a^9*b^5*c^4*d^8*f^2 - 792*B*C*a^1
1*b^3*c^2*d^10*f^2 - 792*B*C*a^3*b^11*c^4*d^8*f^2 - 776*B*C*a^8*b^6*c^9*d^3
*f^2 + 688*B*C*a^4*b^10*c^7*d^5*f^2 - 672*B*C*a^3*b^11*c^10*d^2*f^2 - 592*B
*C*a^9*b^5*c^2*d^10*f^2 + 544*B*C*a^7*b^7*c^10*d^2*f^2 - 492*B*C*a^5*b^9*c^
2*d^10*f^2 + 480*B*C*a^10*b^4*c^5*d^7*f^2 - 392*B*C*a^5*b^9*c^10*d^2*f^2 +
332*B*C*a^9*b^5*c^8*d^4*f^2 - 328*B*C*a^11*b^3*c^6*d^6*f^2 + 320*B*C*a^2*b^
12*c^9*d^3*f^2 + 272*B*C*a^12*b^2*c^3*d^9*f^2 - 248*B*C*a^4*b^10*c^5*d^7*f^
2 - 248*B*C*a^3*b^11*c^2*d^10*f^2 - 208*B*C*a^10*b^4*c^7*d^5*f^2 - 192*B*C*
a^2*b^12*c^5*d^7*f^2 + 144*B*C*a^7*b^7*c^2*d^10*f^2 - 96*B*C*a^4*b^10*c^3*d
^9*f^2 + 88*B*C*a^12*b^2*c^5*d^7*f^2 - 72*B*C*a^11*b^3*c^8*d^4*f^2 - 48*B*C
*a^12*b^2*c^7*d^5*f^2 + 48*B*C*a^10*b^4*c^9*d^3*f^2 - 48*B*C*a^2*b^12*c^7*d
^5*f^2 - 48*B*C*a^2*b^12*c^3*d^9*f^2 - 12*B*C*a^9*b^5*c^10*d^2*f^2 + 4*B*C*
a^5*b^9*c^6*d^6*f^2 + 5824*A*C*a^5*b^9*c^7*d^5*f^2 - 4378*A*C*a^6*b^8*c^8*d
^4*f^2 + 4296*A*C*a^5*b^9*c^5*d^7*f^2 - 3912*A*C*a^6*b^8*c^6*d^6*f^2 - 3672
*A*C*a^9*b^5*c^5*d^7*f^2 + 3594*A*C*a^8*b^6*c^4*d^8*f^2 + 3236*A*C*a^8*b^6*
c^6*d^6*f^2 + 2816*A*C*a^5*b^9*c^9*d^3*f^2 + 2624*A*C*a^5*b^9*c^3*d^9*f^2 +
2432*A*C*a^7*b^7*c^7*d^5*f^2 - 2366*A*C*a^4*b^10*c^8*d^4*f^2 + 2298*A*C*a^
10*b^4*c^4*d^8*f^2 + 1872*A*C*a^7*b^7*c^3*d^9*f^2 + 1848*A*C*a^10*b^4*c^6*d

^6*f^2 - 1644*A*C*a^4*b^10*c^6*d^6*f^2 - 1488*A*C*a^9*b^5*c^7*d^5*f^2 - 140
8*A*C*a^9*b^5*c^3*d^9*f^2 - 1308*A*C*a^6*b^8*c^4*d^8*f^2 + 1248*A*C*a^7*b^7
*c^5*d^7*f^2 - 1012*A*C*a^6*b^8*c^10*d^2*f^2 + 1008*A*C*a^3*b^11*c^7*d^5*f^
2 + 992*A*C*a^3*b^11*c^5*d^7*f^2 + 928*A*C*a^3*b^11*c^3*d^9*f^2 + 848*A*C*a
^7*b^7*c^9*d^3*f^2 + 636*A*C*a^8*b^6*c^2*d^10*f^2 - 628*A*C*a^4*b^10*c^10*d
^2*f^2 - 600*A*C*a^6*b^8*c^2*d^10*f^2 - 576*A*C*a^11*b^3*c^5*d^7*f^2 + 572*
A*C*a^10*b^4*c^2*d^10*f^2 + 464*A*C*a^8*b^6*c^8*d^4*f^2 - 304*A*C*a^4*b^10*
c^4*d^8*f^2 + 304*A*C*a^2*b^12*c^6*d^6*f^2 + 296*A*C*a^2*b^12*c^4*d^8*f^2 +
260*A*C*a^10*b^4*c^8*d^4*f^2 - 232*A*C*a^12*b^2*c^2*d^10*f^2 - 232*A*C*a^9
*b^5*c^9*d^3*f^2 + 228*A*C*a^2*b^12*c^10*d^2*f^2 - 188*A*C*a^4*b^10*c^2*d^1
0*f^2 + 144*A*C*a^11*b^3*c^3*d^9*f^2 + 116*A*C*a^12*b^2*c^6*d^6*f^2 - 112*A
*C*a^11*b^3*c^7*d^5*f^2 + 112*A*C*a^3*b^11*c^9*d^3*f^2 + 92*A*C*a^8*b^6*c^1
0*d^2*f^2 + 74*A*C*a^12*b^2*c^4*d^8*f^2 + 62*A*C*a^2*b^12*c^8*d^4*f^2 + 40*
A*C*a^2*b^12*c^2*d^10*f^2 - 7008*A*B*a^7*b^7*c^6*d^6*f^2 - 4032*A*B*a^7*b^7
*c^4*d^8*f^2 + 3952*A*B*a^8*b^6*c^7*d^5*f^2 + 3648*A*B*a^8*b^6*c^5*d^7*f^2
- 3392*A*B*a^7*b^7*c^8*d^4*f^2 + 3264*A*B*a^6*b^8*c^7*d^5*f^2 - 2992*A*B*a^
4*b^10*c^5*d^7*f^2 - 2368*A*B*a^4*b^10*c^7*d^5*f^2 - 2304*A*B*a^4*b^10*c^3*
d^9*f^2 - 1968*A*B*a^9*b^5*c^6*d^6*f^2 - 1872*A*B*a^4*b^10*c^9*d^3*f^2 - 17
28*A*B*a^7*b^7*c^2*d^10*f^2 + 1712*A*B*a^3*b^11*c^8*d^4*f^2 - 1536*A*B*a^10
*b^4*c^3*d^9*f^2 + 1536*A*B*a^6*b^8*c^5*d^7*f^2 - 1392*A*B*a^2*b^12*c^5*d^7
*f^2 + 1328*A*B*a^3*b^11*c^6*d^6*f^2 - 1104*A*B*a^2*b^12*c^3*d^9*f^2 - 1056
*A*B*a^6*b^8*c^3*d^9*f^2 + 976*A*B*a^6*b^8*c^9*d^3*f^2 + 960*A*B*a^11*b^3*c
^4*d^8*f^2 + 936*A*B*a^5*b^9*c^8*d^4*f^2 - 912*A*B*a^10*b^4*c^5*d^7*f^2 + 8
48*A*B*a^8*b^6*c^9*d^3*f^2 + 816*A*B*a^3*b^11*c^4*d^8*f^2 - 816*A*B*a^2*b^1
2*c^7*d^5*f^2 + 768*A*B*a^3*b^11*c^10*d^2*f^2 + 672*A*B*a^8*b^6*c^3*d^9*f^2
- 632*A*B*a^9*b^5*c^8*d^4*f^2 - 608*A*B*a^9*b^5*c^2*d^10*f^2 - 552*A*B*a^9
*b^5*c^4*d^8*f^2 - 544*A*B*a^7*b^7*c^10*d^2*f^2 - 480*A*B*a^5*b^9*c^2*d^10*
f^2 + 464*A*B*a^5*b^9*c^10*d^2*f^2 - 464*A*B*a^2*b^12*c^9*d^3*f^2 + 432*A*B
*a^11*b^3*c^2*d^10*f^2 - 368*A*B*a^12*b^2*c^3*d^9*f^2 - 256*A*B*a^5*b^9*c^6
*d^6*f^2 - 208*A*B*a^12*b^2*c^5*d^7*f^2 + 176*A*B*a^5*b^9*c^4*d^8*f^2 + 112
*A*B*a^11*b^3*c^6*d^6*f^2 + 112*A*B*a^10*b^4*c^7*d^5*f^2 - 16*A*B*a^3*b^11*
c^2*d^10*f^2 - 576*B*C*a^8*b^6*c*d^11*f^2 + 400*B*C*a^4*b^10*c^11*d*f^2 - 2
88*B*C*a^6*b^8*c*d^11*f^2 - 176*B*C*a^6*b^8*c^11*d*f^2 + 128*B*C*a^10*b^4*c
*d^11*f^2 - 108*B*C*a*b^13*c^4*d^8*f^2 - 104*B*C*a^4*b^10*c*d^11*f^2 - 92*B
*C*a^13*b*c^4*d^8*f^2 - 60*B*C*a*b^13*c^8*d^4*f^2 - 60*B*C*a*b^13*c^6*d^6*f
^2 + 48*B*C*a^2*b^12*c^11*d*f^2 - 40*B*C*a*b^13*c^2*d^10*f^2 - 28*B*C*a^13*
b*c^2*d^10*f^2 - 24*B*C*a^12*b^2*c*d^11*f^2 + 20*B*C*a*b^13*c^10*d^2*f^2 -
16*B*C*a^2*b^12*c*d^11*f^2 + 12*B*C*a^13*b*c^6*d^6*f^2 + 912*A*C*a^7*b^7*c*
d^11*f^2 + 808*A*C*a^5*b^9*c*d^11*f^2 + 432*A*C*a^5*b^9*c^11*d*f^2 + 336*A*
C*a^3*b^11*c*d^11*f^2 + 224*A*C*a^11*b^3*c*d^11*f^2 - 112*A*C*a^3*b^11*c^11
*d*f^2 + 112*A*C*a*b^13*c^3*d^9*f^2 - 88*A*C*a*b^13*c^9*d^3*f^2 + 80*A*C*a^
13*b*c^3*d^9*f^2 + 56*A*C*a*b^13*c^5*d^7*f^2 + 48*A*C*a^9*b^5*c*d^11*f^2 -
40*A*C*a^13*b*c^5*d^7*f^2 - 16*A*C*a^7*b^7*c^11*d*f^2 + 16*A*C*a*b^13*c^7*d
^5*f^2 - 496*A*B*a^4*b^10*c*d^11*f^2 - 400*A*B*a^4*b^10*c^11*d*f^2 + 288*A*
B*a^8*b^6*c*d^11*f^2 - 288*A*B*a^6*b^8*c*d^11*f^2 - 272*A*B*a^2*b^12*c*d^11

*f^2 + 240*A*B*a*b^13*c^6*d^6*f^2 - 224*A*B*a^10*b^4*c*d^11*f^2 + 192*A*B*a
*b^13*c^8*d^4*f^2 + 192*A*B*a*b^13*c^4*d^8*f^2 + 176*A*B*a^6*b^8*c^11*d*f^2
 + 104*A*B*a^13*b*c^4*d^8*f^2 - 48*A*B*a^2*b^12*c^11*d*f^2 + 16*A*B*a^13*b*
c^2*d^10*f^2 + 16*A*B*a*b^13*c^10*d^2*f^2 + 16*A*B*a*b^13*c^2*d^10*f^2 - 96
*B*C*b^14*c^7*d^5*f^2 - 72*B*C*b^14*c^5*d^7*f^2 - 24*B*C*b^14*c^9*d^3*f^2 -
 16*B*C*b^14*c^3*d^9*f^2 + 116*A*C*b^14*c^6*d^6*f^2 + 100*A*C*b^14*c^4*d^8*
f^2 + 24*A*C*b^14*c^2*d^10*f^2 + 22*A*C*b^14*c^8*d^4*f^2 + 16*B*C*a^14*c^3*
d^9*f^2 + 8*A*C*b^14*c^10*d^2*f^2 - 192*A*B*b^14*c^5*d^7*f^2 - 176*A*B*b^14
*c^3*d^9*f^2 - 112*B*C*a^11*b^3*d^12*f^2 - 48*A*B*b^14*c^7*d^5*f^2 - 28*A*C
*a^14*c^2*d^10*f^2 + 4*B*C*a^5*b^9*d^12*f^2 + 2*A*C*a^14*c^4*d^8*f^2 + 150*
A*C*a^10*b^4*d^12*f^2 - 80*B*C*a^3*b^11*c^12*f^2 + 66*A*C*a^8*b^6*d^12*f^2
- 30*A*C*a^12*b^2*d^12*f^2 + 24*B*C*a^5*b^9*c^12*f^2 - 16*A*B*a^14*c^3*d^9*
f^2 - 12*A*C*a^4*b^10*d^12*f^2 - 576*A*B*a^7*b^7*d^12*f^2 - 432*A*B*a^9*b^5
*d^12*f^2 - 400*A*B*a^5*b^9*d^12*f^2 - 144*A*B*a^3*b^11*d^12*f^2 - 66*A*C*a
^4*b^10*c^12*f^2 + 54*A*C*a^2*b^12*c^12*f^2 - 32*A*B*a^11*b^3*d^12*f^2 + 2*
A*C*a^6*b^8*c^12*f^2 + 80*A*B*a^3*b^11*c^12*f^2 - 24*A*B*a^5*b^9*c^12*f^2 +
 2508*C^2*a^6*b^8*c^6*d^6*f^2 + 2376*C^2*a^9*b^5*c^5*d^7*f^2 + 2357*C^2*a^6
*b^8*c^8*d^4*f^2 - 2048*C^2*a^5*b^9*c^7*d^5*f^2 + 1304*C^2*a^9*b^5*c^3*d^9*
f^2 + 1303*C^2*a^4*b^10*c^8*d^4*f^2 + 1212*C^2*a^4*b^10*c^6*d^6*f^2 - 1203*
C^2*a^8*b^6*c^4*d^8*f^2 - 1192*C^2*a^5*b^9*c^9*d^3*f^2 + 1062*C^2*a^6*b^8*c
^4*d^8*f^2 + 984*C^2*a^9*b^5*c^7*d^5*f^2 - 952*C^2*a^8*b^6*c^6*d^6*f^2 + 76
8*C^2*a^7*b^7*c^5*d^7*f^2 - 681*C^2*a^10*b^4*c^4*d^8*f^2 - 672*C^2*a^5*b^9*
c^5*d^7*f^2 - 480*C^2*a^10*b^4*c^6*d^6*f^2 + 458*C^2*a^6*b^8*c^10*d^2*f^2 -
 448*C^2*a^7*b^7*c^7*d^5*f^2 + 422*C^2*a^4*b^10*c^4*d^8*f^2 + 372*C^2*a^6*b
^8*c^2*d^10*f^2 + 360*C^2*a^11*b^3*c^5*d^7*f^2 + 312*C^2*a^7*b^7*c^3*d^9*f^
2 + 278*C^2*a^4*b^10*c^10*d^2*f^2 - 232*C^2*a^7*b^7*c^9*d^3*f^2 + 194*C^2*a
^12*b^2*c^2*d^10*f^2 + 176*C^2*a^9*b^5*c^9*d^3*f^2 + 152*C^2*a^3*b^11*c^5*d
^7*f^2 + 124*C^2*a^4*b^10*c^2*d^10*f^2 - 120*C^2*a^3*b^11*c^7*d^5*f^2 - 114
*C^2*a^2*b^12*c^10*d^2*f^2 - 102*C^2*a^8*b^6*c^2*d^10*f^2 + 101*C^2*a^12*b^
2*c^4*d^8*f^2 + 100*C^2*a^2*b^12*c^6*d^6*f^2 - 88*C^2*a^5*b^9*c^3*d^9*f^2 +
 77*C^2*a^2*b^12*c^8*d^4*f^2 + 72*C^2*a^11*b^3*c^3*d^9*f^2 - 64*C^2*a^8*b^6
*c^10*d^2*f^2 + 64*C^2*a^3*b^11*c^3*d^9*f^2 - 58*C^2*a^10*b^4*c^2*d^10*f^2
+ 56*C^2*a^12*b^2*c^6*d^6*f^2 + 56*C^2*a^11*b^3*c^7*d^5*f^2 + 40*C^2*a^3*b^
11*c^9*d^3*f^2 + 36*C^2*a^12*b^2*c^8*d^4*f^2 + 32*C^2*a^2*b^12*c^4*d^8*f^2
+ 26*C^2*a^10*b^4*c^8*d^4*f^2 + 16*C^2*a^2*b^12*c^2*d^10*f^2 + 2*C^2*a^8*b^
6*c^8*d^4*f^2 + 2277*B^2*a^8*b^6*c^4*d^8*f^2 + 2144*B^2*a^5*b^9*c^7*d^5*f^2
 - 2112*B^2*a^9*b^5*c^5*d^7*f^2 + 2028*B^2*a^8*b^6*c^6*d^6*f^2 - 1671*B^2*a
^6*b^8*c^8*d^4*f^2 + 1275*B^2*a^10*b^4*c^4*d^8*f^2 + 1176*B^2*a^5*b^9*c^5*d
^7*f^2 + 1096*B^2*a^5*b^9*c^9*d^3*f^2 - 1044*B^2*a^6*b^8*c^6*d^6*f^2 + 984*
B^2*a^10*b^4*c^6*d^6*f^2 - 968*B^2*a^9*b^5*c^3*d^9*f^2 - 888*B^2*a^9*b^5*c^
7*d^5*f^2 + 672*B^2*a^7*b^7*c^7*d^5*f^2 + 664*B^2*a^5*b^9*c^3*d^9*f^2 - 649
*B^2*a^4*b^10*c^8*d^4*f^2 + 618*B^2*a^8*b^6*c^2*d^10*f^2 + 514*B^2*a^4*b^10
*c^4*d^8*f^2 + 460*B^2*a^2*b^12*c^6*d^6*f^2 + 422*B^2*a^8*b^6*c^8*d^4*f^2 +
 406*B^2*a^10*b^4*c^2*d^10*f^2 - 382*B^2*a^6*b^8*c^10*d^2*f^2 + 368*B^2*a^2
*b^12*c^4*d^8*f^2 - 312*B^2*a^11*b^3*c^5*d^7*f^2 + 312*B^2*a^7*b^7*c^3*d^9*

f^2 + 248*B^2*a^7*b^7*c^9*d^3*f^2 + 245*B^2*a^2*b^12*c^8*d^4*f^2 - 192*B^2*a^7*b^7*c^5*d^7*f^2 - 184*B^2*a^3*b^11*c^9*d^3*f^2 + 182*B^2*a^2*b^12*c^10*d^2*f^2 + 176*B^2*a^3*b^11*c^3*d^9*f^2 + 174*B^2*a^6*b^8*c^4*d^8*f^2 - 170*B^2*a^4*b^10*c^10*d^2*f^2 - 152*B^2*a^9*b^5*c^9*d^3*f^2 + 152*B^2*a^4*b^10*c^2*d^10*f^2 + 142*B^2*a^10*b^4*c^8*d^4*f^2 - 90*B^2*a^12*b^2*c^2*d^10*f^2 + 88*B^2*a^2*b^12*c^2*d^10*f^2 + 84*B^2*a^8*b^6*c^10*d^2*f^2 + 84*B^2*a^6*b^8*c^2*d^10*f^2 + 60*B^2*a^12*b^2*c^6*d^6*f^2 - 56*B^2*a^11*b^3*c^7*d^5*f^2 + 53*B^2*a^12*b^2*c^4*d^8*f^2 + 24*B^2*a^11*b^3*c^3*d^9*f^2 + 24*B^2*a^4*b^10*c^6*d^6*f^2 + 24*B^2*a^3*b^11*c^7*d^5*f^2 - 8*B^2*a^3*b^11*c^5*d^7*f^2 + 4566*A^2*a^6*b^8*c^4*d^8*f^2 + 4284*A^2*a^6*b^8*c^6*d^6*f^2 - 3776*A^2*a^5*b^9*c^7*d^5*f^2 - 3624*A^2*a^5*b^9*c^5*d^7*f^2 + 3122*A^2*a^4*b^10*c^4*d^8*f^2 + 3108*A^2*a^6*b^8*c^2*d^10*f^2 + 2741*A^2*a^6*b^8*c^8*d^4*f^2 + 2592*A^2*a^4*b^10*c^6*d^6*f^2 - 2536*A^2*a^5*b^9*c^3*d^9*f^2 + 2224*A^2*a^4*b^10*c^2*d^10*f^2 - 2184*A^2*a^7*b^7*c^3*d^9*f^2 - 2016*A^2*a^7*b^7*c^5*d^7*f^2 - 1984*A^2*a^7*b^7*c^7*d^5*f^2 + 1626*A^2*a^8*b^6*c^2*d^10*f^2 - 1624*A^2*a^5*b^9*c^9*d^3*f^2 + 1603*A^2*a^4*b^10*c^8*d^4*f^2 + 1296*A^2*a^9*b^5*c^5*d^7*f^2 - 1144*A^2*a^3*b^11*c^5*d^7*f^2 - 992*A^2*a^3*b^11*c^3*d^9*f^2 + 968*A^2*a^2*b^12*c^4*d^8*f^2 - 888*A^2*a^3*b^11*c^7*d^5*f^2 + 849*A^2*a^8*b^6*c^4*d^8*f^2 + 808*A^2*a^2*b^12*c^2*d^10*f^2 - 616*A^2*a^7*b^7*c^9*d^3*f^2 + 554*A^2*a^6*b^8*c^10*d^2*f^2 - 504*A^2*a^10*b^4*c^6*d^6*f^2 + 504*A^2*a^9*b^5*c^7*d^5*f^2 + 460*A^2*a^2*b^12*c^6*d^6*f^2 + 350*A^2*a^10*b^4*c^2*d^10*f^2 + 350*A^2*a^4*b^10*c^10*d^2*f^2 - 321*A^2*a^10*b^4*c^4*d^8*f^2 + 216*A^2*a^11*b^3*c^5*d^7*f^2 - 216*A^2*a^11*b^3*c^3*d^9*f^2 + 182*A^2*a^12*b^2*c^2*d^10*f^2 - 152*A^2*a^3*b^11*c^9*d^3*f^2 - 124*A^2*a^8*b^6*c^6*d^6*f^2 - 114*A^2*a^2*b^12*c^10*d^2*f^2 + 104*A^2*a^9*b^5*c^3*d^9*f^2 + 77*A^2*a^2*b^12*c^8*d^4*f^2 + 74*A^2*a^8*b^6*c^8*d^4*f^2 - 70*A^2*a^10*b^4*c^8*d^4*f^2 + 56*A^2*a^11*b^3*c^7*d^5*f^2 + 56*A^2*a^9*b^5*c^9*d^3*f^2 + 41*A^2*a^12*b^2*c^4*d^8*f^2 - 28*A^2*a^12*b^2*c^6*d^6*f^2 - 28*A^2*a^8*b^6*c^10*d^2*f^2 - 16*B*C*b^14*c^11*d*f^2 - 16*B*C*a^14*c*d^11*f^2 - 48*A*B*b^14*c*d^11*f^2 + 16*A*B*b^14*c^11*d*f^2 + 12*B*C*a^13*b*d^12*f^2 + 24*B*C*a*b^13*c^12*f^2 + 16*A*B*a^14*c*d^11*f^2 - 24*A*B*a^13*b*d^12*f^2 - 24*A*B*a*b^13*d^12*f^2 - 24*A*B*a*b^13*c^12*f^2 + 216*C^2*a^9*b^5*c*d^11*f^2 - 216*C^2*a^5*b^9*c^11*d*f^2 + 56*C^2*a^3*b^11*c^11*d*f^2 + 56*C^2*a*b^13*c^9*d^3*f^2 + 56*C^2*a*b^13*c^5*d^7*f^2 - 40*C^2*a^11*b^3*c*d^11*f^2 + 40*C^2*a*b^13*c^7*d^5*f^2 + 32*C^2*a^13*b*c^5*d^7*f^2 - 24*C^2*a^7*b^7*c*d^11*f^2 - 16*C^2*a^13*b*c^3*d^9*f^2 + 16*C^2*a*b^13*c^3*d^9*f^2 + 8*C^2*a^7*b^7*c^11*d*f^2 - 8*C^2*a^5*b^9*c*d^11*f^2 + 264*B^2*a^7*b^7*c*d^11*f^2 + 224*B^2*a^5*b^9*c*d^11*f^2 + 168*B^2*a^5*b^9*c^11*d*f^2 - 112*B^2*a*b^13*c^9*d^3*f^2 - 104*B^2*a^3*b^11*c^11*d*f^2 - 104*B^2*a*b^13*c^7*d^5*f^2 + 96*B^2*a^3*b^11*c*d^11*f^2 + 88*B^2*a^11*b^3*c*d^11*f^2 - 72*B^2*a^9*b^5*c*d^11*f^2 - 64*B^2*a*b^13*c^5*d^7*f^2 + 32*B^2*a^13*b*c^3*d^9*f^2 - 24*B^2*a^13*b*c^5*d^7*f^2 - 24*B^2*a^7*b^7*c^11*d*f^2 + 16*B^2*a*b^13*c^3*d^9*f^2 - 888*A^2*a^7*b^7*c*d^11*f^2 - 800*A^2*a^5*b^9*c*d^11*f^2 - 336*A^2*a^3*b^11*c*d^11*f^2 - 264*A^2*a^9*b^5*c*d^11*f^2 - 216*A^2*a^5*b^9*c^11*d*f^2 - 184*A^2*a^11*b^3*c*d^11*f^2 - 128*A^2*a*b^13*c^3*d^9*f^2 - 112*A^2*a*b^13*c^5*d^7*f^2 - 64*A^2*a^13*b*c^3*d^9*f

^2 + 56*A^2*a^3*b^11*c^11*d*f^2 - 56*A^2*a*b^13*c^7*d^5*f^2 + 32*A^2*a*b^13
*c^9*d^3*f^2 + 8*A^2*a^13*b*c^5*d^7*f^2 + 8*A^2*a^7*b^7*c^11*d*f^2 + 24*C^2
*a*b^13*c^11*d*f^2 - 16*C^2*a^13*b*c*d^11*f^2 - 40*B^2*a*b^13*c^11*d*f^2 +
24*B^2*a^13*b*c*d^11*f^2 + 16*B^2*a*b^13*c*d^11*f^2 - 48*A^2*a*b^13*c*d^11*
f^2 - 40*A^2*a^13*b*c*d^11*f^2 + 24*A^2*a*b^13*c^11*d*f^2 - 6*A*C*b^14*c^12
*f^2 + 2*A*C*a^14*d^12*f^2 + 31*C^2*b^14*c^8*d^4*f^2 + 20*C^2*b^14*c^6*d^6*
f^2 + 4*C^2*b^14*c^4*d^8*f^2 + 2*C^2*b^14*c^10*d^2*f^2 + 80*B^2*b^14*c^6*d^
6*f^2 + 64*B^2*b^14*c^4*d^8*f^2 + 31*B^2*b^14*c^8*d^4*f^2 + 16*B^2*b^14*c^2
*d^10*f^2 + 14*C^2*a^14*c^2*d^10*f^2 + 14*B^2*b^14*c^10*d^2*f^2 - C^2*a^14*
c^4*d^8*f^2 + 120*A^2*b^14*c^2*d^10*f^2 + 112*A^2*b^14*c^4*d^8*f^2 + 33*C^2
*a^12*b^2*d^12*f^2 - 27*C^2*a^10*b^4*d^12*f^2 - 17*A^2*b^14*c^8*d^4*f^2 - 1
0*B^2*a^14*c^2*d^10*f^2 - 10*A^2*b^14*c^10*d^2*f^2 + 8*A^2*b^14*c^6*d^6*f^2
 + 3*C^2*a^8*b^6*d^12*f^2 + 3*B^2*a^14*c^4*d^8*f^2 + 117*B^2*a^10*b^4*d^12*
f^2 + 111*B^2*a^8*b^6*d^12*f^2 + 72*B^2*a^6*b^8*d^12*f^2 + 33*C^2*a^4*b^10*
c^12*f^2 - 27*C^2*a^2*b^12*c^12*f^2 + 24*B^2*a^4*b^10*d^12*f^2 + 14*A^2*a^1
4*c^2*d^10*f^2 + 4*B^2*a^2*b^12*d^12*f^2 - 3*B^2*a^12*b^2*d^12*f^2 - C^2*a^
6*b^8*c^12*f^2 - A^2*a^14*c^4*d^8*f^2 + 720*A^2*a^6*b^8*d^12*f^2 + 552*A^2*
a^4*b^10*d^12*f^2 + 471*A^2*a^8*b^6*d^12*f^2 + 216*A^2*a^2*b^12*d^12*f^2 +
93*A^2*a^10*b^4*d^12*f^2 + 33*B^2*a^2*b^12*c^12*f^2 + 33*A^2*a^12*b^2*d^12*
f^2 - 27*B^2*a^4*b^10*c^12*f^2 + 3*B^2*a^6*b^8*c^12*f^2 + 33*A^2*a^4*b^10*c
^12*f^2 - 27*A^2*a^2*b^12*c^12*f^2 - A^2*a^6*b^8*c^12*f^2 + 3*C^2*b^14*c^12
*f^2 - C^2*a^14*d^12*f^2 + 36*A^2*b^14*d^12*f^2 + 3*B^2*a^14*d^12*f^2 - B^2
*b^14*c^12*f^2 + 3*A^2*b^14*c^12*f^2 - A^2*a^14*d^12*f^2 - 44*A*B*C*a^10*b*
c*d^9*f + 3816*A*B*C*a^4*b^7*c^5*d^5*f + 2920*A*B*C*a^5*b^6*c^2*d^8*f - 273
6*A*B*C*a^6*b^5*c^3*d^7*f - 2672*A*B*C*a^3*b^8*c^4*d^6*f + 1996*A*B*C*a^7*b
^4*c^4*d^6*f - 1412*A*B*C*a^5*b^6*c^6*d^4*f + 1120*A*B*C*a^2*b^9*c^3*d^7*f
+ 1080*A*B*C*a^7*b^4*c^2*d^8*f + 1040*A*B*C*a^2*b^9*c^5*d^5*f + 684*A*B*C*a
^5*b^6*c^4*d^6*f + 592*A*B*C*a^4*b^7*c^3*d^7*f - 560*A*B*C*a^2*b^9*c^7*d^3*
f - 448*A*B*C*a^3*b^8*c^2*d^8*f - 400*A*B*C*a^8*b^3*c^5*d^5*f - 398*A*B*C*a
^9*b^2*c^2*d^8*f - 312*A*B*C*a^3*b^8*c^6*d^4*f + 166*A*B*C*a^3*b^8*c^8*d^2*
f + 136*A*B*C*a^6*b^5*c^5*d^5*f + 128*A*B*C*a^6*b^5*c^7*d^3*f - 100*A*B*C*a
^7*b^4*c^6*d^4*f - 64*A*B*C*a^9*b^2*c^4*d^6*f + 64*A*B*C*a^4*b^7*c^7*d^3*f
- 32*A*B*C*a^8*b^3*c^3*d^7*f - 16*A*B*C*a^5*b^6*c^8*d^2*f - 1312*A*B*C*a^4*
b^7*c*d^9*f + 996*A*B*C*a^8*b^3*c*d^9*f + 728*A*B*C*a*b^10*c^6*d^4*f - 624*
A*B*C*a^6*b^5*c*d^9*f - 584*A*B*C*a*b^10*c^2*d^8*f - 512*A*B*C*a*b^10*c^4*d
^6*f - 320*A*B*C*a^2*b^9*c*d^9*f - 98*A*B*C*a*b^10*c^8*d^2*f + 36*A*B*C*a^2
*b^9*c^9*d*f + 32*A*B*C*a^10*b*c^3*d^7*f - 16*A*B*C*a^4*b^7*c^9*d*f + 46*B*
C^2*a^10*b*c*d^9*f - 16*B^2*C*a*b^10*c*d^9*f - 2*B^2*C*a*b^10*c^9*d*f + 312
*A^2*C*a*b^10*c*d^9*f - 48*A*C^2*a*b^10*c*d^9*f - 6*A^2*C*a*b^10*c^9*d*f +
6*A*C^2*a*b^10*c^9*d*f + 208*A*B^2*a*b^10*c*d^9*f - 2*A^2*B*a^10*b*c*d^9*f
+ 2*A*B^2*a*b^10*c^9*d*f - 224*A*B*C*b^11*c^5*d^5*f + 80*A*B*C*b^11*c^7*d^3
*f - 32*A*B*C*b^11*c^3*d^7*f + 2*A*B*C*a^11*c^2*d^8*f - 480*A*B*C*a^7*b^4*d
^10*f + 78*A*B*C*a^9*b^2*d^10*f - 64*A*B*C*a^5*b^6*d^10*f + 2*A*B*C*a^3*b^8
*c^10*f - 1692*B*C^2*a^4*b^7*c^5*d^5*f - 1500*B^2*C*a^5*b^6*c^5*d^5*f - 146
4*B^2*C*a^5*b^6*c^3*d^7*f + 1426*B*C^2*a^5*b^6*c^6*d^4*f - 1158*B^2*C*a^4*b

^7*c^6*d^4*f + 1152*B*C^2*a^6*b^5*c^3*d^7*f + 1026*B^2*C*a^6*b^5*c^4*d^6*f
- 974*B*C^2*a^7*b^4*c^4*d^6*f + 960*B^2*C*a^3*b^8*c^5*d^5*f - 884*B*C^2*a^5
*b^6*c^2*d^8*f - 764*B^2*C*a^7*b^4*c^5*d^5*f + 752*B^2*C*a^4*b^7*c^2*d^8*f
- 752*B*C^2*a^4*b^7*c^3*d^7*f + 738*B^2*C*a^4*b^7*c^4*d^6*f - 688*B^2*C*a^2
*b^9*c^6*d^4*f - 675*B^2*C*a^8*b^3*c^2*d^8*f + 560*B*C^2*a^8*b^3*c^5*d^5*f
+ 496*B*C^2*a^3*b^8*c^4*d^6*f + 496*B*C^2*a^2*b^9*c^7*d^3*f - 468*B*C^2*a^7
*b^4*c^2*d^8*f + 456*B^2*C*a^3*b^8*c^7*d^3*f - 452*B^2*C*a^8*b^3*c^4*d^6*f
- 416*B*C^2*a^2*b^9*c^3*d^7*f + 378*B*C^2*a^5*b^6*c^4*d^6*f + 376*B*C^2*a^8
*b^3*c^3*d^7*f - 360*B^2*C*a^6*b^5*c^2*d^8*f + 355*B*C^2*a^9*b^2*c^2*d^8*f
+ 346*B^2*C*a^6*b^5*c^6*d^4*f - 320*B^2*C*a^2*b^9*c^4*d^6*f + 268*B^2*C*a^2
*b^9*c^2*d^8*f + 216*B^2*C*a^7*b^4*c^3*d^7*f - 203*B*C^2*a^3*b^8*c^8*d^2*f
- 184*B*C^2*a^6*b^5*c^7*d^3*f + 170*B*C^2*a^7*b^4*c^6*d^4*f + 160*B^2*C*a^5
*b^6*c^7*d^3*f - 160*B*C^2*a^2*b^9*c^5*d^5*f - 140*B^2*C*a^4*b^7*c^8*d^2*f
- 136*B*C^2*a^3*b^8*c^2*d^8*f + 112*B^2*C*a^9*b^2*c^3*d^7*f + 91*B^2*C*a^2*
b^9*c^8*d^2*f + 88*B*C^2*a^4*b^7*c^7*d^3*f + 72*B^2*C*a^8*b^3*c^6*d^4*f - 6
4*B^2*C*a^3*b^8*c^3*d^7*f - 60*B*C^2*a^3*b^8*c^6*d^4*f + 56*B*C^2*a^9*b^2*c
^4*d^6*f + 52*B*C^2*a^6*b^5*c^5*d^5*f + 48*B^2*C*a^9*b^2*c^5*d^5*f - 48*B^2
*C*a^7*b^4*c^7*d^3*f + 44*B*C^2*a^5*b^6*c^8*d^2*f - 36*B*C^2*a^9*b^2*c^6*d^
4*f + 12*B^2*C*a^6*b^5*c^8*d^2*f - 2958*A^2*C*a^4*b^7*c^4*d^6*f - 1932*A^2*
C*a^4*b^7*c^2*d^8*f + 1848*A^2*C*a^5*b^6*c^3*d^7*f + 1728*A^2*C*a^3*b^8*c^3
*d^7*f + 1524*A^2*C*a^5*b^6*c^5*d^5*f + 1374*A*C^2*a^4*b^7*c^4*d^6*f - 1272
*A*C^2*a^5*b^6*c^3*d^7*f - 1236*A*C^2*a^5*b^6*c^5*d^5*f + 1116*A*C^2*a^4*b^
7*c^2*d^8*f - 1110*A^2*C*a^6*b^5*c^4*d^6*f + 1038*A*C^2*a^6*b^5*c^4*d^6*f -
768*A^2*C*a^2*b^9*c^2*d^8*f - 696*A^2*C*a^7*b^4*c^3*d^7*f - 666*A*C^2*a^4*
b^7*c^6*d^4*f + 564*A^2*C*a^6*b^5*c^2*d^8*f - 564*A*C^2*a^7*b^4*c^5*d^5*f -
555*A*C^2*a^8*b^3*c^2*d^8*f + 519*A^2*C*a^8*b^3*c^2*d^8*f - 480*A*C^2*a^3*
b^8*c^3*d^7*f + 456*A*C^2*a^3*b^8*c^5*d^5*f - 420*A*C^2*a^2*b^9*c^6*d^4*f +
408*A*C^2*a^7*b^4*c^3*d^7*f + 408*A*C^2*a^2*b^9*c^2*d^8*f + 348*A^2*C*a^2*
b^9*c^6*d^4*f - 348*A*C^2*a^6*b^5*c^2*d^8*f + 342*A*C^2*a^6*b^5*c^6*d^4*f -
336*A*C^2*a^8*b^3*c^4*d^6*f + 324*A^2*C*a^7*b^4*c^5*d^5*f - 312*A^2*C*a^2*
b^9*c^4*d^6*f + 264*A^2*C*a^8*b^3*c^4*d^6*f + 240*A*C^2*a^5*b^6*c^7*d^3*f +
195*A*C^2*a^2*b^9*c^8*d^2*f - 174*A^2*C*a^6*b^5*c^6*d^4*f + 144*A*C^2*a^9*
b^2*c^3*d^7*f - 123*A^2*C*a^2*b^9*c^8*d^2*f + 120*A*C^2*a^3*b^8*c^7*d^3*f +
108*A*C^2*a^8*b^3*c^6*d^4*f - 102*A^2*C*a^4*b^7*c^6*d^4*f - 96*A^2*C*a^4*b
^7*c^8*d^2*f + 72*A^2*C*a^3*b^8*c^7*d^3*f + 72*A*C^2*a^9*b^2*c^5*d^5*f - 48
*A^2*C*a^9*b^2*c^3*d^7*f + 48*A^2*C*a^5*b^6*c^7*d^3*f - 48*A*C^2*a^2*b^9*c^
4*d^6*f - 24*A^2*C*a^3*b^8*c^5*d^5*f - 12*A*C^2*a^4*b^7*c^8*d^2*f + 2736*A^
2*B*a^6*b^5*c^3*d^7*f + 2464*A^2*B*a^3*b^8*c^4*d^6*f - 2298*A*B^2*a^4*b^7*c
^4*d^6*f - 2252*A^2*B*a^5*b^6*c^2*d^8*f - 1692*A^2*B*a^4*b^7*c^5*d^5*f - 15
92*A*B^2*a^4*b^7*c^2*d^8*f - 1338*A*B^2*a^6*b^5*c^4*d^6*f + 1320*A*B^2*a^5*
b^6*c^3*d^7*f + 1212*A*B^2*a^5*b^6*c^5*d^5*f - 1056*A*B^2*a^3*b^8*c^5*d^5*f
+ 1024*A^2*B*a^4*b^7*c^3*d^7*f - 1022*A^2*B*a^7*b^4*c^4*d^6*f - 880*A^2*B*
a^2*b^9*c^5*d^5*f - 846*A^2*B*a^5*b^6*c^4*d^6*f - 840*A*B^2*a^7*b^4*c^3*d^7
*f + 760*A*B^2*a^2*b^9*c^6*d^4*f - 704*A^2*B*a^2*b^9*c^3*d^7*f + 688*A*B^2*
a^3*b^8*c^3*d^7*f + 660*A^2*B*a^3*b^8*c^6*d^4*f - 612*A^2*B*a^7*b^4*c^2*d^8

*f + 462*A*B^2*a^4*b^7*c^6*d^4*f + 459*A*B^2*a^8*b^3*c^2*d^8*f - 412*A*B^2*
a^2*b^9*c^2*d^8*f - 408*A*B^2*a^3*b^8*c^7*d^3*f + 388*A^2*B*a^6*b^5*c^5*d^5
*f + 296*A^2*B*a^3*b^8*c^2*d^8*f + 288*A*B^2*a^6*b^5*c^2*d^8*f + 284*A*B^2*
a^7*b^4*c^5*d^5*f + 236*A*B^2*a^8*b^3*c^4*d^6*f - 226*A*B^2*a^6*b^5*c^6*d^4
*f + 212*A*B^2*a^2*b^9*c^4*d^6*f + 202*A^2*B*a^5*b^6*c^6*d^4*f - 152*A^2*B*
a^4*b^7*c^7*d^3*f + 88*A^2*B*a^8*b^3*c^3*d^7*f + 79*A^2*B*a^9*b^2*c^2*d^8*f
 - 70*A^2*B*a^7*b^4*c^6*d^4*f + 68*A*B^2*a^4*b^7*c^8*d^2*f + 64*A^2*B*a^2*b
^9*c^7*d^3*f - 64*A*B^2*a^9*b^2*c^3*d^7*f + 56*A^2*B*a^8*b^3*c^5*d^5*f + 56
*A^2*B*a^6*b^5*c^7*d^3*f + 37*A^2*B*a^3*b^8*c^8*d^2*f - 28*A^2*B*a^9*b^2*c^
4*d^6*f - 28*A^2*B*a^5*b^6*c^8*d^2*f + 17*A*B^2*a^2*b^9*c^8*d^2*f - 16*A*B^
2*a^5*b^6*c^7*d^3*f + 48*A*B*C*b^11*c*d^9*f + 4*A*B*C*b^11*c^9*d*f + 24*A*B
*C*a*b^10*d^10*f - 6*A*B*C*a*b^10*c^10*f + 432*B^2*C*a^7*b^4*c*d^9*f - 376*
B*C^2*a*b^10*c^6*d^4*f - 354*B*C^2*a^8*b^3*c*d^9*f + 352*B^2*C*a*b^10*c^5*d
^5*f + 320*B^2*C*a^5*b^6*c*d^9*f + 256*B^2*C*a*b^10*c^3*d^7*f - 232*B^2*C*a
*b^10*c^7*d^3*f - 210*B^2*C*a^9*b^2*c*d^9*f - 152*B*C^2*a*b^10*c^4*d^6*f +
85*B*C^2*a*b^10*c^8*d^2*f + 72*B^2*C*a^3*b^8*c*d^9*f - 48*B*C^2*a^6*b^5*c*d
^9*f - 40*B*C^2*a^10*b*c^3*d^7*f + 40*B*C^2*a*b^10*c^2*d^8*f + 37*B^2*C*a^1
0*b*c^2*d^8*f + 22*B^2*C*a^3*b^8*c^9*d*f - 18*B*C^2*a^2*b^9*c^9*d*f + 16*B*
C^2*a^2*b^9*c*d^9*f - 12*B^2*C*a^10*b*c^4*d^6*f + 8*B*C^2*a^4*b^7*c^9*d*f +
 8*B*C^2*a^4*b^7*c*d^9*f - 984*A^2*C*a^7*b^4*c*d^9*f + 672*A^2*C*a^3*b^8*c*
d^9*f + 552*A*C^2*a^7*b^4*c*d^9*f - 504*A^2*C*a*b^10*c^5*d^5*f - 408*A^2*C*
a^5*b^6*c*d^9*f + 408*A*C^2*a^5*b^6*c*d^9*f + 336*A*C^2*a*b^10*c^5*d^5*f -
216*A*C^2*a*b^10*c^7*d^3*f + 192*A*C^2*a*b^10*c^3*d^7*f - 162*A*C^2*a^9*b^2
*c*d^9*f + 120*A^2*C*a*b^10*c^7*d^3*f + 96*A^2*C*a*b^10*c^3*d^7*f + 90*A^2*
C*a^9*b^2*c*d^9*f + 66*A^2*C*a^3*b^8*c^9*d*f - 66*A*C^2*a^3*b^8*c^9*d*f + 5
7*A*C^2*a^10*b*c^2*d^8*f - 48*A*C^2*a^3*b^8*c*d^9*f - 9*A^2*C*a^10*b*c^2*d^
8*f + 1736*A^2*B*a^4*b^7*c*d^9*f + 1248*A^2*B*a^6*b^5*c*d^9*f - 1008*A*B^2*
a^7*b^4*c*d^9*f + 772*A^2*B*a*b^10*c^4*d^6*f - 688*A*B^2*a*b^10*c^5*d^5*f -
 608*A*B^2*a^5*b^6*c*d^9*f + 436*A^2*B*a*b^10*c^2*d^8*f - 426*A^2*B*a^8*b^3
*c*d^9*f + 312*A*B^2*a^3*b^8*c*d^9*f + 304*A^2*B*a^2*b^9*c*d^9*f - 244*A^2*
B*a*b^10*c^6*d^4*f - 160*A*B^2*a*b^10*c^3*d^7*f + 114*A*B^2*a^9*b^2*c*d^9*f
 + 88*A*B^2*a*b^10*c^7*d^3*f - 22*A*B^2*a^3*b^8*c^9*d*f - 18*A^2*B*a^2*b^9*
c^9*d*f + 13*A^2*B*a*b^10*c^8*d^2*f - 13*A*B^2*a^10*b*c^2*d^8*f + 8*A^2*B*a
^10*b*c^3*d^7*f + 8*A^2*B*a^4*b^7*c^9*d*f + 112*B^2*C*b^11*c^6*d^4*f - 64*B
*C^2*b^11*c^7*d^3*f + 16*B^2*C*b^11*c^4*d^6*f - 16*B^2*C*b^11*c^2*d^8*f + 1
6*B*C^2*b^11*c^5*d^5*f + 16*B*C^2*b^11*c^3*d^7*f - B^2*C*b^11*c^8*d^2*f + 9
6*A^2*C*b^11*c^4*d^6*f - 84*A^2*C*b^11*c^6*d^4*f + 72*A*C^2*b^11*c^6*d^4*f
- 24*A*C^2*b^11*c^4*d^6*f - 24*A*C^2*b^11*c^2*d^8*f - 21*A*C^2*b^11*c^8*d^2
*f + 12*A^2*C*b^11*c^2*d^8*f + 9*A^2*C*b^11*c^8*d^2*f - B*C^2*a^11*c^2*d^8*
f + 176*A*B^2*b^11*c^4*d^6*f + 136*A^2*B*b^11*c^5*d^5*f - 128*A^2*B*b^11*c^
3*d^7*f + 112*A*B^2*b^11*c^2*d^8*f + 111*B^2*C*a^8*b^3*d^10*f - 64*A*B^2*b^
11*c^6*d^4*f - 39*B*C^2*a^9*b^2*d^10*f + 24*B*C^2*a^7*b^4*d^10*f - 16*A^2*B
*b^11*c^7*d^3*f - 4*B^2*C*a^2*b^9*d^10*f - 4*B*C^2*a^5*b^6*d^10*f + 432*A^2
*C*a^6*b^5*d^10*f + 192*A^2*C*a^4*b^7*d^10*f - 111*A^2*C*a^8*b^3*d^10*f + 1
11*A*C^2*a^8*b^3*d^10*f - 72*A*C^2*a^6*b^5*d^10*f + 12*A*C^2*a^4*b^7*d^10*f

- 3*B^2*C*a^2*b^9*c^10*f - A^2*B*a^11*c^2*d^8*f - B*C^2*a^3*b^8*c^10*f + 4
56*A^2*B*a^7*b^4*d^10*f - 288*A^2*B*a^3*b^8*d^10*f + 252*A*B^2*a^6*b^5*d^10
*f + 192*A*B^2*a^4*b^7*d^10*f - 183*A*B^2*a^8*b^3*d^10*f - 148*A^2*B*a^5*b^
6*d^10*f + 76*A*B^2*a^2*b^9*d^10*f - 9*A^2*C*a^2*b^9*c^10*f + 9*A*C^2*a^2*b
^9*c^10*f - 3*A^2*B*a^9*b^2*d^10*f + 3*A*B^2*a^2*b^9*c^10*f - A^2*B*a^3*b^8
*c^10*f - 2*C^3*a*b^10*c^9*d*f - 2*B^3*a^10*b*c*d^9*f - 264*A^3*a*b^10*c*d^
9*f + 2*A^3*a*b^10*c^9*d*f - 2*B*C^2*b^11*c^9*d*f - 2*B^2*C*a^11*c*d^9*f -
120*A^2*B*b^11*c*d^9*f - 9*B^2*C*a^10*b*d^10*f - 6*A^2*C*a^11*c*d^9*f + 6*A
*C^2*a^11*c*d^9*f - 2*A^2*B*b^11*c^9*d*f + 9*A^2*C*a^10*b*d^10*f - 9*A*C^2*
a^10*b*d^10*f + 3*B*C^2*a*b^10*c^10*f + 2*A*B^2*a^11*c*d^9*f - 132*A^2*B*a*
b^10*d^10*f - 3*A*B^2*a^10*b*d^10*f + 3*A^2*B*a*b^10*c^10*f + 520*C^3*a^5*b
^6*c^3*d^7*f + 460*C^3*a^5*b^6*c^5*d^5*f - 418*C^3*a^6*b^5*c^4*d^6*f + 406*
C^3*a^4*b^7*c^6*d^4*f + 268*C^3*a^7*b^4*c^5*d^5*f - 266*C^3*a^6*b^5*c^6*d^4
*f + 233*C^3*a^8*b^3*c^2*d^8*f - 176*C^3*a^5*b^6*c^7*d^3*f + 164*C^3*a^2*b^
9*c^6*d^4*f + 140*C^3*a^6*b^5*c^2*d^8*f + 136*C^3*a^2*b^9*c^4*d^6*f - 128*C
^3*a^9*b^2*c^3*d^7*f + 128*C^3*a^3*b^8*c^3*d^7*f - 108*C^3*a^8*b^3*c^6*d^4*
f - 104*C^3*a^3*b^8*c^7*d^3*f - 104*C^3*a^3*b^8*c^5*d^5*f + 100*C^3*a^8*b^3
*c^4*d^6*f - 89*C^3*a^2*b^9*c^8*d^2*f - 72*C^3*a^9*b^2*c^5*d^5*f - 40*C^3*a
^7*b^4*c^3*d^7*f + 40*C^3*a^4*b^7*c^8*d^2*f - 28*C^3*a^4*b^7*c^2*d^8*f - 16
*C^3*a^2*b^9*c^2*d^8*f - 2*C^3*a^4*b^7*c^4*d^6*f + 828*B^3*a^4*b^7*c^5*d^5*
f + 408*B^3*a^5*b^6*c^2*d^8*f + 390*B^3*a^7*b^4*c^4*d^6*f - 372*B^3*a^3*b^8
*c^4*d^6*f - 336*B^3*a^6*b^5*c^3*d^7*f - 314*B^3*a^5*b^6*c^6*d^4*f + 288*B^
3*a^4*b^7*c^3*d^7*f + 216*B^3*a^7*b^4*c^2*d^8*f - 176*B^3*a^2*b^9*c^7*d^3*f
 + 128*B^3*a^2*b^9*c^3*d^7*f + 108*B^3*a^6*b^5*c^5*d^5*f + 88*B^3*a^4*b^7*c
^7*d^3*f + 72*B^3*a^2*b^9*c^5*d^5*f - 68*B^3*a^3*b^8*c^2*d^8*f - 65*B^3*a^9
*b^2*c^2*d^8*f - 56*B^3*a^8*b^3*c^5*d^5*f + 40*B^3*a^6*b^5*c^7*d^3*f + 37*B
^3*a^3*b^8*c^8*d^2*f + 30*B^3*a^5*b^6*c^4*d^6*f - 28*B^3*a^5*b^6*c^8*d^2*f
+ 24*B^3*a^8*b^3*c^3*d^7*f - 4*B^3*a^9*b^2*c^4*d^6*f - 2*B^3*a^7*b^4*c^6*d^
4*f + 1586*A^3*a^4*b^7*c^4*d^6*f - 1376*A^3*a^3*b^8*c^3*d^7*f - 1096*A^3*a^
5*b^6*c^3*d^7*f + 844*A^3*a^4*b^7*c^2*d^8*f - 748*A^3*a^5*b^6*c^5*d^5*f + 4
90*A^3*a^6*b^5*c^4*d^6*f + 376*A^3*a^2*b^9*c^2*d^8*f + 362*A^3*a^4*b^7*c^6*
d^4*f - 356*A^3*a^6*b^5*c^2*d^8*f + 328*A^3*a^7*b^4*c^3*d^7*f - 328*A^3*a^3
*b^8*c^5*d^5*f + 224*A^3*a^2*b^9*c^4*d^6*f - 197*A^3*a^8*b^3*c^2*d^8*f - 11
2*A^3*a^5*b^6*c^7*d^3*f + 98*A^3*a^6*b^5*c^6*d^4*f - 92*A^3*a^2*b^9*c^6*d^4
*f - 88*A^3*a^3*b^8*c^7*d^3*f + 68*A^3*a^4*b^7*c^8*d^2*f + 32*A^3*a^9*b^2*c
^3*d^7*f - 28*A^3*a^8*b^3*c^4*d^6*f - 28*A^3*a^7*b^4*c^5*d^5*f + 17*A^3*a^2
*b^9*c^8*d^2*f + 104*C^3*a*b^10*c^7*d^3*f + 54*C^3*a^9*b^2*c*d^9*f - 40*C^3
*a^7*b^4*c*d^9*f - 35*C^3*a^10*b*c^2*d^8*f + 22*C^3*a^3*b^8*c^9*d*f + 16*C^
3*a*b^10*c^5*d^5*f - 16*C^3*a*b^10*c^3*d^7*f + 8*C^3*a^5*b^6*c*d^9*f - 2*A*
B*C*a^11*d^10*f + 198*B^3*a^8*b^3*c*d^9*f + 192*B^3*a*b^10*c^6*d^4*f - 128*
B^3*a^4*b^7*c*d^9*f - 80*B^3*a*b^10*c^2*d^8*f - 56*B^3*a^2*b^9*c*d^9*f - 24
*B^3*a^6*b^5*c*d^9*f - 18*B^3*a^2*b^9*c^9*d*f - 16*B^3*a*b^10*c^4*d^6*f + 1
3*B^3*a*b^10*c^8*d^2*f + 8*B^3*a^10*b*c^3*d^7*f + 8*B^3*a^4*b^7*c^9*d*f - 6
24*A^3*a^3*b^8*c*d^9*f + 472*A^3*a^7*b^4*c*d^9*f - 272*A^3*a*b^10*c^3*d^7*f
 + 152*A^3*a*b^10*c^5*d^5*f - 22*A^3*a^3*b^8*c^9*d*f + 18*A^3*a^9*b^2*c*d^9

*f - 13*A^3*a^10*b*c^2*d^8*f - 8*A^3*a^5*b^6*c*d^9*f - 8*A^3*a*b^10*c^7*d^3
*f + A*B^2*b^11*c^8*d^2*f + 11*C^3*b^11*c^8*d^2*f - 8*C^3*b^11*c^6*d^4*f -
4*C^3*b^11*c^4*d^6*f - 64*B^3*b^11*c^5*d^5*f - 32*B^3*b^11*c^3*d^7*f - 68*A
^3*b^11*c^4*d^6*f + 20*A^3*b^11*c^6*d^4*f + 12*A^3*b^11*c^2*d^8*f - C^3*a^8
*b^3*d^10*f - B^3*a^11*c^2*d^8*f - 60*B^3*a^7*b^4*d^10*f - 32*B^3*a^5*b^6*d
^10*f + 21*B^3*a^9*b^2*d^10*f - 12*B^3*a^3*b^8*d^10*f - 3*C^3*a^2*b^9*c^10*
f - 360*A^3*a^6*b^5*d^10*f - 204*A^3*a^4*b^7*d^10*f - B^3*a^3*b^8*c^10*f +
3*A^3*a^2*b^9*c^10*f - 2*C^3*a^11*c*d^9*f - 2*B^3*b^11*c^9*d*f + 3*C^3*a^10
*b*d^10*f + 2*A^3*a^11*c*d^9*f + 3*B^3*a*b^10*c^10*f - 3*A^3*a^10*b*d^10*f
- 36*A^2*C*b^11*d^10*f + 3*A^2*C*b^11*c^10*f - 3*A*C^2*b^11*c^10*f - A*B^2*
b^11*c^10*f + 36*A^3*b^11*d^10*f - A^3*b^11*c^10*f + A^3*b^11*c^8*d^2*f + A
^3*a^8*b^3*d^10*f + B^2*C*b^11*c^10*f + B*C^2*a^11*d^10*f + A^2*B*a^11*d^10
*f + C^3*b^11*c^10*f + B^3*a^11*d^10*f - 6*A*B^2*C*a^7*b*c*d^7 + 4*A*B^2*C*
a*b^7*c*d^7 + 168*A^2*B*C*a^2*b^6*c^3*d^5 + 144*A*B*C^2*a^3*b^5*c^4*d^4 - 1
29*A^2*B*C*a^3*b^5*c^4*d^4 - 96*A*B*C^2*a^2*b^6*c^3*d^5 + 84*A*B*C^2*a^3*b^
5*c^2*d^6 + 72*A^2*B*C*a^4*b^4*c^3*d^5 - 72*A^2*B*C*a^3*b^5*c^2*d^6 + 64*A*
B^2*C*a^4*b^4*c^4*d^4 - 60*A*B*C^2*a^4*b^4*c^3*d^5 + 57*A^2*B*C*a^5*b^3*c^2
*d^6 - 56*A*B^2*C*a^5*b^3*c^3*d^5 - 39*A*B^2*C*a^2*b^6*c^4*d^4 - 38*A*B^2*C
*a^3*b^5*c^5*d^3 + 36*A*B^2*C*a^3*b^5*c^3*d^5 + 36*A*B*C^2*a^5*b^3*c^4*d^4
- 30*A*B*C^2*a^5*b^3*c^2*d^6 + 27*A*B^2*C*a^6*b^2*c^2*d^6 - 24*A*B^2*C*a^2*
b^6*c^2*d^6 + 24*A*B*C^2*a^6*b^2*c^3*d^5 - 24*A*B*C^2*a^4*b^4*c^5*d^3 - 18*
A^2*B*C*a^5*b^3*c^4*d^4 + 18*A^2*B*C*a^2*b^6*c^5*d^3 - 15*A*B^2*C*a^4*b^4*c
^2*d^6 - 12*A^2*B*C*a^6*b^2*c^3*d^5 + 12*A^2*B*C*a^4*b^4*c^5*d^3 + 9*A*B^2*
C*a^2*b^6*c^6*d^2 + 6*A*B*C^2*a^3*b^5*c^6*d^2 - 3*A^2*B*C*a^3*b^5*c^6*d^2 +
60*A^2*B*C*a^2*b^6*c*d^7 - 51*A^2*B*C*a*b^7*c^4*d^4 + 48*A*B*C^2*a^6*b^2*c
*d^7 - 42*A^2*B*C*a^6*b^2*c*d^7 - 42*A^2*B*C*a*b^7*c^2*d^6 + 36*A*B*C^2*a^4
*b^4*c*d^7 + 36*A*B*C^2*a*b^7*c^4*d^4 + 36*A*B*C^2*a*b^7*c^2*d^6 - 30*A^2*B
*C*a^4*b^4*c*d^7 + 24*A*B^2*C*a^3*b^5*c*d^7 - 24*A*B*C^2*a^2*b^6*c*d^7 + 18
*A*B^2*C*a*b^7*c^5*d^3 - 18*A*B*C^2*a*b^7*c^6*d^2 + 12*A*B^2*C*a*b^7*c^3*d^
5 + 9*A^2*B*C*a*b^7*c^6*d^2 + 6*A*B^2*C*a^5*b^3*c*d^7 - 6*A*B*C^2*a^7*b*c^2
*d^6 + 3*A^2*B*C*a^7*b*c^2*d^6 - 18*B^3*C*a^6*b^2*c*d^7 - 18*B*C^3*a^6*b^2*
c*d^7 - 14*B^3*C*a^4*b^4*c*d^7 - 14*B*C^3*a^4*b^4*c*d^7 - 10*B^3*C*a*b^7*c^
2*d^6 - 10*B*C^3*a*b^7*c^2*d^6 + 9*B^3*C*a*b^7*c^6*d^2 + 9*B*C^3*a*b^7*c^6*
d^2 - 7*B^3*C*a*b^7*c^4*d^4 - 7*B*C^3*a*b^7*c^4*d^4 + 6*B^2*C^2*a^7*b*c*d^7
- 4*B^3*C*a^2*b^6*c*d^7 + 4*B^2*C^2*a*b^7*c*d^7 - 4*B*C^3*a^2*b^6*c*d^7 +
3*B^3*C*a^7*b*c^2*d^6 + 3*B*C^3*a^7*b*c^2*d^6 + 144*A^3*C*a^3*b^5*c*d^7 + 6
2*A^3*C*a^5*b^3*c*d^7 + 48*A*C^3*a^3*b^5*c*d^7 - 36*A^2*C^2*a*b^7*c*d^7 + 2
6*A*C^3*a^5*b^3*c*d^7 + 20*A^3*C*a*b^7*c^3*d^5 + 18*A^2*C^2*a^7*b*c*d^7 - 1
8*A*C^3*a*b^7*c^5*d^3 - 6*A^3*C*a*b^7*c^5*d^3 - 4*A*C^3*a*b^7*c^3*d^5 - 32*
A^3*B*a^2*b^6*c*d^7 - 32*A*B^3*a^2*b^6*c*d^7 + 22*A^3*B*a*b^7*c^4*d^4 + 22*
A*B^3*a*b^7*c^4*d^4 + 16*A^3*B*a*b^7*c^2*d^6 + 16*A*B^3*a*b^7*c^2*d^6 + 12*
A^3*B*a^6*b^2*c*d^7 + 12*A*B^3*a^6*b^2*c*d^7 + 8*A^3*B*a^4*b^4*c*d^7 - 8*A^
2*B^2*a*b^7*c*d^7 + 8*A*B^3*a^4*b^4*c*d^7 + 36*A^2*B*C*b^8*c^3*d^5 + 24*A*B
*C^2*b^8*c^5*d^3 - 18*A^2*B*C*b^8*c^5*d^3 - 12*A*B*C^2*b^8*c^3*d^5 - 3*A*B^
2*C*b^8*c^6*d^2 - 3*A*B^2*C*b^8*c^4*d^4 - 2*A*B^2*C*b^8*c^2*d^6 + 57*A^2*B*

C*a^5*b^3*d^8 + 36*A^2*B*C*a^3*b^5*d^8 - 30*A*B*C^2*a^5*b^3*d^8 - 18*A*B*C^
2*a^3*b^5*d^8 - 9*A*B^2*C*a^4*b^4*d^8 - 3*A*B^2*C*a^6*b^2*d^8 - 2*A*B^2*C*a
^2*b^6*d^8 + 34*B^2*C^2*a^3*b^5*c^5*d^3 + 28*B^2*C^2*a^5*b^3*c^3*d^5 + 24*B
^2*C^2*a^2*b^6*c^4*d^4 - 20*B^2*C^2*a^4*b^4*c^4*d^4 + 12*B^2*C^2*a^3*b^5*c^
3*d^5 + 12*B^2*C^2*a^2*b^6*c^2*d^6 + 9*B^2*C^2*a^6*b^2*c^4*d^4 + 9*B^2*C^2*
a^4*b^4*c^2*d^6 - 9*B^2*C^2*a^2*b^6*c^6*d^2 - 3*B^2*C^2*a^6*b^2*c^2*d^6 + 1
59*A^2*C^2*a^4*b^4*c^2*d^6 - 156*A^2*C^2*a^3*b^5*c^3*d^5 + 90*A^2*C^2*a^3*b
^5*c^5*d^3 + 78*A^2*C^2*a^2*b^6*c^2*d^6 - 63*A^2*C^2*a^4*b^4*c^4*d^4 - 27*A
^2*C^2*a^6*b^2*c^2*d^6 - 27*A^2*C^2*a^2*b^6*c^6*d^2 - 18*A^2*C^2*a^2*b^6*c^
4*d^4 + 9*A^2*C^2*a^6*b^2*c^4*d^4 + 66*A^2*B^2*a^2*b^6*c^2*d^6 + 60*A^2*B^2
*a^4*b^4*c^2*d^6 - 48*A^2*B^2*a^3*b^5*c^3*d^5 + 42*A^2*B^2*a^2*b^6*c^4*d^4
+ 28*A^2*B^2*a^5*b^3*c^3*d^5 - 17*A^2*B^2*a^4*b^4*c^4*d^4 - 6*A^2*B^2*a^6*b
^2*c^2*d^6 + 4*A^2*B^2*a^3*b^5*c^5*d^3 + 36*A^3*C*a*b^7*c*d^7 - 18*A*C^3*a^
7*b*c*d^7 + 12*A*C^3*a*b^7*c*d^7 - 6*A^3*C*a^7*b*c*d^7 + 24*A^2*B*C*b^8*c*d
^7 - 12*A*B*C^2*b^8*c*d^7 + 12*A^2*B*C*a*b^7*d^8 + 6*A*B*C^2*a^7*b*d^8 - 6*
A*B*C^2*a*b^7*d^8 - 3*A^2*B*C*a^7*b*d^8 - 53*B^3*C*a^3*b^5*c^4*d^4 - 53*B*C
^3*a^3*b^5*c^4*d^4 - 32*B^3*C*a^3*b^5*c^2*d^6 - 32*B*C^3*a^3*b^5*c^2*d^6 -
18*B^3*C*a^5*b^3*c^4*d^4 - 18*B*C^3*a^5*b^3*c^4*d^4 + 16*B^3*C*a^4*b^4*c^3*
d^5 + 16*B*C^3*a^4*b^4*c^3*d^5 - 12*B^3*C*a^6*b^2*c^3*d^5 + 12*B^3*C*a^4*b^
4*c^5*d^3 + 12*B^2*C^2*a^3*b^5*c*d^7 - 12*B*C^3*a^6*b^2*c^3*d^5 + 12*B*C^3*
a^4*b^4*c^5*d^3 + 8*B^3*C*a^2*b^6*c^3*d^5 + 8*B*C^3*a^2*b^6*c^3*d^5 - 6*B^3
*C*a^2*b^6*c^5*d^3 + 6*B^2*C^2*a^5*b^3*c*d^7 - 6*B^2*C^2*a*b^7*c^5*d^3 - 6*
B*C^3*a^2*b^6*c^5*d^3 - 3*B^3*C*a^3*b^5*c^6*d^2 - 3*B*C^3*a^3*b^5*c^6*d^2 -
175*A^3*C*a^4*b^4*c^2*d^6 + 164*A^3*C*a^3*b^5*c^3*d^5 - 144*A^2*C^2*a^3*b^
5*c*d^7 - 124*A^3*C*a^2*b^6*c^2*d^6 - 90*A*C^3*a^3*b^5*c^5*d^3 - 73*A*C^3*a
^4*b^4*c^2*d^6 - 66*A^2*C^2*a^5*b^3*c*d^7 + 44*A*C^3*a^3*b^5*c^3*d^5 + 36*A
*C^3*a^4*b^4*c^4*d^4 + 30*A^3*C*a^4*b^4*c^4*d^4 - 30*A^3*C*a^3*b^5*c^5*d^3
+ 27*A*C^3*a^2*b^6*c^6*d^2 + 21*A*C^3*a^2*b^6*c^4*d^4 + 18*A^2*C^2*a*b^7*c^
5*d^3 - 18*A*C^3*a^6*b^2*c^4*d^4 - 16*A*C^3*a^2*b^6*c^2*d^6 + 15*A^3*C*a^6*
b^2*c^2*d^6 - 15*A^3*C*a^2*b^6*c^4*d^4 - 12*A^2*C^2*a*b^7*c^3*d^5 + 9*A^3*C
*a^2*b^6*c^6*d^2 + 9*A*C^3*a^6*b^2*c^2*d^6 - 80*A^3*B*a^2*b^6*c^3*d^5 - 80*
A*B^3*a^2*b^6*c^3*d^5 + 38*A^3*B*a^3*b^5*c^4*d^4 + 38*A*B^3*a^3*b^5*c^4*d^4
- 36*A^2*B^2*a^3*b^5*c*d^7 - 28*A^3*B*a^5*b^3*c^2*d^6 - 28*A^3*B*a^4*b^4*c
^3*d^5 - 28*A*B^3*a^5*b^3*c^2*d^6 - 28*A*B^3*a^4*b^4*c^3*d^5 + 20*A^3*B*a^3
*b^5*c^2*d^6 + 20*A*B^3*a^3*b^5*c^2*d^6 - 12*A^3*B*a^2*b^6*c^5*d^3 - 12*A^2
*B^2*a^5*b^3*c*d^7 - 12*A^2*B^2*a*b^7*c^5*d^3 - 12*A^2*B^2*a*b^7*c^3*d^5 -
12*A*B^3*a^2*b^6*c^5*d^3 + 9*B^2*C^2*b^8*c^4*d^4 + 4*B^2*C^2*b^8*c^2*d^6 +
3*B^2*C^2*b^8*c^6*d^2 - 30*A^2*C^2*b^8*c^4*d^4 + 9*A^2*C^2*b^8*c^6*d^2 + 16
*A^2*B^2*b^8*c^2*d^6 + 6*B^2*C^2*a^6*b^2*d^8 + 3*B^2*C^2*a^4*b^4*d^8 + 3*A^
2*B^2*b^8*c^4*d^4 + 36*A^2*C^2*a^4*b^4*d^8 + 27*A^2*C^2*a^2*b^6*d^8 - 18*A^
2*C^2*a^6*b^2*d^8 + 33*A^2*B^2*a^4*b^4*d^8 + 28*A^2*B^2*a^2*b^6*d^8 + 6*A^2
*B^2*a^6*b^2*d^8 + 6*C^4*a*b^7*c^5*d^3 + 4*C^4*a*b^7*c^3*d^5 - 2*C^4*a^5*b^
3*c*d^7 + 12*B^4*a^3*b^5*c*d^7 - 12*B^4*a*b^7*c^5*d^3 + 8*B^4*a^5*b^3*c*d^7
- 4*B^4*a*b^7*c^3*d^5 - 48*A^4*a^3*b^5*c*d^7 - 20*A^4*a^5*b^3*c*d^7 - 8*A^
4*a*b^7*c^3*d^5 - 10*B^3*C*b^8*c^5*d^3 - 10*B*C^3*b^8*c^5*d^3 - 4*B^3*C*b^8

*c^3*d^5 - 4*B*C^3*b^8*c^3*d^5 + 23*A^3*C*b^8*c^4*d^4 - 18*A^3*C*b^8*c^2*d^
6 + 11*A*C^3*b^8*c^4*d^4 - 9*A*C^3*b^8*c^6*d^2 + 6*A*C^3*b^8*c^2*d^6 - 3*A^
3*C*b^8*c^6*d^2 - 20*A^3*B*b^8*c^3*d^5 - 20*A*B^3*b^8*c^3*d^5 + 4*A^3*B*b^8
*c^5*d^3 + 4*A*B^3*b^8*c^5*d^3 - 63*A^3*C*a^4*b^4*d^8 - 54*A^3*C*a^2*b^6*d^
8 + 9*A^3*C*a^6*b^2*d^8 + 9*A*C^3*a^6*b^2*d^8 - 3*A*C^3*a^4*b^4*d^8 - 28*A^
3*B*a^5*b^3*d^8 - 28*A*B^3*a^5*b^3*d^8 - 18*A^3*B*a^3*b^5*d^8 - 18*A*B^3*a^
3*b^5*d^8 + B^3*C*a^5*b^3*c^2*d^6 + B*C^3*a^5*b^3*c^2*d^6 + 6*C^4*a^7*b*c*d
^7 + 4*B^4*a*b^7*c*d^7 - 12*A^4*a*b^7*c*d^7 - 12*A^3*B*b^8*c*d^7 - 12*A*B^3
*b^8*c*d^7 - 3*B^3*C*a^7*b*d^8 - 3*B*C^3*a^7*b*d^8 - 6*A^3*B*a*b^7*d^8 - 6*
A*B^3*a*b^7*d^8 + 30*C^4*a^3*b^5*c^5*d^3 + 19*C^4*a^4*b^4*c^2*d^6 + 9*C^4*a
^6*b^2*c^4*d^4 - 9*C^4*a^2*b^6*c^6*d^2 + 4*C^4*a^3*b^5*c^3*d^5 + 4*C^4*a^2*
b^6*c^2*d^6 + 3*C^4*a^6*b^2*c^2*d^6 - 3*C^4*a^4*b^4*c^4*d^4 - 3*C^4*a^2*b^6
*c^4*d^4 + 28*B^4*a^5*b^3*c^3*d^5 + 27*B^4*a^2*b^6*c^4*d^4 - 17*B^4*a^4*b^4
*c^4*d^4 - 10*B^4*a^4*b^4*c^2*d^6 + 8*B^4*a^3*b^5*c^3*d^5 + 8*B^4*a^2*b^6*c
^2*d^6 - 6*B^4*a^6*b^2*c^2*d^6 + 4*B^4*a^3*b^5*c^5*d^3 + 70*A^4*a^4*b^4*c^2
*d^6 + 58*A^4*a^2*b^6*c^2*d^6 - 56*A^4*a^3*b^5*c^3*d^5 + 15*A^4*a^2*b^6*c^4
*d^4 + B^2*C^2*a^2*b^6*d^8 - 18*A^3*C*b^8*d^8 + B^3*C*a^5*b^3*d^8 + B*C^3*a
^5*b^3*d^8 + 3*C^4*b^8*c^6*d^2 + 8*B^4*b^8*c^4*d^4 + 4*B^4*b^8*c^2*d^6 + 12
*A^4*b^8*c^2*d^6 - 5*A^4*b^8*c^4*d^4 + 6*B^4*a^6*b^2*d^8 + 3*B^4*a^4*b^4*d^
8 + 30*A^4*a^4*b^4*d^8 + 27*A^4*a^2*b^6*d^8 + 9*A^2*C^2*b^8*d^8 + 9*A^2*B^2
*b^8*d^8 + 9*A^4*b^8*d^8 + C^4*b^8*c^4*d^4 + B^4*a^2*b^6*d^8, f, k), k, 1,
4) - ((2*A*a^6*d^4 - A*b^6*c^4 - B*a*b^5*c^4 - 2*B*a^6*c*d^3 - 5*A*a^2*b^4*
c^4 + 2*A*a^2*b^4*d^4 + 4*A*a^4*b^2*d^4 + 3*B*a^3*b^3*c^4 + 3*C*a^2*b^4*c^4
 - C*a^4*b^2*c^4 - A*b^6*c^2*d^2 + 2*C*a^6*c^2*d^2 + 9*A*a^3*b^3*c*d^3 + 9*
A*a^3*b^3*c^3*d - B*a*b^5*c^2*d^2 - 5*B*a^2*b^4*c*d^3 - 3*B*a^2*b^4*c^3*d -
 11*B*a^4*b^2*c*d^3 - 7*B*a^4*b^2*c^3*d + C*a^3*b^3*c*d^3 + C*a^3*b^3*c^3*d
 - 5*A*a^2*b^4*c^2*d^2 + 3*B*a^3*b^3*c^2*d^2 + 5*C*a^2*b^4*c^2*d^2 + 3*C*a^
4*b^2*c^2*d^2 + 5*A*a*b^5*c*d^3 + 5*A*a*b^5*c^3*d + 5*C*a^5*b*c*d^3 + 5*C*a
^5*b*c^3*d)/(2*(a^3*d^3 - b^3*c^3 + 3*a*b^2*c^2*d - 3*a^2*b*c*d^2)*(a^4*c^2
 + a^4*d^2 + b^4*c^2 + b^4*d^2 + 2*a^2*b^2*c^2 + 2*a^2*b^2*d^2)) + (tan(e +
 f*x)*(9*A*a*b^5*d^4 - 4*A*a*b^5*c^4 - 2*B*b^6*c^4 + 4*A*a^5*b*d^4 + 4*C*a*
b^5*c^4 + 3*A*b^6*c*d^3 + 3*A*b^6*c^3*d + 5*C*a^5*b*d^4 + 17*A*a^3*b^3*d^4
+ 2*B*a^2*b^4*c^4 - 3*B*a^2*b^4*d^4 - 7*B*a^4*b^2*d^4 + C*a^3*b^3*d^4 - 2*B
*b^6*c^2*d^2 + A*a*b^5*c^2*d^2 + 3*A*a^2*b^4*c*d^3 + 3*A*a^2*b^4*c^3*d - 11
*B*a^3*b^3*c*d^3 - 3*B*a^3*b^3*c^3*d + 8*C*a*b^5*c^2*d^2 + 3*C*a^2*b^4*c*d^
3 + 3*C*a^2*b^4*c^3*d + 3*C*a^4*b^2*c*d^3 + 3*C*a^4*b^2*c^3*d + 9*C*a^5*b*c
^2*d^2 + 9*A*a^3*b^3*c^2*d^2 - B*a^2*b^4*c^2*d^2 - 7*B*a^4*b^2*c^2*d^2 + 9*
C*a^3*b^3*c^2*d^2 - 7*B*a*b^5*c*d^3 - 3*B*a*b^5*c^3*d - 4*B*a^5*b*c*d^3))/(
2*(a^3*d^3 - b^3*c^3 + 3*a*b^2*c^2*d - 3*a^2*b*c*d^2)*(a^4*c^2 + a^4*d^2 +
b^4*c^2 + b^4*d^2 + 2*a^2*b^2*c^2 + 2*a^2*b^2*d^2)) + (tan(e + f*x)^2*(3*A*
b^6*d^4 - B*a*b^5*d^4 - 2*B*b^6*c*d^3 - B*b^6*c^3*d + 6*A*a^2*b^4*d^4 + A*a
^4*b^2*d^4 - 3*B*a^3*b^3*d^4 + 2*A*b^6*c^2*d^2 + 2*C*a^4*b^2*d^4 + C*b^6*c^
2*d^2 - B*a*b^5*c^2*d^2 - B*a^2*b^4*c*d^3 + B*a^2*b^4*c^3*d - B*a^4*b^2*c*d
^3 + 4*A*a^2*b^4*c^2*d^2 - 3*B*a^3*b^3*c^2*d^2 + 2*C*a^2*b^4*c^2*d^2 + 3*C*
a^4*b^2*c^2*d^2 - 2*A*a*b^5*c*d^3 - 2*A*a*b^5*c^3*d + 2*C*a*b^5*c*d^3 + 2*C

```
*a*b^5*c^3*d))/((a^3*d^3 - b^3*c^3 + 3*a*b^2*c^2*d - 3*a^2*b*c*d^2)*(a^4*c^
2 + a^4*d^2 + b^4*c^2 + b^4*d^2 + 2*a^2*b^2*c^2 + 2*a^2*b^2*d^2)))/(tan(e +
 f*x)*(a^2*d + 2*a*b*c) + a^2*c + tan(e + f*x)^2*(b^2*c + 2*a*b*d) + b^2*d*
tan(e + f*x)^3))/f
```

3.84 $\int \frac{(a+b\tan(e+fx))^3(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^3} dx$

Optimal result 957
Rubi [A] (verified) 958
Mathematica [C] (verified) 962
Maple [A] (verified) 962
Fricas [B] (verification not implemented) 963
Sympy [F(-2)] 965
Maxima [A] (verification not implemented) 965
Giac [B] (verification not implemented) 966
Mupad [B] (verification not implemented) 967

Optimal result

Integrand size = 45, antiderivative size = 804

$$\int \frac{(a+b\tan(e+fx))^3(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^3} dx =$$

$$-\frac{(3ab^2(Ac^3 - c^3C + 3Bc^2d - 3Acd^2 + 3cCd^2 - Bd^3) + a^3(c^3C - 3Bc^2d - 3cCd^2 + Bd^3 - A(c^3 - 3cd^2}{(c^2+d^2)^3}$$

$$-\frac{(3a^2b(Ac^3 - c^3C + 3Bc^2d - 3Acd^2 + 3cCd^2 - Bd^3) - b^3(Ac^3 - c^3C + 3Bc^2d - 3Acd^2 + 3cCd^2 - Bd^3}{(c^2}$$

$$-\frac{(bc-ad)\left(b^2(3c^6C - Bc^5d + 9c^4Cd^2 - 3Bc^3d^3 - c^2(A-10C)d^4 - 6Bcd^5 + 3Ad^6) + a^2d^3((A-C)d(3\right.}{d^4(c^2+d^2)}$$

$$+\frac{b^2(b(3c^4C - Bc^3d + 6c^2Cd^2 - 3Bcd^3 + (2A+C)d^4) + ad^2(2c(A-C)d - B(c^2-d^2)))\tan(e+fx)}{d^3(c^2+d^2)^2 f}$$

$$-\frac{(c^2C - Bcd + Ad^2)(a+b\tan(e+fx))^3}{2d(c^2+d^2)f(c+d\tan(e+fx))^2}$$

$$-\frac{(b(3c^4C - Bc^3d - c^2(A-7C)d^2 - 5Bcd^3 + 3Ad^4) + 2ad^2(2c(A-C)d - B(c^2-d^2)))(a+b\tan(e+}{2d^2(c^2+d^2)^2 f(c+d\tan(e+fx))}$$

```
[Out] -(3*a*b^2*(A*c^3-3*A*c*d^2+3*B*c^2*d-B*d^3-C*c^3+3*C*c*d^2)+a^3*(c^3*C-3*B*
c^2*d-3*C*c*d^2+B*d^3-A*(c^3-3*c*d^2))-3*a^2*b*((A-C)*d*(3*c^2-d^2)-B*(c^3-
3*c*d^2))+b^3*((A-C)*d*(3*c^2-d^2)-B*(c^3-3*c*d^2)))*x/(c^2+d^2)^3-(3*a^2*b
*(A*c^3-3*A*c*d^2+3*B*c^2*d-B*d^3-C*c^3+3*C*c*d^2)-b^3*(A*c^3-3*A*c*d^2+3*B
*c^2*d-B*d^3-C*c^3+3*C*c*d^2)-a^3*((A-C)*d*(3*c^2-d^2)-B*(c^3-3*c*d^2))+3*a
*b^2*((A-C)*d*(3*c^2-d^2)-B*(c^3-3*c*d^2)))*ln(cos(f*x+e))/(c^2+d^2)^3/f-(-
a*d+b*c)*(b^2*(3*c^6*C-B*c^5*d+9*c^4*C*d^2-3*B*c^3*d^3-c^2*(A-10*C)*d^4-6*B
*c*d^5+3*A*d^6)+a^2*d^3*((A-C)*d*(3*c^2-d^2)-B*(c^3-3*c*d^2))+a*b*d^2*(8*c*
(A-C)*d^3-B*(c^4+6*c^2*d^2-3*d^4)))*ln(c+d*tan(f*x+e))/d^4/(c^2+d^2)^3/f+b^
2*(b*(3*c^4*C-B*c^3*d+6*C*c^2*d^2-3*B*c*d^3+(2*A+C)*d^4)+a*d^2*(2*c*(A-C)*d
```

```
-B*(c^2-d^2)))*tan(f*x+e)/d^3/(c^2+d^2)^2/f-1/2*(A*d^2-B*c*d+C*c^2)*(a+b*ta
n(f*x+e))^3/d/(c^2+d^2)/f/(c+d*tan(f*x+e))^2-1/2*(b*(3*c^4*C-B*c^3*d-c^2*(A
-7*C)*d^2-5*B*c*d^3+3*A*d^4)+2*a*d^2*(2*c*(A-C)*d-B*(c^2-d^2)))*(a+b*tan(f*
x+e))^2/d^2/(c^2+d^2)^2/f/(c+d*tan(f*x+e))
```

Rubi [A] (verified)

Time = 2.97 (sec) , antiderivative size = 804, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}} = 0.133$, Rules used = {3726, 3718, 3707, 3698, 31, 3556}

$$\int \frac{(a+b\tan(e+fx))^3 (A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^3}\,dx$$
$$= -\frac{(Cc^2-Bdc+Ad^2)(a+b\tan(e+fx))^3}{2d(c^2+d^2)f(c+d\tan(e+fx))^2}$$
$$-\frac{(2a(2c(A-C)d-B(c^2-d^2))d^2+b(3Cc^4-Bdc^3-(A-7C)d^2c^2-5Bd^3c+3Ad^4))(a+b\tan(e+f$$
$$2d^2(c^2+d^2)^2 f(c+d\tan(e+fx))$$
$$-\frac{((Cc^3-3Bdc^2-3Cd^2c+Bd^3-A(c^3-3cd^2))a^3-3b((A-C)d(3c^2-d^2)-B(c^3-3cd^2))a^2+3b^2(A}{(c^2+d^2)^3}$$
$$-\frac{(-(((A-C)d(3c^2-d^2)-B(c^3-3cd^2))a^3)+3b(Ac^3-Cc^3+3Bdc^2-3Ad^2c+3Cd^2c-Bd^3)a^2+3b}{(c^2}$$
$$-\frac{(bc-ad)(a^2((A-C)d(3c^2-d^2)-B(c^3-3cd^2))d^3+ab(8c(A-C)d^3-B(c^4+6d^2c^2-3d^4))d^2+b^2(}{d^4(c^2+d^2)^3}$$
$$+\frac{b^2(a(2c(A-C)d-B(c^2-d^2))d^2+b(3Cc^4-Bdc^3+6Cd^2c^2-3Bd^3c+(2A+C)d^4))\tan(e+fx)}{d^3(c^2+d^2)^2 f}$$

```
[In] Int[((a + b*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c + d
*Tan[e + f*x])^3,x]
```

```
[Out] -(((3*a*b^2*(A*c^3 - c^3*C + 3*B*c^2*d - 3*A*c*d^2 + 3*c*C*d^2 - B*d^3) + a
^3*(c^3*C - 3*B*c^2*d - 3*c*C*d^2 + B*d^3 - A*(c^3 - 3*c*d^2)) - 3*a^2*b*((
A - C)*d*(3*c^2 - d^2) - B*(c^3 - 3*c*d^2)) + b^3*((A - C)*d*(3*c^2 - d^2)
- B*(c^3 - 3*c*d^2)))*x)/(c^2 + d^2)^3) - ((3*a^2*b*(A*c^3 - c^3*C + 3*B*c^
2*d - 3*A*c*d^2 + 3*c*C*d^2 - B*d^3) - b^3*(A*c^3 - c^3*C + 3*B*c^2*d - 3*A
*c*d^2 + 3*c*C*d^2 - B*d^3) - a^3*((A - C)*d*(3*c^2 - d^2) - B*(c^3 - 3*c*d
^2)) + 3*a*b^2*((A - C)*d*(3*c^2 - d^2) - B*(c^3 - 3*c*d^2)))*Log[Cos[e + f
*x]])/((c^2 + d^2)^3*f) - ((b*c - a*d)*(b^2*(3*c^6*C - B*c^5*d + 9*c^4*C*d^
2 - 3*B*c^3*d^3 - c^2*(A - 10*C)*d^4 - 6*B*c*d^5 + 3*A*d^6) + a^2*d^3*((A -
 C)*d*(3*c^2 - d^2) - B*(c^3 - 3*c*d^2)) + a*b*d^2*(8*c*(A - C)*d^3 - B*(c^
4 + 6*c^2*d^2 - 3*d^4)))*Log[c + d*Tan[e + f*x]])/(d^4*(c^2 + d^2)^3*f) + (
b^2*(b*(3*c^4*C - B*c^3*d + 6*c^2*C*d^2 - 3*B*c*d^3 + (2*A + C)*d^4) + a*d^
2*(2*c*(A - C)*d - B*(c^2 - d^2)))*Tan[e + f*x])/(d^3*(c^2 + d^2)^2*f) - ((
c^2*C - B*c*d + A*d^2)*(a + b*Tan[e + f*x])^3)/(2*d*(c^2 + d^2)*f*(c + d*Ta
```

```
n[e + f*x])^2) - ((b*(3*c^4*C - B*c^3*d - c^2*(A - 7*C)*d^2 - 5*B*c*d^3 + 3
*A*d^4) + 2*a*d^2*(2*c*(A - C)*d - B*(c^2 - d^2)))*(a + b*Tan[e + f*x])^2)/
(2*d^2*(c^2 + d^2)^2*f*(c + d*Tan[e + f*x]))
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3698

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3707

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[(a*A + b*B - a
*C)*(x/(a^2 + b^2)), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3726

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*d^2 + c*(c*C - B*d))*(a + b*Tan[e +
 f*x])^m*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dis
```

```
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rubi steps

$$\text{integral} = -\frac{(c^2C - Bcd + Ad^2)(a + b\tan(e + fx))^3}{2d(c^2 + d^2)f(c + d\tan(e + fx))^2} + \frac{\int \frac{(a+b\tan(e+fx))^2(Ad(2ac+3bd)+(3bc-2ad)(cC-Bd)+2d((A-C)(bc-ad)+B(ac+bd))\tan(e+fx)+b(3c^2C-Bcd+(A+2C)d^2)\tan^2(e+fx)}{(c+d\tan(e+fx))^2}}{2d(c^2 + d^2)}$$

$$= -\frac{(c^2C - Bcd + Ad^2)(a + b\tan(e + fx))^3}{2d(c^2 + d^2)f(c + d\tan(e + fx))^2} - \frac{(b(3c^4C - Bc^3d - c^2(A - 7C)d^2 - 5Bcd^3 + 3Ad^4) + 2ad^2(2c(A - C)d - B(c^2 - d^2)))(a + b\tan}{2d^2(c^2 + d^2)^2 f(c + d\tan(e + fx))} + \frac{\int \frac{(a+b\tan(e+fx))(d(ac+2bd)(Ad(2ac+3bd)+(3bc-2ad)(cC-Bd))-(2bc-ad)(2ad^2(Bc-(A-C)d)-b(3c^3C-Bc^2d-c(A-4C)d^2-}{}}{}$$

$$= \frac{b^2(b(3c^4C - Bc^3d + 6c^2Cd^2 - 3Bcd^3 + (2A + C)d^4) + ad^2(2c(A - C)d - B(c^2 - d^2)))\tan(e + fx}{d^3(c^2 + d^2)^2 f} - \frac{(c^2C - Bcd + Ad^2)(a + b\tan(e + fx))^3}{2d(c^2 + d^2)f(c + d\tan(e + fx))^2} - \frac{(b(3c^4C - Bc^3d - c^2(A - 7C)d^2 - 5Bcd^3 + 3Ad^4) + 2ad^2(2c(A - C)d - B(c^2 - d^2)))(a + b\tan}{2d^2(c^2 + d^2)^2 f(c + d\tan(e + fx))} - \frac{\int \frac{-2(3ab^2d(c^4C-c^2(A-3C)d^2-2Bcd^3+Ad^4)-b^3c(3c^4C-Bc^3d+6c^2Cd^2-3Bcd^3+(2A+C)d^4)-a^3d^3(c^2C-2Bcd-Cd^2-A(c^2-d}{}}{}$$

$=$

$$-\frac{(3ab^2(Ac^3-c^3C+3Bc^2d-3Acd^2+3cCd^2-Bd^3)+a^3(c^3C-3Bc^2d-3cCd^2+Bd^3-A(c^3}{(c^2}$$

$$+\frac{b^2(b(3c^4C-Bc^3d+6c^2Cd^2-3Bcd^3+(2A+C)d^4)+ad^2(2c(A-C)d-B(c^2-d^2)))\tan(e+}{d^3\left(c^2+d^2\right)^2 f}$$

$$-\frac{\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^3}{2d\left(c^2+d^2\right)f(c+d\tan(e+fx))^2}$$

$$-\frac{(b(3c^4C-Bc^3d-c^2(A-7C)d^2-5Bcd^3+3Ad^4)+2ad^2(2c(A-C)d-B(c^2-d^2)))(a+b\,\mathrm{ta}}{2d^2\left(c^2+d^2\right)^2 f(c+d\tan(e+fx))}$$

$$+\frac{(3a^2b(Ac^3-c^3C+3Bc^2d-3Acd^2+3cCd^2-Bd^3)-b^3(Ac^3-c^3C+3Bc^2d-3Acd^2+3cCd}{}$$

$$-\frac{((bc-ad)\left(b^2(3c^6C-Bc^5d+9c^4Cd^2-3Bc^3d^3-c^2(A-10C)d^4-6Bcd^5+3Ad^6)+a^2d^3((A\right.}{d^3\left(c^2\right.}$$

$=$

$$-\frac{(3ab^2(Ac^3-c^3C+3Bc^2d-3Acd^2+3cCd^2-Bd^3)+a^3(c^3C-3Bc^2d-3cCd^2+Bd^3-A(c^3}{(c^2}$$

$$-\frac{(3a^2b(Ac^3-c^3C+3Bc^2d-3Acd^2+3cCd^2-Bd^3)-b^3(Ac^3-c^3C+3Bc^2d-3Acd^2+3cCd}{}$$

$$+\frac{b^2(b(3c^4C-Bc^3d+6c^2Cd^2-3Bcd^3+(2A+C)d^4)+ad^2(2c(A-C)d-B(c^2-d^2)))\tan(e+}{d^3\left(c^2+d^2\right)^2 f}$$

$$-\frac{\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^3}{2d\left(c^2+d^2\right)f(c+d\tan(e+fx))^2}$$

$$-\frac{(b(3c^4C-Bc^3d-c^2(A-7C)d^2-5Bcd^3+3Ad^4)+2ad^2(2c(A-C)d-B(c^2-d^2)))(a+b\,\mathrm{ta}}{2d^2\left(c^2+d^2\right)^2 f(c+d\tan(e+fx))}$$

$$-\frac{((bc-ad)\left(b^2(3c^6C-Bc^5d+9c^4Cd^2-3Bc^3d^3-c^2(A-10C)d^4-6Bcd^5+3Ad^6)+a^2d^3((A\right.}{}$$

$=$

$$-\frac{(3ab^2(Ac^3-c^3C+3Bc^2d-3Acd^2+3cCd^2-Bd^3)+a^3(c^3C-3Bc^2d-3cCd^2+Bd^3-A(c^3}{(c^2}$$

$$-\frac{(3a^2b(Ac^3-c^3C+3Bc^2d-3Acd^2+3cCd^2-Bd^3)-b^3(Ac^3-c^3C+3Bc^2d-3Acd^2+3cCd}{}$$

$$-\frac{(bc-ad)\left(b^2(3c^6C-Bc^5d+9c^4Cd^2-3Bc^3d^3-c^2(A-10C)d^4-6Bcd^5+3Ad^6)+a^2d^3((A-\right.}{d^4\left(\right.}$$

$$+\frac{b^2(b(3c^4C-Bc^3d+6c^2Cd^2-3Bcd^3+(2A+C)d^4)+ad^2(2c(A-C)d-B(c^2-d^2)))\tan(e+}{d^3\left(c^2+d^2\right)^2 f}$$

$$-\frac{\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^3}{2d\left(c^2+d^2\right)f(c+d\tan(e+fx))^2}$$

$$-\frac{(b(3c^4C-Bc^3d-c^2(A-7C)d^2-5Bcd^3+3Ad^4)+2ad^2(2c(A-C)d-B(c^2-d^2)))(a+b\,\mathrm{ta}}{2d^2\left(c^2+d^2\right)^2 f(c+d\tan(e+fx))}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 7.37 (sec) , antiderivative size = 454, normalized size of antiderivative = 0.56

$$\int \frac{(a+b\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^3}\,dx$$
$$= \frac{\frac{(a+ib)^3(A+iB-C)\log(i-\tan(e+fx))}{(-ic+d)^3}+\frac{(a-ib)^3(A-iB-C)\log(i+\tan(e+fx))}{(ic+d)^3}+\frac{2(-bc+ad)\left(b^2\left(3c^6C-Bc^5d+9c^4Cd^2-3Bc^3d^3-c^2(A-10C)\right.\right.}{}$$

```
[In] Integrate[((a + b*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(c + d*Tan[e + f*x])^3,x]
```

```
[Out] (((a + I*b)^3*(A + I*B - C)*Log[I - Tan[e + f*x]])/((-I)*c + d)^3 + ((a - I
*b)^3*(A - I*B - C)*Log[I + Tan[e + f*x]])/(I*c + d)^3 + (2*(-(b*c) + a*d)*
(b^2*(3*c^6*C - B*c^5*d + 9*c^4*C*d^2 - 3*B*c^3*d^3 - c^2*(A - 10*C)*d^4 -
6*B*c*d^5 + 3*A*d^6) + a^2*d^3*(-((A - C)*d*(-3*c^2 + d^2)) - B*(c^3 - 3*c*
d^2)) - a*b*d^2*(8*c*(-A + C)*d^3 + B*(c^4 + 6*c^2*d^2 - 3*d^4)))*Log[c + d
*Tan[e + f*x]])/(d^4*(c^2 + d^2)^3) + ((b*c - a*d)^3*(3*c^2*C - B*c*d + (A
+ 2*C)*d^2))/(d^4*(c^2 + d^2)*(c + d*Tan[e + f*x])^2) + (2*C*(a + b*Tan[e +
 f*x])^3)/(d*(c + d*Tan[e + f*x])^2) - (2*(b*c - a*d)^2*(b*(6*c^4*C - 2*B*c
^3*d + c^2*(A + 11*C)*d^2 - 4*B*c*d^3 + 3*(A + C)*d^4) + a*d^2*(2*c*(A - C)
*d + B*(-c^2 + d^2))))/(d^4*(c^2 + d^2)^2*(c + d*Tan[e + f*x])))/(2*f)
```

Maple [A] (verified)

Time = 0.58 (sec) , antiderivative size = 1271, normalized size of antiderivative = 1.58

method	result	size
derivativedivides	Expression too large to display	1271
default	Expression too large to display	1271
norman	Expression too large to display	2076
parallelrisch	Expression too large to display	6687
risch	Expression too large to display	6825

```
[In] int((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^3,x
,method=_RETURNVERBOSE)
```

```
[Out] 1/f*(tan(f*x+e)*C*b^3/d^3+1/(c^2+d^2)^3*(1/2*(-3*A*a^3*c^2*d+A*a^3*d^3+3*A*
a^2*b*c^3-9*A*a^2*b*c*d^2+9*A*a*b^2*c^2*d-3*A*a*b^2*d^3-A*b^3*c^3+3*A*b^3*c
*d^2+B*a^3*c^3-3*B*a^3*c*d^2+9*B*a^2*b*c^2*d-3*B*a^2*b*d^3-3*B*a*b^2*c^3+9*
B*a*b^2*c*d^2-3*B*b^3*c^2*d+B*b^3*d^3+3*C*a^3*c^2*d-C*a^3*d^3-3*C*a^2*b*c^3
+9*C*a^2*b*c*d^2-9*C*a*b^2*c^2*d+3*C*a*b^2*d^3+C*b^3*c^3-3*C*b^3*c*d^2)*ln(
```

```
1+tan(f*x+e)^2)+(A*a^3*c^3-3*A*a^3*c*d^2+9*A*a^2*b*c^2*d-3*A*a^2*b*d^3-3*A*
a*b^2*c^3+9*A*a*b^2*c*d^2-3*A*b^3*c^2*d+A*b^3*d^3+3*B*a^3*c^2*d-B*a^3*d^3-3
*B*a^2*b*c^3+9*B*a^2*b*c*d^2-9*B*a*b^2*c^2*d+3*B*a*b^2*d^3+B*b^3*c^3-3*B*b^
3*c*d^2-C*a^3*c^3+3*C*a^3*c*d^2-9*C*a^2*b*c^2*d+3*C*a^2*b*d^3+3*C*a*b^2*c^3
-9*C*a*b^2*c*d^2+3*C*b^3*c^2*d-C*b^3*d^3)*arctan(tan(f*x+e)))-1/2/d^4*(A*a^
3*d^5-3*A*a^2*b*c*d^4+3*A*a*b^2*c^2*d^3-A*b^3*c^3*d^2-B*a^3*c*d^4+3*B*a^2*b
*c^2*d^3-3*B*a*b^2*c^3*d^2+B*b^3*c^4*d+C*a^3*c^2*d^3-3*C*a^2*b*c^3*d^2+3*C*
a*b^2*c^4*d-C*b^3*c^5)/(c^2+d^2)/(c+d*tan(f*x+e))^2-1/d^4*(2*A*a^3*c*d^5-3*
A*a^2*b*c^2*d^4+3*A*a^2*b*d^6-6*A*a*b^2*c*d^5+A*b^3*c^4*d^2+3*A*b^3*c^2*d^4
-B*a^3*c^2*d^4+B*a^3*d^6-6*B*a^2*b*c*d^5+3*B*a*b^2*c^4*d^2+9*B*a*b^2*c^2*d^
4-2*B*b^3*c^5*d-4*B*b^3*c^3*d^3-2*C*a^3*c*d^5+3*C*a^2*b*c^4*d^2+9*C*a^2*b*c
^2*d^4-6*C*a*b^2*c^5*d-12*C*a*b^2*c^3*d^3+3*C*b^3*c^6+5*C*b^3*c^4*d^2)/(c^2
+d^2)^2/(c+d*tan(f*x+e))+1/d^4*(3*A*a^3*c^2*d^5-A*a^3*d^7-3*A*a^2*b*c^3*d^4
+9*A*a^2*b*c*d^6-9*A*a*b^2*c^2*d^5+3*A*a*b^2*d^7+A*b^3*c^3*d^4-3*A*b^3*c*d^
6-B*a^3*c^3*d^4+3*B*a^3*c*d^6-9*B*a^2*b*c^2*d^5+3*B*a^2*b*d^7+3*B*a*b^2*c^3
*d^4-9*B*a*b^2*c*d^6+B*b^3*c^6*d+3*B*b^3*c^4*d^3+6*B*b^3*c^2*d^5-3*C*a^3*c^
2*d^5+C*a^3*d^7+3*C*a^2*b*c^3*d^4-9*C*a^2*b*c*d^6+3*C*a*b^2*c^6*d+9*C*a*b^2
*c^4*d^3+18*C*a*b^2*c^2*d^5-3*C*b^3*c^7-9*C*b^3*c^5*d^2-10*C*b^3*c^3*d^4)/(
c^2+d^2)^3*ln(c+d*tan(f*x+e)))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 2490 vs. $2(797) = 1594$.

Time = 1.33 (sec) , antiderivative size = 2490, normalized size of antiderivative = 3.10

$$\int \frac{(a+b\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^3}\,dx = \text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^3,x, algorithm="fricas")
```

```
[Out] -1/2*(3*C*b^3*c^7*d^2 + A*a^3*d^9 - (3*C*a*b^2 + B*b^3)*c^6*d^3 - (3*C*a^2*
b + 3*B*a*b^2 + (A - 9*C)*b^3)*c^5*d^4 + (3*C*a^3 + 9*B*a^2*b + 3*(3*A - 7*
C)*a*b^2 - 7*B*b^3)*c^4*d^5 - 5*(B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - A*b^
3)*c^3*d^6 + ((7*A - 3*C)*a^3 - 9*B*a^2*b - 9*A*a*b^2)*c^2*d^7 + (B*a^3 + 3
*A*a^2*b)*c*d^8 - 2*(C*b^3*c^6*d^3 + 3*C*b^3*c^4*d^5 + 3*C*b^3*c^2*d^7 + C*
b^3*d^9)*tan(f*x + e)^3 - 2*(((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B
*b^3)*c^5*d^4 + 3*(B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*c^4*d
^5 - 3*((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b^3)*c^3*d^6 - (B*a^3
 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*c^2*d^7)*f*x - (9*C*b^3*c^7*d
^2 - A*a^3*d^9 - 3*(3*C*a*b^2 + B*b^3)*c^6*d^3 + (3*C*a^2*b + 3*B*a*b^2 + (
A + 23*C)*b^3)*c^5*d^4 + (C*a^3 + 3*B*a^2*b + 3*(A - 9*C)*a*b^2 - 9*B*b^3)*
c^4*d^5 - (3*B*a^3 + 3*(3*A - 7*C)*a^2*b - 21*B*a*b^2 - (7*A + 12*C)*b^3)*c
^3*d^6 + 5*((A - C)*a^3 - 3*B*a^2*b - 3*A*a*b^2)*c^2*d^7 + (3*B*a^3 + 9*A*a
^2*b + 4*C*b^3)*c*d^8 + 2*(((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b
```

^3)*c^3*d^6 + 3*(B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*c^2*d^7
 - 3*((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b^3)*c*d^8 - (B*a^3 + 3
*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*d^9)*f*x)*tan(f*x + e)^2 + (3*C*b
^3*c^9 + 9*C*b^3*c^7*d^2 - (3*C*a*b^2 + B*b^3)*c^8*d - 3*(3*C*a*b^2 + B*b^3
)*c^6*d^3 + (B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - 10*C)*b^3)*c^5*d^4
 - 3*((A - C)*a^3 - 3*B*a^2*b - 3*(A - 2*C)*a*b^2 + 2*B*b^3)*c^4*d^5 - 3*(B*
a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - A*b^3)*c^3*d^6 + ((A - C)*a^3 - 3*B*a^2
*b - 3*A*a*b^2)*c^2*d^7 + (3*C*b^3*c^7*d^2 + 9*C*b^3*c^5*d^4 - (3*C*a*b^2 +
 B*b^3)*c^6*d^3 - 3*(3*C*a*b^2 + B*b^3)*c^4*d^5 + (B*a^3 + 3*(A - C)*a^2*b
 - 3*B*a*b^2 - (A - 10*C)*b^3)*c^3*d^6 - 3*((A - C)*a^3 - 3*B*a^2*b - 3*(A -
 2*C)*a*b^2 + 2*B*b^3)*c^2*d^7 - 3*(B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - A
*b^3)*c*d^8 + ((A - C)*a^3 - 3*B*a^2*b - 3*A*a*b^2)*d^9)*tan(f*x + e)^2 + 2
*(3*C*b^3*c^8*d + 9*C*b^3*c^6*d^3 - (3*C*a*b^2 + B*b^3)*c^7*d^2 - 3*(3*C*a*
b^2 + B*b^3)*c^5*d^4 + (B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - 10*C)*b^
3)*c^4*d^5 - 3*((A - C)*a^3 - 3*B*a^2*b - 3*(A - 2*C)*a*b^2 + 2*B*b^3)*c^3*
d^6 - 3*(B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - A*b^3)*c^2*d^7 + ((A - C)*a^
3 - 3*B*a^2*b - 3*A*a*b^2)*c*d^8)*tan(f*x + e))*log((d^2*tan(f*x + e)^2 + 2
*c*d*tan(f*x + e) + c^2)/(tan(f*x + e)^2 + 1)) - (3*C*b^3*c^9 + 9*C*b^3*c^7
*d^2 + 9*C*b^3*c^5*d^4 + 3*C*b^3*c^3*d^6 - (3*C*a*b^2 + B*b^3)*c^8*d - 3*(3
*C*a*b^2 + B*b^3)*c^6*d^3 - 3*(3*C*a*b^2 + B*b^3)*c^4*d^5 - (3*C*a*b^2 + B*
b^3)*c^2*d^7 + (3*C*b^3*c^7*d^2 + 9*C*b^3*c^5*d^4 + 9*C*b^3*c^3*d^6 + 3*C*b
^3*c*d^8 - (3*C*a*b^2 + B*b^3)*c^6*d^3 - 3*(3*C*a*b^2 + B*b^3)*c^4*d^5 - 3*
(3*C*a*b^2 + B*b^3)*c^2*d^7 - (3*C*a*b^2 + B*b^3)*d^9)*tan(f*x + e)^2 + 2*(
3*C*b^3*c^8*d + 9*C*b^3*c^6*d^3 + 9*C*b^3*c^4*d^5 + 3*C*b^3*c^2*d^7 - (3*C*
a*b^2 + B*b^3)*c^7*d^2 - 3*(3*C*a*b^2 + B*b^3)*c^5*d^4 - 3*(3*C*a*b^2 + B*b
^3)*c^3*d^6 - (3*C*a*b^2 + B*b^3)*c*d^8)*tan(f*x + e))*log(1/(tan(f*x + e)^
2 + 1)) - 2*(3*C*b^3*c^8*d + 6*C*b^3*c^6*d^3 - (3*C*a*b^2 + B*b^3)*c^7*d^2
 + (C*a^3 + 3*B*a^2*b + 3*(A - 3*C)*a*b^2 - 3*B*b^3)*c^5*d^4 - (2*B*a^3 + 3*
(2*A - 3*C)*a^2*b - 9*B*a*b^2 - (3*A - 2*C)*b^3)*c^4*d^5 + (3*(A - C)*a^3 -
 9*B*a^2*b - 3*(3*A - 4*C)*a*b^2 + 4*B*b^3)*c^3*d^6 + (3*B*a^3 + 9*(A - C)*
a^2*b - 9*B*a*b^2 - (3*A - C)*b^3)*c^2*d^7 - ((3*A - 2*C)*a^3 - 6*B*a^2*b -
 6*A*a*b^2)*c*d^8 - (B*a^3 + 3*A*a^2*b)*d^9 + 2*(((A - C)*a^3 - 3*B*a^2*b -
 3*(A - C)*a*b^2 + B*b^3)*c^4*d^5 + 3*(B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2
 - (A - C)*b^3)*c^3*d^6 - 3*((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b
^3)*c^2*d^7 - (B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*c*d^8)*f*
x)*tan(f*x + e))/((c^6*d^6 + 3*c^4*d^8 + 3*c^2*d^10 + d^12)*f*tan(f*x + e)^
2 + 2*(c^7*d^5 + 3*c^5*d^7 + 3*c^3*d^9 + c*d^11)*f*tan(f*x + e) + (c^8*d^4
 + 3*c^6*d^6 + 3*c^4*d^8 + c^2*d^10)*f)

Sympy [F(-2)]

Exception generated.

$$\int \frac{(a+b\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^3}\,dx$$
$$= \text{Exception raised: AttributeError}$$

```
[In] integrate((a+b*tan(f*x+e))**3*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x
+e))**3,x)
```

```
[Out] Exception raised: AttributeError >> 'NoneType' object has no attribute 'pri
mitive'
```

Maxima [A] (verification not implemented)

none

Time = 0.37 (sec) , antiderivative size = 1110, normalized size of antiderivative = 1.38

$$\int \frac{(a+b\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^3}\,dx = \text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^3,x, algorithm="maxima")
```

```
[Out] 1/2*(2*C*b^3*tan(f*x + e)/d^3 + 2*(((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b
^2 + B*b^3)*c^3 + 3*(B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*c^2
*d - 3*((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b^3)*c*d^2 - (B*a^3 +
 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*d^3)*(f*x + e)/(c^6 + 3*c^4*d^2
 + 3*c^2*d^4 + d^6) - 2*(3*C*b^3*c^7 + 9*C*b^3*c^5*d^2 - (3*C*a*b^2 + B*b^3
)*c^6*d - 3*(3*C*a*b^2 + B*b^3)*c^4*d^3 + (B*a^3 + 3*(A - C)*a^2*b - 3*B*a*
b^2 - (A - 10*C)*b^3)*c^3*d^4 - 3*((A - C)*a^3 - 3*B*a^2*b - 3*(A - 2*C)*a*
b^2 + 2*B*b^3)*c^2*d^5 - 3*(B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - A*b^3)*c*
d^6 + ((A - C)*a^3 - 3*B*a^2*b - 3*A*a*b^2)*d^7)*log(d*tan(f*x + e) + c)/(c
^6*d^4 + 3*c^4*d^6 + 3*c^2*d^8 + d^10) + ((B*a^3 + 3*(A - C)*a^2*b - 3*B*a*
b^2 - (A - C)*b^3)*c^3 - 3*((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b
^3)*c^2*d - 3*(B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*c*d^2 + (
(A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b^3)*d^3)*log(tan(f*x + e)^2
+ 1)/(c^6 + 3*c^4*d^2 + 3*c^2*d^4 + d^6) - (5*C*b^3*c^7 + A*a^3*d^7 - 3*(3*
C*a*b^2 + B*b^3)*c^6*d + (3*C*a^2*b + 3*B*a*b^2 + (A + 9*C)*b^3)*c^5*d^2 +
(C*a^3 + 3*B*a^2*b + 3*(A - 7*C)*a*b^2 - 7*B*b^3)*c^4*d^3 - (3*B*a^3 + 3*(3
*A - 5*C)*a^2*b - 15*B*a*b^2 - 5*A*b^3)*c^3*d^4 + ((5*A - 3*C)*a^3 - 9*B*a^
2*b - 9*A*a*b^2)*c^2*d^5 + (B*a^3 + 3*A*a^2*b)*c*d^6 + 2*(3*C*b^3*c^6*d - 2
*(3*C*a*b^2 + B*b^3)*c^5*d^2 + (3*C*a^2*b + 3*B*a*b^2 + (A + 5*C)*b^3)*c^4*
d^3 - 4*(3*C*a*b^2 + B*b^3)*c^3*d^4 - (B*a^3 + 3*(A - 3*C)*a^2*b - 9*B*a*b^
```

```
2 - 3*A*b^3)*c^2*d^5 + 2*((A - C)*a^3 - 3*B*a^2*b - 3*A*a*b^2)*c*d^6 + (B*a
^3 + 3*A*a^2*b)*d^7)*tan(f*x + e))/(c^6*d^4 + 2*c^4*d^6 + c^2*d^8 + (c^4*d^
6 + 2*c^2*d^8 + d^10)*tan(f*x + e)^2 + 2*(c^5*d^5 + 2*c^3*d^7 + c*d^9)*tan(
f*x + e)))/f
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 2441 vs. $2(797) = 1594$.

Time = 1.39 (sec) , antiderivative size = 2441, normalized size of antiderivative = 3.04

$$\int \frac{(a+b\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^3}\,dx = \text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^3,x, algorithm="giac")
```

```
[Out] 1/2*(2*C*b^3*tan(f*x + e)/d^3 + 2*(A*a^3*c^3 - C*a^3*c^3 - 3*B*a^2*b*c^3 -
3*A*a*b^2*c^3 + 3*C*a*b^2*c^3 + B*b^3*c^3 + 3*B*a^3*c^2*d + 9*A*a^2*b*c^2*d
 - 9*C*a^2*b*c^2*d - 9*B*a*b^2*c^2*d - 3*A*b^3*c^2*d + 3*C*b^3*c^2*d - 3*A*
a^3*c*d^2 + 3*C*a^3*c*d^2 + 9*B*a^2*b*c*d^2 + 9*A*a*b^2*c*d^2 - 9*C*a*b^2*c
*d^2 - 3*B*b^3*c*d^2 - B*a^3*d^3 - 3*A*a^2*b*d^3 + 3*C*a^2*b*d^3 + 3*B*a*b^
2*d^3 + A*b^3*d^3 - C*b^3*d^3)*(f*x + e)/(c^6 + 3*c^4*d^2 + 3*c^2*d^4 + d^6
) + (B*a^3*c^3 + 3*A*a^2*b*c^3 - 3*C*a^2*b*c^3 - 3*B*a*b^2*c^3 - A*b^3*c^3
+ C*b^3*c^3 - 3*A*a^3*c^2*d + 3*C*a^3*c^2*d + 9*B*a^2*b*c^2*d + 9*A*a*b^2*c
^2*d - 9*C*a*b^2*c^2*d - 3*B*b^3*c^2*d - 3*B*a^3*c*d^2 - 9*A*a^2*b*c*d^2 +
9*C*a^2*b*c*d^2 + 9*B*a*b^2*c*d^2 + 3*A*b^3*c*d^2 - 3*C*b^3*c*d^2 + A*a^3*d
^3 - C*a^3*d^3 - 3*B*a^2*b*d^3 - 3*A*a*b^2*d^3 + 3*C*a*b^2*d^3 + B*b^3*d^3)
*log(tan(f*x + e)^2 + 1)/(c^6 + 3*c^4*d^2 + 3*c^2*d^4 + d^6) - 2*(3*C*b^3*c
^7 - 3*C*a*b^2*c^6*d - B*b^3*c^6*d + 9*C*b^3*c^5*d^2 - 9*C*a*b^2*c^4*d^3 -
3*B*b^3*c^4*d^3 + B*a^3*c^3*d^4 + 3*A*a^2*b*c^3*d^4 - 3*C*a^2*b*c^3*d^4 - 3
*B*a*b^2*c^3*d^4 - A*b^3*c^3*d^4 + 10*C*b^3*c^3*d^4 - 3*A*a^3*c^2*d^5 + 3*C
*a^3*c^2*d^5 + 9*B*a^2*b*c^2*d^5 + 9*A*a*b^2*c^2*d^5 - 18*C*a*b^2*c^2*d^5 -
 6*B*b^3*c^2*d^5 - 3*B*a^3*c*d^6 - 9*A*a^2*b*c*d^6 + 9*C*a^2*b*c*d^6 + 9*B*
a*b^2*c*d^6 + 3*A*b^3*c*d^6 + A*a^3*d^7 - C*a^3*d^7 - 3*B*a^2*b*d^7 - 3*A*a
*b^2*d^7)*log(abs(d*tan(f*x + e) + c))/(c^6*d^4 + 3*c^4*d^6 + 3*c^2*d^8 + d
^10) + (9*C*b^3*c^7*d^2*tan(f*x + e)^2 - 9*C*a*b^2*c^6*d^3*tan(f*x + e)^2 -
 3*B*b^3*c^6*d^3*tan(f*x + e)^2 + 27*C*b^3*c^5*d^4*tan(f*x + e)^2 - 27*C*a*
b^2*c^4*d^5*tan(f*x + e)^2 - 9*B*b^3*c^4*d^5*tan(f*x + e)^2 + 3*B*a^3*c^3*d
^6*tan(f*x + e)^2 + 9*A*a^2*b*c^3*d^6*tan(f*x + e)^2 - 9*C*a^2*b*c^3*d^6*ta
n(f*x + e)^2 - 9*B*a*b^2*c^3*d^6*tan(f*x + e)^2 - 3*A*b^3*c^3*d^6*tan(f*x +
 e)^2 + 30*C*b^3*c^3*d^6*tan(f*x + e)^2 - 9*A*a^3*c^2*d^7*tan(f*x + e)^2 +
9*C*a^3*c^2*d^7*tan(f*x + e)^2 + 27*B*a^2*b*c^2*d^7*tan(f*x + e)^2 + 27*A*a
*b^2*c^2*d^7*tan(f*x + e)^2 - 54*C*a*b^2*c^2*d^7*tan(f*x + e)^2 - 18*B*b^3*
c^2*d^7*tan(f*x + e)^2 - 9*B*a^3*c*d^8*tan(f*x + e)^2 - 27*A*a^2*b*c*d^8*ta
n(f*x + e)^2 + 27*C*a^2*b*c*d^8*tan(f*x + e)^2 + 27*B*a*b^2*c*d^8*tan(f*x +
```

```
 e)^2 + 9*A*b^3*c*d^8*tan(f*x + e)^2 + 3*A*a^3*d^9*tan(f*x + e)^2 - 3*C*a^3
*d^9*tan(f*x + e)^2 - 9*B*a^2*b*d^9*tan(f*x + e)^2 - 9*A*a*b^2*d^9*tan(f*x
+ e)^2 + 12*C*b^3*c^8*d*tan(f*x + e) - 6*C*a*b^2*c^7*d^2*tan(f*x + e) - 2*B
*b^3*c^7*d^2*tan(f*x + e) - 6*C*a^2*b*c^6*d^3*tan(f*x + e) - 6*B*a*b^2*c^6*
d^3*tan(f*x + e) - 2*A*b^3*c^6*d^3*tan(f*x + e) + 38*C*b^3*c^6*d^3*tan(f*x
+ e) - 18*C*a*b^2*c^5*d^4*tan(f*x + e) - 6*B*b^3*c^5*d^4*tan(f*x + e) + 8*B
*a^3*c^4*d^5*tan(f*x + e) + 24*A*a^2*b*c^4*d^5*tan(f*x + e) - 42*C*a^2*b*c^
4*d^5*tan(f*x + e) - 42*B*a*b^2*c^4*d^5*tan(f*x + e) - 14*A*b^3*c^4*d^5*tan
(f*x + e) + 50*C*b^3*c^4*d^5*tan(f*x + e) - 22*A*a^3*c^3*d^6*tan(f*x + e) +
 22*C*a^3*c^3*d^6*tan(f*x + e) + 66*B*a^2*b*c^3*d^6*tan(f*x + e) + 66*A*a*b
^2*c^3*d^6*tan(f*x + e) - 84*C*a*b^2*c^3*d^6*tan(f*x + e) - 28*B*b^3*c^3*d^
6*tan(f*x + e) - 18*B*a^3*c^2*d^7*tan(f*x + e) - 54*A*a^2*b*c^2*d^7*tan(f*x
 + e) + 36*C*a^2*b*c^2*d^7*tan(f*x + e) + 36*B*a*b^2*c^2*d^7*tan(f*x + e) +
 12*A*b^3*c^2*d^7*tan(f*x + e) + 2*A*a^3*c*d^8*tan(f*x + e) - 2*C*a^3*c*d^8
*tan(f*x + e) - 6*B*a^2*b*c*d^8*tan(f*x + e) - 6*A*a*b^2*c*d^8*tan(f*x + e)
 - 2*B*a^3*d^9*tan(f*x + e) - 6*A*a^2*b*d^9*tan(f*x + e) + 4*C*b^3*c^9 - 3*
C*a^2*b*c^7*d^2 - 3*B*a*b^2*c^7*d^2 - A*b^3*c^7*d^2 + 13*C*b^3*c^7*d^2 - C*
a^3*c^6*d^3 - 3*B*a^2*b*c^6*d^3 - 3*A*a*b^2*c^6*d^3 + 3*C*a*b^2*c^6*d^3 + B
*b^3*c^6*d^3 + 6*B*a^3*c^5*d^4 + 18*A*a^2*b*c^5*d^4 - 27*C*a^2*b*c^5*d^4 -
27*B*a*b^2*c^5*d^4 - 9*A*b^3*c^5*d^4 + 21*C*b^3*c^5*d^4 - 14*A*a^3*c^4*d^5
+ 11*C*a^3*c^4*d^5 + 33*B*a^2*b*c^4*d^5 + 33*A*a*b^2*c^4*d^5 - 33*C*a*b^2*c
^4*d^5 - 11*B*b^3*c^4*d^5 - 7*B*a^3*c^3*d^6 - 21*A*a^2*b*c^3*d^6 + 12*C*a^2
*b*c^3*d^6 + 12*B*a*b^2*c^3*d^6 + 4*A*b^3*c^3*d^6 - 3*A*a^3*c^2*d^7 - B*a^3
*c*d^8 - 3*A*a^2*b*c*d^8 - A*a^3*d^9)/((c^6*d^4 + 3*c^4*d^6 + 3*c^2*d^8 + d
^10)*(d*tan(f*x + e) + c)^2))/f
```

Mupad [B] (verification not implemented)

Time = 18.17 (sec) , antiderivative size = 1172, normalized size of antiderivative = 1.46

$$\int \frac{(a+b\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^3}\,dx$$

$$= \frac{\ln(c+d\tan(e+f\,x))\,\left(d^3\,(3\,B\,b^3\,c^4+9\,C\,a\,b^2\,c^4)-d^6\,(3\,A\,b^3\,c-3\,B\,a^3\,c-9\,A\,a^2\,b\,c+9\,B\,a\,b^2\,c+9\,C\,a\right.}{}$$

$$+\frac{\ln(\tan(e+f\,x)+1\mathrm{i})\,(A\,a^3+A\,b^3\,1\mathrm{i}-B\,a^3\,1\mathrm{i}+B\,b^3-C\,a^3-C\,b^3\,1\mathrm{i}-3\,A\,a\,b^2-A\,a^2\,b\,3\mathrm{i}+B\,a\,b^2\,3\mathrm{i}}{2\,f\,(-c^3\,1\mathrm{i}-3\,c^2\,d+c\,d^2\,3\mathrm{i}+d^3)}$$

$$-\frac{\frac{A\,a^3\,d^7+5\,C\,b^3\,c^7+B\,a^3\,c\,d^6-3\,B\,b^3\,c^6\,d+5\,A\,a^3\,c^2\,d^5+5\,A\,b^3\,c^3\,d^4+A\,b^3\,c^5\,d^2-3\,B\,a^3\,c^3\,d^4-7\,B\,b^3\,c^4\,d^3-3\,C\,a^3\,c^2\,d^5+C\,a^3\,c^4\,d^3+9\,C\,b^3\,c^5}{}}{}$$

$$+\frac{\ln(\tan(e+f\,x)-\mathrm{i})\,(A\,b^3-B\,a^3-C\,b^3-3\,A\,a^2\,b+3\,B\,a\,b^2+3\,C\,a^2\,b+A\,a^3\,1\mathrm{i}+B\,b^3\,1\mathrm{i}-C\,a^3\,1\mathrm{i}-}{2\,f\,(-c^3-c^2\,d\,3\mathrm{i}+3\,c\,d^2+d^3\,1\mathrm{i})}$$

$$+\frac{C\,b^3\tan(e+f\,x)}{d^3\,f}$$

```
[In] int(((a + b*tan(e + f*x))^3*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d
*tan(e + f*x))^3,x)
```

```
[Out] (log(tan(e + f*x) + 1i)*(A*a^3 + A*b^3*1i - B*a^3*1i + B*b^3 - C*a^3 - C*b^
3*1i - 3*A*a*b^2 - A*a^2*b*3i + B*a*b^2*3i - 3*B*a^2*b + 3*C*a*b^2 + C*a^2*
b*3i))/(2*f*(c*d^2*3i - 3*c^2*d - c^3*1i + d^3)) - ((A*a^3*d^7 + 5*C*b^3*c^
7 + B*a^3*c*d^6 - 3*B*b^3*c^6*d + 5*A*a^3*c^2*d^5 + 5*A*b^3*c^3*d^4 + A*b^3
*c^5*d^2 - 3*B*a^3*c^3*d^4 - 7*B*b^3*c^4*d^3 - 3*C*a^3*c^2*d^5 + C*a^3*c^4*
d^3 + 9*C*b^3*c^5*d^2 - 9*A*a*b^2*c^2*d^5 + 3*A*a*b^2*c^4*d^3 - 9*A*a^2*b*c
^3*d^4 + 15*B*a*b^2*c^3*d^4 + 3*B*a*b^2*c^5*d^2 - 9*B*a^2*b*c^2*d^5 + 3*B*a
^2*b*c^4*d^3 - 21*C*a*b^2*c^4*d^3 + 15*C*a^2*b*c^3*d^4 + 3*C*a^2*b*c^5*d^2
+ 3*A*a^2*b*c*d^6 - 9*C*a*b^2*c^6*d)/(2*d*(c^4 + d^4 + 2*c^2*d^2)) + (tan(e
 + f*x)*(B*a^3*d^6 + 3*C*b^3*c^6 + 3*A*a^2*b*d^6 + 2*A*a^3*c*d^5 - 2*B*b^3*
c^5*d - 2*C*a^3*c*d^5 + 3*A*b^3*c^2*d^4 + A*b^3*c^4*d^2 - B*a^3*c^2*d^4 - 4
*B*b^3*c^3*d^3 + 5*C*b^3*c^4*d^2 - 3*A*a^2*b*c^2*d^4 + 9*B*a*b^2*c^2*d^4 +
3*B*a*b^2*c^4*d^2 - 12*C*a*b^2*c^3*d^3 + 9*C*a^2*b*c^2*d^4 + 3*C*a^2*b*c^4*
d^2 - 6*A*a*b^2*c*d^5 - 6*B*a^2*b*c*d^5 - 6*C*a*b^2*c^5*d))/(c^4 + d^4 + 2*
c^2*d^2))/(f*(c^2*d^3 + d^5*tan(e + f*x)^2 + 2*c*d^4*tan(e + f*x))) + (log(
c + d*tan(e + f*x))*(d^3*(3*B*b^3*c^4 + 9*C*a*b^2*c^4) - d^6*(3*A*b^3*c - 3
*B*a^3*c - 9*A*a^2*b*c + 9*B*a*b^2*c + 9*C*a^2*b*c) + d^5*(3*A*a^3*c^2 + 6*
B*b^3*c^2 - 3*C*a^3*c^2 - 9*A*a*b^2*c^2 - 9*B*a^2*b*c^2 + 18*C*a*b^2*c^2) +
 d^4*(A*b^3*c^3 - B*a^3*c^3 - 10*C*b^3*c^3 - 3*A*a^2*b*c^3 + 3*B*a*b^2*c^3
+ 3*C*a^2*b*c^3) + d^7*(C*a^3 - A*a^3 + 3*A*a*b^2 + 3*B*a^2*b) + d*(B*b^3*c
^6 + 3*C*a*b^2*c^6) - 3*C*b^3*c^7 - 9*C*b^3*c^5*d^2))/(f*(d^10 + 3*c^2*d^8
+ 3*c^4*d^6 + c^6*d^4)) + (log(tan(e + f*x) - 1i)*(A*a^3*1i + A*b^3 - B*a^3
 + B*b^3*1i - C*a^3*1i - C*b^3 - A*a*b^2*3i - 3*A*a^2*b + 3*B*a*b^2 - B*a^2
*b*3i + C*a*b^2*3i + 3*C*a^2*b))/(2*f*(3*c*d^2 - c^2*d*3i - c^3 + d^3*1i))
+ (C*b^3*tan(e + f*x))/(d^3*f)
```

3.85 $\int \frac{(a+b\tan(e+fx))^2(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^3}\,dx$

Optimal result 969
Rubi [A] (verified) 970
Mathematica [C] (verified) 973
Maple [A] (verified) 974
Fricas [B] (verification not implemented) 975
Sympy [F(-2)] 976
Maxima [A] (verification not implemented) 976
Giac [B] (verification not implemented) 977
Mupad [B] (verification not implemented) 978

Optimal result

Integrand size = 45, antiderivative size = 597

$$\int \frac{(a+b\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^3}\,dx =$$

$$-\frac{(b^2(Ac^3-c^3C+3Bc^2d-3Acd^2+3cCd^2-Bd^3)+a^2(c^3C-3Bc^2d-3cCd^2+Bd^3-A(c^3-3cd^2))}{(c^2+d^2)^3}$$

$$-\frac{(2ab(Ac^3-c^3C+3Bc^2d-3Acd^2+3cCd^2-Bd^3)-a^2((A-C)d(3c^2-d^2)-B(c^3-3cd^2))+b^2((A}{(c^2+d^2)^3\,f}$$

$$-\frac{(2abd^3(Ac^3-c^3C+3Bc^2d-3Acd^2+3cCd^2-Bd^3)-b^2(c^6C+3c^4Cd^2+Bc^3d^3-3c^2(A-2C)d^4-}{d^3\,(c^2+d^2)^3\,f}$$

$$-\frac{(c^2C-Bcd+Ad^2)\,(a+b\tan(e+fx))^2}{2d\,(c^2+d^2)\,f(c+d\tan(e+fx))^2}$$

$$+\frac{(bc-ad)\,(b(c^4C-c^2(A-3C)d^2-2Bcd^3+Ad^4)+ad^2(2c(A-C)d-B(c^2-d^2)))}{d^3\,(c^2+d^2)^2\,f(c+d\tan(e+fx))}$$

```
[Out] -(b^2*(A*c^3-3*A*c*d^2+3*B*c^2*d-B*d^3-C*c^3+3*C*c*d^2)+a^2*(c^3*C-3*B*c^2*
d-3*C*c*d^2+B*d^3-A*(c^3-3*c*d^2))-2*a*b*((A-C)*d*(3*c^2-d^2)-B*(c^3-3*c*d^
2)))*x/(c^2+d^2)^3-(2*a*b*(A*c^3-3*A*c*d^2+3*B*c^2*d-B*d^3-C*c^3+3*C*c*d^2)
-a^2*((A-C)*d*(3*c^2-d^2)-B*(c^3-3*c*d^2))+b^2*((A-C)*d*(3*c^2-d^2)-B*(c^3-
3*c*d^2)))*ln(cos(f*x+e))/(c^2+d^2)^3/f-(2*a*b*d^3*(A*c^3-3*A*c*d^2+3*B*c^2
*d-B*d^3-C*c^3+3*C*c*d^2)-b^2*(c^6*C+3*c^4*C*d^2+B*c^3*d^3-3*c^2*(A-2*C)*d^
4-3*B*c*d^5+A*d^6)-a^2*d^3*((A-C)*d*(3*c^2-d^2)-B*(c^3-3*c*d^2)))*ln(c+d*ta
n(f*x+e))/d^3/(c^2+d^2)^3/f-1/2*(A*d^2-B*c*d+C*c^2)*(a+b*tan(f*x+e))^2/d/(c
^2+d^2)/f/(c+d*tan(f*x+e))^2+(-a*d+b*c)*(b*(c^4*C-c^2*(A-3*C)*d^2-2*B*c*d^3
+A*d^4)+a*d^2*(2*c*(A-C)*d-B*(c^2-d^2)))/d^3/(c^2+d^2)^2/f/(c+d*tan(f*x+e))
```

Rubi [A] (verified)

Time = 1.45 (sec) , antiderivative size = 597, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.133, Rules used = {3726, 3716, 3707, 3698, 31, 3556}

$$\int \frac{(a+b\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^3}\,dx =$$

$$-\frac{\log(\cos(e+fx))\,(-(a^2(d(A-C)\,(3c^2-d^2)-B(c^3-3cd^2)))+2ab(Ac^3-3Acd^2+3Bc^2d-Bd^3-c^3C}{f\,(c^2+d^2)^3}$$

$$-\frac{x(a^2(-A(c^3-3cd^2)-3Bc^2d+Bd^3+c^3C-3cCd^2)-2ab(d(A-C)\,(3c^2-d^2)-B(c^3-3cd^2))+b^2(A}{(c^2+d^2)^3}$$

$$-\frac{(-a^2d^3(d(A-C)\,(3c^2-d^2)-B(c^3-3cd^2))+2abd^3(Ac^3-3Acd^2+3Bc^2d-Bd^3-c^3C+3cCd^2)-b}{d^3f\,(c^2+d^2)^3}$$

$$-\frac{(Ad^2-Bcd+c^2C)\,(a+b\tan(e+fx))^2}{2df\,(c^2+d^2)\,(c+d\tan(e+fx))^2}$$

$$+\frac{(bc-ad)\,(ad^2(2cd(A-C)-B(c^2-d^2))+b(-c^2d^2(A-3C)+Ad^4-2Bcd^3+c^4C))}{d^3f\,(c^2+d^2)^2\,(c+d\tan(e+fx))}$$

```
[In] Int[((a + b*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c + d
*Tan[e + f*x])^3,x]
```

```
[Out] -(((b^2*(A*c^3 - c^3*C + 3*B*c^2*d - 3*A*c*d^2 + 3*c*C*d^2 - B*d^3) + a^2*(
c^3*C - 3*B*c^2*d - 3*c*C*d^2 + B*d^3 - A*(c^3 - 3*c*d^2)) - 2*a*b*((A - C)
*d*(3*c^2 - d^2) - B*(c^3 - 3*c*d^2)))*x)/(c^2 + d^2)^3) - ((2*a*b*(A*c^3 -
 c^3*C + 3*B*c^2*d - 3*A*c*d^2 + 3*c*C*d^2 - B*d^3) - a^2*((A - C)*d*(3*c^2
 - d^2) - B*(c^3 - 3*c*d^2)) + b^2*((A - C)*d*(3*c^2 - d^2) - B*(c^3 - 3*c*
d^2)))*Log[Cos[e + f*x]])/((c^2 + d^2)^3*f) - ((2*a*b*d^3*(A*c^3 - c^3*C +
3*B*c^2*d - 3*A*c*d^2 + 3*c*C*d^2 - B*d^3) - b^2*(c^6*C + 3*c^4*C*d^2 + B*c
^3*d^3 - 3*c^2*(A - 2*C)*d^4 - 3*B*c*d^5 + A*d^6) - a^2*d^3*((A - C)*d*(3*c
^2 - d^2) - B*(c^3 - 3*c*d^2)))*Log[c + d*Tan[e + f*x]])/(d^3*(c^2 + d^2)^3
*f) - ((c^2*C - B*c*d + A*d^2)*(a + b*Tan[e + f*x])^2)/(2*d*(c^2 + d^2)*f*(
c + d*Tan[e + f*x])^2) + ((b*c - a*d)*(b*(c^4*C - c^2*(A - 3*C)*d^2 - 2*B*c
*d^3 + A*d^4) + a*d^2*(2*c*(A - C)*d - B*(c^2 - d^2))))/(d^3*(c^2 + d^2)^2*
f*(c + d*Tan[e + f*x]))
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3556

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> Simp[-Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3698

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3707

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[(a*A + b*B - a
*C)*(x/(a^2 + b^2)), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3716

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.
)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(-(b*c - a*d))*(c^2*C - B*c*d + A*d^2)*((c
+ d*Tan[e + f*x])^(n + 1)/(d^2*f*(n + 1)*(c^2 + d^2))), x] + Dist[1/(d*(c^2
 + d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)*Simp[a*d*(A*c - c*C + B*d) + b*(
c^2*C - B*c*d + A*d^2) + d*(A*b*c + a*B*c - b*c*C - a*A*d + b*B*d + a*C*d)*
Tan[e + f*x] + b*C*(c^2 + d^2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c
, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] && LtQ[n,
 -1]
```

Rule 3726

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*d^2 + c*(c*C - B*d))*(a + b*Tan[e +
 f*x])^m*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rubi steps

$$\text{integral} = -\frac{(c^2C-Bcd+Ad^2)(a+b\tan(e+fx))^2}{2d(c^2+d^2)f(c+d\tan(e+fx))^2} + \frac{\int \frac{(a+b\tan(e+fx))(2(Ad(ac+bd)+(bc-ad)(cC-Bd))+2d((A-C)(bc-ad)+B(ac+bd))\tan(e+fx)+2bC(c^2+d^2)\tan^2(e+fx))}{(c+d\tan(e+fx))^2}dx}{2d(c^2+d^2)}$$

$$= -\frac{(c^2C-Bcd+Ad^2)(a+b\tan(e+fx))^2}{2d(c^2+d^2)f(c+d\tan(e+fx))^2} + \frac{(bc-ad)(b(c^4C-c^2(A-3C)d^2-2Bcd^3+Ad^4)+ad^2(2c(A-C)d-B(c^2-d^2)))}{d^3(c^2+d^2)^2f(c+d\tan(e+fx))} + \frac{\int \frac{2(b^2(c^4C-c^2(A-3C)d^2-2Bcd^3+Ad^4)-a^2d^2(c^2C-2Bcd-Cd^2-A(c^2-d^2))+2abd^2(2c(A-C)d-B(c^2-d^2)))-2d^2(2ab(c^2C-2}{c+d\tan(e+}}{2d^2(c^2+}$$

$$= \frac{(a^2(Ac^3-c^3C+3Bc^2d-3Acd^2+3cCd^2-Bd^3)+b^2(c^3C-3Bc^2d-3cCd^2+Bd^3-A(c^3-3cd}{(c^2+d^2)^3} - \frac{(c^2C-Bcd+Ad^2)(a+b\tan(e+fx))^2}{2d(c^2+d^2)f(c+d\tan(e+fx))^2} + \frac{(bc-ad)(b(c^4C-c^2(A-3C)d^2-2Bcd^3+Ad^4)+ad^2(2c(A-C)d-B(c^2-d^2)))}{d^3(c^2+d^2)^2f(c+d\tan(e+fx))} + \frac{(2ab(Ac^3-c^3C+3Bc^2d-3Acd^2+3cCd^2-Bd^3)-a^2((A-C)d(3c^2-d^2)-B(c^3-3cd^2))+}{(c^2+d^2)^3} - \frac{(2abd^3(Ac^3-c^3C+3Bc^2d-3Acd^2+3cCd^2-Bd^3)-b^2(c^6C+3c^4Cd^2+Bc^3d^3-3c^2(A-2C}{d^2(c^2+d^2)^3}$$

$$= \frac{(a^2(Ac^3-c^3C+3Bc^2d-3Acd^2+3cCd^2-Bd^3)+b^2(c^3C-3Bc^2d-3cCd^2+Bd^3-A(c^3-3cd}{(c^2+d^2)^3} - \frac{(2ab(Ac^3-c^3C+3Bc^2d-3Acd^2+3cCd^2-Bd^3)-a^2((A-C)d(3c^2-d^2)-B(c^3-3cd^2))+}{(c^2+d^2)^3f} - \frac{(c^2C-Bcd+Ad^2)(a+b\tan(e+fx))^2}{2d(c^2+d^2)f(c+d\tan(e+fx))^2} + \frac{(bc-ad)(b(c^4C-c^2(A-3C)d^2-2Bcd^3+Ad^4)+ad^2(2c(A-C)d-B(c^2-d^2)))}{d^3(c^2+d^2)^2f(c+d\tan(e+fx))} - \frac{(2abd^3(Ac^3-c^3C+3Bc^2d-3Acd^2+3cCd^2-Bd^3)-b^2(c^6C+3c^4Cd^2+Bc^3d^3-3c^2(A-2C}{d^3(c^2}$$

$$= \frac{(a^2(Ac^3 - c^3C + 3Bc^2d - 3Acd^2 + 3cCd^2 - Bd^3) + b^2(c^3C - 3Bc^2d - 3cCd^2 + Bd^3 - A(c^3 - 3c}{(c^2+d^2)^3}$$

$$- \frac{(2ab(Ac^3 - c^3C + 3Bc^2d - 3Acd^2 + 3cCd^2 - Bd^3) - a^2((A-C)d(3c^2-d^2) - B(c^3-3cd^2)) +}{(c^2+d^2)^3 f}$$

$$- \frac{(2abd^3(Ac^3 - c^3C + 3Bc^2d - 3Acd^2 + 3cCd^2 - Bd^3) - b^2(c^6C + 3c^4Cd^2 + Bc^3d^3 - 3c^2(A-2}{d^3(c^2+d}$$

$$- \frac{(c^2C - Bcd + Ad^2)(a + b\tan(e+fx))^2}{2d(c^2+d^2)f(c+d\tan(e+fx))^2}$$

$$+ \frac{(bc-ad)(b(c^4C - c^2(A-3C)d^2 - 2Bcd^3 + Ad^4) + ad^2(2c(A-C)d - B(c^2-d^2)))}{d^3(c^2+d^2)^2 f(c+d\tan(e+fx))}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 6.98 (sec) , antiderivative size = 1044, normalized size of antiderivative = 1.75

$$\int \frac{(a+b\tan(e+fx))^2 (A + B\tan(e+fx) + C\tan^2(e+fx))}{(c+d\tan(e+fx))^3}\,dx =$$

$$- \frac{(-2aAbc^3 - a^2Bc^3 + b^2Bc^3 + 2abc^3C + 3a^2Ac^2d - 3Ab^2c^2d - 6abBc^2d - 3a^2c^2Cd + 3b^2c^2Cd + 6aAbc}{}$$

$$+ \frac{(2aAbc^3 + a^2Bc^3 - b^2Bc^3 - 2abc^3C - 3a^2Ac^2d + 3Ab^2c^2d + 6abBc^2d + 3a^2c^2Cd - 3b^2c^2Cd - 6aAbcd}{}$$

$$- \frac{(2abd^3(Ac^3 - c^3C + 3Bc^2d - 3Acd^2 + 3cCd^2 - Bd^3) - b^2(c^6C + 3c^4Cd^2 + Bc^3d^3 - 3c^2(A-2C)d^4 -}{d^3(c^2+d^2)^3 f}$$

$$- \frac{(bc-ad)^2(c^2C - Bcd + Ad^2)}{2d^3(c^2+d^2)f(c+d\tan(e+fx))^2}$$

$$+ \frac{(bc-ad)(b(2c^4C - Bc^3d + 4c^2Cd^2 - 3Bcd^3 + 2Ad^4) + ad^2(2c(A-C)d - B(c^2-d^2)))}{d^3(c^2+d^2)^2 f(c+d\tan(e+fx))}$$

```
[In] Integrate[((a + b*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(c + d*Tan[e + f*x])^3,x]
```

```
[Out] -1/2*((-2*a*A*b*c^3 - a^2*B*c^3 + b^2*B*c^3 + 2*a*b*c^3*C + 3*a^2*A*c^2*d -
 3*A*b^2*c^2*d - 6*a*b*B*c^2*d - 3*a^2*c^2*C*d + 3*b^2*c^2*C*d + 6*a*A*b*c*
d^2 + 3*a^2*B*c*d^2 - 3*b^2*B*c*d^2 - 6*a*b*c*C*d^2 - a^2*A*d^3 + A*b^2*d^3
 + 2*a*b*B*d^3 + a^2*C*d^3 - b^2*C*d^3 + I*(a^2*A*c^3 - A*b^2*c^3 - 2*a*b*B
*c^3 - a^2*c^3*C + b^2*c^3*C + 6*a*A*b*c^2*d + 3*a^2*B*c^2*d - 3*b^2*B*c^2*
d - 6*a*b*c^2*C*d - 3*a^2*A*c*d^2 + 3*A*b^2*c*d^2 + 6*a*b*B*c*d^2 + 3*a^2*c
*C*d^2 - 3*b^2*c*C*d^2 - 2*a*A*b*d^3 - a^2*B*d^3 + b^2*B*d^3 + 2*a*b*C*d^3)
)*Log[I - Tan[e + f*x]])/((c^2 + d^2)^3*f) + ((2*a*A*b*c^3 + a^2*B*c^3 - b^
2*B*c^3 - 2*a*b*c^3*C - 3*a^2*A*c^2*d + 3*A*b^2*c^2*d + 6*a*b*B*c^2*d + 3*a
```

```
^2*c^2*C*d - 3*b^2*c^2*C*d - 6*a*A*b*c*d^2 - 3*a^2*B*c*d^2 + 3*b^2*B*c*d^2
+ 6*a*b*c*C*d^2 + a^2*A*d^3 - A*b^2*d^3 - 2*a*b*B*d^3 - a^2*C*d^3 + b^2*C*d
^3 + I*(a^2*A*c^3 - A*b^2*c^3 - 2*a*b*B*c^3 - a^2*c^3*C + b^2*c^3*C + 6*a*A
*b*c^2*d + 3*a^2*B*c^2*d - 3*b^2*B*c^2*d - 6*a*b*c^2*C*d - 3*a^2*A*c*d^2 +
3*A*b^2*c*d^2 + 6*a*b*B*c*d^2 + 3*a^2*c*C*d^2 - 3*b^2*c*C*d^2 - 2*a*A*b*d^3
 - a^2*B*d^3 + b^2*B*d^3 + 2*a*b*C*d^3))*Log[I + Tan[e + f*x]])/(2*(c^2 + d
^2)^3*f) - ((2*a*b*d^3*(A*c^3 - c^3*C + 3*B*c^2*d - 3*A*c*d^2 + 3*c*C*d^2 -
 B*d^3) - b^2*(c^6*C + 3*c^4*C*d^2 + B*c^3*d^3 - 3*c^2*(A - 2*C)*d^4 - 3*B*
c*d^5 + A*d^6) - a^2*d^3*((A - C)*d*(3*c^2 - d^2) - B*(c^3 - 3*c*d^2)))*Log
[c + d*Tan[e + f*x]])/(d^3*(c^2 + d^2)^3*f) - ((b*c - a*d)^2*(c^2*C - B*c*d
 + A*d^2))/(2*d^3*(c^2 + d^2)*f*(c + d*Tan[e + f*x])^2) + ((b*c - a*d)*(b*(
2*c^4*C - B*c^3*d + 4*c^2*C*d^2 - 3*B*c*d^3 + 2*A*d^4) + a*d^2*(2*c*(A - C)
*d - B*(c^2 - d^2))))/(d^3*(c^2 + d^2)^2*f*(c + d*Tan[e + f*x]))
```

Maple [A] (verified)

Time = 0.40 (sec) , antiderivative size = 865, normalized size of antiderivative = 1.45

method	result
derivativedivides	$\frac{\frac{\left(-3A\,a^2c^2d+A\,a^2d^3+2Aab\,c^3-6Aabc\,d^2+3A\,b^2c^2d-A\,b^2d^3+B\,a^2c^3-3B\,a^2c\,d^2+6Bab\,c^2d-2Bab\,d^3-B\,b^2c^3+3B\,b^2c\,d^2+3C\,a^2c^2d-\cdots}{2}}{\cdots}$
default	$\frac{\frac{\left(-3A\,a^2c^2d+A\,a^2d^3+2Aab\,c^3-6Aabc\,d^2+3A\,b^2c^2d-A\,b^2d^3+B\,a^2c^3-3B\,a^2c\,d^2+6Bab\,c^2d-2Bab\,d^3-B\,b^2c^3+3B\,b^2c\,d^2+3C\,a^2c^2d-\cdots}{2}}{\cdots}$
norman	Expression too large to display
risch	Expression too large to display
parallelrisch	Expression too large to display

```
[In] int((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^3,x
,method=_RETURNVERBOSE)
```

```
[Out] 1/f*(1/(c^2+d^2)^3*(1/2*(-3*A*a^2*c^2*d+A*a^2*d^3+2*A*a*b*c^3-6*A*a*b*c*d^2
+3*A*b^2*c^2*d-A*b^2*d^3+B*a^2*c^3-3*B*a^2*c*d^2+6*B*a*b*c^2*d-2*B*a*b*d^3-
B*b^2*c^3+3*B*b^2*c*d^2+3*C*a^2*c^2*d-C*a^2*d^3-2*C*a*b*c^3+6*C*a*b*c*d^2-3
*C*b^2*c^2*d+C*b^2*d^3)*ln(1+tan(f*x+e)^2)+(A*a^2*c^3-3*A*a^2*c*d^2+6*A*a*b
*c^2*d-2*A*a*b*d^3-A*b^2*c^3+3*A*b^2*c*d^2+3*B*a^2*c^2*d-B*a^2*d^3-2*B*a*b*
c^3+6*B*a*b*c*d^2-3*B*b^2*c^2*d+B*b^2*d^3-C*a^2*c^3+3*C*a^2*c*d^2-6*C*a*b*c
^2*d+2*C*a*b*d^3+C*b^2*c^3-3*C*b^2*c*d^2)*arctan(tan(f*x+e)))-1/2*(A*a^2*d^
4-2*A*a*b*c*d^3+A*b^2*c^2*d^2-B*a^2*c*d^3+2*B*a*b*c^2*d^2-B*b^2*c^3*d+C*a^2
*c^2*d^2-2*C*a*b*c^3*d+C*b^2*c^4)/d^3/(c^2+d^2)/(c+d*tan(f*x+e))^2-(2*A*a^2
*c*d^4-2*A*a*b*c^2*d^3+2*A*a*b*d^5-2*A*b^2*c*d^4-B*a^2*c^2*d^3+B*a^2*d^5-4*
B*a*b*c*d^4+B*b^2*c^4*d+3*B*b^2*c^2*d^3-2*C*a^2*c*d^4+2*C*a*b*c^4*d+6*C*a*b
*c^2*d^3-2*C*b^2*c^5-4*C*b^2*c^3*d^2)/d^3/(c^2+d^2)^2/(c+d*tan(f*x+e))+(3*A
*a^2*c^2*d^4-A*a^2*d^6-2*A*a*b*c^3*d^3+6*A*a*b*c*d^5-3*A*b^2*c^2*d^4+A*b^2*
```

```
d^6-B*a^2*c^3*d^3+3*B*a^2*c*d^5-6*B*a*b*c^2*d^4+2*B*a*b*d^6+B*b^2*c^3*d^3-3
*B*b^2*c*d^5-3*C*a^2*c^2*d^4+C*a^2*d^6+2*C*a*b*c^3*d^3-6*C*a*b*c*d^5+C*b^2*
c^6+3*C*b^2*c^4*d^2+6*C*b^2*c^2*d^4)/(c^2+d^2)^3/d^3*ln(c+d*tan(f*x+e)))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1618 vs. 2(591) = 1182.

Time = 0.65 (sec) , antiderivative size = 1618, normalized size of antiderivative = 2.71

$$\int \frac{(a+b\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^3}\,dx = \text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^3,x, algorithm="fricas")
```

```
[Out] 1/2*(C*b^2*c^6*d^2 - A*a^2*d^8 + (2*C*a*b + B*b^2)*c^5*d^3 - (3*C*a^2 + 6*B
*a*b + (3*A - 7*C)*b^2)*c^4*d^4 + 5*(B*a^2 + 2*(A - C)*a*b - B*b^2)*c^3*d^5
 - ((7*A - 3*C)*a^2 - 6*B*a*b - 3*A*b^2)*c^2*d^6 - (B*a^2 + 2*A*a*b)*c*d^7
+ 2*(((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*c^5*d^3 + 3*(B*a^2 + 2*(A - C)*a
*b - B*b^2)*c^4*d^4 - 3*((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*c^3*d^5 - (B*
a^2 + 2*(A - C)*a*b - B*b^2)*c^2*d^6)*f*x - (3*C*b^2*c^6*d^2 + A*a^2*d^8 -
(2*C*a*b + B*b^2)*c^5*d^3 - (C*a^2 + 2*B*a*b + (A - 9*C)*b^2)*c^4*d^4 + (3*
B*a^2 + 2*(3*A - 7*C)*a*b - 7*B*b^2)*c^3*d^5 - 5*((A - C)*a^2 - 2*B*a*b - A
*b^2)*c^2*d^6 - 3*(B*a^2 + 2*A*a*b)*c*d^7 - 2*(((A - C)*a^2 - 2*B*a*b - (A
- C)*b^2)*c^3*d^5 + 3*(B*a^2 + 2*(A - C)*a*b - B*b^2)*c^2*d^6 - 3*((A - C)*
a^2 - 2*B*a*b - (A - C)*b^2)*c*d^7 - (B*a^2 + 2*(A - C)*a*b - B*b^2)*d^8)*f
*x)*tan(f*x + e)^2 + (C*b^2*c^8 + 3*C*b^2*c^6*d^2 - (B*a^2 + 2*(A - C)*a*b
- B*b^2)*c^5*d^3 + 3*((A - C)*a^2 - 2*B*a*b - (A - 2*C)*b^2)*c^4*d^4 + 3*(B
*a^2 + 2*(A - C)*a*b - B*b^2)*c^3*d^5 - ((A - C)*a^2 - 2*B*a*b - A*b^2)*c^2
*d^6 + (C*b^2*c^6*d^2 + 3*C*b^2*c^4*d^4 - (B*a^2 + 2*(A - C)*a*b - B*b^2)*c
^3*d^5 + 3*((A - C)*a^2 - 2*B*a*b - (A - 2*C)*b^2)*c^2*d^6 + 3*(B*a^2 + 2*(
A - C)*a*b - B*b^2)*c*d^7 - ((A - C)*a^2 - 2*B*a*b - A*b^2)*d^8)*tan(f*x +
e)^2 + 2*(C*b^2*c^7*d + 3*C*b^2*c^5*d^3 - (B*a^2 + 2*(A - C)*a*b - B*b^2)*c
^4*d^4 + 3*((A - C)*a^2 - 2*B*a*b - (A - 2*C)*b^2)*c^3*d^5 + 3*(B*a^2 + 2*(
A - C)*a*b - B*b^2)*c^2*d^6 - ((A - C)*a^2 - 2*B*a*b - A*b^2)*c*d^7)*tan(f*
x + e))*log((d^2*tan(f*x + e)^2 + 2*c*d*tan(f*x + e) + c^2)/(tan(f*x + e)^2
 + 1)) - (C*b^2*c^8 + 3*C*b^2*c^6*d^2 + 3*C*b^2*c^4*d^4 + C*b^2*c^2*d^6 + (
C*b^2*c^6*d^2 + 3*C*b^2*c^4*d^4 + 3*C*b^2*c^2*d^6 + C*b^2*d^8)*tan(f*x + e)
^2 + 2*(C*b^2*c^7*d + 3*C*b^2*c^5*d^3 + 3*C*b^2*c^3*d^5 + C*b^2*c*d^7)*tan(
f*x + e))*log(1/(tan(f*x + e)^2 + 1)) - 2*(C*b^2*c^7*d - (C*a^2 + 2*B*a*b +
 (A - 3*C)*b^2)*c^5*d^3 + (2*B*a^2 + 2*(2*A - 3*C)*a*b - 3*B*b^2)*c^4*d^4 -
 (3*(A - C)*a^2 - 6*B*a*b - (3*A - 4*C)*b^2)*c^3*d^5 - 3*(B*a^2 + 2*(A - C)
*a*b - B*b^2)*c^2*d^6 + ((3*A - 2*C)*a^2 - 4*B*a*b - 2*A*b^2)*c*d^7 + (B*a^
2 + 2*A*a*b)*d^8 - 2*(((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*c^4*d^4 + 3*(B*
a^2 + 2*(A - C)*a*b - B*b^2)*c^3*d^5 - 3*((A - C)*a^2 - 2*B*a*b - (A - C)*b
```

```
^2)*c^2*d^6 - (B*a^2 + 2*(A - C)*a*b - B*b^2)*c*d^7)*f*x)*tan(f*x + e))/((c
^6*d^5 + 3*c^4*d^7 + 3*c^2*d^9 + d^11)*f*tan(f*x + e)^2 + 2*(c^7*d^4 + 3*c^
5*d^6 + 3*c^3*d^8 + c*d^10)*f*tan(f*x + e) + (c^8*d^3 + 3*c^6*d^5 + 3*c^4*d
^7 + c^2*d^9)*f)
```

Sympy [F(-2)]

Exception generated.

$$\int \frac{(a+b\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^3}\,dx$$
$$= \text{Exception raised: AttributeError}$$

```
[In] integrate((a+b*tan(f*x+e))**2*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x
+e))**3,x)
```

```
[Out] Exception raised: AttributeError >> 'NoneType' object has no attribute 'pri
mitive'
```

Maxima [A] (verification not implemented)

none

Time = 0.34 (sec) , antiderivative size = 827, normalized size of antiderivative = 1.39

$$\int \frac{(a+b\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^3}\,dx$$
$$= \frac{\frac{2\left(\left(\left((A-C)a^2-2\,Bab-(A-C)b^2\right)c^3+3\left(Ba^2+2\,(A-C)ab-Bb^2\right)c^2d-3\left((A-C)a^2-2\,Bab-(A-C)b^2\right)cd^2-\left(Ba^2+2\,(A-C)ab-Bb^2\right)d^3\right)(fx+e)}{c^6+3\,c^4d^2+3\,c^2d^4+d^6}+}{}$$

```
[In] integrate((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^3,x, algorithm="maxima")
```

```
[Out] 1/2*(2*(((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*c^3 + 3*(B*a^2 + 2*(A - C)*a*
b - B*b^2)*c^2*d - 3*((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*c*d^2 - (B*a^2 +
 2*(A - C)*a*b - B*b^2)*d^3)*(f*x + e)/(c^6 + 3*c^4*d^2 + 3*c^2*d^4 + d^6)
+ 2*(C*b^2*c^6 + 3*C*b^2*c^4*d^2 - (B*a^2 + 2*(A - C)*a*b - B*b^2)*c^3*d^3
+ 3*((A - C)*a^2 - 2*B*a*b - (A - 2*C)*b^2)*c^2*d^4 + 3*(B*a^2 + 2*(A - C)*
a*b - B*b^2)*c*d^5 - ((A - C)*a^2 - 2*B*a*b - A*b^2)*d^6)*log(d*tan(f*x + e
) + c)/(c^6*d^3 + 3*c^4*d^5 + 3*c^2*d^7 + d^9) + ((B*a^2 + 2*(A - C)*a*b -
B*b^2)*c^3 - 3*((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*c^2*d - 3*(B*a^2 + 2*(
A - C)*a*b - B*b^2)*c*d^2 + ((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*d^3)*log(
tan(f*x + e)^2 + 1)/(c^6 + 3*c^4*d^2 + 3*c^2*d^4 + d^6) + (3*C*b^2*c^6 - A*
a^2*d^6 - (2*C*a*b + B*b^2)*c^5*d - (C*a^2 + 2*B*a*b + (A - 7*C)*b^2)*c^4*d
```

```
^2 + (3*B*a^2 + 2*(3*A - 5*C)*a*b - 5*B*b^2)*c^3*d^3 - ((5*A - 3*C)*a^2 - 6
*B*a*b - 3*A*b^2)*c^2*d^4 - (B*a^2 + 2*A*a*b)*c*d^5 + 2*(2*C*b^2*c^5*d + 4*
C*b^2*c^3*d^3 - (2*C*a*b + B*b^2)*c^4*d^2 + (B*a^2 + 2*(A - 3*C)*a*b - 3*B*
b^2)*c^2*d^4 - 2*((A - C)*a^2 - 2*B*a*b - A*b^2)*c*d^5 - (B*a^2 + 2*A*a*b)*
d^6)*tan(f*x + e))/(c^6*d^3 + 2*c^4*d^5 + c^2*d^7 + (c^4*d^5 + 2*c^2*d^7 +
d^9)*tan(f*x + e)^2 + 2*(c^5*d^4 + 2*c^3*d^6 + c*d^8)*tan(f*x + e)))/f
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1663 vs. $2(591) = 1182$.

Time = 1.05 (sec) , antiderivative size = 1663, normalized size of antiderivative = 2.79

$$\int \frac{(a+b\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^3}\,dx = \text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^3,x, algorithm="giac")
```

```
[Out] 1/2*(2*(A*a^2*c^3 - C*a^2*c^3 - 2*B*a*b*c^3 - A*b^2*c^3 + C*b^2*c^3 + 3*B*a
^2*c^2*d + 6*A*a*b*c^2*d - 6*C*a*b*c^2*d - 3*B*b^2*c^2*d - 3*A*a^2*c*d^2 +
3*C*a^2*c*d^2 + 6*B*a*b*c*d^2 + 3*A*b^2*c*d^2 - 3*C*b^2*c*d^2 - B*a^2*d^3 -
 2*A*a*b*d^3 + 2*C*a*b*d^3 + B*b^2*d^3)*(f*x + e)/(c^6 + 3*c^4*d^2 + 3*c^2*
d^4 + d^6) + (B*a^2*c^3 + 2*A*a*b*c^3 - 2*C*a*b*c^3 - B*b^2*c^3 - 3*A*a^2*c
^2*d + 3*C*a^2*c^2*d + 6*B*a*b*c^2*d + 3*A*b^2*c^2*d - 3*C*b^2*c^2*d - 3*B*
a^2*c*d^2 - 6*A*a*b*c*d^2 + 6*C*a*b*c*d^2 + 3*B*b^2*c*d^2 + A*a^2*d^3 - C*a
^2*d^3 - 2*B*a*b*d^3 - A*b^2*d^3 + C*b^2*d^3)*log(tan(f*x + e)^2 + 1)/(c^6
+ 3*c^4*d^2 + 3*c^2*d^4 + d^6) + 2*(C*b^2*c^6 + 3*C*b^2*c^4*d^2 - B*a^2*c^3
*d^3 - 2*A*a*b*c^3*d^3 + 2*C*a*b*c^3*d^3 + B*b^2*c^3*d^3 + 3*A*a^2*c^2*d^4
- 3*C*a^2*c^2*d^4 - 6*B*a*b*c^2*d^4 - 3*A*b^2*c^2*d^4 + 6*C*b^2*c^2*d^4 + 3
*B*a^2*c*d^5 + 6*A*a*b*c*d^5 - 6*C*a*b*c*d^5 - 3*B*b^2*c*d^5 - A*a^2*d^6 +
C*a^2*d^6 + 2*B*a*b*d^6 + A*b^2*d^6)*log(abs(d*tan(f*x + e) + c))/(c^6*d^3
+ 3*c^4*d^5 + 3*c^2*d^7 + d^9) - (3*C*b^2*c^6*d*tan(f*x + e)^2 + 9*C*b^2*c^
4*d^3*tan(f*x + e)^2 - 3*B*a^2*c^3*d^4*tan(f*x + e)^2 - 6*A*a*b*c^3*d^4*tan
(f*x + e)^2 + 6*C*a*b*c^3*d^4*tan(f*x + e)^2 + 3*B*b^2*c^3*d^4*tan(f*x + e)
^2 + 9*A*a^2*c^2*d^5*tan(f*x + e)^2 - 9*C*a^2*c^2*d^5*tan(f*x + e)^2 - 18*B
*a*b*c^2*d^5*tan(f*x + e)^2 - 9*A*b^2*c^2*d^5*tan(f*x + e)^2 + 18*C*b^2*c^2
*d^5*tan(f*x + e)^2 + 9*B*a^2*c*d^6*tan(f*x + e)^2 + 18*A*a*b*c*d^6*tan(f*x
 + e)^2 - 18*C*a*b*c*d^6*tan(f*x + e)^2 - 9*B*b^2*c*d^6*tan(f*x + e)^2 - 3*
A*a^2*d^7*tan(f*x + e)^2 + 3*C*a^2*d^7*tan(f*x + e)^2 + 6*B*a*b*d^7*tan(f*x
 + e)^2 + 3*A*b^2*d^7*tan(f*x + e)^2 + 2*C*b^2*c^7*tan(f*x + e) + 4*C*a*b*c
^6*d*tan(f*x + e) + 2*B*b^2*c^6*d*tan(f*x + e) + 6*C*b^2*c^5*d^2*tan(f*x +
e) - 8*B*a^2*c^4*d^3*tan(f*x + e) - 16*A*a*b*c^4*d^3*tan(f*x + e) + 28*C*a*
b*c^4*d^3*tan(f*x + e) + 14*B*b^2*c^4*d^3*tan(f*x + e) + 22*A*a^2*c^3*d^4*t
an(f*x + e) - 22*C*a^2*c^3*d^4*tan(f*x + e) - 44*B*a*b*c^3*d^4*tan(f*x + e)
 - 22*A*b^2*c^3*d^4*tan(f*x + e) + 28*C*b^2*c^3*d^4*tan(f*x + e) + 18*B*a^2
```

*c^2*d^5*tan(f*x + e) + 36*A*a*b*c^2*d^5*tan(f*x + e) - 24*C*a*b*c^2*d^5*ta
n(f*x + e) - 12*B*b^2*c^2*d^5*tan(f*x + e) - 2*A*a^2*c*d^6*tan(f*x + e) + 2
*C*a^2*c*d^6*tan(f*x + e) + 4*B*a*b*c*d^6*tan(f*x + e) + 2*A*b^2*c*d^6*tan(
f*x + e) + 2*B*a^2*d^7*tan(f*x + e) + 4*A*a*b*d^7*tan(f*x + e) + 2*C*a*b*c^
7 + B*b^2*c^7 + C*a^2*c^6*d + 2*B*a*b*c^6*d + A*b^2*c^6*d - C*b^2*c^6*d - 6
*B*a^2*c^5*d^2 - 12*A*a*b*c^5*d^2 + 18*C*a*b*c^5*d^2 + 9*B*b^2*c^5*d^2 + 14
*A*a^2*c^4*d^3 - 11*C*a^2*c^4*d^3 - 22*B*a*b*c^4*d^3 - 11*A*b^2*c^4*d^3 + 1
1*C*b^2*c^4*d^3 + 7*B*a^2*c^3*d^4 + 14*A*a*b*c^3*d^4 - 8*C*a*b*c^3*d^4 - 4*
B*b^2*c^3*d^4 + 3*A*a^2*c^2*d^5 + B*a^2*c*d^6 + 2*A*a*b*c*d^6 + A*a^2*d^7)/
((c^6*d^2 + 3*c^4*d^4 + 3*c^2*d^6 + d^8)*(d*tan(f*x + e) + c)^2))/f

Mupad [B] (verification not implemented)

Time = 27.06 (sec) , antiderivative size = 807, normalized size of antiderivative = 1.35

$$\int \frac{(a+b\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^3}\,dx =$$

$$-\frac{\frac{A\,a^2\,d^6-3\,C\,b^2\,c^6+B\,a^2\,c\,d^5+B\,b^2\,c^5\,d+5\,A\,a^2\,c^2\,d^4-3\,A\,b^2\,c^2\,d^4+A\,b^2\,c^4\,d^2-3\,B\,a^2\,c^3\,d^3+5\,B\,b^2\,c^3\,d^3-3\,C\,a^2\,c^2\,d^4+C\,a^2\,c^4\,d^2-7\,C\,b^2\,c^4\,d^2+}{2\,d^3\,(c^4+2\,c^2\,d^2+d^4)}}{}$$

$$-\frac{\ln\left(c+d\tan(e+f\,x)\right)\,\left(\frac{c^2\,\left(d^4\,\left(3\,A\,b^2-3\,A\,a^2+3\,C\,a^2-6\,C\,b^2+6\,B\,a\,b\right)+3\,C\,b^2\,d^4\right)-d^6\,\left(A\,b^2-A\,a^2+C\,a^2+2\,B\,a\,b\right)+C\,b^2\,d^6-c\,d^5\,(3\,B}{c^6\,d^3+3\,c^4\,d^5+3\,c^2\,d^7+d^9}\right.}{f}$$

$$-\frac{\ln\left(\tan(e+f\,x)-\mathrm{i}\right)\,\left(B\,a^2-B\,b^2+2\,A\,a\,b-2\,C\,a\,b-A\,a^2\,1\mathrm{i}+A\,b^2\,1\mathrm{i}+C\,a^2\,1\mathrm{i}-C\,b^2\,1\mathrm{i}+B\,a\,b\,2\mathrm{i}\right)}{2\,f\,\left(-c^3-c^2\,d\,3\mathrm{i}+3\,c\,d^2+d^3\,1\mathrm{i}\right)}$$

$$-\frac{\ln\left(\tan(e+f\,x)+1\mathrm{i}\right)\,\left(A\,b^2-A\,a^2+B\,a^2\,1\mathrm{i}-B\,b^2\,1\mathrm{i}+C\,a^2-C\,b^2+A\,a\,b\,2\mathrm{i}+2\,B\,a\,b-C\,a\,b\,2\mathrm{i}\right)}{2\,f\,\left(-c^3\,1\mathrm{i}-3\,c^2\,d+c\,d^2\,3\mathrm{i}+d^3\right)}$$

[In] int(((a + b*tan(e + f*x))^2*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d
*tan(e + f*x))^3,x)

[Out] - ((A*a^2*d^6 - 3*C*b^2*c^6 + B*a^2*c*d^5 + B*b^2*c^5*d + 5*A*a^2*c^2*d^4 -
3*A*b^2*c^2*d^4 + A*b^2*c^4*d^2 - 3*B*a^2*c^3*d^3 + 5*B*b^2*c^3*d^3 - 3*C*
a^2*c^2*d^4 + C*a^2*c^4*d^2 - 7*C*b^2*c^4*d^2 + 2*A*a*b*c*d^5 + 2*C*a*b*c^5
*d - 6*A*a*b*c^3*d^3 - 6*B*a*b*c^2*d^4 + 2*B*a*b*c^4*d^2 + 10*C*a*b*c^3*d^3
)/(2*d^3*(c^4 + d^4 + 2*c^2*d^2)) + (tan(e + f*x)*(B*a^2*d^5 - 2*C*b^2*c^5
+ 2*A*a*b*d^5 + 2*A*a^2*c*d^4 - 2*A*b^2*c*d^4 + B*b^2*c^4*d - 2*C*a^2*c*d^4
- B*a^2*c^2*d^3 + 3*B*b^2*c^2*d^3 - 4*C*b^2*c^3*d^2 - 4*B*a*b*c*d^4 + 2*C*
a*b*c^4*d - 2*A*a*b*c^2*d^3 + 6*C*a*b*c^2*d^3))/(d^2*(c^4 + d^4 + 2*c^2*d^2
)))/(f*(c^2 + d^2*tan(e + f*x)^2 + 2*c*d*tan(e + f*x))) - (log(c + d*tan(e
+ f*x))*((c^2*(d^4*(3*A*b^2 - 3*A*a^2 + 3*C*a^2 - 6*C*b^2 + 6*B*a*b) + 3*C*
b^2*d^4) - d^6*(A*b^2 - A*a^2 + C*a^2 + 2*B*a*b) + C*b^2*d^6 - c*d^5*(3*B*a
^2 - 3*B*b^2 + 6*A*a*b - 6*C*a*b) + c^3*d^3*(B*a^2 - B*b^2 + 2*A*a*b - 2*C*
a*b))/(d^9 + 3*c^2*d^7 + 3*c^4*d^5 + c^6*d^3) - (C*b^2)/d^3))/f - (log(tan(
e + f*x) - 1i)*(A*b^2*1i - A*a^2*1i + B*a^2 - B*b^2 + C*a^2*1i - C*b^2*1i +

```
2*A*a*b + B*a*b*2i - 2*C*a*b))/(2*f*(3*c*d^2 - c^2*d*3i - c^3 + d^3*1i)) -
 (log(tan(e + f*x) + 1i)*(A*b^2 - A*a^2 + B*a^2*1i - B*b^2*1i + C*a^2 - C*b
^2 + A*a*b*2i + 2*B*a*b - C*a*b*2i))/(2*f*(c*d^2*3i - 3*c^2*d - c^3*1i + d^
3))
```

3.86 $\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^3}\,dx$

Optimal result 980
Rubi [A] (verified) 981
Mathematica [C] (verified) 983
Maple [A] (verified) 984
Fricas [B] (verification not implemented) 984
Sympy [F(-2)] 985
Maxima [A] (verification not implemented) 985
Giac [B] (verification not implemented) 986
Mupad [B] (verification not implemented) 987

Optimal result

Integrand size = 43, antiderivative size = 352

$$\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^3}\,dx = -\frac{(a(c^3C-3Bc^2d-3cCd^2+Bd^3-A(c^3-3cd^2))-b((A-C)d(3c^2-d^2)-B(c^3-3cd^2)))\,x}{(c^2+d^2)^3} + \frac{(b(c^3C-3Bc^2d-3cCd^2+Bd^3)-a(Bc^3+3c^2Cd-3Bcd^2-Cd^3)+A(ad(3c^2-d^2)-b(c^3-3cd^2)))\,\mathrm{l}}{(c^2+d^2)^3 f} + \frac{(bc-ad)\left(c^2C-Bcd+Ad^2\right)}{2d^2\left(c^2+d^2\right)f(c+d\tan(e+fx))^2} - \frac{b(c^4C-c^2(A-3C)d^2-2Bcd^3+Ad^4)+ad^2(2c(A-C)d-B(c^2-d^2))}{d^2\left(c^2+d^2\right)^2 f(c+d\tan(e+fx))}$$

```
[Out] -(a*(c^3*C-3*B*c^2*d-3*C*c*d^2+B*d^3-A*(c^3-3*c*d^2))-b*((A-C)*d*(3*c^2-d^2
)-B*(c^3-3*c*d^2)))*x/(c^2+d^2)^3+(b*(-3*B*c^2*d+B*d^3+C*c^3-3*C*c*d^2)-a*(
B*c^3-3*B*c*d^2+3*C*c^2*d-C*d^3)+A*(a*d*(3*c^2-d^2)-b*(c^3-3*c*d^2)))*ln(c*
cos(f*x+e)+d*sin(f*x+e))/(c^2+d^2)^3/f+1/2*(-a*d+b*c)*(A*d^2-B*c*d+C*c^2)/d
^2/(c^2+d^2)/f/(c+d*tan(f*x+e))^2+(-b*(c^4*C-c^2*(A-3*C)*d^2-2*B*c*d^3+A*d^
4)-a*d^2*(2*c*(A-C)*d-B*(c^2-d^2)))/d^2/(c^2+d^2)^2/f/(c+d*tan(f*x+e))
```

Rubi [A] (verified)

Time = 0.74 (sec) , antiderivative size = 349, normalized size of antiderivative = 0.99, number of steps used = 4, number of rules used = 4, $\frac{\text{number of rules}}{\text{integrand size}} = 0.093$, Rules used = {3716, 3709, 3612, 3611}

$$\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^3}\,dx$$
$$= \frac{(bc-ad)\left(Ad^2-Bcd+c^2C\right)}{2d^2f\left(c^2+d^2\right)(c+d\tan(e+fx))^2}$$
$$-\frac{ad^2(2cd(A-C)-B(c^2-d^2))+b(-c^2d^2(A-3C)+Ad^4-2Bcd^3+c^4C)}{d^2f\left(c^2+d^2\right)^2(c+d\tan(e+fx))}$$
$$+\frac{(aAd(3c^2-d^2)-a(Bc^3-3Bcd^2+3c^2Cd-Cd^3)-Ab(c^3-3cd^2)+b(-3Bc^2d+Bd^3+c^3C-3cCd^2}{f\left(c^2+d^2\right)^3}$$
$$+\frac{x(-a(-A(c^3-3cd^2)-3Bc^2d+Bd^3+c^3C-3cCd^2)+bd(A-C)\left(3c^2-d^2\right)-bB(c^3-3cd^2))}{\left(c^2+d^2\right)^3}$$

```
[In] Int[((a + b*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c + d*T
an[e + f*x])^3,x]
```

```
[Out] ((b*(A - C)*d*(3*c^2 - d^2) - b*B*(c^3 - 3*c*d^2) - a*(c^3*C - 3*B*c^2*d -
3*c*C*d^2 + B*d^3 - A*(c^3 - 3*c*d^2)))*x)/(c^2 + d^2)^3 + ((a*A*d*(3*c^2 -
 d^2) - A*b*(c^3 - 3*c*d^2) + b*(c^3*C - 3*B*c^2*d - 3*c*C*d^2 + B*d^3) - a
*(B*c^3 + 3*c^2*C*d - 3*B*c*d^2 - C*d^3))*Log[c*Cos[e + f*x] + d*Sin[e + f*
x]])/((c^2 + d^2)^3*f) + ((b*c - a*d)*(c^2*C - B*c*d + A*d^2))/(2*d^2*(c^2
+ d^2)*f*(c + d*Tan[e + f*x])^2) - (b*(c^4*C - c^2*(A - 3*C)*d^2 - 2*B*c*d^
3 + A*d^4) + a*d^2*(2*c*(A - C)*d - B*(c^2 - d^2)))/(d^2*(c^2 + d^2)^2*f*(c
 + d*Tan[e + f*x]))
```

Rule 3611

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c/(b*f))*Log[RemoveContent[a*Cos[e + f*x] + b*Si
n[e + f*x], x]], x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3612

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[(a*c + b*d)*(x/(a^2 + b^2)), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3709

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 -
 a*b*B + a^2*C)*((a + b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1)*(a^2 + b^2))), x
] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[b*B + a*(A -
C) - (A*b - a*B - b*C)*Tan[e + f*x], x], x], x] /; FreeQ[{a, b, e, f, A, B,
 C}, x] && NeQ[A*b^2 - a*b*B + a^2*C, 0] && LtQ[m, -1] && NeQ[a^2 + b^2, 0]
```

Rule 3716

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.
)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(-(b*c - a*d))*(c^2*C - B*c*d + A*d^2)*((c
+ d*Tan[e + f*x])^(n + 1)/(d^2*f*(n + 1)*(c^2 + d^2))), x] + Dist[1/(d*(c^2
 + d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)*Simp[a*d*(A*c - c*C + B*d) + b*(
c^2*C - B*c*d + A*d^2) + d*(A*b*c + a*B*c - b*c*C - a*A*d + b*B*d + a*C*d)*
Tan[e + f*x] + b*C*(c^2 + d^2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c
, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] && LtQ[n,
 -1]
```

Rubi steps

$$\text{integral} = \frac{(bc-ad)\,(c^2C-Bcd+Ad^2)}{2d^2\,(c^2+d^2)\,f(c+d\tan(e+fx))^2} + \frac{\int \frac{ad(Ac-cC+Bd)+b(c^2C-Bcd+Ad^2)+d(Abc+aBc-bcC-aAd+bBd+aCd)\tan(e+fx)+bC(c^2+d^2)\tan^2(e+fx)}{(c+d\tan(e+fx))^2}\,dx}{d\,(c^2+d^2)}$$

$$= \frac{(bc-ad)\,(c^2C-Bcd+Ad^2)}{2d^2\,(c^2+d^2)\,f(c+d\tan(e+fx))^2} - \frac{b(c^4C-c^2(A-3C)d^2-2Bcd^3+Ad^4)+ad^2(2c(A-C)d-B(c^2-d^2))}{d^2\,(c^2+d^2)^2\,f(c+d\tan(e+fx))} + \frac{\int \frac{-d\left(a\left(c^2C-2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-b\left(2c(A-C)d-B\left(c^2-d^2\right)\right)\right)-d\left(2aAcd-2acCd-Ab\left(c^2-d^2\right)-aB\left(c^2-d^2\right)+b\left(c^2C-2Bcd-C\right.\right.}{c+d\tan(e+fx)}}{d\,(c^2+d^2)^2}$$

$$= \frac{(b(A-C)d(3c^2-d^2)-bB(c^3-3cd^2)-a(c^3C-3Bc^2d-3cCd^2+Bd^3-A(c^3-3cd^2)))\,x}{(c^2+d^2)^3} + \frac{(bc-ad)\,(c^2C-Bcd+Ad^2)}{2d^2\,(c^2+d^2)\,f(c+d\tan(e+fx))^2} - \frac{b(c^4C-c^2(A-3C)d^2-2Bcd^3+Ad^4)+ad^2(2c(A-C)d-B(c^2-d^2))}{d^2\,(c^2+d^2)^2\,f(c+d\tan(e+fx))} + \frac{(aAd(3c^2-d^2)-Ab(c^3-3cd^2)+b(c^3C-3Bc^2d-3cCd^2+Bd^3)-a(Bc^3+3c^2Cd-3Bcd^2-}{(c^2+d^2)^3}$$

$$= \frac{(b(A-C)d(3c^2-d^2)-bB(c^3-3cd^2)-a(c^3C-3Bc^2d-3cCd^2+Bd^3-A(c^3-3cd^2)))\,x}{(c^2+d^2)^3}$$
$$+ \frac{(aAd(3c^2-d^2)-Ab(c^3-3cd^2)+b(c^3C-3Bc^2d-3cCd^2+Bd^3)-a(Bc^3+3c^2Cd-3Bcd^2}{(c^2+d^2)^3 f}$$
$$+ \frac{(bc-ad)\,(c^2C-Bcd+Ad^2)}{2d^2\,(c^2+d^2)\,f(c+d\tan(e+fx))^2}$$
$$- \frac{b(c^4C-c^2(A-3C)d^2-2Bcd^3+Ad^4)+ad^2(2c(A-C)d-B(c^2-d^2))}{d^2\,(c^2+d^2)^2\,f(c+d\tan(e+fx))}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 6.40 (sec) , antiderivative size = 378, normalized size of antiderivative = 1.07

$$\int \frac{(a+b\tan(e+fx))\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^3}\,dx = -\frac{C(a+b\tan(e+fx))}{df(c+d\tan(e+fx))^2}$$
$$-\frac{\frac{bcC+bBd-aCd}{2df(c+d\tan(e+fx))^2}+\frac{\frac{\left(2c(Ab+aB-bC)d^2+2(bB-a(A-C))d^3\right)\left(-\frac{\log(i-\tan(e+fx))}{2(ic-d)^3}+\frac{\log(i+\tan(e+fx))}{2(ic+d)^3}+\frac{d\left(3c^2-d^2\right)\log(c+d\tan(e+fx))}{\left(c^2+d^2\right)^3}-\frac{}{2\left(c^2+d^2\right.}}{d}}{}}{}$$

```
[In] Integrate[((a + b*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c
 + d*Tan[e + f*x])^3,x]
```

```
[Out] -((C*(a + b*Tan[e + f*x]))/(d*f*(c + d*Tan[e + f*x])^2)) - ((b*c*C + b*B*d
- a*C*d)/(2*d*f*(c + d*Tan[e + f*x])^2) + (((2*c*(A*b + a*B - b*C)*d^2 + 2*
(b*B - a*(A - C))*d^3)*(-1/2*Log[I - Tan[e + f*x]]/(I*c - d)^3 + Log[I + Ta
n[e + f*x]]/(2*(I*c + d)^3) + (d*(3*c^2 - d^2)*Log[c + d*Tan[e + f*x]])/(c^
2 + d^2)^3 - d/(2*(c^2 + d^2)*(c + d*Tan[e + f*x])^2) - (2*c*d)/((c^2 + d^2
)^2*(c + d*Tan[e + f*x]))))/d - 2*(A*b + a*B - b*C)*d*(((-1/2*I)*Log[I - Ta
n[e + f*x]])/(c + I*d)^2 + ((I/2)*Log[I + Tan[e + f*x]])/(c - I*d)^2 + (2*c
*d*Log[c + d*Tan[e + f*x]])/(c^2 + d^2)^2 - d/((c^2 + d^2)*(c + d*Tan[e + f
*x]))))/(2*d*f))/d
```

Maple [A] (verified)

Time = 0.17 (sec) , antiderivative size = 493, normalized size of antiderivative = 1.40

method	result
derivativedivides	$\frac{\frac{\left(-3Aa\,c^2d+Aa\,d^3+Ab\,c^3-3Abc\,d^2+Ba\,c^3-3Bac\,d^2+3Bb\,c^2d-Bb\,d^3+3Ca\,c^2d-Ca\,d^3-Cb\,c^3+3Cbc\,d^2\right)\ln\left(1+\tan(fx+e)^2\right)}{2}+\left(Aa\,c^3\right.}{\left(c^2+d^2\right)^3}$
default	$\frac{\frac{\left(-3Aa\,c^2d+Aa\,d^3+Ab\,c^3-3Abc\,d^2+Ba\,c^3-3Bac\,d^2+3Bb\,c^2d-Bb\,d^3+3Ca\,c^2d-Ca\,d^3-Cb\,c^3+3Cbc\,d^2\right)\ln\left(1+\tan(fx+e)^2\right)}{2}+\left(Aa\,c^3\right.}{\left(c^2+d^2\right)^3}$
norman	$\frac{\left(Aa\,c^3-3Aac\,d^2+3Ab\,c^2d-Ab\,d^3+3Ba\,c^2d-Ba\,d^3-Bb\,c^3+3Bbc\,d^2-Ca\,c^3+3Cac\,d^2-3Cb\,c^2d+Cb\,d^3\right)c^2x}{\left(c^4+2c^2d^2+d^4\right)\left(c^2+d^2\right)}+\frac{d^2\left(Aa\,c^3-3Aac\,d^2+3Ab\right.}{}$
risch	Expression too large to display
parallelrisch	Expression too large to display

```
[In] int((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^3,x,m
ethod=_RETURNVERBOSE)
```

```
[Out] 1/f*(1/(c^2+d^2)^3*(1/2*(-3*A*a*c^2*d+A*a*d^3+A*b*c^3-3*A*b*c*d^2+B*a*c^3-3
*B*a*c*d^2+3*B*b*c^2*d-B*b*d^3+3*C*a*c^2*d-C*a*d^3-C*b*c^3+3*C*b*c*d^2)*ln(
1+tan(f*x+e)^2)+(A*a*c^3-3*A*a*c*d^2+3*A*b*c^2*d-A*b*d^3+3*B*a*c^2*d-B*a*d^
3-B*b*c^3+3*B*b*c*d^2-C*a*c^3+3*C*a*c*d^2-3*C*b*c^2*d+C*b*d^3)*arctan(tan(f
*x+e)))+(3*A*a*c^2*d-A*a*d^3-A*b*c^3+3*A*b*c*d^2-B*a*c^3+3*B*a*c*d^2-3*B*b*
c^2*d+B*b*d^3-3*C*a*c^2*d+C*a*d^3+C*b*c^3-3*C*b*c*d^2)/(c^2+d^2)^3*ln(c+d*t
an(f*x+e))-1/2*(A*a*d^3-A*b*c*d^2-B*a*c*d^2+B*b*c^2*d+C*a*c^2*d-C*b*c^3)/d^
2/(c^2+d^2)/(c+d*tan(f*x+e))^2-(2*A*a*c*d^3-A*b*c^2*d^2+A*b*d^4-B*a*c^2*d^2
+B*a*d^4-2*B*b*c*d^3-2*C*a*c*d^3+C*b*c^4+3*C*b*c^2*d^2)/(c^2+d^2)^2/d^2/(c+
d*tan(f*x+e)))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 897 vs. 2(350) = 700.

Time = 0.29 (sec) , antiderivative size = 897, normalized size of antiderivative = 2.55

$$\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^3}\,dx$$
$$=\frac{Cbc^5-Aad^5-3\,(Ca+Bb)c^4d+5\,(Ba+(A-C)b)c^3d^2-((7\,A-3\,C)a-3\,Bb)c^2d^3-(Ba+Ab)cd^4+}{}$$

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^3,x, algorithm="fricas")
```

```
[Out] 1/2*(C*b*c^5 - A*a*d^5 - 3*(C*a + B*b)*c^4*d + 5*(B*a + (A - C)*b)*c^3*d^2
- ((7*A - 3*C)*a - 3*B*b)*c^2*d^3 - (B*a + A*b)*c*d^4 + 2*(((A - C)*a - B*b
)*c^5 + 3*(B*a + (A - C)*b)*c^4*d - 3*((A - C)*a - B*b)*c^3*d^2 - (B*a + (A
 - C)*b)*c^2*d^3)*f*x + (C*b*c^5 - A*a*d^5 + (C*a + B*b)*c^4*d - (3*B*a + (
3*A - 7*C)*b)*c^3*d^2 + 5*((A - C)*a - B*b)*c^2*d^3 + 3*(B*a + A*b)*c*d^4 +
 2*(((A - C)*a - B*b)*c^3*d^2 + 3*(B*a + (A - C)*b)*c^2*d^3 - 3*((A - C)*a
- B*b)*c*d^4 - (B*a + (A - C)*b)*d^5)*f*x)*tan(f*x + e)^2 - ((B*a + (A - C)
*b)*c^5 - 3*((A - C)*a - B*b)*c^4*d - 3*(B*a + (A - C)*b)*c^3*d^2 + ((A - C
)*a - B*b)*c^2*d^3 + ((B*a + (A - C)*b)*c^3*d^2 - 3*((A - C)*a - B*b)*c^2*d
^3 - 3*(B*a + (A - C)*b)*c*d^4 + ((A - C)*a - B*b)*d^5)*tan(f*x + e)^2 + 2*
((B*a + (A - C)*b)*c^4*d - 3*((A - C)*a - B*b)*c^3*d^2 - 3*(B*a + (A - C)*b
)*c^2*d^3 + ((A - C)*a - B*b)*c*d^4)*tan(f*x + e))*log((d^2*tan(f*x + e)^2
+ 2*c*d*tan(f*x + e) + c^2)/(tan(f*x + e)^2 + 1)) + 2*((C*a + B*b)*c^5 - (2
*B*a + (2*A - 3*C)*b)*c^4*d + 3*((A - C)*a - B*b)*c^3*d^2 + 3*(B*a + (A - C
)*b)*c^2*d^3 - ((3*A - 2*C)*a - 2*B*b)*c*d^4 - (B*a + A*b)*d^5 + 2*(((A - C
)*a - B*b)*c^4*d + 3*(B*a + (A - C)*b)*c^3*d^2 - 3*((A - C)*a - B*b)*c^2*d^
3 - (B*a + (A - C)*b)*c*d^4)*f*x)*tan(f*x + e))/((c^6*d^2 + 3*c^4*d^4 + 3*c
^2*d^6 + d^8)*f*tan(f*x + e)^2 + 2*(c^7*d + 3*c^5*d^3 + 3*c^3*d^5 + c*d^7)*
f*tan(f*x + e) + (c^8 + 3*c^6*d^2 + 3*c^4*d^4 + c^2*d^6)*f)
```

Sympy [F(-2)]

Exception generated.

$$\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^3}\,dx$$
$$= \text{Exception raised: AttributeError}$$

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x+e)
)**3,x)
```

```
[Out] Exception raised: AttributeError >> 'NoneType' object has no attribute 'pri
mitive'
```

Maxima [A] (verification not implemented)

none

Time = 0.34 (sec) , antiderivative size = 543, normalized size of antiderivative = 1.54

$$\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^3}\,dx$$
$$= \frac{\frac{2\left(((A-C)a-Bb)c^3+3(Ba+(A-C)b)c^2d-3((A-C)a-Bb)cd^2-(Ba+(A-C)b)d^3\right)(fx+e)}{c^6+3c^4d^2+3c^2d^4+d^6} - \frac{2\left((Ba+(A-C)b)c^3-3((A-C)a-Bb)c^2d-3(B\cdots\right.}{c^6+3c^4\cdots}}{\cdots}$$

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^3,x, algorithm="maxima")
```

```
[Out] 1/2*(2*(((A - C)*a - B*b)*c^3 + 3*(B*a + (A - C)*b)*c^2*d - 3*((A - C)*a -
B*b)*c*d^2 - (B*a + (A - C)*b)*d^3)*(f*x + e)/(c^6 + 3*c^4*d^2 + 3*c^2*d^4
+ d^6) - 2*((B*a + (A - C)*b)*c^3 - 3*((A - C)*a - B*b)*c^2*d - 3*(B*a + (A
 - C)*b)*c*d^2 + ((A - C)*a - B*b)*d^3)*log(d*tan(f*x + e) + c)/(c^6 + 3*c^
4*d^2 + 3*c^2*d^4 + d^6) + ((B*a + (A - C)*b)*c^3 - 3*((A - C)*a - B*b)*c^2
*d - 3*(B*a + (A - C)*b)*c*d^2 + ((A - C)*a - B*b)*d^3)*log(tan(f*x + e)^2
+ 1)/(c^6 + 3*c^4*d^2 + 3*c^2*d^4 + d^6) - (C*b*c^5 + A*a*d^5 + (C*a + B*b)
*c^4*d - (3*B*a + (3*A - 5*C)*b)*c^3*d^2 + ((5*A - 3*C)*a - 3*B*b)*c^2*d^3
+ (B*a + A*b)*c*d^4 + 2*(C*b*c^4*d - (B*a + (A - 3*C)*b)*c^2*d^3 + 2*((A -
C)*a - B*b)*c*d^4 + (B*a + A*b)*d^5)*tan(f*x + e))/(c^6*d^2 + 2*c^4*d^4 + c
^2*d^6 + (c^4*d^4 + 2*c^2*d^6 + d^8)*tan(f*x + e)^2 + 2*(c^5*d^3 + 2*c^3*d^
5 + c*d^7)*tan(f*x + e)))/f
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1006 vs. $2(350) = 700$.

Time = 0.81 (sec) , antiderivative size = 1006, normalized size of antiderivative = 2.86

$$\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^3}\,dx$$
$$= \frac{\frac{2\left(Aac^3-Cac^3-Bbc^3+3\,Bac^2d+3\,Abc^2d-3\,Cbc^2d-3\,Aacd^2+3\,Cacd^2+3\,Bbcd^2-Bad^3-Abd^3+Cbd^3\right)(fx+e)}{c^6+3\,c^4d^2+3\,c^2d^4+d^6}+\frac{(Bac^3+Abc^3-Cbc^3-3\,Aac^2d+}{}}{}$$

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^3,x, algorithm="giac")
```

```
[Out] 1/2*(2*(A*a*c^3 - C*a*c^3 - B*b*c^3 + 3*B*a*c^2*d + 3*A*b*c^2*d - 3*C*b*c^2
*d - 3*A*a*c*d^2 + 3*C*a*c*d^2 + 3*B*b*c*d^2 - B*a*d^3 - A*b*d^3 + C*b*d^3)
*(f*x + e)/(c^6 + 3*c^4*d^2 + 3*c^2*d^4 + d^6) + (B*a*c^3 + A*b*c^3 - C*b*c
^3 - 3*A*a*c^2*d + 3*C*a*c^2*d + 3*B*b*c^2*d - 3*B*a*c*d^2 - 3*A*b*c*d^2 +
3*C*b*c*d^2 + A*a*d^3 - C*a*d^3 - B*b*d^3)*log(tan(f*x + e)^2 + 1)/(c^6 + 3
*c^4*d^2 + 3*c^2*d^4 + d^6) - 2*(B*a*c^3*d + A*b*c^3*d - C*b*c^3*d - 3*A*a*
c^2*d^2 + 3*C*a*c^2*d^2 + 3*B*b*c^2*d^2 - 3*B*a*c*d^3 - 3*A*b*c*d^3 + 3*C*b
*c*d^3 + A*a*d^4 - C*a*d^4 - B*b*d^4)*log(abs(d*tan(f*x + e) + c))/(c^6*d +
 3*c^4*d^3 + 3*c^2*d^5 + d^7) + (3*B*a*c^3*d^4*tan(f*x + e)^2 + 3*A*b*c^3*d
^4*tan(f*x + e)^2 - 3*C*b*c^3*d^4*tan(f*x + e)^2 - 9*A*a*c^2*d^5*tan(f*x +
e)^2 + 9*C*a*c^2*d^5*tan(f*x + e)^2 + 9*B*b*c^2*d^5*tan(f*x + e)^2 - 9*B*a*
c*d^6*tan(f*x + e)^2 - 9*A*b*c*d^6*tan(f*x + e)^2 + 9*C*b*c*d^6*tan(f*x + e
)^2 + 3*A*a*d^7*tan(f*x + e)^2 - 3*C*a*d^7*tan(f*x + e)^2 - 3*B*b*d^7*tan(f
*x + e)^2 - 2*C*b*c^6*d*tan(f*x + e) + 8*B*a*c^4*d^3*tan(f*x + e) + 8*A*b*c
^4*d^3*tan(f*x + e) - 14*C*b*c^4*d^3*tan(f*x + e) - 22*A*a*c^3*d^4*tan(f*x
```

```
+ e) + 22*C*a*c^3*d^4*tan(f*x + e) + 22*B*b*c^3*d^4*tan(f*x + e) - 18*B*a*c
^2*d^5*tan(f*x + e) - 18*A*b*c^2*d^5*tan(f*x + e) + 12*C*b*c^2*d^5*tan(f*x
+ e) + 2*A*a*c*d^6*tan(f*x + e) - 2*C*a*c*d^6*tan(f*x + e) - 2*B*b*c*d^6*ta
n(f*x + e) - 2*B*a*d^7*tan(f*x + e) - 2*A*b*d^7*tan(f*x + e) - C*b*c^7 - C*
a*c^6*d - B*b*c^6*d + 6*B*a*c^5*d^2 + 6*A*b*c^5*d^2 - 9*C*b*c^5*d^2 - 14*A*
a*c^4*d^3 + 11*C*a*c^4*d^3 + 11*B*b*c^4*d^3 - 7*B*a*c^3*d^4 - 7*A*b*c^3*d^4
 + 4*C*b*c^3*d^4 - 3*A*a*c^2*d^5 - B*a*c*d^6 - A*b*c*d^6 - A*a*d^7)/((c^6*d
^2 + 3*c^4*d^4 + 3*c^2*d^6 + d^8)*(d*tan(f*x + e) + c)^2))/f
```

Mupad [B] (verification not implemented)

Time = 15.42 (sec) , antiderivative size = 502, normalized size of antiderivative = 1.43

$$\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^3}\,dx =$$

$$-\frac{\frac{A\,a\,d^5+C\,b\,c^5+A\,b\,c\,d^4+B\,a\,c\,d^4+B\,b\,c^4\,d+C\,a\,c^4\,d+5\,A\,a\,c^2\,d^3-3\,A\,b\,c^3\,d^2-3\,B\,a\,c^3\,d^2-3\,B\,b\,c^2\,d^3-3\,C\,a\,c^2\,d^3+5\,C\,b\,c^3\,d^2}{2\,d^2\,\left(c^4+2\,c^2\,d^2+d^4\right)}+\frac{\tan(e+f\,x)\,(\ldots}{\ldots}}{f\,\left(c^2+2\,c\,d\tan\left(e+f\,x\right)+d^2\tan\left(e+f\ldots\right.\right.}$$

$$-\frac{\ln\left(\tan(e+f\,x)+1\mathrm{i}\right)\,\left(B\,b+A\,b\,1\mathrm{i}+B\,a\,1\mathrm{i}-A\,a+C\,a-C\,b\,1\mathrm{i}\right)}{2\,f\,\left(-c^3\,1\mathrm{i}-3\,c^2\,d+c\,d^2\,3\mathrm{i}+d^3\right)}$$

$$-\frac{\ln\left(\tan(e+f\,x)-\mathrm{i}\right)\,\left(A\,b+B\,a-C\,b-A\,a\,1\mathrm{i}+B\,b\,1\mathrm{i}+C\,a\,1\mathrm{i}\right)}{2\,f\,\left(-c^3-c^2\,d\,3\mathrm{i}+3\,c\,d^2+d^3\,1\mathrm{i}\right)}$$

$$-\frac{\ln\left(c+d\tan(e+f\,x)\right)\,\left(\left(A\,b+B\,a-C\,b\right)\,c^3+\left(3\,B\,b-3\,A\,a+3\,C\,a\right)\,c^2\,d+\left(3\,C\,b-3\,B\,a-3\,A\,b\right)\,c\,d\ldots\right.}{f\,\left(c^6+3\,c^4\,d^2+3\,c^2\,d^4+d^6\right)}$$

```
[In] int(((a + b*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d*t
an(e + f*x))^3,x)
```

```
[Out] - ((A*a*d^5 + C*b*c^5 + A*b*c*d^4 + B*a*c*d^4 + B*b*c^4*d + C*a*c^4*d + 5*A
*a*c^2*d^3 - 3*A*b*c^3*d^2 - 3*B*a*c^3*d^2 - 3*B*b*c^2*d^3 - 3*C*a*c^2*d^3
+ 5*C*b*c^3*d^2)/(2*d^2*(c^4 + d^4 + 2*c^2*d^2)) + (tan(e + f*x)*(A*b*d^4 +
 B*a*d^4 + C*b*c^4 + 2*A*a*c*d^3 - 2*B*b*c*d^3 - 2*C*a*c*d^3 - A*b*c^2*d^2
- B*a*c^2*d^2 + 3*C*b*c^2*d^2))/(d*(c^4 + d^4 + 2*c^2*d^2)))/(f*(c^2 + d^2*
tan(e + f*x)^2 + 2*c*d*tan(e + f*x))) - (log(tan(e + f*x) + 1i)*(A*b*1i - A
*a + B*a*1i + B*b + C*a - C*b*1i))/(2*f*(c*d^2*3i - 3*c^2*d - c^3*1i + d^3)
) - (log(tan(e + f*x) - 1i)*(A*b - A*a*1i + B*a + B*b*1i + C*a*1i - C*b))/(
2*f*(3*c*d^2 - c^2*d*3i - c^3 + d^3*1i)) - (log(c + d*tan(e + f*x))*(c^3*(A
*b + B*a - C*b) - d^3*(B*b - A*a + C*a) + c^2*d*(3*B*b - 3*A*a + 3*C*a) - c
*d^2*(3*A*b + 3*B*a - 3*C*b)))/(f*(c^6 + d^6 + 3*c^2*d^4 + 3*c^4*d^2))
```

3.87 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^3}\,dx$

Optimal result 988
Rubi [A] (verified) 988
Mathematica [C] (verified) 990
Maple [A] (verified) 991
Fricas [B] (verification not implemented) 991
Sympy [F(-2)] 992
Maxima [A] (verification not implemented) 992
Giac [B] (verification not implemented) 993
Mupad [B] (verification not implemented) 994

Optimal result

Integrand size = 33, antiderivative size = 209

$$\begin{aligned}
&\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^3}\,dx \\
&= -\frac{(c^3C-3Bc^2d-3cCd^2+Bd^3-A(c^3-3cd^2))\,x}{(c^2+d^2)^3} \\
&\quad+\frac{((A-C)d(3c^2-d^2)-B(c^3-3cd^2))\log(c\cos(e+fx)+d\sin(e+fx))}{(c^2+d^2)^3 f} \\
&\quad-\frac{c^2C-Bcd+Ad^2}{2d\,(c^2+d^2)\,f(c+d\tan(e+fx))^2}-\frac{2c(A-C)d-B(c^2-d^2)}{(c^2+d^2)^2\,f(c+d\tan(e+fx))}
\end{aligned}$$

```
[Out] -(c^3*C-3*B*c^2*d-3*C*c*d^2+B*d^3-A*(c^3-3*c*d^2))*x/(c^2+d^2)^3+((A-C)*d*(
3*c^2-d^2)-B*(c^3-3*c*d^2))*ln(c*cos(f*x+e)+d*sin(f*x+e))/(c^2+d^2)^3/f+1/2
*(-A*d^2+B*c*d-C*c^2)/d/(c^2+d^2)/f/(c+d*tan(f*x+e))^2+(-2*c*(A-C)*d+B*(c^2
-d^2))/(c^2+d^2)^2/f/(c+d*tan(f*x+e))
```

Rubi [A] (verified)

Time = 0.45 (sec) , antiderivative size = 209, normalized size of antiderivative = 1.00, number of steps used = 4, number of rules used = 4, $\frac{\text{number of rules}}{\text{integrand size}} = 0.121$, Rules used

$= \{3709, 3610, 3612, 3611\}$

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^3}\,dx$$
$$= -\frac{Ad^2-Bcd+c^2C}{2df\,(c^2+d^2)\,(c+d\tan(e+fx))^2} - \frac{2cd(A-C)-B(c^2-d^2)}{f\,(c^2+d^2)^2\,(c+d\tan(e+fx))}$$
$$+ \frac{(d(A-C)\,(3c^2-d^2)-B(c^3-3cd^2))\log(c\cos(e+fx)+d\sin(e+fx))}{f\,(c^2+d^2)^3}$$
$$- \frac{x(-A(c^3-3cd^2)-3Bc^2d+Bd^3+c^3C-3cCd^2)}{(c^2+d^2)^3}$$

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/(c + d*Tan[e + f*x])^3,x]
```

```
[Out] -((((c^3*C - 3*B*c^2*d - 3*c*C*d^2 + B*d^3 - A*(c^3 - 3*c*d^2))*x)/(c^2 + d^
2)^3) + (((A - C)*d*(3*c^2 - d^2) - B*(c^3 - 3*c*d^2))*Log[c*Cos[e + f*x] +
 d*Sin[e + f*x]])/((c^2 + d^2)^3*f) - (c^2*C - B*c*d + A*d^2)/(2*d*(c^2 + d
^2)*f*(c + d*Tan[e + f*x])^2) - (2*c*(A - C)*d - B*(c^2 - d^2))/((c^2 + d^2
)^2*f*(c + d*Tan[e + f*x]))
```

Rule 3610

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(b*c - a*d)*((a + b*Tan[e + f*x])^(m + 1)/
(f*(m + 1)*(a^2 + b^2))), x] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])
^(m + 1)*Simp[a*c + b*d - (b*c - a*d)*Tan[e + f*x], x], x], x] /; FreeQ[{a,
 b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && LtQ[m, -1]
```

Rule 3611

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c/(b*f))*Log[RemoveContent[a*Cos[e + f*x] + b*Si
n[e + f*x], x]], x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3612

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[(a*c + b*d)*(x/(a^2 + b^2)), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3709

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 -
```

```
 a*b*B + a^2*C)*((a + b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1)*(a^2 + b^2))), x
] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[b*B + a*(A -
C) - (A*b - a*B - b*C)*Tan[e + f*x], x], x], x] /; FreeQ[{a, b, e, f, A, B,
 C}, x] && NeQ[A*b^2 - a*b*B + a^2*C, 0] && LtQ[m, -1] && NeQ[a^2 + b^2, 0]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= -\frac{c^2C-Bcd+Ad^2}{2d\left(c^2+d^2\right)f(c+d\tan(e+fx))^2}+\frac{\int\frac{Ac-cC+Bd+(Bc-(A-C)d)\tan(e+fx)}{(c+d\tan(e+fx))^2}\,dx}{c^2+d^2}\\
&= -\frac{c^2C-Bcd+Ad^2}{2d\left(c^2+d^2\right)f(c+d\tan(e+fx))^2}-\frac{2c(A-C)d-B(c^2-d^2)}{\left(c^2+d^2\right)^2f(c+d\tan(e+fx))}\\
&\quad+\frac{\int\frac{-c^2C+2Bcd+Cd^2+A(c^2-d^2)-(2c(A-C)d-B(c^2-d^2))\tan(e+fx)}{c+d\tan(e+fx)}\,dx}{\left(c^2+d^2\right)^2}\\
&= \frac{\left(Ac^3-c^3C+3Bc^2d-3Acd^2+3cCd^2-Bd^3\right)x}{\left(c^2+d^2\right)^3}-\frac{c^2C-Bcd+Ad^2}{2d\left(c^2+d^2\right)f(c+d\tan(e+fx))^2}\\
&\quad-\frac{2c(A-C)d-B(c^2-d^2)}{\left(c^2+d^2\right)^2f(c+d\tan(e+fx))}+\frac{\left((A-C)d(3c^2-d^2)-B(c^3-3cd^2)\right)\int\frac{d-c\tan(e+fx)}{c+d\tan(e+fx)}\,dx}{\left(c^2+d^2\right)^3}\\
&= \frac{\left(Ac^3-c^3C+3Bc^2d-3Acd^2+3cCd^2-Bd^3\right)x}{\left(c^2+d^2\right)^3}\\
&\quad+\frac{\left((A-C)d(3c^2-d^2)-B(c^3-3cd^2)\right)\log(c\cos(e+fx)+d\sin(e+fx))}{\left(c^2+d^2\right)^3f}\\
&\quad-\frac{c^2C-Bcd+Ad^2}{2d\left(c^2+d^2\right)f(c+d\tan(e+fx))^2}-\frac{2c(A-C)d-B(c^2-d^2)}{\left(c^2+d^2\right)^2f(c+d\tan(e+fx))}
\end{aligned}$$

Mathematica [C] (verified)

Result contains complex when optimal does not.

Time = 5.52 (sec) , antiderivative size = 261, normalized size of antiderivative = 1.25

$$\int\frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^3}\,dx=$$

$$-\frac{\frac{C}{(c+d\tan(e+fx))^2}+B\left(\frac{i\log(i-\tan(e+fx))}{(c+id)^2}-\frac{i\log(i+\tan(e+fx))}{(c-id)^2}+\frac{2d\left(-2c\log(c+d\tan(e+fx))+\frac{c^2+d^2}{c+d\tan(e+fx)}\right)}{(c^2+d^2)^2}\right)-(Bc+(-}{2df}$$

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/(c + d*Tan[e + f*x])^3,x]
```

```
[Out] -1/2*(C/(c + d*Tan[e + f*x])^2 + B*((I*Log[I - Tan[e + f*x]])/(c + I*d)^2 -
 (I*Log[I + Tan[e + f*x]])/(c - I*d)^2 + (2*d*(-2*c*Log[c + d*Tan[e + f*x]]
 + (c^2 + d^2)/(c + d*Tan[e + f*x])))/(c^2 + d^2)^2) - (B*c + (-A + C)*d)*(
(I*Log[I - Tan[e + f*x]])/(c + I*d)^3 - Log[I + Tan[e + f*x]]/(I*c + d)^3 +
 (d*((-6*c^2 + 2*d^2)*Log[c + d*Tan[e + f*x]] + ((c^2 + d^2)*(5*c^2 + d^2 +
 4*c*d*Tan[e + f*x]))/(c + d*Tan[e + f*x])^2))/(c^2 + d^2)^3))/(d*f)
```

Maple [A] (verified)

Time = 0.14 (sec) , antiderivative size = 262, normalized size of antiderivative = 1.25

method	result
derivativedivides	$\frac{\frac{\frac{\left(-3A c^2 d+A d^3+B c^3-3Bc d^2+3C c^2 d-C d^3\right)\ln\left(1+\tan(fx+e)^2\right)}{2}+\left(A c^3-3Ac d^2+3B c^2 d-B d^3-c^3 C+3Cc d^2\right)\arctan(\tan(fx+e)}{\left(c^2+d^2\right)^3}}{f}$
default	$\frac{\frac{\frac{\left(-3A c^2 d+A d^3+B c^3-3Bc d^2+3C c^2 d-C d^3\right)\ln\left(1+\tan(fx+e)^2\right)}{2}+\left(A c^3-3Ac d^2+3B c^2 d-B d^3-c^3 C+3Cc d^2\right)\arctan(\tan(fx+e)}{\left(c^2+d^2\right)^3}}{f}$
norman	$\frac{\frac{\left(A c^3-3Ac d^2+3B c^2 d-B d^3-c^3 C+3Cc d^2\right)c^2 x}{\left(c^4+2c^2 d^2+d^4\right)\left(c^2+d^2\right)}+\frac{d^2\left(A c^3-3Ac d^2+3B c^2 d-B d^3-c^3 C+3Cc d^2\right)x\tan(fx+e)^2}{\left(c^4+2c^2 d^2+d^4\right)\left(c^2+d^2\right)}-\frac{5A c^2 d^3+A d^5-3}{2f c}}{(c+d}$
risch	Expression too large to display
parallelrisch	Expression too large to display

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^3,x,method=_RETURNVERB
OSE)
```

```
[Out] 1/f*(1/(c^2+d^2)^3*(1/2*(-3*A*c^2*d+A*d^3+B*c^3-3*B*c*d^2+3*C*c^2*d-C*d^3)*
ln(1+tan(f*x+e)^2)+(A*c^3-3*A*c*d^2+3*B*c^2*d-B*d^3-C*c^3+3*C*c*d^2)*arctan
(tan(f*x+e)))-1/2*(A*d^2-B*c*d+C*c^2)/(c^2+d^2)/d/(c+d*tan(f*x+e))^2-(2*A*c
*d-B*c^2+B*d^2-2*C*c*d)/(c^2+d^2)^2/(c+d*tan(f*x+e))+(3*A*c^2*d-A*d^3-B*c^3
+3*B*c*d^2-3*C*c^2*d+C*d^3)/(c^2+d^2)^3*ln(c+d*tan(f*x+e)))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 566 vs. 2(207) = 414.

Time = 0.30 (sec) , antiderivative size = 566, normalized size of antiderivative = 2.71

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^3}\,dx =$$

$$-\frac{3\,Cc^4d-5\,Bc^3d^2+(7\,A-3\,C)c^2d^3+Bcd^4+Ad^5-2\,((A-C)c^5+3\,Bc^4d-3\,(A-C)c^3d^2-Bc^2d^3}{}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^3,x, algorithm="
fricas")
```

```
[Out] -1/2*(3*C*c^4*d - 5*B*c^3*d^2 + (7*A - 3*C)*c^2*d^3 + B*c*d^4 + A*d^5 - 2*(
(A - C)*c^5 + 3*B*c^4*d - 3*(A - C)*c^3*d^2 - B*c^2*d^3)*f*x - (C*c^4*d - 3
*B*c^3*d^2 + 5*(A - C)*c^2*d^3 + 3*B*c*d^4 - A*d^5 + 2*((A - C)*c^3*d^2 + 3
*B*c^2*d^3 - 3*(A - C)*c*d^4 - B*d^5)*f*x)*tan(f*x + e)^2 + (B*c^5 - 3*(A -
 C)*c^4*d - 3*B*c^3*d^2 + (A - C)*c^2*d^3 + (B*c^3*d^2 - 3*(A - C)*c^2*d^3
- 3*B*c*d^4 + (A - C)*d^5)*tan(f*x + e)^2 + 2*(B*c^4*d - 3*(A - C)*c^3*d^2
- 3*B*c^2*d^3 + (A - C)*c*d^4)*tan(f*x + e))*log((d^2*tan(f*x + e)^2 + 2*c*
d*tan(f*x + e) + c^2)/(tan(f*x + e)^2 + 1)) - 2*(C*c^5 - 2*B*c^4*d + 3*(A -
 C)*c^3*d^2 + 3*B*c^2*d^3 - (3*A - 2*C)*c*d^4 - B*d^5 + 2*((A - C)*c^4*d +
3*B*c^3*d^2 - 3*(A - C)*c^2*d^3 - B*c*d^4)*f*x)*tan(f*x + e))/((c^6*d^2 + 3
*c^4*d^4 + 3*c^2*d^6 + d^8)*f*tan(f*x + e)^2 + 2*(c^7*d + 3*c^5*d^3 + 3*c^3
*d^5 + c*d^7)*f*tan(f*x + e) + (c^8 + 3*c^6*d^2 + 3*c^4*d^4 + c^2*d^6)*f)
```

Sympy [F(-2)]

Exception generated.

$$\int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{(c + d\tan(e + fx))^3}\,dx = \text{Exception raised: AttributeError}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x+e))**3,x)
```

```
[Out] Exception raised: AttributeError >> 'NoneType' object has no attribute 'pri
mitive'
```

Maxima [A] (verification not implemented)

none

Time = 0.33 (sec) , antiderivative size = 367, normalized size of antiderivative = 1.76

$$\int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{(c + d\tan(e + fx))^3}\,dx = \frac{\frac{2\,((A-C)c^3+3\,Bc^2d-3\,(A-C)cd^2-Bd^3)(fx+e)}{c^6+3\,c^4d^2+3\,c^2d^4+d^6} - \frac{2\,(Bc^3-3\,(A-C)c^2d-3\,Bcd^2+(A-C)d^3)\log(d\tan(fx+e)+c)}{c^6+3\,c^4d^2+3\,c^2d^4+d^6} + \frac{(Bc^3-3\,(A-C)c^2d-3\,Bc}{c^6+3}}{2\,f}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^3,x, algorithm="
maxima")
```

```
[Out] 1/2*(2*((A - C)*c^3 + 3*B*c^2*d - 3*(A - C)*c*d^2 - B*d^3)*(f*x + e)/(c^6 +
 3*c^4*d^2 + 3*c^2*d^4 + d^6) - 2*(B*c^3 - 3*(A - C)*c^2*d - 3*B*c*d^2 + (A
 - C)*d^3)*log(d*tan(f*x + e) + c)/(c^6 + 3*c^4*d^2 + 3*c^2*d^4 + d^6) + (B
```

```
*c^3 - 3*(A - C)*c^2*d - 3*B*c*d^2 + (A - C)*d^3)*log(tan(f*x + e)^2 + 1)/(
c^6 + 3*c^4*d^2 + 3*c^2*d^4 + d^6) - (C*c^4 - 3*B*c^3*d + (5*A - 3*C)*c^2*d
^2 + B*c*d^3 + A*d^4 - 2*(B*c^2*d^2 - 2*(A - C)*c*d^3 - B*d^4)*tan(f*x + e)
)/(c^6*d + 2*c^4*d^3 + c^2*d^5 + (c^4*d^3 + 2*c^2*d^5 + d^7)*tan(f*x + e)^2
 + 2*(c^5*d^2 + 2*c^3*d^4 + c*d^6)*tan(f*x + e)))/f
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 531 vs. $2(207) = 414$.

Time = 0.69 (sec) , antiderivative size = 531, normalized size of antiderivative = 2.54

$$\int \frac{A + B\tan(e+fx) + C\tan^2(e+fx)}{(c+d\tan(e+fx))^3}\,dx$$
$$= \frac{\frac{2\,(Ac^3 - Cc^3 + 3\,Bc^2d - 3\,Acd^2 + 3\,Ccd^2 - Bd^3)(fx+e)}{c^6 + 3\,c^4d^2 + 3\,c^2d^4 + d^6} + \frac{(Bc^3 - 3\,Ac^2d + 3\,Cc^2d - 3\,Bcd^2 + Ad^3 - Cd^3)\log\left(\tan(fx+e)^2 + 1\right)}{c^6 + 3\,c^4d^2 + 3\,c^2d^4 + d^6} - \frac{2\,(Bc^3d - 3\,Ac^2d^2}{}}{}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^3,x, algorithm="
giac")
```

```
[Out] 1/2*(2*(A*c^3 - C*c^3 + 3*B*c^2*d - 3*A*c*d^2 + 3*C*c*d^2 - B*d^3)*(f*x + e
)/(c^6 + 3*c^4*d^2 + 3*c^2*d^4 + d^6) + (B*c^3 - 3*A*c^2*d + 3*C*c^2*d - 3*
B*c*d^2 + A*d^3 - C*d^3)*log(tan(f*x + e)^2 + 1)/(c^6 + 3*c^4*d^2 + 3*c^2*d
^4 + d^6) - 2*(B*c^3*d - 3*A*c^2*d^2 + 3*C*c^2*d^2 - 3*B*c*d^3 + A*d^4 - C*
d^4)*log(abs(d*tan(f*x + e) + c))/(c^6*d + 3*c^4*d^3 + 3*c^2*d^5 + d^7) + (
3*B*c^3*d^3*tan(f*x + e)^2 - 9*A*c^2*d^4*tan(f*x + e)^2 + 9*C*c^2*d^4*tan(f
*x + e)^2 - 9*B*c*d^5*tan(f*x + e)^2 + 3*A*d^6*tan(f*x + e)^2 - 3*C*d^6*tan
(f*x + e)^2 + 8*B*c^4*d^2*tan(f*x + e) - 22*A*c^3*d^3*tan(f*x + e) + 22*C*c
^3*d^3*tan(f*x + e) - 18*B*c^2*d^4*tan(f*x + e) + 2*A*c*d^5*tan(f*x + e) -
2*C*c*d^5*tan(f*x + e) - 2*B*d^6*tan(f*x + e) - C*c^6 + 6*B*c^5*d - 14*A*c^
4*d^2 + 11*C*c^4*d^2 - 7*B*c^3*d^3 - 3*A*c^2*d^4 - B*c*d^5 - A*d^6)/((c^6*d
 + 3*c^4*d^3 + 3*c^2*d^5 + d^7)*(d*tan(f*x + e) + c)^2))/f
```

Mupad [B] (verification not implemented)

Time = 10.86 (sec) , antiderivative size = 327, normalized size of antiderivative = 1.56

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^3}\,dx$$
$$= -\frac{\frac{\tan(e+f\,x)\,\left(B\,d^3+2\,A\,c\,d^2-B\,c^2\,d-2\,C\,c\,d^2\right)}{c^4+2\,c^2\,d^2+d^4}+\frac{A\,d^4+C\,c^4+5\,A\,c^2\,d^2-3\,C\,c^2\,d^2+B\,c\,d^3-3\,B\,c^3\,d}{2\,d\,\left(c^4+2\,c^2\,d^2+d^4\right)}}{f\,\left(c^2+2\,c\,d\tan\left(e+f\,x\right)+d^2\tan\left(e+f\,x\right)^2\right)}$$
$$-\frac{\ln\left(\tan(e+f\,x)-\mathrm{i}\right)\,\left(B-A\,1\mathrm{i}+C\,1\mathrm{i}\right)}{2\,f\,\left(-c^3-c^2\,d\,3\mathrm{i}+3\,c\,d^2+d^3\,1\mathrm{i}\right)}$$
$$-\frac{\ln\left(c+d\tan(e+f\,x)\right)\,\left(B\,c^3+\left(3\,C-3\,A\right)\,c^2\,d-3\,B\,c\,d^2+\left(A-C\right)\,d^3\right)}{f\,\left(c^6+3\,c^4\,d^2+3\,c^2\,d^4+d^6\right)}$$
$$-\frac{\ln\left(\tan(e+f\,x)+1\mathrm{i}\right)\,\left(C-A+B\,1\mathrm{i}\right)}{2\,f\,\left(-c^3\,1\mathrm{i}-3\,c^2\,d+c\,d^2\,3\mathrm{i}+d^3\right)}$$

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/(c + d*tan(e + f*x))^3,x)
```

```
[Out] - ((tan(e + f*x)*(B*d^3 + 2*A*c*d^2 - B*c^2*d - 2*C*c*d^2))/(c^4 + d^4 + 2*
c^2*d^2) + (A*d^4 + C*c^4 + 5*A*c^2*d^2 - 3*C*c^2*d^2 + B*c*d^3 - 3*B*c^3*d
)/(2*d*(c^4 + d^4 + 2*c^2*d^2)))/(f*(c^2 + d^2*tan(e + f*x)^2 + 2*c*d*tan(e
 + f*x))) - (log(tan(e + f*x) - 1i)*(B - A*1i + C*1i))/(2*f*(3*c*d^2 - c^2*
d*3i - c^3 + d^3*1i)) - (log(c + d*tan(e + f*x))*(B*c^3 + d^3*(A - C) - c^2
*d*(3*A - 3*C) - 3*B*c*d^2))/(f*(c^6 + d^6 + 3*c^2*d^4 + 3*c^4*d^2)) - (log
(tan(e + f*x) + 1i)*(B*1i - A + C))/(2*f*(c*d^2*3i - 3*c^2*d - c^3*1i + d^3
))
```

3.88 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^3}\,dx$

Optimal result 995
Rubi [A] (verified) 996
Mathematica [A] (verified) 998
Maple [A] (verified) 999
Fricas [B] (verification not implemented) 1000
Sympy [F(-2)] 1002
Maxima [B] (verification not implemented) 1002
Giac [B] (verification not implemented) 1003
Mupad [B] (verification not implemented) 1004

Optimal result

Integrand size = 45, antiderivative size = 487

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^3}\,dx =$$
$$-\frac{(a(c^3C-3Bc^2d-3cCd^2+Bd^3-A(c^3-3cd^2))+b((A-C)d(3c^2-d^2)-B(c^3-3cd^2)))\,x}{(a^2+b^2)\,(c^2+d^2)^3}$$
$$+\frac{b^2(Ab^2-a(bB-aC))\log(a\cos(e+fx)+b\sin(e+fx))}{(a^2+b^2)\,(bc-ad)^3f}$$
$$-\frac{(b^2(c^6C-3Bc^5d+3c^4(2A-C)d^2+Bc^3d^3+3Ac^2d^4+Ad^6)+a^2d^3((A-C)d(3c^2-d^2)-B(c^3-3c}{(bc-ad)^3\,(c^2+d^2)^3}$$
$$+\frac{c^2C-Bcd+Ad^2}{2(bc-ad)\,(c^2+d^2)\,f(c+d\tan(e+fx))^2}$$
$$+\frac{b(c^4C-2Bc^3d+c^2(3A-C)d^2+Ad^4)-ad^2(2c(A-C)d-B(c^2-d^2))}{(bc-ad)^2\,(c^2+d^2)^2\,f(c+d\tan(e+fx))}$$

```
[Out] -(a*(c^3*C-3*B*c^2*d-3*C*c*d^2+B*d^3-A*(c^3-3*c*d^2))+b*((A-C)*d*(3*c^2-d^2
)-B*(c^3-3*c*d^2)))*x/(a^2+b^2)/(c^2+d^2)^3+b^2*(A*b^2-a*(B*b-C*a))*ln(a*co
s(f*x+e)+b*sin(f*x+e))/(a^2+b^2)/(-a*d+b*c)^3/f-(b^2*(c^6*C-3*B*c^5*d+3*c^4
*(2*A-C)*d^2+B*c^3*d^3+3*A*c^2*d^4+A*d^6)+a^2*d^3*((A-C)*d*(3*c^2-d^2)-B*(c
^3-3*c*d^2))-a*b*d^2*(8*c^3*(A-C)*d-B*(3*c^4-6*c^2*d^2-d^4)))*ln(c*cos(f*x+
e)+d*sin(f*x+e))/(-a*d+b*c)^3/(c^2+d^2)^3/f+1/2*(A*d^2-B*c*d+C*c^2)/(-a*d+b
*c)/(c^2+d^2)/f/(c+d*tan(f*x+e))^2+(b*(c^4*C-2*B*c^3*d+c^2*(3*A-C)*d^2+A*d^
4)-a*d^2*(2*c*(A-C)*d-B*(c^2-d^2)))/(-a*d+b*c)^2/(c^2+d^2)^2/f/(c+d*tan(f*x
+e))
```

Rubi [A] (verified)

Time = 2.06 (sec) , antiderivative size = 487, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 3, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.067, Rules used = {3730, 3732, 3611}

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^3}\,dx =$$

$$-\frac{x(a(-A(c^3-3cd^2)-3Bc^2d+Bd^3+c^3C-3cCd^2)+b(d(A-C)\left(3c^2-d^2\right)-B(c^3-3cd^2)))}{\left(a^2+b^2\right)\left(c^2+d^2\right)^3}$$
$$-\frac{(a^2d^3(d(A-C)\left(3c^2-d^2\right)-B(c^3-3cd^2))-abd^2(8c^3d(A-C)-B(3c^4-6c^2d^2-d^4))+b^2(3c^4d^2(2A}{f\left(c^2+d^2\right)^3(bc-ad)^{}}$$
$$+\frac{b^2(Ab^2-a(bB-aC))\log(a\cos(e+fx)+b\sin(e+fx))}{f\left(a^2+b^2\right)(bc-ad)^3}$$
$$+\frac{Ad^2-Bcd+c^2C}{2f\left(c^2+d^2\right)(bc-ad)(c+d\tan(e+fx))^2}$$
$$+\frac{b(c^2d^2(3A-C)+Ad^4-2Bc^3d+c^4C)-ad^2(2cd(A-C)-B(c^2-d^2))}{f\left(c^2+d^2\right)^2(bc-ad)^2(c+d\tan(e+fx))}$$

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])*(c + d*Ta
n[e + f*x])^3),x]
```

```
[Out] -(((a*(c^3*C - 3*B*c^2*d - 3*c*C*d^2 + B*d^3 - A*(c^3 - 3*c*d^2)) + b*((A -
 C)*d*(3*c^2 - d^2) - B*(c^3 - 3*c*d^2)))*x)/((a^2 + b^2)*(c^2 + d^2)^3)) +
 (b^2*(A*b^2 - a*(b*B - a*C))*Log[a*Cos[e + f*x] + b*Sin[e + f*x]])/((a^2 +
 b^2)*(b*c - a*d)^3*f) - ((b^2*(c^6*C - 3*B*c^5*d + 3*c^4*(2*A - C)*d^2 + B
*c^3*d^3 + 3*A*c^2*d^4 + A*d^6) + a^2*d^3*((A - C)*d*(3*c^2 - d^2) - B*(c^3
 - 3*c*d^2)) - a*b*d^2*(8*c^3*(A - C)*d - B*(3*c^4 - 6*c^2*d^2 - d^4)))*Log
[c*Cos[e + f*x] + d*Sin[e + f*x]])/((b*c - a*d)^3*(c^2 + d^2)^3*f) + (c^2*C
 - B*c*d + A*d^2)/(2*(b*c - a*d)*(c^2 + d^2)*f*(c + d*Tan[e + f*x])^2) + (b
*(c^4*C - 2*B*c^3*d + c^2*(3*A - C)*d^2 + A*d^4) - a*d^2*(2*c*(A - C)*d - B
*(c^2 - d^2)))/((b*c - a*d)^2*(c^2 + d^2)^2*f*(c + d*Tan[e + f*x]))
```

Rule 3611

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c/(b*f))*Log[RemoveContent[a*Cos[e + f*x] + b*Si
n[e + f*x], x]], x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3730

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 - a*(b*B - a*C))*(a + b*Tan[e +
```

```
 f*x])^(m + 1)*((c + d*Tan[e + f*x])^(n + 1)/(f*(m + 1)*(b*c - a*d)*(a^2 +
b^2))), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3732

```
Int[((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^
2)/(((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])), x_Symbol] :> Simp[(a*(A*c - c*C + B*d) + b*(B*c - A*d + C*d))*(x/
((a^2 + b^2)*(c^2 + d^2))), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/((b*c - a*d)
*(a^2 + b^2)), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] - Dist
[(c^2*C - B*c*d + A*d^2)/((b*c - a*d)*(c^2 + d^2)), Int[(d - c*Tan[e + f*x]
)/(c + d*Tan[e + f*x]), x], x]) /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] &&
 NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0]
```

Rubi steps

integral

$$= \frac{c^2C - Bcd + Ad^2}{2(bc-ad)\left(c^2+d^2\right)f(c+d\tan(e+fx))^2} + \frac{\int \frac{-2\left(aAcd-ad(cC-Bd)-Ab\left(c^2+d^2\right)\right)+2(bc-ad)(Bc-(A-C)d)\tan(e+fx)+2b\left(c^2C-Bcd+Ad^2\right)\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^2}\,dx}{2(bc-ad)\left(c^2+d^2\right)}$$

$$= \frac{c^2C - Bcd + Ad^2}{2(bc-ad)\left(c^2+d^2\right)f(c+d\tan(e+fx))^2} + \frac{b\left(c^4C-2Bc^3d+c^2(3A-C)d^2+Ad^4\right)-ad^2\left(2c(A-C)d-B\left(c^2-d^2\right)\right)}{(bc-ad)^2\left(c^2+d^2\right)^2 f(c+d\tan(e+fx))} + \frac{\int \frac{-2\left(A\left(2abc^3d-a^2d^2\left(c^2-d^2\right)-b^2\left(c^2+d^2\right)^2\right)+ad\left(ad\left(c^2C-2Bcd-Cd^2\right)-b\left(2c^3C-3Bc^2d-Bd^3\right)\right)\right)-2(bc-ad)^2\left(2c(A-C)d-B\left(c^2-d^2\right)\right)\,\mathrm{ta}}{(a+b\tan(e+fx))(c+d\tan(e+fx))}}{2(bc-ad)^2\left(c^2+d^2\right)^2}$$

$$= -\frac{(b(A-C)d(3c^2-d^2)-bB(c^3-3cd^2)-a(Ac^3-c^3C+3Bc^2d-3Acd^2+3cCd^2-Bd^3))\,x}{(a^2+b^2)\,(c^2+d^2)^3} + \frac{c^2C-Bcd+Ad^2}{2(bc-ad)\,(c^2+d^2)\,f(c+d\tan(e+fx))^2} + \frac{b(c^4C-2Bc^3d+c^2(3A-C)d^2+Ad^4)-ad^2(2c(A-C)d-B(c^2-d^2))}{(bc-ad)^2\,(c^2+d^2)^2\,f(c+d\tan(e+fx))} + \frac{(b^2(Ab^2-a(bB-aC)))\int\frac{b-a\tan(e+fx)}{a+b\tan(e+fx)}\,dx}{(a^2+b^2)\,(bc-ad)^3} - \frac{(b^2(c^6C-3Bc^5d+3c^4(2A-C)d^2+Bc^3d^3+3Ac^2d^4+Ad^6)+a^2d^3((A-C)d(3c^2-d^2)-B(c^3-3cd^2}{(bc-ad)^3\,(c^2+d^2)^3}$$

$$= -\frac{(b(A-C)d(3c^2-d^2)-bB(c^3-3cd^2)-a(Ac^3-c^3C+3Bc^2d-3Acd^2+3cCd^2-Bd^3))\,x}{(a^2+b^2)\,(c^2+d^2)^3} + \frac{b^2(Ab^2-a(bB-aC))\log(a\cos(e+fx)+b\sin(e+fx))}{(a^2+b^2)\,(bc-ad)^3f} - \frac{(b^2(c^6C-3Bc^5d+3c^4(2A-C)d^2+Bc^3d^3+3Ac^2d^4+Ad^6)+a^2d^3((A-C)d(3c^2-d^2)-B(c^3-3cd^2}{(bc-ad)^3\,(c^2+d^2)^3\,f} + \frac{c^2C-Bcd+Ad^2}{2(bc-ad)\,(c^2+d^2)\,f(c+d\tan(e+fx))^2} + \frac{b(c^4C-2Bc^3d+c^2(3A-C)d^2+Ad^4)-ad^2(2c(A-C)d-B(c^2-d^2))}{(bc-ad)^2\,(c^2+d^2)^2\,f(c+d\tan(e+fx))}$$

Mathematica [A] (verified)

Time = 9.02 (sec) , antiderivative size = 912, normalized size of antiderivative = 1.87

$$\int\frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^3}\,dx = -\frac{Ad^2-c(-cC+Bd)}{2(-bc+ad)\,(c^2+d^2)\,f(c+d\tan(e+fx))^2} - \frac{-\frac{-\frac{b(bc-ad)^2\left(Abc^3-aBc^3-bc^3C+3aAc^2d+3bBc^2d-3ac^2Cd-3Abcd^2+3aBcd^2+3bcCd^2-aAd^3-bBd^3+aCd^3-\frac{\sqrt{-b^2}\left(a\left(c^3C-3Bc^2d-3cCd^2+Bd^3-A\left(c^3\right.\right.\right.}{}\right.}{(a^2+b^2)(c^2+d^2)}}{}}{}$$

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])*(c
+ d*Tan[e + f*x])^3),x]
```

```
[Out] -1/2*(A*d^2 - c*(-(c*C) + B*d))/((-(b*c) + a*d)*(c^2 + d^2)*f*(c + d*Tan[e
+ f*x])^2) - (-((-((b*(b*c - a*d)^2*(A*b*c^3 - a*B*c^3 - b*c^3*C + 3*a*A*c^
```

2*d + 3*b*B*c^2*d - 3*a*c^2*C*d - 3*A*b*c*d^2 + 3*a*B*c*d^2 + 3*b*c*C*d^2 - a*A*d^3 - b*B*d^3 + a*C*d^3 - (Sqrt[-b^2]*(a*(c^3*C - 3*B*c^2*d - 3*c*C*d^2 + B*d^3 - A*(c^3 - 3*c*d^2)) + b*((A - C)*d*(3*c^2 - d^2) - B*(c^3 - 3*c*d^2))))/b)*Log[Sqrt[-b^2] - b*Tan[e + f*x]])/((a^2 + b^2)*(c^2 + d^2))) + (2*b^3*(A*b^2 - a*(b*B - a*C))*(c^2 + d^2)^2*Log[a + b*Tan[e + f*x]])/((a^2 + b^2)*(b*c - a*d)) - (b*(b*c - a*d)^2*(A*b*c^3 - a*B*c^3 - b*c^3*C + 3*a*A*c^2*d + 3*b*B*c^2*d - 3*a*c^2*C*d - 3*A*b*c*d^2 + 3*a*B*c*d^2 + 3*b*c*C*d^2 - a*A*d^3 - b*B*d^3 + a*C*d^3 + (Sqrt[-b^2]*(b*(A - C)*d*(3*c^2 - d^2) - b*B*(c^3 - 3*c*d^2) - a*(A*c^3 - c^3*C + 3*B*c^2*d - 3*A*c*d^2 + 3*c*C*d^2 - B*d^3)))/b)*Log[Sqrt[-b^2] + b*Tan[e + f*x]])/((a^2 + b^2)*(c^2 + d^2)) - (2*b*(b^2*(c^6*C - 3*B*c^5*d + 3*c^4*(2*A - C)*d^2 + B*c^3*d^3 + 3*A*c^2*d^4 + A*d^6) + a^2*d^3*((A - C)*d*(3*c^2 - d^2) - B*(c^3 - 3*c*d^2)) - a*b*d^2*(8*c^3*(A - C)*d - B*(3*c^4 - 6*c^2*d^2 - d^4)))*Log[c + d*Tan[e + f*x]])/((b*c - a*d)*(c^2 + d^2)))/(b*(-(b*c) + a*d)*(c^2 + d^2)*f)) - (-2*d^2*(a*A*c*d - a*d*(c*C - B*d) - A*b*(c^2 + d^2)) - c*(2*d*(b*c - a*d)*(B*c - (A - C)*d) - 2*b*c*(c^2*C - B*c*d + A*d^2)))/((-(b*c) + a*d)*(c^2 + d^2)*f*(c + d*Tan[e + f*x])))/(2*(-(b*c) + a*d)*(c^2 + d^2))

Maple [A] (verified)

Time = 1.77 (sec) , antiderivative size = 649, normalized size of antiderivative = 1.33

method	result
derivativedivides	$\frac{-\frac{\left(A\,b^{2}-Bab+C\,a^{2}\right)b^{2}\ln\left(a+b\tan\left(fx+e\right)\right)}{\left(ad-bc\right)^{3}\left(a^{2}+b^{2}\right)}+\frac{\frac{\left(-3Aa\,c^{2}d+Aa\,d^{3}-Ab\,c^{3}+3Abc\,d^{2}+Ba\,c^{3}-3Bac\,d^{2}-3Bb\,c^{2}d+Bb\,d^{3}+3Ca\,c^{2}d-Ca\,d^{3}}{2}}{}}{}$
default	$\frac{-\frac{\left(A\,b^{2}-Bab+C\,a^{2}\right)b^{2}\ln\left(a+b\tan\left(fx+e\right)\right)}{\left(ad-bc\right)^{3}\left(a^{2}+b^{2}\right)}+\frac{\frac{\left(-3Aa\,c^{2}d+Aa\,d^{3}-Ab\,c^{3}+3Abc\,d^{2}+Ba\,c^{3}-3Bac\,d^{2}-3Bb\,c^{2}d+Bb\,d^{3}+3Ca\,c^{2}d-Ca\,d^{3}}{2}}{}}{}$
norman	Expression too large to display
parallelrisch	Expression too large to display
risch	Expression too large to display

[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))^3,x,method=_RETURNVERBOSE)

[Out] 1/f*(-(A*b^2-B*a*b+C*a^2)*b^2/(a*d-b*c)^3/(a^2+b^2)*ln(a+b*tan(f*x+e))+1/(a^2+b^2)/(c^2+d^2)^3*(1/2*(-3*A*a*c^2*d+A*a*d^3-A*b*c^3+3*A*b*c*d^2+B*a*c^3-3*B*a*c*d^2-3*B*b*c^2*d+B*b*d^3+3*C*a*c^2*d-C*a*d^3+C*b*c^3-3*C*b*c*d^2)*ln(1+tan(f*x+e)^2)+(A*a*c^3-3*A*a*c*d^2-3*A*b*c^2*d+A*b*d^3+3*B*a*c^2*d-B*a*d^3+B*b*c^3-3*B*b*c*d^2-C*a*c^3+3*C*a*c*d^2+3*C*b*c^2*d-C*b*d^3)*arctan(tan(f*x+e)))-(2*A*a*c*d^3-3*A*b*c^2*d^2-A*b*d^4-B*a*c^2*d^2+B*a*d^4+2*B*b*c^3*d-2*C*a*c*d^3-C*b*c^4+C*b*c^2*d^2)/(a*d-b*c)^2/(c^2+d^2)^2/(c+d*tan(f*x+e))+(3*A*a^2*c^2*d^4-A*a^2*d^6-8*A*a*b*c^3*d^3+6*A*b^2*c^4*d^2+3*A*b^2*c^2*d^4+A*b^2*d^6-B*a^2*c^3*d^3+3*B*a^2*c*d^5+3*B*a*b*c^4*d^2-6*B*a*b*c^2*d^4-B*a*b

```
*d^6-3*B*b^2*c^5*d+B*b^2*c^3*d^3-3*C*a^2*c^2*d^4+C*a^2*d^6+8*C*a*b*c^3*d^3+
C*b^2*c^6-3*C*b^2*c^4*d^2)/(a*d-b*c)^3/(c^2+d^2)^3*ln(c+d*tan(f*x+e))-1/2*(
A*d^2-B*c*d+C*c^2)/(a*d-b*c)/(c^2+d^2)/(c+d*tan(f*x+e))^2)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 3496 vs. $2(485) = 970$.

Time = 4.01 (sec) , antiderivative size = 3496, normalized size of antiderivative = 7.18

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^3}\,dx=\text{Too large to display}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))
^3,x, algorithm="fricas")
```

```
[Out] 1/2*(5*(C*a^2*b^2 + C*b^4)*c^6*d^2 - (8*C*a^3*b + 7*B*a^2*b^2 + 8*C*a*b^3 +
 7*B*b^4)*c^5*d^3 + (3*C*a^4 + 12*B*a^3*b + (9*A + 2*C)*a^2*b^2 + 12*B*a*b^
3 + (9*A - C)*b^4)*c^4*d^4 - (5*B*a^4 + 4*(4*A - C)*a^3*b + 6*B*a^2*b^2 + 4
*(4*A - C)*a*b^3 + B*b^4)*c^3*d^5 + ((7*A - 3*C)*a^4 + (10*A - 3*C)*a^2*b^2
 + 3*A*b^4)*c^2*d^6 + (B*a^4 - 4*A*a^3*b + B*a^2*b^2 - 4*A*a*b^3)*c*d^7 + (
A*a^4 + A*a^2*b^2)*d^8 + 2*(((A - C)*a*b^3 + B*b^4)*c^8 - 3*((A - C)*a^2*b^
2 + (A - C)*b^4)*c^7*d + 3*((A - C)*a^3*b - 2*B*a^2*b^2 + 2*(A - C)*a*b^3 -
 B*b^4)*c^6*d^2 - ((A - C)*a^4 - 8*B*a^3*b - 8*B*a*b^3 - (A - C)*b^4)*c^5*d
^3 - 3*(B*a^4 + 2*(A - C)*a^3*b + 2*B*a^2*b^2 + (A - C)*a*b^3)*c^4*d^4 + 3*
((A - C)*a^4 + (A - C)*a^2*b^2)*c^3*d^5 + (B*a^4 - (A - C)*a^3*b)*c^2*d^6)*
f*x - (3*(C*a^2*b^2 + C*b^4)*c^6*d^2 - (4*C*a^3*b + 5*B*a^2*b^2 + 4*C*a*b^3
 + 5*B*b^4)*c^5*d^3 + (C*a^4 + 8*B*a^3*b + (7*A - 2*C)*a^2*b^2 + 8*B*a*b^3
+ (7*A - 3*C)*b^4)*c^4*d^4 - (3*B*a^4 + 4*(3*A - 2*C)*a^3*b + 2*B*a^2*b^2 +
 4*(3*A - 2*C)*a*b^3 - B*b^4)*c^3*d^5 + (5*(A - C)*a^4 - 4*B*a^3*b + (6*A -
 5*C)*a^2*b^2 - 4*B*a*b^3 + A*b^4)*c^2*d^6 + 3*(B*a^4 + B*a^2*b^2)*c*d^7 -
(A*a^4 + A*a^2*b^2)*d^8 - 2*(((A - C)*a*b^3 + B*b^4)*c^6*d^2 - 3*((A - C)*a
^2*b^2 + (A - C)*b^4)*c^5*d^3 + 3*((A - C)*a^3*b - 2*B*a^2*b^2 + 2*(A - C)*
a*b^3 - B*b^4)*c^4*d^4 - ((A - C)*a^4 - 8*B*a^3*b - 8*B*a*b^3 - (A - C)*b^4
)*c^3*d^5 - 3*(B*a^4 + 2*(A - C)*a^3*b + 2*B*a^2*b^2 + (A - C)*a*b^3)*c^2*d
^6 + 3*((A - C)*a^4 + (A - C)*a^2*b^2)*c*d^7 + (B*a^4 - (A - C)*a^3*b)*d^8)
*f*x)*tan(f*x + e)^2 + ((C*a^2*b^2 - B*a*b^3 + A*b^4)*c^8 + 3*(C*a^2*b^2 -
B*a*b^3 + A*b^4)*c^6*d^2 + 3*(C*a^2*b^2 - B*a*b^3 + A*b^4)*c^4*d^4 + (C*a^2
*b^2 - B*a*b^3 + A*b^4)*c^2*d^6 + ((C*a^2*b^2 - B*a*b^3 + A*b^4)*c^6*d^2 +
3*(C*a^2*b^2 - B*a*b^3 + A*b^4)*c^4*d^4 + 3*(C*a^2*b^2 - B*a*b^3 + A*b^4)*c
^2*d^6 + (C*a^2*b^2 - B*a*b^3 + A*b^4)*d^8)*tan(f*x + e)^2 + 2*((C*a^2*b^2
- B*a*b^3 + A*b^4)*c^7*d + 3*(C*a^2*b^2 - B*a*b^3 + A*b^4)*c^5*d^3 + 3*(C*a
^2*b^2 - B*a*b^3 + A*b^4)*c^3*d^5 + (C*a^2*b^2 - B*a*b^3 + A*b^4)*c*d^7)*ta
n(f*x + e))*log((b^2*tan(f*x + e)^2 + 2*a*b*tan(f*x + e) + a^2)/(tan(f*x +
e)^2 + 1)) - ((C*a^2*b^2 + C*b^4)*c^8 - 3*(B*a^2*b^2 + B*b^4)*c^7*d + 3*(B*
a^3*b + (2*A - C)*a^2*b^2 + B*a*b^3 + (2*A - C)*b^4)*c^6*d^2 - (B*a^4 + 8*(
```

```
A - C)*a^3*b + 8*(A - C)*a*b^3 - B*b^4)*c^5*d^3 + 3*((A - C)*a^4 - 2*B*a^3*
b + (2*A - C)*a^2*b^2 - 2*B*a*b^3 + A*b^4)*c^4*d^4 + 3*(B*a^4 + B*a^2*b^2)*
c^3*d^5 - ((A - C)*a^4 + B*a^3*b - C*a^2*b^2 + B*a*b^3 - A*b^4)*c^2*d^6 + (
(C*a^2*b^2 + C*b^4)*c^6*d^2 - 3*(B*a^2*b^2 + B*b^4)*c^5*d^3 + 3*(B*a^3*b +
(2*A - C)*a^2*b^2 + B*a*b^3 + (2*A - C)*b^4)*c^4*d^4 - (B*a^4 + 8*(A - C)*a
^3*b + 8*(A - C)*a*b^3 - B*b^4)*c^3*d^5 + 3*((A - C)*a^4 - 2*B*a^3*b + (2*A
 - C)*a^2*b^2 - 2*B*a*b^3 + A*b^4)*c^2*d^6 + 3*(B*a^4 + B*a^2*b^2)*c*d^7 -
((A - C)*a^4 + B*a^3*b - C*a^2*b^2 + B*a*b^3 - A*b^4)*d^8)*tan(f*x + e)^2 +
 2*((C*a^2*b^2 + C*b^4)*c^7*d - 3*(B*a^2*b^2 + B*b^4)*c^6*d^2 + 3*(B*a^3*b
+ (2*A - C)*a^2*b^2 + B*a*b^3 + (2*A - C)*b^4)*c^5*d^3 - (B*a^4 + 8*(A - C)
*a^3*b + 8*(A - C)*a*b^3 - B*b^4)*c^4*d^4 + 3*((A - C)*a^4 - 2*B*a^3*b + (2
*A - C)*a^2*b^2 - 2*B*a*b^3 + A*b^4)*c^3*d^5 + 3*(B*a^4 + B*a^2*b^2)*c^2*d^
6 - ((A - C)*a^4 + B*a^3*b - C*a^2*b^2 + B*a*b^3 - A*b^4)*c*d^7)*tan(f*x +
e))*log((d^2*tan(f*x + e)^2 + 2*c*d*tan(f*x + e) + c^2)/(tan(f*x + e)^2 + 1
)) - 2*(2*(C*a^2*b^2 + C*b^4)*c^7*d - 3*(C*a^3*b + B*a^2*b^2 + C*a*b^3 + B*
b^4)*c^6*d^2 + (C*a^4 + 5*B*a^3*b + 2*(2*A - C)*a^2*b^2 + 5*B*a*b^3 + (4*A
- 3*C)*b^4)*c^5*d^3 - (2*B*a^4 + (7*A - 6*C)*a^3*b - B*a^2*b^2 + (7*A - 6*C
)*a*b^3 - 3*B*b^4)*c^4*d^4 + (3*(A - C)*a^4 - 6*B*a^3*b - 2*C*a^2*b^2 - 6*B
*a*b^3 - (3*A - C)*b^4)*c^3*d^5 + 3*(B*a^4 + (2*A - C)*a^3*b + B*a^2*b^2 +
(2*A - C)*a*b^3)*c^2*d^6 - ((3*A - 2*C)*a^4 - B*a^3*b + 2*(2*A - C)*a^2*b^2
 - B*a*b^3 + A*b^4)*c*d^7 - (B*a^4 - A*a^3*b + B*a^2*b^2 - A*a*b^3)*d^8 - 2
*(((A - C)*a*b^3 + B*b^4)*c^7*d - 3*((A - C)*a^2*b^2 + (A - C)*b^4)*c^6*d^2
 + 3*((A - C)*a^3*b - 2*B*a^2*b^2 + 2*(A - C)*a*b^3 - B*b^4)*c^5*d^3 - ((A
- C)*a^4 - 8*B*a^3*b - 8*B*a*b^3 - (A - C)*b^4)*c^4*d^4 - 3*(B*a^4 + 2*(A -
 C)*a^3*b + 2*B*a^2*b^2 + (A - C)*a*b^3)*c^3*d^5 + 3*((A - C)*a^4 + (A - C)
*a^2*b^2)*c^2*d^6 + (B*a^4 - (A - C)*a^3*b)*c*d^7)*f*x)*tan(f*x + e))/(((a^
2*b^3 + b^5)*c^9*d^2 - 3*(a^3*b^2 + a*b^4)*c^8*d^3 + 3*(a^4*b + 2*a^2*b^3 +
 b^5)*c^7*d^4 - (a^5 + 10*a^3*b^2 + 9*a*b^4)*c^6*d^5 + 3*(3*a^4*b + 4*a^2*b
^3 + b^5)*c^5*d^6 - 3*(a^5 + 4*a^3*b^2 + 3*a*b^4)*c^4*d^7 + (9*a^4*b + 10*a
^2*b^3 + b^5)*c^3*d^8 - 3*(a^5 + 2*a^3*b^2 + a*b^4)*c^2*d^9 + 3*(a^4*b + a^
2*b^3)*c*d^10 - (a^5 + a^3*b^2)*d^11)*f*tan(f*x + e)^2 + 2*((a^2*b^3 + b^5)
*c^10*d - 3*(a^3*b^2 + a*b^4)*c^9*d^2 + 3*(a^4*b + 2*a^2*b^3 + b^5)*c^8*d^3
 - (a^5 + 10*a^3*b^2 + 9*a*b^4)*c^7*d^4 + 3*(3*a^4*b + 4*a^2*b^3 + b^5)*c^6
*d^5 - 3*(a^5 + 4*a^3*b^2 + 3*a*b^4)*c^5*d^6 + (9*a^4*b + 10*a^2*b^3 + b^5)
*c^4*d^7 - 3*(a^5 + 2*a^3*b^2 + a*b^4)*c^3*d^8 + 3*(a^4*b + a^2*b^3)*c^2*d^
9 - (a^5 + a^3*b^2)*c*d^10)*f*tan(f*x + e) + ((a^2*b^3 + b^5)*c^11 - 3*(a^3
*b^2 + a*b^4)*c^10*d + 3*(a^4*b + 2*a^2*b^3 + b^5)*c^9*d^2 - (a^5 + 10*a^3*
b^2 + 9*a*b^4)*c^8*d^3 + 3*(3*a^4*b + 4*a^2*b^3 + b^5)*c^7*d^4 - 3*(a^5 + 4
*a^3*b^2 + 3*a*b^4)*c^6*d^5 + (9*a^4*b + 10*a^2*b^3 + b^5)*c^5*d^6 - 3*(a^5
 + 2*a^3*b^2 + a*b^4)*c^4*d^7 + 3*(a^4*b + a^2*b^3)*c^3*d^8 - (a^5 + a^3*b^
2)*c^2*d^9)*f)
```

Sympy [F(-2)]

Exception generated.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^3}\,dx = \text{Exception raised: NotImplementedError}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e)
)**3,x)
[Out] Exception raised: NotImplementedError >> no valid subset found
```

Maxima [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 1078 vs. $2(485) = 970$.

Time = 0.38 (sec) , antiderivative size = 1078, normalized size of antiderivative = 2.21

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^3}\,dx = \text{Too large to display}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))
^3,x, algorithm="maxima")
[Out] 1/2*(2*(((A - C)*a + B*b)*c^3 + 3*(B*a - (A - C)*b)*c^2*d - 3*((A - C)*a +
B*b)*c*d^2 - (B*a - (A - C)*b)*d^3)*(f*x + e)/((a^2 + b^2)*c^6 + 3*(a^2 + b
^2)*c^4*d^2 + 3*(a^2 + b^2)*c^2*d^4 + (a^2 + b^2)*d^6) + 2*(C*a^2*b^2 - B*a
*b^3 + A*b^4)*log(b*tan(f*x + e) + a)/((a^2*b^3 + b^5)*c^3 - 3*(a^3*b^2 + a
*b^4)*c^2*d + 3*(a^4*b + a^2*b^3)*c*d^2 - (a^5 + a^3*b^2)*d^3) - 2*(C*b^2*c
^6 - 3*B*b^2*c^5*d + 3*B*a^2*c*d^5 + 3*(B*a*b + (2*A - C)*b^2)*c^4*d^2 - (B
*a^2 + 8*(A - C)*a*b - B*b^2)*c^3*d^3 + 3*((A - C)*a^2 - 2*B*a*b + A*b^2)*c
^2*d^4 - ((A - C)*a^2 + B*a*b - A*b^2)*d^6)*log(d*tan(f*x + e) + c)/(b^3*c^
9 - 3*a*b^2*c^8*d + 3*a^2*b*c*d^8 - a^3*d^9 + 3*(a^2*b + b^3)*c^7*d^2 - (a^
3 + 9*a*b^2)*c^6*d^3 + 3*(3*a^2*b + b^3)*c^5*d^4 - 3*(a^3 + 3*a*b^2)*c^4*d^
5 + (9*a^2*b + b^3)*c^3*d^6 - 3*(a^3 + a*b^2)*c^2*d^7) + ((B*a - (A - C)*b)
*c^3 - 3*((A - C)*a + B*b)*c^2*d - 3*(B*a - (A - C)*b)*c*d^2 + ((A - C)*a +
 B*b)*d^3)*log(tan(f*x + e)^2 + 1)/((a^2 + b^2)*c^6 + 3*(a^2 + b^2)*c^4*d^2
 + 3*(a^2 + b^2)*c^2*d^4 + (a^2 + b^2)*d^6) + (3*C*b*c^5 - A*a*d^5 - (C*a +
 5*B*b)*c^4*d + (3*B*a + (7*A - C)*b)*c^3*d^2 - ((5*A - 3*C)*a + B*b)*c^2*d
^3 - (B*a - 3*A*b)*c*d^4 + 2*(C*b*c^4*d - 2*B*b*c^3*d^2 - 2*(A - C)*a*c*d^4
 + (B*a + (3*A - C)*b)*c^2*d^3 - (B*a - A*b)*d^5)*tan(f*x + e))/(b^2*c^8 -
2*a*b*c^7*d - 4*a*b*c^5*d^3 - 2*a*b*c^3*d^5 + a^2*c^2*d^6 + (a^2 + 2*b^2)*c
^6*d^2 + (2*a^2 + b^2)*c^4*d^4 + (b^2*c^6*d^2 - 2*a*b*c^5*d^3 - 4*a*b*c^3*d
^5 - 2*a*b*c*d^7 + a^2*d^8 + (a^2 + 2*b^2)*c^4*d^4 + (2*a^2 + b^2)*c^2*d^6)
*tan(f*x + e)^2 + 2*(b^2*c^7*d - 2*a*b*c^6*d^2 - 4*a*b*c^4*d^4 - 2*a*b*c^2*
d^6 + a^2*c*d^7 + (a^2 + 2*b^2)*c^5*d^3 + (2*a^2 + b^2)*c^3*d^5)*tan(f*x +
e)))/f
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 2078 vs. $2(485) = 970$.

Time = 1.04 (sec) , antiderivative size = 2078, normalized size of antiderivative = 4.27

$$\int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{(a + b\tan(e + fx))(c + d\tan(e + fx))^3}\,dx = \text{Too large to display}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))
^3,x, algorithm="giac")
```

```
[Out] 1/2*(2*(A*a*c^3 - C*a*c^3 + B*b*c^3 + 3*B*a*c^2*d - 3*A*b*c^2*d + 3*C*b*c^2
*d - 3*A*a*c*d^2 + 3*C*a*c*d^2 - 3*B*b*c*d^2 - B*a*d^3 + A*b*d^3 - C*b*d^3)
*(f*x + e)/(a^2*c^6 + b^2*c^6 + 3*a^2*c^4*d^2 + 3*b^2*c^4*d^2 + 3*a^2*c^2*d
^4 + 3*b^2*c^2*d^4 + a^2*d^6 + b^2*d^6) + (B*a*c^3 - A*b*c^3 + C*b*c^3 - 3*
A*a*c^2*d + 3*C*a*c^2*d - 3*B*b*c^2*d - 3*B*a*c*d^2 + 3*A*b*c*d^2 - 3*C*b*c
*d^2 + A*a*d^3 - C*a*d^3 + B*b*d^3)*log(tan(f*x + e)^2 + 1)/(a^2*c^6 + b^2*
c^6 + 3*a^2*c^4*d^2 + 3*b^2*c^4*d^2 + 3*a^2*c^2*d^4 + 3*b^2*c^2*d^4 + a^2*d
^6 + b^2*d^6) + 2*(C*a^2*b^3 - B*a*b^4 + A*b^5)*log(abs(b*tan(f*x + e) + a)
)/(a^2*b^4*c^3 + b^6*c^3 - 3*a^3*b^3*c^2*d - 3*a*b^5*c^2*d + 3*a^4*b^2*c*d^
2 + 3*a^2*b^4*c*d^2 - a^5*b*d^3 - a^3*b^3*d^3) - 2*(C*b^2*c^6*d - 3*B*b^2*c
^5*d^2 + 3*B*a*b*c^4*d^3 + 6*A*b^2*c^4*d^3 - 3*C*b^2*c^4*d^3 - B*a^2*c^3*d^
4 - 8*A*a*b*c^3*d^4 + 8*C*a*b*c^3*d^4 + B*b^2*c^3*d^4 + 3*A*a^2*c^2*d^5 - 3
*C*a^2*c^2*d^5 - 6*B*a*b*c^2*d^5 + 3*A*b^2*c^2*d^5 + 3*B*a^2*c*d^6 - A*a^2*
d^7 + C*a^2*d^7 - B*a*b*d^7 + A*b^2*d^7)*log(abs(d*tan(f*x + e) + c))/(b^3*
c^9*d - 3*a*b^2*c^8*d^2 + 3*a^2*b*c^7*d^3 + 3*b^3*c^7*d^3 - a^3*c^6*d^4 - 9
*a*b^2*c^6*d^4 + 9*a^2*b*c^5*d^5 + 3*b^3*c^5*d^5 - 3*a^3*c^4*d^6 - 9*a*b^2*
c^4*d^6 + 9*a^2*b*c^3*d^7 + b^3*c^3*d^7 - 3*a^3*c^2*d^8 - 3*a*b^2*c^2*d^8 +
 3*a^2*b*c*d^9 - a^3*d^10) + (3*C*b^2*c^6*d^2*tan(f*x + e)^2 - 9*B*b^2*c^5*
d^3*tan(f*x + e)^2 + 9*B*a*b*c^4*d^4*tan(f*x + e)^2 + 18*A*b^2*c^4*d^4*tan(
f*x + e)^2 - 9*C*b^2*c^4*d^4*tan(f*x + e)^2 - 3*B*a^2*c^3*d^5*tan(f*x + e)^
2 - 24*A*a*b*c^3*d^5*tan(f*x + e)^2 + 24*C*a*b*c^3*d^5*tan(f*x + e)^2 + 3*B
*b^2*c^3*d^5*tan(f*x + e)^2 + 9*A*a^2*c^2*d^6*tan(f*x + e)^2 - 9*C*a^2*c^2*
d^6*tan(f*x + e)^2 - 18*B*a*b*c^2*d^6*tan(f*x + e)^2 + 9*A*b^2*c^2*d^6*tan(
f*x + e)^2 + 9*B*a^2*c*d^7*tan(f*x + e)^2 - 3*A*a^2*d^8*tan(f*x + e)^2 + 3*
C*a^2*d^8*tan(f*x + e)^2 - 3*B*a*b*d^8*tan(f*x + e)^2 + 3*A*b^2*d^8*tan(f*x
 + e)^2 + 8*C*b^2*c^7*d*tan(f*x + e) - 2*C*a*b*c^6*d^2*tan(f*x + e) - 22*B*
b^2*c^6*d^2*tan(f*x + e) + 24*B*a*b*c^5*d^3*tan(f*x + e) + 42*A*b^2*c^5*d^3
*tan(f*x + e) - 18*C*b^2*c^5*d^3*tan(f*x + e) - 8*B*a^2*c^4*d^4*tan(f*x + e
) - 58*A*a*b*c^4*d^4*tan(f*x + e) + 52*C*a*b*c^4*d^4*tan(f*x + e) + 2*B*b^2
*c^4*d^4*tan(f*x + e) + 22*A*a^2*c^3*d^5*tan(f*x + e) - 22*C*a^2*c^3*d^5*ta
n(f*x + e) - 32*B*a*b*c^3*d^5*tan(f*x + e) + 26*A*b^2*c^3*d^5*tan(f*x + e)
- 2*C*b^2*c^3*d^5*tan(f*x + e) + 18*B*a^2*c^2*d^6*tan(f*x + e) - 12*A*a*b*c
^2*d^6*tan(f*x + e) + 6*C*a*b*c^2*d^6*tan(f*x + e) - 2*A*a^2*c*d^7*tan(f*x
+ e) + 2*C*a^2*c*d^7*tan(f*x + e) - 8*B*a*b*c*d^7*tan(f*x + e) + 8*A*b^2*c*
```

```
d^7*tan(f*x + e) + 2*B*a^2*d^8*tan(f*x + e) - 2*A*a*b*d^8*tan(f*x + e) + 6*
C*b^2*c^8 - 4*C*a*b*c^7*d - 14*B*b^2*c^7*d + C*a^2*c^6*d^2 + 17*B*a*b*c^6*d
^2 + 25*A*b^2*c^6*d^2 - 7*C*b^2*c^6*d^2 - 6*B*a^2*c^5*d^3 - 36*A*a*b*c^5*d^
3 + 24*C*a*b*c^5*d^3 - 3*B*b^2*c^5*d^3 + 14*A*a^2*c^4*d^4 - 11*C*a^2*c^4*d^
4 - 10*B*a*b*c^4*d^4 + 19*A*b^2*c^4*d^4 - C*b^2*c^4*d^4 + 7*B*a^2*c^3*d^5 -
 16*A*a*b*c^3*d^5 + 4*C*a*b*c^3*d^5 - B*b^2*c^3*d^5 + 3*A*a^2*c^2*d^6 - 3*B
*a*b*c^2*d^6 + 6*A*b^2*c^2*d^6 + B*a^2*c*d^7 - 4*A*a*b*c*d^7 + A*a^2*d^8)/(
(b^3*c^9 - 3*a*b^2*c^8*d + 3*a^2*b*c^7*d^2 + 3*b^3*c^7*d^2 - a^3*c^6*d^3 -
9*a*b^2*c^6*d^3 + 9*a^2*b*c^5*d^4 + 3*b^3*c^5*d^4 - 3*a^3*c^4*d^5 - 9*a*b^2
*c^4*d^5 + 9*a^2*b*c^3*d^6 + b^3*c^3*d^6 - 3*a^3*c^2*d^7 - 3*a*b^2*c^2*d^7
+ 3*a^2*b*c*d^8 - a^3*d^9)*(d*tan(f*x + e) + c)^2))/f
```

Mupad [B] (verification not implemented)

Time = 23.07 (sec) , antiderivative size = 65817, normalized size of antiderivative = 135.15

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^3}\,dx=\text{Too large to display}$$

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))*(c + d*ta
n(e + f*x))^3),x)
```

```
[Out] (symsum(log(- root(480*a^9*b*c^7*d^11*f^4 + 480*a*b^9*c^11*d^7*f^4 + 360*a^
9*b*c^9*d^9*f^4 + 360*a^9*b*c^5*d^13*f^4 + 360*a*b^9*c^13*d^5*f^4 + 360*a*b
^9*c^9*d^9*f^4 + 144*a^9*b*c^11*d^7*f^4 + 144*a^9*b*c^3*d^15*f^4 + 144*a*b^
9*c^15*d^3*f^4 + 144*a*b^9*c^7*d^11*f^4 + 48*a^7*b^3*c*d^17*f^4 + 48*a^3*b^
7*c^17*d*f^4 + 24*a^9*b*c^13*d^5*f^4 + 24*a^5*b^5*c^17*d*f^4 + 24*a^5*b^5*c
*d^17*f^4 + 24*a*b^9*c^5*d^13*f^4 + 24*a^9*b*c*d^17*f^4 + 24*a*b^9*c^17*d*f
^4 + 3920*a^5*b^5*c^9*d^9*f^4 - 3360*a^6*b^4*c^8*d^10*f^4 - 3360*a^4*b^6*c^
10*d^8*f^4 - 3024*a^6*b^4*c^10*d^8*f^4 + 3024*a^5*b^5*c^11*d^7*f^4 + 3024*a
^5*b^5*c^7*d^11*f^4 - 3024*a^4*b^6*c^8*d^10*f^4 + 2320*a^7*b^3*c^9*d^9*f^4
+ 2320*a^3*b^7*c^9*d^9*f^4 - 2240*a^6*b^4*c^6*d^12*f^4 - 2240*a^4*b^6*c^12*
d^6*f^4 + 2160*a^7*b^3*c^7*d^11*f^4 + 2160*a^3*b^7*c^11*d^7*f^4 - 1624*a^6*
b^4*c^12*d^6*f^4 - 1624*a^4*b^6*c^6*d^12*f^4 + 1488*a^7*b^3*c^11*d^7*f^4 +
1488*a^3*b^7*c^7*d^11*f^4 + 1344*a^5*b^5*c^13*d^5*f^4 + 1344*a^5*b^5*c^5*d^
13*f^4 - 1320*a^8*b^2*c^8*d^10*f^4 - 1320*a^2*b^8*c^10*d^8*f^4 + 1200*a^7*b
^3*c^5*d^13*f^4 + 1200*a^3*b^7*c^13*d^5*f^4 - 1060*a^8*b^2*c^6*d^12*f^4 - 1
060*a^2*b^8*c^12*d^6*f^4 - 948*a^8*b^2*c^10*d^8*f^4 - 948*a^2*b^8*c^8*d^10*
f^4 - 840*a^6*b^4*c^4*d^14*f^4 - 840*a^4*b^6*c^14*d^4*f^4 + 528*a^7*b^3*c^1
3*d^5*f^4 + 528*a^3*b^7*c^5*d^13*f^4 - 480*a^8*b^2*c^4*d^14*f^4 - 480*a^6*b
^4*c^14*d^4*f^4 - 480*a^4*b^6*c^4*d^14*f^4 - 480*a^2*b^8*c^14*d^4*f^4 - 368
*a^8*b^2*c^12*d^6*f^4 + 368*a^7*b^3*c^3*d^15*f^4 + 368*a^3*b^7*c^15*d^3*f^4
 - 368*a^2*b^8*c^6*d^12*f^4 + 304*a^5*b^5*c^15*d^3*f^4 + 304*a^5*b^5*c^3*d^
15*f^4 - 144*a^6*b^4*c^2*d^16*f^4 - 144*a^4*b^6*c^16*d^2*f^4 - 108*a^8*b^2*
c^2*d^16*f^4 - 108*a^2*b^8*c^16*d^2*f^4 + 80*a^7*b^3*c^15*d^3*f^4 + 80*a^3*
```

b^7*c^3*d^15*f^4 - 60*a^8*b^2*c^14*d^4*f^4 - 60*a^6*b^4*c^16*d^2*f^4 - 60*a
^4*b^6*c^2*d^16*f^4 - 60*a^2*b^8*c^4*d^14*f^4 - 80*b^10*c^12*d^6*f^4 - 60*b
^10*c^14*d^4*f^4 - 60*b^10*c^10*d^8*f^4 - 24*b^10*c^16*d^2*f^4 - 24*b^10*c^
8*d^10*f^4 - 4*b^10*c^6*d^12*f^4 - 80*a^10*c^6*d^12*f^4 - 60*a^10*c^8*d^10*
f^4 - 60*a^10*c^4*d^14*f^4 - 24*a^10*c^10*d^8*f^4 - 24*a^10*c^2*d^16*f^4 -
4*a^10*c^12*d^6*f^4 - 8*a^8*b^2*d^18*f^4 - 4*a^6*b^4*d^18*f^4 - 8*a^2*b^8*c
^18*f^4 - 4*a^4*b^6*c^18*f^4 - 4*b^10*c^18*f^4 - 4*a^10*d^18*f^4 - 12*A*C*a
^7*b*c*d^11*f^2 - 12*A*C*a*b^7*c^11*d*f^2 - 912*B*C*a^4*b^4*c^5*d^7*f^2 + 7
92*B*C*a^5*b^3*c^4*d^8*f^2 - 792*B*C*a^3*b^5*c^8*d^4*f^2 + 720*B*C*a^4*b^4*
c^7*d^5*f^2 - 480*B*C*a^6*b^2*c^5*d^7*f^2 - 408*B*C*a^2*b^6*c^5*d^7*f^2 + 3
84*B*C*a^2*b^6*c^7*d^5*f^2 - 336*B*C*a^5*b^3*c^8*d^4*f^2 + 324*B*C*a^3*b^5*
c^4*d^8*f^2 + 312*B*C*a^6*b^2*c^7*d^5*f^2 - 248*B*C*a^6*b^2*c^3*d^9*f^2 + 2
16*B*C*a^2*b^6*c^9*d^3*f^2 - 196*B*C*a^4*b^4*c^3*d^9*f^2 + 132*B*C*a^4*b^4*
c^9*d^3*f^2 + 80*B*C*a^3*b^5*c^6*d^6*f^2 - 64*B*C*a^5*b^3*c^6*d^6*f^2 - 36*
B*C*a^3*b^5*c^2*d^10*f^2 - 28*B*C*a^2*b^6*c^3*d^9*f^2 + 12*B*C*a^5*b^3*c^10
*d^2*f^2 - 12*B*C*a^5*b^3*c^2*d^10*f^2 - 12*B*C*a^3*b^5*c^10*d^2*f^2 - 4*B*
C*a^6*b^2*c^9*d^3*f^2 - 1468*A*C*a^4*b^4*c^6*d^6*f^2 + 996*A*C*a^3*b^5*c^7*
d^5*f^2 + 900*A*C*a^5*b^3*c^5*d^7*f^2 - 676*A*C*a^6*b^2*c^6*d^6*f^2 - 660*A
*C*a^2*b^6*c^6*d^6*f^2 + 636*A*C*a^3*b^5*c^5*d^7*f^2 + 540*A*C*a^5*b^3*c^7*
d^5*f^2 - 236*A*C*a^5*b^3*c^3*d^9*f^2 - 204*A*C*a^3*b^5*c^9*d^3*f^2 + 156*A
*C*a^2*b^6*c^10*d^2*f^2 + 132*A*C*a^6*b^2*c^2*d^10*f^2 - 72*A*C*a^6*b^2*c^4
*d^8*f^2 - 72*A*C*a^5*b^3*c^9*d^3*f^2 + 66*A*C*a^2*b^6*c^4*d^8*f^2 + 54*A*C
*a^4*b^4*c^10*d^2*f^2 + 54*A*C*a^4*b^4*c^2*d^10*f^2 - 48*A*C*a^4*b^4*c^4*d^
8*f^2 - 48*A*C*a^2*b^6*c^8*d^4*f^2 + 42*A*C*a^6*b^2*c^8*d^4*f^2 - 40*A*C*a^
3*b^5*c^3*d^9*f^2 - 36*A*C*a^4*b^4*c^8*d^4*f^2 + 24*A*C*a^2*b^6*c^2*d^10*f^
2 + 960*A*B*a^4*b^4*c^5*d^7*f^2 - 864*A*B*a^5*b^3*c^4*d^8*f^2 + 756*A*B*a^3
*b^5*c^8*d^4*f^2 - 744*A*B*a^4*b^4*c^7*d^5*f^2 - 528*A*B*a^3*b^5*c^4*d^8*f^
2 + 504*A*B*a^6*b^2*c^5*d^7*f^2 - 432*A*B*a^2*b^6*c^7*d^5*f^2 + 432*A*B*a^2
*b^6*c^5*d^7*f^2 + 348*A*B*a^5*b^3*c^8*d^4*f^2 - 312*A*B*a^6*b^2*c^7*d^5*f^
2 - 284*A*B*a^2*b^6*c^9*d^3*f^2 + 280*A*B*a^6*b^2*c^3*d^9*f^2 + 264*A*B*a^4
*b^4*c^3*d^9*f^2 - 240*A*B*a^3*b^5*c^6*d^6*f^2 - 172*A*B*a^4*b^4*c^9*d^3*f^
2 + 68*A*B*a^2*b^6*c^3*d^9*f^2 - 60*A*B*a^3*b^5*c^2*d^10*f^2 + 24*A*B*a^5*b
^3*c^6*d^6*f^2 - 24*A*B*a^5*b^3*c^2*d^10*f^2 + 12*A*B*a^3*b^5*c^10*d^2*f^2
+ 360*B*C*a^7*b*c^4*d^8*f^2 - 336*B*C*a*b^7*c^8*d^4*f^2 + 168*B*C*a*b^7*c^6
*d^6*f^2 - 136*B*C*a^7*b*c^6*d^6*f^2 + 36*B*C*a^6*b^2*c*d^11*f^2 - 36*B*C*a
^2*b^6*c^11*d*f^2 - 24*B*C*a^7*b*c^2*d^10*f^2 + 24*B*C*a*b^7*c^10*d^2*f^2 -
12*B*C*a^4*b^4*c^11*d*f^2 + 12*B*C*a^4*b^4*c*d^11*f^2 + 12*B*C*a*b^7*c^4*d
^8*f^2 + 444*A*C*a*b^7*c^7*d^5*f^2 + 348*A*C*a^7*b*c^5*d^7*f^2 - 164*A*C*a^
7*b*c^3*d^9*f^2 - 132*A*C*a*b^7*c^9*d^3*f^2 + 84*A*C*a*b^7*c^5*d^7*f^2 + 32
*A*C*a*b^7*c^3*d^9*f^2 - 12*A*C*a^7*b*c^7*d^5*f^2 - 12*A*C*a^5*b^3*c*d^11*f
^2 - 12*A*C*a^3*b^5*c^11*d*f^2 - 360*A*B*a^7*b*c^4*d^8*f^2 + 288*A*B*a*b^7*
c^8*d^4*f^2 - 288*A*B*a*b^7*c^6*d^6*f^2 - 144*A*B*a*b^7*c^4*d^8*f^2 + 136*A
*B*a^7*b*c^6*d^6*f^2 - 60*A*B*a*b^7*c^2*d^10*f^2 - 36*A*B*a*b^7*c^10*d^2*f^
2 + 24*A*B*a^7*b*c^2*d^10*f^2 - 24*A*B*a^6*b^2*c*d^11*f^2 + 12*A*B*a^4*b^4*
c*d^11*f^2 + 12*A*B*a^2*b^6*c^11*d*f^2 + 12*A*B*a^2*b^6*c*d^11*f^2 + 80*B*C

*b^8*c^9*d^3*f^2 - 24*B*C*b^8*c^7*d^5*f^2 - 90*A*C*b^8*c^8*d^4*f^2 - 80*B*C
*a^8*c^3*d^9*f^2 + 54*A*C*b^8*c^10*d^2*f^2 - 30*A*C*b^8*c^6*d^6*f^2 + 24*B*
C*a^8*c^5*d^7*f^2 - 12*A*C*b^8*c^4*d^8*f^2 - 112*A*B*b^8*c^9*d^3*f^2 - 66*A
*C*a^8*c^4*d^8*f^2 + 54*A*C*a^8*c^2*d^10*f^2 - 8*B*C*a^5*b^3*d^12*f^2 - 8*B
*C*a^3*b^5*d^12*f^2 + 4*A*B*b^8*c^3*d^9*f^2 + 2*A*C*a^8*c^6*d^6*f^2 + 80*A*
B*a^8*c^3*d^9*f^2 - 24*A*B*a^8*c^5*d^7*f^2 + 8*A*C*a^2*b^6*d^12*f^2 - 4*B*C
*a^3*b^5*c^12*f^2 + 4*A*C*a^4*b^4*d^12*f^2 - 2*A*C*a^6*b^2*d^12*f^2 + 6*A*C
*a^2*b^6*c^12*f^2 + 4*A*B*a^5*b^3*d^12*f^2 - 4*A*B*a^3*b^5*d^12*f^2 + 726*C
^2*a^4*b^4*c^6*d^6*f^2 - 402*C^2*a^5*b^3*c^5*d^7*f^2 - 402*C^2*a^3*b^5*c^7*
d^5*f^2 + 322*C^2*a^6*b^2*c^6*d^6*f^2 + 322*C^2*a^2*b^6*c^6*d^6*f^2 - 222*C
^2*a^5*b^3*c^7*d^5*f^2 - 222*C^2*a^3*b^5*c^5*d^7*f^2 + 134*C^2*a^5*b^3*c^3*
d^9*f^2 + 134*C^2*a^3*b^5*c^9*d^3*f^2 - 66*C^2*a^6*b^2*c^2*d^10*f^2 - 66*C^
2*a^2*b^6*c^10*d^2*f^2 + 52*C^2*a^5*b^3*c^9*d^3*f^2 + 52*C^2*a^3*b^5*c^3*d^
9*f^2 - 27*C^2*a^6*b^2*c^8*d^4*f^2 - 27*C^2*a^2*b^6*c^4*d^8*f^2 + 24*C^2*a^
6*b^2*c^4*d^8*f^2 + 24*C^2*a^4*b^4*c^8*d^4*f^2 + 24*C^2*a^4*b^4*c^4*d^8*f^2
 + 24*C^2*a^2*b^6*c^8*d^4*f^2 - 15*C^2*a^4*b^4*c^10*d^2*f^2 - 15*C^2*a^4*b^
4*c^2*d^10*f^2 - 570*B^2*a^4*b^4*c^6*d^6*f^2 + 366*B^2*a^3*b^5*c^7*d^5*f^2
+ 318*B^2*a^5*b^3*c^5*d^7*f^2 - 262*B^2*a^6*b^2*c^6*d^6*f^2 - 222*B^2*a^2*b
^6*c^6*d^6*f^2 - 210*B^2*a^5*b^3*c^3*d^9*f^2 + 186*B^2*a^5*b^3*c^7*d^5*f^2
+ 162*B^2*a^3*b^5*c^5*d^7*f^2 - 142*B^2*a^3*b^5*c^9*d^3*f^2 + 132*B^2*a^4*b
^4*c^4*d^8*f^2 + 117*B^2*a^2*b^6*c^4*d^8*f^2 + 102*B^2*a^6*b^2*c^2*d^10*f^2
 - 96*B^2*a^3*b^5*c^3*d^9*f^2 + 90*B^2*a^2*b^6*c^10*d^2*f^2 + 81*B^2*a^4*b^
4*c^2*d^10*f^2 - 56*B^2*a^5*b^3*c^9*d^3*f^2 + 48*B^2*a^6*b^2*c^4*d^8*f^2 +
48*B^2*a^4*b^4*c^8*d^4*f^2 + 45*B^2*a^6*b^2*c^8*d^4*f^2 + 36*B^2*a^2*b^6*c^
8*d^4*f^2 + 36*B^2*a^2*b^6*c^2*d^10*f^2 + 33*B^2*a^4*b^4*c^10*d^2*f^2 + 822
*A^2*a^4*b^4*c^6*d^6*f^2 - 594*A^2*a^3*b^5*c^7*d^5*f^2 - 498*A^2*a^5*b^3*c^
5*d^7*f^2 + 498*A^2*a^2*b^6*c^6*d^6*f^2 - 414*A^2*a^3*b^5*c^5*d^7*f^2 + 354
*A^2*a^6*b^2*c^6*d^6*f^2 - 318*A^2*a^5*b^3*c^7*d^5*f^2 + 144*A^2*a^2*b^6*c^
8*d^4*f^2 + 102*A^2*a^5*b^3*c^3*d^9*f^2 + 84*A^2*a^4*b^4*c^4*d^8*f^2 + 81*A
^2*a^2*b^6*c^4*d^8*f^2 + 72*A^2*a^4*b^4*c^8*d^4*f^2 + 70*A^2*a^3*b^5*c^9*d^
3*f^2 - 66*A^2*a^6*b^2*c^2*d^10*f^2 + 48*A^2*a^6*b^2*c^4*d^8*f^2 - 42*A^2*a
^2*b^6*c^10*d^2*f^2 + 24*A^2*a^2*b^6*c^2*d^10*f^2 + 20*A^2*a^5*b^3*c^9*d^3*
f^2 - 15*A^2*a^6*b^2*c^8*d^4*f^2 - 15*A^2*a^4*b^4*c^10*d^2*f^2 - 15*A^2*a^4
*b^4*c^2*d^10*f^2 - 12*A^2*a^3*b^5*c^3*d^9*f^2 - 24*B*C*b^8*c^11*d*f^2 + 24
*B*C*a^8*c*d^11*f^2 + 12*A*B*b^8*c^11*d*f^2 - 8*B*C*a^7*b*d^12*f^2 - 24*A*B
*a^8*c*d^11*f^2 + 4*B*C*a*b^7*c^12*f^2 + 8*A*B*a^7*b*d^12*f^2 - 8*A*B*a*b^7
*d^12*f^2 - 8*A*B*a*b^7*c^12*f^2 - 174*C^2*a^7*b*c^5*d^7*f^2 - 174*C^2*a*b^
7*c^7*d^5*f^2 + 82*C^2*a^7*b*c^3*d^9*f^2 + 82*C^2*a*b^7*c^9*d^3*f^2 + 6*C^2
*a^7*b*c^7*d^5*f^2 + 6*C^2*a^5*b^3*c*d^11*f^2 + 6*C^2*a^3*b^5*c^11*d*f^2 +
6*C^2*a*b^7*c^5*d^7*f^2 + 162*B^2*a*b^7*c^7*d^5*f^2 + 138*B^2*a^7*b*c^5*d^7
*f^2 - 118*B^2*a^7*b*c^3*d^9*f^2 - 86*B^2*a*b^7*c^9*d^3*f^2 - 30*B^2*a^5*b^
3*c*d^11*f^2 - 18*B^2*a^7*b*c^7*d^5*f^2 - 18*B^2*a*b^7*c^5*d^7*f^2 - 12*B^2
*a^3*b^5*c*d^11*f^2 - 6*B^2*a^3*b^5*c^11*d*f^2 - 4*B^2*a*b^7*c^3*d^9*f^2 -
270*A^2*a*b^7*c^7*d^5*f^2 - 174*A^2*a^7*b*c^5*d^7*f^2 - 90*A^2*a*b^7*c^5*d^
7*f^2 + 82*A^2*a^7*b*c^3*d^9*f^2 + 50*A^2*a*b^7*c^9*d^3*f^2 - 32*A^2*a*b^7*

c^3*d^9*f^2 + 6*A^2*a^7*b*c^7*d^5*f^2 + 6*A^2*a^5*b^3*c*d^11*f^2 + 6*A^2*a^3*b^5*c^11*d*f^2 + 6*C^2*a^7*b*c*d^11*f^2 + 6*C^2*a*b^7*c^11*d*f^2 - 18*B^2*a^7*b*c*d^11*f^2 - 6*B^2*a*b^7*c^11*d*f^2 + 6*A^2*a^7*b*c*d^11*f^2 + 6*A^2*a*b^7*c^11*d*f^2 - 6*A*C*a^8*d^12*f^2 - 2*A*C*b^8*c^12*f^2 + 33*C^2*b^8*c^8*d^4*f^2 - 27*C^2*b^8*c^10*d^2*f^2 - C^2*b^8*c^6*d^6*f^2 + 33*C^2*a^8*c^4*d^8*f^2 + 33*B^2*b^8*c^10*d^2*f^2 - 27*C^2*a^8*c^2*d^10*f^2 - 27*B^2*b^8*c^8*d^4*f^2 + 3*B^2*b^8*c^6*d^6*f^2 - C^2*a^8*c^6*d^6*f^2 + 117*A^2*b^8*c^8*d^4*f^2 + 111*A^2*b^8*c^6*d^6*f^2 + 72*A^2*b^8*c^4*d^8*f^2 + 33*B^2*a^8*c^2*d^10*f^2 - 27*B^2*a^8*c^4*d^8*f^2 + 24*A^2*b^8*c^2*d^10*f^2 + 4*C^2*a^4*b^4*d^12*f^2 + 3*C^2*a^6*b^2*d^12*f^2 + 3*B^2*a^8*c^6*d^6*f^2 - 3*A^2*b^8*c^10*d^2*f^2 + 33*A^2*a^8*c^4*d^8*f^2 - 27*A^2*a^8*c^2*d^10*f^2 + 4*C^2*a^4*b^4*c^12*f^2 + 4*B^2*a^4*b^4*d^12*f^2 + 4*B^2*a^2*b^6*d^12*f^2 + 3*C^2*a^2*b^6*c^12*f^2 + 3*B^2*a^6*b^2*d^12*f^2 - A^2*a^8*c^6*d^6*f^2 - 4*A^2*a^4*b^4*d^12*f^2 + 3*B^2*a^2*b^6*c^12*f^2 - A^2*a^6*b^2*d^12*f^2 - A^2*a^2*b^6*c^12*f^2 + 3*C^2*b^8*c^12*f^2 + 3*C^2*a^8*d^12*f^2 + 4*A^2*b^8*d^12*f^2 - B^2*b^8*c^12*f^2 - B^2*a^8*d^12*f^2 + 3*A^2*b^8*c^12*f^2 + 3*A^2*a^8*d^12*f^2 - 24*A*B*C*a*b^6*c*d^8*f + 342*A*B*C*a^2*b^5*c^4*d^5*f - 186*A*B*C*a^3*b^4*c^5*d^4*f - 66*A*B*C*a^4*b^3*c^2*d^7*f + 48*A*B*C*a^2*b^5*c^2*d^7*f + 42*A*B*C*a^2*b^5*c^6*d^3*f + 26*A*B*C*a^5*b^2*c^3*d^6*f + 24*A*B*C*a^4*b^3*c^6*d^3*f - 18*A*B*C*a^4*b^3*c^4*d^5*f - 18*A*B*C*a^3*b^4*c^7*d^2*f - 8*A*B*C*a^3*b^4*c^3*d^6*f + 6*A*B*C*a^5*b^2*c^5*d^4*f - 128*A*B*C*a*b^6*c^3*d^6*f + 126*A*B*C*a*b^6*c^7*d^2*f + 72*A*B*C*a^3*b^4*c*d^8*f - 36*A*B*C*a^5*b^2*c*d^8*f - 36*A*B*C*a^2*b^5*c^8*d*f + 30*A*B*C*a^6*b*c^2*d^7*f - 12*A*B*C*a^6*b*c^4*d^5*f - 12*A*B*C*a*b^6*c^5*d^4*f - 21*B^2*C*a*b^6*c^8*d*f - 3*B^2*C*a^6*b*c*d^8*f + 21*A^2*C*a*b^6*c^8*d*f - 21*A*C^2*a*b^6*c^8*d*f - 9*A^2*C*a^6*b*c*d^8*f + 9*A*C^2*a^6*b*c*d^8*f + 36*A^2*B*a*b^6*c*d^8*f + 21*A*B^2*a*b^6*c^8*d*f + 3*A*B^2*a^6*b*c*d^8*f - 78*A*B*C*b^7*c^6*d^3*f + 24*A*B*C*b^7*c^4*d^5*f + 2*A*B*C*a^7*c^3*d^6*f + 16*A*B*C*a^4*b^3*d^9*f - 16*A*B*C*a^2*b^5*d^9*f - 237*B^2*C*a^3*b^4*c^4*d^5*f + 165*B*C^2*a^3*b^4*c^5*d^4*f + 92*B^2*C*a^2*b^5*c^3*d^6*f - 81*B^2*C*a^2*b^5*c^7*d^2*f + 77*B^2*C*a^4*b^3*c^3*d^6*f - 75*B*C^2*a^2*b^5*c^4*d^5*f + 69*B^2*C*a^4*b^3*c^5*d^4*f + 69*B*C^2*a^4*b^3*c^4*d^5*f - 68*B*C^2*a^3*b^4*c^3*d^6*f - 63*B^2*C*a^5*b^2*c^4*d^5*f - 61*B*C^2*a^2*b^5*c^6*d^3*f + 57*B*C^2*a^4*b^3*c^2*d^7*f - 53*B*C^2*a^5*b^2*c^3*d^6*f - 44*B*C^2*a^4*b^3*c^6*d^3*f - 36*B^2*C*a^3*b^4*c^2*d^7*f + 35*B^2*C*a^3*b^4*c^6*d^3*f + 33*B^2*C*a^5*b^2*c^2*d^7*f - 33*B^2*C*a^2*b^5*c^5*d^4*f + 33*B*C^2*a^3*b^4*c^7*d^2*f - 12*B^2*C*a^4*b^3*c^7*d^2*f + 9*B*C^2*a^5*b^2*c^5*d^4*f + 4*B^2*C*a^5*b^2*c^6*d^3*f + 225*A^2*C*a^2*b^5*c^5*d^4*f - 105*A*C^2*a^2*b^5*c^5*d^4*f - 99*A^2*C*a^3*b^4*c^4*d^5*f - 81*A^2*C*a^5*b^2*c^4*d^5*f + 67*A^2*C*a^4*b^3*c^3*d^6*f - 59*A*C^2*a^4*b^3*c^3*d^6*f + 57*A*C^2*a^5*b^2*c^2*d^7*f - 57*A*C^2*a^2*b^5*c^7*d^2*f + 51*A^2*C*a^4*b^3*c^5*d^4*f + 48*A^2*C*a^3*b^4*c^2*d^7*f + 45*A*C^2*a^5*b^2*c^4*d^5*f - 35*A^2*C*a^3*b^4*c^6*d^3*f - 33*A^2*C*a^5*b^2*c^2*d^7*f + 33*A^2*C*a^2*b^5*c^7*d^2*f + 33*A*C^2*a^4*b^3*c^5*d^4*f + 27*A*C^2*a^3*b^4*c^6*d^3*f - 24*A*C^2*a^3*b^4*c^2*d^7*f + 24*A*C^2*a^2*b^5*c^3*d^6*f - 21*A*C^2*a^3*b^4*c^4*d^5*f - 16*A^2*C*a^2*b^5*c^3*d^6*f - 243*A^2*B*a^2*b^5*c^4*d^5*f - 156*A*B^2*a^2*b^5*c^

3*d^6*f + 141*A*B^2*a^3*b^4*c^4*d^5*f + 108*A^2*B*a^3*b^4*c^3*d^6*f - 105*A
*B^2*a^4*b^3*c^3*d^6*f + 84*A*B^2*a^3*b^4*c^2*d^7*f + 81*A*B^2*a^2*b^5*c^5*
d^4*f - 51*A^2*B*a^4*b^3*c^4*d^5*f + 51*A^2*B*a^2*b^5*c^6*d^3*f - 48*A^2*B*
a^2*b^5*c^2*d^7*f + 45*A^2*B*a^3*b^4*c^5*d^4*f + 39*A*B^2*a^5*b^2*c^4*d^5*f
 - 35*A*B^2*a^3*b^4*c^6*d^3*f + 33*A*B^2*a^2*b^5*c^7*d^2*f + 27*A^2*B*a^5*b
^2*c^3*d^6*f - 21*A*B^2*a^4*b^3*c^5*d^4*f + 20*A^2*B*a^4*b^3*c^6*d^3*f - 15
*A^2*B*a^5*b^2*c^5*d^4*f - 15*A^2*B*a^3*b^4*c^7*d^2*f + 9*A^2*B*a^4*b^3*c^2
*d^7*f + 3*A*B^2*a^5*b^2*c^2*d^7*f + 18*A*B*C*b^7*c^8*d*f - 6*A*B*C*a^7*c*d
^8*f + 2*A*B*C*a^6*b*d^9*f - 6*A*B*C*a*b^6*c^9*f + 63*B^2*C*a*b^6*c^6*d^3*f
 - 48*B^2*C*a^4*b^3*c*d^8*f + 42*B*C^2*a^2*b^5*c^8*d*f + 42*B*C^2*a*b^6*c^5
*d^4*f - 39*B*C^2*a*b^6*c^7*d^2*f + 30*B*C^2*a^5*b^2*c*d^8*f - 24*B^2*C*a*b
^6*c^4*d^5*f - 24*B*C^2*a^3*b^4*c*d^8*f + 17*B^2*C*a^6*b*c^3*d^6*f - 15*B*C
^2*a^6*b*c^2*d^7*f + 12*B^2*C*a^3*b^4*c^8*d*f + 12*B^2*C*a^2*b^5*c*d^8*f +
6*B*C^2*a^6*b*c^4*d^5*f - 192*A^2*C*a*b^6*c^4*d^5*f - 99*A^2*C*a*b^6*c^6*d^
3*f + 84*A*C^2*a*b^6*c^4*d^5*f + 59*A*C^2*a*b^6*c^6*d^3*f + 51*A^2*C*a^6*b*
c^3*d^6*f - 51*A*C^2*a^6*b*c^3*d^6*f - 36*A^2*C*a^2*b^5*c*d^8*f - 24*A*C^2*
a^4*b^3*c*d^8*f + 24*A*C^2*a^2*b^5*c*d^8*f + 12*A^2*C*a^4*b^3*c*d^8*f + 12*
A*C^2*a^3*b^4*c^8*d*f + 160*A^2*B*a*b^6*c^3*d^6*f - 99*A*B^2*a*b^6*c^6*d^3*
f - 87*A^2*B*a*b^6*c^7*d^2*f - 72*A*B^2*a*b^6*c^4*d^5*f - 48*A*B^2*a^2*b^5*
c*d^8*f - 36*A^2*B*a^3*b^4*c*d^8*f + 24*A*B^2*a^4*b^3*c*d^8*f - 17*A*B^2*a^
6*b*c^3*d^6*f - 15*A^2*B*a^6*b*c^2*d^7*f + 12*A*B^2*a*b^6*c^2*d^7*f + 6*A^2
*B*a^6*b*c^4*d^5*f + 6*A^2*B*a^5*b^2*c*d^8*f + 6*A^2*B*a^2*b^5*c^8*d*f - 6*
A^2*B*a*b^6*c^5*d^4*f + 3*B^2*C*b^7*c^7*d^2*f - B*C^2*b^7*c^6*d^3*f + 96*A^
2*C*b^7*c^5*d^4*f - 39*A^2*C*b^7*c^7*d^2*f - 36*A*C^2*b^7*c^5*d^4*f + 32*A^
2*C*b^7*c^3*d^6*f + 15*A*C^2*b^7*c^7*d^2*f - 3*B^2*C*a^7*c^2*d^7*f - B*C^2*
a^7*c^3*d^6*f + 111*A^2*B*b^7*c^6*d^3*f - 39*A*B^2*b^7*c^7*d^2*f + 24*A*B^2
*b^7*c^5*d^4*f + 12*B^2*C*a^3*b^4*d^9*f - 12*B*C^2*a^4*b^3*d^9*f - 9*A^2*C*
a^7*c^2*d^7*f + 9*A*C^2*a^7*c^2*d^7*f - 4*A*B^2*b^7*c^3*d^6*f - 12*A^2*C*a^
3*b^4*d^9*f - 8*A*C^2*a^5*b^2*d^9*f + 8*A*C^2*a^3*b^4*d^9*f + 4*B^2*C*a^2*b
^5*c^9*f + 4*A^2*C*a^5*b^2*d^9*f - 4*B*C^2*a^3*b^4*c^9*f + 3*A*B^2*a^7*c^2*
d^7*f - A^2*B*a^7*c^3*d^6*f + 12*A^2*B*a^2*b^5*d^9*f - 8*A*B^2*a^3*b^4*d^9*
f - 4*A^2*B*a^4*b^3*d^9*f + 4*A*C^2*a^2*b^5*c^9*f - 3*C^3*a^6*b*c*d^8*f + 3
*C^3*a*b^6*c^8*d*f + 3*A^3*a^6*b*c*d^8*f - 3*A^3*a*b^6*c^8*d*f + 3*B*C^2*b^
7*c^8*d*f + 12*A^2*C*b^7*c*d^8*f + 3*B*C^2*a^7*c*d^8*f - 9*A^2*B*b^7*c^8*d*
f - B*C^2*a^6*b*d^9*f + 4*A^2*C*a*b^6*d^9*f + 3*A^2*B*a^7*c*d^8*f + 3*B*C^2
*a*b^6*c^9*f + 8*A*B^2*a*b^6*d^9*f - A^2*B*a^6*b*d^9*f - A^2*B*a*b^6*c^9*f
 - 39*C^3*a^4*b^3*c^5*d^4*f + 39*C^3*a^3*b^4*c^4*d^5*f - 27*C^3*a^5*b^2*c^2*
d^7*f + 27*C^3*a^2*b^5*c^7*d^2*f + 17*C^3*a^4*b^3*c^3*d^6*f - 17*C^3*a^3*b^
4*c^6*d^3*f - 3*C^3*a^5*b^2*c^4*d^5*f + 3*C^3*a^2*b^5*c^5*d^4*f - 63*B^3*a^
3*b^4*c^5*d^4*f + 57*B^3*a^2*b^5*c^4*d^5*f - 51*B^3*a^4*b^3*c^2*d^7*f + 48*
B^3*a^3*b^4*c^3*d^6*f + 31*B^3*a^2*b^5*c^6*d^3*f + 27*B^3*a^5*b^2*c^3*d^6*f
 + 16*B^3*a^4*b^3*c^6*d^3*f - 15*B^3*a^5*b^2*c^5*d^4*f - 12*B^3*a^2*b^5*c^2
*d^7*f + 9*B^3*a^4*b^3*c^4*d^5*f - 3*B^3*a^3*b^4*c^7*d^2*f - 123*A^3*a^2*b^
5*c^5*d^4*f + 81*A^3*a^3*b^4*c^4*d^5*f - 45*A^3*a^4*b^3*c^5*d^4*f + 39*A^3*
a^5*b^2*c^4*d^5*f - 25*A^3*a^4*b^3*c^3*d^6*f + 25*A^3*a^3*b^4*c^6*d^3*f - 2

4*A^3*a^3*b^4*c^2*d^7*f - 8*A^3*a^2*b^5*c^3*d^6*f + 3*A^3*a^5*b^2*c^2*d^7*f - 3*A^3*a^2*b^5*c^7*d^2*f + 17*C^3*a^6*b*c^3*d^6*f - 17*C^3*a*b^6*c^6*d^3*f + 12*C^3*a^4*b^3*c*d^8*f - 12*C^3*a^3*b^4*c^8*d*f + 24*B^3*a^3*b^4*c*d^8*f + 21*B^3*a*b^6*c^7*d^2*f - 18*B^3*a*b^6*c^5*d^4*f - 15*B^3*a^6*b*c^2*d^7*f + 6*B^3*a^6*b*c^4*d^5*f + 6*B^3*a^5*b^2*c*d^8*f - 6*B^3*a^2*b^5*c^8*d*f + 4*B^3*a*b^6*c^3*d^6*f + 108*A^3*a*b^6*c^4*d^5*f + 57*A^3*a*b^6*c^6*d^3*f - 17*A^3*a^6*b*c^3*d^6*f + 12*A^3*a^2*b^5*c*d^8*f + 3*C^3*b^7*c^7*d^2*f - 3*C^3*a^7*c^2*d^7*f - B^3*b^7*c^6*d^3*f - 60*A^3*b^7*c^5*d^4*f - 32*A^3*b^7*c^3*d^6*f + 21*A^3*b^7*c^7*d^2*f + 4*C^3*a^5*b^2*d^9*f - B^3*a^7*c^3*d^6*f - 4*C^3*a^2*b^5*c^9*f - 4*B^3*a^2*b^5*d^9*f + 3*A^3*a^7*c^2*d^7*f + 4*A^3*a^3*b^4*d^9*f + 3*B^3*b^7*c^8*d*f - 12*A^3*b^7*c*d^8*f + 3*B^3*a^7*c*d^8*f - B^3*a^6*b*d^9*f - 4*A^3*a*b^6*d^9*f - B^3*a*b^6*c^9*f - B^2*C*b^7*c^9*f - 4*A^2*B*b^7*d^9*f + 3*A^2*C*a^7*d^9*f - 3*A*C^2*a^7*d^9*f - A*C^2*b^7*c^9*f - A*B^2*a^7*d^9*f - C^3*b^7*c^9*f - A^3*a^7*d^9*f + B^2*C*a^7*d^9*f + A^2*C*b^7*c^9*f + A*B^2*b^7*c^9*f + C^3*a^7*d^9*f + A^3*b^7*c^9*f - 6*A*B^2*C*a*b^5*c^5*d - 21*A^2*B*C*a^2*b^4*c^3*d^3 + 21*A*B*C^2*a^2*b^4*c^3*d^3 + 12*A*B^2*C*a^2*b^4*c^4*d^2 - 12*A*B^2*C*a^2*b^4*c^2*d^4 - 10*A*B^2*C*a^3*b^3*c^3*d^3 - 6*A*B*C^2*a^3*b^3*c^4*d^2 + 3*A^2*B*C*a^3*b^3*c^4*d^2 + 3*A^2*B*C*a^3*b^3*c^2*d^4 + 3*A*B^2*C*a^4*b^2*c^2*d^4 + 3*A*B*C^2*a^3*b^3*c^2*d^4 + 2*A*B*C^2*a^4*b^2*c^3*d^3 - A^2*B*C*a^4*b^2*c^3*d^3 + 18*A^2*B*C*a*b^5*c^2*d^4 + 10*A*B^2*C*a*b^5*c^3*d^3 + 9*A^2*B*C*a*b^5*c^4*d^2 - 9*A*B*C^2*a*b^5*c^4*d^2 - 9*A*B*C^2*a*b^5*c^2*d^4 - 6*A^2*B*C*a^2*b^4*c*d^5 + 6*A*B^2*C*a^3*b^3*c*d^5 - 6*A*B*C^2*a^4*b^2*c*d^5 + 6*A*B*C^2*a^2*b^4*c^5*d + 3*A^2*B*C*a^4*b^2*c*d^5 - 3*A^2*B*C*a^2*b^4*c^5*d + 3*A*B*C^2*a^2*b^4*c*d^5 + 3*B^3*C*a^4*b^2*c*d^5 - 3*B^3*C*a^2*b^4*c^5*d + 3*B^3*C*a*b^5*c^4*d^2 + 3*B^2*C^2*a*b^5*c^5*d + 3*B*C^3*a^4*b^2*c*d^5 - 3*B*C^3*a^2*b^4*c^5*d + 3*B*C^3*a*b^5*c^4*d^2 + 24*A^3*C*a*b^5*c^3*d^3 + 8*A*C^3*a*b^5*c^3*d^3 - 9*A^3*B*a*b^5*c^2*d^4 - 9*A*B^3*a*b^5*c^2*d^4 + 3*A^3*B*a^2*b^4*c*d^5 - 3*A^3*B*a*b^5*c^4*d^2 + 3*A^2*B^2*a*b^5*c^5*d + 3*A*B^3*a^2*b^4*c*d^5 - 3*A*B^3*a*b^5*c^4*d^2 - 3*A*B^2*C*b^6*c^4*d^2 - 2*A^2*B*C*b^6*c^3*d^3 + 5*A*B*C^2*a^3*b^3*d^6 - 4*A^2*B*C*a^3*b^3*d^6 - A*B^2*C*a^4*b^2*d^6 + 9*B^2*C^2*a^3*b^3*c^3*d^3 - 6*B^2*C^2*a^2*b^4*c^4*d^2 + 6*B^2*C^2*a^2*b^4*c^2*d^4 - 3*B^2*C^2*a^4*b^2*c^2*d^4 + 24*A^2*C^2*a^3*b^3*c^3*d^3 - 15*A^2*C^2*a^2*b^4*c^4*d^2 - 9*A^2*C^2*a^4*b^2*c^2*d^4 + 3*A^2*C^2*a^2*b^4*c^2*d^4 + 9*A^2*B^2*a^2*b^4*c^2*d^4 - 3*A^2*B^2*a^2*b^4*c^4*d^2 + 6*A^2*B*C*b^6*c^5*d - 3*A*B*C^2*b^6*c^5*d + 4*A^2*B*C*a*b^5*d^6 - 2*A*B*C^2*a*b^5*d^6 + 2*A*B*C^2*a*b^5*c^6 - A^2*B*C*a*b^5*c^6 - 7*B^3*C*a^2*b^4*c^3*d^3 - 7*B*C^3*a^2*b^4*c^3*d^3 + 3*B^3*C*a^3*b^3*c^4*d^2 - 3*B^3*C*a^3*b^3*c^2*d^4 - 3*B^2*C^2*a^3*b^3*c*d^5 + 3*B*C^3*a^3*b^3*c^4*d^2 - 3*B*C^3*a^3*b^3*c^2*d^4 - B^3*C*a^4*b^2*c^3*d^3 - B^2*C^2*a*b^5*c^3*d^3 - B*C^3*a^4*b^2*c^3*d^3 - 24*A^2*C^2*a*b^5*c^3*d^3 - 24*A*C^3*a^3*b^3*c^3*d^3 + 12*A*C^3*a^2*b^4*c^4*d^2 + 9*A*C^3*a^4*b^2*c^2*d^4 - 8*A^3*C*a^3*b^3*c^3*d^3 + 6*A^3*C*a^2*b^4*c^4*d^2 - 6*A^3*C*a^2*b^4*c^2*d^4 + 3*A^3*C*a^4*b^2*c^2*d^4 - 9*A^2*B^2*a*b^5*c^3*d^3 + 7*A^3*B*a^2*b^4*c^3*d^3 + 7*A*B^3*a^2*b^4*c^3*d^3 - 3*A^3*B*a^3*b^3*c^2*d^4 - 3*A^2*B^2*a^3*b^3*c*d^5 - 3*A*B^3*a^3*b^3*c^2*d^4 + 12*A^2*C^2*b^6*c^4*d^2 + 3*A^2*C^2*b^6*c^2*d^4 +

6*A^2*B^2*b^6*c^4*d^2 + 3*A^2*B^2*b^6*c^2*d^4 - 5*A^2*C^2*a^2*b^4*d^6 + 3*A
^2*C^2*a^4*b^2*d^6 + A*B*C^2*b^6*c^3*d^3 - 3*B^4*a^3*b^3*c*d^5 - B^4*a*b^5*
c^3*d^3 + A^2*B^2*a^3*b^3*c^3*d^3 - 8*A^4*a*b^5*c^3*d^3 - 15*A^3*C*b^6*c^4*
d^2 - 6*A^3*C*b^6*c^2*d^4 - 3*A*C^3*b^6*c^4*d^2 - 2*B^3*C*a^3*b^3*d^6 - 2*B
*C^3*a^3*b^3*d^6 + 4*A^3*C*a^2*b^4*d^6 - 3*A*C^3*a^4*b^2*d^6 + 2*A*C^3*a^2*
b^4*d^6 - A^3*C*a^4*b^2*d^6 - 2*A*C^3*a^2*b^4*c^6 + 3*B^4*a*b^5*c^5*d - 3*A
^3*B*b^6*c^5*d - 3*A*B^3*b^6*c^5*d - B^3*C*a*b^5*c^6 - B*C^3*a*b^5*c^6 - 2*
A^3*B*a*b^5*d^6 - 2*A*B^3*a*b^5*d^6 + 8*C^4*a^3*b^3*c^3*d^3 - 3*C^4*a^4*b^2
*c^2*d^4 - 3*C^4*a^2*b^4*c^4*d^2 + 6*B^4*a^2*b^4*c^2*d^4 - 3*B^4*a^2*b^4*c^
4*d^2 + 3*A^4*a^2*b^4*c^2*d^4 + B^2*C^2*a^4*b^2*d^6 + B^2*C^2*a^2*b^4*d^6 +
 B^2*C^2*a^2*b^4*c^6 + A^2*C^2*a^2*b^4*c^6 - 2*A^3*C*b^6*d^6 + A^3*B*b^6*c^
3*d^3 + A*B^3*b^6*c^3*d^3 + A^3*B*a^3*b^3*d^6 + A*B^3*a^3*b^3*d^6 + 6*A^4*b
^6*c^4*d^2 + 3*A^4*b^6*c^2*d^4 - A^4*a^2*b^4*d^6 - 2*A^2*C^2*b^6*c^6 + A*B^
2*C*b^6*c^6 + B^4*a^3*b^3*c^3*d^3 + A^3*C*b^6*c^6 + A*C^3*b^6*c^6 + C^4*a^4
*b^2*d^6 + C^4*a^2*b^4*c^6 + B^4*a^2*b^4*d^6 + A^2*C^2*b^6*d^6 + A^2*B^2*b^
6*d^6 + A^4*b^6*d^6, f, k)*(root(480*a^9*b*c^7*d^11*f^4 + 480*a*b^9*c^11*d^
7*f^4 + 360*a^9*b*c^9*d^9*f^4 + 360*a^9*b*c^5*d^13*f^4 + 360*a*b^9*c^13*d^5
*f^4 + 360*a*b^9*c^9*d^9*f^4 + 144*a^9*b*c^11*d^7*f^4 + 144*a^9*b*c^3*d^15*
f^4 + 144*a*b^9*c^15*d^3*f^4 + 144*a*b^9*c^7*d^11*f^4 + 48*a^7*b^3*c*d^17*f
^4 + 48*a^3*b^7*c^17*d*f^4 + 24*a^9*b*c^13*d^5*f^4 + 24*a^5*b^5*c^17*d*f^4
+ 24*a^5*b^5*c*d^17*f^4 + 24*a*b^9*c^5*d^13*f^4 + 24*a^9*b*c*d^17*f^4 + 24*
a*b^9*c^17*d*f^4 + 3920*a^5*b^5*c^9*d^9*f^4 - 3360*a^6*b^4*c^8*d^10*f^4 - 3
360*a^4*b^6*c^10*d^8*f^4 - 3024*a^6*b^4*c^10*d^8*f^4 + 3024*a^5*b^5*c^11*d^
7*f^4 + 3024*a^5*b^5*c^7*d^11*f^4 - 3024*a^4*b^6*c^8*d^10*f^4 + 2320*a^7*b^
3*c^9*d^9*f^4 + 2320*a^3*b^7*c^9*d^9*f^4 - 2240*a^6*b^4*c^6*d^12*f^4 - 2240
*a^4*b^6*c^12*d^6*f^4 + 2160*a^7*b^3*c^7*d^11*f^4 + 2160*a^3*b^7*c^11*d^7*f
^4 - 1624*a^6*b^4*c^12*d^6*f^4 - 1624*a^4*b^6*c^6*d^12*f^4 + 1488*a^7*b^3*c
^11*d^7*f^4 + 1488*a^3*b^7*c^7*d^11*f^4 + 1344*a^5*b^5*c^13*d^5*f^4 + 1344*
a^5*b^5*c^5*d^13*f^4 - 1320*a^8*b^2*c^8*d^10*f^4 - 1320*a^2*b^8*c^10*d^8*f^
4 + 1200*a^7*b^3*c^5*d^13*f^4 + 1200*a^3*b^7*c^13*d^5*f^4 - 1060*a^8*b^2*c^
6*d^12*f^4 - 1060*a^2*b^8*c^12*d^6*f^4 - 948*a^8*b^2*c^10*d^8*f^4 - 948*a^2
*b^8*c^8*d^10*f^4 - 840*a^6*b^4*c^4*d^14*f^4 - 840*a^4*b^6*c^14*d^4*f^4 + 5
28*a^7*b^3*c^13*d^5*f^4 + 528*a^3*b^7*c^5*d^13*f^4 - 480*a^8*b^2*c^4*d^14*f
^4 - 480*a^6*b^4*c^14*d^4*f^4 - 480*a^4*b^6*c^4*d^14*f^4 - 480*a^2*b^8*c^14
*d^4*f^4 - 368*a^8*b^2*c^12*d^6*f^4 + 368*a^7*b^3*c^3*d^15*f^4 + 368*a^3*b^
7*c^15*d^3*f^4 - 368*a^2*b^8*c^6*d^12*f^4 + 304*a^5*b^5*c^15*d^3*f^4 + 304*
a^5*b^5*c^3*d^15*f^4 - 144*a^6*b^4*c^2*d^16*f^4 - 144*a^4*b^6*c^16*d^2*f^4
- 108*a^8*b^2*c^2*d^16*f^4 - 108*a^2*b^8*c^16*d^2*f^4 + 80*a^7*b^3*c^15*d^3
*f^4 + 80*a^3*b^7*c^3*d^15*f^4 - 60*a^8*b^2*c^14*d^4*f^4 - 60*a^6*b^4*c^16*
d^2*f^4 - 60*a^4*b^6*c^2*d^16*f^4 - 60*a^2*b^8*c^4*d^14*f^4 - 80*b^10*c^12*
d^6*f^4 - 60*b^10*c^14*d^4*f^4 - 60*b^10*c^10*d^8*f^4 - 24*b^10*c^16*d^2*f^
4 - 24*b^10*c^8*d^10*f^4 - 4*b^10*c^6*d^12*f^4 - 80*a^10*c^6*d^12*f^4 - 60*
a^10*c^8*d^10*f^4 - 60*a^10*c^4*d^14*f^4 - 24*a^10*c^10*d^8*f^4 - 24*a^10*c
^2*d^16*f^4 - 4*a^10*c^12*d^6*f^4 - 8*a^8*b^2*d^18*f^4 - 4*a^6*b^4*d^18*f^4
 - 8*a^2*b^8*c^18*f^4 - 4*a^4*b^6*c^18*f^4 - 4*b^10*c^18*f^4 - 4*a^10*d^18*

f^4 - 12*A*C*a^7*b*c*d^11*f^2 - 12*A*C*a*b^7*c^11*d*f^2 - 912*B*C*a^4*b^4*c
^5*d^7*f^2 + 792*B*C*a^5*b^3*c^4*d^8*f^2 - 792*B*C*a^3*b^5*c^8*d^4*f^2 + 72
0*B*C*a^4*b^4*c^7*d^5*f^2 - 480*B*C*a^6*b^2*c^5*d^7*f^2 - 408*B*C*a^2*b^6*c
^5*d^7*f^2 + 384*B*C*a^2*b^6*c^7*d^5*f^2 - 336*B*C*a^5*b^3*c^8*d^4*f^2 + 32
4*B*C*a^3*b^5*c^4*d^8*f^2 + 312*B*C*a^6*b^2*c^7*d^5*f^2 - 248*B*C*a^6*b^2*c
^3*d^9*f^2 + 216*B*C*a^2*b^6*c^9*d^3*f^2 - 196*B*C*a^4*b^4*c^3*d^9*f^2 + 13
2*B*C*a^4*b^4*c^9*d^3*f^2 + 80*B*C*a^3*b^5*c^6*d^6*f^2 - 64*B*C*a^5*b^3*c^6
*d^6*f^2 - 36*B*C*a^3*b^5*c^2*d^10*f^2 - 28*B*C*a^2*b^6*c^3*d^9*f^2 + 12*B*
C*a^5*b^3*c^10*d^2*f^2 - 12*B*C*a^5*b^3*c^2*d^10*f^2 - 12*B*C*a^3*b^5*c^10*
d^2*f^2 - 4*B*C*a^6*b^2*c^9*d^3*f^2 - 1468*A*C*a^4*b^4*c^6*d^6*f^2 + 996*A*
C*a^3*b^5*c^7*d^5*f^2 + 900*A*C*a^5*b^3*c^5*d^7*f^2 - 676*A*C*a^6*b^2*c^6*d
^6*f^2 - 660*A*C*a^2*b^6*c^6*d^6*f^2 + 636*A*C*a^3*b^5*c^5*d^7*f^2 + 540*A*
C*a^5*b^3*c^7*d^5*f^2 - 236*A*C*a^5*b^3*c^3*d^9*f^2 - 204*A*C*a^3*b^5*c^9*d
^3*f^2 + 156*A*C*a^2*b^6*c^10*d^2*f^2 + 132*A*C*a^6*b^2*c^2*d^10*f^2 - 72*A
*C*a^6*b^2*c^4*d^8*f^2 - 72*A*C*a^5*b^3*c^9*d^3*f^2 + 66*A*C*a^2*b^6*c^4*d^
8*f^2 + 54*A*C*a^4*b^4*c^10*d^2*f^2 + 54*A*C*a^4*b^4*c^2*d^10*f^2 - 48*A*C*
a^4*b^4*c^4*d^8*f^2 - 48*A*C*a^2*b^6*c^8*d^4*f^2 + 42*A*C*a^6*b^2*c^8*d^4*f
^2 - 40*A*C*a^3*b^5*c^3*d^9*f^2 - 36*A*C*a^4*b^4*c^8*d^4*f^2 + 24*A*C*a^2*b
^6*c^2*d^10*f^2 + 960*A*B*a^4*b^4*c^5*d^7*f^2 - 864*A*B*a^5*b^3*c^4*d^8*f^2
 + 756*A*B*a^3*b^5*c^8*d^4*f^2 - 744*A*B*a^4*b^4*c^7*d^5*f^2 - 528*A*B*a^3*
b^5*c^4*d^8*f^2 + 504*A*B*a^6*b^2*c^5*d^7*f^2 - 432*A*B*a^2*b^6*c^7*d^5*f^2
 + 432*A*B*a^2*b^6*c^5*d^7*f^2 + 348*A*B*a^5*b^3*c^8*d^4*f^2 - 312*A*B*a^6*
b^2*c^7*d^5*f^2 - 284*A*B*a^2*b^6*c^9*d^3*f^2 + 280*A*B*a^6*b^2*c^3*d^9*f^2
 + 264*A*B*a^4*b^4*c^3*d^9*f^2 - 240*A*B*a^3*b^5*c^6*d^6*f^2 - 172*A*B*a^4*
b^4*c^9*d^3*f^2 + 68*A*B*a^2*b^6*c^3*d^9*f^2 - 60*A*B*a^3*b^5*c^2*d^10*f^2
+ 24*A*B*a^5*b^3*c^6*d^6*f^2 - 24*A*B*a^5*b^3*c^2*d^10*f^2 + 12*A*B*a^3*b^5
*c^10*d^2*f^2 + 360*B*C*a^7*b*c^4*d^8*f^2 - 336*B*C*a*b^7*c^8*d^4*f^2 + 168
*B*C*a*b^7*c^6*d^6*f^2 - 136*B*C*a^7*b*c^6*d^6*f^2 + 36*B*C*a^6*b^2*c*d^11*
f^2 - 36*B*C*a^2*b^6*c^11*d*f^2 - 24*B*C*a^7*b*c^2*d^10*f^2 + 24*B*C*a*b^7*
c^10*d^2*f^2 - 12*B*C*a^4*b^4*c^11*d*f^2 + 12*B*C*a^4*b^4*c*d^11*f^2 + 12*B
*C*a*b^7*c^4*d^8*f^2 + 444*A*C*a*b^7*c^7*d^5*f^2 + 348*A*C*a^7*b*c^5*d^7*f^
2 - 164*A*C*a^7*b*c^3*d^9*f^2 - 132*A*C*a*b^7*c^9*d^3*f^2 + 84*A*C*a*b^7*c^
5*d^7*f^2 + 32*A*C*a*b^7*c^3*d^9*f^2 - 12*A*C*a^7*b*c^7*d^5*f^2 - 12*A*C*a^
5*b^3*c*d^11*f^2 - 12*A*C*a^3*b^5*c^11*d*f^2 - 360*A*B*a^7*b*c^4*d^8*f^2 +
288*A*B*a*b^7*c^8*d^4*f^2 - 288*A*B*a*b^7*c^6*d^6*f^2 - 144*A*B*a*b^7*c^4*d
^8*f^2 + 136*A*B*a^7*b*c^6*d^6*f^2 - 60*A*B*a*b^7*c^2*d^10*f^2 - 36*A*B*a*b
^7*c^10*d^2*f^2 + 24*A*B*a^7*b*c^2*d^10*f^2 - 24*A*B*a^6*b^2*c*d^11*f^2 + 1
2*A*B*a^4*b^4*c*d^11*f^2 + 12*A*B*a^2*b^6*c^11*d*f^2 + 12*A*B*a^2*b^6*c*d^1
1*f^2 + 80*B*C*b^8*c^9*d^3*f^2 - 24*B*C*b^8*c^7*d^5*f^2 - 90*A*C*b^8*c^8*d^
4*f^2 - 80*B*C*a^8*c^3*d^9*f^2 + 54*A*C*b^8*c^10*d^2*f^2 - 30*A*C*b^8*c^6*d
^6*f^2 + 24*B*C*a^8*c^5*d^7*f^2 - 12*A*C*b^8*c^4*d^8*f^2 - 112*A*B*b^8*c^9*
d^3*f^2 - 66*A*C*a^8*c^4*d^8*f^2 + 54*A*C*a^8*c^2*d^10*f^2 - 8*B*C*a^5*b^3*
d^12*f^2 - 8*B*C*a^3*b^5*d^12*f^2 + 4*A*B*b^8*c^3*d^9*f^2 + 2*A*C*a^8*c^6*d
^6*f^2 + 80*A*B*a^8*c^3*d^9*f^2 - 24*A*B*a^8*c^5*d^7*f^2 + 8*A*C*a^2*b^6*d^
12*f^2 - 4*B*C*a^3*b^5*c^12*f^2 + 4*A*C*a^4*b^4*d^12*f^2 - 2*A*C*a^6*b^2*d^

12*f^2 + 6*A*C*a^2*b^6*c^12*f^2 + 4*A*B*a^5*b^3*d^12*f^2 - 4*A*B*a^3*b^5*d^
12*f^2 + 726*C^2*a^4*b^4*c^6*d^6*f^2 - 402*C^2*a^5*b^3*c^5*d^7*f^2 - 402*C^
2*a^3*b^5*c^7*d^5*f^2 + 322*C^2*a^6*b^2*c^6*d^6*f^2 + 322*C^2*a^2*b^6*c^6*d
^6*f^2 - 222*C^2*a^5*b^3*c^7*d^5*f^2 - 222*C^2*a^3*b^5*c^5*d^7*f^2 + 134*C^
2*a^5*b^3*c^3*d^9*f^2 + 134*C^2*a^3*b^5*c^9*d^3*f^2 - 66*C^2*a^6*b^2*c^2*d^
10*f^2 - 66*C^2*a^2*b^6*c^10*d^2*f^2 + 52*C^2*a^5*b^3*c^9*d^3*f^2 + 52*C^2*
a^3*b^5*c^3*d^9*f^2 - 27*C^2*a^6*b^2*c^8*d^4*f^2 - 27*C^2*a^2*b^6*c^4*d^8*f
^2 + 24*C^2*a^6*b^2*c^4*d^8*f^2 + 24*C^2*a^4*b^4*c^8*d^4*f^2 + 24*C^2*a^4*b
^4*c^4*d^8*f^2 + 24*C^2*a^2*b^6*c^8*d^4*f^2 - 15*C^2*a^4*b^4*c^10*d^2*f^2 -
15*C^2*a^4*b^4*c^2*d^10*f^2 - 570*B^2*a^4*b^4*c^6*d^6*f^2 + 366*B^2*a^3*b^
5*c^7*d^5*f^2 + 318*B^2*a^5*b^3*c^5*d^7*f^2 - 262*B^2*a^6*b^2*c^6*d^6*f^2 -
222*B^2*a^2*b^6*c^6*d^6*f^2 - 210*B^2*a^5*b^3*c^3*d^9*f^2 + 186*B^2*a^5*b^
3*c^7*d^5*f^2 + 162*B^2*a^3*b^5*c^5*d^7*f^2 - 142*B^2*a^3*b^5*c^9*d^3*f^2 +
132*B^2*a^4*b^4*c^4*d^8*f^2 + 117*B^2*a^2*b^6*c^4*d^8*f^2 + 102*B^2*a^6*b^
2*c^2*d^10*f^2 - 96*B^2*a^3*b^5*c^3*d^9*f^2 + 90*B^2*a^2*b^6*c^10*d^2*f^2 +
81*B^2*a^4*b^4*c^2*d^10*f^2 - 56*B^2*a^5*b^3*c^9*d^3*f^2 + 48*B^2*a^6*b^2*
c^4*d^8*f^2 + 48*B^2*a^4*b^4*c^8*d^4*f^2 + 45*B^2*a^6*b^2*c^8*d^4*f^2 + 36*
B^2*a^2*b^6*c^8*d^4*f^2 + 36*B^2*a^2*b^6*c^2*d^10*f^2 + 33*B^2*a^4*b^4*c^10
*d^2*f^2 + 822*A^2*a^4*b^4*c^6*d^6*f^2 - 594*A^2*a^3*b^5*c^7*d^5*f^2 - 498*
A^2*a^5*b^3*c^5*d^7*f^2 + 498*A^2*a^2*b^6*c^6*d^6*f^2 - 414*A^2*a^3*b^5*c^5
*d^7*f^2 + 354*A^2*a^6*b^2*c^6*d^6*f^2 - 318*A^2*a^5*b^3*c^7*d^5*f^2 + 144*
A^2*a^2*b^6*c^8*d^4*f^2 + 102*A^2*a^5*b^3*c^3*d^9*f^2 + 84*A^2*a^4*b^4*c^4*
d^8*f^2 + 81*A^2*a^2*b^6*c^4*d^8*f^2 + 72*A^2*a^4*b^4*c^8*d^4*f^2 + 70*A^2*
a^3*b^5*c^9*d^3*f^2 - 66*A^2*a^6*b^2*c^2*d^10*f^2 + 48*A^2*a^6*b^2*c^4*d^8*
f^2 - 42*A^2*a^2*b^6*c^10*d^2*f^2 + 24*A^2*a^2*b^6*c^2*d^10*f^2 + 20*A^2*a^
5*b^3*c^9*d^3*f^2 - 15*A^2*a^6*b^2*c^8*d^4*f^2 - 15*A^2*a^4*b^4*c^10*d^2*f^
2 - 15*A^2*a^4*b^4*c^2*d^10*f^2 - 12*A^2*a^3*b^5*c^3*d^9*f^2 - 24*B*C*b^8*c
^11*d*f^2 + 24*B*C*a^8*c*d^11*f^2 + 12*A*B*b^8*c^11*d*f^2 - 8*B*C*a^7*b*d^1
2*f^2 - 24*A*B*a^8*c*d^11*f^2 + 4*B*C*a*b^7*c^12*f^2 + 8*A*B*a^7*b*d^12*f^2
- 8*A*B*a*b^7*d^12*f^2 - 8*A*B*a*b^7*c^12*f^2 - 174*C^2*a^7*b*c^5*d^7*f^2
- 174*C^2*a*b^7*c^7*d^5*f^2 + 82*C^2*a^7*b*c^3*d^9*f^2 + 82*C^2*a*b^7*c^9*d
^3*f^2 + 6*C^2*a^7*b*c^7*d^5*f^2 + 6*C^2*a^5*b^3*c*d^11*f^2 + 6*C^2*a^3*b^5
*c^11*d*f^2 + 6*C^2*a*b^7*c^5*d^7*f^2 + 162*B^2*a*b^7*c^7*d^5*f^2 + 138*B^2
*a^7*b*c^5*d^7*f^2 - 118*B^2*a^7*b*c^3*d^9*f^2 - 86*B^2*a*b^7*c^9*d^3*f^2 -
30*B^2*a^5*b^3*c*d^11*f^2 - 18*B^2*a^7*b*c^7*d^5*f^2 - 18*B^2*a*b^7*c^5*d^
7*f^2 - 12*B^2*a^3*b^5*c*d^11*f^2 - 6*B^2*a^3*b^5*c^11*d*f^2 - 4*B^2*a*b^7*
c^3*d^9*f^2 - 270*A^2*a*b^7*c^7*d^5*f^2 - 174*A^2*a^7*b*c^5*d^7*f^2 - 90*A^
2*a*b^7*c^5*d^7*f^2 + 82*A^2*a^7*b*c^3*d^9*f^2 + 50*A^2*a*b^7*c^9*d^3*f^2 -
32*A^2*a*b^7*c^3*d^9*f^2 + 6*A^2*a^7*b*c^7*d^5*f^2 + 6*A^2*a^5*b^3*c*d^11*
f^2 + 6*A^2*a^3*b^5*c^11*d*f^2 + 6*C^2*a^7*b*c*d^11*f^2 + 6*C^2*a*b^7*c^11*
d*f^2 - 18*B^2*a^7*b*c*d^11*f^2 - 6*B^2*a*b^7*c^11*d*f^2 + 6*A^2*a^7*b*c*d^
11*f^2 + 6*A^2*a*b^7*c^11*d*f^2 - 6*A*C*a^8*d^12*f^2 - 2*A*C*b^8*c^12*f^2 +
33*C^2*b^8*c^8*d^4*f^2 - 27*C^2*b^8*c^10*d^2*f^2 - C^2*b^8*c^6*d^6*f^2 + 3
3*C^2*a^8*c^4*d^8*f^2 + 33*B^2*b^8*c^10*d^2*f^2 - 27*C^2*a^8*c^2*d^10*f^2 -
27*B^2*b^8*c^8*d^4*f^2 + 3*B^2*b^8*c^6*d^6*f^2 - C^2*a^8*c^6*d^6*f^2 + 117

*A^2*b^8*c^8*d^4*f^2 + 111*A^2*b^8*c^6*d^6*f^2 + 72*A^2*b^8*c^4*d^8*f^2 + 3
3*B^2*a^8*c^2*d^10*f^2 - 27*B^2*a^8*c^4*d^8*f^2 + 24*A^2*b^8*c^2*d^10*f^2 +
4*C^2*a^4*b^4*d^12*f^2 + 3*C^2*a^6*b^2*d^12*f^2 + 3*B^2*a^8*c^6*d^6*f^2 -
3*A^2*b^8*c^10*d^2*f^2 + 33*A^2*a^8*c^4*d^8*f^2 - 27*A^2*a^8*c^2*d^10*f^2 +
4*C^2*a^4*b^4*c^12*f^2 + 4*B^2*a^4*b^4*d^12*f^2 + 4*B^2*a^2*b^6*d^12*f^2 +
3*C^2*a^2*b^6*c^12*f^2 + 3*B^2*a^6*b^2*d^12*f^2 - A^2*a^8*c^6*d^6*f^2 - 4*
A^2*a^4*b^4*d^12*f^2 + 3*B^2*a^2*b^6*c^12*f^2 - A^2*a^6*b^2*d^12*f^2 - A^2*
a^2*b^6*c^12*f^2 + 3*C^2*b^8*c^12*f^2 + 3*C^2*a^8*d^12*f^2 + 4*A^2*b^8*d^12
*f^2 - B^2*b^8*c^12*f^2 - B^2*a^8*d^12*f^2 + 3*A^2*b^8*c^12*f^2 + 3*A^2*a^8
*d^12*f^2 - 24*A*B*C*a*b^6*c*d^8*f + 342*A*B*C*a^2*b^5*c^4*d^5*f - 186*A*B*
C*a^3*b^4*c^5*d^4*f - 66*A*B*C*a^4*b^3*c^2*d^7*f + 48*A*B*C*a^2*b^5*c^2*d^7
*f + 42*A*B*C*a^2*b^5*c^6*d^3*f + 26*A*B*C*a^5*b^2*c^3*d^6*f + 24*A*B*C*a^4
*b^3*c^6*d^3*f - 18*A*B*C*a^4*b^3*c^4*d^5*f - 18*A*B*C*a^3*b^4*c^7*d^2*f -
8*A*B*C*a^3*b^4*c^3*d^6*f + 6*A*B*C*a^5*b^2*c^5*d^4*f - 128*A*B*C*a*b^6*c^3
*d^6*f + 126*A*B*C*a*b^6*c^7*d^2*f + 72*A*B*C*a^3*b^4*c*d^8*f - 36*A*B*C*a^
5*b^2*c*d^8*f - 36*A*B*C*a^2*b^5*c^8*d*f + 30*A*B*C*a^6*b*c^2*d^7*f - 12*A*
B*C*a^6*b*c^4*d^5*f - 12*A*B*C*a*b^6*c^5*d^4*f - 21*B^2*C*a*b^6*c^8*d*f - 3
*B^2*C*a^6*b*c*d^8*f + 21*A^2*C*a*b^6*c^8*d*f - 21*A*C^2*a*b^6*c^8*d*f - 9*
A^2*C*a^6*b*c*d^8*f + 9*A*C^2*a^6*b*c*d^8*f + 36*A^2*B*a*b^6*c*d^8*f + 21*A
*B^2*a*b^6*c^8*d*f + 3*A*B^2*a^6*b*c*d^8*f - 78*A*B*C*b^7*c^6*d^3*f + 24*A*
B*C*b^7*c^4*d^5*f + 2*A*B*C*a^7*c^3*d^6*f + 16*A*B*C*a^4*b^3*d^9*f - 16*A*B
*C*a^2*b^5*d^9*f - 237*B^2*C*a^3*b^4*c^4*d^5*f + 165*B*C^2*a^3*b^4*c^5*d^4*
f + 92*B^2*C*a^2*b^5*c^3*d^6*f - 81*B^2*C*a^2*b^5*c^7*d^2*f + 77*B^2*C*a^4*
b^3*c^3*d^6*f - 75*B*C^2*a^2*b^5*c^4*d^5*f + 69*B^2*C*a^4*b^3*c^5*d^4*f + 6
9*B*C^2*a^4*b^3*c^4*d^5*f - 68*B*C^2*a^3*b^4*c^3*d^6*f - 63*B^2*C*a^5*b^2*c
^4*d^5*f - 61*B*C^2*a^2*b^5*c^6*d^3*f + 57*B*C^2*a^4*b^3*c^2*d^7*f - 53*B*C
^2*a^5*b^2*c^3*d^6*f - 44*B*C^2*a^4*b^3*c^6*d^3*f - 36*B^2*C*a^3*b^4*c^2*d^
7*f + 35*B^2*C*a^3*b^4*c^6*d^3*f + 33*B^2*C*a^5*b^2*c^2*d^7*f - 33*B^2*C*a^
2*b^5*c^5*d^4*f + 33*B*C^2*a^3*b^4*c^7*d^2*f - 12*B^2*C*a^4*b^3*c^7*d^2*f +
9*B*C^2*a^5*b^2*c^5*d^4*f + 4*B^2*C*a^5*b^2*c^6*d^3*f + 225*A^2*C*a^2*b^5*
c^5*d^4*f - 105*A*C^2*a^2*b^5*c^5*d^4*f - 99*A^2*C*a^3*b^4*c^4*d^5*f - 81*A
^2*C*a^5*b^2*c^4*d^5*f + 67*A^2*C*a^4*b^3*c^3*d^6*f - 59*A*C^2*a^4*b^3*c^3*
d^6*f + 57*A*C^2*a^5*b^2*c^2*d^7*f - 57*A*C^2*a^2*b^5*c^7*d^2*f + 51*A^2*C*
a^4*b^3*c^5*d^4*f + 48*A^2*C*a^3*b^4*c^2*d^7*f + 45*A*C^2*a^5*b^2*c^4*d^5*f
- 35*A^2*C*a^3*b^4*c^6*d^3*f - 33*A^2*C*a^5*b^2*c^2*d^7*f + 33*A^2*C*a^2*b
^5*c^7*d^2*f + 33*A*C^2*a^4*b^3*c^5*d^4*f + 27*A*C^2*a^3*b^4*c^6*d^3*f - 24
*A*C^2*a^3*b^4*c^2*d^7*f + 24*A*C^2*a^2*b^5*c^3*d^6*f - 21*A*C^2*a^3*b^4*c^
4*d^5*f - 16*A^2*C*a^2*b^5*c^3*d^6*f - 243*A^2*B*a^2*b^5*c^4*d^5*f - 156*A*
B^2*a^2*b^5*c^3*d^6*f + 141*A*B^2*a^3*b^4*c^4*d^5*f + 108*A^2*B*a^3*b^4*c^3
*d^6*f - 105*A*B^2*a^4*b^3*c^3*d^6*f + 84*A*B^2*a^3*b^4*c^2*d^7*f + 81*A*B^
2*a^2*b^5*c^5*d^4*f - 51*A^2*B*a^4*b^3*c^4*d^5*f + 51*A^2*B*a^2*b^5*c^6*d^3
*f - 48*A^2*B*a^2*b^5*c^2*d^7*f + 45*A^2*B*a^3*b^4*c^5*d^4*f + 39*A*B^2*a^5
*b^2*c^4*d^5*f - 35*A*B^2*a^3*b^4*c^6*d^3*f + 33*A*B^2*a^2*b^5*c^7*d^2*f +
27*A^2*B*a^5*b^2*c^3*d^6*f - 21*A*B^2*a^4*b^3*c^5*d^4*f + 20*A^2*B*a^4*b^3*
c^6*d^3*f - 15*A^2*B*a^5*b^2*c^5*d^4*f - 15*A^2*B*a^3*b^4*c^7*d^2*f + 9*A^2

*B*a^4*b^3*c^2*d^7*f + 3*A*B^2*a^5*b^2*c^2*d^7*f + 18*A*B*C*b^7*c^8*d*f - 6
*A*B*C*a^7*c*d^8*f + 2*A*B*C*a^6*b*d^9*f - 6*A*B*C*a*b^6*c^9*f + 63*B^2*C*a
*b^6*c^6*d^3*f - 48*B^2*C*a^4*b^3*c*d^8*f + 42*B*C^2*a^2*b^5*c^8*d*f + 42*B
*C^2*a*b^6*c^5*d^4*f - 39*B*C^2*a*b^6*c^7*d^2*f + 30*B*C^2*a^5*b^2*c*d^8*f
- 24*B^2*C*a*b^6*c^4*d^5*f - 24*B*C^2*a^3*b^4*c*d^8*f + 17*B^2*C*a^6*b*c^3*
d^6*f - 15*B*C^2*a^6*b*c^2*d^7*f + 12*B^2*C*a^3*b^4*c^8*d*f + 12*B^2*C*a^2*
b^5*c*d^8*f + 6*B*C^2*a^6*b*c^4*d^5*f - 192*A^2*C*a*b^6*c^4*d^5*f - 99*A^2*
C*a*b^6*c^6*d^3*f + 84*A*C^2*a*b^6*c^4*d^5*f + 59*A*C^2*a*b^6*c^6*d^3*f + 5
1*A^2*C*a^6*b*c^3*d^6*f - 51*A*C^2*a^6*b*c^3*d^6*f - 36*A^2*C*a^2*b^5*c*d^8
*f - 24*A*C^2*a^4*b^3*c*d^8*f + 24*A*C^2*a^2*b^5*c*d^8*f + 12*A^2*C*a^4*b^3
*c*d^8*f + 12*A*C^2*a^3*b^4*c^8*d*f + 160*A^2*B*a*b^6*c^3*d^6*f - 99*A*B^2*
a*b^6*c^6*d^3*f - 87*A^2*B*a*b^6*c^7*d^2*f - 72*A*B^2*a*b^6*c^4*d^5*f - 48*
A*B^2*a^2*b^5*c*d^8*f - 36*A^2*B*a^3*b^4*c*d^8*f + 24*A*B^2*a^4*b^3*c*d^8*f
 - 17*A*B^2*a^6*b*c^3*d^6*f - 15*A^2*B*a^6*b*c^2*d^7*f + 12*A*B^2*a*b^6*c^2
*d^7*f + 6*A^2*B*a^6*b*c^4*d^5*f + 6*A^2*B*a^5*b^2*c*d^8*f + 6*A^2*B*a^2*b^
5*c^8*d*f - 6*A^2*B*a*b^6*c^5*d^4*f + 3*B^2*C*b^7*c^7*d^2*f - B*C^2*b^7*c^6
*d^3*f + 96*A^2*C*b^7*c^5*d^4*f - 39*A^2*C*b^7*c^7*d^2*f - 36*A*C^2*b^7*c^5
*d^4*f + 32*A^2*C*b^7*c^3*d^6*f + 15*A*C^2*b^7*c^7*d^2*f - 3*B^2*C*a^7*c^2*
d^7*f - B*C^2*a^7*c^3*d^6*f + 111*A^2*B*b^7*c^6*d^3*f - 39*A*B^2*b^7*c^7*d^
2*f + 24*A*B^2*b^7*c^5*d^4*f + 12*B^2*C*a^3*b^4*d^9*f - 12*B*C^2*a^4*b^3*d^
9*f - 9*A^2*C*a^7*c^2*d^7*f + 9*A*C^2*a^7*c^2*d^7*f - 4*A*B^2*b^7*c^3*d^6*f
 - 12*A^2*C*a^3*b^4*d^9*f - 8*A*C^2*a^5*b^2*d^9*f + 8*A*C^2*a^3*b^4*d^9*f +
 4*B^2*C*a^2*b^5*c^9*f + 4*A^2*C*a^5*b^2*d^9*f - 4*B*C^2*a^3*b^4*c^9*f + 3*
A*B^2*a^7*c^2*d^7*f - A^2*B*a^7*c^3*d^6*f + 12*A^2*B*a^2*b^5*d^9*f - 8*A*B^
2*a^3*b^4*d^9*f - 4*A^2*B*a^4*b^3*d^9*f + 4*A*C^2*a^2*b^5*c^9*f - 3*C^3*a^6
*b*c*d^8*f + 3*C^3*a*b^6*c^8*d*f + 3*A^3*a^6*b*c*d^8*f - 3*A^3*a*b^6*c^8*d*
f + 3*B*C^2*b^7*c^8*d*f + 12*A^2*C*b^7*c*d^8*f + 3*B*C^2*a^7*c*d^8*f - 9*A^
2*B*b^7*c^8*d*f - B*C^2*a^6*b*d^9*f + 4*A^2*C*a*b^6*d^9*f + 3*A^2*B*a^7*c*d
^8*f + 3*B*C^2*a*b^6*c^9*f + 8*A*B^2*a*b^6*d^9*f - A^2*B*a^6*b*d^9*f - A^2*
B*a*b^6*c^9*f - 39*C^3*a^4*b^3*c^5*d^4*f + 39*C^3*a^3*b^4*c^4*d^5*f - 27*C^
3*a^5*b^2*c^2*d^7*f + 27*C^3*a^2*b^5*c^7*d^2*f + 17*C^3*a^4*b^3*c^3*d^6*f -
 17*C^3*a^3*b^4*c^6*d^3*f - 3*C^3*a^5*b^2*c^4*d^5*f + 3*C^3*a^2*b^5*c^5*d^4
*f - 63*B^3*a^3*b^4*c^5*d^4*f + 57*B^3*a^2*b^5*c^4*d^5*f - 51*B^3*a^4*b^3*c
^2*d^7*f + 48*B^3*a^3*b^4*c^3*d^6*f + 31*B^3*a^2*b^5*c^6*d^3*f + 27*B^3*a^5
*b^2*c^3*d^6*f + 16*B^3*a^4*b^3*c^6*d^3*f - 15*B^3*a^5*b^2*c^5*d^4*f - 12*B
^3*a^2*b^5*c^2*d^7*f + 9*B^3*a^4*b^3*c^4*d^5*f - 3*B^3*a^3*b^4*c^7*d^2*f -
123*A^3*a^2*b^5*c^5*d^4*f + 81*A^3*a^3*b^4*c^4*d^5*f - 45*A^3*a^4*b^3*c^5*d
^4*f + 39*A^3*a^5*b^2*c^4*d^5*f - 25*A^3*a^4*b^3*c^3*d^6*f + 25*A^3*a^3*b^4
*c^6*d^3*f - 24*A^3*a^3*b^4*c^2*d^7*f - 8*A^3*a^2*b^5*c^3*d^6*f + 3*A^3*a^5
*b^2*c^2*d^7*f - 3*A^3*a^2*b^5*c^7*d^2*f + 17*C^3*a^6*b*c^3*d^6*f - 17*C^3*
a*b^6*c^6*d^3*f + 12*C^3*a^4*b^3*c*d^8*f - 12*C^3*a^3*b^4*c^8*d*f + 24*B^3*
a^3*b^4*c*d^8*f + 21*B^3*a*b^6*c^7*d^2*f - 18*B^3*a*b^6*c^5*d^4*f - 15*B^3*
a^6*b*c^2*d^7*f + 6*B^3*a^6*b*c^4*d^5*f + 6*B^3*a^5*b^2*c*d^8*f - 6*B^3*a^2
*b^5*c^8*d*f + 4*B^3*a*b^6*c^3*d^6*f + 108*A^3*a*b^6*c^4*d^5*f + 57*A^3*a*b
^6*c^6*d^3*f - 17*A^3*a^6*b*c^3*d^6*f + 12*A^3*a^2*b^5*c*d^8*f + 3*C^3*b^7*

c^7*d^2*f - 3*C^3*a^7*c^2*d^7*f - B^3*b^7*c^6*d^3*f - 60*A^3*b^7*c^5*d^4*f - 32*A^3*b^7*c^3*d^6*f + 21*A^3*b^7*c^7*d^2*f + 4*C^3*a^5*b^2*d^9*f - B^3*a^7*c^3*d^6*f - 4*C^3*a^2*b^5*c^9*f - 4*B^3*a^2*b^5*d^9*f + 3*A^3*a^7*c^2*d^7*f + 4*A^3*a^3*b^4*d^9*f + 3*B^3*b^7*c^8*d*f - 12*A^3*b^7*c*d^8*f + 3*B^3*a^7*c*d^8*f - B^3*a^6*b*d^9*f - 4*A^3*a*b^6*d^9*f - B^3*a*b^6*c^9*f - B^2*C*b^7*c^9*f - 4*A^2*B*b^7*d^9*f + 3*A^2*C*a^7*d^9*f - 3*A*C^2*a^7*d^9*f - A*C^2*b^7*c^9*f - A*B^2*a^7*d^9*f - C^3*b^7*c^9*f - A^3*a^7*d^9*f + B^2*C*a^7*d^9*f + A^2*C*b^7*c^9*f + A*B^2*b^7*c^9*f + C^3*a^7*d^9*f + A^3*b^7*c^9*f - 6*A*B^2*C*a*b^5*c^5*d - 21*A^2*B*C*a^2*b^4*c^3*d^3 + 21*A*B*C^2*a^2*b^4*c^3*d^3 + 12*A*B^2*C*a^2*b^4*c^4*d^2 - 12*A*B^2*C*a^2*b^4*c^2*d^4 - 10*A*B^2*C*a^3*b^3*c^3*d^3 - 6*A*B*C^2*a^3*b^3*c^4*d^2 + 3*A^2*B*C*a^3*b^3*c^4*d^2 + 3*A^2*B*C*a^3*b^3*c^2*d^4 + 3*A*B^2*C*a^4*b^2*c^2*d^4 + 3*A*B*C^2*a^3*b^3*c^2*d^4 + 2*A*B*C^2*a^4*b^2*c^3*d^3 - A^2*B*C*a^4*b^2*c^3*d^3 + 18*A^2*B*C*a*b^5*c^2*d^4 + 10*A*B^2*C*a*b^5*c^3*d^3 + 9*A^2*B*C*a*b^5*c^4*d^2 - 9*A*B*C^2*a*b^5*c^4*d^2 - 9*A*B*C^2*a*b^5*c^2*d^4 - 6*A^2*B*C*a^2*b^4*c*d^5 + 6*A*B^2*C*a^3*b^3*c*d^5 - 6*A*B*C^2*a^4*b^2*c*d^5 + 6*A*B*C^2*a^2*b^4*c^5*d + 3*A^2*B*C*a^4*b^2*c*d^5 - 3*A^2*B*C*a^2*b^4*c^5*d + 3*A*B*C^2*a^2*b^4*c*d^5 + 3*B^3*C*a^4*b^2*c*d^5 - 3*B^3*C*a^2*b^4*c^5*d + 3*B^3*C*a*b^5*c^4*d^2 + 3*B^2*C^2*a*b^5*c^5*d + 3*B*C^3*a^4*b^2*c*d^5 - 3*B*C^3*a^2*b^4*c^5*d + 3*B*C^3*a*b^5*c^4*d^2 + 24*A^3*C*a*b^5*c^3*d^3 + 8*A*C^3*a*b^5*c^3*d^3 - 9*A^3*B*a*b^5*c^2*d^4 - 9*A*B^3*a*b^5*c^2*d^4 + 3*A^3*B*a^2*b^4*c*d^5 - 3*A^3*B*a*b^5*c^4*d^2 + 3*A^2*B^2*a*b^5*c^5*d + 3*A*B^3*a^2*b^4*c*d^5 - 3*A*B^3*a*b^5*c^4*d^2 - 3*A*B^2*C*b^6*c^4*d^2 - 2*A^2*B*C*b^6*c^3*d^3 + 5*A*B*C^2*a^3*b^3*d^6 - 4*A^2*B*C*a^3*b^3*d^6 - A*B^2*C*a^4*b^2*d^6 + 9*B^2*C^2*a^3*b^3*c^3*d^3 - 6*B^2*C^2*a^2*b^4*c^4*d^2 + 6*B^2*C^2*a^2*b^4*c^2*d^4 - 3*B^2*C^2*a^4*b^2*c^2*d^4 + 24*A^2*C^2*a^3*b^3*c^3*d^3 - 15*A^2*C^2*a^2*b^4*c^4*d^2 - 9*A^2*C^2*a^4*b^2*c^2*d^4 + 3*A^2*C^2*a^2*b^4*c^2*d^4 + 9*A^2*B^2*a^2*b^4*c^2*d^4 - 3*A^2*B^2*a^2*b^4*c^4*d^2 + 6*A^2*B*C*b^6*c^5*d - 3*A*B*C^2*b^6*c^5*d + 4*A^2*B*C*a*b^5*d^6 - 2*A*B*C^2*a*b^5*d^6 + 2*A*B*C^2*a*b^5*c^6 - A^2*B*C*a*b^5*c^6 - 7*B^3*C*a^2*b^4*c^3*d^3 - 7*B*C^3*a^2*b^4*c^3*d^3 + 3*B^3*C*a^3*b^3*c^4*d^2 - 3*B^3*C*a^3*b^3*c^2*d^4 - 3*B^2*C^2*a^3*b^3*c*d^5 + 3*B*C^3*a^3*b^3*c^4*d^2 - 3*B*C^3*a^3*b^3*c^2*d^4 - B^3*C*a^4*b^2*c^3*d^3 - B^2*C^2*a*b^5*c^3*d^3 - B*C^3*a^4*b^2*c^3*d^3 - 24*A^2*C^2*a*b^5*c^3*d^3 - 24*A*C^3*a^3*b^3*c^3*d^3 + 12*A*C^3*a^2*b^4*c^4*d^2 + 9*A*C^3*a^4*b^2*c^2*d^4 - 8*A^3*C*a^3*b^3*c^3*d^3 + 6*A^3*C*a^2*b^4*c^4*d^2 - 6*A^3*C*a^2*b^4*c^2*d^4 + 3*A^3*C*a^4*b^2*c^2*d^4 - 9*A^2*B^2*a*b^5*c^3*d^3 + 7*A^3*B*a^2*b^4*c^3*d^3 + 7*A*B^3*a^2*b^4*c^3*d^3 - 3*A^3*B*a^3*b^3*c^2*d^4 - 3*A^2*B^2*a^3*b^3*c*d^5 - 3*A*B^3*a^3*b^3*c^2*d^4 + 12*A^2*C^2*b^6*c^4*d^2 + 3*A^2*C^2*b^6*c^2*d^4 + 6*A^2*B^2*b^6*c^4*d^2 + 3*A^2*B^2*b^6*c^2*d^4 - 5*A^2*C^2*a^2*b^4*d^6 + 3*A^2*C^2*a^4*b^2*d^6 + A*B*C^2*b^6*c^3*d^3 - 3*B^4*a^3*b^3*c*d^5 - B^4*a*b^5*c^3*d^3 + A^2*B^2*a^3*b^3*c^3*d^3 - 8*A^4*a*b^5*c^3*d^3 - 15*A^3*C*b^6*c^4*d^2 - 6*A^3*C*b^6*c^2*d^4 - 3*A*C^3*b^6*c^4*d^2 - 2*B^3*C*a^3*b^3*d^6 - 2*B*C^3*a^3*b^3*d^6 + 4*A^3*C*a^2*b^4*d^6 - 3*A*C^3*a^4*b^2*d^6 + 2*A*C^3*a^2*b^4*d^6 - A^3*C*a^4*b^2*d^6 - 2*A*C^3*a^2*b^4*c^6 + 3*B^4*a*b^5*c^5*d - 3*A^3*B*b^6*c^5*d - 3*A*B^3*b^6*c^5*d - B^3*C*a*b^5*c^6 - B*C^3* a*b^5*c^6 - 2*A^3*B*a*b^5*d^6 - 2*A*B^3*a*b^5*d^6 + 8*C^4*a^3*b^3*c^3*d^3 - 3*C^4*a^4*b^2*c^2*d^4 - 3*C^4*a^2*b^4*c^4*d^2 + 6*B^4*a^2*b^4*c^2*d^4 - 3*B^4*a^2*b^4*c^4*d^2 + 3*A^4*a^2*b^4*c^2*d^4 + B^2*C^2*a^4*b^2*d^6 + B^2*C^2*a^2*b^4*d^6 + B^2*C^2*a^2*b^4*c^6 + A^2*C^2*a^2*b^4*c^6 - 2*A^3*C*b^6*d^6 + A^3*B*b^6*c^3*d^3 + A*B^3*b^6*c^3*d^3 + A^3*B*a^3*b^3*d^6 + A*B^3*a^3*b^3*d^6 + 6*A^4*b^6*c^4*d^2 + 3*A^4*b^6*c^2*d^4 - A^4*a^2*b^4*d^6 - 2*A^2*C^2*b^6*c^6 + A*B^2*C*b^6*c^6 + B^4*a^3*b^3*c^3*d^3 + A^3*C*b^6*c^6 + A*C^3*b^6*c^6 + C^4*a^4*b^2*d^6 + C^4*a^2*b^4*c^6 + B^4*a^2*b^4*d^6 + A^2*C^2*b^6*d^6 + A^2*B^2*b^6*d^6 + A^4*b^6*d^6, f, k)*(root(480*a^9*b*c^7*d^11*f^4 + 480*a*b^9*c^11*d^7*f^4 + 360*a^9*b*c^9*d^9*f^4 + 360*a^9*b*c^5*d^13*f^4 + 360*a*b^9*c^13*d^5*f^4 + 360*a*b^9*c^9*d^9*f^4 + 144*a^9*b*c^11*d^7*f^4 + 144*a^9*b*c^3*d^15*f^4 + 144*a*b^9*c^15*d^3*f^4 + 144*a*b^9*c^7*d^11*f^4 + 48*a^7*b^3*c*d^17*f^4 + 48*a^3*b^7*c^17*d*f^4 + 24*a^9*b*c^13*d^5*f^4 + 24*a^5*b^5*c^17*d*f^4 + 24*a^5*b^5*c*d^17*f^4 + 24*a*b^9*c^5*d^13*f^4 + 24*a^9*b*c*d^17*f^4 + 24*a*b^9*c^17*d*f^4 + 3920*a^5*b^5*c^9*d^9*f^4 - 3360*a^6*b^4*c^8*d^10*f^4 - 3360*a^4*b^6*c^10*d^8*f^4 - 3024*a^6*b^4*c^10*d^8*f^4 + 3024*a^5*b^5*c^11*d^7*f^4 + 3024*a^5*b^5*c^7*d^11*f^4 - 3024*a^4*b^6*c^8*d^10*f^4 + 2320*a^7*b^3*c^9*d^9*f^4 + 2320*a^3*b^7*c^9*d^9*f^4 - 2240*a^6*b^4*c^6*d^12*f^4 - 2240*a^4*b^6*c^12*d^6*f^4 + 2160*a^7*b^3*c^7*d^11*f^4 + 2160*a^3*b^7*c^11*d^7*f^4 - 1624*a^6*b^4*c^12*d^6*f^4 - 1624*a^4*b^6*c^6*d^12*f^4 + 1488*a^7*b^3*c^11*d^7*f^4 + 1488*a^3*b^7*c^7*d^11*f^4 + 1344*a^5*b^5*c^13*d^5*f^4 + 1344*a^5*b^5*c^5*d^13*f^4 - 1320*a^8*b^2*c^8*d^10*f^4 - 1320*a^2*b^8*c^10*d^8*f^4 + 1200*a^7*b^3*c^5*d^13*f^4 + 1200*a^3*b^7*c^13*d^5*f^4 - 1060*a^8*b^2*c^6*d^12*f^4 - 1060*a^2*b^8*c^12*d^6*f^4 - 948*a^8*b^2*c^10*d^8*f^4 - 948*a^2*b^8*c^8*d^10*f^4 - 840*a^6*b^4*c^4*d^14*f^4 - 840*a^4*b^6*c^14*d^4*f^4 + 528*a^7*b^3*c^13*d^5*f^4 + 528*a^3*b^7*c^5*d^13*f^4 - 480*a^8*b^2*c^4*d^14*f^4 - 480*a^6*b^4*c^14*d^4*f^4 - 480*a^4*b^6*c^4*d^14*f^4 - 480*a^2*b^8*c^14*d^4*f^4 - 368*a^8*b^2*c^12*d^6*f^4 + 368*a^7*b^3*c^3*d^15*f^4 + 368*a^3*b^7*c^15*d^3*f^4 - 368*a^2*b^8*c^6*d^12*f^4 + 304*a^5*b^5*c^15*d^3*f^4 + 304*a^5*b^5*c^3*d^15*f^4 - 144*a^6*b^4*c^2*d^16*f^4 - 144*a^4*b^6*c^16*d^2*f^4 - 108*a^8*b^2*c^2*d^16*f^4 - 108*a^2*b^8*c^16*d^2*f^4 + 80*a^7*b^3*c^15*d^3*f^4 + 80*a^3*b^7*c^3*d^15*f^4 - 60*a^8*b^2*c^14*d^4*f^4 - 60*a^6*b^4*c^16*d^2*f^4 - 60*a^4*b^6*c^2*d^16*f^4 - 60*a^2*b^8*c^4*d^14*f^4 - 80*b^10*c^12*d^6*f^4 - 60*b^10*c^14*d^4*f^4 - 60*b^10*c^10*d^8*f^4 - 24*b^10*c^16*d^2*f^4 - 24*b^10*c^8*d^10*f^4 - 4*b^10*c^6*d^12*f^4 - 80*a^10*c^6*d^12*f^4 - 60*a^10*c^8*d^10*f^4 - 60*a^10*c^4*d^14*f^4 - 24*a^10*c^10*d^8*f^4 - 24*a^10*c^2*d^16*f^4 - 4*a^10*c^12*d^6*f^4 - 8*a^8*b^2*d^18*f^4 - 4*a^6*b^4*d^18*f^4 - 8*a^2*b^8*c^18*f^4 - 4*a^4*b^6*c^18*f^4 - 4*b^10*c^18*f^4 - 4*a^10*d^18*f^4 - 12*A*C*a^7*b*c*d^11*f^2 - 12*A*C*a*b^7*c^11*d*f^2 - 912*B*C*a^4*b^4*c^5*d^7*f^2 + 792*B*C*a^5*b^3*c^4*d^8*f^2 - 792*B*C*a^3*b^5*c^8*d^4*f^2 + 720*B*C*a^4*b^4*c^7*d^5*f^2 - 480*B*C*a^6*b^2*c^5*d^7*f^2 - 408*B*C*a^2*b^6*c^5*d^7*f^2 + 384*B*C*a^2*b^6*c^7*d^5*f^2 - 336*B*C*a^5*b^3*c^8*d^4*f^2 + 324*B*C*a^3*b^5*c^4*d^8*f^2 + 312*B*C*a^6*b^2*c^7*d^5*f^2 - 248*B*C*a^6*b^2*c^3*d^9*f^2 + 216*B*C*a^2*b^6*c^9*d^3*f^2 - 196*B*C*a^4*b^4*c^3*d^9*f^2 + 132*B*C*a^4*b^4*c^9*d^3*f^2 + 80*B*C*a^3*b^5*c^6*d^6*f^2 - 64*B

*C*a^5*b^3*c^6*d^6*f^2 - 36*B*C*a^3*b^5*c^2*d^10*f^2 - 28*B*C*a^2*b^6*c^3*d
^9*f^2 + 12*B*C*a^5*b^3*c^10*d^2*f^2 - 12*B*C*a^5*b^3*c^2*d^10*f^2 - 12*B*C
*a^3*b^5*c^10*d^2*f^2 - 4*B*C*a^6*b^2*c^9*d^3*f^2 - 1468*A*C*a^4*b^4*c^6*d^
6*f^2 + 996*A*C*a^3*b^5*c^7*d^5*f^2 + 900*A*C*a^5*b^3*c^5*d^7*f^2 - 676*A*C
*a^6*b^2*c^6*d^6*f^2 - 660*A*C*a^2*b^6*c^6*d^6*f^2 + 636*A*C*a^3*b^5*c^5*d^
7*f^2 + 540*A*C*a^5*b^3*c^7*d^5*f^2 - 236*A*C*a^5*b^3*c^3*d^9*f^2 - 204*A*C
*a^3*b^5*c^9*d^3*f^2 + 156*A*C*a^2*b^6*c^10*d^2*f^2 + 132*A*C*a^6*b^2*c^2*d
^10*f^2 - 72*A*C*a^6*b^2*c^4*d^8*f^2 - 72*A*C*a^5*b^3*c^9*d^3*f^2 + 66*A*C*
a^2*b^6*c^4*d^8*f^2 + 54*A*C*a^4*b^4*c^10*d^2*f^2 + 54*A*C*a^4*b^4*c^2*d^10
*f^2 - 48*A*C*a^4*b^4*c^4*d^8*f^2 - 48*A*C*a^2*b^6*c^8*d^4*f^2 + 42*A*C*a^6
*b^2*c^8*d^4*f^2 - 40*A*C*a^3*b^5*c^3*d^9*f^2 - 36*A*C*a^4*b^4*c^8*d^4*f^2
+ 24*A*C*a^2*b^6*c^2*d^10*f^2 + 960*A*B*a^4*b^4*c^5*d^7*f^2 - 864*A*B*a^5*b
^3*c^4*d^8*f^2 + 756*A*B*a^3*b^5*c^8*d^4*f^2 - 744*A*B*a^4*b^4*c^7*d^5*f^2
- 528*A*B*a^3*b^5*c^4*d^8*f^2 + 504*A*B*a^6*b^2*c^5*d^7*f^2 - 432*A*B*a^2*b
^6*c^7*d^5*f^2 + 432*A*B*a^2*b^6*c^5*d^7*f^2 + 348*A*B*a^5*b^3*c^8*d^4*f^2
- 312*A*B*a^6*b^2*c^7*d^5*f^2 - 284*A*B*a^2*b^6*c^9*d^3*f^2 + 280*A*B*a^6*b
^2*c^3*d^9*f^2 + 264*A*B*a^4*b^4*c^3*d^9*f^2 - 240*A*B*a^3*b^5*c^6*d^6*f^2
- 172*A*B*a^4*b^4*c^9*d^3*f^2 + 68*A*B*a^2*b^6*c^3*d^9*f^2 - 60*A*B*a^3*b^5
*c^2*d^10*f^2 + 24*A*B*a^5*b^3*c^6*d^6*f^2 - 24*A*B*a^5*b^3*c^2*d^10*f^2 +
12*A*B*a^3*b^5*c^10*d^2*f^2 + 360*B*C*a^7*b*c^4*d^8*f^2 - 336*B*C*a*b^7*c^8
*d^4*f^2 + 168*B*C*a*b^7*c^6*d^6*f^2 - 136*B*C*a^7*b*c^6*d^6*f^2 + 36*B*C*a
^6*b^2*c*d^11*f^2 - 36*B*C*a^2*b^6*c^11*d*f^2 - 24*B*C*a^7*b*c^2*d^10*f^2 +
24*B*C*a*b^7*c^10*d^2*f^2 - 12*B*C*a^4*b^4*c^11*d*f^2 + 12*B*C*a^4*b^4*c*d
^11*f^2 + 12*B*C*a*b^7*c^4*d^8*f^2 + 444*A*C*a*b^7*c^7*d^5*f^2 + 348*A*C*a^
7*b*c^5*d^7*f^2 - 164*A*C*a^7*b*c^3*d^9*f^2 - 132*A*C*a*b^7*c^9*d^3*f^2 + 8
4*A*C*a*b^7*c^5*d^7*f^2 + 32*A*C*a*b^7*c^3*d^9*f^2 - 12*A*C*a^7*b*c^7*d^5*f
^2 - 12*A*C*a^5*b^3*c*d^11*f^2 - 12*A*C*a^3*b^5*c^11*d*f^2 - 360*A*B*a^7*b*
c^4*d^8*f^2 + 288*A*B*a*b^7*c^8*d^4*f^2 - 288*A*B*a*b^7*c^6*d^6*f^2 - 144*A
*B*a*b^7*c^4*d^8*f^2 + 136*A*B*a^7*b*c^6*d^6*f^2 - 60*A*B*a*b^7*c^2*d^10*f^
2 - 36*A*B*a*b^7*c^10*d^2*f^2 + 24*A*B*a^7*b*c^2*d^10*f^2 - 24*A*B*a^6*b^2*
c*d^11*f^2 + 12*A*B*a^4*b^4*c*d^11*f^2 + 12*A*B*a^2*b^6*c^11*d*f^2 + 12*A*B
*a^2*b^6*c*d^11*f^2 + 80*B*C*b^8*c^9*d^3*f^2 - 24*B*C*b^8*c^7*d^5*f^2 - 90*
A*C*b^8*c^8*d^4*f^2 - 80*B*C*a^8*c^3*d^9*f^2 + 54*A*C*b^8*c^10*d^2*f^2 - 30
*A*C*b^8*c^6*d^6*f^2 + 24*B*C*a^8*c^5*d^7*f^2 - 12*A*C*b^8*c^4*d^8*f^2 - 11
2*A*B*b^8*c^9*d^3*f^2 - 66*A*C*a^8*c^4*d^8*f^2 + 54*A*C*a^8*c^2*d^10*f^2 -
8*B*C*a^5*b^3*d^12*f^2 - 8*B*C*a^3*b^5*d^12*f^2 + 4*A*B*b^8*c^3*d^9*f^2 + 2
*A*C*a^8*c^6*d^6*f^2 + 80*A*B*a^8*c^3*d^9*f^2 - 24*A*B*a^8*c^5*d^7*f^2 + 8*
A*C*a^2*b^6*d^12*f^2 - 4*B*C*a^3*b^5*c^12*f^2 + 4*A*C*a^4*b^4*d^12*f^2 - 2*
A*C*a^6*b^2*d^12*f^2 + 6*A*C*a^2*b^6*c^12*f^2 + 4*A*B*a^5*b^3*d^12*f^2 - 4*
A*B*a^3*b^5*d^12*f^2 + 726*C^2*a^4*b^4*c^6*d^6*f^2 - 402*C^2*a^5*b^3*c^5*d^
7*f^2 - 402*C^2*a^3*b^5*c^7*d^5*f^2 + 322*C^2*a^6*b^2*c^6*d^6*f^2 + 322*C^2
*a^2*b^6*c^6*d^6*f^2 - 222*C^2*a^5*b^3*c^7*d^5*f^2 - 222*C^2*a^3*b^5*c^5*d^
7*f^2 + 134*C^2*a^5*b^3*c^3*d^9*f^2 + 134*C^2*a^3*b^5*c^9*d^3*f^2 - 66*C^2*
a^6*b^2*c^2*d^10*f^2 - 66*C^2*a^2*b^6*c^10*d^2*f^2 + 52*C^2*a^5*b^3*c^9*d^3
*f^2 + 52*C^2*a^3*b^5*c^3*d^9*f^2 - 27*C^2*a^6*b^2*c^8*d^4*f^2 - 27*C^2*a^2

*b^6*c^4*d^8*f^2 + 24*C^2*a^6*b^2*c^4*d^8*f^2 + 24*C^2*a^4*b^4*c^8*d^4*f^2
+ 24*C^2*a^4*b^4*c^4*d^8*f^2 + 24*C^2*a^2*b^6*c^8*d^4*f^2 - 15*C^2*a^4*b^4*
c^10*d^2*f^2 - 15*C^2*a^4*b^4*c^2*d^10*f^2 - 570*B^2*a^4*b^4*c^6*d^6*f^2 +
366*B^2*a^3*b^5*c^7*d^5*f^2 + 318*B^2*a^5*b^3*c^5*d^7*f^2 - 262*B^2*a^6*b^2
*c^6*d^6*f^2 - 222*B^2*a^2*b^6*c^6*d^6*f^2 - 210*B^2*a^5*b^3*c^3*d^9*f^2 +
186*B^2*a^5*b^3*c^7*d^5*f^2 + 162*B^2*a^3*b^5*c^5*d^7*f^2 - 142*B^2*a^3*b^5
*c^9*d^3*f^2 + 132*B^2*a^4*b^4*c^4*d^8*f^2 + 117*B^2*a^2*b^6*c^4*d^8*f^2 +
102*B^2*a^6*b^2*c^2*d^10*f^2 - 96*B^2*a^3*b^5*c^3*d^9*f^2 + 90*B^2*a^2*b^6*
c^10*d^2*f^2 + 81*B^2*a^4*b^4*c^2*d^10*f^2 - 56*B^2*a^5*b^3*c^9*d^3*f^2 + 4
8*B^2*a^6*b^2*c^4*d^8*f^2 + 48*B^2*a^4*b^4*c^8*d^4*f^2 + 45*B^2*a^6*b^2*c^8
*d^4*f^2 + 36*B^2*a^2*b^6*c^8*d^4*f^2 + 36*B^2*a^2*b^6*c^2*d^10*f^2 + 33*B^
2*a^4*b^4*c^10*d^2*f^2 + 822*A^2*a^4*b^4*c^6*d^6*f^2 - 594*A^2*a^3*b^5*c^7*
d^5*f^2 - 498*A^2*a^5*b^3*c^5*d^7*f^2 + 498*A^2*a^2*b^6*c^6*d^6*f^2 - 414*A
^2*a^3*b^5*c^5*d^7*f^2 + 354*A^2*a^6*b^2*c^6*d^6*f^2 - 318*A^2*a^5*b^3*c^7*
d^5*f^2 + 144*A^2*a^2*b^6*c^8*d^4*f^2 + 102*A^2*a^5*b^3*c^3*d^9*f^2 + 84*A^
2*a^4*b^4*c^4*d^8*f^2 + 81*A^2*a^2*b^6*c^4*d^8*f^2 + 72*A^2*a^4*b^4*c^8*d^4
*f^2 + 70*A^2*a^3*b^5*c^9*d^3*f^2 - 66*A^2*a^6*b^2*c^2*d^10*f^2 + 48*A^2*a^
6*b^2*c^4*d^8*f^2 - 42*A^2*a^2*b^6*c^10*d^2*f^2 + 24*A^2*a^2*b^6*c^2*d^10*f
^2 + 20*A^2*a^5*b^3*c^9*d^3*f^2 - 15*A^2*a^6*b^2*c^8*d^4*f^2 - 15*A^2*a^4*b
^4*c^10*d^2*f^2 - 15*A^2*a^4*b^4*c^2*d^10*f^2 - 12*A^2*a^3*b^5*c^3*d^9*f^2
- 24*B*C*b^8*c^11*d*f^2 + 24*B*C*a^8*c*d^11*f^2 + 12*A*B*b^8*c^11*d*f^2 - 8
*B*C*a^7*b*d^12*f^2 - 24*A*B*a^8*c*d^11*f^2 + 4*B*C*a*b^7*c^12*f^2 + 8*A*B*
a^7*b*d^12*f^2 - 8*A*B*a*b^7*d^12*f^2 - 8*A*B*a*b^7*c^12*f^2 - 174*C^2*a^7*
b*c^5*d^7*f^2 - 174*C^2*a*b^7*c^7*d^5*f^2 + 82*C^2*a^7*b*c^3*d^9*f^2 + 82*C
^2*a*b^7*c^9*d^3*f^2 + 6*C^2*a^7*b*c^7*d^5*f^2 + 6*C^2*a^5*b^3*c*d^11*f^2 +
6*C^2*a^3*b^5*c^11*d*f^2 + 6*C^2*a*b^7*c^5*d^7*f^2 + 162*B^2*a*b^7*c^7*d^5
*f^2 + 138*B^2*a^7*b*c^5*d^7*f^2 - 118*B^2*a^7*b*c^3*d^9*f^2 - 86*B^2*a*b^7
*c^9*d^3*f^2 - 30*B^2*a^5*b^3*c*d^11*f^2 - 18*B^2*a^7*b*c^7*d^5*f^2 - 18*B^
2*a*b^7*c^5*d^7*f^2 - 12*B^2*a^3*b^5*c*d^11*f^2 - 6*B^2*a^3*b^5*c^11*d*f^2
- 4*B^2*a*b^7*c^3*d^9*f^2 - 270*A^2*a*b^7*c^7*d^5*f^2 - 174*A^2*a^7*b*c^5*d
^7*f^2 - 90*A^2*a*b^7*c^5*d^7*f^2 + 82*A^2*a^7*b*c^3*d^9*f^2 + 50*A^2*a*b^7
*c^9*d^3*f^2 - 32*A^2*a*b^7*c^3*d^9*f^2 + 6*A^2*a^7*b*c^7*d^5*f^2 + 6*A^2*a
^5*b^3*c*d^11*f^2 + 6*A^2*a^3*b^5*c^11*d*f^2 + 6*C^2*a^7*b*c*d^11*f^2 + 6*C
^2*a*b^7*c^11*d*f^2 - 18*B^2*a^7*b*c*d^11*f^2 - 6*B^2*a*b^7*c^11*d*f^2 + 6*
A^2*a^7*b*c*d^11*f^2 + 6*A^2*a*b^7*c^11*d*f^2 - 6*A*C*a^8*d^12*f^2 - 2*A*C*
b^8*c^12*f^2 + 33*C^2*b^8*c^8*d^4*f^2 - 27*C^2*b^8*c^10*d^2*f^2 - C^2*b^8*c
^6*d^6*f^2 + 33*C^2*a^8*c^4*d^8*f^2 + 33*B^2*b^8*c^10*d^2*f^2 - 27*C^2*a^8*
c^2*d^10*f^2 - 27*B^2*b^8*c^8*d^4*f^2 + 3*B^2*b^8*c^6*d^6*f^2 - C^2*a^8*c^6
*d^6*f^2 + 117*A^2*b^8*c^8*d^4*f^2 + 111*A^2*b^8*c^6*d^6*f^2 + 72*A^2*b^8*c
^4*d^8*f^2 + 33*B^2*a^8*c^2*d^10*f^2 - 27*B^2*a^8*c^4*d^8*f^2 + 24*A^2*b^8*
c^2*d^10*f^2 + 4*C^2*a^4*b^4*d^12*f^2 + 3*C^2*a^6*b^2*d^12*f^2 + 3*B^2*a^8*
c^6*d^6*f^2 - 3*A^2*b^8*c^10*d^2*f^2 + 33*A^2*a^8*c^4*d^8*f^2 - 27*A^2*a^8*
c^2*d^10*f^2 + 4*C^2*a^4*b^4*c^12*f^2 + 4*B^2*a^4*b^4*d^12*f^2 + 4*B^2*a^2*
b^6*d^12*f^2 + 3*C^2*a^2*b^6*c^12*f^2 + 3*B^2*a^6*b^2*d^12*f^2 - A^2*a^8*c^
6*d^6*f^2 - 4*A^2*a^4*b^4*d^12*f^2 + 3*B^2*a^2*b^6*c^12*f^2 - A^2*a^6*b^2*d

^12*f^2 - A^2*a^2*b^6*c^12*f^2 + 3*C^2*b^8*c^12*f^2 + 3*C^2*a^8*d^12*f^2 + 4*A^2*b^8*d^12*f^2 - B^2*b^8*c^12*f^2 - B^2*a^8*d^12*f^2 + 3*A^2*b^8*c^12*f ^2 + 3*A^2*a^8*d^12*f^2 - 24*A*B*C*a*b^6*c*d^8*f + 342*A*B*C*a^2*b^5*c^4*d^ 5*f - 186*A*B*C*a^3*b^4*c^5*d^4*f - 66*A*B*C*a^4*b^3*c^2*d^7*f + 48*A*B*C*a ^2*b^5*c^2*d^7*f + 42*A*B*C*a^2*b^5*c^6*d^3*f + 26*A*B*C*a^5*b^2*c^3*d^6*f + 24*A*B*C*a^4*b^3*c^6*d^3*f - 18*A*B*C*a^4*b^3*c^4*d^5*f - 18*A*B*C*a^3*b^ 4*c^7*d^2*f - 8*A*B*C*a^3*b^4*c^3*d^6*f + 6*A*B*C*a^5*b^2*c^5*d^4*f - 128*A *B*C*a*b^6*c^3*d^6*f + 126*A*B*C*a*b^6*c^7*d^2*f + 72*A*B*C*a^3*b^4*c*d^8*f - 36*A*B*C*a^5*b^2*c*d^8*f - 36*A*B*C*a^2*b^5*c^8*d*f + 30*A*B*C*a^6*b*c^2 *d^7*f - 12*A*B*C*a^6*b*c^4*d^5*f - 12*A*B*C*a*b^6*c^5*d^4*f - 21*B^2*C*a*b ^6*c^8*d*f - 3*B^2*C*a^6*b*c*d^8*f + 21*A^2*C*a*b^6*c^8*d*f - 21*A*C^2*a*b^ 6*c^8*d*f - 9*A^2*C*a^6*b*c*d^8*f + 9*A*C^2*a^6*b*c*d^8*f + 36*A^2*B*a*b^6* c*d^8*f + 21*A*B^2*a*b^6*c^8*d*f + 3*A*B^2*a^6*b*c*d^8*f - 78*A*B*C*b^7*c^6 *d^3*f + 24*A*B*C*b^7*c^4*d^5*f + 2*A*B*C*a^7*c^3*d^6*f + 16*A*B*C*a^4*b^3* d^9*f - 16*A*B*C*a^2*b^5*d^9*f - 237*B^2*C*a^3*b^4*c^4*d^5*f + 165*B*C^2*a^ 3*b^4*c^5*d^4*f + 92*B^2*C*a^2*b^5*c^3*d^6*f - 81*B^2*C*a^2*b^5*c^7*d^2*f + 77*B^2*C*a^4*b^3*c^3*d^6*f - 75*B*C^2*a^2*b^5*c^4*d^5*f + 69*B^2*C*a^4*b^3 *c^5*d^4*f + 69*B*C^2*a^4*b^3*c^4*d^5*f - 68*B*C^2*a^3*b^4*c^3*d^6*f - 63*B ^2*C*a^5*b^2*c^4*d^5*f - 61*B*C^2*a^2*b^5*c^6*d^3*f + 57*B*C^2*a^4*b^3*c^2* d^7*f - 53*B*C^2*a^5*b^2*c^3*d^6*f - 44*B*C^2*a^4*b^3*c^6*d^3*f - 36*B^2*C* a^3*b^4*c^2*d^7*f + 35*B^2*C*a^3*b^4*c^6*d^3*f + 33*B^2*C*a^5*b^2*c^2*d^7*f - 33*B^2*C*a^2*b^5*c^5*d^4*f + 33*B*C^2*a^3*b^4*c^7*d^2*f - 12*B^2*C*a^4*b ^3*c^7*d^2*f + 9*B*C^2*a^5*b^2*c^5*d^4*f + 4*B^2*C*a^5*b^2*c^6*d^3*f + 225* A^2*C*a^2*b^5*c^5*d^4*f - 105*A*C^2*a^2*b^5*c^5*d^4*f - 99*A^2*C*a^3*b^4*c^ 4*d^5*f - 81*A^2*C*a^5*b^2*c^4*d^5*f + 67*A^2*C*a^4*b^3*c^3*d^6*f - 59*A*C^ 2*a^4*b^3*c^3*d^6*f + 57*A*C^2*a^5*b^2*c^2*d^7*f - 57*A*C^2*a^2*b^5*c^7*d^2 *f + 51*A^2*C*a^4*b^3*c^5*d^4*f + 48*A^2*C*a^3*b^4*c^2*d^7*f + 45*A*C^2*a^5 *b^2*c^4*d^5*f - 35*A^2*C*a^3*b^4*c^6*d^3*f - 33*A^2*C*a^5*b^2*c^2*d^7*f + 33*A^2*C*a^2*b^5*c^7*d^2*f + 33*A*C^2*a^4*b^3*c^5*d^4*f + 27*A*C^2*a^3*b^4* c^6*d^3*f - 24*A*C^2*a^3*b^4*c^2*d^7*f + 24*A*C^2*a^2*b^5*c^3*d^6*f - 21*A* C^2*a^3*b^4*c^4*d^5*f - 16*A^2*C*a^2*b^5*c^3*d^6*f - 243*A^2*B*a^2*b^5*c^4* d^5*f - 156*A*B^2*a^2*b^5*c^3*d^6*f + 141*A*B^2*a^3*b^4*c^4*d^5*f + 108*A^2 *B*a^3*b^4*c^3*d^6*f - 105*A*B^2*a^4*b^3*c^3*d^6*f + 84*A*B^2*a^3*b^4*c^2*d ^7*f + 81*A*B^2*a^2*b^5*c^5*d^4*f - 51*A^2*B*a^4*b^3*c^4*d^5*f + 51*A^2*B*a ^2*b^5*c^6*d^3*f - 48*A^2*B*a^2*b^5*c^2*d^7*f + 45*A^2*B*a^3*b^4*c^5*d^4*f + 39*A*B^2*a^5*b^2*c^4*d^5*f - 35*A*B^2*a^3*b^4*c^6*d^3*f + 33*A*B^2*a^2*b^ 5*c^7*d^2*f + 27*A^2*B*a^5*b^2*c^3*d^6*f - 21*A*B^2*a^4*b^3*c^5*d^4*f + 20* A^2*B*a^4*b^3*c^6*d^3*f - 15*A^2*B*a^5*b^2*c^5*d^4*f - 15*A^2*B*a^3*b^4*c^7 *d^2*f + 9*A^2*B*a^4*b^3*c^2*d^7*f + 3*A*B^2*a^5*b^2*c^2*d^7*f + 18*A*B*C*b ^7*c^8*d*f - 6*A*B*C*a^7*c*d^8*f + 2*A*B*C*a^6*b*d^9*f - 6*A*B*C*a*b^6*c^9* f + 63*B^2*C*a*b^6*c^6*d^3*f - 48*B^2*C*a^4*b^3*c*d^8*f + 42*B*C^2*a^2*b^5* c^8*d*f + 42*B*C^2*a*b^6*c^5*d^4*f - 39*B*C^2*a*b^6*c^7*d^2*f + 30*B*C^2*a^ 5*b^2*c*d^8*f - 24*B^2*C*a*b^6*c^4*d^5*f - 24*B*C^2*a^3*b^4*c*d^8*f + 17*B^ 2*C*a^6*b*c^3*d^6*f - 15*B*C^2*a^6*b*c^2*d^7*f + 12*B^2*C*a^3*b^4*c^8*d*f + 12*B^2*C*a^2*b^5*c*d^8*f + 6*B*C^2*a^6*b*c^4*d^5*f - 192*A^2*C*a*b^6*c^4*d

^5*f - 99*A^2*C*a*b^6*c^6*d^3*f + 84*A*C^2*a*b^6*c^4*d^5*f + 59*A*C^2*a*b^6
*c^6*d^3*f + 51*A^2*C*a^6*b*c^3*d^6*f - 51*A*C^2*a^6*b*c^3*d^6*f - 36*A^2*C
*a^2*b^5*c*d^8*f - 24*A*C^2*a^4*b^3*c*d^8*f + 24*A*C^2*a^2*b^5*c*d^8*f + 12
*A^2*C*a^4*b^3*c*d^8*f + 12*A*C^2*a^3*b^4*c^8*d*f + 160*A^2*B*a*b^6*c^3*d^6
*f - 99*A*B^2*a*b^6*c^6*d^3*f - 87*A^2*B*a*b^6*c^7*d^2*f - 72*A*B^2*a*b^6*c
^4*d^5*f - 48*A*B^2*a^2*b^5*c*d^8*f - 36*A^2*B*a^3*b^4*c*d^8*f + 24*A*B^2*a
^4*b^3*c*d^8*f - 17*A*B^2*a^6*b*c^3*d^6*f - 15*A^2*B*a^6*b*c^2*d^7*f + 12*A
*B^2*a*b^6*c^2*d^7*f + 6*A^2*B*a^6*b*c^4*d^5*f + 6*A^2*B*a^5*b^2*c*d^8*f +
6*A^2*B*a^2*b^5*c^8*d*f - 6*A^2*B*a*b^6*c^5*d^4*f + 3*B^2*C*b^7*c^7*d^2*f -
B*C^2*b^7*c^6*d^3*f + 96*A^2*C*b^7*c^5*d^4*f - 39*A^2*C*b^7*c^7*d^2*f - 36
*A*C^2*b^7*c^5*d^4*f + 32*A^2*C*b^7*c^3*d^6*f + 15*A*C^2*b^7*c^7*d^2*f - 3*
B^2*C*a^7*c^2*d^7*f - B*C^2*a^7*c^3*d^6*f + 111*A^2*B*b^7*c^6*d^3*f - 39*A*
B^2*b^7*c^7*d^2*f + 24*A*B^2*b^7*c^5*d^4*f + 12*B^2*C*a^3*b^4*d^9*f - 12*B*
C^2*a^4*b^3*d^9*f - 9*A^2*C*a^7*c^2*d^7*f + 9*A*C^2*a^7*c^2*d^7*f - 4*A*B^2
*b^7*c^3*d^6*f - 12*A^2*C*a^3*b^4*d^9*f - 8*A*C^2*a^5*b^2*d^9*f + 8*A*C^2*a
^3*b^4*d^9*f + 4*B^2*C*a^2*b^5*c^9*f + 4*A^2*C*a^5*b^2*d^9*f - 4*B*C^2*a^3*
b^4*c^9*f + 3*A*B^2*a^7*c^2*d^7*f - A^2*B*a^7*c^3*d^6*f + 12*A^2*B*a^2*b^5*
d^9*f - 8*A*B^2*a^3*b^4*d^9*f - 4*A^2*B*a^4*b^3*d^9*f + 4*A*C^2*a^2*b^5*c^9
*f - 3*C^3*a^6*b*c*d^8*f + 3*C^3*a*b^6*c^8*d*f + 3*A^3*a^6*b*c*d^8*f - 3*A^
3*a*b^6*c^8*d*f + 3*B*C^2*b^7*c^8*d*f + 12*A^2*C*b^7*c*d^8*f + 3*B*C^2*a^7*
c*d^8*f - 9*A^2*B*b^7*c^8*d*f - B*C^2*a^6*b*d^9*f + 4*A^2*C*a*b^6*d^9*f + 3
*A^2*B*a^7*c*d^8*f + 3*B*C^2*a*b^6*c^9*f + 8*A*B^2*a*b^6*d^9*f - A^2*B*a^6*
b*d^9*f - A^2*B*a*b^6*c^9*f - 39*C^3*a^4*b^3*c^5*d^4*f + 39*C^3*a^3*b^4*c^4
*d^5*f - 27*C^3*a^5*b^2*c^2*d^7*f + 27*C^3*a^2*b^5*c^7*d^2*f + 17*C^3*a^4*b
^3*c^3*d^6*f - 17*C^3*a^3*b^4*c^6*d^3*f - 3*C^3*a^5*b^2*c^4*d^5*f + 3*C^3*a
^2*b^5*c^5*d^4*f - 63*B^3*a^3*b^4*c^5*d^4*f + 57*B^3*a^2*b^5*c^4*d^5*f - 51
*B^3*a^4*b^3*c^2*d^7*f + 48*B^3*a^3*b^4*c^3*d^6*f + 31*B^3*a^2*b^5*c^6*d^3*
f + 27*B^3*a^5*b^2*c^3*d^6*f + 16*B^3*a^4*b^3*c^6*d^3*f - 15*B^3*a^5*b^2*c^
5*d^4*f - 12*B^3*a^2*b^5*c^2*d^7*f + 9*B^3*a^4*b^3*c^4*d^5*f - 3*B^3*a^3*b^
4*c^7*d^2*f - 123*A^3*a^2*b^5*c^5*d^4*f + 81*A^3*a^3*b^4*c^4*d^5*f - 45*A^3
*a^4*b^3*c^5*d^4*f + 39*A^3*a^5*b^2*c^4*d^5*f - 25*A^3*a^4*b^3*c^3*d^6*f +
25*A^3*a^3*b^4*c^6*d^3*f - 24*A^3*a^3*b^4*c^2*d^7*f - 8*A^3*a^2*b^5*c^3*d^6
*f + 3*A^3*a^5*b^2*c^2*d^7*f - 3*A^3*a^2*b^5*c^7*d^2*f + 17*C^3*a^6*b*c^3*d
^6*f - 17*C^3*a*b^6*c^6*d^3*f + 12*C^3*a^4*b^3*c*d^8*f - 12*C^3*a^3*b^4*c^8
*d*f + 24*B^3*a^3*b^4*c*d^8*f + 21*B^3*a*b^6*c^7*d^2*f - 18*B^3*a*b^6*c^5*d
^4*f - 15*B^3*a^6*b*c^2*d^7*f + 6*B^3*a^6*b*c^4*d^5*f + 6*B^3*a^5*b^2*c*d^8
*f - 6*B^3*a^2*b^5*c^8*d*f + 4*B^3*a*b^6*c^3*d^6*f + 108*A^3*a*b^6*c^4*d^5*
f + 57*A^3*a*b^6*c^6*d^3*f - 17*A^3*a^6*b*c^3*d^6*f + 12*A^3*a^2*b^5*c*d^8*
f + 3*C^3*b^7*c^7*d^2*f - 3*C^3*a^7*c^2*d^7*f - B^3*b^7*c^6*d^3*f - 60*A^3*
b^7*c^5*d^4*f - 32*A^3*b^7*c^3*d^6*f + 21*A^3*b^7*c^7*d^2*f + 4*C^3*a^5*b^2
*d^9*f - B^3*a^7*c^3*d^6*f - 4*C^3*a^2*b^5*c^9*f - 4*B^3*a^2*b^5*d^9*f + 3*
A^3*a^7*c^2*d^7*f + 4*A^3*a^3*b^4*d^9*f + 3*B^3*b^7*c^8*d*f - 12*A^3*b^7*c*
d^8*f + 3*B^3*a^7*c*d^8*f - B^3*a^6*b*d^9*f - 4*A^3*a*b^6*d^9*f - B^3*a*b^6
*c^9*f - B^2*C*b^7*c^9*f - 4*A^2*B*b^7*d^9*f + 3*A^2*C*a^7*d^9*f - 3*A*C^2*
a^7*d^9*f - A*C^2*b^7*c^9*f - A*B^2*a^7*d^9*f - C^3*b^7*c^9*f - A^3*a^7*d^9

*f + B^2*C*a^7*d^9*f + A^2*C*b^7*c^9*f + A*B^2*b^7*c^9*f + C^3*a^7*d^9*f +
A^3*b^7*c^9*f - 6*A*B^2*C*a*b^5*c^5*d - 21*A^2*B*C*a^2*b^4*c^3*d^3 + 21*A*B
*C^2*a^2*b^4*c^3*d^3 + 12*A*B^2*C*a^2*b^4*c^4*d^2 - 12*A*B^2*C*a^2*b^4*c^2*
d^4 - 10*A*B^2*C*a^3*b^3*c^3*d^3 - 6*A*B*C^2*a^3*b^3*c^4*d^2 + 3*A^2*B*C*a^
3*b^3*c^4*d^2 + 3*A^2*B*C*a^3*b^3*c^2*d^4 + 3*A*B^2*C*a^4*b^2*c^2*d^4 + 3*A
*B*C^2*a^3*b^3*c^2*d^4 + 2*A*B*C^2*a^4*b^2*c^3*d^3 - A^2*B*C*a^4*b^2*c^3*d^
3 + 18*A^2*B*C*a*b^5*c^2*d^4 + 10*A*B^2*C*a*b^5*c^3*d^3 + 9*A^2*B*C*a*b^5*c
^4*d^2 - 9*A*B*C^2*a*b^5*c^4*d^2 - 9*A*B*C^2*a*b^5*c^2*d^4 - 6*A^2*B*C*a^2*
b^4*c*d^5 + 6*A*B^2*C*a^3*b^3*c*d^5 - 6*A*B*C^2*a^4*b^2*c*d^5 + 6*A*B*C^2*a
^2*b^4*c^5*d + 3*A^2*B*C*a^4*b^2*c*d^5 - 3*A^2*B*C*a^2*b^4*c^5*d + 3*A*B*C^
2*a^2*b^4*c*d^5 + 3*B^3*C*a^4*b^2*c*d^5 - 3*B^3*C*a^2*b^4*c^5*d + 3*B^3*C*a
*b^5*c^4*d^2 + 3*B^2*C^2*a*b^5*c^5*d + 3*B*C^3*a^4*b^2*c*d^5 - 3*B*C^3*a^2*
b^4*c^5*d + 3*B*C^3*a*b^5*c^4*d^2 + 24*A^3*C*a*b^5*c^3*d^3 + 8*A*C^3*a*b^5*
c^3*d^3 - 9*A^3*B*a*b^5*c^2*d^4 - 9*A*B^3*a*b^5*c^2*d^4 + 3*A^3*B*a^2*b^4*c
*d^5 - 3*A^3*B*a*b^5*c^4*d^2 + 3*A^2*B^2*a*b^5*c^5*d + 3*A*B^3*a^2*b^4*c*d^
5 - 3*A*B^3*a*b^5*c^4*d^2 - 3*A*B^2*C*b^6*c^4*d^2 - 2*A^2*B*C*b^6*c^3*d^3 +
 5*A*B*C^2*a^3*b^3*d^6 - 4*A^2*B*C*a^3*b^3*d^6 - A*B^2*C*a^4*b^2*d^6 + 9*B^
2*C^2*a^3*b^3*c^3*d^3 - 6*B^2*C^2*a^2*b^4*c^4*d^2 + 6*B^2*C^2*a^2*b^4*c^2*d
^4 - 3*B^2*C^2*a^4*b^2*c^2*d^4 + 24*A^2*C^2*a^3*b^3*c^3*d^3 - 15*A^2*C^2*a^
2*b^4*c^4*d^2 - 9*A^2*C^2*a^4*b^2*c^2*d^4 + 3*A^2*C^2*a^2*b^4*c^2*d^4 + 9*A
^2*B^2*a^2*b^4*c^2*d^4 - 3*A^2*B^2*a^2*b^4*c^4*d^2 + 6*A^2*B*C*b^6*c^5*d -
3*A*B*C^2*b^6*c^5*d + 4*A^2*B*C*a*b^5*d^6 - 2*A*B*C^2*a*b^5*d^6 + 2*A*B*C^2
*a*b^5*c^6 - A^2*B*C*a*b^5*c^6 - 7*B^3*C*a^2*b^4*c^3*d^3 - 7*B*C^3*a^2*b^4*
c^3*d^3 + 3*B^3*C*a^3*b^3*c^4*d^2 - 3*B^3*C*a^3*b^3*c^2*d^4 - 3*B^2*C^2*a^3
*b^3*c*d^5 + 3*B*C^3*a^3*b^3*c^4*d^2 - 3*B*C^3*a^3*b^3*c^2*d^4 - B^3*C*a^4*
b^2*c^3*d^3 - B^2*C^2*a*b^5*c^3*d^3 - B*C^3*a^4*b^2*c^3*d^3 - 24*A^2*C^2*a*
b^5*c^3*d^3 - 24*A*C^3*a^3*b^3*c^3*d^3 + 12*A*C^3*a^2*b^4*c^4*d^2 + 9*A*C^3
*a^4*b^2*c^2*d^4 - 8*A^3*C*a^3*b^3*c^3*d^3 + 6*A^3*C*a^2*b^4*c^4*d^2 - 6*A^
3*C*a^2*b^4*c^2*d^4 + 3*A^3*C*a^4*b^2*c^2*d^4 - 9*A^2*B^2*a*b^5*c^3*d^3 + 7
*A^3*B*a^2*b^4*c^3*d^3 + 7*A*B^3*a^2*b^4*c^3*d^3 - 3*A^3*B*a^3*b^3*c^2*d^4
- 3*A^2*B^2*a^3*b^3*c*d^5 - 3*A*B^3*a^3*b^3*c^2*d^4 + 12*A^2*C^2*b^6*c^4*d^
2 + 3*A^2*C^2*b^6*c^2*d^4 + 6*A^2*B^2*b^6*c^4*d^2 + 3*A^2*B^2*b^6*c^2*d^4 -
 5*A^2*C^2*a^2*b^4*d^6 + 3*A^2*C^2*a^4*b^2*d^6 + A*B*C^2*b^6*c^3*d^3 - 3*B^
4*a^3*b^3*c*d^5 - B^4*a*b^5*c^3*d^3 + A^2*B^2*a^3*b^3*c^3*d^3 - 8*A^4*a*b^5
*c^3*d^3 - 15*A^3*C*b^6*c^4*d^2 - 6*A^3*C*b^6*c^2*d^4 - 3*A*C^3*b^6*c^4*d^2
 - 2*B^3*C*a^3*b^3*d^6 - 2*B*C^3*a^3*b^3*d^6 + 4*A^3*C*a^2*b^4*d^6 - 3*A*C^
3*a^4*b^2*d^6 + 2*A*C^3*a^2*b^4*d^6 - A^3*C*a^4*b^2*d^6 - 2*A*C^3*a^2*b^4*c
^6 + 3*B^4*a*b^5*c^5*d - 3*A^3*B*b^6*c^5*d - 3*A*B^3*b^6*c^5*d - B^3*C*a*b^
5*c^6 - B*C^3*a*b^5*c^6 - 2*A^3*B*a*b^5*d^6 - 2*A*B^3*a*b^5*d^6 + 8*C^4*a^3
*b^3*c^3*d^3 - 3*C^4*a^4*b^2*c^2*d^4 - 3*C^4*a^2*b^4*c^4*d^2 + 6*B^4*a^2*b^
4*c^2*d^4 - 3*B^4*a^2*b^4*c^4*d^2 + 3*A^4*a^2*b^4*c^2*d^4 + B^2*C^2*a^4*b^2
*d^6 + B^2*C^2*a^2*b^4*d^6 + B^2*C^2*a^2*b^4*c^6 + A^2*C^2*a^2*b^4*c^6 - 2*
A^3*C*b^6*d^6 + A^3*B*b^6*c^3*d^3 + A*B^3*b^6*c^3*d^3 + A^3*B*a^3*b^3*d^6 +
 A*B^3*a^3*b^3*d^6 + 6*A^4*b^6*c^4*d^2 + 3*A^4*b^6*c^2*d^4 - A^4*a^2*b^4*d^
6 - 2*A^2*C^2*b^6*c^6 + A*B^2*C*b^6*c^6 + B^4*a^3*b^3*c^3*d^3 + A^3*C*b^6*c

^6 + A*C^3*b^6*c^6 + C^4*a^4*b^2*d^6 + C^4*a^2*b^4*c^6 + B^4*a^2*b^4*d^6 +
A^2*C^2*b^6*d^6 + A^2*B^2*b^6*d^6 + A^4*b^6*d^6, f, k)*((4*a^5*b^4*d^17 - 4
*a^7*b^2*d^17 + 4*b^9*c^5*d^12 + 12*b^9*c^7*d^10 + 8*b^9*c^9*d^8 - 8*b^9*c^
11*d^6 - 12*b^9*c^13*d^4 - 4*b^9*c^15*d^2 - 12*a*b^8*c^4*d^13 - 20*a*b^8*c^
6*d^11 + 48*a*b^8*c^8*d^9 + 152*a*b^8*c^10*d^7 + 148*a*b^8*c^12*d^5 + 60*a*
b^8*c^14*d^3 - 12*a^4*b^5*c*d^16 + 28*a^6*b^3*c*d^16 + 32*a^8*b*c^3*d^14 +
48*a^8*b*c^5*d^12 + 32*a^8*b*c^7*d^10 + 8*a^8*b*c^9*d^8 + 8*a^2*b^7*c^3*d^1
4 - 28*a^2*b^7*c^5*d^12 - 228*a^2*b^7*c^7*d^10 - 472*a^2*b^7*c^9*d^8 - 448*
a^2*b^7*c^11*d^6 - 204*a^2*b^7*c^13*d^4 - 36*a^2*b^7*c^15*d^2 + 8*a^3*b^6*c
^2*d^15 + 68*a^3*b^6*c^4*d^13 + 252*a^3*b^6*c^6*d^11 + 488*a^3*b^6*c^8*d^9
+ 512*a^3*b^6*c^10*d^7 + 276*a^3*b^6*c^12*d^5 + 60*a^3*b^6*c^14*d^3 - 12*a^
4*b^5*c^3*d^14 + 40*a^4*b^5*c^5*d^12 + 40*a^4*b^5*c^7*d^10 - 60*a^4*b^5*c^9
*d^8 - 92*a^4*b^5*c^11*d^6 - 32*a^4*b^5*c^13*d^4 - 44*a^5*b^4*c^2*d^15 - 24
8*a^5*b^4*c^4*d^13 - 472*a^5*b^4*c^6*d^11 - 428*a^5*b^4*c^8*d^9 - 188*a^5*b
^4*c^10*d^7 - 32*a^5*b^4*c^12*d^5 + 172*a^6*b^3*c^3*d^14 + 408*a^6*b^3*c^5*
d^12 + 472*a^6*b^3*c^7*d^10 + 268*a^6*b^3*c^9*d^8 + 60*a^6*b^3*c^11*d^6 - 5
2*a^7*b^2*c^2*d^15 - 168*a^7*b^2*c^4*d^13 - 232*a^7*b^2*c^6*d^11 - 148*a^7*
b^2*c^8*d^9 - 36*a^7*b^2*c^10*d^7 + 8*a*b^8*c^16*d + 8*a^8*b*c*d^16)/(a^4*d
^12 + b^4*c^12 + 4*a^4*c^2*d^10 + 6*a^4*c^4*d^8 + 4*a^4*c^6*d^6 + a^4*c^8*d
^4 + b^4*c^4*d^8 + 4*b^4*c^6*d^6 + 6*b^4*c^8*d^4 + 4*b^4*c^10*d^2 - 4*a*b^3
*c^3*d^9 - 16*a*b^3*c^5*d^7 - 24*a*b^3*c^7*d^5 - 16*a*b^3*c^9*d^3 - 16*a^3*
b*c^3*d^9 - 24*a^3*b*c^5*d^7 - 16*a^3*b*c^7*d^5 - 4*a^3*b*c^9*d^3 + 6*a^2*b
^2*c^2*d^10 + 24*a^2*b^2*c^4*d^8 + 36*a^2*b^2*c^6*d^6 + 24*a^2*b^2*c^8*d^4
+ 6*a^2*b^2*c^10*d^2 - 4*a*b^3*c^11*d - 4*a^3*b*c*d^11) + (tan(e + f*x)*(6*
a^8*b*d^17 + 6*b^9*c^16*d + 8*a^4*b^5*d^17 + 6*a^6*b^3*d^17 + 8*b^9*c^4*d^1
3 + 38*b^9*c^6*d^11 + 78*b^9*c^8*d^9 + 92*b^9*c^10*d^7 + 68*b^9*c^12*d^5 +
30*b^9*c^14*d^3 - 32*a*b^8*c^3*d^14 - 148*a*b^8*c^5*d^12 - 292*a*b^8*c^7*d^
10 - 328*a*b^8*c^9*d^8 - 232*a*b^8*c^11*d^6 - 100*a*b^8*c^13*d^4 - 20*a*b^8
*c^15*d^2 - 2*a^2*b^7*c^16*d - 32*a^3*b^6*c*d^16 - 20*a^5*b^4*c*d^16 - 20*a
^7*b^2*c*d^16 + 22*a^8*b*c^2*d^15 + 28*a^8*b*c^4*d^13 + 12*a^8*b*c^6*d^11 -
2*a^8*b*c^8*d^9 - 2*a^8*b*c^10*d^7 + 48*a^2*b^7*c^2*d^15 + 218*a^2*b^7*c^4
*d^13 + 400*a^2*b^7*c^6*d^11 + 378*a^2*b^7*c^8*d^9 + 192*a^2*b^7*c^10*d^7 +
46*a^2*b^7*c^12*d^5 - 152*a^3*b^6*c^3*d^14 - 236*a^3*b^6*c^5*d^12 - 52*a^3
*b^6*c^7*d^10 + 232*a^3*b^6*c^9*d^8 + 256*a^3*b^6*c^11*d^6 + 100*a^3*b^6*c^
13*d^4 + 12*a^3*b^6*c^15*d^2 + 58*a^4*b^5*c^2*d^15 + 60*a^4*b^5*c^4*d^13 -
210*a^4*b^5*c^6*d^11 - 560*a^4*b^5*c^8*d^9 - 522*a^4*b^5*c^10*d^7 - 212*a^4
*b^5*c^12*d^5 - 30*a^4*b^5*c^14*d^3 - 28*a^5*b^4*c^3*d^14 + 128*a^5*b^4*c^5
*d^12 + 392*a^5*b^4*c^7*d^10 + 428*a^5*b^4*c^9*d^8 + 212*a^5*b^4*c^11*d^6 +
40*a^5*b^4*c^13*d^4 + 32*a^6*b^3*c^2*d^15 + 38*a^6*b^3*c^4*d^13 - 48*a^6*b
^3*c^6*d^11 - 142*a^6*b^3*c^8*d^9 - 112*a^6*b^3*c^10*d^7 - 30*a^6*b^3*c^12*
d^5 - 68*a^7*b^2*c^3*d^14 - 72*a^7*b^2*c^5*d^12 - 8*a^7*b^2*c^7*d^10 + 28*a
^7*b^2*c^9*d^8 + 12*a^7*b^2*c^11*d^6))/(a^4*d^12 + b^4*c^12 + 4*a^4*c^2*d^1
0 + 6*a^4*c^4*d^8 + 4*a^4*c^6*d^6 + a^4*c^8*d^4 + b^4*c^4*d^8 + 4*b^4*c^6*d
^6 + 6*b^4*c^8*d^4 + 4*b^4*c^10*d^2 - 4*a*b^3*c^3*d^9 - 16*a*b^3*c^5*d^7 -
24*a*b^3*c^7*d^5 - 16*a*b^3*c^9*d^3 - 16*a^3*b*c^3*d^9 - 24*a^3*b*c^5*d^7 -

16*a^3*b*c^7*d^5 - 4*a^3*b*c^9*d^3 + 6*a^2*b^2*c^2*d^10 + 24*a^2*b^2*c^4*d
^8 + 36*a^2*b^2*c^6*d^6 + 24*a^2*b^2*c^8*d^4 + 6*a^2*b^2*c^10*d^2 - 4*a*b^3
*c^11*d - 4*a^3*b*c*d^11)) + (B*a^7*b*d^14 - B*b^8*c^13*d - 4*A*a^2*b^6*d^1
4 + 4*A*a^4*b^4*d^14 - 3*A*a^6*b^2*d^14 + 4*B*a^3*b^5*d^14 - 4*B*a^5*b^3*d^
14 - 4*A*b^8*c^2*d^12 - 16*A*b^8*c^4*d^10 - 35*A*b^8*c^6*d^8 - 33*A*b^8*c^8
*d^6 - 5*A*b^8*c^10*d^4 + 5*A*b^8*c^12*d^2 - 4*C*a^4*b^4*d^14 + 3*C*a^6*b^2
*d^14 - 4*B*b^8*c^5*d^9 + 3*B*b^8*c^7*d^7 + 17*B*b^8*c^9*d^5 + 9*B*b^8*c^11
*d^3 + 11*C*b^8*c^6*d^8 + 17*C*b^8*c^8*d^6 + C*b^8*c^10*d^4 - 5*C*b^8*c^12*
d^2 + 40*A*a*b^7*c^3*d^11 + 122*A*a*b^7*c^5*d^9 + 175*A*a*b^7*c^7*d^7 + 105
*A*a*b^7*c^9*d^5 + 21*A*a*b^7*c^11*d^3 - 6*A*a^5*b^3*c*d^13 + 3*A*a^7*b*c^3
*d^11 + 3*A*a^7*b*c^5*d^9 + A*a^7*b*c^7*d^7 + 4*B*a*b^7*c^2*d^12 + 32*B*a*b
^7*c^4*d^10 + 31*B*a*b^7*c^6*d^8 - 27*B*a*b^7*c^8*d^6 - 39*B*a*b^7*c^10*d^4
 - 9*B*a*b^7*c^12*d^2 - 8*B*a^2*b^6*c*d^13 - 4*B*a^4*b^4*c*d^13 + 5*B*a^6*b
^2*c*d^13 + 3*B*a^7*b*c^2*d^12 + 3*B*a^7*b*c^4*d^10 + B*a^7*b*c^6*d^8 - 38*
C*a*b^7*c^5*d^9 - 79*C*a*b^7*c^7*d^7 - 41*C*a*b^7*c^9*d^5 + 3*C*a*b^7*c^11*
d^3 + 8*C*a^3*b^5*c*d^13 + 10*C*a^5*b^3*c*d^13 - 3*C*a^7*b*c^3*d^11 - 3*C*a
^7*b*c^5*d^9 - C*a^7*b*c^7*d^7 - 28*A*a^2*b^6*c^2*d^12 - 117*A*a^2*b^6*c^4*
d^10 - 245*A*a^2*b^6*c^6*d^8 - 237*A*a^2*b^6*c^8*d^6 - 91*A*a^2*b^6*c^10*d^
4 - 6*A*a^2*b^6*c^12*d^2 - 4*A*a^3*b^5*c^3*d^11 + 67*A*a^3*b^5*c^5*d^9 + 16
1*A*a^3*b^5*c^7*d^7 + 105*A*a^3*b^5*c^9*d^5 + 15*A*a^3*b^5*c^11*d^3 + 43*A*
a^4*b^4*c^2*d^12 + 69*A*a^4*b^4*c^4*d^10 + 5*A*a^4*b^4*c^6*d^8 - 45*A*a^4*b
^4*c^8*d^6 - 20*A*a^4*b^4*c^10*d^4 - 35*A*a^5*b^3*c^3*d^11 - 37*A*a^5*b^3*c
^5*d^9 + 7*A*a^5*b^3*c^7*d^7 + 15*A*a^5*b^3*c^9*d^5 + A*a^6*b^2*c^2*d^12 +
5*A*a^6*b^2*c^4*d^10 - 5*A*a^6*b^2*c^6*d^8 - 6*A*a^6*b^2*c^8*d^6 - 64*B*a^2
*b^6*c^3*d^11 - 145*B*a^2*b^6*c^5*d^9 - 115*B*a^2*b^6*c^7*d^7 - 11*B*a^2*b^
6*c^9*d^5 + 15*B*a^2*b^6*c^11*d^3 + 44*B*a^3*b^5*c^2*d^12 + 187*B*a^3*b^5*c
^4*d^10 + 273*B*a^3*b^5*c^6*d^8 + 141*B*a^3*b^5*c^8*d^6 + 15*B*a^3*b^5*c^10
*d^4 - 71*B*a^4*b^4*c^3*d^11 - 173*B*a^4*b^4*c^5*d^9 - 149*B*a^4*b^4*c^7*d^
7 - 43*B*a^4*b^4*c^9*d^5 - 11*B*a^5*b^3*c^2*d^12 + 23*B*a^5*b^3*c^4*d^10 +
63*B*a^5*b^3*c^6*d^8 + 33*B*a^5*b^3*c^8*d^6 - B*a^6*b^2*c^3*d^11 - 17*B*a^6
*b^2*c^5*d^9 - 11*B*a^6*b^2*c^7*d^7 - 4*C*a^2*b^6*c^2*d^12 + 25*C*a^2*b^6*c
^4*d^10 + 117*C*a^2*b^6*c^6*d^8 + 145*C*a^2*b^6*c^8*d^6 + 59*C*a^2*b^6*c^10
*d^4 + 2*C*a^2*b^6*c^12*d^2 + 36*C*a^3*b^5*c^3*d^11 - 19*C*a^3*b^5*c^5*d^9
- 129*C*a^3*b^5*c^7*d^7 - 97*C*a^3*b^5*c^9*d^5 - 15*C*a^3*b^5*c^11*d^3 - 47
*C*a^4*b^4*c^2*d^12 - 85*C*a^4*b^4*c^4*d^10 - 29*C*a^4*b^4*c^6*d^8 + 29*C*a
^4*b^4*c^8*d^6 + 16*C*a^4*b^4*c^10*d^4 + 51*C*a^5*b^3*c^3*d^11 + 61*C*a^5*b
^3*c^5*d^9 + 9*C*a^5*b^3*c^7*d^7 - 11*C*a^5*b^3*c^9*d^5 - C*a^6*b^2*c^2*d^1
2 - 5*C*a^6*b^2*c^4*d^10 + 5*C*a^6*b^2*c^6*d^8 + 6*C*a^6*b^2*c^8*d^6 + 8*A*
a*b^7*c*d^13 + A*a*b^7*c^13*d + A*a^7*b*c*d^13 + 3*C*a*b^7*c^13*d - C*a^7*b
*c*d^13)/(a^4*d^12 + b^4*c^12 + 4*a^4*c^2*d^10 + 6*a^4*c^4*d^8 + 4*a^4*c^6*
d^6 + a^4*c^8*d^4 + b^4*c^4*d^8 + 4*b^4*c^6*d^6 + 6*b^4*c^8*d^4 + 4*b^4*c^1
0*d^2 - 4*a*b^3*c^3*d^9 - 16*a*b^3*c^5*d^7 - 24*a*b^3*c^7*d^5 - 16*a*b^3*c^
9*d^3 - 16*a^3*b*c^3*d^9 - 24*a^3*b*c^5*d^7 - 16*a^3*b*c^7*d^5 - 4*a^3*b*c^
9*d^3 + 6*a^2*b^2*c^2*d^10 + 24*a^2*b^2*c^4*d^8 + 36*a^2*b^2*c^6*d^6 + 24*a
^2*b^2*c^8*d^4 + 6*a^2*b^2*c^10*d^2 - 4*a*b^3*c^11*d - 4*a^3*b*c*d^11) + (t

an(e + f*x)*(3*A*b^8*c^13*d - 3*A*a^7*b*d^14 + 3*C*a^7*b*d^14 + C*b^8*c^13*
d + 8*A*a^3*b^5*d^14 - 8*A*a^5*b^3*d^14 - 12*B*a^4*b^4*d^14 - B*a^6*b^2*d^1
4 + 8*A*b^8*c^3*d^11 + 24*A*b^8*c^5*d^9 + 51*A*b^8*c^7*d^7 + 65*A*b^8*c^9*d
^5 + 33*A*b^8*c^11*d^3 + 12*C*a^5*b^3*d^14 - 4*B*b^8*c^4*d^10 + 7*B*b^8*c^6
*d^8 + 21*B*b^8*c^8*d^6 + 5*B*b^8*c^10*d^4 - 5*B*b^8*c^12*d^2 + 12*C*b^8*c^
5*d^9 + 13*C*b^8*c^7*d^7 - 9*C*b^8*c^9*d^5 - 9*C*b^8*c^11*d^3 - 8*A*a*b^7*c
^2*d^12 + 8*A*a*b^7*c^4*d^10 + 3*A*a*b^7*c^6*d^8 - 63*A*a*b^7*c^8*d^6 - 63*
A*a*b^7*c^10*d^4 - 13*A*a*b^7*c^12*d^2 - 8*A*a^2*b^6*c*d^13 + 8*A*a^4*b^4*c
*d^13 + 13*A*a^6*b^2*c*d^13 - A*a^7*b*c^2*d^12 + 7*A*a^7*b*c^4*d^10 + 5*A*a
^7*b*c^6*d^8 + 8*B*a*b^7*c^3*d^11 - 50*B*a*b^7*c^5*d^9 - 143*B*a*b^7*c^7*d^
7 - 105*B*a*b^7*c^9*d^5 - 21*B*a*b^7*c^11*d^3 + 24*B*a^3*b^5*c*d^13 + 30*B*
a^5*b^3*c*d^13 + 13*B*a^7*b*c^3*d^11 + 5*B*a^7*b*c^5*d^9 - B*a^7*b*c^7*d^7
- 44*C*a*b^7*c^4*d^10 - 67*C*a*b^7*c^6*d^8 + 7*C*a*b^7*c^8*d^6 + 39*C*a*b^7
*c^10*d^4 + 9*C*a*b^7*c^12*d^2 - 12*C*a^4*b^4*c*d^13 - 13*C*a^6*b^2*c*d^13
+ C*a^7*b*c^2*d^12 - 7*C*a^7*b*c^4*d^10 - 5*C*a^7*b*c^6*d^8 - 96*A*a^2*b^6*
c^3*d^11 - 233*A*a^2*b^6*c^5*d^9 - 195*A*a^2*b^6*c^7*d^7 - 35*A*a^2*b^6*c^9
*d^5 + 15*A*a^2*b^6*c^11*d^3 + 64*A*a^3*b^5*c^2*d^12 + 263*A*a^3*b^5*c^4*d^
10 + 381*A*a^3*b^5*c^6*d^8 + 189*A*a^3*b^5*c^8*d^6 + 15*A*a^3*b^5*c^10*d^4
- 87*A*a^4*b^4*c^3*d^11 - 253*A*a^4*b^4*c^5*d^9 - 213*A*a^4*b^4*c^7*d^7 - 5
5*A*a^4*b^4*c^9*d^5 - 7*A*a^5*b^3*c^2*d^12 + 67*A*a^5*b^3*c^4*d^10 + 123*A*
a^5*b^3*c^6*d^8 + 57*A*a^5*b^3*c^8*d^6 - A*a^6*b^2*c^3*d^11 - 41*A*a^6*b^2*
c^5*d^9 - 27*A*a^6*b^2*c^7*d^7 - 16*B*a^2*b^6*c^2*d^12 + 17*B*a^2*b^6*c^4*d
^10 + 161*B*a^2*b^6*c^6*d^8 + 213*B*a^2*b^6*c^8*d^6 + 91*B*a^2*b^6*c^10*d^4
 + 6*B*a^2*b^6*c^12*d^2 + 116*B*a^3*b^5*c^3*d^11 + 85*B*a^3*b^5*c^5*d^9 - 9
7*B*a^3*b^5*c^7*d^7 - 105*B*a^3*b^5*c^9*d^5 - 15*B*a^3*b^5*c^11*d^3 - 119*B
*a^4*b^4*c^2*d^12 - 209*B*a^4*b^4*c^4*d^10 - 89*B*a^4*b^4*c^6*d^8 + 33*B*a^
4*b^4*c^8*d^6 + 20*B*a^4*b^4*c^10*d^4 + 115*B*a^5*b^3*c^3*d^11 + 125*B*a^5*
b^3*c^5*d^9 + 25*B*a^5*b^3*c^7*d^7 - 15*B*a^5*b^3*c^9*d^5 - 37*B*a^6*b^2*c^
2*d^12 - 65*B*a^6*b^2*c^4*d^10 - 23*B*a^6*b^2*c^6*d^8 + 6*B*a^6*b^2*c^8*d^6
 + 64*C*a^2*b^6*c^3*d^11 + 185*C*a^2*b^6*c^5*d^9 + 163*C*a^2*b^6*c^7*d^7 +
27*C*a^2*b^6*c^9*d^5 - 15*C*a^2*b^6*c^11*d^3 - 32*C*a^3*b^5*c^2*d^12 - 215*
C*a^3*b^5*c^4*d^10 - 349*C*a^3*b^5*c^6*d^8 - 181*C*a^3*b^5*c^8*d^6 - 15*C*a
^3*b^5*c^10*d^4 + 71*C*a^4*b^4*c^3*d^11 + 229*C*a^4*b^4*c^5*d^9 + 197*C*a^4
*b^4*c^7*d^7 + 51*C*a^4*b^4*c^9*d^5 + 23*C*a^5*b^3*c^2*d^12 - 43*C*a^5*b^3*
c^4*d^10 - 107*C*a^5*b^3*c^6*d^8 - 53*C*a^5*b^3*c^8*d^6 + C*a^6*b^2*c^3*d^1
1 + 41*C*a^6*b^2*c^5*d^9 + 27*C*a^6*b^2*c^7*d^7 - B*a*b^7*c^13*d + 7*B*a^7*
b*c*d^13))/(a^4*d^12 + b^4*c^12 + 4*a^4*c^2*d^10 + 6*a^4*c^4*d^8 + 4*a^4*c^
6*d^6 + a^4*c^8*d^4 + b^4*c^4*d^8 + 4*b^4*c^6*d^6 + 6*b^4*c^8*d^4 + 4*b^4*c
^10*d^2 - 4*a*b^3*c^3*d^9 - 16*a*b^3*c^5*d^7 - 24*a*b^3*c^7*d^5 - 16*a*b^3*
c^9*d^3 - 16*a^3*b*c^3*d^9 - 24*a^3*b*c^5*d^7 - 16*a^3*b*c^7*d^5 - 4*a^3*b*
c^9*d^3 + 6*a^2*b^2*c^2*d^10 + 24*a^2*b^2*c^4*d^8 + 36*a^2*b^2*c^6*d^6 + 24
*a^2*b^2*c^8*d^4 + 6*a^2*b^2*c^10*d^2 - 4*a*b^3*c^11*d - 4*a^3*b*c*d^11)) -
 (4*A^2*a^3*b^4*d^11 - A^2*a^5*b^2*d^11 - B^2*a^5*b^2*d^11 - 28*A^2*b^7*c^3
*d^8 - 45*A^2*b^7*c^5*d^6 - 24*A^2*b^7*c^7*d^4 + A^2*b^7*c^9*d^2 - C^2*a^5*
b^2*d^11 - B^2*b^7*c^5*d^6 - 3*B^2*b^7*c^9*d^2 - C^2*b^7*c^5*d^6 - 4*C^2*b^

7*c^7*d^4 + C^2*b^7*c^9*d^2 - 4*A^2*a*b^6*d^11 - 4*A^2*b^7*c*d^10 + 14*A^2*a^2*b^5*c^3*d^8 - 154*A^2*a^2*b^5*c^5*d^6 + 28*A^2*a^2*b^5*c^7*d^4 - 26*A^2*a^3*b^4*c^2*d^9 + 72*A^2*a^3*b^4*c^4*d^7 - 42*A^2*a^3*b^4*c^6*d^5 - 24*A^2*a^4*b^3*c^3*d^8 + 33*A^2*a^4*b^3*c^5*d^6 + 10*A^2*a^5*b^2*c^2*d^9 - 13*A^2*a^5*b^2*c^4*d^7 - 46*B^2*a^2*b^5*c^3*d^8 + 102*B^2*a^2*b^5*c^5*d^6 - 52*B^2*a^2*b^5*c^7*d^4 + 34*B^2*a^3*b^4*c^2*d^9 - 68*B^2*a^3*b^4*c^4*d^7 + 42*B^2*a^3*b^4*c^6*d^5 + 36*B^2*a^4*b^3*c^3*d^8 - 27*B^2*a^4*b^3*c^5*d^6 - 14*B^2*a^5*b^2*c^2*d^9 + 11*B^2*a^5*b^2*c^4*d^7 + 10*C^2*a^2*b^5*c^3*d^8 - 134*C^2*a^2*b^5*c^5*d^6 + 48*C^2*a^2*b^5*c^7*d^4 + 4*C^2*a^2*b^5*c^9*d^2 - 22*C^2*a^3*b^4*c^2*d^9 + 92*C^2*a^3*b^4*c^4*d^7 - 30*C^2*a^3*b^4*c^6*d^5 - 24*C^2*a^4*b^3*c^3*d^8 + 33*C^2*a^4*b^3*c^5*d^6 + 10*C^2*a^5*b^2*c^2*d^9 - 13*C^2*a^5*b^2*c^4*d^7 + 4*A*B*a^2*b^5*d^11 - 4*A*C*a^3*b^4*d^11 + 2*A*C*a^5*b^2*d^11 - 4*A*B*b^7*c^2*d^9 + 4*A*B*b^7*c^4*d^7 + 19*A*B*b^7*c^6*d^5 + 18*A*B*b^7*c^8*d^3 + 12*A*C*b^7*c^3*d^8 + 22*A*C*b^7*c^5*d^6 + 12*A*C*b^7*c^7*d^4 - 6*A*C*b^7*c^9*d^2 + B*C*b^7*c^6*d^5 - 6*B*C*b^7*c^8*d^3 - 2*A^2*a^6*b*c*d^10 + 2*B^2*a^6*b*c*d^10 + 4*C^2*a*b^6*c^10*d - 2*C^2*a^6*b*c*d^10 + 8*A^2*a*b^6*c^2*d^9 + 63*A^2*a*b^6*c^4*d^7 + 130*A^2*a*b^6*c^6*d^5 - 9*A^2*a*b^6*c^8*d^3 + 8*A^2*a^2*b^5*c*d^10 + 3*A^2*a^4*b^3*c*d^10 + 2*A^2*a^6*b*c^3*d^8 + 4*B^2*a*b^6*c^2*d^9 + 3*B^2*a*b^6*c^4*d^7 - 50*B^2*a*b^6*c^6*d^5 + 39*B^2*a*b^6*c^8*d^3 - 12*B^2*a^2*b^5*c*d^10 + 3*B^2*a^4*b^3*c*d^10 - 2*B^2*a^6*b*c^3*d^8 + 3*C^2*a*b^6*c^4*d^7 + 54*C^2*a*b^6*c^6*d^5 - 33*C^2*a*b^6*c^8*d^3 + 3*C^2*a^4*b^3*c*d^10 + 2*C^2*a^6*b*c^3*d^8 - A*B*a^6*b*d^11 - A*B*b^7*c^10*d + B*C*a^6*b*d^11 + B*C*b^7*c^10*d + 16*A*B*a*b^6*c*d^10 + 4*A*C*a^6*b*c*d^10 + 56*A*B*a*b^6*c^3*d^8 + 70*A*B*a*b^6*c^5*d^6 - 140*A*B*a*b^6*c^7*d^4 + 6*A*B*a*b^6*c^9*d^2 - 24*A*B*a^3*b^4*c*d^10 + 6*A*B*a^5*b^2*c*d^10 + 6*A*B*a^6*b*c^2*d^9 - A*B*a^6*b*c^4*d^7 - 20*A*C*a*b^6*c^2*d^9 - 74*A*C*a*b^6*c^4*d^7 - 176*A*C*a*b^6*c^6*d^5 + 54*A*C*a*b^6*c^8*d^3 - 4*A*C*a^2*b^5*c*d^10 - 6*A*C*a^4*b^3*c*d^10 - 4*A*C*a^6*b*c^3*d^8 - 12*B*C*a*b^6*c^3*d^8 - 50*B*C*a*b^6*c^5*d^6 + 112*B*C*a*b^6*c^7*d^4 - 26*B*C*a*b^6*c^9*d^2 + 12*B*C*a^3*b^4*c*d^10 - 6*B*C*a^5*b^2*c*d^10 - 6*B*C*a^6*b*c^2*d^9 + B*C*a^6*b*c^4*d^7 - 20*A*B*a^2*b^5*c^2*d^9 - 195*A*B*a^2*b^5*c^4*d^7 + 190*A*B*a^2*b^5*c^6*d^5 - 15*A*B*a^2*b^5*c^8*d^3 + 100*A*B*a^3*b^4*c^3*d^8 - 144*A*B*a^3*b^4*c^5*d^6 + 20*A*B*a^3*b^4*c^7*d^4 - 15*A*B*a^4*b^3*c^2*d^9 + 90*A*B*a^4*b^3*c^4*d^7 - 15*A*B*a^4*b^3*c^6*d^5 - 36*A*B*a^5*b^2*c^3*d^8 + 6*A*B*a^5*b^2*c^5*d^6 - 8*A*C*a^2*b^5*c^3*d^8 + 312*A*C*a^2*b^5*c^5*d^6 - 60*A*C*a^2*b^5*c^7*d^4 + 48*A*C*a^3*b^4*c^2*d^9 - 164*A*C*a^3*b^4*c^4*d^7 + 72*A*C*a^3*b^4*c^6*d^5 + 48*A*C*a^4*b^3*c^3*d^8 - 66*A*C*a^4*b^3*c^5*d^6 - 20*A*C*a^5*b^2*c^2*d^9 + 26*A*C*a^5*b^2*c^4*d^7 + 16*B*C*a^2*b^5*c^2*d^9 + 175*B*C*a^2*b^5*c^4*d^7 - 202*B*C*a^2*b^5*c^6*d^5 + 15*B*C*a^2*b^5*c^8*d^3 - 120*B*C*a^3*b^4*c^3*d^8 + 140*B*C*a^3*b^4*c^5*d^6 - 16*B*C*a^3*b^4*c^7*d^4 + 15*B*C*a^4*b^3*c^2*d^9 - 90*B*C*a^4*b^3*c^4*d^7 + 15*B*C*a^4*b^3*c^6*d^5 + 36*B*C*a^5*b^2*c^3*d^8 - 6*B*C*a^5*b^2*c^5*d^6)/(a^4*d^12 + b^4*c^12 + 4*a^4*c^2*d^10 + 6*a^4*c^4*d^8 + 4*a^4*c^6*d^6 + a^4*c^8*d^4 + b^4*c^4*d^8 + 4*b^4*c^6*d^6 + 6*b^4*c^8*d^4 + 4*b^4*c^10*d^2 - 4*a*b^3*c^3*d^9 - 16*a*b^3*c^5*d^7 - 24*a*b^3*c^7*d^5 - 16*a*b^3*c^9*d^3 - 16*a^3*b*c^3*d^9 - 24*a^3*b*c^5*d^

7 - 16*a^3*b*c^7*d^5 - 4*a^3*b*c^9*d^3 + 6*a^2*b^2*c^2*d^10 + 24*a^2*b^2*c^
4*d^8 + 36*a^2*b^2*c^6*d^6 + 24*a^2*b^2*c^8*d^4 + 6*a^2*b^2*c^10*d^2 - 4*a*
b^3*c^11*d - 4*a^3*b*c*d^11) + (tan(e + f*x)*(2*A^2*b^7*d^11 - 6*A^2*a^2*b^
5*d^11 + 2*A^2*a^4*b^3*d^11 + 2*B^2*a^2*b^5*d^11 + 2*B^2*a^4*b^3*d^11 + 6*A
^2*b^7*c^2*d^9 - 12*A^2*b^7*c^4*d^7 - 66*A^2*b^7*c^6*d^5 + 18*A^2*b^7*c^8*d
^3 + 4*C^2*a^4*b^3*d^11 - 2*B^2*b^7*c^4*d^7 + 29*B^2*b^7*c^6*d^5 - 36*B^2*b
^7*c^8*d^3 + 2*C^2*b^7*c^4*d^7 - 32*C^2*b^7*c^6*d^5 + 30*C^2*b^7*c^8*d^3 +
B^2*a^6*b*d^11 + B^2*b^7*c^10*d - 4*C^2*b^7*c^10*d + 38*A^2*a^2*b^5*c^2*d^9
 - 2*A^2*a^2*b^5*c^4*d^7 + 78*A^2*a^2*b^5*c^6*d^5 - 16*A^2*a^3*b^4*c^3*d^8
- 88*A^2*a^3*b^4*c^5*d^6 + 4*A^2*a^4*b^3*c^2*d^9 + 62*A^2*a^4*b^3*c^4*d^7 -
 24*A^2*a^5*b^2*c^3*d^8 - 8*B^2*a^2*b^5*c^2*d^9 + 83*B^2*a^2*b^5*c^4*d^7 -
22*B^2*a^2*b^5*c^6*d^5 + 9*B^2*a^2*b^5*c^8*d^3 - 46*B^2*a^3*b^4*c^3*d^8 + 3
0*B^2*a^3*b^4*c^5*d^6 - 18*B^2*a^3*b^4*c^7*d^4 + 19*B^2*a^4*b^3*c^2*d^9 - 2
8*B^2*a^4*b^3*c^4*d^7 + 15*B^2*a^4*b^3*c^6*d^5 + 12*B^2*a^5*b^2*c^3*d^8 - 6
*B^2*a^5*b^2*c^5*d^6 + 12*C^2*a^2*b^5*c^2*d^9 - 82*C^2*a^2*b^5*c^4*d^7 + 22
*C^2*a^2*b^5*c^6*d^5 - 6*C^2*a^2*b^5*c^8*d^3 - 56*C^2*a^3*b^4*c^5*d^6 + 16*
C^2*a^3*b^4*c^7*d^4 + 2*C^2*a^4*b^3*c^2*d^9 + 52*C^2*a^4*b^3*c^4*d^7 - 6*C^
2*a^4*b^3*c^6*d^5 - 24*C^2*a^5*b^2*c^3*d^8 + 2*A*B*a^3*b^4*d^11 + 4*A*C*a^2
*b^5*d^11 - 6*A*C*a^4*b^3*d^11 - 6*A*B*b^7*c^3*d^8 - 18*A*B*b^7*c^5*d^6 + 1
14*A*B*b^7*c^7*d^4 - 10*A*B*b^7*c^9*d^2 - 4*B*C*a^3*b^4*d^11 + 14*A*C*b^7*c
^4*d^7 + 94*A*C*b^7*c^6*d^5 - 54*A*C*b^7*c^8*d^3 + 24*B*C*b^7*c^5*d^6 - 84*
B*C*b^7*c^7*d^4 + 28*B*C*b^7*c^9*d^2 - 8*A^2*a*b^6*c*d^10 - 40*A^2*a*b^6*c^
3*d^8 + 72*A^2*a*b^6*c^5*d^6 - 48*A^2*a*b^6*c^7*d^4 - 8*A^2*a^3*b^4*c*d^10
+ 4*A^2*a^6*b*c^2*d^9 + 14*B^2*a*b^6*c^3*d^8 - 100*B^2*a*b^6*c^5*d^6 + 38*B
^2*a*b^6*c^7*d^4 - 14*B^2*a^3*b^4*c*d^10 - 6*B^2*a^5*b^2*c*d^10 - 2*B^2*a^6
*b*c^2*d^9 + B^2*a^6*b*c^4*d^7 - 8*C^2*a*b^6*c^3*d^8 + 104*C^2*a*b^6*c^5*d^
6 - 48*C^2*a*b^6*c^7*d^4 - 8*C^2*a*b^6*c^9*d^2 + 2*C^2*a^2*b^5*c^10*d - 8*C
^2*a^3*b^4*c*d^10 + 4*C^2*a^6*b*c^2*d^9 - 4*A*B*a*b^6*d^11 + 2*A*C*b^7*c^10
*d + 4*A*B*a^6*b*c*d^10 - 2*B*C*a*b^6*c^10*d - 4*B*C*a^6*b*c*d^10 - 10*A*B*
a*b^6*c^2*d^9 + 114*A*B*a*b^6*c^4*d^7 - 166*A*B*a*b^6*c^6*d^5 + 18*A*B*a*b^
6*c^8*d^3 + 30*A*B*a^2*b^5*c*d^10 - 4*A*B*a^6*b*c^3*d^8 + 16*A*C*a*b^6*c^3*
d^8 - 224*A*C*a*b^6*c^5*d^6 + 64*A*C*a*b^6*c^7*d^4 + 16*A*C*a^3*b^4*c*d^10
- 8*A*C*a^6*b*c^2*d^9 - 106*B*C*a*b^6*c^4*d^7 + 194*B*C*a*b^6*c^6*d^5 - 6*B
*C*a*b^6*c^8*d^3 + 6*B*C*a^4*b^3*c*d^10 + 4*B*C*a^6*b*c^3*d^8 - 54*A*B*a^2*
b^5*c^3*d^8 + 118*A*B*a^2*b^5*c^5*d^6 - 46*A*B*a^2*b^5*c^7*d^4 - 2*A*B*a^3*
b^4*c^2*d^9 - 90*A*B*a^3*b^4*c^4*d^7 + 74*A*B*a^3*b^4*c^6*d^5 + 60*A*B*a^4*
b^3*c^3*d^8 - 60*A*B*a^4*b^3*c^5*d^6 - 24*A*B*a^5*b^2*c^2*d^9 + 24*A*B*a^5*
b^2*c^4*d^7 - 56*A*C*a^2*b^5*c^2*d^9 + 80*A*C*a^2*b^5*c^4*d^7 - 96*A*C*a^2*
b^5*c^6*d^5 + 12*A*C*a^2*b^5*c^8*d^3 + 16*A*C*a^3*b^4*c^3*d^8 + 144*A*C*a^3
*b^4*c^5*d^6 - 16*A*C*a^3*b^4*c^7*d^4 - 6*A*C*a^4*b^3*c^2*d^9 - 114*A*C*a^4
*b^3*c^4*d^7 + 6*A*C*a^4*b^3*c^6*d^5 + 48*A*C*a^5*b^2*c^3*d^8 + 106*B*C*a^2
*b^5*c^3*d^8 - 110*B*C*a^2*b^5*c^5*d^6 + 26*B*C*a^2*b^5*c^7*d^4 - 6*B*C*a^2
*b^5*c^9*d^2 - 14*B*C*a^3*b^4*c^2*d^9 + 70*B*C*a^3*b^4*c^4*d^7 - 74*B*C*a^3
*b^4*c^6*d^5 + 6*B*C*a^3*b^4*c^8*d^3 - 50*B*C*a^4*b^3*c^3*d^8 + 62*B*C*a^4*
b^3*c^5*d^6 - 2*B*C*a^4*b^3*c^7*d^4 + 24*B*C*a^5*b^2*c^2*d^9 - 24*B*C*a^5*b

^2*c^4*d^7))/(a^4*d^12 + b^4*c^12 + 4*a^4*c^2*d^10 + 6*a^4*c^4*d^8 + 4*a^4*
c^6*d^6 + a^4*c^8*d^4 + b^4*c^4*d^8 + 4*b^4*c^6*d^6 + 6*b^4*c^8*d^4 + 4*b^4
*c^10*d^2 - 4*a*b^3*c^3*d^9 - 16*a*b^3*c^5*d^7 - 24*a*b^3*c^7*d^5 - 16*a*b^
3*c^9*d^3 - 16*a^3*b*c^3*d^9 - 24*a^3*b*c^5*d^7 - 16*a^3*b*c^7*d^5 - 4*a^3*
b*c^9*d^3 + 6*a^2*b^2*c^2*d^10 + 24*a^2*b^2*c^4*d^8 + 36*a^2*b^2*c^6*d^6 +
24*a^2*b^2*c^8*d^4 + 6*a^2*b^2*c^10*d^2 - 4*a*b^3*c^11*d - 4*a^3*b*c*d^11))
 - (A^3*a^2*b^4*d^8 - A^3*b^6*d^8 - 4*A^3*b^6*c^2*d^6 - 7*A^3*b^6*c^4*d^4 +
 A^2*C*b^6*d^8 - 3*A^3*a^2*b^4*c^2*d^6 - B^3*a^2*b^4*c^3*d^5 - C^3*a^2*b^4*
c^2*d^6 + 7*C^3*a^2*b^4*c^4*d^4 - 2*C^3*a^3*b^3*c^3*d^5 + A^2*B*a*b^5*d^8 +
 A^2*B*b^6*c*d^7 + A^3*a*b^5*c*d^7 + C^3*a*b^5*c^7*d + A*C^2*a^2*b^4*d^8 -
2*A^2*C*a^2*b^4*d^8 - A*B^2*b^6*c^2*d^6 - 3*A*B^2*b^6*c^6*d^2 - B*C^2*a^3*b
^3*d^8 + 2*A^2*B*b^6*c^3*d^5 + 9*A^2*B*b^6*c^5*d^3 + B^2*C*a^2*b^4*d^8 - A*
C^2*b^6*c^2*d^6 - 4*A*C^2*b^6*c^4*d^4 + A*C^2*b^6*c^6*d^2 + 5*A^2*C*b^6*c^2
*d^6 + 11*A^2*C*b^6*c^4*d^4 - A^2*C*b^6*c^6*d^2 + 9*A^3*a*b^5*c^3*d^5 + B^3
*a*b^5*c^2*d^6 + B^3*a*b^5*c^4*d^4 - B^3*a^2*b^4*c*d^7 - 3*C^3*a*b^5*c^5*d^
3 + 2*C^3*a^3*b^3*c*d^7 - 2*A*B*C*a*b^5*d^8 + A*B*C*b^6*c^7*d + 3*A*B^2*a^2
*b^4*c^2*d^6 - A*B^2*a^2*b^4*c^4*d^4 + 3*A^2*B*a^2*b^4*c^3*d^5 - A*C^2*a^2*
b^4*c^2*d^6 - 14*A*C^2*a^2*b^4*c^4*d^4 + 4*A*C^2*a^3*b^3*c^3*d^5 + 5*A^2*C*
a^2*b^4*c^2*d^6 + 7*A^2*C*a^2*b^4*c^4*d^4 - 2*A^2*C*a^3*b^3*c^3*d^5 - 15*B*
C^2*a^2*b^4*c^3*d^5 + 3*B*C^2*a^2*b^4*c^5*d^3 + 6*B*C^2*a^3*b^3*c^2*d^6 - B
*C^2*a^3*b^3*c^4*d^4 + 5*B^2*C*a^2*b^4*c^2*d^6 - 4*B^2*C*a^2*b^4*c^4*d^4 +
2*B^2*C*a^3*b^3*c^3*d^5 + A*B*C*a^3*b^3*d^8 + A*B*C*b^6*c^3*d^5 - 6*A*B*C*b
^6*c^5*d^3 + 2*A*C^2*a*b^5*c*d^7 - A*C^2*a*b^5*c^7*d - 3*A^2*C*a*b^5*c*d^7
 - 5*A*B^2*a*b^5*c^3*d^5 + 3*A*B^2*a*b^5*c^5*d^3 + 7*A^2*B*a*b^5*c^2*d^6 - 1
0*A^2*B*a*b^5*c^4*d^4 - 5*A^2*B*a^2*b^4*c*d^7 + 12*A*C^2*a*b^5*c^3*d^5 + 9*
A*C^2*a*b^5*c^5*d^3 - 4*A*C^2*a^3*b^3*c*d^7 - 21*A^2*C*a*b^5*c^3*d^5 - 6*A^
2*C*a*b^5*c^5*d^3 + 2*A^2*C*a^3*b^3*c*d^7 + B*C^2*a*b^5*c^2*d^6 + 5*B*C^2*a
*b^5*c^4*d^4 - 4*B*C^2*a*b^5*c^6*d^2 - 2*B*C^2*a^2*b^4*c*d^7 - B^2*C*a*b^5*
c^3*d^5 + 3*B^2*C*a*b^5*c^5*d^3 - 2*B^2*C*a^3*b^3*c*d^7 + 12*A*B*C*a^2*b^4*
c^3*d^5 - 3*A*B*C*a^2*b^4*c^5*d^3 - 6*A*B*C*a^3*b^3*c^2*d^6 + A*B*C*a^3*b^3
*c^4*d^4 - 11*A*B*C*a*b^5*c^2*d^6 + 2*A*B*C*a*b^5*c^4*d^4 + 3*A*B*C*a*b^5*c
^6*d^2 + 7*A*B*C*a^2*b^4*c*d^7)/(a^4*d^12 + b^4*c^12 + 4*a^4*c^2*d^10 + 6*a
^4*c^4*d^8 + 4*a^4*c^6*d^6 + a^4*c^8*d^4 + b^4*c^4*d^8 + 4*b^4*c^6*d^6 + 6*
b^4*c^8*d^4 + 4*b^4*c^10*d^2 - 4*a*b^3*c^3*d^9 - 16*a*b^3*c^5*d^7 - 24*a*b^
3*c^7*d^5 - 16*a*b^3*c^9*d^3 - 16*a^3*b*c^3*d^9 - 24*a^3*b*c^5*d^7 - 16*a^3
*b*c^7*d^5 - 4*a^3*b*c^9*d^3 + 6*a^2*b^2*c^2*d^10 + 24*a^2*b^2*c^4*d^8 + 36
*a^2*b^2*c^6*d^6 + 24*a^2*b^2*c^8*d^4 + 6*a^2*b^2*c^10*d^2 - 4*a*b^3*c^11*d
 - 4*a^3*b*c*d^11) - (tan(e + f*x)*(B^3*b^6*c^4*d^4 - A^3*b^6*c^3*d^5 - B^3
*a^2*b^4*d^8 - 3*B^3*b^6*c^6*d^2 - 3*C^3*b^6*c^5*d^3 - A^2*B*b^6*d^8 + A^3*
a*b^5*d^8 - A^3*b^6*c*d^7 + C^3*b^6*c^7*d + 2*B^3*a^2*b^4*c^2*d^6 - B^3*a^2
*b^4*c^4*d^4 - 12*C^3*a^2*b^4*c^3*d^5 + 4*C^3*a^3*b^3*c^2*d^6 + 2*A*B^2*a*b
^5*d^8 - A^2*C*a*b^5*d^8 - A*C^2*b^6*c^7*d + A^2*C*b^6*c*d^7 + B^2*C*b^6*c^
7*d + A*B^2*b^6*c^3*d^5 + 9*A*B^2*b^6*c^5*d^3 - 3*A^2*B*b^6*c^2*d^6 - 6*A^2
*B*b^6*c^4*d^4 + B^2*C*a^3*b^3*d^8 + 2*A*C^2*b^6*c^3*d^5 + 9*A*C^2*b^6*c^5*
d^3 - A^2*C*b^6*c^3*d^5 - 6*A^2*C*b^6*c^5*d^3 + B*C^2*b^6*c^4*d^4 - 3*B*C^2

*b^6*c^6*d^2 - 3*B^2*C*b^6*c^5*d^3 + A^3*a*b^5*c^2*d^6 - 5*B^3*a*b^5*c^3*d^5 + 3*B^3*a*b^5*c^5*d^3 + 11*C^3*a*b^5*c^4*d^4 - C^3*a*b^5*c^6*d^2 + 4*A*B^2*a^2*b^4*c^3*d^5 - 4*A^2*B*a^2*b^4*c^2*d^6 + 24*A*C^2*a^2*b^4*c^3*d^5 - 8*A*C^2*a^3*b^3*c^2*d^6 - 12*A^2*C*a^2*b^4*c^3*d^5 + 4*A^2*C*a^3*b^3*c^2*d^6 + 8*B*C^2*a^2*b^4*c^2*d^6 - 12*B*C^2*a^2*b^4*c^4*d^4 + 4*B*C^2*a^3*b^3*c^3*d^5 + 2*B^2*C*a^2*b^4*c^3*d^5 - 3*B^2*C*a^2*b^4*c^5*d^3 - 2*B^2*C*a^3*b^3*c^2*d^6 + B^2*C*a^3*b^3*c^4*d^4 + 2*A*B*C*b^6*c^4*d^4 + 2*A*B*C*b^6*c^6*d^2 + A^2*B*a*b^5*c*d^7 - B*C^2*a*b^5*c^7*d + 7*A*B^2*a*b^5*c^2*d^6 - 11*A*B^2*a*b^5*c^4*d^4 - 4*A*B^2*a^2*b^4*c*d^7 + 9*A^2*B*a*b^5*c^3*d^5 - 2*A*C^2*a*b^5*c^2*d^6 - 25*A*C^2*a*b^5*c^4*d^4 + A*C^2*a*b^5*c^6*d^2 + A^2*C*a*b^5*c^2*d^6 + 14*A^2*C*a*b^5*c^4*d^4 - 6*B*C^2*a*b^5*c^3*d^5 + 9*B*C^2*a*b^5*c^5*d^3 - 4*B*C^2*a^3*b^3*c*d^7 + 7*B^2*C*a*b^5*c^4*d^4 + 3*B^2*C*a*b^5*c^6*d^2 + B^2*C*a^2*b^4*c*d^7 - 4*A*B*C*a^2*b^4*c^2*d^6 + 12*A*B*C*a^2*b^4*c^4*d^4 - 4*A*B*C*a^3*b^3*c^3*d^5 - 2*A*B*C*a*b^5*c*d^7 - 6*A*B*C*a*b^5*c^3*d^5 - 12*A*B*C*a*b^5*c^5*d^3 + 4*A*B*C*a^3*b^3*c*d^7))/(a^4*d^12 + b^4*c^12 + 4*a^4*c^2*d^10 + 6*a^4*c^4*d^8 + 4*a^4*c^6*d^6 + a^4*c^8*d^4 + b^4*c^4*d^8 + 4*b^4*c^6*d^6 + 6*b^4*c^8*d^4 + 4*b^4*c^10*d^2 - 4*a*b^3*c^3*d^9 - 16*a*b^3*c^5*d^7 - 24*a*b^3*c^7*d^5 - 16*a*b^3*c^9*d^3 - 16*a^3*b*c^3*d^9 - 24*a^3*b*c^5*d^7 - 16*a^3*b*c^7*d^5 - 4*a^3*b*c^9*d^3 + 6*a^2*b^2*c^2*d^10 + 24*a^2*b^2*c^4*d^8 + 36*a^2*b^2*c^6*d^6 + 24*a^2*b^2*c^8*d^4 + 6*a^2*b^2*c^10*d^2 - 4*a*b^3*c^11*d - 4*a^3*b*c*d^11))*root(480*a^9*b*c^7*d^11*f^4 + 480*a*b^9*c^11*d^7*f^4 + 360*a^9*b*c^9*d^9*f^4 + 360*a^9*b*c^5*d^13*f^4 + 360*a*b^9*c^13*d^5*f^4 + 360*a*b^9*c^9*d^9*f^4 + 144*a^9*b*c^11*d^7*f^4 + 144*a^9*b*c^3*d^15*f^4 + 144*a*b^9*c^15*d^3*f^4 + 144*a*b^9*c^7*d^11*f^4 + 48*a^7*b^3*c*d^17*f^4 + 48*a^3*b^7*c^17*d*f^4 + 24*a^9*b*c^13*d^5*f^4 + 24*a^5*b^5*c^17*d*f^4 + 24*a^5*b^5*c*d^17*f^4 + 24*a*b^9*c^5*d^13*f^4 + 24*a^9*b*c*d^17*f^4 + 24*a*b^9*c^17*d*f^4 + 3920*a^5*b^5*c^9*d^9*f^4 - 3360*a^6*b^4*c^8*d^10*f^4 - 3360*a^4*b^6*c^10*d^8*f^4 - 3024*a^6*b^4*c^10*d^8*f^4 + 3024*a^5*b^5*c^11*d^7*f^4 + 3024*a^5*b^5*c^7*d^11*f^4 - 3024*a^4*b^6*c^8*d^10*f^4 + 2320*a^7*b^3*c^9*d^9*f^4 + 2320*a^3*b^7*c^9*d^9*f^4 - 2240*a^6*b^4*c^6*d^12*f^4 - 2240*a^4*b^6*c^12*d^6*f^4 + 2160*a^7*b^3*c^7*d^11*f^4 + 2160*a^3*b^7*c^11*d^7*f^4 - 1624*a^6*b^4*c^12*d^6*f^4 - 1624*a^4*b^6*c^6*d^12*f^4 + 1488*a^7*b^3*c^11*d^7*f^4 + 1488*a^3*b^7*c^7*d^11*f^4 + 1344*a^5*b^5*c^13*d^5*f^4 + 1344*a^5*b^5*c^5*d^13*f^4 - 1320*a^8*b^2*c^8*d^10*f^4 - 1320*a^2*b^8*c^10*d^8*f^4 + 1200*a^7*b^3*c^5*d^13*f^4 + 1200*a^3*b^7*c^13*d^5*f^4 - 1060*a^8*b^2*c^6*d^12*f^4 - 1060*a^2*b^8*c^12*d^6*f^4 - 948*a^8*b^2*c^10*d^8*f^4 - 948*a^2*b^8*c^8*d^10*f^4 - 840*a^6*b^4*c^4*d^14*f^4 - 840*a^4*b^6*c^14*d^4*f^4 + 528*a^7*b^3*c^13*d^5*f^4 + 528*a^3*b^7*c^5*d^13*f^4 - 480*a^8*b^2*c^4*d^14*f^4 - 480*a^6*b^4*c^14*d^4*f^4 - 480*a^4*b^6*c^4*d^14*f^4 - 480*a^2*b^8*c^14*d^4*f^4 - 368*a^8*b^2*c^12*d^6*f^4 + 368*a^7*b^3*c^3*d^15*f^4 + 368*a^3*b^7*c^15*d^3*f^4 - 368*a^2*b^8*c^6*d^12*f^4 + 304*a^5*b^5*c^15*d^3*f^4 + 304*a^5*b^5*c^3*d^15*f^4 - 144*a^6*b^4*c^2*d^16*f^4 - 144*a^4*b^6*c^16*d^2*f^4 - 108*a^8*b^2*c^2*d^16*f^4 - 108*a^2*b^8*c^16*d^2*f^4 + 80*a^7*b^3*c^15*d^3*f^4 + 80*a^3*b^7*c^3*d^15*f^4 - 60*a^8*b^2*c^14*d^4*f^4 - 60*a^6*b^4*c^16*d^2*f^4 - 60*a^4*b^6*c^2*d^16*f^4 - 60*a^2*b^8*c^4*d^14*f^4 - 80*b^

10*c^12*d^6*f^4 - 60*b^10*c^14*d^4*f^4 - 60*b^10*c^10*d^8*f^4 - 24*b^10*c^1
6*d^2*f^4 - 24*b^10*c^8*d^10*f^4 - 4*b^10*c^6*d^12*f^4 - 80*a^10*c^6*d^12*f
^4 - 60*a^10*c^8*d^10*f^4 - 60*a^10*c^4*d^14*f^4 - 24*a^10*c^10*d^8*f^4 - 2
4*a^10*c^2*d^16*f^4 - 4*a^10*c^12*d^6*f^4 - 8*a^8*b^2*d^18*f^4 - 4*a^6*b^4*
d^18*f^4 - 8*a^2*b^8*c^18*f^4 - 4*a^4*b^6*c^18*f^4 - 4*b^10*c^18*f^4 - 4*a^
10*d^18*f^4 - 12*A*C*a^7*b*c*d^11*f^2 - 12*A*C*a*b^7*c^11*d*f^2 - 912*B*C*a
^4*b^4*c^5*d^7*f^2 + 792*B*C*a^5*b^3*c^4*d^8*f^2 - 792*B*C*a^3*b^5*c^8*d^4*
f^2 + 720*B*C*a^4*b^4*c^7*d^5*f^2 - 480*B*C*a^6*b^2*c^5*d^7*f^2 - 408*B*C*a
^2*b^6*c^5*d^7*f^2 + 384*B*C*a^2*b^6*c^7*d^5*f^2 - 336*B*C*a^5*b^3*c^8*d^4*
f^2 + 324*B*C*a^3*b^5*c^4*d^8*f^2 + 312*B*C*a^6*b^2*c^7*d^5*f^2 - 248*B*C*a
^6*b^2*c^3*d^9*f^2 + 216*B*C*a^2*b^6*c^9*d^3*f^2 - 196*B*C*a^4*b^4*c^3*d^9*
f^2 + 132*B*C*a^4*b^4*c^9*d^3*f^2 + 80*B*C*a^3*b^5*c^6*d^6*f^2 - 64*B*C*a^5
*b^3*c^6*d^6*f^2 - 36*B*C*a^3*b^5*c^2*d^10*f^2 - 28*B*C*a^2*b^6*c^3*d^9*f^2
 + 12*B*C*a^5*b^3*c^10*d^2*f^2 - 12*B*C*a^5*b^3*c^2*d^10*f^2 - 12*B*C*a^3*b
^5*c^10*d^2*f^2 - 4*B*C*a^6*b^2*c^9*d^3*f^2 - 1468*A*C*a^4*b^4*c^6*d^6*f^2
+ 996*A*C*a^3*b^5*c^7*d^5*f^2 + 900*A*C*a^5*b^3*c^5*d^7*f^2 - 676*A*C*a^6*b
^2*c^6*d^6*f^2 - 660*A*C*a^2*b^6*c^6*d^6*f^2 + 636*A*C*a^3*b^5*c^5*d^7*f^2
+ 540*A*C*a^5*b^3*c^7*d^5*f^2 - 236*A*C*a^5*b^3*c^3*d^9*f^2 - 204*A*C*a^3*b
^5*c^9*d^3*f^2 + 156*A*C*a^2*b^6*c^10*d^2*f^2 + 132*A*C*a^6*b^2*c^2*d^10*f^
2 - 72*A*C*a^6*b^2*c^4*d^8*f^2 - 72*A*C*a^5*b^3*c^9*d^3*f^2 + 66*A*C*a^2*b^
6*c^4*d^8*f^2 + 54*A*C*a^4*b^4*c^10*d^2*f^2 + 54*A*C*a^4*b^4*c^2*d^10*f^2 -
 48*A*C*a^4*b^4*c^4*d^8*f^2 - 48*A*C*a^2*b^6*c^8*d^4*f^2 + 42*A*C*a^6*b^2*c
^8*d^4*f^2 - 40*A*C*a^3*b^5*c^3*d^9*f^2 - 36*A*C*a^4*b^4*c^8*d^4*f^2 + 24*A
*C*a^2*b^6*c^2*d^10*f^2 + 960*A*B*a^4*b^4*c^5*d^7*f^2 - 864*A*B*a^5*b^3*c^4
*d^8*f^2 + 756*A*B*a^3*b^5*c^8*d^4*f^2 - 744*A*B*a^4*b^4*c^7*d^5*f^2 - 528*
A*B*a^3*b^5*c^4*d^8*f^2 + 504*A*B*a^6*b^2*c^5*d^7*f^2 - 432*A*B*a^2*b^6*c^7
*d^5*f^2 + 432*A*B*a^2*b^6*c^5*d^7*f^2 + 348*A*B*a^5*b^3*c^8*d^4*f^2 - 312*
A*B*a^6*b^2*c^7*d^5*f^2 - 284*A*B*a^2*b^6*c^9*d^3*f^2 + 280*A*B*a^6*b^2*c^3
*d^9*f^2 + 264*A*B*a^4*b^4*c^3*d^9*f^2 - 240*A*B*a^3*b^5*c^6*d^6*f^2 - 172*
A*B*a^4*b^4*c^9*d^3*f^2 + 68*A*B*a^2*b^6*c^3*d^9*f^2 - 60*A*B*a^3*b^5*c^2*d
^10*f^2 + 24*A*B*a^5*b^3*c^6*d^6*f^2 - 24*A*B*a^5*b^3*c^2*d^10*f^2 + 12*A*B
*a^3*b^5*c^10*d^2*f^2 + 360*B*C*a^7*b*c^4*d^8*f^2 - 336*B*C*a*b^7*c^8*d^4*f
^2 + 168*B*C*a*b^7*c^6*d^6*f^2 - 136*B*C*a^7*b*c^6*d^6*f^2 + 36*B*C*a^6*b^2
*c*d^11*f^2 - 36*B*C*a^2*b^6*c^11*d*f^2 - 24*B*C*a^7*b*c^2*d^10*f^2 + 24*B*
C*a*b^7*c^10*d^2*f^2 - 12*B*C*a^4*b^4*c^11*d*f^2 + 12*B*C*a^4*b^4*c*d^11*f^
2 + 12*B*C*a*b^7*c^4*d^8*f^2 + 444*A*C*a*b^7*c^7*d^5*f^2 + 348*A*C*a^7*b*c^
5*d^7*f^2 - 164*A*C*a^7*b*c^3*d^9*f^2 - 132*A*C*a*b^7*c^9*d^3*f^2 + 84*A*C*
a*b^7*c^5*d^7*f^2 + 32*A*C*a*b^7*c^3*d^9*f^2 - 12*A*C*a^7*b*c^7*d^5*f^2 - 1
2*A*C*a^5*b^3*c*d^11*f^2 - 12*A*C*a^3*b^5*c^11*d*f^2 - 360*A*B*a^7*b*c^4*d^
8*f^2 + 288*A*B*a*b^7*c^8*d^4*f^2 - 288*A*B*a*b^7*c^6*d^6*f^2 - 144*A*B*a*b
^7*c^4*d^8*f^2 + 136*A*B*a^7*b*c^6*d^6*f^2 - 60*A*B*a*b^7*c^2*d^10*f^2 - 36
*A*B*a*b^7*c^10*d^2*f^2 + 24*A*B*a^7*b*c^2*d^10*f^2 - 24*A*B*a^6*b^2*c*d^11
*f^2 + 12*A*B*a^4*b^4*c*d^11*f^2 + 12*A*B*a^2*b^6*c^11*d*f^2 + 12*A*B*a^2*b
^6*c*d^11*f^2 + 80*B*C*b^8*c^9*d^3*f^2 - 24*B*C*b^8*c^7*d^5*f^2 - 90*A*C*b^
8*c^8*d^4*f^2 - 80*B*C*a^8*c^3*d^9*f^2 + 54*A*C*b^8*c^10*d^2*f^2 - 30*A*C*b

^8*c^6*d^6*f^2 + 24*B*C*a^8*c^5*d^7*f^2 - 12*A*C*b^8*c^4*d^8*f^2 - 112*A*B*
b^8*c^9*d^3*f^2 - 66*A*C*a^8*c^4*d^8*f^2 + 54*A*C*a^8*c^2*d^10*f^2 - 8*B*C*
a^5*b^3*d^12*f^2 - 8*B*C*a^3*b^5*d^12*f^2 + 4*A*B*b^8*c^3*d^9*f^2 + 2*A*C*a
^8*c^6*d^6*f^2 + 80*A*B*a^8*c^3*d^9*f^2 - 24*A*B*a^8*c^5*d^7*f^2 + 8*A*C*a^
2*b^6*d^12*f^2 - 4*B*C*a^3*b^5*c^12*f^2 + 4*A*C*a^4*b^4*d^12*f^2 - 2*A*C*a^
6*b^2*d^12*f^2 + 6*A*C*a^2*b^6*c^12*f^2 + 4*A*B*a^5*b^3*d^12*f^2 - 4*A*B*a^
3*b^5*d^12*f^2 + 726*C^2*a^4*b^4*c^6*d^6*f^2 - 402*C^2*a^5*b^3*c^5*d^7*f^2
- 402*C^2*a^3*b^5*c^7*d^5*f^2 + 322*C^2*a^6*b^2*c^6*d^6*f^2 + 322*C^2*a^2*b
^6*c^6*d^6*f^2 - 222*C^2*a^5*b^3*c^7*d^5*f^2 - 222*C^2*a^3*b^5*c^5*d^7*f^2
+ 134*C^2*a^5*b^3*c^3*d^9*f^2 + 134*C^2*a^3*b^5*c^9*d^3*f^2 - 66*C^2*a^6*b^
2*c^2*d^10*f^2 - 66*C^2*a^2*b^6*c^10*d^2*f^2 + 52*C^2*a^5*b^3*c^9*d^3*f^2 +
52*C^2*a^3*b^5*c^3*d^9*f^2 - 27*C^2*a^6*b^2*c^8*d^4*f^2 - 27*C^2*a^2*b^6*c
^4*d^8*f^2 + 24*C^2*a^6*b^2*c^4*d^8*f^2 + 24*C^2*a^4*b^4*c^8*d^4*f^2 + 24*C
^2*a^4*b^4*c^4*d^8*f^2 + 24*C^2*a^2*b^6*c^8*d^4*f^2 - 15*C^2*a^4*b^4*c^10*d
^2*f^2 - 15*C^2*a^4*b^4*c^2*d^10*f^2 - 570*B^2*a^4*b^4*c^6*d^6*f^2 + 366*B^
2*a^3*b^5*c^7*d^5*f^2 + 318*B^2*a^5*b^3*c^5*d^7*f^2 - 262*B^2*a^6*b^2*c^6*d
^6*f^2 - 222*B^2*a^2*b^6*c^6*d^6*f^2 - 210*B^2*a^5*b^3*c^3*d^9*f^2 + 186*B^
2*a^5*b^3*c^7*d^5*f^2 + 162*B^2*a^3*b^5*c^5*d^7*f^2 - 142*B^2*a^3*b^5*c^9*d
^3*f^2 + 132*B^2*a^4*b^4*c^4*d^8*f^2 + 117*B^2*a^2*b^6*c^4*d^8*f^2 + 102*B^
2*a^6*b^2*c^2*d^10*f^2 - 96*B^2*a^3*b^5*c^3*d^9*f^2 + 90*B^2*a^2*b^6*c^10*d
^2*f^2 + 81*B^2*a^4*b^4*c^2*d^10*f^2 - 56*B^2*a^5*b^3*c^9*d^3*f^2 + 48*B^2*
a^6*b^2*c^4*d^8*f^2 + 48*B^2*a^4*b^4*c^8*d^4*f^2 + 45*B^2*a^6*b^2*c^8*d^4*f
^2 + 36*B^2*a^2*b^6*c^8*d^4*f^2 + 36*B^2*a^2*b^6*c^2*d^10*f^2 + 33*B^2*a^4*
b^4*c^10*d^2*f^2 + 822*A^2*a^4*b^4*c^6*d^6*f^2 - 594*A^2*a^3*b^5*c^7*d^5*f^
2 - 498*A^2*a^5*b^3*c^5*d^7*f^2 + 498*A^2*a^2*b^6*c^6*d^6*f^2 - 414*A^2*a^3
*b^5*c^5*d^7*f^2 + 354*A^2*a^6*b^2*c^6*d^6*f^2 - 318*A^2*a^5*b^3*c^7*d^5*f^
2 + 144*A^2*a^2*b^6*c^8*d^4*f^2 + 102*A^2*a^5*b^3*c^3*d^9*f^2 + 84*A^2*a^4*
b^4*c^4*d^8*f^2 + 81*A^2*a^2*b^6*c^4*d^8*f^2 + 72*A^2*a^4*b^4*c^8*d^4*f^2 +
70*A^2*a^3*b^5*c^9*d^3*f^2 - 66*A^2*a^6*b^2*c^2*d^10*f^2 + 48*A^2*a^6*b^2*
c^4*d^8*f^2 - 42*A^2*a^2*b^6*c^10*d^2*f^2 + 24*A^2*a^2*b^6*c^2*d^10*f^2 + 2
0*A^2*a^5*b^3*c^9*d^3*f^2 - 15*A^2*a^6*b^2*c^8*d^4*f^2 - 15*A^2*a^4*b^4*c^1
0*d^2*f^2 - 15*A^2*a^4*b^4*c^2*d^10*f^2 - 12*A^2*a^3*b^5*c^3*d^9*f^2 - 24*B
*C*b^8*c^11*d*f^2 + 24*B*C*a^8*c*d^11*f^2 + 12*A*B*b^8*c^11*d*f^2 - 8*B*C*a
^7*b*d^12*f^2 - 24*A*B*a^8*c*d^11*f^2 + 4*B*C*a*b^7*c^12*f^2 + 8*A*B*a^7*b*
d^12*f^2 - 8*A*B*a*b^7*d^12*f^2 - 8*A*B*a*b^7*c^12*f^2 - 174*C^2*a^7*b*c^5*
d^7*f^2 - 174*C^2*a*b^7*c^7*d^5*f^2 + 82*C^2*a^7*b*c^3*d^9*f^2 + 82*C^2*a*b
^7*c^9*d^3*f^2 + 6*C^2*a^7*b*c^7*d^5*f^2 + 6*C^2*a^5*b^3*c*d^11*f^2 + 6*C^2
*a^3*b^5*c^11*d*f^2 + 6*C^2*a*b^7*c^5*d^7*f^2 + 162*B^2*a*b^7*c^7*d^5*f^2 +
138*B^2*a^7*b*c^5*d^7*f^2 - 118*B^2*a^7*b*c^3*d^9*f^2 - 86*B^2*a*b^7*c^9*d
^3*f^2 - 30*B^2*a^5*b^3*c*d^11*f^2 - 18*B^2*a^7*b*c^7*d^5*f^2 - 18*B^2*a*b^
7*c^5*d^7*f^2 - 12*B^2*a^3*b^5*c*d^11*f^2 - 6*B^2*a^3*b^5*c^11*d*f^2 - 4*B^
2*a*b^7*c^3*d^9*f^2 - 270*A^2*a*b^7*c^7*d^5*f^2 - 174*A^2*a^7*b*c^5*d^7*f^2
- 90*A^2*a*b^7*c^5*d^7*f^2 + 82*A^2*a^7*b*c^3*d^9*f^2 + 50*A^2*a*b^7*c^9*d
^3*f^2 - 32*A^2*a*b^7*c^3*d^9*f^2 + 6*A^2*a^7*b*c^7*d^5*f^2 + 6*A^2*a^5*b^3
*c*d^11*f^2 + 6*A^2*a^3*b^5*c^11*d*f^2 + 6*C^2*a^7*b*c*d^11*f^2 + 6*C^2*a*b

^7*c^11*d*f^2 - 18*B^2*a^7*b*c*d^11*f^2 - 6*B^2*a*b^7*c^11*d*f^2 + 6*A^2*a^
7*b*c*d^11*f^2 + 6*A^2*a*b^7*c^11*d*f^2 - 6*A*C*a^8*d^12*f^2 - 2*A*C*b^8*c^
12*f^2 + 33*C^2*b^8*c^8*d^4*f^2 - 27*C^2*b^8*c^10*d^2*f^2 - C^2*b^8*c^6*d^6
*f^2 + 33*C^2*a^8*c^4*d^8*f^2 + 33*B^2*b^8*c^10*d^2*f^2 - 27*C^2*a^8*c^2*d^
10*f^2 - 27*B^2*b^8*c^8*d^4*f^2 + 3*B^2*b^8*c^6*d^6*f^2 - C^2*a^8*c^6*d^6*f
^2 + 117*A^2*b^8*c^8*d^4*f^2 + 111*A^2*b^8*c^6*d^6*f^2 + 72*A^2*b^8*c^4*d^8
*f^2 + 33*B^2*a^8*c^2*d^10*f^2 - 27*B^2*a^8*c^4*d^8*f^2 + 24*A^2*b^8*c^2*d^
10*f^2 + 4*C^2*a^4*b^4*d^12*f^2 + 3*C^2*a^6*b^2*d^12*f^2 + 3*B^2*a^8*c^6*d^
6*f^2 - 3*A^2*b^8*c^10*d^2*f^2 + 33*A^2*a^8*c^4*d^8*f^2 - 27*A^2*a^8*c^2*d^
10*f^2 + 4*C^2*a^4*b^4*c^12*f^2 + 4*B^2*a^4*b^4*d^12*f^2 + 4*B^2*a^2*b^6*d^
12*f^2 + 3*C^2*a^2*b^6*c^12*f^2 + 3*B^2*a^6*b^2*d^12*f^2 - A^2*a^8*c^6*d^6*
f^2 - 4*A^2*a^4*b^4*d^12*f^2 + 3*B^2*a^2*b^6*c^12*f^2 - A^2*a^6*b^2*d^12*f^
2 - A^2*a^2*b^6*c^12*f^2 + 3*C^2*b^8*c^12*f^2 + 3*C^2*a^8*d^12*f^2 + 4*A^2*
b^8*d^12*f^2 - B^2*b^8*c^12*f^2 - B^2*a^8*d^12*f^2 + 3*A^2*b^8*c^12*f^2 + 3
*A^2*a^8*d^12*f^2 - 24*A*B*C*a*b^6*c*d^8*f + 342*A*B*C*a^2*b^5*c^4*d^5*f -
186*A*B*C*a^3*b^4*c^5*d^4*f - 66*A*B*C*a^4*b^3*c^2*d^7*f + 48*A*B*C*a^2*b^5
*c^2*d^7*f + 42*A*B*C*a^2*b^5*c^6*d^3*f + 26*A*B*C*a^5*b^2*c^3*d^6*f + 24*A
*B*C*a^4*b^3*c^6*d^3*f - 18*A*B*C*a^4*b^3*c^4*d^5*f - 18*A*B*C*a^3*b^4*c^7*
d^2*f - 8*A*B*C*a^3*b^4*c^3*d^6*f + 6*A*B*C*a^5*b^2*c^5*d^4*f - 128*A*B*C*a
*b^6*c^3*d^6*f + 126*A*B*C*a*b^6*c^7*d^2*f + 72*A*B*C*a^3*b^4*c*d^8*f - 36*
A*B*C*a^5*b^2*c*d^8*f - 36*A*B*C*a^2*b^5*c^8*d*f + 30*A*B*C*a^6*b*c^2*d^7*f
 - 12*A*B*C*a^6*b*c^4*d^5*f - 12*A*B*C*a*b^6*c^5*d^4*f - 21*B^2*C*a*b^6*c^8
*d*f - 3*B^2*C*a^6*b*c*d^8*f + 21*A^2*C*a*b^6*c^8*d*f - 21*A*C^2*a*b^6*c^8*
d*f - 9*A^2*C*a^6*b*c*d^8*f + 9*A*C^2*a^6*b*c*d^8*f + 36*A^2*B*a*b^6*c*d^8*
f + 21*A*B^2*a*b^6*c^8*d*f + 3*A*B^2*a^6*b*c*d^8*f - 78*A*B*C*b^7*c^6*d^3*f
 + 24*A*B*C*b^7*c^4*d^5*f + 2*A*B*C*a^7*c^3*d^6*f + 16*A*B*C*a^4*b^3*d^9*f
- 16*A*B*C*a^2*b^5*d^9*f - 237*B^2*C*a^3*b^4*c^4*d^5*f + 165*B*C^2*a^3*b^4*
c^5*d^4*f + 92*B^2*C*a^2*b^5*c^3*d^6*f - 81*B^2*C*a^2*b^5*c^7*d^2*f + 77*B^
2*C*a^4*b^3*c^3*d^6*f - 75*B*C^2*a^2*b^5*c^4*d^5*f + 69*B^2*C*a^4*b^3*c^5*d
^4*f + 69*B*C^2*a^4*b^3*c^4*d^5*f - 68*B*C^2*a^3*b^4*c^3*d^6*f - 63*B^2*C*a
^5*b^2*c^4*d^5*f - 61*B*C^2*a^2*b^5*c^6*d^3*f + 57*B*C^2*a^4*b^3*c^2*d^7*f
- 53*B*C^2*a^5*b^2*c^3*d^6*f - 44*B*C^2*a^4*b^3*c^6*d^3*f - 36*B^2*C*a^3*b^
4*c^2*d^7*f + 35*B^2*C*a^3*b^4*c^6*d^3*f + 33*B^2*C*a^5*b^2*c^2*d^7*f - 33*
B^2*C*a^2*b^5*c^5*d^4*f + 33*B*C^2*a^3*b^4*c^7*d^2*f - 12*B^2*C*a^4*b^3*c^7
*d^2*f + 9*B*C^2*a^5*b^2*c^5*d^4*f + 4*B^2*C*a^5*b^2*c^6*d^3*f + 225*A^2*C*
a^2*b^5*c^5*d^4*f - 105*A*C^2*a^2*b^5*c^5*d^4*f - 99*A^2*C*a^3*b^4*c^4*d^5*
f - 81*A^2*C*a^5*b^2*c^4*d^5*f + 67*A^2*C*a^4*b^3*c^3*d^6*f - 59*A*C^2*a^4*
b^3*c^3*d^6*f + 57*A*C^2*a^5*b^2*c^2*d^7*f - 57*A*C^2*a^2*b^5*c^7*d^2*f + 5
1*A^2*C*a^4*b^3*c^5*d^4*f + 48*A^2*C*a^3*b^4*c^2*d^7*f + 45*A*C^2*a^5*b^2*c
^4*d^5*f - 35*A^2*C*a^3*b^4*c^6*d^3*f - 33*A^2*C*a^5*b^2*c^2*d^7*f + 33*A^2
*C*a^2*b^5*c^7*d^2*f + 33*A*C^2*a^4*b^3*c^5*d^4*f + 27*A*C^2*a^3*b^4*c^6*d^
3*f - 24*A*C^2*a^3*b^4*c^2*d^7*f + 24*A*C^2*a^2*b^5*c^3*d^6*f - 21*A*C^2*a^
3*b^4*c^4*d^5*f - 16*A^2*C*a^2*b^5*c^3*d^6*f - 243*A^2*B*a^2*b^5*c^4*d^5*f
- 156*A*B^2*a^2*b^5*c^3*d^6*f + 141*A*B^2*a^3*b^4*c^4*d^5*f + 108*A^2*B*a^3
*b^4*c^3*d^6*f - 105*A*B^2*a^4*b^3*c^3*d^6*f + 84*A*B^2*a^3*b^4*c^2*d^7*f +

81*A*B^2*a^2*b^5*c^5*d^4*f - 51*A^2*B*a^4*b^3*c^4*d^5*f + 51*A^2*B*a^2*b^5
*c^6*d^3*f - 48*A^2*B*a^2*b^5*c^2*d^7*f + 45*A^2*B*a^3*b^4*c^5*d^4*f + 39*A
*B^2*a^5*b^2*c^4*d^5*f - 35*A*B^2*a^3*b^4*c^6*d^3*f + 33*A*B^2*a^2*b^5*c^7*
d^2*f + 27*A^2*B*a^5*b^2*c^3*d^6*f - 21*A*B^2*a^4*b^3*c^5*d^4*f + 20*A^2*B*
a^4*b^3*c^6*d^3*f - 15*A^2*B*a^5*b^2*c^5*d^4*f - 15*A^2*B*a^3*b^4*c^7*d^2*f
 + 9*A^2*B*a^4*b^3*c^2*d^7*f + 3*A*B^2*a^5*b^2*c^2*d^7*f + 18*A*B*C*b^7*c^8
*d*f - 6*A*B*C*a^7*c*d^8*f + 2*A*B*C*a^6*b*d^9*f - 6*A*B*C*a*b^6*c^9*f + 63
*B^2*C*a*b^6*c^6*d^3*f - 48*B^2*C*a^4*b^3*c*d^8*f + 42*B*C^2*a^2*b^5*c^8*d*
f + 42*B*C^2*a*b^6*c^5*d^4*f - 39*B*C^2*a*b^6*c^7*d^2*f + 30*B*C^2*a^5*b^2*
c*d^8*f - 24*B^2*C*a*b^6*c^4*d^5*f - 24*B*C^2*a^3*b^4*c*d^8*f + 17*B^2*C*a^
6*b*c^3*d^6*f - 15*B*C^2*a^6*b*c^2*d^7*f + 12*B^2*C*a^3*b^4*c^8*d*f + 12*B^
2*C*a^2*b^5*c*d^8*f + 6*B*C^2*a^6*b*c^4*d^5*f - 192*A^2*C*a*b^6*c^4*d^5*f -
 99*A^2*C*a*b^6*c^6*d^3*f + 84*A*C^2*a*b^6*c^4*d^5*f + 59*A*C^2*a*b^6*c^6*d
^3*f + 51*A^2*C*a^6*b*c^3*d^6*f - 51*A*C^2*a^6*b*c^3*d^6*f - 36*A^2*C*a^2*b
^5*c*d^8*f - 24*A*C^2*a^4*b^3*c*d^8*f + 24*A*C^2*a^2*b^5*c*d^8*f + 12*A^2*C
*a^4*b^3*c*d^8*f + 12*A*C^2*a^3*b^4*c^8*d*f + 160*A^2*B*a*b^6*c^3*d^6*f - 9
9*A*B^2*a*b^6*c^6*d^3*f - 87*A^2*B*a*b^6*c^7*d^2*f - 72*A*B^2*a*b^6*c^4*d^5
*f - 48*A*B^2*a^2*b^5*c*d^8*f - 36*A^2*B*a^3*b^4*c*d^8*f + 24*A*B^2*a^4*b^3
*c*d^8*f - 17*A*B^2*a^6*b*c^3*d^6*f - 15*A^2*B*a^6*b*c^2*d^7*f + 12*A*B^2*a
*b^6*c^2*d^7*f + 6*A^2*B*a^6*b*c^4*d^5*f + 6*A^2*B*a^5*b^2*c*d^8*f + 6*A^2*
B*a^2*b^5*c^8*d*f - 6*A^2*B*a*b^6*c^5*d^4*f + 3*B^2*C*b^7*c^7*d^2*f - B*C^2
*b^7*c^6*d^3*f + 96*A^2*C*b^7*c^5*d^4*f - 39*A^2*C*b^7*c^7*d^2*f - 36*A*C^2
*b^7*c^5*d^4*f + 32*A^2*C*b^7*c^3*d^6*f + 15*A*C^2*b^7*c^7*d^2*f - 3*B^2*C*
a^7*c^2*d^7*f - B*C^2*a^7*c^3*d^6*f + 111*A^2*B*b^7*c^6*d^3*f - 39*A*B^2*b^
7*c^7*d^2*f + 24*A*B^2*b^7*c^5*d^4*f + 12*B^2*C*a^3*b^4*d^9*f - 12*B*C^2*a^
4*b^3*d^9*f - 9*A^2*C*a^7*c^2*d^7*f + 9*A*C^2*a^7*c^2*d^7*f - 4*A*B^2*b^7*c
^3*d^6*f - 12*A^2*C*a^3*b^4*d^9*f - 8*A*C^2*a^5*b^2*d^9*f + 8*A*C^2*a^3*b^4
*d^9*f + 4*B^2*C*a^2*b^5*c^9*f + 4*A^2*C*a^5*b^2*d^9*f - 4*B*C^2*a^3*b^4*c^
9*f + 3*A*B^2*a^7*c^2*d^7*f - A^2*B*a^7*c^3*d^6*f + 12*A^2*B*a^2*b^5*d^9*f
 - 8*A*B^2*a^3*b^4*d^9*f - 4*A^2*B*a^4*b^3*d^9*f + 4*A*C^2*a^2*b^5*c^9*f - 3
*C^3*a^6*b*c*d^8*f + 3*C^3*a*b^6*c^8*d*f + 3*A^3*a^6*b*c*d^8*f - 3*A^3*a*b^
6*c^8*d*f + 3*B*C^2*b^7*c^8*d*f + 12*A^2*C*b^7*c*d^8*f + 3*B*C^2*a^7*c*d^8*
f - 9*A^2*B*b^7*c^8*d*f - B*C^2*a^6*b*d^9*f + 4*A^2*C*a*b^6*d^9*f + 3*A^2*B
*a^7*c*d^8*f + 3*B*C^2*a*b^6*c^9*f + 8*A*B^2*a*b^6*d^9*f - A^2*B*a^6*b*d^9*
f - A^2*B*a*b^6*c^9*f - 39*C^3*a^4*b^3*c^5*d^4*f + 39*C^3*a^3*b^4*c^4*d^5*f
 - 27*C^3*a^5*b^2*c^2*d^7*f + 27*C^3*a^2*b^5*c^7*d^2*f + 17*C^3*a^4*b^3*c^3
*d^6*f - 17*C^3*a^3*b^4*c^6*d^3*f - 3*C^3*a^5*b^2*c^4*d^5*f + 3*C^3*a^2*b^5
*c^5*d^4*f - 63*B^3*a^3*b^4*c^5*d^4*f + 57*B^3*a^2*b^5*c^4*d^5*f - 51*B^3*a
^4*b^3*c^2*d^7*f + 48*B^3*a^3*b^4*c^3*d^6*f + 31*B^3*a^2*b^5*c^6*d^3*f + 27
*B^3*a^5*b^2*c^3*d^6*f + 16*B^3*a^4*b^3*c^6*d^3*f - 15*B^3*a^5*b^2*c^5*d^4*
f - 12*B^3*a^2*b^5*c^2*d^7*f + 9*B^3*a^4*b^3*c^4*d^5*f - 3*B^3*a^3*b^4*c^7*
d^2*f - 123*A^3*a^2*b^5*c^5*d^4*f + 81*A^3*a^3*b^4*c^4*d^5*f - 45*A^3*a^4*b
^3*c^5*d^4*f + 39*A^3*a^5*b^2*c^4*d^5*f - 25*A^3*a^4*b^3*c^3*d^6*f + 25*A^3
*a^3*b^4*c^6*d^3*f - 24*A^3*a^3*b^4*c^2*d^7*f - 8*A^3*a^2*b^5*c^3*d^6*f + 3
*A^3*a^5*b^2*c^2*d^7*f - 3*A^3*a^2*b^5*c^7*d^2*f + 17*C^3*a^6*b*c^3*d^6*f -

17*C^3*a*b^6*c^6*d^3*f + 12*C^3*a^4*b^3*c*d^8*f - 12*C^3*a^3*b^4*c^8*d*f +
24*B^3*a^3*b^4*c*d^8*f + 21*B^3*a*b^6*c^7*d^2*f - 18*B^3*a*b^6*c^5*d^4*f -
15*B^3*a^6*b*c^2*d^7*f + 6*B^3*a^6*b*c^4*d^5*f + 6*B^3*a^5*b^2*c*d^8*f - 6
*B^3*a^2*b^5*c^8*d*f + 4*B^3*a*b^6*c^3*d^6*f + 108*A^3*a*b^6*c^4*d^5*f + 57
*A^3*a*b^6*c^6*d^3*f - 17*A^3*a^6*b*c^3*d^6*f + 12*A^3*a^2*b^5*c*d^8*f + 3*
C^3*b^7*c^7*d^2*f - 3*C^3*a^7*c^2*d^7*f - B^3*b^7*c^6*d^3*f - 60*A^3*b^7*c^
5*d^4*f - 32*A^3*b^7*c^3*d^6*f + 21*A^3*b^7*c^7*d^2*f + 4*C^3*a^5*b^2*d^9*f
 - B^3*a^7*c^3*d^6*f - 4*C^3*a^2*b^5*c^9*f - 4*B^3*a^2*b^5*d^9*f + 3*A^3*a^
7*c^2*d^7*f + 4*A^3*a^3*b^4*d^9*f + 3*B^3*b^7*c^8*d*f - 12*A^3*b^7*c*d^8*f
+ 3*B^3*a^7*c*d^8*f - B^3*a^6*b*d^9*f - 4*A^3*a*b^6*d^9*f - B^3*a*b^6*c^9*f
 - B^2*C*b^7*c^9*f - 4*A^2*B*b^7*d^9*f + 3*A^2*C*a^7*d^9*f - 3*A*C^2*a^7*d^
9*f - A*C^2*b^7*c^9*f - A*B^2*a^7*d^9*f - C^3*b^7*c^9*f - A^3*a^7*d^9*f + B
^2*C*a^7*d^9*f + A^2*C*b^7*c^9*f + A*B^2*b^7*c^9*f + C^3*a^7*d^9*f + A^3*b^
7*c^9*f - 6*A*B^2*C*a*b^5*c^5*d - 21*A^2*B*C*a^2*b^4*c^3*d^3 + 21*A*B*C^2*a
^2*b^4*c^3*d^3 + 12*A*B^2*C*a^2*b^4*c^4*d^2 - 12*A*B^2*C*a^2*b^4*c^2*d^4 -
10*A*B^2*C*a^3*b^3*c^3*d^3 - 6*A*B*C^2*a^3*b^3*c^4*d^2 + 3*A^2*B*C*a^3*b^3*
c^4*d^2 + 3*A^2*B*C*a^3*b^3*c^2*d^4 + 3*A*B^2*C*a^4*b^2*c^2*d^4 + 3*A*B*C^2
*a^3*b^3*c^2*d^4 + 2*A*B*C^2*a^4*b^2*c^3*d^3 - A^2*B*C*a^4*b^2*c^3*d^3 + 18
*A^2*B*C*a*b^5*c^2*d^4 + 10*A*B^2*C*a*b^5*c^3*d^3 + 9*A^2*B*C*a*b^5*c^4*d^2
 - 9*A*B*C^2*a*b^5*c^4*d^2 - 9*A*B*C^2*a*b^5*c^2*d^4 - 6*A^2*B*C*a^2*b^4*c*
d^5 + 6*A*B^2*C*a^3*b^3*c*d^5 - 6*A*B*C^2*a^4*b^2*c*d^5 + 6*A*B*C^2*a^2*b^4
*c^5*d + 3*A^2*B*C*a^4*b^2*c*d^5 - 3*A^2*B*C*a^2*b^4*c^5*d + 3*A*B*C^2*a^2*
b^4*c*d^5 + 3*B^3*C*a^4*b^2*c*d^5 - 3*B^3*C*a^2*b^4*c^5*d + 3*B^3*C*a*b^5*c
^4*d^2 + 3*B^2*C^2*a*b^5*c^5*d + 3*B*C^3*a^4*b^2*c*d^5 - 3*B*C^3*a^2*b^4*c^
5*d + 3*B*C^3*a*b^5*c^4*d^2 + 24*A^3*C*a*b^5*c^3*d^3 + 8*A*C^3*a*b^5*c^3*d^
3 - 9*A^3*B*a*b^5*c^2*d^4 - 9*A*B^3*a*b^5*c^2*d^4 + 3*A^3*B*a^2*b^4*c*d^5 -
 3*A^3*B*a*b^5*c^4*d^2 + 3*A^2*B^2*a*b^5*c^5*d + 3*A*B^3*a^2*b^4*c*d^5 - 3*
A*B^3*a*b^5*c^4*d^2 - 3*A*B^2*C*b^6*c^4*d^2 - 2*A^2*B*C*b^6*c^3*d^3 + 5*A*B
*C^2*a^3*b^3*d^6 - 4*A^2*B*C*a^3*b^3*d^6 - A*B^2*C*a^4*b^2*d^6 + 9*B^2*C^2*
a^3*b^3*c^3*d^3 - 6*B^2*C^2*a^2*b^4*c^4*d^2 + 6*B^2*C^2*a^2*b^4*c^2*d^4 - 3
*B^2*C^2*a^4*b^2*c^2*d^4 + 24*A^2*C^2*a^3*b^3*c^3*d^3 - 15*A^2*C^2*a^2*b^4*
c^4*d^2 - 9*A^2*C^2*a^4*b^2*c^2*d^4 + 3*A^2*C^2*a^2*b^4*c^2*d^4 + 9*A^2*B^2
*a^2*b^4*c^2*d^4 - 3*A^2*B^2*a^2*b^4*c^4*d^2 + 6*A^2*B*C*b^6*c^5*d - 3*A*B*
C^2*b^6*c^5*d + 4*A^2*B*C*a*b^5*d^6 - 2*A*B*C^2*a*b^5*d^6 + 2*A*B*C^2*a*b^5
*c^6 - A^2*B*C*a*b^5*c^6 - 7*B^3*C*a^2*b^4*c^3*d^3 - 7*B*C^3*a^2*b^4*c^3*d^
3 + 3*B^3*C*a^3*b^3*c^4*d^2 - 3*B^3*C*a^3*b^3*c^2*d^4 - 3*B^2*C^2*a^3*b^3*c
*d^5 + 3*B*C^3*a^3*b^3*c^4*d^2 - 3*B*C^3*a^3*b^3*c^2*d^4 - B^3*C*a^4*b^2*c^
3*d^3 - B^2*C^2*a*b^5*c^3*d^3 - B*C^3*a^4*b^2*c^3*d^3 - 24*A^2*C^2*a*b^5*c^
3*d^3 - 24*A*C^3*a^3*b^3*c^3*d^3 + 12*A*C^3*a^2*b^4*c^4*d^2 + 9*A*C^3*a^4*b
^2*c^2*d^4 - 8*A^3*C*a^3*b^3*c^3*d^3 + 6*A^3*C*a^2*b^4*c^4*d^2 - 6*A^3*C*a^
2*b^4*c^2*d^4 + 3*A^3*C*a^4*b^2*c^2*d^4 - 9*A^2*B^2*a*b^5*c^3*d^3 + 7*A^3*B
*a^2*b^4*c^3*d^3 + 7*A*B^3*a^2*b^4*c^3*d^3 - 3*A^3*B*a^3*b^3*c^2*d^4 - 3*A^
2*B^2*a^3*b^3*c*d^5 - 3*A*B^3*a^3*b^3*c^2*d^4 + 12*A^2*C^2*b^6*c^4*d^2 + 3*
A^2*C^2*b^6*c^2*d^4 + 6*A^2*B^2*b^6*c^4*d^2 + 3*A^2*B^2*b^6*c^2*d^4 - 5*A^2
*C^2*a^2*b^4*d^6 + 3*A^2*C^2*a^4*b^2*d^6 + A*B*C^2*b^6*c^3*d^3 - 3*B^4*a^3*

```
b^3*c*d^5 - B^4*a*b^5*c^3*d^3 + A^2*B^2*a^3*b^3*c^3*d^3 - 8*A^4*a*b^5*c^3*d
^3 - 15*A^3*C*b^6*c^4*d^2 - 6*A^3*C*b^6*c^2*d^4 - 3*A*C^3*b^6*c^4*d^2 - 2*B
^3*C*a^3*b^3*d^6 - 2*B*C^3*a^3*b^3*d^6 + 4*A^3*C*a^2*b^4*d^6 - 3*A*C^3*a^4*
b^2*d^6 + 2*A*C^3*a^2*b^4*d^6 - A^3*C*a^4*b^2*d^6 - 2*A*C^3*a^2*b^4*c^6 + 3
*B^4*a*b^5*c^5*d - 3*A^3*B*b^6*c^5*d - 3*A*B^3*b^6*c^5*d - B^3*C*a*b^5*c^6
- B*C^3*a*b^5*c^6 - 2*A^3*B*a*b^5*d^6 - 2*A*B^3*a*b^5*d^6 + 8*C^4*a^3*b^3*c
^3*d^3 - 3*C^4*a^4*b^2*c^2*d^4 - 3*C^4*a^2*b^4*c^4*d^2 + 6*B^4*a^2*b^4*c^2*
d^4 - 3*B^4*a^2*b^4*c^4*d^2 + 3*A^4*a^2*b^4*c^2*d^4 + B^2*C^2*a^4*b^2*d^6 +
 B^2*C^2*a^2*b^4*d^6 + B^2*C^2*a^2*b^4*c^6 + A^2*C^2*a^2*b^4*c^6 - 2*A^3*C*
b^6*d^6 + A^3*B*b^6*c^3*d^3 + A*B^3*b^6*c^3*d^3 + A^3*B*a^3*b^3*d^6 + A*B^3
*a^3*b^3*d^6 + 6*A^4*b^6*c^4*d^2 + 3*A^4*b^6*c^2*d^4 - A^4*a^2*b^4*d^6 - 2*
A^2*C^2*b^6*c^6 + A*B^2*C*b^6*c^6 + B^4*a^3*b^3*c^3*d^3 + A^3*C*b^6*c^6 + A
*C^3*b^6*c^6 + C^4*a^4*b^2*d^6 + C^4*a^2*b^4*c^6 + B^4*a^2*b^4*d^6 + A^2*C^
2*b^6*d^6 + A^2*B^2*b^6*d^6 + A^4*b^6*d^6, f, k), k, 1, 4) - ((A*a*d^5 - 3*
C*b*c^5 - 3*A*b*c*d^4 + B*a*c*d^4 + 5*B*b*c^4*d + C*a*c^4*d + 5*A*a*c^2*d^3
 - 7*A*b*c^3*d^2 - 3*B*a*c^3*d^2 + B*b*c^2*d^3 - 3*C*a*c^2*d^3 + C*b*c^3*d^
2)/(2*(a^2*d^2 + b^2*c^2 - 2*a*b*c*d)*(c^4 + d^4 + 2*c^2*d^2)) - (tan(e + f
*x)*(A*b*d^5 - B*a*d^5 - 2*A*a*c*d^4 + 2*C*a*c*d^4 + C*b*c^4*d + 3*A*b*c^2*
d^3 + B*a*c^2*d^3 - 2*B*b*c^3*d^2 - C*b*c^2*d^3))/((a^2*d^2 + b^2*c^2 - 2*a
*b*c*d)*(c^4 + d^4 + 2*c^2*d^2)))/(c^2 + d^2*tan(e + f*x)^2 + 2*c*d*tan(e +
 f*x)))/f
```

3.89 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^3}\,dx$

Optimal result 1035
Rubi [A] (verified) 1036
Mathematica [B] (verified) 1039
Maple [A] (verified) 1040
Fricas [B] (verification not implemented) 1041
Sympy [F(-1)] 1041
Maxima [B] (verification not implemented) 1042
Giac [B] (verification not implemented) 1043
Mupad [B] (verification not implemented) 1045

Optimal result

Integrand size = 45, antiderivative size = 861

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^3}\,dx =$$

$$-\frac{(b^2(Ac^3-c^3C+3Bc^2d-3Acd^2+3cCd^2-Bd^3)+a^2(c^3C-3Bc^2d-3cCd^2+Bd^3-A(c^3-3cd^2))}{(a^2+b^2)^2(c^2+d^2)^3}$$

$$+\frac{b^2(4a^3bBd-3a^4Cd+b^4(Bc-3Ad)+2ab^3(Ac-cC+Bd)-a^2b^2(Bc+(5A+C)d))\log(a\cos(e+f}{(a^2+b^2)^2(bc-ad)^4f}$$

$$+\frac{d(b^2(3c^6C-6Bc^5d+c^4(10A-C)d^2-3Bc^3d^3+9Ac^2d^4-Bcd^5+3Ad^6)+a^2d^3((A-C)d(3c^2-d^2)}{(bc-ad)}$$

$$-\frac{d(b^2c(cC-Bd)-2abB(c^2+d^2)+a^2(3c^2C-Bcd+2Cd^2)+A(a^2d^2+b^2(2c^2+3d^2)))}{2(a^2+b^2)(bc-ad)^2(c^2+d^2)f(c+d\tan(e+fx))^2}$$

$$-\frac{Ab^2-a(bB-aC)}{(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))(c+d\tan(e+fx))^2}$$

$$-\frac{d(b^3c(2c^3C-3Bc^2d-Bd^3)+a^2b(3c^4C-3Bc^3d+2c^2Cd^2-Bcd^3+Cd^4)+a^3d^2(2cCd+B(c^2-d^2)}{(a^2+b^2)(bc-ad)^3(c^2+}$$

```
[Out] -(b^2*(A*c^3-3*A*c*d^2+3*B*c^2*d-B*d^3-C*c^3+3*C*c*d^2)+a^2*(c^3*C-3*B*c^2*
d-3*C*c*d^2+B*d^3-A*(c^3-3*c*d^2))+2*a*b*((A-C)*d*(3*c^2-d^2)-B*(c^3-3*c*d^
2)))*x/(a^2+b^2)^2/(c^2+d^2)^3+b^2*(4*a^3*b*B*d-3*a^4*C*d+b^4*(-3*A*d+B*c)+
2*a*b^3*(A*c+B*d-C*c)-a^2*b^2*(B*c+(5*A+C)*d))*ln(a*cos(f*x+e)+b*sin(f*x+e)
)/(a^2+b^2)^2/(-a*d+b*c)^4/f+d*(b^2*(3*c^6*C-6*B*c^5*d+c^4*(10*A-C)*d^2-3*B
*c^3*d^3+9*A*c^2*d^4-B*c*d^5+3*A*d^6)+a^2*d^3*((A-C)*d*(3*c^2-d^2)-B*(c^3-3
*c*d^2))-2*a*b*d^2*(c*(A-C)*d*(5*c^2+d^2)-B*(2*c^4-3*c^2*d^2-d^4)))*ln(c*co
s(f*x+e)+d*sin(f*x+e))/(-a*d+b*c)^4/(c^2+d^2)^3/f-1/2*d*(b^2*c*(-B*d+C*c)-2
*a*b*B*(c^2+d^2)+a^2*(-B*c*d+3*C*c^2+2*C*d^2)+A*(a^2*d^2+b^2*(2*c^2+3*d^2))
)/(a^2+b^2)/(-a*d+b*c)^2/(c^2+d^2)/f/(c+d*tan(f*x+e))^2+(-A*b^2+a*(B*b-C*a)
```

```
)/(a^2+b^2)/(-a*d+b*c)/f/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))^2-d*(b^3*c*(-3*B
*c^2*d-B*d^3+2*C*c^3)+a^2*b*(-3*B*c^3*d-B*c*d^3+3*C*c^4+2*C*c^2*d^2+C*d^4)+
a^3*d^2*(2*C*c*d+B*(c^2-d^2))+a*b^2*(2*c*C*d^3-B*(c^4+c^2*d^2+2*d^4))-A*(2*
a^3*c*d^3+2*a*b^2*c*d^3-2*a^2*b*d^2*(2*c^2+d^2)-b^3*(c^4+6*c^2*d^2+3*d^4)))
/(a^2+b^2)/(-a*d+b*c)^3/(c^2+d^2)^2/f/(c+d*tan(f*x+e))
```

Rubi [A] (verified)

Time = 4.76 (sec) , antiderivative size = 860, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 3, $\frac{\text{number of rules}}{\text{integrand size}} = 0.067$, Rules used = {3730, 3732, 3611}

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^3}\,dx$$
$$= \frac{(-3Cda^4+4bBda^3-b^2(Bc+(5A+C)d)a^2+2b^3(Ac-Cc+Bd)a+b^4(Bc-3Ad))\log(a\cos(e+fx)+}{(a^2+b^2)^2(bc-ad)^4 f}$$
$$-\frac{((Cc^3-3Bdc^2-3Cd^2c+Bd^3-A(c^3-3cd^2))a^2+2b((A-C)d(3c^2-d^2)-B(c^3-3cd^2))a+b^2(Ac^3}{(a^2+b^2)^2(c^2+d^2)^3}$$
$$+\frac{d(a^2((A-C)d(3c^2-d^2)-B(c^3-3cd^2))d^3-2ab(c(A-C)d(5c^2+d^2)-B(2c^4-3d^2c^2-d^4))d^2+b^2}{(bc-ad)^4}$$
$$-\frac{d(d^2(2cCd+B(c^2-d^2))a^3+b(3Cc^4-3Bdc^3+2Cd^2c^2-Bd^3c+Cd^4)a^2+b^2(2cCd^3-B(c^4+d^2c^2+}{(a^2+b^2)(bc-ad)^3(c^2}$$
$$-\frac{d(Ad^2a^2+(3Cc^2-Bdc+2Cd^2)a^2-2bB(c^2+d^2)a+b^2c(cC-Bd)+Ab^2(2c^2+3d^2))}{2(a^2+b^2)(bc-ad)^2(c^2+d^2)f(c+d\tan(e+fx))^2}$$
$$-\frac{Ab^2-a(bB-aC)}{(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))(c+d\tan(e+fx))^2}$$

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^2*(c + d*
Tan[e + f*x])^3),x]
```

```
[Out] -(((b^2*(A*c^3 - c^3*C + 3*B*c^2*d - 3*A*c*d^2 + 3*c*C*d^2 - B*d^3) + a^2*(
c^3*C - 3*B*c^2*d - 3*c*C*d^2 + B*d^3 - A*(c^3 - 3*c*d^2)) + 2*a*b*((A - C)
*d*(3*c^2 - d^2) - B*(c^3 - 3*c*d^2)))*x)/((a^2 + b^2)^2*(c^2 + d^2)^3)) +
(b^2*(4*a^3*b*B*d - 3*a^4*C*d + b^4*(B*c - 3*A*d) + 2*a*b^3*(A*c - c*C + B*
d) - a^2*b^2*(B*c + (5*A + C)*d))*Log[a*Cos[e + f*x] + b*Sin[e + f*x]])/((a
^2 + b^2)^2*(b*c - a*d)^4*f) + (d*(b^2*(3*c^6*C - 6*B*c^5*d + c^4*(10*A - C
)*d^2 - 3*B*c^3*d^3 + 9*A*c^2*d^4 - B*c*d^5 + 3*A*d^6) + a^2*d^3*((A - C)*d
*(3*c^2 - d^2) - B*(c^3 - 3*c*d^2)) - 2*a*b*d^2*(c*(A - C)*d*(5*c^2 + d^2)
- B*(2*c^4 - 3*c^2*d^2 - d^4)))*Log[c*Cos[e + f*x] + d*Sin[e + f*x]])/((b*c
 - a*d)^4*(c^2 + d^2)^3*f) - (d*(a^2*A*d^2 + b^2*c*(c*C - B*d) - 2*a*b*B*(c
^2 + d^2) + A*b^2*(2*c^2 + 3*d^2) + a^2*(3*c^2*C - B*c*d + 2*C*d^2)))/(2*(a
^2 + b^2)*(b*c - a*d)^2*(c^2 + d^2)*f*(c + d*Tan[e + f*x])^2) - (A*b^2 - a*
(b*B - a*C))/((a^2 + b^2)*(b*c - a*d)*f*(a + b*Tan[e + f*x])*(c + d*Tan[e +
```

```
 f*x])^2) - (d*(b^3*c*(2*c^3*C - 3*B*c^2*d - B*d^3) + a^2*b*(3*c^4*C - 3*B*
c^3*d + 2*c^2*C*d^2 - B*c*d^3 + C*d^4) + a^3*d^2*(2*c*C*d + B*(c^2 - d^2))
+ a*b^2*(2*c*C*d^3 - B*(c^4 + c^2*d^2 + 2*d^4)) - A*(2*a^3*c*d^3 + 2*a*b^2*
c*d^3 - 2*a^2*b*d^2*(2*c^2 + d^2) - b^3*(c^4 + 6*c^2*d^2 + 3*d^4))))/((a^2
+ b^2)*(b*c - a*d)^3*(c^2 + d^2)^2*f*(c + d*Tan[e + f*x]))
```

Rule 3611

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c/(b*f))*Log[RemoveContent[a*Cos[e + f*x] + b*Si
n[e + f*x], x]], x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3730

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 - a*(b*B - a*C))*(a + b*Tan[e +
 f*x])^(m + 1)*((c + d*Tan[e + f*x])^(n + 1)/(f*(m + 1)*(b*c - a*d)*(a^2 +
b^2))), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3732

```
Int[((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^
2)/(((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])), x_Symbol] :> Simp[(a*(A*c - c*C + B*d) + b*(B*c - A*d + C*d))*(x/
((a^2 + b^2)*(c^2 + d^2))), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/((b*c - a*d)
*(a^2 + b^2)), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] - Dist
[(c^2*C - B*c*d + A*d^2)/((b*c - a*d)*(c^2 + d^2)), Int[(d - c*Tan[e + f*x]
)/(c + d*Tan[e + f*x]), x], x]) /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] &&
 NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0]
```

Rubi steps

integral

$$= -\frac{Ab^2 - a(bB - aC)}{(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))(c+d\tan(e+fx))^2} - \frac{\int \frac{3Ab^2d - aA(bc-ad) - (bB-aC)(bc+2ad) + (Ab-aB-bC)(bc-ad)\tan(e+fx) + 3(Ab^2 - a(bB-aC))d\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^3}\,dx}{(a^2+b^2)(bc-ad)}$$

$$= -\frac{d(a^2Ad^2 + b^2c(cC - Bd) - 2abB(c^2 + d^2) + Ab^2(2c^2 + 3d^2) + a^2(3c^2C - Bcd + 2Cd^2))}{2(a^2 + b^2)(bc - ad)^2(c^2 + d^2)f(c + d\tan(e + fx))^2}$$
$$- \frac{Ab^2 - a(bB - aC)}{(a^2 + b^2)(bc - ad)f(a + b\tan(e + fx))(c + d\tan(e + fx))^2}$$
$$- \frac{\int \frac{-2(a^3d^2(Ac-cC+Bd)-a^2b(2A+C)d(c^2+d^2)+b^3(Bc-3Ad)(c^2+d^2)+ab^2(Ac-cC+Bd)(c^2+2d^2))-2(bc-ad)^2(bcC-bBd-A(bc+ad)+a(B}{(a+b\tan(e+fx))(c+d\tan(e+}}{2(a^2 + b^2)(bc - ad)^2(c}$$

$$= -\frac{d(a^2Ad^2 + b^2c(cC - Bd) - 2abB(c^2 + d^2) + Ab^2(2c^2 + 3d^2) + a^2(3c^2C - Bcd + 2Cd^2))}{2(a^2 + b^2)(bc - ad)^2(c^2 + d^2)f(c + d\tan(e + fx))^2}$$
$$- \frac{Ab^2 - a(bB - aC)}{(a^2 + b^2)(bc - ad)f(a + b\tan(e + fx))(c + d\tan(e + fx))^2}$$
$$- \frac{d(b^3c(2c^3C - 3Bc^2d - Bd^3) + a^2b(3c^4C - 3Bc^3d + 2c^2Cd^2 - Bcd^3 + Cd^4) + a^3d^2(2cCd + B(c^2 - d^2))}{(a^2 + b^2)(bc - ad)^3(c^2 +}$$
$$- \frac{\int \frac{2((acd-b(c^2+d^2))(a^3d^2(Ac-cC+Bd)-a^2b(2A+C)d(c^2+d^2)+b^3(Bc-3Ad)(c^2+d^2)+ab^2(Ac-cC+Bd)(c^2+2d^2))+ad^2(2b^3c^2(cC-Bd)-}{}}{}$$

$$=$$
$$-\frac{(b^2(Ac^3 - c^3C + 3Bc^2d - 3Acd^2 + 3cCd^2 - Bd^3) + a^2(c^3C - 3Bc^2d - 3cCd^2 + Bd^3 - A(c^3 - 3cd^2)) +}{(a^2 + b^2)^2(c^2 + d^2)^3}$$
$$- \frac{d(a^2Ad^2 + b^2c(cC - Bd) - 2abB(c^2 + d^2) + Ab^2(2c^2 + 3d^2) + a^2(3c^2C - Bcd + 2Cd^2))}{2(a^2 + b^2)(bc - ad)^2(c^2 + d^2)f(c + d\tan(e + fx))^2}$$
$$- \frac{Ab^2 - a(bB - aC)}{(a^2 + b^2)(bc - ad)f(a + b\tan(e + fx))(c + d\tan(e + fx))^2}$$
$$- \frac{d(b^3c(2c^3C - 3Bc^2d - Bd^3) + a^2b(3c^4C - 3Bc^3d + 2c^2Cd^2 - Bcd^3 + Cd^4) + a^3d^2(2cCd + B(c^2 - d^2))}{(a^2 + b^2)(bc - ad)^3(c^2 +}$$
$$+ \frac{(b^2(4a^3bBd - 3a^4Cd + b^4(Bc - 3Ad) + 2ab^3(Ac - cC + Bd) - a^2b^2(Bc + (5A + C)d))) \int \frac{b-a\tan(e+fx)}{a+b\tan(e+fx)}\, d}{(a^2 + b^2)^2(bc - ad)^4}$$
$$+ \frac{(d(b^2(3c^6C - 6Bc^5d + c^4(10A - C)d^2 - 3Bc^3d^3 + 9Ac^2d^4 - Bcd^5 + 3Ad^6) + a^2d^3((A - C)d(3c^2 - d^2) +}{(bc - ad)^4(c^2 + d}$$

$$= -\frac{(b^2(Ac^3 - c^3C + 3Bc^2d - 3Acd^2 + 3cCd^2 - Bd^3) + a^2(c^3C - 3Bc^2d - 3cCd^2 + Bd^3 - A(c^3 - 3cd^2))}{(a^2+b^2)^2(c^2+d^2)^3}$$

$$+ \frac{b^2(4a^3bBd - 3a^4Cd + b^4(Bc - 3Ad) + 2ab^3(Ac - cC + Bd) - a^2b^2(Bc + (5A + C)d))\log(a\cos(e + f}{(a^2+b^2)^2(bc-ad)^4f}$$

$$+ \frac{d(b^2(3c^6C - 6Bc^5d + c^4(10A - C)d^2 - 3Bc^3d^3 + 9Ac^2d^4 - Bcd^5 + 3Ad^6) + a^2d^3((A - C)d(3c^2 - d^2)}{(bc-ad)}$$

$$- \frac{d(a^2Ad^2 + b^2c(cC - Bd) - 2abB(c^2 + d^2) + Ab^2(2c^2 + 3d^2) + a^2(3c^2C - Bcd + 2Cd^2))}{2(a^2+b^2)(bc-ad)^2(c^2+d^2)f(c+d\tan(e+fx))^2}$$

$$- \frac{Ab^2 - a(bB - aC)}{(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))(c+d\tan(e+fx))^2}$$

$$- \frac{d(b^3c(2c^3C - 3Bc^2d - Bd^3) + a^2b(3c^4C - 3Bc^3d + 2c^2Cd^2 - Bcd^3 + Cd^4) + a^3d^2(2cCd + B(c^2 - d^2)}{(a^2+b^2)(bc-ad)^3(c^2+}$$

Mathematica [B] (verified)

Leaf count is larger than twice the leaf count of optimal. 1732 vs. 2(861) = 1722.

Time = 8.63 (sec) , antiderivative size = 1732, normalized size of antiderivative = 2.01

$$\int \frac{A + B\tan(e+fx) + C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^3}\,dx$$

$$= -\frac{Ab^2 - a(bB - aC)}{(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))(c+d\tan(e+fx))^2}$$

$$- \frac{-\frac{-c(-3c(Ab^2-a(bB-aC))d+(Ab-aB-bC)d(bc-ad))+d^2(3Ab^2d-aA(bc-ad)-(bB-aC)(bc+2ad))}{2(-bc+ad)(c^2+d^2)f(c+d\tan(e+fx))^2} - \frac{-\frac{(bc-ad)^3\left(-b^2\left(2aAbc^3-a^2Bc^3\right.\right.}{}}{}}{}$$

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^2*(
c + d*Tan[e + f*x])^3),x]
```

```
[Out] -((A*b^2 - a*(b*B - a*C))/((a^2 + b^2)*(b*c - a*d)*f*(a + b*Tan[e + f*x])*(
c + d*Tan[e + f*x])^2)) - (-1/2*(-(c*(-3*c*(A*b^2 - a*(b*B - a*C))*d + (A*b
 - a*B - b*C)*d*(b*c - a*d))) + d^2*(3*A*b^2*d - a*A*(b*c - a*d) - (b*B - a
*C)*(b*c + 2*a*d)))/((-(b*c) + a*d)*(c^2 + d^2)*f*(c + d*Tan[e + f*x])^2) -
 (-((-(((b*c - a*d)^3*(-(b^2*(2*a*A*b*c^3 - a^2*B*c^3 + b^2*B*c^3 - 2*a*b*c
^3*C + 3*a^2*A*c^2*d - 3*A*b^2*c^2*d + 6*a*b*B*c^2*d - 3*a^2*c^2*C*d + 3*b^
2*c^2*C*d - 6*a*A*b*c*d^2 + 3*a^2*B*c*d^2 - 3*b^2*B*c*d^2 + 6*a*b*c*C*d^2 -
 a^2*A*d^3 + A*b^2*d^3 - 2*a*b*B*d^3 + a^2*C*d^3 - b^2*C*d^3)) + Sqrt[-b^2]
*(-(a^2*A*b*c^3) + A*b^3*c^3 - 2*a*b^2*B*c^3 + a^2*b*c^3*C - b^3*c^3*C + 6*
a*A*b^2*c^2*d - 3*a^2*b*B*c^2*d + 3*b^3*B*c^2*d - 6*a*b^2*c^2*C*d + 3*a^2*A
*b*c*d^2 - 3*A*b^3*c*d^2 + 6*a*b^2*B*c*d^2 - 3*a^2*b*c*C*d^2 + 3*b^3*c*C*d^
2 - 2*a*A*b^2*d^3 + a^2*b*B*d^3 - b^3*B*d^3 + 2*a*b^2*C*d^3))*Log[Sqrt[-b^2
```

```
] - b*Tan[e + f*x]])/(b*(a^2 + b^2)*(c^2 + d^2))) - (2*b^3*(c^2 + d^2)^2*(4
*a^3*b*B*d - 3*a^4*C*d + b^4*(B*c - 3*A*d) + 2*a*b^3*(A*c - c*C + B*d) - a^
2*b^2*(B*c + (5*A + C)*d))*Log[a + b*Tan[e + f*x]])/((a^2 + b^2)*(b*c - a*d
)) + ((b*c - a*d)^3*(b^2*(2*a*A*b*c^3 - a^2*B*c^3 + b^2*B*c^3 - 2*a*b*c^3*C
 + 3*a^2*A*c^2*d - 3*A*b^2*c^2*d + 6*a*b*B*c^2*d - 3*a^2*c^2*C*d + 3*b^2*c^
2*C*d - 6*a*A*b*c*d^2 + 3*a^2*B*c*d^2 - 3*b^2*B*c*d^2 + 6*a*b*c*C*d^2 - a^2
*A*d^3 + A*b^2*d^3 - 2*a*b*B*d^3 + a^2*C*d^3 - b^2*C*d^3) + Sqrt[-b^2]*(-(a
^2*A*b*c^3) + A*b^3*c^3 - 2*a*b^2*B*c^3 + a^2*b*c^3*C - b^3*c^3*C + 6*a*A*b
^2*c^2*d - 3*a^2*b*B*c^2*d + 3*b^3*B*c^2*d - 6*a*b^2*c^2*C*d + 3*a^2*A*b*c*
d^2 - 3*A*b^3*c*d^2 + 6*a*b^2*B*c*d^2 - 3*a^2*b*c*C*d^2 + 3*b^3*c*C*d^2 - 2
*a*A*b^2*d^3 + a^2*b*B*d^3 - b^3*B*d^3 + 2*a*b^2*C*d^3))*Log[Sqrt[-b^2] + b
*Tan[e + f*x]])/(b*(a^2 + b^2)*(c^2 + d^2)) - (2*b*(a^2 + b^2)*d*(b^2*(3*c^
6*C - 6*B*c^5*d + c^4*(10*A - C)*d^2 - 3*B*c^3*d^3 + 9*A*c^2*d^4 - B*c*d^5
+ 3*A*d^6) + a^2*d^3*((A - C)*d*(3*c^2 - d^2) - B*(c^3 - 3*c*d^2)) - 2*a*b*
d^2*(c*(A - C)*d*(5*c^2 + d^2) - B*(2*c^4 - 3*c^2*d^2 - d^4)))*Log[c + d*Ta
n[e + f*x]])/((b*c - a*d)*(c^2 + d^2)))/(b*(-(b*c) + a*d)*(c^2 + d^2)*f)) -
 (d^2*(-2*a*d*(-3*c*(A*b^2 - a*(b*B - a*C))*d + (A*b - a*B - b*C)*d*(b*c -
a*d)) + (2*b*d^2 - 2*c*(-(b*c) + a*d))*(3*A*b^2*d - a*A*(b*c - a*d) - (b*B
- a*C)*(b*c + 2*a*d))) - c*(2*d*(-(b*c) + a*d)*(-3*(A*b^2 - a*(b*B - a*C))*
d^2 - c*(A*b - a*B - b*C)*(b*c - a*d) + d*(3*A*b^2*d - a*A*(b*c - a*d) - (b
*B - a*C)*(b*c + 2*a*d))) - 2*b*c*(-(c*(-3*c*(A*b^2 - a*(b*B - a*C))*d + (A
*b - a*B - b*C)*d*(b*c - a*d))) + d^2*(3*A*b^2*d - a*A*(b*c - a*d) - (b*B -
 a*C)*(b*c + 2*a*d)))))/((-(b*c) + a*d)*(c^2 + d^2)*f*(c + d*Tan[e + f*x]))
)/(2*(-(b*c) + a*d)*(c^2 + d^2)))/((a^2 + b^2)*(b*c - a*d))
```

Maple [A] (verified)

Time = 5.79 (sec) , antiderivative size = 949, normalized size of antiderivative = 1.10

method	result
derivativedivides	$-\frac{b^2\left(5A\,a^2b^2d-2Aa\,b^3c+3A\,b^4d-4a^3bBd+B\,a^2b^2c-2Ba\,b^3d-B\,b^4c+3a^4Cd+C\,a^2b^2d+2Ca\,b^3c\right)\ln(a+b\tan(fx+e))}{(ad-bc)^4\left(a^2+b^2\right)^2}+\frac{\left(A\,b^2-\dots\right.}{(ad-bc)^3\left(a^2+\dots\right.}$ …
default	$-\frac{b^2\left(5A\,a^2b^2d-2Aa\,b^3c+3A\,b^4d-4a^3bBd+B\,a^2b^2c-2Ba\,b^3d-B\,b^4c+3a^4Cd+C\,a^2b^2d+2Ca\,b^3c\right)\ln(a+b\tan(fx+e))}{(ad-bc)^4\left(a^2+b^2\right)^2}+\frac{\left(A\,b^2-\dots\right.}{(ad-bc)^3\left(a^2+\dots\right.}$ …
norman	Expression too large to display
risch	Expression too large to display
parallelrisch	Expression too large to display

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e))^3,x
,method=_RETURNVERBOSE)
```

```
[Out] 1/f*(-b^2*(5*A*a^2*b^2*d-2*A*a*b^3*c+3*A*b^4*d-4*B*a^3*b*d+B*a^2*b^2*c-2*B*
a*b^3*d-B*b^4*c+3*C*a^4*d+C*a^2*b^2*d+2*C*a*b^3*c)/(a*d-b*c)^4/(a^2+b^2)^2*
```

```
ln(a+b*tan(f*x+e))+(A*b^2-B*a*b+C*a^2)*b^2/(a*d-b*c)^3/(a^2+b^2)/(a+b*tan(f
*x+e))+1/(a^2+b^2)^2/(c^2+d^2)^3*(1/2*(-3*A*a^2*c^2*d+A*a^2*d^3-2*A*a*b*c^3
+6*A*a*b*c*d^2+3*A*b^2*c^2*d-A*b^2*d^3+B*a^2*c^3-3*B*a^2*c*d^2-6*B*a*b*c^2*
d+2*B*a*b*d^3-B*b^2*c^3+3*B*b^2*c*d^2+3*C*a^2*c^2*d-C*a^2*d^3+2*C*a*b*c^3-6
*C*a*b*c*d^2-3*C*b^2*c^2*d+C*b^2*d^3)*ln(1+tan(f*x+e)^2)+(A*a^2*c^3-3*A*a^2
*c*d^2-6*A*a*b*c^2*d+2*A*a*b*d^3-A*b^2*c^3+3*A*b^2*c*d^2+3*B*a^2*c^2*d-B*a^
2*d^3+2*B*a*b*c^3-6*B*a*b*c*d^2-3*B*b^2*c^2*d+B*b^2*d^3-C*a^2*c^3+3*C*a^2*c
*d^2+6*C*a*b*c^2*d-2*C*a*b*d^3+C*b^2*c^3-3*C*b^2*c*d^2)*arctan(tan(f*x+e)))
-d*(2*A*a*c*d^3-4*A*b*c^2*d^2-2*A*b*d^4-B*a*c^2*d^2+B*a*d^4+3*B*b*c^3*d+B*b
*c*d^3-2*C*a*c*d^3-2*C*b*c^4)/(a*d-b*c)^3/(c^2+d^2)^2/(c+d*tan(f*x+e))+d*(3
*A*a^2*c^2*d^4-A*a^2*d^6-10*A*a*b*c^3*d^3-2*A*a*b*c*d^5+10*A*b^2*c^4*d^2+9*
A*b^2*c^2*d^4+3*A*b^2*d^6-B*a^2*c^3*d^3+3*B*a^2*c*d^5+4*B*a*b*c^4*d^2-6*B*a
*b*c^2*d^4-2*B*a*b*d^6-6*B*b^2*c^5*d-3*B*b^2*c^3*d^3-B*b^2*c*d^5-3*C*a^2*c^
2*d^4+C*a^2*d^6+10*C*a*b*c^3*d^3+2*C*a*b*c*d^5+3*C*b^2*c^6-C*b^2*c^4*d^2)/(
a*d-b*c)^4/(c^2+d^2)^3*ln(c+d*tan(f*x+e))-1/2*(A*d^2-B*c*d+C*c^2)*d/(a*d-b*
c)^2/(c^2+d^2)/(c+d*tan(f*x+e))^2)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 9567 vs. $2(862) = 1724$.

Time = 12.03 (sec) , antiderivative size = 9567, normalized size of antiderivative = 11.11

$$\int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{(a + b\tan(e + fx))^2(c + d\tan(e + fx))^3}\,dx = \text{Too large to display}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e
))^3,x, algorithm="fricas")
```

```
[Out] Too large to include
```

Sympy [F(-1)]

Timed out.

$$\int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{(a + b\tan(e + fx))^2(c + d\tan(e + fx))^3}\,dx = \text{Timed out}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e))**2/(c+d*tan(f*x
+e))**3,x)
```

```
[Out] Timed out
```

Maxima [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 2537 vs. 2(862) = 1724.

Time = 0.55 (sec) , antiderivative size = 2537, normalized size of antiderivative = 2.95

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^3}\,dx = \text{Too large to display}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e
))^3,x, algorithm="maxima")
```

```
[Out] 1/2*(2*(((A - C)*a^2 + 2*B*a*b - (A - C)*b^2)*c^3 + 3*(B*a^2 - 2*(A - C)*a*
b - B*b^2)*c^2*d - 3*((A - C)*a^2 + 2*B*a*b - (A - C)*b^2)*c*d^2 - (B*a^2 -
 2*(A - C)*a*b - B*b^2)*d^3)*(f*x + e)/((a^4 + 2*a^2*b^2 + b^4)*c^6 + 3*(a^
4 + 2*a^2*b^2 + b^4)*c^4*d^2 + 3*(a^4 + 2*a^2*b^2 + b^4)*c^2*d^4 + (a^4 + 2
*a^2*b^2 + b^4)*d^6) - 2*((B*a^2*b^4 - 2*(A - C)*a*b^5 - B*b^6)*c + (3*C*a^
4*b^2 - 4*B*a^3*b^3 + (5*A + C)*a^2*b^4 - 2*B*a*b^5 + 3*A*b^6)*d)*log(b*tan
(f*x + e) + a)/((a^4*b^4 + 2*a^2*b^6 + b^8)*c^4 - 4*(a^5*b^3 + 2*a^3*b^5 +
a*b^7)*c^3*d + 6*(a^6*b^2 + 2*a^4*b^4 + a^2*b^6)*c^2*d^2 - 4*(a^7*b + 2*a^5
*b^3 + a^3*b^5)*c*d^3 + (a^8 + 2*a^6*b^2 + a^4*b^4)*d^4) + 2*(3*C*b^2*c^6*d
 - 6*B*b^2*c^5*d^2 + (4*B*a*b + (10*A - C)*b^2)*c^4*d^3 - (B*a^2 + 10*(A -
C)*a*b + 3*B*b^2)*c^3*d^4 + 3*((A - C)*a^2 - 2*B*a*b + 3*A*b^2)*c^2*d^5 + (
3*B*a^2 - 2*(A - C)*a*b - B*b^2)*c*d^6 - ((A - C)*a^2 + 2*B*a*b - 3*A*b^2)*
d^7)*log(d*tan(f*x + e) + c)/(b^4*c^10 - 4*a*b^3*c^9*d - 4*a^3*b*c*d^9 + a^
4*d^10 + 3*(2*a^2*b^2 + b^4)*c^8*d^2 - 4*(a^3*b + 3*a*b^3)*c^7*d^3 + (a^4 +
 18*a^2*b^2 + 3*b^4)*c^6*d^4 - 12*(a^3*b + a*b^3)*c^5*d^5 + (3*a^4 + 18*a^2
*b^2 + b^4)*c^4*d^6 - 4*(3*a^3*b + a*b^3)*c^3*d^7 + 3*(a^4 + 2*a^2*b^2)*c^2
*d^8) + ((B*a^2 - 2*(A - C)*a*b - B*b^2)*c^3 - 3*((A - C)*a^2 + 2*B*a*b - (
A - C)*b^2)*c^2*d - 3*(B*a^2 - 2*(A - C)*a*b - B*b^2)*c*d^2 + ((A - C)*a^2
+ 2*B*a*b - (A - C)*b^2)*d^3)*log(tan(f*x + e)^2 + 1)/((a^4 + 2*a^2*b^2 + b
^4)*c^6 + 3*(a^4 + 2*a^2*b^2 + b^4)*c^4*d^2 + 3*(a^4 + 2*a^2*b^2 + b^4)*c^2
*d^4 + (a^4 + 2*a^2*b^2 + b^4)*d^6) - (2*(C*a^2*b^2 - B*a*b^3 + A*b^4)*c^6
+ 5*(C*a^3*b + C*a*b^3)*c^5*d - (C*a^4 + 7*B*a^3*b - 3*C*a^2*b^2 + 11*B*a*b
^3 - 4*A*b^4)*c^4*d^2 + (3*B*a^4 + (9*A + C)*a^3*b + 3*B*a^2*b^2 + (9*A + C
)*a*b^3)*c^3*d^3 - ((5*A - 3*C)*a^4 + 3*B*a^3*b + 5*(A - C)*a^2*b^2 + 5*B*a
*b^3 - 2*A*b^4)*c^2*d^4 - (B*a^4 - 5*A*a^3*b + B*a^2*b^2 - 5*A*a*b^3)*c*d^5
 - (A*a^4 + A*a^2*b^2)*d^6 + 2*((3*C*a^2*b^2 - B*a*b^3 + (A + 2*C)*b^4)*c^4
*d^2 - 3*(B*a^2*b^2 + B*b^4)*c^3*d^3 + (B*a^3*b + 2*(2*A + C)*a^2*b^2 - B*a
*b^3 + 6*A*b^4)*c^2*d^4 - (2*(A - C)*a^3*b + B*a^2*b^2 + 2*(A - C)*a*b^3 +
B*b^4)*c*d^5 - (B*a^3*b - (2*A + C)*a^2*b^2 + 2*B*a*b^3 - 3*A*b^4)*d^6)*tan
(f*x + e)^2 + ((9*C*a^2*b^2 - 4*B*a*b^3 + (4*A + 5*C)*b^4)*c^5*d + (3*C*a^3
*b - 7*B*a^2*b^2 + 3*C*a*b^3 - 7*B*b^4)*c^4*d^2 - (3*B*a^3*b - 9*(A + C)*a^
2*b^2 + 11*B*a*b^3 - (17*A + C)*b^4)*c^3*d^3 + (2*B*a^4 + 3*(A + C)*a^3*b -
 B*a^2*b^2 + 3*(A + C)*a*b^3 - 3*B*b^4)*c^2*d^4 - (4*(A - C)*a^4 + 3*B*a^3*
b - (A + 8*C)*a^2*b^2 + 7*B*a*b^3 - 9*A*b^4)*c*d^5 - (2*B*a^4 - 3*A*a^3*b +
```

```
 2*B*a^2*b^2 - 3*A*a*b^3)*d^6)*tan(f*x + e))/((a^3*b^3 + a*b^5)*c^9 - 3*(a^
4*b^2 + a^2*b^4)*c^8*d + (3*a^5*b + 5*a^3*b^3 + 2*a*b^5)*c^7*d^2 - (a^6 + 7
*a^4*b^2 + 6*a^2*b^4)*c^6*d^3 + (6*a^5*b + 7*a^3*b^3 + a*b^5)*c^5*d^4 - (2*
a^6 + 5*a^4*b^2 + 3*a^2*b^4)*c^4*d^5 + 3*(a^5*b + a^3*b^3)*c^3*d^6 - (a^6 +
 a^4*b^2)*c^2*d^7 + ((a^2*b^4 + b^6)*c^7*d^2 - 3*(a^3*b^3 + a*b^5)*c^6*d^3
+ (3*a^4*b^2 + 5*a^2*b^4 + 2*b^6)*c^5*d^4 - (a^5*b + 7*a^3*b^3 + 6*a*b^5)*c
^4*d^5 + (6*a^4*b^2 + 7*a^2*b^4 + b^6)*c^3*d^6 - (2*a^5*b + 5*a^3*b^3 + 3*a
*b^5)*c^2*d^7 + 3*(a^4*b^2 + a^2*b^4)*c*d^8 - (a^5*b + a^3*b^3)*d^9)*tan(f*
x + e)^3 + (2*(a^2*b^4 + b^6)*c^8*d - 5*(a^3*b^3 + a*b^5)*c^7*d^2 + (3*a^4*
b^2 + 7*a^2*b^4 + 4*b^6)*c^6*d^3 + (a^5*b - 9*a^3*b^3 - 10*a*b^5)*c^5*d^4 -
 (a^6 - 5*a^4*b^2 - 8*a^2*b^4 - 2*b^6)*c^4*d^5 + (2*a^5*b - 3*a^3*b^3 - 5*a
*b^5)*c^3*d^6 - (2*a^6 - a^4*b^2 - 3*a^2*b^4)*c^2*d^7 + (a^5*b + a^3*b^3)*c
*d^8 - (a^6 + a^4*b^2)*d^9)*tan(f*x + e)^2 + ((a^2*b^4 + b^6)*c^9 - (a^3*b^
3 + a*b^5)*c^8*d - (3*a^4*b^2 + a^2*b^4 - 2*b^6)*c^7*d^2 + (5*a^5*b + 3*a^3
*b^3 - 2*a*b^5)*c^6*d^3 - (2*a^6 + 8*a^4*b^2 + 5*a^2*b^4 - b^6)*c^5*d^4 + (
10*a^5*b + 9*a^3*b^3 - a*b^5)*c^4*d^5 - (4*a^6 + 7*a^4*b^2 + 3*a^2*b^4)*c^3
*d^6 + 5*(a^5*b + a^3*b^3)*c^2*d^7 - 2*(a^6 + a^4*b^2)*c*d^8)*tan(f*x + e))
)/f
```

Giac [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 3115 vs. $2(862) = 1724$.

Time = 1.10 (sec) , antiderivative size = 3115, normalized size of antiderivative = 3.62

$$\int \frac{A + B\tan(e+fx) + C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^3}\,dx = \text{Too large to display}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e
))^3,x, algorithm="giac")
```

```
[Out] 1/2*(2*(A*a^2*c^3 - C*a^2*c^3 + 2*B*a*b*c^3 - A*b^2*c^3 + C*b^2*c^3 + 3*B*a
^2*c^2*d - 6*A*a*b*c^2*d + 6*C*a*b*c^2*d - 3*B*b^2*c^2*d - 3*A*a^2*c*d^2 +
3*C*a^2*c*d^2 - 6*B*a*b*c*d^2 + 3*A*b^2*c*d^2 - 3*C*b^2*c*d^2 - B*a^2*d^3 +
 2*A*a*b*d^3 - 2*C*a*b*d^3 + B*b^2*d^3)*(f*x + e)/(a^4*c^6 + 2*a^2*b^2*c^6
+ b^4*c^6 + 3*a^4*c^4*d^2 + 6*a^2*b^2*c^4*d^2 + 3*b^4*c^4*d^2 + 3*a^4*c^2*d
^4 + 6*a^2*b^2*c^2*d^4 + 3*b^4*c^2*d^4 + a^4*d^6 + 2*a^2*b^2*d^6 + b^4*d^6)
 + (B*a^2*c^3 - 2*A*a*b*c^3 + 2*C*a*b*c^3 - B*b^2*c^3 - 3*A*a^2*c^2*d + 3*C
*a^2*c^2*d - 6*B*a*b*c^2*d + 3*A*b^2*c^2*d - 3*C*b^2*c^2*d - 3*B*a^2*c*d^2
+ 6*A*a*b*c*d^2 - 6*C*a*b*c*d^2 + 3*B*b^2*c*d^2 + A*a^2*d^3 - C*a^2*d^3 + 2
*B*a*b*d^3 - A*b^2*d^3 + C*b^2*d^3)*log(tan(f*x + e)^2 + 1)/(a^4*c^6 + 2*a^
2*b^2*c^6 + b^4*c^6 + 3*a^4*c^4*d^2 + 6*a^2*b^2*c^4*d^2 + 3*b^4*c^4*d^2 + 3
*a^4*c^2*d^4 + 6*a^2*b^2*c^2*d^4 + 3*b^4*c^2*d^4 + a^4*d^6 + 2*a^2*b^2*d^6
+ b^4*d^6) - 2*(B*a^2*b^5*c - 2*A*a*b^6*c + 2*C*a*b^6*c - B*b^7*c + 3*C*a^4
*b^3*d - 4*B*a^3*b^4*d + 5*A*a^2*b^5*d + C*a^2*b^5*d - 2*B*a*b^6*d + 3*A*b^
7*d)*log(abs(b*tan(f*x + e) + a))/(a^4*b^5*c^4 + 2*a^2*b^7*c^4 + b^9*c^4 -
```

4*a^5*b^4*c^3*d - 8*a^3*b^6*c^3*d - 4*a*b^8*c^3*d + 6*a^6*b^3*c^2*d^2 + 12*
a^4*b^5*c^2*d^2 + 6*a^2*b^7*c^2*d^2 - 4*a^7*b^2*c*d^3 - 8*a^5*b^4*c*d^3 - 4
*a^3*b^6*c*d^3 + a^8*b*d^4 + 2*a^6*b^3*d^4 + a^4*b^5*d^4) + 2*(3*C*b^2*c^6*
d^2 - 6*B*b^2*c^5*d^3 + 4*B*a*b*c^4*d^4 + 10*A*b^2*c^4*d^4 - C*b^2*c^4*d^4
- B*a^2*c^3*d^5 - 10*A*a*b*c^3*d^5 + 10*C*a*b*c^3*d^5 - 3*B*b^2*c^3*d^5 + 3
*A*a^2*c^2*d^6 - 3*C*a^2*c^2*d^6 - 6*B*a*b*c^2*d^6 + 9*A*b^2*c^2*d^6 + 3*B*
a^2*c*d^7 - 2*A*a*b*c*d^7 + 2*C*a*b*c*d^7 - B*b^2*c*d^7 - A*a^2*d^8 + C*a^2
*d^8 - 2*B*a*b*d^8 + 3*A*b^2*d^8)*log(abs(d*tan(f*x + e) + c))/(b^4*c^10*d
- 4*a*b^3*c^9*d^2 + 6*a^2*b^2*c^8*d^3 + 3*b^4*c^8*d^3 - 4*a^3*b*c^7*d^4 - 1
2*a*b^3*c^7*d^4 + a^4*c^6*d^5 + 18*a^2*b^2*c^6*d^5 + 3*b^4*c^6*d^5 - 12*a^3
*b*c^5*d^6 - 12*a*b^3*c^5*d^6 + 3*a^4*c^4*d^7 + 18*a^2*b^2*c^4*d^7 + b^4*c^
4*d^7 - 12*a^3*b*c^3*d^8 - 4*a*b^3*c^3*d^8 + 3*a^4*c^2*d^9 + 6*a^2*b^2*c^2*
d^9 - 4*a^3*b*c*d^10 + a^4*d^11) + 2*(B*a^2*b^5*c*tan(f*x + e) - 2*A*a*b^6*
c*tan(f*x + e) + 2*C*a*b^6*c*tan(f*x + e) - B*b^7*c*tan(f*x + e) + 3*C*a^4*
b^3*d*tan(f*x + e) - 4*B*a^3*b^4*d*tan(f*x + e) + 5*A*a^2*b^5*d*tan(f*x + e
) + C*a^2*b^5*d*tan(f*x + e) - 2*B*a*b^6*d*tan(f*x + e) + 3*A*b^7*d*tan(f*x
 + e) - C*a^4*b^3*c + 2*B*a^3*b^4*c - 3*A*a^2*b^5*c + C*a^2*b^5*c - A*b^7*c
 + 4*C*a^5*b^2*d - 5*B*a^4*b^3*d + 6*A*a^3*b^4*d + 2*C*a^3*b^4*d - 3*B*a^2*
b^5*d + 4*A*a*b^6*d)/((a^4*b^4*c^4 + 2*a^2*b^6*c^4 + b^8*c^4 - 4*a^5*b^3*c^
3*d - 8*a^3*b^5*c^3*d - 4*a*b^7*c^3*d + 6*a^6*b^2*c^2*d^2 + 12*a^4*b^4*c^2*
d^2 + 6*a^2*b^6*c^2*d^2 - 4*a^7*b*c*d^3 - 8*a^5*b^3*c*d^3 - 4*a^3*b^5*c*d^3
 + a^8*d^4 + 2*a^6*b^2*d^4 + a^4*b^4*d^4)*(b*tan(f*x + e) + a)) - (9*C*b^2*
c^6*d^3*tan(f*x + e)^2 - 18*B*b^2*c^5*d^4*tan(f*x + e)^2 + 12*B*a*b*c^4*d^5
*tan(f*x + e)^2 + 30*A*b^2*c^4*d^5*tan(f*x + e)^2 - 3*C*b^2*c^4*d^5*tan(f*x
 + e)^2 - 3*B*a^2*c^3*d^6*tan(f*x + e)^2 - 30*A*a*b*c^3*d^6*tan(f*x + e)^2
+ 30*C*a*b*c^3*d^6*tan(f*x + e)^2 - 9*B*b^2*c^3*d^6*tan(f*x + e)^2 + 9*A*a^
2*c^2*d^7*tan(f*x + e)^2 - 9*C*a^2*c^2*d^7*tan(f*x + e)^2 - 18*B*a*b*c^2*d^
7*tan(f*x + e)^2 + 27*A*b^2*c^2*d^7*tan(f*x + e)^2 + 9*B*a^2*c*d^8*tan(f*x
+ e)^2 - 6*A*a*b*c*d^8*tan(f*x + e)^2 + 6*C*a*b*c*d^8*tan(f*x + e)^2 - 3*B*
b^2*c*d^8*tan(f*x + e)^2 - 3*A*a^2*d^9*tan(f*x + e)^2 + 3*C*a^2*d^9*tan(f*x
 + e)^2 - 6*B*a*b*d^9*tan(f*x + e)^2 + 9*A*b^2*d^9*tan(f*x + e)^2 + 22*C*b^
2*c^7*d^2*tan(f*x + e) - 4*C*a*b*c^6*d^3*tan(f*x + e) - 42*B*b^2*c^6*d^3*ta
n(f*x + e) + 32*B*a*b*c^5*d^4*tan(f*x + e) + 68*A*b^2*c^5*d^4*tan(f*x + e)
- 2*C*b^2*c^5*d^4*tan(f*x + e) - 8*B*a^2*c^4*d^5*tan(f*x + e) - 72*A*a*b*c^
4*d^5*tan(f*x + e) + 60*C*a*b*c^4*d^5*tan(f*x + e) - 26*B*b^2*c^4*d^5*tan(f
*x + e) + 22*A*a^2*c^3*d^6*tan(f*x + e) - 22*C*a^2*c^3*d^6*tan(f*x + e) - 2
8*B*a*b*c^3*d^6*tan(f*x + e) + 66*A*b^2*c^3*d^6*tan(f*x + e) + 18*B*a^2*c^2
*d^7*tan(f*x + e) - 28*A*a*b*c^2*d^7*tan(f*x + e) + 16*C*a*b*c^2*d^7*tan(f*
x + e) - 8*B*b^2*c^2*d^7*tan(f*x + e) - 2*A*a^2*c*d^8*tan(f*x + e) + 2*C*a^
2*c*d^8*tan(f*x + e) - 12*B*a*b*c*d^8*tan(f*x + e) + 22*A*b^2*c*d^8*tan(f*x
 + e) + 2*B*a^2*d^9*tan(f*x + e) - 4*A*a*b*d^9*tan(f*x + e) + 14*C*b^2*c^8*
d - 6*C*a*b*c^7*d^2 - 25*B*b^2*c^7*d^2 + C*a^2*c^6*d^3 + 22*B*a*b*c^6*d^3 +
 39*A*b^2*c^6*d^3 + 3*C*b^2*c^6*d^3 - 6*B*a^2*c^5*d^4 - 44*A*a*b*c^5*d^4 +
26*C*a*b*c^5*d^4 - 19*B*b^2*c^5*d^4 + 14*A*a^2*c^4*d^5 - 11*C*a^2*c^4*d^5 -
 6*B*a*b*c^4*d^5 + 41*A*b^2*c^4*d^5 + C*b^2*c^4*d^5 + 7*B*a^2*c^3*d^6 - 26*

```
A*a*b*c^3*d^6 + 8*C*a*b*c^3*d^6 - 6*B*b^2*c^3*d^6 + 3*A*a^2*c^2*d^7 - 4*B*a
*b*c^2*d^7 + 14*A*b^2*c^2*d^7 + B*a^2*c*d^8 - 6*A*a*b*c*d^8 + A*a^2*d^9)/((
b^4*c^10 - 4*a*b^3*c^9*d + 6*a^2*b^2*c^8*d^2 + 3*b^4*c^8*d^2 - 4*a^3*b*c^7*
d^3 - 12*a*b^3*c^7*d^3 + a^4*c^6*d^4 + 18*a^2*b^2*c^6*d^4 + 3*b^4*c^6*d^4 -
 12*a^3*b*c^5*d^5 - 12*a*b^3*c^5*d^5 + 3*a^4*c^4*d^6 + 18*a^2*b^2*c^4*d^6 +
 b^4*c^4*d^6 - 12*a^3*b*c^3*d^7 - 4*a*b^3*c^3*d^7 + 3*a^4*c^2*d^8 + 6*a^2*b
^2*c^2*d^8 - 4*a^3*b*c*d^9 + a^4*d^10)*(d*tan(f*x + e) + c)^2))/f
```

Mupad [B] (verification not implemented)

Time = 43.73 (sec) , antiderivative size = 128666, normalized size of antiderivative = 149.44

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^3}\,dx = \text{Too large to display}$$

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))^2*(c + d*
tan(e + f*x))^3),x)
```

```
[Out] (((2*A*b^4*c^6 - A*a^4*d^6 - 2*B*a*b^3*c^6 - B*a^4*c*d^5 - A*a^2*b^2*d^6 -
5*A*a^4*c^2*d^4 + 2*C*a^2*b^2*c^6 + 2*A*b^4*c^2*d^4 + 4*A*b^4*c^4*d^2 + 3*B
*a^4*c^3*d^3 + 3*C*a^4*c^2*d^4 - C*a^4*c^4*d^2 + 9*A*a*b^3*c^3*d^3 + 9*A*a^
3*b*c^3*d^3 - 5*B*a*b^3*c^2*d^4 - 11*B*a*b^3*c^4*d^2 - B*a^2*b^2*c*d^5 - 3*
B*a^3*b*c^2*d^4 - 7*B*a^3*b*c^4*d^2 + C*a*b^3*c^3*d^3 + C*a^3*b*c^3*d^3 - 5
*A*a^2*b^2*c^2*d^4 + 3*B*a^2*b^2*c^3*d^3 + 5*C*a^2*b^2*c^2*d^4 + 3*C*a^2*b^
2*c^4*d^2 + 5*A*a*b^3*c*d^5 + 5*A*a^3*b*c*d^5 + 5*C*a*b^3*c^5*d + 5*C*a^3*b
*c^5*d)/(2*(a^3*d^3 - b^3*c^3 + 3*a*b^2*c^2*d - 3*a^2*b*c*d^2)*(a^2*c^4 + a
^2*d^4 + b^2*c^4 + b^2*d^4 + 2*a^2*c^2*d^2 + 2*b^2*c^2*d^2)) + (tan(e + f*x
)*(3*A*a*b^3*d^6 - 2*B*a^4*d^6 + 3*A*a^3*b*d^6 - 4*A*a^4*c*d^5 + 9*A*b^4*c*
d^5 + 4*A*b^4*c^5*d + 4*C*a^4*c*d^5 + 5*C*b^4*c^5*d - 2*B*a^2*b^2*d^6 + 17*
A*b^4*c^3*d^3 + 2*B*a^4*c^2*d^4 - 3*B*b^4*c^2*d^4 - 7*B*b^4*c^4*d^2 + C*b^4
*c^3*d^3 + 3*A*a*b^3*c^2*d^4 + A*a^2*b^2*c*d^5 + 3*A*a^3*b*c^2*d^4 - 11*B*a
*b^3*c^3*d^3 - 3*B*a^3*b*c^3*d^3 + 3*C*a*b^3*c^2*d^4 + 3*C*a*b^3*c^4*d^2 +
8*C*a^2*b^2*c*d^5 + 9*C*a^2*b^2*c^5*d + 3*C*a^3*b*c^2*d^4 + 3*C*a^3*b*c^4*d
^2 + 9*A*a^2*b^2*c^3*d^3 - B*a^2*b^2*c^2*d^4 - 7*B*a^2*b^2*c^4*d^2 + 9*C*a^
2*b^2*c^3*d^3 - 7*B*a*b^3*c*d^5 - 4*B*a*b^3*c^5*d - 3*B*a^3*b*c*d^5))/(2*(a
^3*d^3 - b^3*c^3 + 3*a*b^2*c^2*d - 3*a^2*b*c*d^2)*(a^2*c^4 + a^2*d^4 + b^2*
c^4 + b^2*d^4 + 2*a^2*c^2*d^2 + 2*b^2*c^2*d^2)) + (tan(e + f*x)^2*(3*A*b^4*
d^6 - 2*B*a*b^3*d^6 - B*a^3*b*d^6 - B*b^4*c*d^5 + 2*A*a^2*b^2*d^6 + 6*A*b^4
*c^2*d^4 + A*b^4*c^4*d^2 + C*a^2*b^2*d^6 - 3*B*b^4*c^3*d^3 + 2*C*b^4*c^4*d^
2 - B*a*b^3*c^2*d^4 - B*a*b^3*c^4*d^2 - B*a^2*b^2*c*d^5 + B*a^3*b*c^2*d^4 +
 4*A*a^2*b^2*c^2*d^4 - 3*B*a^2*b^2*c^3*d^3 + 2*C*a^2*b^2*c^2*d^4 + 3*C*a^2*
b^2*c^4*d^2 - 2*A*a*b^3*c*d^5 - 2*A*a^3*b*c*d^5 + 2*C*a*b^3*c*d^5 + 2*C*a^3
*b*c*d^5))/((a^3*d^3 - b^3*c^3 + 3*a*b^2*c^2*d - 3*a^2*b*c*d^2)*(a^2*c^4 +
a^2*d^4 + b^2*c^4 + b^2*d^4 + 2*a^2*c^2*d^2 + 2*b^2*c^2*d^2)))/(tan(e + f*x
)*(b*c^2 + 2*a*c*d) + a*c^2 + tan(e + f*x)^2*(a*d^2 + 2*b*c*d) + b*d^2*tan(
```

e + f*x)^3) + symsum(log((3*A^3*a^3*b^6*d^10 - A^3*a^5*b^4*d^10 + 4*B^3*a^2
*b^7*d^10 + 6*B^3*a^4*b^5*d^10 + 24*A^3*b^9*c^3*d^7 + 27*A^3*b^9*c^5*d^5 +
C^3*a^5*b^4*d^10 + B^3*b^9*c^2*d^8 + 4*B^3*b^9*c^4*d^6 + 7*B^3*b^9*c^6*d^4
+ 9*A^2*B*b^9*d^10 + 9*A^3*b^9*c*d^9 + 26*A^3*a^2*b^7*c^3*d^7 + 31*A^3*a^2*
b^7*c^5*d^5 + 16*A^3*a^3*b^6*c^2*d^8 - 11*A^3*a^3*b^6*c^4*d^6 - 6*A^3*a^4*b
^5*c^3*d^7 + 3*A^3*a^5*b^4*c^2*d^8 + 5*B^3*a^2*b^7*c^2*d^8 - 14*B^3*a^2*b^7
*c^4*d^6 + 9*B^3*a^2*b^7*c^6*d^4 + 28*B^3*a^3*b^6*c^3*d^7 + 19*B^3*a^3*b^6*
c^5*d^5 + 6*B^3*a^4*b^5*c^2*d^8 - 20*B^3*a^4*b^5*c^4*d^6 + 7*B^3*a^5*b^4*c^
3*d^7 + C^3*a^2*b^7*c^3*d^7 - 4*C^3*a^2*b^7*c^5*d^5 - 9*C^3*a^2*b^7*c^7*d^3
 - 7*C^3*a^3*b^6*c^2*d^8 - 28*C^3*a^3*b^6*c^4*d^6 + 3*C^3*a^3*b^6*c^6*d^4 +
 15*C^3*a^4*b^5*c^3*d^7 - 9*C^3*a^4*b^5*c^7*d^3 - 3*C^3*a^5*b^4*c^2*d^8 - 2
4*C^3*a^5*b^4*c^4*d^6 + 6*C^3*a^6*b^3*c^3*d^7 - 12*A*B^2*a*b^8*d^10 - 6*A*B
^2*b^9*c*d^9 - 9*A^2*C*b^9*c*d^9 + 4*B^3*a*b^8*c*d^9 - 17*A*B^2*a^3*b^6*d^1
0 + 3*A*B^2*a^5*b^4*d^10 + 12*A^2*B*a^2*b^7*d^10 - 7*A^2*B*a^4*b^5*d^10 + 3
*A*C^2*a^3*b^6*d^10 - 3*A*C^2*a^5*b^4*d^10 - 6*A^2*C*a^3*b^6*d^10 + 3*A^2*C
*a^5*b^4*d^10 - 20*A*B^2*b^9*c^3*d^7 - 28*A*B^2*b^9*c^5*d^5 + 6*A*B^2*b^9*c
^7*d^3 - B*C^2*a^4*b^5*d^10 + 3*B*C^2*a^6*b^3*d^10 + 21*A^2*B*b^9*c^2*d^8 +
 13*A^2*B*b^9*c^4*d^6 - 27*A^2*B*b^9*c^6*d^4 - 4*B^2*C*a^3*b^6*d^10 - 9*B^2
*C*a^5*b^4*d^10 - 3*A*C^2*b^9*c^3*d^7 - 9*A*C^2*b^9*c^7*d^3 - 21*A^2*C*b^9*
c^3*d^7 - 27*A^2*C*b^9*c^5*d^5 + 9*A^2*C*b^9*c^7*d^3 + B*C^2*b^9*c^4*d^6 +
3*B*C^2*b^9*c^8*d^2 - B^2*C*b^9*c^3*d^7 - 2*B^2*C*b^9*c^5*d^5 - 9*B^2*C*b^9
*c^7*d^3 - 3*A^3*a*b^8*c^2*d^8 - 31*A^3*a*b^8*c^4*d^6 - 8*A^3*a*b^8*c^6*d^4
 + 3*A^3*a^2*b^7*c*d^9 - 10*A^3*a^4*b^5*c*d^9 + 11*B^3*a*b^8*c^3*d^7 + 5*B^
3*a*b^8*c^5*d^5 - 6*B^3*a*b^8*c^7*d^3 + B^3*a^3*b^6*c*d^9 - 5*B^3*a^5*b^4*c
*d^9 - 2*C^3*a*b^8*c^4*d^6 - C^3*a*b^8*c^6*d^4 - 3*C^3*a*b^8*c^8*d^2 - 2*C^
3*a^4*b^5*c*d^9 - 6*C^3*a^6*b^3*c*d^9 - 4*A*B^2*a^2*b^7*c^3*d^7 - 77*A*B^2*
a^2*b^7*c^5*d^5 - 6*A*B^2*a^2*b^7*c^7*d^3 - 60*A*B^2*a^3*b^6*c^2*d^8 + 25*A
*B^2*a^3*b^6*c^4*d^6 + 28*A*B^2*a^3*b^6*c^6*d^4 + 44*A*B^2*a^4*b^5*c^3*d^7
 - 17*A*B^2*a^4*b^5*c^5*d^5 - 21*A*B^2*a^5*b^4*c^2*d^8 + 4*A*B^2*a^5*b^4*c^4
*d^6 + 71*A^2*B*a^2*b^7*c^2*d^8 + 86*A^2*B*a^2*b^7*c^4*d^6 - 13*A^2*B*a^2*b
^7*c^6*d^4 - 116*A^2*B*a^3*b^6*c^3*d^7 - 37*A^2*B*a^3*b^6*c^5*d^5 + 16*A^2*
B*a^4*b^5*c^2*d^8 + 35*A^2*B*a^4*b^5*c^4*d^6 - 9*A^2*B*a^5*b^4*c^3*d^7 - 30
*A*C^2*a^2*b^7*c^3*d^7 - 15*A*C^2*a^2*b^7*c^5*d^5 + 30*A*C^2*a^3*b^6*c^2*d^
8 + 45*A*C^2*a^3*b^6*c^4*d^6 - 6*A*C^2*a^3*b^6*c^6*d^4 - 63*A*C^2*a^4*b^5*c
^3*d^7 - 27*A*C^2*a^4*b^5*c^5*d^5 + 9*A*C^2*a^4*b^5*c^7*d^3 + 9*A*C^2*a^5*b
^4*c^2*d^8 + 48*A*C^2*a^5*b^4*c^4*d^6 - 12*A*C^2*a^6*b^3*c^3*d^7 + 3*A^2*C*
a^2*b^7*c^3*d^7 - 12*A^2*C*a^2*b^7*c^5*d^5 + 9*A^2*C*a^2*b^7*c^7*d^3 - 39*A
^2*C*a^3*b^6*c^2*d^8 - 6*A^2*C*a^3*b^6*c^4*d^6 + 3*A^2*C*a^3*b^6*c^6*d^4 +
54*A^2*C*a^4*b^5*c^3*d^7 + 27*A^2*C*a^4*b^5*c^5*d^5 - 9*A^2*C*a^5*b^4*c^2*d
^8 - 24*A^2*C*a^5*b^4*c^4*d^6 + 6*A^2*C*a^6*b^3*c^3*d^7 + 11*B*C^2*a^2*b^7*
c^2*d^8 + 47*B*C^2*a^2*b^7*c^4*d^6 + 17*B*C^2*a^2*b^7*c^6*d^4 - 3*B*C^2*a^2
*b^7*c^8*d^2 + 16*B*C^2*a^3*b^6*c^3*d^7 - 25*B*C^2*a^3*b^6*c^5*d^5 + 12*B*C
^2*a^3*b^6*c^7*d^3 - 17*B*C^2*a^4*b^5*c^2*d^8 + 47*B*C^2*a^4*b^5*c^4*d^6 +
27*B*C^2*a^4*b^5*c^6*d^4 + 39*B*C^2*a^5*b^4*c^3*d^7 - 12*B*C^2*a^5*b^4*c^5*
d^5 - 18*B*C^2*a^6*b^3*c^2*d^8 + 3*B*C^2*a^6*b^3*c^4*d^6 - 35*B^2*C*a^2*b^7

*c^3*d^7 + 26*B^2*C*a^2*b^7*c^5*d^5 + 3*B^2*C*a^2*b^7*c^7*d^3 + 9*B^2*C*a^3
*b^6*c^2*d^8 - 16*B^2*C*a^3*b^6*c^4*d^6 - 37*B^2*C*a^3*b^6*c^6*d^4 - 68*B^2
*C*a^4*b^5*c^3*d^7 - 4*B^2*C*a^4*b^5*c^5*d^5 + 9*B^2*C*a^5*b^4*c^2*d^8 + 14
*B^2*C*a^5*b^4*c^4*d^6 - 6*B^2*C*a^6*b^3*c^3*d^7 + 6*A*B*C*a^2*b^7*d^10 + 1
7*A*B*C*a^4*b^5*d^10 - 3*A*B*C*a^6*b^3*d^10 + 6*A*B*C*b^9*c^2*d^8 + 13*A*B*
C*b^9*c^4*d^6 + 36*A*B*C*b^9*c^6*d^4 - 3*A*B*C*b^9*c^8*d^2 - 24*A^2*B*a*b^8
*c*d^9 - 19*A*B^2*a*b^8*c^2*d^8 + 37*A*B^2*a*b^8*c^4*d^6 + 32*A*B^2*a*b^8*c
^6*d^4 + 11*A*B^2*a^2*b^7*c*d^9 + 25*A*B^2*a^4*b^5*c*d^9 - 81*A^2*B*a*b^8*c
^3*d^7 - 15*A^2*B*a*b^8*c^5*d^5 + 6*A^2*B*a*b^8*c^7*d^3 - 23*A^2*B*a^3*b^6*
c*d^9 + 11*A^2*B*a^5*b^4*c*d^9 - 3*A*C^2*a*b^8*c^2*d^8 - 27*A*C^2*a*b^8*c^4
*d^6 - 6*A*C^2*a*b^8*c^6*d^4 + 6*A*C^2*a*b^8*c^8*d^2 - 15*A*C^2*a^2*b^7*c*d
^9 - 15*A*C^2*a^4*b^5*c*d^9 + 12*A*C^2*a^6*b^3*c*d^9 + 6*A^2*C*a*b^8*c^2*d^
8 + 60*A^2*C*a*b^8*c^4*d^6 + 15*A^2*C*a*b^8*c^6*d^4 - 3*A^2*C*a*b^8*c^8*d^2
 + 12*A^2*C*a^2*b^7*c*d^9 + 27*A^2*C*a^4*b^5*c*d^9 - 6*A^2*C*a^6*b^3*c*d^9
+ 3*B*C^2*a*b^8*c^3*d^7 + 9*B*C^2*a*b^8*c^5*d^5 + 18*B*C^2*a*b^8*c^7*d^3 +
13*B*C^2*a^3*b^6*c*d^9 + 23*B*C^2*a^5*b^4*c*d^9 - 8*B^2*C*a*b^8*c^2*d^8 - 2
8*B^2*C*a*b^8*c^4*d^6 - 29*B^2*C*a*b^8*c^6*d^4 + 3*B^2*C*a*b^8*c^8*d^2 - 14
*B^2*C*a^2*b^7*c*d^9 - 16*B^2*C*a^4*b^5*c*d^9 + 6*B^2*C*a^6*b^3*c*d^9 - 28*
A*B*C*a^2*b^7*c^2*d^8 - 79*A*B*C*a^2*b^7*c^4*d^6 + 14*A*B*C*a^2*b^7*c^6*d^4
 + 3*A*B*C*a^2*b^7*c^8*d^2 + 100*A*B*C*a^3*b^6*c^3*d^7 + 62*A*B*C*a^3*b^6*c
^5*d^5 - 12*A*B*C*a^3*b^6*c^7*d^3 + 28*A*B*C*a^4*b^5*c^2*d^8 - 55*A*B*C*a^4
*b^5*c^4*d^6 - 18*A*B*C*a^4*b^5*c^6*d^4 - 30*A*B*C*a^5*b^4*c^3*d^7 + 12*A*B
*C*a^5*b^4*c^5*d^5 + 18*A*B*C*a^6*b^3*c^2*d^8 - 3*A*B*C*a^6*b^3*c^4*d^6 + 2
4*A*B*C*a*b^8*c*d^9 + 78*A*B*C*a*b^8*c^3*d^7 + 6*A*B*C*a*b^8*c^5*d^5 - 24*A
*B*C*a*b^8*c^7*d^3 + 10*A*B*C*a^3*b^6*c*d^9 - 34*A*B*C*a^5*b^4*c*d^9)/(a^10
*d^14 + b^10*c^14 + 2*a^2*b^8*c^14 + a^4*b^6*c^14 + a^6*b^4*d^14 + 2*a^8*b^
2*d^14 + 4*a^10*c^2*d^12 + 6*a^10*c^4*d^10 + 4*a^10*c^6*d^8 + a^10*c^8*d^6
+ b^10*c^6*d^8 + 4*b^10*c^8*d^6 + 6*b^10*c^10*d^4 + 4*b^10*c^12*d^2 - 6*a*b
^9*c^5*d^9 - 24*a*b^9*c^7*d^7 - 36*a*b^9*c^9*d^5 - 24*a*b^9*c^11*d^3 - 12*a
^3*b^7*c^13*d - 6*a^5*b^5*c*d^13 - 6*a^5*b^5*c^13*d - 12*a^7*b^3*c*d^13 - 2
4*a^9*b*c^3*d^11 - 36*a^9*b*c^5*d^9 - 24*a^9*b*c^7*d^7 - 6*a^9*b*c^9*d^5 +
15*a^2*b^8*c^4*d^10 + 62*a^2*b^8*c^6*d^8 + 98*a^2*b^8*c^8*d^6 + 72*a^2*b^8*
c^10*d^4 + 23*a^2*b^8*c^12*d^2 - 20*a^3*b^7*c^3*d^11 - 92*a^3*b^7*c^5*d^9 -
 168*a^3*b^7*c^7*d^7 - 152*a^3*b^7*c^9*d^5 - 68*a^3*b^7*c^11*d^3 + 15*a^4*b
^6*c^2*d^12 + 90*a^4*b^6*c^4*d^10 + 211*a^4*b^6*c^6*d^8 + 244*a^4*b^6*c^8*d
^6 + 141*a^4*b^6*c^10*d^4 + 34*a^4*b^6*c^12*d^2 - 64*a^5*b^5*c^3*d^11 - 202
*a^5*b^5*c^5*d^9 - 288*a^5*b^5*c^7*d^7 - 202*a^5*b^5*c^9*d^5 - 64*a^5*b^5*c
^11*d^3 + 34*a^6*b^4*c^2*d^12 + 141*a^6*b^4*c^4*d^10 + 244*a^6*b^4*c^6*d^8
+ 211*a^6*b^4*c^8*d^6 + 90*a^6*b^4*c^10*d^4 + 15*a^6*b^4*c^12*d^2 - 68*a^7*
b^3*c^3*d^11 - 152*a^7*b^3*c^5*d^9 - 168*a^7*b^3*c^7*d^7 - 92*a^7*b^3*c^9*d
^5 - 20*a^7*b^3*c^11*d^3 + 23*a^8*b^2*c^2*d^12 + 72*a^8*b^2*c^4*d^10 + 98*a
^8*b^2*c^6*d^8 + 62*a^8*b^2*c^8*d^6 + 15*a^8*b^2*c^10*d^4 - 6*a*b^9*c^13*d
- 6*a^9*b*c*d^13) - root(640*a^15*b*c^7*d^13*f^4 + 640*a*b^15*c^13*d^7*f^4
+ 480*a^15*b*c^9*d^11*f^4 + 480*a^15*b*c^5*d^15*f^4 + 480*a*b^15*c^15*d^5*f
^4 + 480*a*b^15*c^11*d^9*f^4 + 192*a^15*b*c^11*d^9*f^4 + 192*a^15*b*c^3*d^1

7*f^4 + 192*a^11*b^5*c*d^19*f^4 + 192*a^5*b^11*c^19*d*f^4 + 192*a*b^15*c^17
*d^3*f^4 + 192*a*b^15*c^9*d^11*f^4 + 128*a^13*b^3*c*d^19*f^4 + 128*a^9*b^7*
c*d^19*f^4 + 128*a^7*b^9*c^19*d*f^4 + 128*a^3*b^13*c^19*d*f^4 + 32*a^15*b*c
^13*d^7*f^4 + 32*a^9*b^7*c^19*d*f^4 + 32*a^7*b^9*c*d^19*f^4 + 32*a*b^15*c^7
*d^13*f^4 + 32*a^15*b*c*d^19*f^4 + 32*a*b^15*c^19*d*f^4 - 47088*a^8*b^8*c^1
0*d^10*f^4 + 42432*a^9*b^7*c^9*d^11*f^4 + 42432*a^7*b^9*c^11*d^9*f^4 + 3932
8*a^9*b^7*c^11*d^9*f^4 + 39328*a^7*b^9*c^9*d^11*f^4 - 36912*a^8*b^8*c^12*d^
8*f^4 - 36912*a^8*b^8*c^8*d^12*f^4 - 34256*a^10*b^6*c^10*d^10*f^4 - 34256*a
^6*b^10*c^10*d^10*f^4 - 31152*a^10*b^6*c^8*d^12*f^4 - 31152*a^6*b^10*c^12*d
^8*f^4 + 28128*a^9*b^7*c^7*d^13*f^4 + 28128*a^7*b^9*c^13*d^7*f^4 + 24160*a^
11*b^5*c^9*d^11*f^4 + 24160*a^5*b^11*c^11*d^9*f^4 - 23088*a^10*b^6*c^12*d^8
*f^4 - 23088*a^6*b^10*c^8*d^12*f^4 + 22272*a^9*b^7*c^13*d^7*f^4 + 22272*a^7
*b^9*c^7*d^13*f^4 + 19072*a^11*b^5*c^11*d^9*f^4 + 19072*a^5*b^11*c^9*d^11*f
^4 + 18624*a^11*b^5*c^7*d^13*f^4 + 18624*a^5*b^11*c^13*d^7*f^4 - 17328*a^8*
b^8*c^14*d^6*f^4 - 17328*a^8*b^8*c^6*d^14*f^4 - 17232*a^10*b^6*c^6*d^14*f^4
 - 17232*a^6*b^10*c^14*d^6*f^4 - 13520*a^12*b^4*c^8*d^12*f^4 - 13520*a^4*b^
12*c^12*d^8*f^4 - 12464*a^12*b^4*c^10*d^10*f^4 - 12464*a^4*b^12*c^10*d^10*f
^4 + 10880*a^9*b^7*c^5*d^15*f^4 + 10880*a^7*b^9*c^15*d^5*f^4 - 9072*a^10*b^
6*c^14*d^6*f^4 - 9072*a^6*b^10*c^6*d^14*f^4 + 8928*a^11*b^5*c^13*d^7*f^4 +
8928*a^5*b^11*c^7*d^13*f^4 - 8880*a^12*b^4*c^6*d^14*f^4 - 8880*a^4*b^12*c^1
4*d^6*f^4 + 8480*a^11*b^5*c^5*d^15*f^4 + 8480*a^5*b^11*c^15*d^5*f^4 + 7200*
a^9*b^7*c^15*d^5*f^4 + 7200*a^7*b^9*c^5*d^15*f^4 - 6912*a^12*b^4*c^12*d^8*f
^4 - 6912*a^4*b^12*c^8*d^12*f^4 + 6400*a^13*b^3*c^9*d^11*f^4 + 6400*a^3*b^1
3*c^11*d^9*f^4 + 5920*a^13*b^3*c^7*d^13*f^4 + 5920*a^3*b^13*c^13*d^7*f^4 -
5392*a^10*b^6*c^4*d^16*f^4 - 5392*a^6*b^10*c^16*d^4*f^4 - 4428*a^8*b^8*c^16
*d^4*f^4 - 4428*a^8*b^8*c^4*d^16*f^4 + 4128*a^13*b^3*c^11*d^9*f^4 + 4128*a^
3*b^13*c^9*d^11*f^4 - 3328*a^12*b^4*c^4*d^16*f^4 - 3328*a^4*b^12*c^16*d^4*f
^4 + 3264*a^13*b^3*c^5*d^15*f^4 + 3264*a^3*b^13*c^15*d^5*f^4 - 2480*a^14*b^
2*c^8*d^12*f^4 - 2480*a^2*b^14*c^12*d^8*f^4 + 2240*a^11*b^5*c^15*d^5*f^4 +
2240*a^5*b^11*c^5*d^15*f^4 - 2128*a^12*b^4*c^14*d^6*f^4 - 2128*a^4*b^12*c^6
*d^14*f^4 + 2112*a^9*b^7*c^3*d^17*f^4 + 2112*a^7*b^9*c^17*d^3*f^4 + 2048*a^
11*b^5*c^3*d^17*f^4 + 2048*a^5*b^11*c^17*d^3*f^4 - 2000*a^14*b^2*c^6*d^14*f
^4 - 2000*a^2*b^14*c^14*d^6*f^4 - 1792*a^10*b^6*c^16*d^4*f^4 - 1792*a^6*b^1
0*c^4*d^16*f^4 - 1776*a^14*b^2*c^10*d^10*f^4 - 1776*a^2*b^14*c^10*d^10*f^4
+ 1472*a^13*b^3*c^13*d^7*f^4 + 1472*a^3*b^13*c^7*d^13*f^4 + 1088*a^9*b^7*c^
17*d^3*f^4 + 1088*a^7*b^9*c^3*d^17*f^4 + 992*a^13*b^3*c^3*d^17*f^4 + 992*a^
3*b^13*c^17*d^3*f^4 - 912*a^14*b^2*c^4*d^16*f^4 - 912*a^2*b^14*c^16*d^4*f^4
 - 768*a^10*b^6*c^2*d^18*f^4 - 768*a^6*b^10*c^18*d^2*f^4 - 688*a^14*b^2*c^1
2*d^8*f^4 - 688*a^2*b^14*c^8*d^12*f^4 - 592*a^12*b^4*c^2*d^18*f^4 - 592*a^4
*b^12*c^18*d^2*f^4 - 472*a^8*b^8*c^18*d^2*f^4 - 472*a^8*b^8*c^2*d^18*f^4 -
280*a^12*b^4*c^16*d^4*f^4 - 280*a^4*b^12*c^4*d^16*f^4 + 224*a^13*b^3*c^15*d
^5*f^4 + 224*a^11*b^5*c^17*d^3*f^4 + 224*a^5*b^11*c^3*d^17*f^4 + 224*a^3*b^
13*c^5*d^15*f^4 - 208*a^14*b^2*c^2*d^18*f^4 - 208*a^2*b^14*c^18*d^2*f^4 - 1
12*a^14*b^2*c^14*d^6*f^4 - 112*a^10*b^6*c^18*d^2*f^4 - 112*a^6*b^10*c^2*d^1
8*f^4 - 112*a^2*b^14*c^6*d^14*f^4 - 80*b^16*c^14*d^6*f^4 - 60*b^16*c^16*d^4

*f^4 - 60*b^16*c^12*d^8*f^4 - 24*b^16*c^18*d^2*f^4 - 24*b^16*c^10*d^10*f^4
- 4*b^16*c^8*d^12*f^4 - 80*a^16*c^6*d^14*f^4 - 60*a^16*c^8*d^12*f^4 - 60*a^
16*c^4*d^16*f^4 - 24*a^16*c^10*d^10*f^4 - 24*a^16*c^2*d^18*f^4 - 4*a^16*c^1
2*d^8*f^4 - 24*a^12*b^4*d^20*f^4 - 16*a^14*b^2*d^20*f^4 - 16*a^10*b^6*d^20*
f^4 - 4*a^8*b^8*d^20*f^4 - 24*a^4*b^12*c^20*f^4 - 16*a^6*b^10*c^20*f^4 - 16
*a^2*b^14*c^20*f^4 - 4*a^8*b^8*c^20*f^4 - 4*b^16*c^20*f^4 - 4*a^16*d^20*f^4
+ 56*A*C*a*b^11*c^13*d*f^2 - 48*A*C*a^11*b*c*d^13*f^2 + 48*A*C*a*b^11*c*d^
13*f^2 + 5904*B*C*a^6*b^6*c^7*d^7*f^2 - 5016*B*C*a^5*b^7*c^8*d^6*f^2 - 4608
*B*C*a^7*b^5*c^6*d^8*f^2 - 4512*B*C*a^5*b^7*c^6*d^8*f^2 - 4384*B*C*a^7*b^5*
c^8*d^6*f^2 + 3056*B*C*a^8*b^4*c^7*d^7*f^2 + 2256*B*C*a^4*b^8*c^7*d^7*f^2 -
1824*B*C*a^3*b^9*c^8*d^6*f^2 + 1632*B*C*a^9*b^3*c^4*d^10*f^2 - 1400*B*C*a^
8*b^4*c^3*d^11*f^2 - 1320*B*C*a^4*b^8*c^11*d^3*f^2 - 1248*B*C*a^3*b^9*c^6*d
^8*f^2 + 1152*B*C*a^3*b^9*c^10*d^4*f^2 - 1072*B*C*a^9*b^3*c^6*d^8*f^2 + 106
8*B*C*a^6*b^6*c^9*d^5*f^2 - 1004*B*C*a^4*b^8*c^5*d^9*f^2 - 968*B*C*a^6*b^6*
c^3*d^11*f^2 - 864*B*C*a^8*b^4*c^5*d^9*f^2 - 828*B*C*a^4*b^8*c^9*d^5*f^2 -
792*B*C*a^4*b^8*c^3*d^11*f^2 - 792*B*C*a^2*b^10*c^11*d^3*f^2 - 776*B*C*a^9*
b^3*c^8*d^6*f^2 + 688*B*C*a^7*b^5*c^4*d^10*f^2 - 672*B*C*a^10*b^2*c^3*d^11*
f^2 - 592*B*C*a^2*b^10*c^9*d^5*f^2 + 544*B*C*a^10*b^2*c^7*d^7*f^2 - 492*B*C
*a^2*b^10*c^5*d^9*f^2 + 480*B*C*a^5*b^7*c^10*d^4*f^2 - 392*B*C*a^10*b^2*c^5
*d^9*f^2 + 332*B*C*a^8*b^4*c^9*d^5*f^2 - 328*B*C*a^6*b^6*c^11*d^3*f^2 + 320
*B*C*a^9*b^3*c^2*d^12*f^2 + 272*B*C*a^3*b^9*c^12*d^2*f^2 - 248*B*C*a^5*b^7*
c^4*d^10*f^2 - 248*B*C*a^2*b^10*c^3*d^11*f^2 - 208*B*C*a^7*b^5*c^10*d^4*f^2
- 192*B*C*a^5*b^7*c^2*d^12*f^2 + 144*B*C*a^2*b^10*c^7*d^7*f^2 - 96*B*C*a^3
*b^9*c^4*d^10*f^2 + 88*B*C*a^5*b^7*c^12*d^2*f^2 - 72*B*C*a^8*b^4*c^11*d^3*f
^2 + 48*B*C*a^9*b^3*c^10*d^4*f^2 - 48*B*C*a^7*b^5*c^12*d^2*f^2 - 48*B*C*a^7
*b^5*c^2*d^12*f^2 - 48*B*C*a^3*b^9*c^2*d^12*f^2 - 12*B*C*a^10*b^2*c^9*d^5*f
^2 + 4*B*C*a^6*b^6*c^5*d^9*f^2 + 5824*A*C*a^7*b^5*c^5*d^9*f^2 - 4378*A*C*a^
8*b^4*c^6*d^8*f^2 + 4296*A*C*a^5*b^7*c^5*d^9*f^2 - 3912*A*C*a^6*b^6*c^6*d^8
*f^2 - 3672*A*C*a^5*b^7*c^9*d^5*f^2 + 3594*A*C*a^4*b^8*c^8*d^6*f^2 + 3236*A
*C*a^6*b^6*c^8*d^6*f^2 + 2816*A*C*a^9*b^3*c^5*d^9*f^2 + 2624*A*C*a^3*b^9*c^
5*d^9*f^2 + 2432*A*C*a^7*b^5*c^7*d^7*f^2 - 2366*A*C*a^8*b^4*c^4*d^10*f^2 +
2298*A*C*a^4*b^8*c^10*d^4*f^2 + 1872*A*C*a^3*b^9*c^7*d^7*f^2 + 1848*A*C*a^6
*b^6*c^10*d^4*f^2 - 1644*A*C*a^6*b^6*c^4*d^10*f^2 - 1488*A*C*a^7*b^5*c^9*d^
5*f^2 - 1408*A*C*a^3*b^9*c^9*d^5*f^2 - 1308*A*C*a^4*b^8*c^6*d^8*f^2 + 1248*
A*C*a^5*b^7*c^7*d^7*f^2 - 1012*A*C*a^10*b^2*c^6*d^8*f^2 + 1008*A*C*a^7*b^5*
c^3*d^11*f^2 + 992*A*C*a^5*b^7*c^3*d^11*f^2 + 928*A*C*a^3*b^9*c^3*d^11*f^2
+ 848*A*C*a^9*b^3*c^7*d^7*f^2 + 636*A*C*a^2*b^10*c^8*d^6*f^2 - 628*A*C*a^10
*b^2*c^4*d^10*f^2 - 600*A*C*a^2*b^10*c^6*d^8*f^2 - 576*A*C*a^5*b^7*c^11*d^3
*f^2 + 572*A*C*a^2*b^10*c^10*d^4*f^2 + 464*A*C*a^8*b^4*c^8*d^6*f^2 + 304*A*
C*a^6*b^6*c^2*d^12*f^2 - 304*A*C*a^4*b^8*c^4*d^10*f^2 + 296*A*C*a^4*b^8*c^2
*d^12*f^2 + 260*A*C*a^8*b^4*c^10*d^4*f^2 - 232*A*C*a^9*b^3*c^9*d^5*f^2 - 23
2*A*C*a^2*b^10*c^12*d^2*f^2 + 228*A*C*a^10*b^2*c^2*d^12*f^2 - 188*A*C*a^2*b
^10*c^4*d^10*f^2 + 144*A*C*a^3*b^9*c^11*d^3*f^2 + 116*A*C*a^6*b^6*c^12*d^2*
f^2 + 112*A*C*a^9*b^3*c^3*d^11*f^2 - 112*A*C*a^7*b^5*c^11*d^3*f^2 + 92*A*C*
a^10*b^2*c^8*d^6*f^2 + 74*A*C*a^4*b^8*c^12*d^2*f^2 + 62*A*C*a^8*b^4*c^2*d^1

2*f^2 + 40*A*C*a^2*b^10*c^2*d^12*f^2 - 7008*A*B*a^6*b^6*c^7*d^7*f^2 - 4032*
A*B*a^4*b^8*c^7*d^7*f^2 + 3952*A*B*a^7*b^5*c^8*d^6*f^2 + 3648*A*B*a^5*b^7*c
^8*d^6*f^2 - 3392*A*B*a^8*b^4*c^7*d^7*f^2 + 3264*A*B*a^7*b^5*c^6*d^8*f^2 -
2992*A*B*a^5*b^7*c^4*d^10*f^2 - 2368*A*B*a^7*b^5*c^4*d^10*f^2 - 2304*A*B*a^
3*b^9*c^4*d^10*f^2 - 1968*A*B*a^6*b^6*c^9*d^5*f^2 - 1872*A*B*a^9*b^3*c^4*d^
10*f^2 - 1728*A*B*a^2*b^10*c^7*d^7*f^2 + 1712*A*B*a^8*b^4*c^3*d^11*f^2 + 15
36*A*B*a^5*b^7*c^6*d^8*f^2 - 1536*A*B*a^3*b^9*c^10*d^4*f^2 - 1392*A*B*a^5*b
^7*c^2*d^12*f^2 + 1328*A*B*a^6*b^6*c^3*d^11*f^2 - 1104*A*B*a^3*b^9*c^2*d^12
*f^2 - 1056*A*B*a^3*b^9*c^6*d^8*f^2 + 976*A*B*a^9*b^3*c^6*d^8*f^2 + 960*A*B
*a^4*b^8*c^11*d^3*f^2 + 936*A*B*a^8*b^4*c^5*d^9*f^2 - 912*A*B*a^5*b^7*c^10*
d^4*f^2 + 848*A*B*a^9*b^3*c^8*d^6*f^2 - 816*A*B*a^7*b^5*c^2*d^12*f^2 + 816*
A*B*a^4*b^8*c^3*d^11*f^2 + 768*A*B*a^10*b^2*c^3*d^11*f^2 + 672*A*B*a^3*b^9*
c^8*d^6*f^2 - 632*A*B*a^8*b^4*c^9*d^5*f^2 - 608*A*B*a^2*b^10*c^9*d^5*f^2 -
552*A*B*a^4*b^8*c^9*d^5*f^2 - 544*A*B*a^10*b^2*c^7*d^7*f^2 - 480*A*B*a^2*b^
10*c^5*d^9*f^2 + 464*A*B*a^10*b^2*c^5*d^9*f^2 - 464*A*B*a^9*b^3*c^2*d^12*f^
2 + 432*A*B*a^2*b^10*c^11*d^3*f^2 - 368*A*B*a^3*b^9*c^12*d^2*f^2 - 256*A*B*
a^6*b^6*c^5*d^9*f^2 - 208*A*B*a^5*b^7*c^12*d^2*f^2 + 176*A*B*a^4*b^8*c^5*d^
9*f^2 + 112*A*B*a^7*b^5*c^10*d^4*f^2 + 112*A*B*a^6*b^6*c^11*d^3*f^2 - 16*A*
B*a^2*b^10*c^3*d^11*f^2 - 576*B*C*a*b^11*c^8*d^6*f^2 + 400*B*C*a^11*b*c^4*d
^10*f^2 - 288*B*C*a*b^11*c^6*d^8*f^2 - 176*B*C*a^11*b*c^6*d^8*f^2 + 128*B*C
*a*b^11*c^10*d^4*f^2 - 108*B*C*a^4*b^8*c*d^13*f^2 - 104*B*C*a*b^11*c^4*d^10
*f^2 - 92*B*C*a^4*b^8*c^13*d*f^2 - 60*B*C*a^8*b^4*c*d^13*f^2 - 60*B*C*a^6*b
^6*c*d^13*f^2 + 48*B*C*a^11*b*c^2*d^12*f^2 - 40*B*C*a^2*b^10*c*d^13*f^2 - 2
8*B*C*a^2*b^10*c^13*d*f^2 - 24*B*C*a*b^11*c^12*d^2*f^2 + 20*B*C*a^10*b^2*c*
d^13*f^2 - 16*B*C*a*b^11*c^2*d^12*f^2 + 12*B*C*a^6*b^6*c^13*d*f^2 + 912*A*C
*a*b^11*c^7*d^7*f^2 + 808*A*C*a*b^11*c^5*d^9*f^2 + 432*A*C*a^11*b*c^5*d^9*f
^2 + 336*A*C*a*b^11*c^3*d^11*f^2 + 224*A*C*a*b^11*c^11*d^3*f^2 - 112*A*C*a^
11*b*c^3*d^11*f^2 + 112*A*C*a^3*b^9*c*d^13*f^2 - 88*A*C*a^9*b^3*c*d^13*f^2
+ 80*A*C*a^3*b^9*c^13*d*f^2 + 56*A*C*a^5*b^7*c*d^13*f^2 + 48*A*C*a*b^11*c^9
*d^5*f^2 - 40*A*C*a^5*b^7*c^13*d*f^2 - 16*A*C*a^11*b*c^7*d^7*f^2 + 16*A*C*a
^7*b^5*c*d^13*f^2 - 496*A*B*a*b^11*c^4*d^10*f^2 - 400*A*B*a^11*b*c^4*d^10*f
^2 + 288*A*B*a*b^11*c^8*d^6*f^2 - 288*A*B*a*b^11*c^6*d^8*f^2 - 272*A*B*a*b^
11*c^2*d^12*f^2 + 240*A*B*a^6*b^6*c*d^13*f^2 - 224*A*B*a*b^11*c^10*d^4*f^2
+ 192*A*B*a^8*b^4*c*d^13*f^2 + 192*A*B*a^4*b^8*c*d^13*f^2 + 176*A*B*a^11*b*
c^6*d^8*f^2 + 104*A*B*a^4*b^8*c^13*d*f^2 - 48*A*B*a^11*b*c^2*d^12*f^2 + 16*
A*B*a^10*b^2*c*d^13*f^2 + 16*A*B*a^2*b^10*c^13*d*f^2 + 16*A*B*a^2*b^10*c*d^
13*f^2 - 112*B*C*b^12*c^11*d^3*f^2 + 4*B*C*b^12*c^5*d^9*f^2 + 150*A*C*b^12*
c^10*d^4*f^2 - 80*B*C*a^12*c^3*d^11*f^2 + 66*A*C*b^12*c^8*d^6*f^2 - 30*A*C*
b^12*c^12*d^2*f^2 + 24*B*C*a^12*c^5*d^9*f^2 - 12*A*C*b^12*c^4*d^10*f^2 - 57
6*A*B*b^12*c^7*d^7*f^2 - 432*A*B*b^12*c^9*d^5*f^2 - 400*A*B*b^12*c^5*d^9*f^
2 - 144*A*B*b^12*c^3*d^11*f^2 - 96*B*C*a^7*b^5*d^14*f^2 - 72*B*C*a^5*b^7*d^
14*f^2 - 66*A*C*a^12*c^4*d^10*f^2 + 54*A*C*a^12*c^2*d^12*f^2 - 32*A*B*b^12*
c^11*d^3*f^2 - 24*B*C*a^9*b^3*d^14*f^2 - 16*B*C*a^3*b^9*d^14*f^2 + 2*A*C*a^
12*c^6*d^8*f^2 + 116*A*C*a^6*b^6*d^14*f^2 + 100*A*C*a^4*b^8*d^14*f^2 + 80*A
*B*a^12*c^3*d^11*f^2 + 24*A*C*a^2*b^10*d^14*f^2 - 24*A*B*a^12*c^5*d^9*f^2 +

22*A*C*a^8*b^4*d^14*f^2 + 16*B*C*a^3*b^9*c^14*f^2 + 8*A*C*a^10*b^2*d^14*f^2 - 192*A*B*a^5*b^7*d^14*f^2 - 176*A*B*a^3*b^9*d^14*f^2 - 48*A*B*a^7*b^5*d^14*f^2 - 28*A*C*a^2*b^10*c^14*f^2 + 2*A*C*a^4*b^8*c^14*f^2 - 16*A*B*a^3*b^9*c^14*f^2 + 2508*C^2*a^6*b^6*c^6*d^8*f^2 + 2376*C^2*a^5*b^7*c^9*d^5*f^2 + 2357*C^2*a^8*b^4*c^6*d^8*f^2 - 2048*C^2*a^7*b^5*c^5*d^9*f^2 + 1304*C^2*a^3*b^9*c^9*d^5*f^2 + 1303*C^2*a^8*b^4*c^4*d^10*f^2 + 1212*C^2*a^6*b^6*c^4*d^10*f^2 - 1203*C^2*a^4*b^8*c^8*d^6*f^2 - 1192*C^2*a^9*b^3*c^5*d^9*f^2 + 1062*C^2*a^4*b^8*c^6*d^8*f^2 + 984*C^2*a^7*b^5*c^9*d^5*f^2 - 952*C^2*a^6*b^6*c^8*d^6*f^2 + 768*C^2*a^5*b^7*c^7*d^7*f^2 - 681*C^2*a^4*b^8*c^10*d^4*f^2 - 672*C^2*a^5*b^7*c^5*d^9*f^2 - 480*C^2*a^6*b^6*c^10*d^4*f^2 + 458*C^2*a^10*b^2*c^6*d^8*f^2 - 448*C^2*a^7*b^5*c^7*d^7*f^2 + 422*C^2*a^4*b^8*c^4*d^10*f^2 + 372*C^2*a^2*b^10*c^6*d^8*f^2 + 360*C^2*a^5*b^7*c^11*d^3*f^2 + 312*C^2*a^3*b^9*c^7*d^7*f^2 + 278*C^2*a^10*b^2*c^4*d^10*f^2 - 232*C^2*a^9*b^3*c^7*d^7*f^2 + 194*C^2*a^2*b^10*c^12*d^2*f^2 + 176*C^2*a^9*b^3*c^9*d^5*f^2 + 152*C^2*a^5*b^7*c^3*d^11*f^2 + 124*C^2*a^2*b^10*c^4*d^10*f^2 - 120*C^2*a^7*b^5*c^3*d^11*f^2 - 114*C^2*a^10*b^2*c^2*d^12*f^2 - 102*C^2*a^2*b^10*c^8*d^6*f^2 + 101*C^2*a^4*b^8*c^12*d^2*f^2 + 100*C^2*a^6*b^6*c^2*d^12*f^2 - 88*C^2*a^3*b^9*c^5*d^9*f^2 + 77*C^2*a^8*b^4*c^2*d^12*f^2 + 72*C^2*a^3*b^9*c^11*d^3*f^2 - 64*C^2*a^10*b^2*c^8*d^6*f^2 + 64*C^2*a^3*b^9*c^3*d^11*f^2 - 58*C^2*a^2*b^10*c^10*d^4*f^2 + 56*C^2*a^7*b^5*c^11*d^3*f^2 + 56*C^2*a^6*b^6*c^12*d^2*f^2 + 40*C^2*a^9*b^3*c^3*d^11*f^2 + 36*C^2*a^8*b^4*c^12*d^2*f^2 + 32*C^2*a^4*b^8*c^2*d^12*f^2 + 26*C^2*a^8*b^4*c^10*d^4*f^2 + 16*C^2*a^2*b^10*c^2*d^12*f^2 + 2*C^2*a^8*b^4*c^8*d^6*f^2 + 2277*B^2*a^4*b^8*c^8*d^6*f^2 + 2144*B^2*a^7*b^5*c^5*d^9*f^2 - 2112*B^2*a^5*b^7*c^9*d^5*f^2 + 2028*B^2*a^6*b^6*c^8*d^6*f^2 - 1671*B^2*a^8*b^4*c^6*d^8*f^2 + 1275*B^2*a^4*b^8*c^10*d^4*f^2 + 1176*B^2*a^5*b^7*c^5*d^9*f^2 + 1096*B^2*a^9*b^3*c^5*d^9*f^2 - 1044*B^2*a^6*b^6*c^6*d^8*f^2 + 984*B^2*a^6*b^6*c^10*d^4*f^2 - 968*B^2*a^3*b^9*c^9*d^5*f^2 - 888*B^2*a^7*b^5*c^9*d^5*f^2 + 672*B^2*a^7*b^5*c^7*d^7*f^2 + 664*B^2*a^3*b^9*c^5*d^9*f^2 - 649*B^2*a^8*b^4*c^4*d^10*f^2 + 618*B^2*a^2*b^10*c^8*d^6*f^2 + 514*B^2*a^4*b^8*c^4*d^10*f^2 + 460*B^2*a^6*b^6*c^2*d^12*f^2 + 422*B^2*a^8*b^4*c^8*d^6*f^2 + 406*B^2*a^2*b^10*c^10*d^4*f^2 - 382*B^2*a^10*b^2*c^6*d^8*f^2 + 368*B^2*a^4*b^8*c^2*d^12*f^2 - 312*B^2*a^5*b^7*c^11*d^3*f^2 + 312*B^2*a^3*b^9*c^7*d^7*f^2 + 248*B^2*a^9*b^3*c^7*d^7*f^2 + 245*B^2*a^8*b^4*c^2*d^12*f^2 - 192*B^2*a^5*b^7*c^7*d^7*f^2 - 184*B^2*a^9*b^3*c^3*d^11*f^2 + 182*B^2*a^10*b^2*c^2*d^12*f^2 + 176*B^2*a^3*b^9*c^3*d^11*f^2 + 174*B^2*a^4*b^8*c^6*d^8*f^2 - 170*B^2*a^10*b^2*c^4*d^10*f^2 - 152*B^2*a^9*b^3*c^9*d^5*f^2 + 152*B^2*a^2*b^10*c^4*d^10*f^2 + 142*B^2*a^8*b^4*c^10*d^4*f^2 - 90*B^2*a^2*b^10*c^12*d^2*f^2 + 88*B^2*a^2*b^10*c^2*d^12*f^2 + 84*B^2*a^10*b^2*c^8*d^6*f^2 + 84*B^2*a^2*b^10*c^6*d^8*f^2 + 60*B^2*a^6*b^6*c^12*d^2*f^2 - 56*B^2*a^7*b^5*c^11*d^3*f^2 + 53*B^2*a^4*b^8*c^12*d^2*f^2 + 24*B^2*a^7*b^5*c^3*d^11*f^2 + 24*B^2*a^6*b^6*c^4*d^10*f^2 + 24*B^2*a^3*b^9*c^11*d^3*f^2 - 8*B^2*a^5*b^7*c^3*d^11*f^2 + 4566*A^2*a^4*b^8*c^6*d^8*f^2 + 4284*A^2*a^6*b^6*c^6*d^8*f^2 - 3776*A^2*a^7*b^5*c^5*d^9*f^2 - 3624*A^2*a^5*b^7*c^5*d^9*f^2 + 3122*A^2*a^4*b^8*c^4*d^10*f^2 + 3108*A^2*a^2*b^10*c^6*d^8*f^2 + 2741*A^2*a^8*b^4*c^6*d^8*f^2 + 2592*A^2*a^6*b^6*c^4*d^10*f^2 - 2536*A^2*a^3*b^9*c^5*d^9*f^2 + 2224*

A^2*a^2*b^10*c^4*d^10*f^2 - 2184*A^2*a^3*b^9*c^7*d^7*f^2 - 2016*A^2*a^5*b^7
*c^7*d^7*f^2 - 1984*A^2*a^7*b^5*c^7*d^7*f^2 + 1626*A^2*a^2*b^10*c^8*d^6*f^2
 - 1624*A^2*a^9*b^3*c^5*d^9*f^2 + 1603*A^2*a^8*b^4*c^4*d^10*f^2 + 1296*A^2*
a^5*b^7*c^9*d^5*f^2 - 1144*A^2*a^5*b^7*c^3*d^11*f^2 - 992*A^2*a^3*b^9*c^3*d
^11*f^2 + 968*A^2*a^4*b^8*c^2*d^12*f^2 - 888*A^2*a^7*b^5*c^3*d^11*f^2 + 849
*A^2*a^4*b^8*c^8*d^6*f^2 + 808*A^2*a^2*b^10*c^2*d^12*f^2 - 616*A^2*a^9*b^3*
c^7*d^7*f^2 + 554*A^2*a^10*b^2*c^6*d^8*f^2 + 504*A^2*a^7*b^5*c^9*d^5*f^2 -
504*A^2*a^6*b^6*c^10*d^4*f^2 + 460*A^2*a^6*b^6*c^2*d^12*f^2 + 350*A^2*a^10*
b^2*c^4*d^10*f^2 + 350*A^2*a^2*b^10*c^10*d^4*f^2 - 321*A^2*a^4*b^8*c^10*d^4
*f^2 + 216*A^2*a^5*b^7*c^11*d^3*f^2 - 216*A^2*a^3*b^9*c^11*d^3*f^2 + 182*A^
2*a^2*b^10*c^12*d^2*f^2 - 152*A^2*a^9*b^3*c^3*d^11*f^2 - 124*A^2*a^6*b^6*c^
8*d^6*f^2 - 114*A^2*a^10*b^2*c^2*d^12*f^2 + 104*A^2*a^3*b^9*c^9*d^5*f^2 + 7
7*A^2*a^8*b^4*c^2*d^12*f^2 + 74*A^2*a^8*b^4*c^8*d^6*f^2 - 70*A^2*a^8*b^4*c^
10*d^4*f^2 + 56*A^2*a^9*b^3*c^9*d^5*f^2 + 56*A^2*a^7*b^5*c^11*d^3*f^2 + 41*
A^2*a^4*b^8*c^12*d^2*f^2 - 28*A^2*a^10*b^2*c^8*d^6*f^2 - 28*A^2*a^6*b^6*c^1
2*d^2*f^2 + 12*B*C*b^12*c^13*d*f^2 + 24*B*C*a^12*c*d^13*f^2 - 24*A*B*b^12*c
^13*d*f^2 - 24*A*B*b^12*c*d^13*f^2 - 16*B*C*a^11*b*d^14*f^2 - 24*A*B*a^12*c
*d^13*f^2 - 16*B*C*a*b^11*c^14*f^2 - 48*A*B*a*b^11*d^14*f^2 + 16*A*B*a^11*b
*d^14*f^2 + 16*A*B*a*b^11*c^14*f^2 - 216*C^2*a^11*b*c^5*d^9*f^2 + 216*C^2*a
*b^11*c^9*d^5*f^2 + 56*C^2*a^11*b*c^3*d^11*f^2 + 56*C^2*a^9*b^3*c*d^13*f^2
+ 56*C^2*a^5*b^7*c*d^13*f^2 + 40*C^2*a^7*b^5*c*d^13*f^2 - 40*C^2*a*b^11*c^1
1*d^3*f^2 + 32*C^2*a^5*b^7*c^13*d*f^2 - 24*C^2*a*b^11*c^7*d^7*f^2 - 16*C^2*
a^3*b^9*c^13*d*f^2 + 16*C^2*a^3*b^9*c*d^13*f^2 + 8*C^2*a^11*b*c^7*d^7*f^2 -
 8*C^2*a*b^11*c^5*d^9*f^2 + 264*B^2*a*b^11*c^7*d^7*f^2 + 224*B^2*a*b^11*c^5
*d^9*f^2 + 168*B^2*a^11*b*c^5*d^9*f^2 - 112*B^2*a^9*b^3*c*d^13*f^2 - 104*B^
2*a^11*b*c^3*d^11*f^2 - 104*B^2*a^7*b^5*c*d^13*f^2 + 96*B^2*a*b^11*c^3*d^11
*f^2 + 88*B^2*a*b^11*c^11*d^3*f^2 - 72*B^2*a*b^11*c^9*d^5*f^2 - 64*B^2*a^5*
b^7*c*d^13*f^2 + 32*B^2*a^3*b^9*c^13*d*f^2 - 24*B^2*a^11*b*c^7*d^7*f^2 - 24
*B^2*a^5*b^7*c^13*d*f^2 + 16*B^2*a^3*b^9*c*d^13*f^2 - 888*A^2*a*b^11*c^7*d^
7*f^2 - 800*A^2*a*b^11*c^5*d^9*f^2 - 336*A^2*a*b^11*c^3*d^11*f^2 - 264*A^2*
a*b^11*c^9*d^5*f^2 - 216*A^2*a^11*b*c^5*d^9*f^2 - 184*A^2*a*b^11*c^11*d^3*f
^2 - 128*A^2*a^3*b^9*c*d^13*f^2 - 112*A^2*a^5*b^7*c*d^13*f^2 - 64*A^2*a^3*b
^9*c^13*d*f^2 + 56*A^2*a^11*b*c^3*d^11*f^2 - 56*A^2*a^7*b^5*c*d^13*f^2 + 32
*A^2*a^9*b^3*c*d^13*f^2 + 8*A^2*a^11*b*c^7*d^7*f^2 + 8*A^2*a^5*b^7*c^13*d*f
^2 + 24*C^2*a^11*b*c*d^13*f^2 - 16*C^2*a*b^11*c^13*d*f^2 - 40*B^2*a^11*b*c*
d^13*f^2 + 24*B^2*a*b^11*c^13*d*f^2 + 16*B^2*a*b^11*c*d^13*f^2 - 48*A^2*a*b
^11*c*d^13*f^2 - 40*A^2*a*b^11*c^13*d*f^2 + 24*A^2*a^11*b*c*d^13*f^2 - 6*A*
C*a^12*d^14*f^2 + 2*A*C*b^12*c^14*f^2 + 33*C^2*b^12*c^12*d^2*f^2 - 27*C^2*b
^12*c^10*d^4*f^2 + 3*C^2*b^12*c^8*d^6*f^2 + 117*B^2*b^12*c^10*d^4*f^2 + 111
*B^2*b^12*c^8*d^6*f^2 + 72*B^2*b^12*c^6*d^8*f^2 + 33*C^2*a^12*c^4*d^10*f^2
- 27*C^2*a^12*c^2*d^12*f^2 + 24*B^2*b^12*c^4*d^10*f^2 + 4*B^2*b^12*c^2*d^12
*f^2 - 3*B^2*b^12*c^12*d^2*f^2 - C^2*a^12*c^6*d^8*f^2 + 720*A^2*b^12*c^6*d^
8*f^2 + 552*A^2*b^12*c^4*d^10*f^2 + 471*A^2*b^12*c^8*d^6*f^2 + 216*A^2*b^12
*c^2*d^12*f^2 + 93*A^2*b^12*c^10*d^4*f^2 + 33*B^2*a^12*c^2*d^12*f^2 + 33*A^
2*b^12*c^12*d^2*f^2 + 31*C^2*a^8*b^4*d^14*f^2 - 27*B^2*a^12*c^4*d^10*f^2 +

20*C^2*a^6*b^6*d^14*f^2 + 4*C^2*a^4*b^8*d^14*f^2 + 3*B^2*a^12*c^6*d^8*f^2 +
 2*C^2*a^10*b^2*d^14*f^2 + 80*B^2*a^6*b^6*d^14*f^2 + 64*B^2*a^4*b^8*d^14*f^
2 + 33*A^2*a^12*c^4*d^10*f^2 + 31*B^2*a^8*b^4*d^14*f^2 - 27*A^2*a^12*c^2*d^
12*f^2 + 16*B^2*a^2*b^10*d^14*f^2 + 14*C^2*a^2*b^10*c^14*f^2 + 14*B^2*a^10*
b^2*d^14*f^2 - C^2*a^4*b^8*c^14*f^2 - A^2*a^12*c^6*d^8*f^2 + 120*A^2*a^2*b^
10*d^14*f^2 + 112*A^2*a^4*b^8*d^14*f^2 - 17*A^2*a^8*b^4*d^14*f^2 - 10*B^2*a
^2*b^10*c^14*f^2 - 10*A^2*a^10*b^2*d^14*f^2 + 8*A^2*a^6*b^6*d^14*f^2 + 3*B^
2*a^4*b^8*c^14*f^2 + 14*A^2*a^2*b^10*c^14*f^2 - A^2*a^4*b^8*c^14*f^2 + 3*C^
2*a^12*d^14*f^2 - C^2*b^12*c^14*f^2 + 36*A^2*b^12*d^14*f^2 + 3*B^2*b^12*c^1
4*f^2 - B^2*a^12*d^14*f^2 + 3*A^2*a^12*d^14*f^2 - A^2*b^12*c^14*f^2 - 44*A*
B*C*a*b^9*c^10*d*f + 3816*A*B*C*a^5*b^5*c^4*d^7*f + 2920*A*B*C*a^2*b^8*c^5*
d^6*f - 2736*A*B*C*a^3*b^7*c^6*d^5*f - 2672*A*B*C*a^4*b^6*c^3*d^8*f + 1996*
A*B*C*a^4*b^6*c^7*d^4*f - 1412*A*B*C*a^6*b^4*c^5*d^6*f + 1120*A*B*C*a^3*b^7
*c^2*d^9*f + 1080*A*B*C*a^2*b^8*c^7*d^4*f + 1040*A*B*C*a^5*b^5*c^2*d^9*f +
684*A*B*C*a^4*b^6*c^5*d^6*f + 592*A*B*C*a^3*b^7*c^4*d^7*f - 560*A*B*C*a^7*b
^3*c^2*d^9*f - 448*A*B*C*a^2*b^8*c^3*d^8*f - 400*A*B*C*a^5*b^5*c^8*d^3*f -
398*A*B*C*a^2*b^8*c^9*d^2*f - 312*A*B*C*a^6*b^4*c^3*d^8*f + 166*A*B*C*a^8*b
^2*c^3*d^8*f + 136*A*B*C*a^5*b^5*c^6*d^5*f + 128*A*B*C*a^7*b^3*c^6*d^5*f -
100*A*B*C*a^6*b^4*c^7*d^4*f + 64*A*B*C*a^7*b^3*c^4*d^7*f - 64*A*B*C*a^4*b^6
*c^9*d^2*f - 32*A*B*C*a^3*b^7*c^8*d^3*f - 16*A*B*C*a^8*b^2*c^5*d^6*f - 1312
*A*B*C*a*b^9*c^4*d^7*f + 996*A*B*C*a*b^9*c^8*d^3*f + 728*A*B*C*a^6*b^4*c*d^
10*f - 624*A*B*C*a*b^9*c^6*d^5*f - 584*A*B*C*a^2*b^8*c*d^10*f - 512*A*B*C*a
^4*b^6*c*d^10*f - 320*A*B*C*a*b^9*c^2*d^9*f - 98*A*B*C*a^8*b^2*c*d^10*f + 3
6*A*B*C*a^9*b*c^2*d^9*f + 32*A*B*C*a^3*b^7*c^10*d*f - 16*A*B*C*a^9*b*c^4*d^
7*f + 46*B*C^2*a*b^9*c^10*d*f - 16*B^2*C*a*b^9*c*d^10*f - 2*B^2*C*a^9*b*c*d
^10*f + 312*A^2*C*a*b^9*c*d^10*f - 48*A*C^2*a*b^9*c*d^10*f - 6*A^2*C*a^9*b*
c*d^10*f + 6*A*C^2*a^9*b*c*d^10*f + 208*A*B^2*a*b^9*c*d^10*f - 2*A^2*B*a*b^
9*c^10*d*f + 2*A*B^2*a^9*b*c*d^10*f - 480*A*B*C*b^10*c^7*d^4*f + 78*A*B*C*b
^10*c^9*d^2*f - 64*A*B*C*b^10*c^5*d^6*f + 2*A*B*C*a^10*c^3*d^8*f - 224*A*B*
C*a^5*b^5*d^11*f + 80*A*B*C*a^7*b^3*d^11*f - 32*A*B*C*a^3*b^7*d^11*f + 2*A*
B*C*a^2*b^8*c^11*f - 1692*B*C^2*a^5*b^5*c^4*d^7*f - 1500*B^2*C*a^5*b^5*c^5*
d^6*f - 1464*B^2*C*a^3*b^7*c^5*d^6*f + 1426*B*C^2*a^6*b^4*c^5*d^6*f - 1158*
B^2*C*a^6*b^4*c^4*d^7*f + 1152*B*C^2*a^3*b^7*c^6*d^5*f + 1026*B^2*C*a^4*b^6
*c^6*d^5*f - 974*B*C^2*a^4*b^6*c^7*d^4*f + 960*B^2*C*a^5*b^5*c^3*d^8*f - 88
4*B*C^2*a^2*b^8*c^5*d^6*f - 764*B^2*C*a^5*b^5*c^7*d^4*f + 752*B^2*C*a^2*b^8
*c^4*d^7*f - 752*B*C^2*a^3*b^7*c^4*d^7*f + 738*B^2*C*a^4*b^6*c^4*d^7*f - 68
8*B^2*C*a^6*b^4*c^2*d^9*f - 675*B^2*C*a^2*b^8*c^8*d^3*f + 560*B*C^2*a^5*b^5
*c^8*d^3*f + 496*B*C^2*a^7*b^3*c^2*d^9*f + 496*B*C^2*a^4*b^6*c^3*d^8*f - 46
8*B*C^2*a^2*b^8*c^7*d^4*f + 456*B^2*C*a^7*b^3*c^3*d^8*f - 452*B^2*C*a^4*b^6
*c^8*d^3*f - 416*B*C^2*a^3*b^7*c^2*d^9*f + 378*B*C^2*a^4*b^6*c^5*d^6*f + 37
6*B*C^2*a^3*b^7*c^8*d^3*f - 360*B^2*C*a^2*b^8*c^6*d^5*f + 355*B*C^2*a^2*b^8
*c^9*d^2*f + 346*B^2*C*a^6*b^4*c^6*d^5*f - 320*B^2*C*a^4*b^6*c^2*d^9*f + 26
8*B^2*C*a^2*b^8*c^2*d^9*f + 216*B^2*C*a^3*b^7*c^7*d^4*f - 203*B*C^2*a^8*b^2
*c^3*d^8*f - 184*B*C^2*a^7*b^3*c^6*d^5*f + 170*B*C^2*a^6*b^4*c^7*d^4*f + 16
0*B^2*C*a^7*b^3*c^5*d^6*f - 160*B*C^2*a^5*b^5*c^2*d^9*f - 140*B^2*C*a^8*b^2

*c^4*d^7*f - 136*B*C^2*a^2*b^8*c^3*d^8*f + 112*B^2*C*a^3*b^7*c^9*d^2*f + 91
*B^2*C*a^8*b^2*c^2*d^9*f + 88*B*C^2*a^7*b^3*c^4*d^7*f + 72*B^2*C*a^6*b^4*c^
8*d^3*f - 64*B^2*C*a^3*b^7*c^3*d^8*f - 60*B*C^2*a^6*b^4*c^3*d^8*f + 56*B*C^
2*a^4*b^6*c^9*d^2*f + 52*B*C^2*a^5*b^5*c^6*d^5*f - 48*B^2*C*a^7*b^3*c^7*d^4
*f + 48*B^2*C*a^5*b^5*c^9*d^2*f + 44*B*C^2*a^8*b^2*c^5*d^6*f - 36*B*C^2*a^6
*b^4*c^9*d^2*f + 12*B^2*C*a^8*b^2*c^6*d^5*f - 2958*A^2*C*a^4*b^6*c^4*d^7*f
- 1932*A^2*C*a^2*b^8*c^4*d^7*f + 1848*A^2*C*a^3*b^7*c^5*d^6*f + 1728*A^2*C*
a^3*b^7*c^3*d^8*f + 1524*A^2*C*a^5*b^5*c^5*d^6*f + 1374*A*C^2*a^4*b^6*c^4*d
^7*f - 1272*A*C^2*a^3*b^7*c^5*d^6*f - 1236*A*C^2*a^5*b^5*c^5*d^6*f + 1116*A
*C^2*a^2*b^8*c^4*d^7*f - 1110*A^2*C*a^4*b^6*c^6*d^5*f + 1038*A*C^2*a^4*b^6*
c^6*d^5*f - 768*A^2*C*a^2*b^8*c^2*d^9*f - 696*A^2*C*a^3*b^7*c^7*d^4*f - 666
*A*C^2*a^6*b^4*c^4*d^7*f + 564*A^2*C*a^2*b^8*c^6*d^5*f - 564*A*C^2*a^5*b^5*
c^7*d^4*f - 555*A*C^2*a^2*b^8*c^8*d^3*f + 519*A^2*C*a^2*b^8*c^8*d^3*f - 480
*A*C^2*a^3*b^7*c^3*d^8*f + 456*A*C^2*a^5*b^5*c^3*d^8*f - 420*A*C^2*a^6*b^4*
c^2*d^9*f + 408*A*C^2*a^3*b^7*c^7*d^4*f + 408*A*C^2*a^2*b^8*c^2*d^9*f + 348
*A^2*C*a^6*b^4*c^2*d^9*f - 348*A*C^2*a^2*b^8*c^6*d^5*f + 342*A*C^2*a^6*b^4*
c^6*d^5*f - 336*A*C^2*a^4*b^6*c^8*d^3*f + 324*A^2*C*a^5*b^5*c^7*d^4*f - 312
*A^2*C*a^4*b^6*c^2*d^9*f + 264*A^2*C*a^4*b^6*c^8*d^3*f + 240*A*C^2*a^7*b^3*
c^5*d^6*f + 195*A*C^2*a^8*b^2*c^2*d^9*f - 174*A^2*C*a^6*b^4*c^6*d^5*f + 144
*A*C^2*a^3*b^7*c^9*d^2*f - 123*A^2*C*a^8*b^2*c^2*d^9*f + 120*A*C^2*a^7*b^3*
c^3*d^8*f + 108*A*C^2*a^6*b^4*c^8*d^3*f - 102*A^2*C*a^6*b^4*c^4*d^7*f - 96*
A^2*C*a^8*b^2*c^4*d^7*f + 72*A^2*C*a^7*b^3*c^3*d^8*f + 72*A*C^2*a^5*b^5*c^9
*d^2*f + 48*A^2*C*a^7*b^3*c^5*d^6*f - 48*A^2*C*a^3*b^7*c^9*d^2*f - 48*A*C^2
*a^4*b^6*c^2*d^9*f - 24*A^2*C*a^5*b^5*c^3*d^8*f - 12*A*C^2*a^8*b^2*c^4*d^7*
f + 2736*A^2*B*a^3*b^7*c^6*d^5*f + 2464*A^2*B*a^4*b^6*c^3*d^8*f - 2298*A*B^
2*a^4*b^6*c^4*d^7*f - 2252*A^2*B*a^2*b^8*c^5*d^6*f - 1692*A^2*B*a^5*b^5*c^4
*d^7*f - 1592*A*B^2*a^2*b^8*c^4*d^7*f - 1338*A*B^2*a^4*b^6*c^6*d^5*f + 1320
*A*B^2*a^3*b^7*c^5*d^6*f + 1212*A*B^2*a^5*b^5*c^5*d^6*f - 1056*A*B^2*a^5*b^
5*c^3*d^8*f + 1024*A^2*B*a^3*b^7*c^4*d^7*f - 1022*A^2*B*a^4*b^6*c^7*d^4*f -
880*A^2*B*a^5*b^5*c^2*d^9*f - 846*A^2*B*a^4*b^6*c^5*d^6*f - 840*A*B^2*a^3*
b^7*c^7*d^4*f + 760*A*B^2*a^6*b^4*c^2*d^9*f - 704*A^2*B*a^3*b^7*c^2*d^9*f +
688*A*B^2*a^3*b^7*c^3*d^8*f + 660*A^2*B*a^6*b^4*c^3*d^8*f - 612*A^2*B*a^2*
b^8*c^7*d^4*f + 462*A*B^2*a^6*b^4*c^4*d^7*f + 459*A*B^2*a^2*b^8*c^8*d^3*f -
412*A*B^2*a^2*b^8*c^2*d^9*f - 408*A*B^2*a^7*b^3*c^3*d^8*f + 388*A^2*B*a^5*
b^5*c^6*d^5*f + 296*A^2*B*a^2*b^8*c^3*d^8*f + 288*A*B^2*a^2*b^8*c^6*d^5*f +
284*A*B^2*a^5*b^5*c^7*d^4*f + 236*A*B^2*a^4*b^6*c^8*d^3*f - 226*A*B^2*a^6*
b^4*c^6*d^5*f + 212*A*B^2*a^4*b^6*c^2*d^9*f + 202*A^2*B*a^6*b^4*c^5*d^6*f -
152*A^2*B*a^7*b^3*c^4*d^7*f + 88*A^2*B*a^3*b^7*c^8*d^3*f + 79*A^2*B*a^2*b^
8*c^9*d^2*f - 70*A^2*B*a^6*b^4*c^7*d^4*f + 68*A*B^2*a^8*b^2*c^4*d^7*f + 64*
A^2*B*a^7*b^3*c^2*d^9*f - 64*A*B^2*a^3*b^7*c^9*d^2*f + 56*A^2*B*a^7*b^3*c^6
*d^5*f + 56*A^2*B*a^5*b^5*c^8*d^3*f + 37*A^2*B*a^8*b^2*c^3*d^8*f - 28*A^2*B
*a^8*b^2*c^5*d^6*f - 28*A^2*B*a^4*b^6*c^9*d^2*f + 17*A*B^2*a^8*b^2*c^2*d^9*
f - 16*A*B^2*a^7*b^3*c^5*d^6*f + 24*A*B*C*b^10*c*d^10*f - 6*A*B*C*a^10*c*d^
10*f + 48*A*B*C*a*b^9*d^11*f + 4*A*B*C*a^9*b*d^11*f + 432*B^2*C*a*b^9*c^7*d
^4*f - 376*B*C^2*a^6*b^4*c*d^10*f - 354*B*C^2*a*b^9*c^8*d^3*f + 352*B^2*C*a

^5*b^5*c*d^10*f + 320*B^2*C*a*b^9*c^5*d^6*f + 256*B^2*C*a^3*b^7*c*d^10*f -
232*B^2*C*a^7*b^3*c*d^10*f - 210*B^2*C*a*b^9*c^9*d^2*f - 152*B*C^2*a^4*b^6*
c*d^10*f + 85*B*C^2*a^8*b^2*c*d^10*f + 72*B^2*C*a*b^9*c^3*d^8*f - 48*B*C^2*
a*b^9*c^6*d^5*f - 40*B*C^2*a^3*b^7*c^10*d*f + 40*B*C^2*a^2*b^8*c*d^10*f + 3
7*B^2*C*a^2*b^8*c^10*d*f + 22*B^2*C*a^9*b*c^3*d^8*f - 18*B*C^2*a^9*b*c^2*d^
9*f + 16*B*C^2*a*b^9*c^2*d^9*f - 12*B^2*C*a^4*b^6*c^10*d*f + 8*B*C^2*a^9*b*
c^4*d^7*f + 8*B*C^2*a*b^9*c^4*d^7*f - 984*A^2*C*a*b^9*c^7*d^4*f + 672*A^2*C
*a*b^9*c^3*d^8*f + 552*A*C^2*a*b^9*c^7*d^4*f - 504*A^2*C*a^5*b^5*c*d^10*f -
 408*A^2*C*a*b^9*c^5*d^6*f + 408*A*C^2*a*b^9*c^5*d^6*f + 336*A*C^2*a^5*b^5*
c*d^10*f - 216*A*C^2*a^7*b^3*c*d^10*f + 192*A*C^2*a^3*b^7*c*d^10*f - 162*A*
C^2*a*b^9*c^9*d^2*f + 120*A^2*C*a^7*b^3*c*d^10*f + 96*A^2*C*a^3*b^7*c*d^10*
f + 90*A^2*C*a*b^9*c^9*d^2*f + 66*A^2*C*a^9*b*c^3*d^8*f - 66*A*C^2*a^9*b*c^
3*d^8*f + 57*A*C^2*a^2*b^8*c^10*d*f - 48*A*C^2*a*b^9*c^3*d^8*f - 9*A^2*C*a^
2*b^8*c^10*d*f + 1736*A^2*B*a*b^9*c^4*d^7*f + 1248*A^2*B*a*b^9*c^6*d^5*f -
1008*A*B^2*a*b^9*c^7*d^4*f + 772*A^2*B*a^4*b^6*c*d^10*f - 688*A*B^2*a^5*b^5
*c*d^10*f - 608*A*B^2*a*b^9*c^5*d^6*f + 436*A^2*B*a^2*b^8*c*d^10*f - 426*A^
2*B*a*b^9*c^8*d^3*f + 312*A*B^2*a*b^9*c^3*d^8*f + 304*A^2*B*a*b^9*c^2*d^9*f
 - 244*A^2*B*a^6*b^4*c*d^10*f - 160*A*B^2*a^3*b^7*c*d^10*f + 114*A*B^2*a*b^
9*c^9*d^2*f + 88*A*B^2*a^7*b^3*c*d^10*f - 22*A*B^2*a^9*b*c^3*d^8*f - 18*A^2
*B*a^9*b*c^2*d^9*f + 13*A^2*B*a^8*b^2*c*d^10*f - 13*A*B^2*a^2*b^8*c^10*d*f
+ 8*A^2*B*a^9*b*c^4*d^7*f + 8*A^2*B*a^3*b^7*c^10*d*f + 111*B^2*C*b^10*c^8*d
^3*f - 39*B*C^2*b^10*c^9*d^2*f + 24*B*C^2*b^10*c^7*d^4*f - 4*B^2*C*b^10*c^2
*d^9*f - 4*B*C^2*b^10*c^5*d^6*f + 432*A^2*C*b^10*c^6*d^5*f + 192*A^2*C*b^10
*c^4*d^7*f - 111*A^2*C*b^10*c^8*d^3*f + 111*A*C^2*b^10*c^8*d^3*f - 72*A*C^2
*b^10*c^6*d^5*f + 12*A*C^2*b^10*c^4*d^7*f - 3*B^2*C*a^10*c^2*d^9*f - B*C^2*
a^10*c^3*d^8*f + 456*A^2*B*b^10*c^7*d^4*f - 288*A^2*B*b^10*c^3*d^8*f + 252*
A*B^2*b^10*c^6*d^5*f + 192*A*B^2*b^10*c^4*d^7*f - 183*A*B^2*b^10*c^8*d^3*f
- 148*A^2*B*b^10*c^5*d^6*f + 112*B^2*C*a^6*b^4*d^11*f + 76*A*B^2*b^10*c^2*d
^9*f - 64*B*C^2*a^7*b^3*d^11*f + 16*B^2*C*a^4*b^6*d^11*f - 16*B^2*C*a^2*b^8
*d^11*f + 16*B*C^2*a^5*b^5*d^11*f + 16*B*C^2*a^3*b^7*d^11*f - 9*A^2*C*a^10*
c^2*d^9*f + 9*A*C^2*a^10*c^2*d^9*f - 3*A^2*B*b^10*c^9*d^2*f - B^2*C*a^8*b^2
*d^11*f + 96*A^2*C*a^4*b^6*d^11*f - 84*A^2*C*a^6*b^4*d^11*f + 72*A*C^2*a^6*
b^4*d^11*f - 24*A*C^2*a^4*b^6*d^11*f - 24*A*C^2*a^2*b^8*d^11*f - 21*A*C^2*a
^8*b^2*d^11*f + 12*A^2*C*a^2*b^8*d^11*f + 9*A^2*C*a^8*b^2*d^11*f + 3*A*B^2*
a^10*c^2*d^9*f - A^2*B*a^10*c^3*d^8*f - B*C^2*a^2*b^8*c^11*f + 176*A*B^2*a^
4*b^6*d^11*f + 136*A^2*B*a^5*b^5*d^11*f - 128*A^2*B*a^3*b^7*d^11*f + 112*A*
B^2*a^2*b^8*d^11*f - 64*A*B^2*a^6*b^4*d^11*f - 16*A^2*B*a^7*b^3*d^11*f - A^
2*B*a^2*b^8*c^11*f - 2*C^3*a^9*b*c*d^10*f - 2*B^3*a*b^9*c^10*d*f - 264*A^3*
a*b^9*c*d^10*f + 2*A^3*a^9*b*c*d^10*f - 9*B^2*C*b^10*c^10*d*f + 9*A^2*C*b^1
0*c^10*d*f - 9*A*C^2*b^10*c^10*d*f + 3*B*C^2*a^10*c*d^10*f - 132*A^2*B*b^10
*c*d^10*f - 3*A*B^2*b^10*c^10*d*f - 2*B*C^2*a^9*b*d^11*f + 3*A^2*B*a^10*c*d
^10*f - 2*B^2*C*a*b^9*c^11*f - 120*A^2*B*a*b^9*d^11*f - 6*A^2*C*a*b^9*c^11*
f + 6*A*C^2*a*b^9*c^11*f - 2*A^2*B*a^9*b*d^11*f + 2*A*B^2*a*b^9*c^11*f + 52
0*C^3*a^3*b^7*c^5*d^6*f + 460*C^3*a^5*b^5*c^5*d^6*f - 418*C^3*a^4*b^6*c^6*d
^5*f + 406*C^3*a^6*b^4*c^4*d^7*f + 268*C^3*a^5*b^5*c^7*d^4*f - 266*C^3*a^6*

b^4*c^6*d^5*f + 233*C^3*a^2*b^8*c^8*d^3*f - 176*C^3*a^7*b^3*c^5*d^6*f + 164 *C^3*a^6*b^4*c^2*d^9*f + 140*C^3*a^2*b^8*c^6*d^5*f + 136*C^3*a^4*b^6*c^2*d^ 9*f - 128*C^3*a^3*b^7*c^9*d^2*f + 128*C^3*a^3*b^7*c^3*d^8*f - 108*C^3*a^6*b ^4*c^8*d^3*f - 104*C^3*a^7*b^3*c^3*d^8*f - 104*C^3*a^5*b^5*c^3*d^8*f + 100* C^3*a^4*b^6*c^8*d^3*f - 89*C^3*a^8*b^2*c^2*d^9*f - 72*C^3*a^5*b^5*c^9*d^2*f + 40*C^3*a^8*b^2*c^4*d^7*f - 40*C^3*a^3*b^7*c^7*d^4*f - 28*C^3*a^2*b^8*c^4 *d^7*f - 16*C^3*a^2*b^8*c^2*d^9*f - 2*C^3*a^4*b^6*c^4*d^7*f + 828*B^3*a^5*b ^5*c^4*d^7*f + 408*B^3*a^2*b^8*c^5*d^6*f + 390*B^3*a^4*b^6*c^7*d^4*f - 372* B^3*a^4*b^6*c^3*d^8*f - 336*B^3*a^3*b^7*c^6*d^5*f - 314*B^3*a^6*b^4*c^5*d^6 *f + 288*B^3*a^3*b^7*c^4*d^7*f + 216*B^3*a^2*b^8*c^7*d^4*f - 176*B^3*a^7*b^ 3*c^2*d^9*f + 128*B^3*a^3*b^7*c^2*d^9*f + 108*B^3*a^5*b^5*c^6*d^5*f + 88*B^ 3*a^7*b^3*c^4*d^7*f + 72*B^3*a^5*b^5*c^2*d^9*f - 68*B^3*a^2*b^8*c^3*d^8*f - 65*B^3*a^2*b^8*c^9*d^2*f - 56*B^3*a^5*b^5*c^8*d^3*f + 40*B^3*a^7*b^3*c^6*d ^5*f + 37*B^3*a^8*b^2*c^3*d^8*f + 30*B^3*a^4*b^6*c^5*d^6*f - 28*B^3*a^8*b^2 *c^5*d^6*f + 24*B^3*a^3*b^7*c^8*d^3*f - 4*B^3*a^4*b^6*c^9*d^2*f - 2*B^3*a^6 *b^4*c^7*d^4*f + 1586*A^3*a^4*b^6*c^4*d^7*f - 1376*A^3*a^3*b^7*c^3*d^8*f - 1096*A^3*a^3*b^7*c^5*d^6*f + 844*A^3*a^2*b^8*c^4*d^7*f - 748*A^3*a^5*b^5*c^ 5*d^6*f + 490*A^3*a^4*b^6*c^6*d^5*f + 376*A^3*a^2*b^8*c^2*d^9*f + 362*A^3*a ^6*b^4*c^4*d^7*f - 356*A^3*a^2*b^8*c^6*d^5*f - 328*A^3*a^5*b^5*c^3*d^8*f + 328*A^3*a^3*b^7*c^7*d^4*f + 224*A^3*a^4*b^6*c^2*d^9*f - 197*A^3*a^2*b^8*c^8 *d^3*f - 112*A^3*a^7*b^3*c^5*d^6*f + 98*A^3*a^6*b^4*c^6*d^5*f - 92*A^3*a^6* b^4*c^2*d^9*f - 88*A^3*a^7*b^3*c^3*d^8*f + 68*A^3*a^8*b^2*c^4*d^7*f + 32*A^ 3*a^3*b^7*c^9*d^2*f - 28*A^3*a^5*b^5*c^7*d^4*f - 28*A^3*a^4*b^6*c^8*d^3*f + 17*A^3*a^8*b^2*c^2*d^9*f + 104*C^3*a^7*b^3*c*d^10*f + 54*C^3*a*b^9*c^9*d^2 *f - 40*C^3*a*b^9*c^7*d^4*f - 35*C^3*a^2*b^8*c^10*d*f + 22*C^3*a^9*b*c^3*d^ 8*f + 16*C^3*a^5*b^5*c*d^10*f - 16*C^3*a^3*b^7*c*d^10*f + 8*C^3*a*b^9*c^5*d ^6*f - 2*A*B*C*b^10*c^11*f + 198*B^3*a*b^9*c^8*d^3*f + 192*B^3*a^6*b^4*c*d^ 10*f - 128*B^3*a*b^9*c^4*d^7*f - 80*B^3*a^2*b^8*c*d^10*f - 56*B^3*a*b^9*c^2 *d^9*f - 24*B^3*a*b^9*c^6*d^5*f - 18*B^3*a^9*b*c^2*d^9*f - 16*B^3*a^4*b^6*c *d^10*f + 13*B^3*a^8*b^2*c*d^10*f + 8*B^3*a^9*b*c^4*d^7*f + 8*B^3*a^3*b^7*c ^10*d*f - 624*A^3*a*b^9*c^3*d^8*f + 472*A^3*a*b^9*c^7*d^4*f - 272*A^3*a^3*b ^7*c*d^10*f + 152*A^3*a^5*b^5*c*d^10*f - 22*A^3*a^9*b*c^3*d^8*f + 18*A^3*a* b^9*c^9*d^2*f - 13*A^3*a^2*b^8*c^10*d*f - 8*A^3*a^7*b^3*c*d^10*f - 8*A^3*a* b^9*c^5*d^6*f + A*B^2*a^8*b^2*d^11*f - C^3*b^10*c^8*d^3*f - 60*B^3*b^10*c^7 *d^4*f - 32*B^3*b^10*c^5*d^6*f + 21*B^3*b^10*c^9*d^2*f - 12*B^3*b^10*c^3*d^ 8*f - 3*C^3*a^10*c^2*d^9*f - 360*A^3*b^10*c^6*d^5*f - 204*A^3*b^10*c^4*d^7* f + 11*C^3*a^8*b^2*d^11*f - 8*C^3*a^6*b^4*d^11*f - 4*C^3*a^4*b^6*d^11*f - B ^3*a^10*c^3*d^8*f - 64*B^3*a^5*b^5*d^11*f - 32*B^3*a^3*b^7*d^11*f + 3*A^3*a ^10*c^2*d^9*f - 68*A^3*a^4*b^6*d^11*f + 20*A^3*a^6*b^4*d^11*f + 12*A^3*a^2* b^8*d^11*f - B^3*a^2*b^8*c^11*f + 3*C^3*b^10*c^10*d*f + 3*B^3*a^10*c*d^10*f - 3*A^3*b^10*c^10*d*f - 2*C^3*a*b^9*c^11*f - 2*B^3*a^9*b*d^11*f + 2*A^3*a* b^9*c^11*f - 36*A^2*C*b^10*d^11*f + 3*A^2*C*a^10*d^11*f - 3*A*C^2*a^10*d^11 *f - A*B^2*a^10*d^11*f + 36*A^3*b^10*d^11*f - A^3*a^10*d^11*f + A^3*b^10*c^ 8*d^3*f + A^3*a^8*b^2*d^11*f + B^2*C*a^10*d^11*f + B*C^2*b^10*c^11*f + A^2* B*b^10*c^11*f + C^3*a^10*d^11*f + B^3*b^10*c^11*f - 6*A*B^2*C*a*b^7*c^7*d +

4*A*B^2*C*a*b^7*c*d^7 + 168*A^2*B*C*a^3*b^5*c^2*d^6 + 144*A*B*C^2*a^4*b^4*
c^3*d^5 - 129*A^2*B*C*a^4*b^4*c^3*d^5 - 96*A*B*C^2*a^3*b^5*c^2*d^6 + 84*A*B
*C^2*a^2*b^6*c^3*d^5 + 72*A^2*B*C*a^3*b^5*c^4*d^4 - 72*A^2*B*C*a^2*b^6*c^3*
d^5 + 64*A*B^2*C*a^4*b^4*c^4*d^4 - 60*A*B*C^2*a^3*b^5*c^4*d^4 + 57*A^2*B*C*
a^2*b^6*c^5*d^3 - 56*A*B^2*C*a^3*b^5*c^5*d^3 - 39*A*B^2*C*a^4*b^4*c^2*d^6 -
38*A*B^2*C*a^5*b^3*c^3*d^5 + 36*A*B^2*C*a^3*b^5*c^3*d^5 + 36*A*B*C^2*a^4*b
^4*c^5*d^3 - 30*A*B*C^2*a^2*b^6*c^5*d^3 + 27*A*B^2*C*a^2*b^6*c^6*d^2 - 24*A
*B^2*C*a^2*b^6*c^2*d^6 - 24*A*B*C^2*a^5*b^3*c^4*d^4 + 24*A*B*C^2*a^3*b^5*c^
6*d^2 + 18*A^2*B*C*a^5*b^3*c^2*d^6 - 18*A^2*B*C*a^4*b^4*c^5*d^3 - 15*A*B^2*
C*a^2*b^6*c^4*d^4 + 12*A^2*B*C*a^5*b^3*c^4*d^4 - 12*A^2*B*C*a^3*b^5*c^6*d^2
+ 9*A*B^2*C*a^6*b^2*c^2*d^6 + 6*A*B*C^2*a^6*b^2*c^3*d^5 - 3*A^2*B*C*a^6*b^
2*c^3*d^5 + 60*A^2*B*C*a*b^7*c^2*d^6 - 51*A^2*B*C*a^4*b^4*c*d^7 + 48*A*B*C^
2*a*b^7*c^6*d^2 - 42*A^2*B*C*a^2*b^6*c*d^7 - 42*A^2*B*C*a*b^7*c^6*d^2 + 36*
A*B*C^2*a^4*b^4*c*d^7 + 36*A*B*C^2*a^2*b^6*c*d^7 + 36*A*B*C^2*a*b^7*c^4*d^4
- 30*A^2*B*C*a*b^7*c^4*d^4 + 24*A*B^2*C*a*b^7*c^3*d^5 - 24*A*B*C^2*a*b^7*c
^2*d^6 + 18*A*B^2*C*a^5*b^3*c*d^7 - 18*A*B*C^2*a^6*b^2*c*d^7 + 12*A*B^2*C*a
^3*b^5*c*d^7 + 9*A^2*B*C*a^6*b^2*c*d^7 + 6*A*B^2*C*a*b^7*c^5*d^3 - 6*A*B*C^
2*a^2*b^6*c^7*d + 3*A^2*B*C*a^2*b^6*c^7*d - 18*B^3*C*a*b^7*c^6*d^2 - 18*B*C
^3*a*b^7*c^6*d^2 - 14*B^3*C*a*b^7*c^4*d^4 - 14*B*C^3*a*b^7*c^4*d^4 - 10*B^3
*C*a^2*b^6*c*d^7 - 10*B*C^3*a^2*b^6*c*d^7 + 9*B^3*C*a^6*b^2*c*d^7 + 9*B*C^3
*a^6*b^2*c*d^7 - 7*B^3*C*a^4*b^4*c*d^7 - 7*B*C^3*a^4*b^4*c*d^7 + 6*B^2*C^2*
a*b^7*c^7*d - 4*B^3*C*a*b^7*c^2*d^6 + 4*B^2*C^2*a*b^7*c*d^7 - 4*B*C^3*a*b^7
*c^2*d^6 + 3*B^3*C*a^2*b^6*c^7*d + 3*B*C^3*a^2*b^6*c^7*d + 144*A^3*C*a*b^7*
c^3*d^5 + 62*A^3*C*a*b^7*c^5*d^3 + 48*A*C^3*a*b^7*c^3*d^5 - 36*A^2*C^2*a*b^
7*c*d^7 + 26*A*C^3*a*b^7*c^5*d^3 + 20*A^3*C*a^3*b^5*c*d^7 + 18*A^2*C^2*a*b^
7*c^7*d - 18*A*C^3*a^5*b^3*c*d^7 - 6*A^3*C*a^5*b^3*c*d^7 - 4*A*C^3*a^3*b^5*
c*d^7 - 32*A^3*B*a*b^7*c^2*d^6 - 32*A*B^3*a*b^7*c^2*d^6 + 22*A^3*B*a^4*b^4*
c*d^7 + 22*A*B^3*a^4*b^4*c*d^7 + 16*A^3*B*a^2*b^6*c*d^7 + 16*A*B^3*a^2*b^6*
c*d^7 + 12*A^3*B*a*b^7*c^6*d^2 + 12*A*B^3*a*b^7*c^6*d^2 + 8*A^3*B*a*b^7*c^4
*d^4 - 8*A^2*B^2*a*b^7*c*d^7 + 8*A*B^3*a*b^7*c^4*d^4 + 57*A^2*B*C*b^8*c^5*d
^3 + 36*A^2*B*C*b^8*c^3*d^5 - 30*A*B*C^2*b^8*c^5*d^3 - 18*A*B*C^2*b^8*c^3*d
^5 - 9*A*B^2*C*b^8*c^4*d^4 - 3*A*B^2*C*b^8*c^6*d^2 - 2*A*B^2*C*b^8*c^2*d^6
+ 36*A^2*B*C*a^3*b^5*d^8 + 24*A*B*C^2*a^5*b^3*d^8 - 18*A^2*B*C*a^5*b^3*d^8
- 12*A*B*C^2*a^3*b^5*d^8 - 3*A*B^2*C*a^6*b^2*d^8 - 3*A*B^2*C*a^4*b^4*d^8 -
2*A*B^2*C*a^2*b^6*d^8 + 34*B^2*C^2*a^5*b^3*c^3*d^5 + 28*B^2*C^2*a^3*b^5*c^5
*d^3 + 24*B^2*C^2*a^4*b^4*c^2*d^6 - 20*B^2*C^2*a^4*b^4*c^4*d^4 + 12*B^2*C^2
*a^3*b^5*c^3*d^5 + 12*B^2*C^2*a^2*b^6*c^2*d^6 - 9*B^2*C^2*a^6*b^2*c^2*d^6 +
9*B^2*C^2*a^4*b^4*c^6*d^2 + 9*B^2*C^2*a^2*b^6*c^4*d^4 - 3*B^2*C^2*a^2*b^6*
c^6*d^2 + 159*A^2*C^2*a^2*b^6*c^4*d^4 - 156*A^2*C^2*a^3*b^5*c^3*d^5 + 90*A^
2*C^2*a^5*b^3*c^3*d^5 + 78*A^2*C^2*a^2*b^6*c^2*d^6 - 63*A^2*C^2*a^4*b^4*c^4
*d^4 - 27*A^2*C^2*a^6*b^2*c^2*d^6 - 27*A^2*C^2*a^2*b^6*c^6*d^2 - 18*A^2*C^2
*a^4*b^4*c^2*d^6 + 9*A^2*C^2*a^4*b^4*c^6*d^2 + 66*A^2*B^2*a^2*b^6*c^2*d^6 +
60*A^2*B^2*a^2*b^6*c^4*d^4 - 48*A^2*B^2*a^3*b^5*c^3*d^5 + 42*A^2*B^2*a^4*b
^4*c^2*d^6 + 28*A^2*B^2*a^3*b^5*c^5*d^3 - 17*A^2*B^2*a^4*b^4*c^4*d^4 - 6*A^
2*B^2*a^2*b^6*c^6*d^2 + 4*A^2*B^2*a^5*b^3*c^3*d^5 + 36*A^3*C*a*b^7*c*d^7 -

18*A*C^3*a*b^7*c^7*d + 12*A*C^3*a*b^7*c*d^7 - 6*A^3*C*a*b^7*c^7*d + 12*A^2*
B*C*b^8*c*d^7 + 6*A*B*C^2*b^8*c^7*d - 6*A*B*C^2*b^8*c*d^7 - 3*A^2*B*C*b^8*c
^7*d + 24*A^2*B*C*a*b^7*d^8 - 12*A*B*C^2*a*b^7*d^8 - 53*B^3*C*a^4*b^4*c^3*d
^5 - 53*B*C^3*a^4*b^4*c^3*d^5 - 32*B^3*C*a^2*b^6*c^3*d^5 - 32*B*C^3*a^2*b^6
*c^3*d^5 - 18*B^3*C*a^4*b^4*c^5*d^3 - 18*B*C^3*a^4*b^4*c^5*d^3 + 16*B^3*C*a
^3*b^5*c^4*d^4 + 16*B*C^3*a^3*b^5*c^4*d^4 + 12*B^3*C*a^5*b^3*c^4*d^4 - 12*B
^3*C*a^3*b^5*c^6*d^2 + 12*B^2*C^2*a*b^7*c^3*d^5 + 12*B*C^3*a^5*b^3*c^4*d^4
- 12*B*C^3*a^3*b^5*c^6*d^2 + 8*B^3*C*a^3*b^5*c^2*d^6 + 8*B*C^3*a^3*b^5*c^2*
d^6 - 6*B^3*C*a^5*b^3*c^2*d^6 - 6*B^2*C^2*a^5*b^3*c*d^7 + 6*B^2*C^2*a*b^7*c
^5*d^3 - 6*B*C^3*a^5*b^3*c^2*d^6 - 3*B^3*C*a^6*b^2*c^3*d^5 - 3*B*C^3*a^6*b^
2*c^3*d^5 - 175*A^3*C*a^2*b^6*c^4*d^4 + 164*A^3*C*a^3*b^5*c^3*d^5 - 144*A^2
*C^2*a*b^7*c^3*d^5 - 124*A^3*C*a^2*b^6*c^2*d^6 - 90*A*C^3*a^5*b^3*c^3*d^5 -
73*A*C^3*a^2*b^6*c^4*d^4 - 66*A^2*C^2*a*b^7*c^5*d^3 + 44*A*C^3*a^3*b^5*c^3
*d^5 + 36*A*C^3*a^4*b^4*c^4*d^4 - 30*A^3*C*a^5*b^3*c^3*d^5 + 30*A^3*C*a^4*b
^4*c^4*d^4 + 27*A*C^3*a^6*b^2*c^2*d^6 + 21*A*C^3*a^4*b^4*c^2*d^6 + 18*A^2*C
^2*a^5*b^3*c*d^7 - 18*A*C^3*a^4*b^4*c^6*d^2 - 16*A*C^3*a^2*b^6*c^2*d^6 - 15
*A^3*C*a^4*b^4*c^2*d^6 + 15*A^3*C*a^2*b^6*c^6*d^2 - 12*A^2*C^2*a^3*b^5*c*d^
7 + 9*A^3*C*a^6*b^2*c^2*d^6 + 9*A*C^3*a^2*b^6*c^6*d^2 - 80*A^3*B*a^3*b^5*c^
2*d^6 - 80*A*B^3*a^3*b^5*c^2*d^6 + 38*A^3*B*a^4*b^4*c^3*d^5 + 38*A*B^3*a^4*
b^4*c^3*d^5 - 36*A^2*B^2*a*b^7*c^3*d^5 - 28*A^3*B*a^3*b^5*c^4*d^4 - 28*A^3*
B*a^2*b^6*c^5*d^3 - 28*A*B^3*a^3*b^5*c^4*d^4 - 28*A*B^3*a^2*b^6*c^5*d^3 + 2
0*A^3*B*a^2*b^6*c^3*d^5 + 20*A*B^3*a^2*b^6*c^3*d^5 - 12*A^3*B*a^5*b^3*c^2*d
^6 - 12*A^2*B^2*a^5*b^3*c*d^7 - 12*A^2*B^2*a^3*b^5*c*d^7 - 12*A^2*B^2*a*b^7
*c^5*d^3 - 12*A*B^3*a^5*b^3*c^2*d^6 + 6*B^2*C^2*b^8*c^6*d^2 + 3*B^2*C^2*b^8
*c^4*d^4 + 36*A^2*C^2*b^8*c^4*d^4 + 27*A^2*C^2*b^8*c^2*d^6 - 18*A^2*C^2*b^8
*c^6*d^2 + 33*A^2*B^2*b^8*c^4*d^4 + 28*A^2*B^2*b^8*c^2*d^6 + 9*B^2*C^2*a^4*
b^4*d^8 + 6*A^2*B^2*b^8*c^6*d^2 + 4*B^2*C^2*a^2*b^6*d^8 + 3*B^2*C^2*a^6*b^2
*d^8 - 30*A^2*C^2*a^4*b^4*d^8 + 9*A^2*C^2*a^6*b^2*d^8 + 16*A^2*B^2*a^2*b^6*
d^8 + 3*A^2*B^2*a^4*b^4*d^8 + 6*C^4*a^5*b^3*c*d^7 + 4*C^4*a^3*b^5*c*d^7 - 2
*C^4*a*b^7*c^5*d^3 - 12*B^4*a^5*b^3*c*d^7 + 12*B^4*a*b^7*c^3*d^5 + 8*B^4*a*
b^7*c^5*d^3 - 4*B^4*a^3*b^5*c*d^7 - 48*A^4*a*b^7*c^3*d^5 - 20*A^4*a*b^7*c^5
*d^3 - 8*A^4*a^3*b^5*c*d^7 - 63*A^3*C*b^8*c^4*d^4 - 54*A^3*C*b^8*c^2*d^6 +
9*A^3*C*b^8*c^6*d^2 + 9*A*C^3*b^8*c^6*d^2 - 3*A*C^3*b^8*c^4*d^4 - 28*A^3*B*
b^8*c^5*d^3 - 28*A*B^3*b^8*c^5*d^3 - 18*A^3*B*b^8*c^3*d^5 - 18*A*B^3*b^8*c^
3*d^5 - 10*B^3*C*a^5*b^3*d^8 - 10*B*C^3*a^5*b^3*d^8 - 4*B^3*C*a^3*b^5*d^8 -
4*B*C^3*a^3*b^5*d^8 + 23*A^3*C*a^4*b^4*d^8 - 18*A^3*C*a^2*b^6*d^8 + 11*A*C
^3*a^4*b^4*d^8 - 9*A*C^3*a^6*b^2*d^8 + 6*A*C^3*a^2*b^6*d^8 - 3*A^3*C*a^6*b^
2*d^8 - 20*A^3*B*a^3*b^5*d^8 - 20*A*B^3*a^3*b^5*d^8 + 4*A^3*B*a^5*b^3*d^8 +
4*A*B^3*a^5*b^3*d^8 + B^3*C*a^2*b^6*c^5*d^3 + B*C^3*a^2*b^6*c^5*d^3 + 6*C^
4*a*b^7*c^7*d + 4*B^4*a*b^7*c*d^7 - 12*A^4*a*b^7*c*d^7 - 3*B^3*C*b^8*c^7*d
- 3*B*C^3*b^8*c^7*d - 6*A^3*B*b^8*c*d^7 - 6*A*B^3*b^8*c*d^7 - 12*A^3*B*a*b^
7*d^8 - 12*A*B^3*a*b^7*d^8 + 30*C^4*a^5*b^3*c^3*d^5 + 19*C^4*a^2*b^6*c^4*d^
4 - 9*C^4*a^6*b^2*c^2*d^6 + 9*C^4*a^4*b^4*c^6*d^2 + 4*C^4*a^3*b^5*c^3*d^5 +
4*C^4*a^2*b^6*c^2*d^6 - 3*C^4*a^4*b^4*c^4*d^4 - 3*C^4*a^4*b^4*c^2*d^6 + 3*
C^4*a^2*b^6*c^6*d^2 + 28*B^4*a^3*b^5*c^5*d^3 + 27*B^4*a^4*b^4*c^2*d^6 - 17*

B^4*a^4*b^4*c^4*d^4 - 10*B^4*a^2*b^6*c^4*d^4 + 8*B^4*a^3*b^5*c^3*d^5 + 8*B^
4*a^2*b^6*c^2*d^6 - 6*B^4*a^2*b^6*c^6*d^2 + 4*B^4*a^5*b^3*c^3*d^5 + 70*A^4*
a^2*b^6*c^4*d^4 + 58*A^4*a^2*b^6*c^2*d^6 - 56*A^4*a^3*b^5*c^3*d^5 + 15*A^4*
a^4*b^4*c^2*d^6 + B^2*C^2*b^8*c^2*d^6 - 18*A^3*C*b^8*d^8 + B^3*C*b^8*c^5*d^
3 + B*C^3*b^8*c^5*d^3 + 6*B^4*b^8*c^6*d^2 + 3*B^4*b^8*c^4*d^4 + 30*A^4*b^8*
c^4*d^4 + 27*A^4*b^8*c^2*d^6 + 3*C^4*a^6*b^2*d^8 + 8*B^4*a^4*b^4*d^8 + 4*B^
4*a^2*b^6*d^8 + 12*A^4*a^2*b^6*d^8 - 5*A^4*a^4*b^4*d^8 + 9*A^2*C^2*b^8*d^8
+ 9*A^2*B^2*b^8*d^8 + 9*A^4*b^8*d^8 + B^4*b^8*c^2*d^6 + C^4*a^4*b^4*d^8, f,
k)*(root(640*a^15*b*c^7*d^13*f^4 + 640*a*b^15*c^13*d^7*f^4 + 480*a^15*b*c^
9*d^11*f^4 + 480*a^15*b*c^5*d^15*f^4 + 480*a*b^15*c^15*d^5*f^4 + 480*a*b^15
*c^11*d^9*f^4 + 192*a^15*b*c^11*d^9*f^4 + 192*a^15*b*c^3*d^17*f^4 + 192*a^1
1*b^5*c*d^19*f^4 + 192*a^5*b^11*c^19*d*f^4 + 192*a*b^15*c^17*d^3*f^4 + 192*
a*b^15*c^9*d^11*f^4 + 128*a^13*b^3*c*d^19*f^4 + 128*a^9*b^7*c*d^19*f^4 + 12
8*a^7*b^9*c^19*d*f^4 + 128*a^3*b^13*c^19*d*f^4 + 32*a^15*b*c^13*d^7*f^4 + 3
2*a^9*b^7*c^19*d*f^4 + 32*a^7*b^9*c*d^19*f^4 + 32*a*b^15*c^7*d^13*f^4 + 32*
a^15*b*c*d^19*f^4 + 32*a*b^15*c^19*d*f^4 - 47088*a^8*b^8*c^10*d^10*f^4 + 42
432*a^9*b^7*c^9*d^11*f^4 + 42432*a^7*b^9*c^11*d^9*f^4 + 39328*a^9*b^7*c^11*
d^9*f^4 + 39328*a^7*b^9*c^9*d^11*f^4 - 36912*a^8*b^8*c^12*d^8*f^4 - 36912*a
^8*b^8*c^8*d^12*f^4 - 34256*a^10*b^6*c^10*d^10*f^4 - 34256*a^6*b^10*c^10*d^
10*f^4 - 31152*a^10*b^6*c^8*d^12*f^4 - 31152*a^6*b^10*c^12*d^8*f^4 + 28128*
a^9*b^7*c^7*d^13*f^4 + 28128*a^7*b^9*c^13*d^7*f^4 + 24160*a^11*b^5*c^9*d^11
*f^4 + 24160*a^5*b^11*c^11*d^9*f^4 - 23088*a^10*b^6*c^12*d^8*f^4 - 23088*a^
6*b^10*c^8*d^12*f^4 + 22272*a^9*b^7*c^13*d^7*f^4 + 22272*a^7*b^9*c^7*d^13*f
^4 + 19072*a^11*b^5*c^11*d^9*f^4 + 19072*a^5*b^11*c^9*d^11*f^4 + 18624*a^11
*b^5*c^7*d^13*f^4 + 18624*a^5*b^11*c^13*d^7*f^4 - 17328*a^8*b^8*c^14*d^6*f^
4 - 17328*a^8*b^8*c^6*d^14*f^4 - 17232*a^10*b^6*c^6*d^14*f^4 - 17232*a^6*b^
10*c^14*d^6*f^4 - 13520*a^12*b^4*c^8*d^12*f^4 - 13520*a^4*b^12*c^12*d^8*f^4
- 12464*a^12*b^4*c^10*d^10*f^4 - 12464*a^4*b^12*c^10*d^10*f^4 + 10880*a^9*
b^7*c^5*d^15*f^4 + 10880*a^7*b^9*c^15*d^5*f^4 - 9072*a^10*b^6*c^14*d^6*f^4
- 9072*a^6*b^10*c^6*d^14*f^4 + 8928*a^11*b^5*c^13*d^7*f^4 + 8928*a^5*b^11*c
^7*d^13*f^4 - 8880*a^12*b^4*c^6*d^14*f^4 - 8880*a^4*b^12*c^14*d^6*f^4 + 848
0*a^11*b^5*c^5*d^15*f^4 + 8480*a^5*b^11*c^15*d^5*f^4 + 7200*a^9*b^7*c^15*d^
5*f^4 + 7200*a^7*b^9*c^5*d^15*f^4 - 6912*a^12*b^4*c^12*d^8*f^4 - 6912*a^4*b
^12*c^8*d^12*f^4 + 6400*a^13*b^3*c^9*d^11*f^4 + 6400*a^3*b^13*c^11*d^9*f^4
+ 5920*a^13*b^3*c^7*d^13*f^4 + 5920*a^3*b^13*c^13*d^7*f^4 - 5392*a^10*b^6*c
^4*d^16*f^4 - 5392*a^6*b^10*c^16*d^4*f^4 - 4428*a^8*b^8*c^16*d^4*f^4 - 4428
*a^8*b^8*c^4*d^16*f^4 + 4128*a^13*b^3*c^11*d^9*f^4 + 4128*a^3*b^13*c^9*d^11
*f^4 - 3328*a^12*b^4*c^4*d^16*f^4 - 3328*a^4*b^12*c^16*d^4*f^4 + 3264*a^13*
b^3*c^5*d^15*f^4 + 3264*a^3*b^13*c^15*d^5*f^4 - 2480*a^14*b^2*c^8*d^12*f^4
- 2480*a^2*b^14*c^12*d^8*f^4 + 2240*a^11*b^5*c^15*d^5*f^4 + 2240*a^5*b^11*c
^5*d^15*f^4 - 2128*a^12*b^4*c^14*d^6*f^4 - 2128*a^4*b^12*c^6*d^14*f^4 + 211
2*a^9*b^7*c^3*d^17*f^4 + 2112*a^7*b^9*c^17*d^3*f^4 + 2048*a^11*b^5*c^3*d^17
*f^4 + 2048*a^5*b^11*c^17*d^3*f^4 - 2000*a^14*b^2*c^6*d^14*f^4 - 2000*a^2*b
^14*c^14*d^6*f^4 - 1792*a^10*b^6*c^16*d^4*f^4 - 1792*a^6*b^10*c^4*d^16*f^4
- 1776*a^14*b^2*c^10*d^10*f^4 - 1776*a^2*b^14*c^10*d^10*f^4 + 1472*a^13*b^3

*c^13*d^7*f^4 + 1472*a^3*b^13*c^7*d^13*f^4 + 1088*a^9*b^7*c^17*d^3*f^4 + 10
88*a^7*b^9*c^3*d^17*f^4 + 992*a^13*b^3*c^3*d^17*f^4 + 992*a^3*b^13*c^17*d^3
*f^4 - 912*a^14*b^2*c^4*d^16*f^4 - 912*a^2*b^14*c^16*d^4*f^4 - 768*a^10*b^6
*c^2*d^18*f^4 - 768*a^6*b^10*c^18*d^2*f^4 - 688*a^14*b^2*c^12*d^8*f^4 - 688
*a^2*b^14*c^8*d^12*f^4 - 592*a^12*b^4*c^2*d^18*f^4 - 592*a^4*b^12*c^18*d^2*
f^4 - 472*a^8*b^8*c^18*d^2*f^4 - 472*a^8*b^8*c^2*d^18*f^4 - 280*a^12*b^4*c^
16*d^4*f^4 - 280*a^4*b^12*c^4*d^16*f^4 + 224*a^13*b^3*c^15*d^5*f^4 + 224*a^
11*b^5*c^17*d^3*f^4 + 224*a^5*b^11*c^3*d^17*f^4 + 224*a^3*b^13*c^5*d^15*f^4
 - 208*a^14*b^2*c^2*d^18*f^4 - 208*a^2*b^14*c^18*d^2*f^4 - 112*a^14*b^2*c^1
4*d^6*f^4 - 112*a^10*b^6*c^18*d^2*f^4 - 112*a^6*b^10*c^2*d^18*f^4 - 112*a^2
*b^14*c^6*d^14*f^4 - 80*b^16*c^14*d^6*f^4 - 60*b^16*c^16*d^4*f^4 - 60*b^16*
c^12*d^8*f^4 - 24*b^16*c^18*d^2*f^4 - 24*b^16*c^10*d^10*f^4 - 4*b^16*c^8*d^
12*f^4 - 80*a^16*c^6*d^14*f^4 - 60*a^16*c^8*d^12*f^4 - 60*a^16*c^4*d^16*f^4
 - 24*a^16*c^10*d^10*f^4 - 24*a^16*c^2*d^18*f^4 - 4*a^16*c^12*d^8*f^4 - 24*
a^12*b^4*d^20*f^4 - 16*a^14*b^2*d^20*f^4 - 16*a^10*b^6*d^20*f^4 - 4*a^8*b^8
*d^20*f^4 - 24*a^4*b^12*c^20*f^4 - 16*a^6*b^10*c^20*f^4 - 16*a^2*b^14*c^20*
f^4 - 4*a^8*b^8*c^20*f^4 - 4*b^16*c^20*f^4 - 4*a^16*d^20*f^4 + 56*A*C*a*b^1
1*c^13*d*f^2 - 48*A*C*a^11*b*c*d^13*f^2 + 48*A*C*a*b^11*c*d^13*f^2 + 5904*B
*C*a^6*b^6*c^7*d^7*f^2 - 5016*B*C*a^5*b^7*c^8*d^6*f^2 - 4608*B*C*a^7*b^5*c^
6*d^8*f^2 - 4512*B*C*a^5*b^7*c^6*d^8*f^2 - 4384*B*C*a^7*b^5*c^8*d^6*f^2 + 3
056*B*C*a^8*b^4*c^7*d^7*f^2 + 2256*B*C*a^4*b^8*c^7*d^7*f^2 - 1824*B*C*a^3*b
^9*c^8*d^6*f^2 + 1632*B*C*a^9*b^3*c^4*d^10*f^2 - 1400*B*C*a^8*b^4*c^3*d^11*
f^2 - 1320*B*C*a^4*b^8*c^11*d^3*f^2 - 1248*B*C*a^3*b^9*c^6*d^8*f^2 + 1152*B
*C*a^3*b^9*c^10*d^4*f^2 - 1072*B*C*a^9*b^3*c^6*d^8*f^2 + 1068*B*C*a^6*b^6*c
^9*d^5*f^2 - 1004*B*C*a^4*b^8*c^5*d^9*f^2 - 968*B*C*a^6*b^6*c^3*d^11*f^2 -
864*B*C*a^8*b^4*c^5*d^9*f^2 - 828*B*C*a^4*b^8*c^9*d^5*f^2 - 792*B*C*a^4*b^8
*c^3*d^11*f^2 - 792*B*C*a^2*b^10*c^11*d^3*f^2 - 776*B*C*a^9*b^3*c^8*d^6*f^2
 + 688*B*C*a^7*b^5*c^4*d^10*f^2 - 672*B*C*a^10*b^2*c^3*d^11*f^2 - 592*B*C*a
^2*b^10*c^9*d^5*f^2 + 544*B*C*a^10*b^2*c^7*d^7*f^2 - 492*B*C*a^2*b^10*c^5*d
^9*f^2 + 480*B*C*a^5*b^7*c^10*d^4*f^2 - 392*B*C*a^10*b^2*c^5*d^9*f^2 + 332*
B*C*a^8*b^4*c^9*d^5*f^2 - 328*B*C*a^6*b^6*c^11*d^3*f^2 + 320*B*C*a^9*b^3*c^
2*d^12*f^2 + 272*B*C*a^3*b^9*c^12*d^2*f^2 - 248*B*C*a^5*b^7*c^4*d^10*f^2 -
248*B*C*a^2*b^10*c^3*d^11*f^2 - 208*B*C*a^7*b^5*c^10*d^4*f^2 - 192*B*C*a^5*
b^7*c^2*d^12*f^2 + 144*B*C*a^2*b^10*c^7*d^7*f^2 - 96*B*C*a^3*b^9*c^4*d^10*f
^2 + 88*B*C*a^5*b^7*c^12*d^2*f^2 - 72*B*C*a^8*b^4*c^11*d^3*f^2 + 48*B*C*a^9
*b^3*c^10*d^4*f^2 - 48*B*C*a^7*b^5*c^12*d^2*f^2 - 48*B*C*a^7*b^5*c^2*d^12*f
^2 - 48*B*C*a^3*b^9*c^2*d^12*f^2 - 12*B*C*a^10*b^2*c^9*d^5*f^2 + 4*B*C*a^6*
b^6*c^5*d^9*f^2 + 5824*A*C*a^7*b^5*c^5*d^9*f^2 - 4378*A*C*a^8*b^4*c^6*d^8*f
^2 + 4296*A*C*a^5*b^7*c^5*d^9*f^2 - 3912*A*C*a^6*b^6*c^6*d^8*f^2 - 3672*A*C
*a^5*b^7*c^9*d^5*f^2 + 3594*A*C*a^4*b^8*c^8*d^6*f^2 + 3236*A*C*a^6*b^6*c^8*
d^6*f^2 + 2816*A*C*a^9*b^3*c^5*d^9*f^2 + 2624*A*C*a^3*b^9*c^5*d^9*f^2 + 243
2*A*C*a^7*b^5*c^7*d^7*f^2 - 2366*A*C*a^8*b^4*c^4*d^10*f^2 + 2298*A*C*a^4*b^
8*c^10*d^4*f^2 + 1872*A*C*a^3*b^9*c^7*d^7*f^2 + 1848*A*C*a^6*b^6*c^10*d^4*f
^2 - 1644*A*C*a^6*b^6*c^4*d^10*f^2 - 1488*A*C*a^7*b^5*c^9*d^5*f^2 - 1408*A*
C*a^3*b^9*c^9*d^5*f^2 - 1308*A*C*a^4*b^8*c^6*d^8*f^2 + 1248*A*C*a^5*b^7*c^7

*d^7*f^2 - 1012*A*C*a^10*b^2*c^6*d^8*f^2 + 1008*A*C*a^7*b^5*c^3*d^11*f^2 +
992*A*C*a^5*b^7*c^3*d^11*f^2 + 928*A*C*a^3*b^9*c^3*d^11*f^2 + 848*A*C*a^9*b
^3*c^7*d^7*f^2 + 636*A*C*a^2*b^10*c^8*d^6*f^2 - 628*A*C*a^10*b^2*c^4*d^10*f
^2 - 600*A*C*a^2*b^10*c^6*d^8*f^2 - 576*A*C*a^5*b^7*c^11*d^3*f^2 + 572*A*C*
a^2*b^10*c^10*d^4*f^2 + 464*A*C*a^8*b^4*c^8*d^6*f^2 + 304*A*C*a^6*b^6*c^2*d
^12*f^2 - 304*A*C*a^4*b^8*c^4*d^10*f^2 + 296*A*C*a^4*b^8*c^2*d^12*f^2 + 260
*A*C*a^8*b^4*c^10*d^4*f^2 - 232*A*C*a^9*b^3*c^9*d^5*f^2 - 232*A*C*a^2*b^10*
c^12*d^2*f^2 + 228*A*C*a^10*b^2*c^2*d^12*f^2 - 188*A*C*a^2*b^10*c^4*d^10*f^
2 + 144*A*C*a^3*b^9*c^11*d^3*f^2 + 116*A*C*a^6*b^6*c^12*d^2*f^2 + 112*A*C*a
^9*b^3*c^3*d^11*f^2 - 112*A*C*a^7*b^5*c^11*d^3*f^2 + 92*A*C*a^10*b^2*c^8*d^
6*f^2 + 74*A*C*a^4*b^8*c^12*d^2*f^2 + 62*A*C*a^8*b^4*c^2*d^12*f^2 + 40*A*C*
a^2*b^10*c^2*d^12*f^2 - 7008*A*B*a^6*b^6*c^7*d^7*f^2 - 4032*A*B*a^4*b^8*c^7
*d^7*f^2 + 3952*A*B*a^7*b^5*c^8*d^6*f^2 + 3648*A*B*a^5*b^7*c^8*d^6*f^2 - 33
92*A*B*a^8*b^4*c^7*d^7*f^2 + 3264*A*B*a^7*b^5*c^6*d^8*f^2 - 2992*A*B*a^5*b^
7*c^4*d^10*f^2 - 2368*A*B*a^7*b^5*c^4*d^10*f^2 - 2304*A*B*a^3*b^9*c^4*d^10*
f^2 - 1968*A*B*a^6*b^6*c^9*d^5*f^2 - 1872*A*B*a^9*b^3*c^4*d^10*f^2 - 1728*A
*B*a^2*b^10*c^7*d^7*f^2 + 1712*A*B*a^8*b^4*c^3*d^11*f^2 + 1536*A*B*a^5*b^7*
c^6*d^8*f^2 - 1536*A*B*a^3*b^9*c^10*d^4*f^2 - 1392*A*B*a^5*b^7*c^2*d^12*f^2
 + 1328*A*B*a^6*b^6*c^3*d^11*f^2 - 1104*A*B*a^3*b^9*c^2*d^12*f^2 - 1056*A*B
*a^3*b^9*c^6*d^8*f^2 + 976*A*B*a^9*b^3*c^6*d^8*f^2 + 960*A*B*a^4*b^8*c^11*d
^3*f^2 + 936*A*B*a^8*b^4*c^5*d^9*f^2 - 912*A*B*a^5*b^7*c^10*d^4*f^2 + 848*A
*B*a^9*b^3*c^8*d^6*f^2 - 816*A*B*a^7*b^5*c^2*d^12*f^2 + 816*A*B*a^4*b^8*c^3
*d^11*f^2 + 768*A*B*a^10*b^2*c^3*d^11*f^2 + 672*A*B*a^3*b^9*c^8*d^6*f^2 - 6
32*A*B*a^8*b^4*c^9*d^5*f^2 - 608*A*B*a^2*b^10*c^9*d^5*f^2 - 552*A*B*a^4*b^8
*c^9*d^5*f^2 - 544*A*B*a^10*b^2*c^7*d^7*f^2 - 480*A*B*a^2*b^10*c^5*d^9*f^2
 + 464*A*B*a^10*b^2*c^5*d^9*f^2 - 464*A*B*a^9*b^3*c^2*d^12*f^2 + 432*A*B*a^2
*b^10*c^11*d^3*f^2 - 368*A*B*a^3*b^9*c^12*d^2*f^2 - 256*A*B*a^6*b^6*c^5*d^9
*f^2 - 208*A*B*a^5*b^7*c^12*d^2*f^2 + 176*A*B*a^4*b^8*c^5*d^9*f^2 + 112*A*B
*a^7*b^5*c^10*d^4*f^2 + 112*A*B*a^6*b^6*c^11*d^3*f^2 - 16*A*B*a^2*b^10*c^3*
d^11*f^2 - 576*B*C*a*b^11*c^8*d^6*f^2 + 400*B*C*a^11*b*c^4*d^10*f^2 - 288*B
*C*a*b^11*c^6*d^8*f^2 - 176*B*C*a^11*b*c^6*d^8*f^2 + 128*B*C*a*b^11*c^10*d^
4*f^2 - 108*B*C*a^4*b^8*c*d^13*f^2 - 104*B*C*a*b^11*c^4*d^10*f^2 - 92*B*C*a
^4*b^8*c^13*d*f^2 - 60*B*C*a^8*b^4*c*d^13*f^2 - 60*B*C*a^6*b^6*c*d^13*f^2 +
 48*B*C*a^11*b*c^2*d^12*f^2 - 40*B*C*a^2*b^10*c*d^13*f^2 - 28*B*C*a^2*b^10*
c^13*d*f^2 - 24*B*C*a*b^11*c^12*d^2*f^2 + 20*B*C*a^10*b^2*c*d^13*f^2 - 16*B
*C*a*b^11*c^2*d^12*f^2 + 12*B*C*a^6*b^6*c^13*d*f^2 + 912*A*C*a*b^11*c^7*d^7
*f^2 + 808*A*C*a*b^11*c^5*d^9*f^2 + 432*A*C*a^11*b*c^5*d^9*f^2 + 336*A*C*a*
b^11*c^3*d^11*f^2 + 224*A*C*a*b^11*c^11*d^3*f^2 - 112*A*C*a^11*b*c^3*d^11*f
^2 + 112*A*C*a^3*b^9*c*d^13*f^2 - 88*A*C*a^9*b^3*c*d^13*f^2 + 80*A*C*a^3*b^
9*c^13*d*f^2 + 56*A*C*a^5*b^7*c*d^13*f^2 + 48*A*C*a*b^11*c^9*d^5*f^2 - 40*A
*C*a^5*b^7*c^13*d*f^2 - 16*A*C*a^11*b*c^7*d^7*f^2 + 16*A*C*a^7*b^5*c*d^13*f
^2 - 496*A*B*a*b^11*c^4*d^10*f^2 - 400*A*B*a^11*b*c^4*d^10*f^2 + 288*A*B*a*
b^11*c^8*d^6*f^2 - 288*A*B*a*b^11*c^6*d^8*f^2 - 272*A*B*a*b^11*c^2*d^12*f^2
 + 240*A*B*a^6*b^6*c*d^13*f^2 - 224*A*B*a*b^11*c^10*d^4*f^2 + 192*A*B*a^8*b
^4*c*d^13*f^2 + 192*A*B*a^4*b^8*c*d^13*f^2 + 176*A*B*a^11*b*c^6*d^8*f^2 + 1

04*A*B*a^4*b^8*c^13*d*f^2 - 48*A*B*a^11*b*c^2*d^12*f^2 + 16*A*B*a^10*b^2*c*
d^13*f^2 + 16*A*B*a^2*b^10*c^13*d*f^2 + 16*A*B*a^2*b^10*c*d^13*f^2 - 112*B*
C*b^12*c^11*d^3*f^2 + 4*B*C*b^12*c^5*d^9*f^2 + 150*A*C*b^12*c^10*d^4*f^2 -
80*B*C*a^12*c^3*d^11*f^2 + 66*A*C*b^12*c^8*d^6*f^2 - 30*A*C*b^12*c^12*d^2*f
^2 + 24*B*C*a^12*c^5*d^9*f^2 - 12*A*C*b^12*c^4*d^10*f^2 - 576*A*B*b^12*c^7*
d^7*f^2 - 432*A*B*b^12*c^9*d^5*f^2 - 400*A*B*b^12*c^5*d^9*f^2 - 144*A*B*b^1
2*c^3*d^11*f^2 - 96*B*C*a^7*b^5*d^14*f^2 - 72*B*C*a^5*b^7*d^14*f^2 - 66*A*C
*a^12*c^4*d^10*f^2 + 54*A*C*a^12*c^2*d^12*f^2 - 32*A*B*b^12*c^11*d^3*f^2 -
24*B*C*a^9*b^3*d^14*f^2 - 16*B*C*a^3*b^9*d^14*f^2 + 2*A*C*a^12*c^6*d^8*f^2
+ 116*A*C*a^6*b^6*d^14*f^2 + 100*A*C*a^4*b^8*d^14*f^2 + 80*A*B*a^12*c^3*d^1
1*f^2 + 24*A*C*a^2*b^10*d^14*f^2 - 24*A*B*a^12*c^5*d^9*f^2 + 22*A*C*a^8*b^4
*d^14*f^2 + 16*B*C*a^3*b^9*c^14*f^2 + 8*A*C*a^10*b^2*d^14*f^2 - 192*A*B*a^5
*b^7*d^14*f^2 - 176*A*B*a^3*b^9*d^14*f^2 - 48*A*B*a^7*b^5*d^14*f^2 - 28*A*C
*a^2*b^10*c^14*f^2 + 2*A*C*a^4*b^8*c^14*f^2 - 16*A*B*a^3*b^9*c^14*f^2 + 250
8*C^2*a^6*b^6*c^6*d^8*f^2 + 2376*C^2*a^5*b^7*c^9*d^5*f^2 + 2357*C^2*a^8*b^4
*c^6*d^8*f^2 - 2048*C^2*a^7*b^5*c^5*d^9*f^2 + 1304*C^2*a^3*b^9*c^9*d^5*f^2
+ 1303*C^2*a^8*b^4*c^4*d^10*f^2 + 1212*C^2*a^6*b^6*c^4*d^10*f^2 - 1203*C^2*
a^4*b^8*c^8*d^6*f^2 - 1192*C^2*a^9*b^3*c^5*d^9*f^2 + 1062*C^2*a^4*b^8*c^6*d
^8*f^2 + 984*C^2*a^7*b^5*c^9*d^5*f^2 - 952*C^2*a^6*b^6*c^8*d^6*f^2 + 768*C^
2*a^5*b^7*c^7*d^7*f^2 - 681*C^2*a^4*b^8*c^10*d^4*f^2 - 672*C^2*a^5*b^7*c^5*
d^9*f^2 - 480*C^2*a^6*b^6*c^10*d^4*f^2 + 458*C^2*a^10*b^2*c^6*d^8*f^2 - 448
*C^2*a^7*b^5*c^7*d^7*f^2 + 422*C^2*a^4*b^8*c^4*d^10*f^2 + 372*C^2*a^2*b^10*
c^6*d^8*f^2 + 360*C^2*a^5*b^7*c^11*d^3*f^2 + 312*C^2*a^3*b^9*c^7*d^7*f^2 +
278*C^2*a^10*b^2*c^4*d^10*f^2 - 232*C^2*a^9*b^3*c^7*d^7*f^2 + 194*C^2*a^2*b
^10*c^12*d^2*f^2 + 176*C^2*a^9*b^3*c^9*d^5*f^2 + 152*C^2*a^5*b^7*c^3*d^11*f
^2 + 124*C^2*a^2*b^10*c^4*d^10*f^2 - 120*C^2*a^7*b^5*c^3*d^11*f^2 - 114*C^2
*a^10*b^2*c^2*d^12*f^2 - 102*C^2*a^2*b^10*c^8*d^6*f^2 + 101*C^2*a^4*b^8*c^1
2*d^2*f^2 + 100*C^2*a^6*b^6*c^2*d^12*f^2 - 88*C^2*a^3*b^9*c^5*d^9*f^2 + 77*
C^2*a^8*b^4*c^2*d^12*f^2 + 72*C^2*a^3*b^9*c^11*d^3*f^2 - 64*C^2*a^10*b^2*c^
8*d^6*f^2 + 64*C^2*a^3*b^9*c^3*d^11*f^2 - 58*C^2*a^2*b^10*c^10*d^4*f^2 + 56
*C^2*a^7*b^5*c^11*d^3*f^2 + 56*C^2*a^6*b^6*c^12*d^2*f^2 + 40*C^2*a^9*b^3*c^
3*d^11*f^2 + 36*C^2*a^8*b^4*c^12*d^2*f^2 + 32*C^2*a^4*b^8*c^2*d^12*f^2 + 26
*C^2*a^8*b^4*c^10*d^4*f^2 + 16*C^2*a^2*b^10*c^2*d^12*f^2 + 2*C^2*a^8*b^4*c^
8*d^6*f^2 + 2277*B^2*a^4*b^8*c^8*d^6*f^2 + 2144*B^2*a^7*b^5*c^5*d^9*f^2 - 2
112*B^2*a^5*b^7*c^9*d^5*f^2 + 2028*B^2*a^6*b^6*c^8*d^6*f^2 - 1671*B^2*a^8*b
^4*c^6*d^8*f^2 + 1275*B^2*a^4*b^8*c^10*d^4*f^2 + 1176*B^2*a^5*b^7*c^5*d^9*f
^2 + 1096*B^2*a^9*b^3*c^5*d^9*f^2 - 1044*B^2*a^6*b^6*c^6*d^8*f^2 + 984*B^2*
a^6*b^6*c^10*d^4*f^2 - 968*B^2*a^3*b^9*c^9*d^5*f^2 - 888*B^2*a^7*b^5*c^9*d^
5*f^2 + 672*B^2*a^7*b^5*c^7*d^7*f^2 + 664*B^2*a^3*b^9*c^5*d^9*f^2 - 649*B^2
*a^8*b^4*c^4*d^10*f^2 + 618*B^2*a^2*b^10*c^8*d^6*f^2 + 514*B^2*a^4*b^8*c^4*
d^10*f^2 + 460*B^2*a^6*b^6*c^2*d^12*f^2 + 422*B^2*a^8*b^4*c^8*d^6*f^2 + 406
*B^2*a^2*b^10*c^10*d^4*f^2 - 382*B^2*a^10*b^2*c^6*d^8*f^2 + 368*B^2*a^4*b^8
*c^2*d^12*f^2 - 312*B^2*a^5*b^7*c^11*d^3*f^2 + 312*B^2*a^3*b^9*c^7*d^7*f^2
+ 248*B^2*a^9*b^3*c^7*d^7*f^2 + 245*B^2*a^8*b^4*c^2*d^12*f^2 - 192*B^2*a^5*
b^7*c^7*d^7*f^2 - 184*B^2*a^9*b^3*c^3*d^11*f^2 + 182*B^2*a^10*b^2*c^2*d^12*

f^2 + 176*B^2*a^3*b^9*c^3*d^11*f^2 + 174*B^2*a^4*b^8*c^6*d^8*f^2 - 170*B^2*
a^10*b^2*c^4*d^10*f^2 - 152*B^2*a^9*b^3*c^9*d^5*f^2 + 152*B^2*a^2*b^10*c^4*
d^10*f^2 + 142*B^2*a^8*b^4*c^10*d^4*f^2 - 90*B^2*a^2*b^10*c^12*d^2*f^2 + 88
*B^2*a^2*b^10*c^2*d^12*f^2 + 84*B^2*a^10*b^2*c^8*d^6*f^2 + 84*B^2*a^2*b^10*
c^6*d^8*f^2 + 60*B^2*a^6*b^6*c^12*d^2*f^2 - 56*B^2*a^7*b^5*c^11*d^3*f^2 + 5
3*B^2*a^4*b^8*c^12*d^2*f^2 + 24*B^2*a^7*b^5*c^3*d^11*f^2 + 24*B^2*a^6*b^6*c
^4*d^10*f^2 + 24*B^2*a^3*b^9*c^11*d^3*f^2 - 8*B^2*a^5*b^7*c^3*d^11*f^2 + 45
66*A^2*a^4*b^8*c^6*d^8*f^2 + 4284*A^2*a^6*b^6*c^6*d^8*f^2 - 3776*A^2*a^7*b^
5*c^5*d^9*f^2 - 3624*A^2*a^5*b^7*c^5*d^9*f^2 + 3122*A^2*a^4*b^8*c^4*d^10*f^
2 + 3108*A^2*a^2*b^10*c^6*d^8*f^2 + 2741*A^2*a^8*b^4*c^6*d^8*f^2 + 2592*A^2
*a^6*b^6*c^4*d^10*f^2 - 2536*A^2*a^3*b^9*c^5*d^9*f^2 + 2224*A^2*a^2*b^10*c^
4*d^10*f^2 - 2184*A^2*a^3*b^9*c^7*d^7*f^2 - 2016*A^2*a^5*b^7*c^7*d^7*f^2 -
1984*A^2*a^7*b^5*c^7*d^7*f^2 + 1626*A^2*a^2*b^10*c^8*d^6*f^2 - 1624*A^2*a^9
*b^3*c^5*d^9*f^2 + 1603*A^2*a^8*b^4*c^4*d^10*f^2 + 1296*A^2*a^5*b^7*c^9*d^5
*f^2 - 1144*A^2*a^5*b^7*c^3*d^11*f^2 - 992*A^2*a^3*b^9*c^3*d^11*f^2 + 968*A
^2*a^4*b^8*c^2*d^12*f^2 - 888*A^2*a^7*b^5*c^3*d^11*f^2 + 849*A^2*a^4*b^8*c^
8*d^6*f^2 + 808*A^2*a^2*b^10*c^2*d^12*f^2 - 616*A^2*a^9*b^3*c^7*d^7*f^2 + 5
54*A^2*a^10*b^2*c^6*d^8*f^2 + 504*A^2*a^7*b^5*c^9*d^5*f^2 - 504*A^2*a^6*b^6
*c^10*d^4*f^2 + 460*A^2*a^6*b^6*c^2*d^12*f^2 + 350*A^2*a^10*b^2*c^4*d^10*f^
2 + 350*A^2*a^2*b^10*c^10*d^4*f^2 - 321*A^2*a^4*b^8*c^10*d^4*f^2 + 216*A^2*
a^5*b^7*c^11*d^3*f^2 - 216*A^2*a^3*b^9*c^11*d^3*f^2 + 182*A^2*a^2*b^10*c^12
*d^2*f^2 - 152*A^2*a^9*b^3*c^3*d^11*f^2 - 124*A^2*a^6*b^6*c^8*d^6*f^2 - 114
*A^2*a^10*b^2*c^2*d^12*f^2 + 104*A^2*a^3*b^9*c^9*d^5*f^2 + 77*A^2*a^8*b^4*c
^2*d^12*f^2 + 74*A^2*a^8*b^4*c^8*d^6*f^2 - 70*A^2*a^8*b^4*c^10*d^4*f^2 + 56
*A^2*a^9*b^3*c^9*d^5*f^2 + 56*A^2*a^7*b^5*c^11*d^3*f^2 + 41*A^2*a^4*b^8*c^1
2*d^2*f^2 - 28*A^2*a^10*b^2*c^8*d^6*f^2 - 28*A^2*a^6*b^6*c^12*d^2*f^2 + 12*
B*C*b^12*c^13*d*f^2 + 24*B*C*a^12*c*d^13*f^2 - 24*A*B*b^12*c^13*d*f^2 - 24*
A*B*b^12*c*d^13*f^2 - 16*B*C*a^11*b*d^14*f^2 - 24*A*B*a^12*c*d^13*f^2 - 16*
B*C*a*b^11*c^14*f^2 - 48*A*B*a*b^11*d^14*f^2 + 16*A*B*a^11*b*d^14*f^2 + 16*
A*B*a*b^11*c^14*f^2 - 216*C^2*a^11*b*c^5*d^9*f^2 + 216*C^2*a*b^11*c^9*d^5*f
^2 + 56*C^2*a^11*b*c^3*d^11*f^2 + 56*C^2*a^9*b^3*c*d^13*f^2 + 56*C^2*a^5*b^
7*c*d^13*f^2 + 40*C^2*a^7*b^5*c*d^13*f^2 - 40*C^2*a*b^11*c^11*d^3*f^2 + 32*
C^2*a^5*b^7*c^13*d*f^2 - 24*C^2*a*b^11*c^7*d^7*f^2 - 16*C^2*a^3*b^9*c^13*d*
f^2 + 16*C^2*a^3*b^9*c*d^13*f^2 + 8*C^2*a^11*b*c^7*d^7*f^2 - 8*C^2*a*b^11*c
^5*d^9*f^2 + 264*B^2*a*b^11*c^7*d^7*f^2 + 224*B^2*a*b^11*c^5*d^9*f^2 + 168*
B^2*a^11*b*c^5*d^9*f^2 - 112*B^2*a^9*b^3*c*d^13*f^2 - 104*B^2*a^11*b*c^3*d^
11*f^2 - 104*B^2*a^7*b^5*c*d^13*f^2 + 96*B^2*a*b^11*c^3*d^11*f^2 + 88*B^2*a
*b^11*c^11*d^3*f^2 - 72*B^2*a*b^11*c^9*d^5*f^2 - 64*B^2*a^5*b^7*c*d^13*f^2
+ 32*B^2*a^3*b^9*c^13*d*f^2 - 24*B^2*a^11*b*c^7*d^7*f^2 - 24*B^2*a^5*b^7*c^
13*d*f^2 + 16*B^2*a^3*b^9*c*d^13*f^2 - 888*A^2*a*b^11*c^7*d^7*f^2 - 800*A^2
*a*b^11*c^5*d^9*f^2 - 336*A^2*a*b^11*c^3*d^11*f^2 - 264*A^2*a*b^11*c^9*d^5*
f^2 - 216*A^2*a^11*b*c^5*d^9*f^2 - 184*A^2*a*b^11*c^11*d^3*f^2 - 128*A^2*a^
3*b^9*c*d^13*f^2 - 112*A^2*a^5*b^7*c*d^13*f^2 - 64*A^2*a^3*b^9*c^13*d*f^2 +
56*A^2*a^11*b*c^3*d^11*f^2 - 56*A^2*a^7*b^5*c*d^13*f^2 + 32*A^2*a^9*b^3*c*
d^13*f^2 + 8*A^2*a^11*b*c^7*d^7*f^2 + 8*A^2*a^5*b^7*c^13*d*f^2 + 24*C^2*a^1

1*b*c*d^13*f^2 - 16*C^2*a*b^11*c^13*d*f^2 - 40*B^2*a^11*b*c*d^13*f^2 + 24*B
^2*a*b^11*c^13*d*f^2 + 16*B^2*a*b^11*c*d^13*f^2 - 48*A^2*a*b^11*c*d^13*f^2
- 40*A^2*a*b^11*c^13*d*f^2 + 24*A^2*a^11*b*c*d^13*f^2 - 6*A*C*a^12*d^14*f^2
+ 2*A*C*b^12*c^14*f^2 + 33*C^2*b^12*c^12*d^2*f^2 - 27*C^2*b^12*c^10*d^4*f^
2 + 3*C^2*b^12*c^8*d^6*f^2 + 117*B^2*b^12*c^10*d^4*f^2 + 111*B^2*b^12*c^8*d
^6*f^2 + 72*B^2*b^12*c^6*d^8*f^2 + 33*C^2*a^12*c^4*d^10*f^2 - 27*C^2*a^12*c
^2*d^12*f^2 + 24*B^2*b^12*c^4*d^10*f^2 + 4*B^2*b^12*c^2*d^12*f^2 - 3*B^2*b^
12*c^12*d^2*f^2 - C^2*a^12*c^6*d^8*f^2 + 720*A^2*b^12*c^6*d^8*f^2 + 552*A^2
*b^12*c^4*d^10*f^2 + 471*A^2*b^12*c^8*d^6*f^2 + 216*A^2*b^12*c^2*d^12*f^2 +
93*A^2*b^12*c^10*d^4*f^2 + 33*B^2*a^12*c^2*d^12*f^2 + 33*A^2*b^12*c^12*d^2
*f^2 + 31*C^2*a^8*b^4*d^14*f^2 - 27*B^2*a^12*c^4*d^10*f^2 + 20*C^2*a^6*b^6*
d^14*f^2 + 4*C^2*a^4*b^8*d^14*f^2 + 3*B^2*a^12*c^6*d^8*f^2 + 2*C^2*a^10*b^2
*d^14*f^2 + 80*B^2*a^6*b^6*d^14*f^2 + 64*B^2*a^4*b^8*d^14*f^2 + 33*A^2*a^12
*c^4*d^10*f^2 + 31*B^2*a^8*b^4*d^14*f^2 - 27*A^2*a^12*c^2*d^12*f^2 + 16*B^2
*a^2*b^10*d^14*f^2 + 14*C^2*a^2*b^10*c^14*f^2 + 14*B^2*a^10*b^2*d^14*f^2 -
C^2*a^4*b^8*c^14*f^2 - A^2*a^12*c^6*d^8*f^2 + 120*A^2*a^2*b^10*d^14*f^2 + 1
12*A^2*a^4*b^8*d^14*f^2 - 17*A^2*a^8*b^4*d^14*f^2 - 10*B^2*a^2*b^10*c^14*f^
2 - 10*A^2*a^10*b^2*d^14*f^2 + 8*A^2*a^6*b^6*d^14*f^2 + 3*B^2*a^4*b^8*c^14*
f^2 + 14*A^2*a^2*b^10*c^14*f^2 - A^2*a^4*b^8*c^14*f^2 + 3*C^2*a^12*d^14*f^2
- C^2*b^12*c^14*f^2 + 36*A^2*b^12*d^14*f^2 + 3*B^2*b^12*c^14*f^2 - B^2*a^1
2*d^14*f^2 + 3*A^2*a^12*d^14*f^2 - A^2*b^12*c^14*f^2 - 44*A*B*C*a*b^9*c^10*
d*f + 3816*A*B*C*a^5*b^5*c^4*d^7*f + 2920*A*B*C*a^2*b^8*c^5*d^6*f - 2736*A*
B*C*a^3*b^7*c^6*d^5*f - 2672*A*B*C*a^4*b^6*c^3*d^8*f + 1996*A*B*C*a^4*b^6*c
^7*d^4*f - 1412*A*B*C*a^6*b^4*c^5*d^6*f + 1120*A*B*C*a^3*b^7*c^2*d^9*f + 10
80*A*B*C*a^2*b^8*c^7*d^4*f + 1040*A*B*C*a^5*b^5*c^2*d^9*f + 684*A*B*C*a^4*b
^6*c^5*d^6*f + 592*A*B*C*a^3*b^7*c^4*d^7*f - 560*A*B*C*a^7*b^3*c^2*d^9*f -
448*A*B*C*a^2*b^8*c^3*d^8*f - 400*A*B*C*a^5*b^5*c^8*d^3*f - 398*A*B*C*a^2*b
^8*c^9*d^2*f - 312*A*B*C*a^6*b^4*c^3*d^8*f + 166*A*B*C*a^8*b^2*c^3*d^8*f +
136*A*B*C*a^5*b^5*c^6*d^5*f + 128*A*B*C*a^7*b^3*c^6*d^5*f - 100*A*B*C*a^6*b
^4*c^7*d^4*f + 64*A*B*C*a^7*b^3*c^4*d^7*f - 64*A*B*C*a^4*b^6*c^9*d^2*f - 32
*A*B*C*a^3*b^7*c^8*d^3*f - 16*A*B*C*a^8*b^2*c^5*d^6*f - 1312*A*B*C*a*b^9*c^
4*d^7*f + 996*A*B*C*a*b^9*c^8*d^3*f + 728*A*B*C*a^6*b^4*c*d^10*f - 624*A*B*
C*a*b^9*c^6*d^5*f - 584*A*B*C*a^2*b^8*c*d^10*f - 512*A*B*C*a^4*b^6*c*d^10*f
- 320*A*B*C*a*b^9*c^2*d^9*f - 98*A*B*C*a^8*b^2*c*d^10*f + 36*A*B*C*a^9*b*c
^2*d^9*f + 32*A*B*C*a^3*b^7*c^10*d*f - 16*A*B*C*a^9*b*c^4*d^7*f + 46*B*C^2*
a*b^9*c^10*d*f - 16*B^2*C*a*b^9*c*d^10*f - 2*B^2*C*a^9*b*c*d^10*f + 312*A^2
*C*a*b^9*c*d^10*f - 48*A*C^2*a*b^9*c*d^10*f - 6*A^2*C*a^9*b*c*d^10*f + 6*A*
C^2*a^9*b*c*d^10*f + 208*A*B^2*a*b^9*c*d^10*f - 2*A^2*B*a*b^9*c^10*d*f + 2*
A*B^2*a^9*b*c*d^10*f - 480*A*B*C*b^10*c^7*d^4*f + 78*A*B*C*b^10*c^9*d^2*f -
64*A*B*C*b^10*c^5*d^6*f + 2*A*B*C*a^10*c^3*d^8*f - 224*A*B*C*a^5*b^5*d^11*
f + 80*A*B*C*a^7*b^3*d^11*f - 32*A*B*C*a^3*b^7*d^11*f + 2*A*B*C*a^2*b^8*c^1
1*f - 1692*B*C^2*a^5*b^5*c^4*d^7*f - 1500*B^2*C*a^5*b^5*c^5*d^6*f - 1464*B^
2*C*a^3*b^7*c^5*d^6*f + 1426*B*C^2*a^6*b^4*c^5*d^6*f - 1158*B^2*C*a^6*b^4*c
^4*d^7*f + 1152*B*C^2*a^3*b^7*c^6*d^5*f + 1026*B^2*C*a^4*b^6*c^6*d^5*f - 97
4*B*C^2*a^4*b^6*c^7*d^4*f + 960*B^2*C*a^5*b^5*c^3*d^8*f - 884*B*C^2*a^2*b^8

*c^5*d^6*f - 764*B^2*C*a^5*b^5*c^7*d^4*f + 752*B^2*C*a^2*b^8*c^4*d^7*f - 75
2*B*C^2*a^3*b^7*c^4*d^7*f + 738*B^2*C*a^4*b^6*c^4*d^7*f - 688*B^2*C*a^6*b^4
*c^2*d^9*f - 675*B^2*C*a^2*b^8*c^8*d^3*f + 560*B*C^2*a^5*b^5*c^8*d^3*f + 49
6*B*C^2*a^7*b^3*c^2*d^9*f + 496*B*C^2*a^4*b^6*c^3*d^8*f - 468*B*C^2*a^2*b^8
*c^7*d^4*f + 456*B^2*C*a^7*b^3*c^3*d^8*f - 452*B^2*C*a^4*b^6*c^8*d^3*f - 41
6*B*C^2*a^3*b^7*c^2*d^9*f + 378*B*C^2*a^4*b^6*c^5*d^6*f + 376*B*C^2*a^3*b^7
*c^8*d^3*f - 360*B^2*C*a^2*b^8*c^6*d^5*f + 355*B*C^2*a^2*b^8*c^9*d^2*f + 34
6*B^2*C*a^6*b^4*c^6*d^5*f - 320*B^2*C*a^4*b^6*c^2*d^9*f + 268*B^2*C*a^2*b^8
*c^2*d^9*f + 216*B^2*C*a^3*b^7*c^7*d^4*f - 203*B*C^2*a^8*b^2*c^3*d^8*f - 18
4*B*C^2*a^7*b^3*c^6*d^5*f + 170*B*C^2*a^6*b^4*c^7*d^4*f + 160*B^2*C*a^7*b^3
*c^5*d^6*f - 160*B*C^2*a^5*b^5*c^2*d^9*f - 140*B^2*C*a^8*b^2*c^4*d^7*f - 13
6*B*C^2*a^2*b^8*c^3*d^8*f + 112*B^2*C*a^3*b^7*c^9*d^2*f + 91*B^2*C*a^8*b^2*
c^2*d^9*f + 88*B*C^2*a^7*b^3*c^4*d^7*f + 72*B^2*C*a^6*b^4*c^8*d^3*f - 64*B^
2*C*a^3*b^7*c^3*d^8*f - 60*B*C^2*a^6*b^4*c^3*d^8*f + 56*B*C^2*a^4*b^6*c^9*d
^2*f + 52*B*C^2*a^5*b^5*c^6*d^5*f - 48*B^2*C*a^7*b^3*c^7*d^4*f + 48*B^2*C*a
^5*b^5*c^9*d^2*f + 44*B*C^2*a^8*b^2*c^5*d^6*f - 36*B*C^2*a^6*b^4*c^9*d^2*f
 + 12*B^2*C*a^8*b^2*c^6*d^5*f - 2958*A^2*C*a^4*b^6*c^4*d^7*f - 1932*A^2*C*a^
2*b^8*c^4*d^7*f + 1848*A^2*C*a^3*b^7*c^5*d^6*f + 1728*A^2*C*a^3*b^7*c^3*d^8
*f + 1524*A^2*C*a^5*b^5*c^5*d^6*f + 1374*A*C^2*a^4*b^6*c^4*d^7*f - 1272*A*C
^2*a^3*b^7*c^5*d^6*f - 1236*A*C^2*a^5*b^5*c^5*d^6*f + 1116*A*C^2*a^2*b^8*c^
4*d^7*f - 1110*A^2*C*a^4*b^6*c^6*d^5*f + 1038*A*C^2*a^4*b^6*c^6*d^5*f - 768
*A^2*C*a^2*b^8*c^2*d^9*f - 696*A^2*C*a^3*b^7*c^7*d^4*f - 666*A*C^2*a^6*b^4*
c^4*d^7*f + 564*A^2*C*a^2*b^8*c^6*d^5*f - 564*A*C^2*a^5*b^5*c^7*d^4*f - 555
*A*C^2*a^2*b^8*c^8*d^3*f + 519*A^2*C*a^2*b^8*c^8*d^3*f - 480*A*C^2*a^3*b^7*
c^3*d^8*f + 456*A*C^2*a^5*b^5*c^3*d^8*f - 420*A*C^2*a^6*b^4*c^2*d^9*f + 408
*A*C^2*a^3*b^7*c^7*d^4*f + 408*A*C^2*a^2*b^8*c^2*d^9*f + 348*A^2*C*a^6*b^4*
c^2*d^9*f - 348*A*C^2*a^2*b^8*c^6*d^5*f + 342*A*C^2*a^6*b^4*c^6*d^5*f - 336
*A*C^2*a^4*b^6*c^8*d^3*f + 324*A^2*C*a^5*b^5*c^7*d^4*f - 312*A^2*C*a^4*b^6*
c^2*d^9*f + 264*A^2*C*a^4*b^6*c^8*d^3*f + 240*A*C^2*a^7*b^3*c^5*d^6*f + 195
*A*C^2*a^8*b^2*c^2*d^9*f - 174*A^2*C*a^6*b^4*c^6*d^5*f + 144*A*C^2*a^3*b^7*
c^9*d^2*f - 123*A^2*C*a^8*b^2*c^2*d^9*f + 120*A*C^2*a^7*b^3*c^3*d^8*f + 108
*A*C^2*a^6*b^4*c^8*d^3*f - 102*A^2*C*a^6*b^4*c^4*d^7*f - 96*A^2*C*a^8*b^2*c
^4*d^7*f + 72*A^2*C*a^7*b^3*c^3*d^8*f + 72*A*C^2*a^5*b^5*c^9*d^2*f + 48*A^2
*C*a^7*b^3*c^5*d^6*f - 48*A^2*C*a^3*b^7*c^9*d^2*f - 48*A*C^2*a^4*b^6*c^2*d^
9*f - 24*A^2*C*a^5*b^5*c^3*d^8*f - 12*A*C^2*a^8*b^2*c^4*d^7*f + 2736*A^2*B*
a^3*b^7*c^6*d^5*f + 2464*A^2*B*a^4*b^6*c^3*d^8*f - 2298*A*B^2*a^4*b^6*c^4*d
^7*f - 2252*A^2*B*a^2*b^8*c^5*d^6*f - 1692*A^2*B*a^5*b^5*c^4*d^7*f - 1592*A
*B^2*a^2*b^8*c^4*d^7*f - 1338*A*B^2*a^4*b^6*c^6*d^5*f + 1320*A*B^2*a^3*b^7*
c^5*d^6*f + 1212*A*B^2*a^5*b^5*c^5*d^6*f - 1056*A*B^2*a^5*b^5*c^3*d^8*f + 1
024*A^2*B*a^3*b^7*c^4*d^7*f - 1022*A^2*B*a^4*b^6*c^7*d^4*f - 880*A^2*B*a^5*
b^5*c^2*d^9*f - 846*A^2*B*a^4*b^6*c^5*d^6*f - 840*A*B^2*a^3*b^7*c^7*d^4*f +
 760*A*B^2*a^6*b^4*c^2*d^9*f - 704*A^2*B*a^3*b^7*c^2*d^9*f + 688*A*B^2*a^3*
b^7*c^3*d^8*f + 660*A^2*B*a^6*b^4*c^3*d^8*f - 612*A^2*B*a^2*b^8*c^7*d^4*f +
 462*A*B^2*a^6*b^4*c^4*d^7*f + 459*A*B^2*a^2*b^8*c^8*d^3*f - 412*A*B^2*a^2*
b^8*c^2*d^9*f - 408*A*B^2*a^7*b^3*c^3*d^8*f + 388*A^2*B*a^5*b^5*c^6*d^5*f +

296*A^2*B*a^2*b^8*c^3*d^8*f + 288*A*B^2*a^2*b^8*c^6*d^5*f + 284*A*B^2*a^5*
b^5*c^7*d^4*f + 236*A*B^2*a^4*b^6*c^8*d^3*f - 226*A*B^2*a^6*b^4*c^6*d^5*f +
212*A*B^2*a^4*b^6*c^2*d^9*f + 202*A^2*B*a^6*b^4*c^5*d^6*f - 152*A^2*B*a^7*
b^3*c^4*d^7*f + 88*A^2*B*a^3*b^7*c^8*d^3*f + 79*A^2*B*a^2*b^8*c^9*d^2*f - 7
0*A^2*B*a^6*b^4*c^7*d^4*f + 68*A*B^2*a^8*b^2*c^4*d^7*f + 64*A^2*B*a^7*b^3*c
^2*d^9*f - 64*A*B^2*a^3*b^7*c^9*d^2*f + 56*A^2*B*a^7*b^3*c^6*d^5*f + 56*A^2
*B*a^5*b^5*c^8*d^3*f + 37*A^2*B*a^8*b^2*c^3*d^8*f - 28*A^2*B*a^8*b^2*c^5*d^
6*f - 28*A^2*B*a^4*b^6*c^9*d^2*f + 17*A*B^2*a^8*b^2*c^2*d^9*f - 16*A*B^2*a^
7*b^3*c^5*d^6*f + 24*A*B*C*b^10*c*d^10*f - 6*A*B*C*a^10*c*d^10*f + 48*A*B*C
*a*b^9*d^11*f + 4*A*B*C*a^9*b*d^11*f + 432*B^2*C*a*b^9*c^7*d^4*f - 376*B*C^
2*a^6*b^4*c*d^10*f - 354*B*C^2*a*b^9*c^8*d^3*f + 352*B^2*C*a^5*b^5*c*d^10*f
+ 320*B^2*C*a*b^9*c^5*d^6*f + 256*B^2*C*a^3*b^7*c*d^10*f - 232*B^2*C*a^7*b
^3*c*d^10*f - 210*B^2*C*a*b^9*c^9*d^2*f - 152*B*C^2*a^4*b^6*c*d^10*f + 85*B
*C^2*a^8*b^2*c*d^10*f + 72*B^2*C*a*b^9*c^3*d^8*f - 48*B*C^2*a*b^9*c^6*d^5*f
- 40*B*C^2*a^3*b^7*c^10*d*f + 40*B*C^2*a^2*b^8*c*d^10*f + 37*B^2*C*a^2*b^8
*c^10*d*f + 22*B^2*C*a^9*b*c^3*d^8*f - 18*B*C^2*a^9*b*c^2*d^9*f + 16*B*C^2*
a*b^9*c^2*d^9*f - 12*B^2*C*a^4*b^6*c^10*d*f + 8*B*C^2*a^9*b*c^4*d^7*f + 8*B
*C^2*a*b^9*c^4*d^7*f - 984*A^2*C*a*b^9*c^7*d^4*f + 672*A^2*C*a*b^9*c^3*d^8*
f + 552*A*C^2*a*b^9*c^7*d^4*f - 504*A^2*C*a^5*b^5*c*d^10*f - 408*A^2*C*a*b^
9*c^5*d^6*f + 408*A*C^2*a*b^9*c^5*d^6*f + 336*A*C^2*a^5*b^5*c*d^10*f - 216*
A*C^2*a^7*b^3*c*d^10*f + 192*A*C^2*a^3*b^7*c*d^10*f - 162*A*C^2*a*b^9*c^9*d
^2*f + 120*A^2*C*a^7*b^3*c*d^10*f + 96*A^2*C*a^3*b^7*c*d^10*f + 90*A^2*C*a*
b^9*c^9*d^2*f + 66*A^2*C*a^9*b*c^3*d^8*f - 66*A*C^2*a^9*b*c^3*d^8*f + 57*A*
C^2*a^2*b^8*c^10*d*f - 48*A*C^2*a*b^9*c^3*d^8*f - 9*A^2*C*a^2*b^8*c^10*d*f
+ 1736*A^2*B*a*b^9*c^4*d^7*f + 1248*A^2*B*a*b^9*c^6*d^5*f - 1008*A*B^2*a*b^
9*c^7*d^4*f + 772*A^2*B*a^4*b^6*c*d^10*f - 688*A*B^2*a^5*b^5*c*d^10*f - 608
*A*B^2*a*b^9*c^5*d^6*f + 436*A^2*B*a^2*b^8*c*d^10*f - 426*A^2*B*a*b^9*c^8*d
^3*f + 312*A*B^2*a*b^9*c^3*d^8*f + 304*A^2*B*a*b^9*c^2*d^9*f - 244*A^2*B*a^
6*b^4*c*d^10*f - 160*A*B^2*a^3*b^7*c*d^10*f + 114*A*B^2*a*b^9*c^9*d^2*f + 8
8*A*B^2*a^7*b^3*c*d^10*f - 22*A*B^2*a^9*b*c^3*d^8*f - 18*A^2*B*a^9*b*c^2*d^
9*f + 13*A^2*B*a^8*b^2*c*d^10*f - 13*A*B^2*a^2*b^8*c^10*d*f + 8*A^2*B*a^9*b
*c^4*d^7*f + 8*A^2*B*a^3*b^7*c^10*d*f + 111*B^2*C*b^10*c^8*d^3*f - 39*B*C^2
*b^10*c^9*d^2*f + 24*B*C^2*b^10*c^7*d^4*f - 4*B^2*C*b^10*c^2*d^9*f - 4*B*C^
2*b^10*c^5*d^6*f + 432*A^2*C*b^10*c^6*d^5*f + 192*A^2*C*b^10*c^4*d^7*f - 11
1*A^2*C*b^10*c^8*d^3*f + 111*A*C^2*b^10*c^8*d^3*f - 72*A*C^2*b^10*c^6*d^5*f
+ 12*A*C^2*b^10*c^4*d^7*f - 3*B^2*C*a^10*c^2*d^9*f - B*C^2*a^10*c^3*d^8*f
+ 456*A^2*B*b^10*c^7*d^4*f - 288*A^2*B*b^10*c^3*d^8*f + 252*A*B^2*b^10*c^6*
d^5*f + 192*A*B^2*b^10*c^4*d^7*f - 183*A*B^2*b^10*c^8*d^3*f - 148*A^2*B*b^1
0*c^5*d^6*f + 112*B^2*C*a^6*b^4*d^11*f + 76*A*B^2*b^10*c^2*d^9*f - 64*B*C^2
*a^7*b^3*d^11*f + 16*B^2*C*a^4*b^6*d^11*f - 16*B^2*C*a^2*b^8*d^11*f + 16*B*
C^2*a^5*b^5*d^11*f + 16*B*C^2*a^3*b^7*d^11*f - 9*A^2*C*a^10*c^2*d^9*f + 9*A
*C^2*a^10*c^2*d^9*f - 3*A^2*B*b^10*c^9*d^2*f - B^2*C*a^8*b^2*d^11*f + 96*A^
2*C*a^4*b^6*d^11*f - 84*A^2*C*a^6*b^4*d^11*f + 72*A*C^2*a^6*b^4*d^11*f - 24
*A*C^2*a^4*b^6*d^11*f - 24*A*C^2*a^2*b^8*d^11*f - 21*A*C^2*a^8*b^2*d^11*f +
12*A^2*C*a^2*b^8*d^11*f + 9*A^2*C*a^8*b^2*d^11*f + 3*A*B^2*a^10*c^2*d^9*f

- A^2*B*a^10*c^3*d^8*f - B*C^2*a^2*b^8*c^11*f + 176*A*B^2*a^4*b^6*d^11*f + 136*A^2*B*a^5*b^5*d^11*f - 128*A^2*B*a^3*b^7*d^11*f + 112*A*B^2*a^2*b^8*d^11*f - 64*A*B^2*a^6*b^4*d^11*f - 16*A^2*B*a^7*b^3*d^11*f - A^2*B*a^2*b^8*c^11*f - 2*C^3*a^9*b*c*d^10*f - 2*B^3*a*b^9*c^10*d*f - 264*A^3*a*b^9*c*d^10*f + 2*A^3*a^9*b*c*d^10*f - 9*B^2*C*b^10*c^10*d*f + 9*A^2*C*b^10*c^10*d*f - 9*A*C^2*b^10*c^10*d*f + 3*B*C^2*a^10*c*d^10*f - 132*A^2*B*b^10*c*d^10*f - 3*A*B^2*b^10*c^10*d*f - 2*B*C^2*a^9*b*d^11*f + 3*A^2*B*a^10*c*d^10*f - 2*B^2*C*a*b^9*c^11*f - 120*A^2*B*a*b^9*d^11*f - 6*A^2*C*a*b^9*c^11*f + 6*A*C^2*a*b^9*c^11*f - 2*A^2*B*a^9*b*d^11*f + 2*A*B^2*a*b^9*c^11*f + 520*C^3*a^3*b^7*c^5*d^6*f + 460*C^3*a^5*b^5*c^5*d^6*f - 418*C^3*a^4*b^6*c^6*d^5*f + 406*C^3*a^6*b^4*c^4*d^7*f + 268*C^3*a^5*b^5*c^7*d^4*f - 266*C^3*a^6*b^4*c^6*d^5*f + 233*C^3*a^2*b^8*c^8*d^3*f - 176*C^3*a^7*b^3*c^5*d^6*f + 164*C^3*a^6*b^4*c^2*d^9*f + 140*C^3*a^2*b^8*c^6*d^5*f + 136*C^3*a^4*b^6*c^2*d^9*f - 128*C^3*a^3*b^7*c^9*d^2*f + 128*C^3*a^3*b^7*c^3*d^8*f - 108*C^3*a^6*b^4*c^8*d^3*f - 104*C^3*a^7*b^3*c^3*d^8*f - 104*C^3*a^5*b^5*c^3*d^8*f + 100*C^3*a^4*b^6*c^8*d^3*f - 89*C^3*a^8*b^2*c^2*d^9*f - 72*C^3*a^5*b^5*c^9*d^2*f + 40*C^3*a^8*b^2*c^4*d^7*f - 40*C^3*a^3*b^7*c^7*d^4*f - 28*C^3*a^2*b^8*c^4*d^7*f - 16*C^3*a^2*b^8*c^2*d^9*f - 2*C^3*a^4*b^6*c^4*d^7*f + 828*B^3*a^5*b^5*c^4*d^7*f + 408*B^3*a^2*b^8*c^5*d^6*f + 390*B^3*a^4*b^6*c^7*d^4*f - 372*B^3*a^4*b^6*c^3*d^8*f - 336*B^3*a^3*b^7*c^6*d^5*f - 314*B^3*a^6*b^4*c^5*d^6*f + 288*B^3*a^3*b^7*c^4*d^7*f + 216*B^3*a^2*b^8*c^7*d^4*f - 176*B^3*a^7*b^3*c^2*d^9*f + 128*B^3*a^3*b^7*c^2*d^9*f + 108*B^3*a^5*b^5*c^6*d^5*f + 88*B^3*a^7*b^3*c^4*d^7*f + 72*B^3*a^5*b^5*c^2*d^9*f - 68*B^3*a^2*b^8*c^3*d^8*f - 65*B^3*a^2*b^8*c^9*d^2*f - 56*B^3*a^5*b^5*c^8*d^3*f + 40*B^3*a^7*b^3*c^6*d^5*f + 37*B^3*a^8*b^2*c^3*d^8*f + 30*B^3*a^4*b^6*c^5*d^6*f - 28*B^3*a^8*b^2*c^5*d^6*f + 24*B^3*a^3*b^7*c^8*d^3*f - 4*B^3*a^4*b^6*c^9*d^2*f - 2*B^3*a^6*b^4*c^7*d^4*f + 1586*A^3*a^4*b^6*c^4*d^7*f - 1376*A^3*a^3*b^7*c^3*d^8*f - 1096*A^3*a^3*b^7*c^5*d^6*f + 844*A^3*a^2*b^8*c^4*d^7*f - 748*A^3*a^5*b^5*c^5*d^6*f + 490*A^3*a^4*b^6*c^6*d^5*f + 376*A^3*a^2*b^8*c^2*d^9*f + 362*A^3*a^6*b^4*c^4*d^7*f - 356*A^3*a^2*b^8*c^6*d^5*f - 328*A^3*a^5*b^5*c^3*d^8*f + 328*A^3*a^3*b^7*c^7*d^4*f + 224*A^3*a^4*b^6*c^2*d^9*f - 197*A^3*a^2*b^8*c^8*d^3*f - 112*A^3*a^7*b^3*c^5*d^6*f + 98*A^3*a^6*b^4*c^6*d^5*f - 92*A^3*a^6*b^4*c^2*d^9*f - 88*A^3*a^7*b^3*c^3*d^8*f + 68*A^3*a^8*b^2*c^4*d^7*f + 32*A^3*a^3*b^7*c^9*d^2*f - 28*A^3*a^5*b^5*c^7*d^4*f - 28*A^3*a^4*b^6*c^8*d^3*f + 17*A^3*a^8*b^2*c^2*d^9*f + 104*C^3*a^7*b^3*c*d^10*f + 54*C^3*a*b^9*c^9*d^2*f - 40*C^3*a*b^9*c^7*d^4*f - 35*C^3*a^2*b^8*c^10*d*f + 22*C^3*a^9*b*c^3*d^8*f + 16*C^3*a^5*b^5*c*d^10*f - 16*C^3*a^3*b^7*c*d^10*f + 8*C^3*a*b^9*c^5*d^6*f - 2*A*B*C*b^10*c^11*f + 198*B^3*a*b^9*c^8*d^3*f + 192*B^3*a^6*b^4*c*d^10*f - 128*B^3*a*b^9*c^4*d^7*f - 80*B^3*a^2*b^8*c*d^10*f - 56*B^3*a*b^9*c^2*d^9*f - 24*B^3*a*b^9*c^6*d^5*f - 18*B^3*a^9*b*c^2*d^9*f - 16*B^3*a^4*b^6*c*d^10*f + 13*B^3*a^8*b^2*c*d^10*f + 8*B^3*a^9*b*c^4*d^7*f + 8*B^3*a^3*b^7*c^10*d*f - 624*A^3*a*b^9*c^3*d^8*f + 472*A^3*a*b^9*c^7*d^4*f - 272*A^3*a^3*b^7*c*d^10*f + 152*A^3*a^5*b^5*c*d^10*f - 22*A^3*a^9*b*c^3*d^8*f + 18*A^3*a*b^9*c^9*d^2*f - 13*A^3*a^2*b^8*c^10*d*f - 8*A^3*a^7*b^3*c*d^10*f - 8*A^3*a*b^9*c^5*d^6*f + A*B^2*a^8*b^2*d^11*f - C^3*b^10*c^8*d^3*f - 60*B^3*b^10*c^7*d^4*f - 32*B^3

*b^10*c^5*d^6*f + 21*B^3*b^10*c^9*d^2*f - 12*B^3*b^10*c^3*d^8*f - 3*C^3*a^1
0*c^2*d^9*f - 360*A^3*b^10*c^6*d^5*f - 204*A^3*b^10*c^4*d^7*f + 11*C^3*a^8*
b^2*d^11*f - 8*C^3*a^6*b^4*d^11*f - 4*C^3*a^4*b^6*d^11*f - B^3*a^10*c^3*d^8
*f - 64*B^3*a^5*b^5*d^11*f - 32*B^3*a^3*b^7*d^11*f + 3*A^3*a^10*c^2*d^9*f -
 68*A^3*a^4*b^6*d^11*f + 20*A^3*a^6*b^4*d^11*f + 12*A^3*a^2*b^8*d^11*f - B^
3*a^2*b^8*c^11*f + 3*C^3*b^10*c^10*d*f + 3*B^3*a^10*c*d^10*f - 3*A^3*b^10*c
^10*d*f - 2*C^3*a*b^9*c^11*f - 2*B^3*a^9*b*d^11*f + 2*A^3*a*b^9*c^11*f - 36
*A^2*C*b^10*d^11*f + 3*A^2*C*a^10*d^11*f - 3*A*C^2*a^10*d^11*f - A*B^2*a^10
*d^11*f + 36*A^3*b^10*d^11*f - A^3*a^10*d^11*f + A^3*b^10*c^8*d^3*f + A^3*a
^8*b^2*d^11*f + B^2*C*a^10*d^11*f + B*C^2*b^10*c^11*f + A^2*B*b^10*c^11*f +
 C^3*a^10*d^11*f + B^3*b^10*c^11*f - 6*A*B^2*C*a*b^7*c^7*d + 4*A*B^2*C*a*b^
7*c*d^7 + 168*A^2*B*C*a^3*b^5*c^2*d^6 + 144*A*B*C^2*a^4*b^4*c^3*d^5 - 129*A
^2*B*C*a^4*b^4*c^3*d^5 - 96*A*B*C^2*a^3*b^5*c^2*d^6 + 84*A*B*C^2*a^2*b^6*c^
3*d^5 + 72*A^2*B*C*a^3*b^5*c^4*d^4 - 72*A^2*B*C*a^2*b^6*c^3*d^5 + 64*A*B^2*
C*a^4*b^4*c^4*d^4 - 60*A*B*C^2*a^3*b^5*c^4*d^4 + 57*A^2*B*C*a^2*b^6*c^5*d^3
 - 56*A*B^2*C*a^3*b^5*c^5*d^3 - 39*A*B^2*C*a^4*b^4*c^2*d^6 - 38*A*B^2*C*a^5
*b^3*c^3*d^5 + 36*A*B^2*C*a^3*b^5*c^3*d^5 + 36*A*B*C^2*a^4*b^4*c^5*d^3 - 30
*A*B*C^2*a^2*b^6*c^5*d^3 + 27*A*B^2*C*a^2*b^6*c^6*d^2 - 24*A*B^2*C*a^2*b^6*
c^2*d^6 - 24*A*B*C^2*a^5*b^3*c^4*d^4 + 24*A*B*C^2*a^3*b^5*c^6*d^2 + 18*A^2*
B*C*a^5*b^3*c^2*d^6 - 18*A^2*B*C*a^4*b^4*c^5*d^3 - 15*A*B^2*C*a^2*b^6*c^4*d
^4 + 12*A^2*B*C*a^5*b^3*c^4*d^4 - 12*A^2*B*C*a^3*b^5*c^6*d^2 + 9*A*B^2*C*a^
6*b^2*c^2*d^6 + 6*A*B*C^2*a^6*b^2*c^3*d^5 - 3*A^2*B*C*a^6*b^2*c^3*d^5 + 60*
A^2*B*C*a*b^7*c^2*d^6 - 51*A^2*B*C*a^4*b^4*c*d^7 + 48*A*B*C^2*a*b^7*c^6*d^2
 - 42*A^2*B*C*a^2*b^6*c*d^7 - 42*A^2*B*C*a*b^7*c^6*d^2 + 36*A*B*C^2*a^4*b^4
*c*d^7 + 36*A*B*C^2*a^2*b^6*c*d^7 + 36*A*B*C^2*a*b^7*c^4*d^4 - 30*A^2*B*C*a
*b^7*c^4*d^4 + 24*A*B^2*C*a*b^7*c^3*d^5 - 24*A*B*C^2*a*b^7*c^2*d^6 + 18*A*B
^2*C*a^5*b^3*c*d^7 - 18*A*B*C^2*a^6*b^2*c*d^7 + 12*A*B^2*C*a^3*b^5*c*d^7 +
9*A^2*B*C*a^6*b^2*c*d^7 + 6*A*B^2*C*a*b^7*c^5*d^3 - 6*A*B*C^2*a^2*b^6*c^7*d
 + 3*A^2*B*C*a^2*b^6*c^7*d - 18*B^3*C*a*b^7*c^6*d^2 - 18*B*C^3*a*b^7*c^6*d^
2 - 14*B^3*C*a*b^7*c^4*d^4 - 14*B*C^3*a*b^7*c^4*d^4 - 10*B^3*C*a^2*b^6*c*d^
7 - 10*B*C^3*a^2*b^6*c*d^7 + 9*B^3*C*a^6*b^2*c*d^7 + 9*B*C^3*a^6*b^2*c*d^7
 - 7*B^3*C*a^4*b^4*c*d^7 - 7*B*C^3*a^4*b^4*c*d^7 + 6*B^2*C^2*a*b^7*c^7*d - 4
*B^3*C*a*b^7*c^2*d^6 + 4*B^2*C^2*a*b^7*c*d^7 - 4*B*C^3*a*b^7*c^2*d^6 + 3*B^
3*C*a^2*b^6*c^7*d + 3*B*C^3*a^2*b^6*c^7*d + 144*A^3*C*a*b^7*c^3*d^5 + 62*A^
3*C*a*b^7*c^5*d^3 + 48*A*C^3*a*b^7*c^3*d^5 - 36*A^2*C^2*a*b^7*c*d^7 + 26*A*
C^3*a*b^7*c^5*d^3 + 20*A^3*C*a^3*b^5*c*d^7 + 18*A^2*C^2*a*b^7*c^7*d - 18*A*
C^3*a^5*b^3*c*d^7 - 6*A^3*C*a^5*b^3*c*d^7 - 4*A*C^3*a^3*b^5*c*d^7 - 32*A^3*
B*a*b^7*c^2*d^6 - 32*A*B^3*a*b^7*c^2*d^6 + 22*A^3*B*a^4*b^4*c*d^7 + 22*A*B^
3*a^4*b^4*c*d^7 + 16*A^3*B*a^2*b^6*c*d^7 + 16*A*B^3*a^2*b^6*c*d^7 + 12*A^3*
B*a*b^7*c^6*d^2 + 12*A*B^3*a*b^7*c^6*d^2 + 8*A^3*B*a*b^7*c^4*d^4 - 8*A^2*B^
2*a*b^7*c*d^7 + 8*A*B^3*a*b^7*c^4*d^4 + 57*A^2*B*C*b^8*c^5*d^3 + 36*A^2*B*C
*b^8*c^3*d^5 - 30*A*B*C^2*b^8*c^5*d^3 - 18*A*B*C^2*b^8*c^3*d^5 - 9*A*B^2*C*
b^8*c^4*d^4 - 3*A*B^2*C*b^8*c^6*d^2 - 2*A*B^2*C*b^8*c^2*d^6 + 36*A^2*B*C*a^
3*b^5*d^8 + 24*A*B*C^2*a^5*b^3*d^8 - 18*A^2*B*C*a^5*b^3*d^8 - 12*A*B*C^2*a^
3*b^5*d^8 - 3*A*B^2*C*a^6*b^2*d^8 - 3*A*B^2*C*a^4*b^4*d^8 - 2*A*B^2*C*a^2*b

^6*d^8 + 34*B^2*C^2*a^5*b^3*c^3*d^5 + 28*B^2*C^2*a^3*b^5*c^5*d^3 + 24*B^2*C
^2*a^4*b^4*c^2*d^6 - 20*B^2*C^2*a^4*b^4*c^4*d^4 + 12*B^2*C^2*a^3*b^5*c^3*d^
5 + 12*B^2*C^2*a^2*b^6*c^2*d^6 - 9*B^2*C^2*a^6*b^2*c^2*d^6 + 9*B^2*C^2*a^4*
b^4*c^6*d^2 + 9*B^2*C^2*a^2*b^6*c^4*d^4 - 3*B^2*C^2*a^2*b^6*c^6*d^2 + 159*A
^2*C^2*a^2*b^6*c^4*d^4 - 156*A^2*C^2*a^3*b^5*c^3*d^5 + 90*A^2*C^2*a^5*b^3*c
^3*d^5 + 78*A^2*C^2*a^2*b^6*c^2*d^6 - 63*A^2*C^2*a^4*b^4*c^4*d^4 - 27*A^2*C
^2*a^6*b^2*c^2*d^6 - 27*A^2*C^2*a^2*b^6*c^6*d^2 - 18*A^2*C^2*a^4*b^4*c^2*d^
6 + 9*A^2*C^2*a^4*b^4*c^6*d^2 + 66*A^2*B^2*a^2*b^6*c^2*d^6 + 60*A^2*B^2*a^2
*b^6*c^4*d^4 - 48*A^2*B^2*a^3*b^5*c^3*d^5 + 42*A^2*B^2*a^4*b^4*c^2*d^6 + 28
*A^2*B^2*a^3*b^5*c^5*d^3 - 17*A^2*B^2*a^4*b^4*c^4*d^4 - 6*A^2*B^2*a^2*b^6*c
^6*d^2 + 4*A^2*B^2*a^5*b^3*c^3*d^5 + 36*A^3*C*a*b^7*c*d^7 - 18*A*C^3*a*b^7*
c^7*d + 12*A*C^3*a*b^7*c*d^7 - 6*A^3*C*a*b^7*c^7*d + 12*A^2*B*C*b^8*c*d^7 +
 6*A*B*C^2*b^8*c^7*d - 6*A*B*C^2*b^8*c*d^7 - 3*A^2*B*C*b^8*c^7*d + 24*A^2*B
*C*a*b^7*d^8 - 12*A*B*C^2*a*b^7*d^8 - 53*B^3*C*a^4*b^4*c^3*d^5 - 53*B*C^3*a
^4*b^4*c^3*d^5 - 32*B^3*C*a^2*b^6*c^3*d^5 - 32*B*C^3*a^2*b^6*c^3*d^5 - 18*B
^3*C*a^4*b^4*c^5*d^3 - 18*B*C^3*a^4*b^4*c^5*d^3 + 16*B^3*C*a^3*b^5*c^4*d^4
 + 16*B*C^3*a^3*b^5*c^4*d^4 + 12*B^3*C*a^5*b^3*c^4*d^4 - 12*B^3*C*a^3*b^5*c^
6*d^2 + 12*B^2*C^2*a*b^7*c^3*d^5 + 12*B*C^3*a^5*b^3*c^4*d^4 - 12*B*C^3*a^3*
b^5*c^6*d^2 + 8*B^3*C*a^3*b^5*c^2*d^6 + 8*B*C^3*a^3*b^5*c^2*d^6 - 6*B^3*C*a
^5*b^3*c^2*d^6 - 6*B^2*C^2*a^5*b^3*c*d^7 + 6*B^2*C^2*a*b^7*c^5*d^3 - 6*B*C^
3*a^5*b^3*c^2*d^6 - 3*B^3*C*a^6*b^2*c^3*d^5 - 3*B*C^3*a^6*b^2*c^3*d^5 - 175
*A^3*C*a^2*b^6*c^4*d^4 + 164*A^3*C*a^3*b^5*c^3*d^5 - 144*A^2*C^2*a*b^7*c^3*
d^5 - 124*A^3*C*a^2*b^6*c^2*d^6 - 90*A*C^3*a^5*b^3*c^3*d^5 - 73*A*C^3*a^2*b
^6*c^4*d^4 - 66*A^2*C^2*a*b^7*c^5*d^3 + 44*A*C^3*a^3*b^5*c^3*d^5 + 36*A*C^3
*a^4*b^4*c^4*d^4 - 30*A^3*C*a^5*b^3*c^3*d^5 + 30*A^3*C*a^4*b^4*c^4*d^4 + 27
*A*C^3*a^6*b^2*c^2*d^6 + 21*A*C^3*a^4*b^4*c^2*d^6 + 18*A^2*C^2*a^5*b^3*c*d^
7 - 18*A*C^3*a^4*b^4*c^6*d^2 - 16*A*C^3*a^2*b^6*c^2*d^6 - 15*A^3*C*a^4*b^4*
c^2*d^6 + 15*A^3*C*a^2*b^6*c^6*d^2 - 12*A^2*C^2*a^3*b^5*c*d^7 + 9*A^3*C*a^6
*b^2*c^2*d^6 + 9*A*C^3*a^2*b^6*c^6*d^2 - 80*A^3*B*a^3*b^5*c^2*d^6 - 80*A*B^
3*a^3*b^5*c^2*d^6 + 38*A^3*B*a^4*b^4*c^3*d^5 + 38*A*B^3*a^4*b^4*c^3*d^5 - 3
6*A^2*B^2*a*b^7*c^3*d^5 - 28*A^3*B*a^3*b^5*c^4*d^4 - 28*A^3*B*a^2*b^6*c^5*d
^3 - 28*A*B^3*a^3*b^5*c^4*d^4 - 28*A*B^3*a^2*b^6*c^5*d^3 + 20*A^3*B*a^2*b^6
*c^3*d^5 + 20*A*B^3*a^2*b^6*c^3*d^5 - 12*A^3*B*a^5*b^3*c^2*d^6 - 12*A^2*B^2
*a^5*b^3*c*d^7 - 12*A^2*B^2*a^3*b^5*c*d^7 - 12*A^2*B^2*a*b^7*c^5*d^3 - 12*A
*B^3*a^5*b^3*c^2*d^6 + 6*B^2*C^2*b^8*c^6*d^2 + 3*B^2*C^2*b^8*c^4*d^4 + 36*A
^2*C^2*b^8*c^4*d^4 + 27*A^2*C^2*b^8*c^2*d^6 - 18*A^2*C^2*b^8*c^6*d^2 + 33*A
^2*B^2*b^8*c^4*d^4 + 28*A^2*B^2*b^8*c^2*d^6 + 9*B^2*C^2*a^4*b^4*d^8 + 6*A^2
*B^2*b^8*c^6*d^2 + 4*B^2*C^2*a^2*b^6*d^8 + 3*B^2*C^2*a^6*b^2*d^8 - 30*A^2*C
^2*a^4*b^4*d^8 + 9*A^2*C^2*a^6*b^2*d^8 + 16*A^2*B^2*a^2*b^6*d^8 + 3*A^2*B^2
*a^4*b^4*d^8 + 6*C^4*a^5*b^3*c*d^7 + 4*C^4*a^3*b^5*c*d^7 - 2*C^4*a*b^7*c^5*
d^3 - 12*B^4*a^5*b^3*c*d^7 + 12*B^4*a*b^7*c^3*d^5 + 8*B^4*a*b^7*c^5*d^3 - 4
*B^4*a^3*b^5*c*d^7 - 48*A^4*a*b^7*c^3*d^5 - 20*A^4*a*b^7*c^5*d^3 - 8*A^4*a^
3*b^5*c*d^7 - 63*A^3*C*b^8*c^4*d^4 - 54*A^3*C*b^8*c^2*d^6 + 9*A^3*C*b^8*c^6
*d^2 + 9*A*C^3*b^8*c^6*d^2 - 3*A*C^3*b^8*c^4*d^4 - 28*A^3*B*b^8*c^5*d^3 - 2
8*A*B^3*b^8*c^5*d^3 - 18*A^3*B*b^8*c^3*d^5 - 18*A*B^3*b^8*c^3*d^5 - 10*B^3*

C*a^5*b^3*d^8 - 10*B*C^3*a^5*b^3*d^8 - 4*B^3*C*a^3*b^5*d^8 - 4*B*C^3*a^3*b^
5*d^8 + 23*A^3*C*a^4*b^4*d^8 - 18*A^3*C*a^2*b^6*d^8 + 11*A*C^3*a^4*b^4*d^8
- 9*A*C^3*a^6*b^2*d^8 + 6*A*C^3*a^2*b^6*d^8 - 3*A^3*C*a^6*b^2*d^8 - 20*A^3*
B*a^3*b^5*d^8 - 20*A*B^3*a^3*b^5*d^8 + 4*A^3*B*a^5*b^3*d^8 + 4*A*B^3*a^5*b^
3*d^8 + B^3*C*a^2*b^6*c^5*d^3 + B*C^3*a^2*b^6*c^5*d^3 + 6*C^4*a*b^7*c^7*d +
4*B^4*a*b^7*c*d^7 - 12*A^4*a*b^7*c*d^7 - 3*B^3*C*b^8*c^7*d - 3*B*C^3*b^8*c
^7*d - 6*A^3*B*b^8*c*d^7 - 6*A*B^3*b^8*c*d^7 - 12*A^3*B*a*b^7*d^8 - 12*A*B^
3*a*b^7*d^8 + 30*C^4*a^5*b^3*c^3*d^5 + 19*C^4*a^2*b^6*c^4*d^4 - 9*C^4*a^6*b
^2*c^2*d^6 + 9*C^4*a^4*b^4*c^6*d^2 + 4*C^4*a^3*b^5*c^3*d^5 + 4*C^4*a^2*b^6*
c^2*d^6 - 3*C^4*a^4*b^4*c^4*d^4 - 3*C^4*a^4*b^4*c^2*d^6 + 3*C^4*a^2*b^6*c^6
*d^2 + 28*B^4*a^3*b^5*c^5*d^3 + 27*B^4*a^4*b^4*c^2*d^6 - 17*B^4*a^4*b^4*c^4
*d^4 - 10*B^4*a^2*b^6*c^4*d^4 + 8*B^4*a^3*b^5*c^3*d^5 + 8*B^4*a^2*b^6*c^2*d
^6 - 6*B^4*a^2*b^6*c^6*d^2 + 4*B^4*a^5*b^3*c^3*d^5 + 70*A^4*a^2*b^6*c^4*d^4
+ 58*A^4*a^2*b^6*c^2*d^6 - 56*A^4*a^3*b^5*c^3*d^5 + 15*A^4*a^4*b^4*c^2*d^6
+ B^2*C^2*b^8*c^2*d^6 - 18*A^3*C*b^8*d^8 + B^3*C*b^8*c^5*d^3 + B*C^3*b^8*c
^5*d^3 + 6*B^4*b^8*c^6*d^2 + 3*B^4*b^8*c^4*d^4 + 30*A^4*b^8*c^4*d^4 + 27*A^
4*b^8*c^2*d^6 + 3*C^4*a^6*b^2*d^8 + 8*B^4*a^4*b^4*d^8 + 4*B^4*a^2*b^6*d^8 +
12*A^4*a^2*b^6*d^8 - 5*A^4*a^4*b^4*d^8 + 9*A^2*C^2*b^8*d^8 + 9*A^2*B^2*b^8
*d^8 + 9*A^4*b^8*d^8 + B^4*b^8*c^2*d^6 + C^4*a^4*b^4*d^8, f, k)*(root(640*a
^15*b*c^7*d^13*f^4 + 640*a*b^15*c^13*d^7*f^4 + 480*a^15*b*c^9*d^11*f^4 + 48
0*a^15*b*c^5*d^15*f^4 + 480*a*b^15*c^15*d^5*f^4 + 480*a*b^15*c^11*d^9*f^4 +
192*a^15*b*c^11*d^9*f^4 + 192*a^15*b*c^3*d^17*f^4 + 192*a^11*b^5*c*d^19*f^
4 + 192*a^5*b^11*c^19*d*f^4 + 192*a*b^15*c^17*d^3*f^4 + 192*a*b^15*c^9*d^11
*f^4 + 128*a^13*b^3*c*d^19*f^4 + 128*a^9*b^7*c*d^19*f^4 + 128*a^7*b^9*c^19*
d*f^4 + 128*a^3*b^13*c^19*d*f^4 + 32*a^15*b*c^13*d^7*f^4 + 32*a^9*b^7*c^19*
d*f^4 + 32*a^7*b^9*c*d^19*f^4 + 32*a*b^15*c^7*d^13*f^4 + 32*a^15*b*c*d^19*f
^4 + 32*a*b^15*c^19*d*f^4 - 47088*a^8*b^8*c^10*d^10*f^4 + 42432*a^9*b^7*c^9
*d^11*f^4 + 42432*a^7*b^9*c^11*d^9*f^4 + 39328*a^9*b^7*c^11*d^9*f^4 + 39328
*a^7*b^9*c^9*d^11*f^4 - 36912*a^8*b^8*c^12*d^8*f^4 - 36912*a^8*b^8*c^8*d^12
*f^4 - 34256*a^10*b^6*c^10*d^10*f^4 - 34256*a^6*b^10*c^10*d^10*f^4 - 31152*
a^10*b^6*c^8*d^12*f^4 - 31152*a^6*b^10*c^12*d^8*f^4 + 28128*a^9*b^7*c^7*d^1
3*f^4 + 28128*a^7*b^9*c^13*d^7*f^4 + 24160*a^11*b^5*c^9*d^11*f^4 + 24160*a^
5*b^11*c^11*d^9*f^4 - 23088*a^10*b^6*c^12*d^8*f^4 - 23088*a^6*b^10*c^8*d^12
*f^4 + 22272*a^9*b^7*c^13*d^7*f^4 + 22272*a^7*b^9*c^7*d^13*f^4 + 19072*a^11
*b^5*c^11*d^9*f^4 + 19072*a^5*b^11*c^9*d^11*f^4 + 18624*a^11*b^5*c^7*d^13*f
^4 + 18624*a^5*b^11*c^13*d^7*f^4 - 17328*a^8*b^8*c^14*d^6*f^4 - 17328*a^8*b
^8*c^6*d^14*f^4 - 17232*a^10*b^6*c^6*d^14*f^4 - 17232*a^6*b^10*c^14*d^6*f^4
- 13520*a^12*b^4*c^8*d^12*f^4 - 13520*a^4*b^12*c^12*d^8*f^4 - 12464*a^12*b
^4*c^10*d^10*f^4 - 12464*a^4*b^12*c^10*d^10*f^4 + 10880*a^9*b^7*c^5*d^15*f^
4 + 10880*a^7*b^9*c^15*d^5*f^4 - 9072*a^10*b^6*c^14*d^6*f^4 - 9072*a^6*b^10
*c^6*d^14*f^4 + 8928*a^11*b^5*c^13*d^7*f^4 + 8928*a^5*b^11*c^7*d^13*f^4 - 8
880*a^12*b^4*c^6*d^14*f^4 - 8880*a^4*b^12*c^14*d^6*f^4 + 8480*a^11*b^5*c^5*
d^15*f^4 + 8480*a^5*b^11*c^15*d^5*f^4 + 7200*a^9*b^7*c^15*d^5*f^4 + 7200*a^
7*b^9*c^5*d^15*f^4 - 6912*a^12*b^4*c^12*d^8*f^4 - 6912*a^4*b^12*c^8*d^12*f^
4 + 6400*a^13*b^3*c^9*d^11*f^4 + 6400*a^3*b^13*c^11*d^9*f^4 + 5920*a^13*b^3

*c^7*d^13*f^4 + 5920*a^3*b^13*c^13*d^7*f^4 - 5392*a^10*b^6*c^4*d^16*f^4 - 5
392*a^6*b^10*c^16*d^4*f^4 - 4428*a^8*b^8*c^16*d^4*f^4 - 4428*a^8*b^8*c^4*d^
16*f^4 + 4128*a^13*b^3*c^11*d^9*f^4 + 4128*a^3*b^13*c^9*d^11*f^4 - 3328*a^1
2*b^4*c^4*d^16*f^4 - 3328*a^4*b^12*c^16*d^4*f^4 + 3264*a^13*b^3*c^5*d^15*f^
4 + 3264*a^3*b^13*c^15*d^5*f^4 - 2480*a^14*b^2*c^8*d^12*f^4 - 2480*a^2*b^14
*c^12*d^8*f^4 + 2240*a^11*b^5*c^15*d^5*f^4 + 2240*a^5*b^11*c^5*d^15*f^4 - 2
128*a^12*b^4*c^14*d^6*f^4 - 2128*a^4*b^12*c^6*d^14*f^4 + 2112*a^9*b^7*c^3*d
^17*f^4 + 2112*a^7*b^9*c^17*d^3*f^4 + 2048*a^11*b^5*c^3*d^17*f^4 + 2048*a^5
*b^11*c^17*d^3*f^4 - 2000*a^14*b^2*c^6*d^14*f^4 - 2000*a^2*b^14*c^14*d^6*f^
4 - 1792*a^10*b^6*c^16*d^4*f^4 - 1792*a^6*b^10*c^4*d^16*f^4 - 1776*a^14*b^2
*c^10*d^10*f^4 - 1776*a^2*b^14*c^10*d^10*f^4 + 1472*a^13*b^3*c^13*d^7*f^4 +
1472*a^3*b^13*c^7*d^13*f^4 + 1088*a^9*b^7*c^17*d^3*f^4 + 1088*a^7*b^9*c^3*
d^17*f^4 + 992*a^13*b^3*c^3*d^17*f^4 + 992*a^3*b^13*c^17*d^3*f^4 - 912*a^14
*b^2*c^4*d^16*f^4 - 912*a^2*b^14*c^16*d^4*f^4 - 768*a^10*b^6*c^2*d^18*f^4 -
768*a^6*b^10*c^18*d^2*f^4 - 688*a^14*b^2*c^12*d^8*f^4 - 688*a^2*b^14*c^8*d
^12*f^4 - 592*a^12*b^4*c^2*d^18*f^4 - 592*a^4*b^12*c^18*d^2*f^4 - 472*a^8*b
^8*c^18*d^2*f^4 - 472*a^8*b^8*c^2*d^18*f^4 - 280*a^12*b^4*c^16*d^4*f^4 - 28
0*a^4*b^12*c^4*d^16*f^4 + 224*a^13*b^3*c^15*d^5*f^4 + 224*a^11*b^5*c^17*d^3
*f^4 + 224*a^5*b^11*c^3*d^17*f^4 + 224*a^3*b^13*c^5*d^15*f^4 - 208*a^14*b^2
*c^2*d^18*f^4 - 208*a^2*b^14*c^18*d^2*f^4 - 112*a^14*b^2*c^14*d^6*f^4 - 112
*a^10*b^6*c^18*d^2*f^4 - 112*a^6*b^10*c^2*d^18*f^4 - 112*a^2*b^14*c^6*d^14*
f^4 - 80*b^16*c^14*d^6*f^4 - 60*b^16*c^16*d^4*f^4 - 60*b^16*c^12*d^8*f^4 -
24*b^16*c^18*d^2*f^4 - 24*b^16*c^10*d^10*f^4 - 4*b^16*c^8*d^12*f^4 - 80*a^1
6*c^6*d^14*f^4 - 60*a^16*c^8*d^12*f^4 - 60*a^16*c^4*d^16*f^4 - 24*a^16*c^10
*d^10*f^4 - 24*a^16*c^2*d^18*f^4 - 4*a^16*c^12*d^8*f^4 - 24*a^12*b^4*d^20*f
^4 - 16*a^14*b^2*d^20*f^4 - 16*a^10*b^6*d^20*f^4 - 4*a^8*b^8*d^20*f^4 - 24*
a^4*b^12*c^20*f^4 - 16*a^6*b^10*c^20*f^4 - 16*a^2*b^14*c^20*f^4 - 4*a^8*b^8
*c^20*f^4 - 4*b^16*c^20*f^4 - 4*a^16*d^20*f^4 + 56*A*C*a*b^11*c^13*d*f^2 -
48*A*C*a^11*b*c*d^13*f^2 + 48*A*C*a*b^11*c*d^13*f^2 + 5904*B*C*a^6*b^6*c^7*
d^7*f^2 - 5016*B*C*a^5*b^7*c^8*d^6*f^2 - 4608*B*C*a^7*b^5*c^6*d^8*f^2 - 451
2*B*C*a^5*b^7*c^6*d^8*f^2 - 4384*B*C*a^7*b^5*c^8*d^6*f^2 + 3056*B*C*a^8*b^4
*c^7*d^7*f^2 + 2256*B*C*a^4*b^8*c^7*d^7*f^2 - 1824*B*C*a^3*b^9*c^8*d^6*f^2
+ 1632*B*C*a^9*b^3*c^4*d^10*f^2 - 1400*B*C*a^8*b^4*c^3*d^11*f^2 - 1320*B*C*
a^4*b^8*c^11*d^3*f^2 - 1248*B*C*a^3*b^9*c^6*d^8*f^2 + 1152*B*C*a^3*b^9*c^10
*d^4*f^2 - 1072*B*C*a^9*b^3*c^6*d^8*f^2 + 1068*B*C*a^6*b^6*c^9*d^5*f^2 - 10
04*B*C*a^4*b^8*c^5*d^9*f^2 - 968*B*C*a^6*b^6*c^3*d^11*f^2 - 864*B*C*a^8*b^4
*c^5*d^9*f^2 - 828*B*C*a^4*b^8*c^9*d^5*f^2 - 792*B*C*a^4*b^8*c^3*d^11*f^2 -
792*B*C*a^2*b^10*c^11*d^3*f^2 - 776*B*C*a^9*b^3*c^8*d^6*f^2 + 688*B*C*a^7*
b^5*c^4*d^10*f^2 - 672*B*C*a^10*b^2*c^3*d^11*f^2 - 592*B*C*a^2*b^10*c^9*d^5
*f^2 + 544*B*C*a^10*b^2*c^7*d^7*f^2 - 492*B*C*a^2*b^10*c^5*d^9*f^2 + 480*B*
C*a^5*b^7*c^10*d^4*f^2 - 392*B*C*a^10*b^2*c^5*d^9*f^2 + 332*B*C*a^8*b^4*c^9
*d^5*f^2 - 328*B*C*a^6*b^6*c^11*d^3*f^2 + 320*B*C*a^9*b^3*c^2*d^12*f^2 + 27
2*B*C*a^3*b^9*c^12*d^2*f^2 - 248*B*C*a^5*b^7*c^4*d^10*f^2 - 248*B*C*a^2*b^1
0*c^3*d^11*f^2 - 208*B*C*a^7*b^5*c^10*d^4*f^2 - 192*B*C*a^5*b^7*c^2*d^12*f^
2 + 144*B*C*a^2*b^10*c^7*d^7*f^2 - 96*B*C*a^3*b^9*c^4*d^10*f^2 + 88*B*C*a^5

*b^7*c^12*d^2*f^2 - 72*B*C*a^8*b^4*c^11*d^3*f^2 + 48*B*C*a^9*b^3*c^10*d^4*f
^2 - 48*B*C*a^7*b^5*c^12*d^2*f^2 - 48*B*C*a^7*b^5*c^2*d^12*f^2 - 48*B*C*a^3
*b^9*c^2*d^12*f^2 - 12*B*C*a^10*b^2*c^9*d^5*f^2 + 4*B*C*a^6*b^6*c^5*d^9*f^2
 + 5824*A*C*a^7*b^5*c^5*d^9*f^2 - 4378*A*C*a^8*b^4*c^6*d^8*f^2 + 4296*A*C*a
^5*b^7*c^5*d^9*f^2 - 3912*A*C*a^6*b^6*c^6*d^8*f^2 - 3672*A*C*a^5*b^7*c^9*d^
5*f^2 + 3594*A*C*a^4*b^8*c^8*d^6*f^2 + 3236*A*C*a^6*b^6*c^8*d^6*f^2 + 2816*
A*C*a^9*b^3*c^5*d^9*f^2 + 2624*A*C*a^3*b^9*c^5*d^9*f^2 + 2432*A*C*a^7*b^5*c
^7*d^7*f^2 - 2366*A*C*a^8*b^4*c^4*d^10*f^2 + 2298*A*C*a^4*b^8*c^10*d^4*f^2
+ 1872*A*C*a^3*b^9*c^7*d^7*f^2 + 1848*A*C*a^6*b^6*c^10*d^4*f^2 - 1644*A*C*a
^6*b^6*c^4*d^10*f^2 - 1488*A*C*a^7*b^5*c^9*d^5*f^2 - 1408*A*C*a^3*b^9*c^9*d
^5*f^2 - 1308*A*C*a^4*b^8*c^6*d^8*f^2 + 1248*A*C*a^5*b^7*c^7*d^7*f^2 - 1012
*A*C*a^10*b^2*c^6*d^8*f^2 + 1008*A*C*a^7*b^5*c^3*d^11*f^2 + 992*A*C*a^5*b^7
*c^3*d^11*f^2 + 928*A*C*a^3*b^9*c^3*d^11*f^2 + 848*A*C*a^9*b^3*c^7*d^7*f^2
+ 636*A*C*a^2*b^10*c^8*d^6*f^2 - 628*A*C*a^10*b^2*c^4*d^10*f^2 - 600*A*C*a^
2*b^10*c^6*d^8*f^2 - 576*A*C*a^5*b^7*c^11*d^3*f^2 + 572*A*C*a^2*b^10*c^10*d
^4*f^2 + 464*A*C*a^8*b^4*c^8*d^6*f^2 + 304*A*C*a^6*b^6*c^2*d^12*f^2 - 304*A
*C*a^4*b^8*c^4*d^10*f^2 + 296*A*C*a^4*b^8*c^2*d^12*f^2 + 260*A*C*a^8*b^4*c^
10*d^4*f^2 - 232*A*C*a^9*b^3*c^9*d^5*f^2 - 232*A*C*a^2*b^10*c^12*d^2*f^2 +
228*A*C*a^10*b^2*c^2*d^12*f^2 - 188*A*C*a^2*b^10*c^4*d^10*f^2 + 144*A*C*a^3
*b^9*c^11*d^3*f^2 + 116*A*C*a^6*b^6*c^12*d^2*f^2 + 112*A*C*a^9*b^3*c^3*d^11
*f^2 - 112*A*C*a^7*b^5*c^11*d^3*f^2 + 92*A*C*a^10*b^2*c^8*d^6*f^2 + 74*A*C*
a^4*b^8*c^12*d^2*f^2 + 62*A*C*a^8*b^4*c^2*d^12*f^2 + 40*A*C*a^2*b^10*c^2*d^
12*f^2 - 7008*A*B*a^6*b^6*c^7*d^7*f^2 - 4032*A*B*a^4*b^8*c^7*d^7*f^2 + 3952
*A*B*a^7*b^5*c^8*d^6*f^2 + 3648*A*B*a^5*b^7*c^8*d^6*f^2 - 3392*A*B*a^8*b^4*
c^7*d^7*f^2 + 3264*A*B*a^7*b^5*c^6*d^8*f^2 - 2992*A*B*a^5*b^7*c^4*d^10*f^2
- 2368*A*B*a^7*b^5*c^4*d^10*f^2 - 2304*A*B*a^3*b^9*c^4*d^10*f^2 - 1968*A*B*
a^6*b^6*c^9*d^5*f^2 - 1872*A*B*a^9*b^3*c^4*d^10*f^2 - 1728*A*B*a^2*b^10*c^7
*d^7*f^2 + 1712*A*B*a^8*b^4*c^3*d^11*f^2 + 1536*A*B*a^5*b^7*c^6*d^8*f^2 - 1
536*A*B*a^3*b^9*c^10*d^4*f^2 - 1392*A*B*a^5*b^7*c^2*d^12*f^2 + 1328*A*B*a^6
*b^6*c^3*d^11*f^2 - 1104*A*B*a^3*b^9*c^2*d^12*f^2 - 1056*A*B*a^3*b^9*c^6*d^
8*f^2 + 976*A*B*a^9*b^3*c^6*d^8*f^2 + 960*A*B*a^4*b^8*c^11*d^3*f^2 + 936*A*
B*a^8*b^4*c^5*d^9*f^2 - 912*A*B*a^5*b^7*c^10*d^4*f^2 + 848*A*B*a^9*b^3*c^8*
d^6*f^2 - 816*A*B*a^7*b^5*c^2*d^12*f^2 + 816*A*B*a^4*b^8*c^3*d^11*f^2 + 768
*A*B*a^10*b^2*c^3*d^11*f^2 + 672*A*B*a^3*b^9*c^8*d^6*f^2 - 632*A*B*a^8*b^4*
c^9*d^5*f^2 - 608*A*B*a^2*b^10*c^9*d^5*f^2 - 552*A*B*a^4*b^8*c^9*d^5*f^2 -
544*A*B*a^10*b^2*c^7*d^7*f^2 - 480*A*B*a^2*b^10*c^5*d^9*f^2 + 464*A*B*a^10*
b^2*c^5*d^9*f^2 - 464*A*B*a^9*b^3*c^2*d^12*f^2 + 432*A*B*a^2*b^10*c^11*d^3*
f^2 - 368*A*B*a^3*b^9*c^12*d^2*f^2 - 256*A*B*a^6*b^6*c^5*d^9*f^2 - 208*A*B*
a^5*b^7*c^12*d^2*f^2 + 176*A*B*a^4*b^8*c^5*d^9*f^2 + 112*A*B*a^7*b^5*c^10*d
^4*f^2 + 112*A*B*a^6*b^6*c^11*d^3*f^2 - 16*A*B*a^2*b^10*c^3*d^11*f^2 - 576*
B*C*a*b^11*c^8*d^6*f^2 + 400*B*C*a^11*b*c^4*d^10*f^2 - 288*B*C*a*b^11*c^6*d
^8*f^2 - 176*B*C*a^11*b*c^6*d^8*f^2 + 128*B*C*a*b^11*c^10*d^4*f^2 - 108*B*C
*a^4*b^8*c*d^13*f^2 - 104*B*C*a*b^11*c^4*d^10*f^2 - 92*B*C*a^4*b^8*c^13*d*f
^2 - 60*B*C*a^8*b^4*c*d^13*f^2 - 60*B*C*a^6*b^6*c*d^13*f^2 + 48*B*C*a^11*b*
c^2*d^12*f^2 - 40*B*C*a^2*b^10*c*d^13*f^2 - 28*B*C*a^2*b^10*c^13*d*f^2 - 24

*B*C*a*b^11*c^12*d^2*f^2 + 20*B*C*a^10*b^2*c*d^13*f^2 - 16*B*C*a*b^11*c^2*d
^12*f^2 + 12*B*C*a^6*b^6*c^13*d*f^2 + 912*A*C*a*b^11*c^7*d^7*f^2 + 808*A*C*
a*b^11*c^5*d^9*f^2 + 432*A*C*a^11*b*c^5*d^9*f^2 + 336*A*C*a*b^11*c^3*d^11*f
^2 + 224*A*C*a*b^11*c^11*d^3*f^2 - 112*A*C*a^11*b*c^3*d^11*f^2 + 112*A*C*a^
3*b^9*c*d^13*f^2 - 88*A*C*a^9*b^3*c*d^13*f^2 + 80*A*C*a^3*b^9*c^13*d*f^2 +
56*A*C*a^5*b^7*c*d^13*f^2 + 48*A*C*a*b^11*c^9*d^5*f^2 - 40*A*C*a^5*b^7*c^13
*d*f^2 - 16*A*C*a^11*b*c^7*d^7*f^2 + 16*A*C*a^7*b^5*c*d^13*f^2 - 496*A*B*a*
b^11*c^4*d^10*f^2 - 400*A*B*a^11*b*c^4*d^10*f^2 + 288*A*B*a*b^11*c^8*d^6*f^
2 - 288*A*B*a*b^11*c^6*d^8*f^2 - 272*A*B*a*b^11*c^2*d^12*f^2 + 240*A*B*a^6*
b^6*c*d^13*f^2 - 224*A*B*a*b^11*c^10*d^4*f^2 + 192*A*B*a^8*b^4*c*d^13*f^2 +
192*A*B*a^4*b^8*c*d^13*f^2 + 176*A*B*a^11*b*c^6*d^8*f^2 + 104*A*B*a^4*b^8*
c^13*d*f^2 - 48*A*B*a^11*b*c^2*d^12*f^2 + 16*A*B*a^10*b^2*c*d^13*f^2 + 16*A
*B*a^2*b^10*c^13*d*f^2 + 16*A*B*a^2*b^10*c*d^13*f^2 - 112*B*C*b^12*c^11*d^3
*f^2 + 4*B*C*b^12*c^5*d^9*f^2 + 150*A*C*b^12*c^10*d^4*f^2 - 80*B*C*a^12*c^3
*d^11*f^2 + 66*A*C*b^12*c^8*d^6*f^2 - 30*A*C*b^12*c^12*d^2*f^2 + 24*B*C*a^1
2*c^5*d^9*f^2 - 12*A*C*b^12*c^4*d^10*f^2 - 576*A*B*b^12*c^7*d^7*f^2 - 432*A
*B*b^12*c^9*d^5*f^2 - 400*A*B*b^12*c^5*d^9*f^2 - 144*A*B*b^12*c^3*d^11*f^2
- 96*B*C*a^7*b^5*d^14*f^2 - 72*B*C*a^5*b^7*d^14*f^2 - 66*A*C*a^12*c^4*d^10*
f^2 + 54*A*C*a^12*c^2*d^12*f^2 - 32*A*B*b^12*c^11*d^3*f^2 - 24*B*C*a^9*b^3*
d^14*f^2 - 16*B*C*a^3*b^9*d^14*f^2 + 2*A*C*a^12*c^6*d^8*f^2 + 116*A*C*a^6*b
^6*d^14*f^2 + 100*A*C*a^4*b^8*d^14*f^2 + 80*A*B*a^12*c^3*d^11*f^2 + 24*A*C*
a^2*b^10*d^14*f^2 - 24*A*B*a^12*c^5*d^9*f^2 + 22*A*C*a^8*b^4*d^14*f^2 + 16*
B*C*a^3*b^9*c^14*f^2 + 8*A*C*a^10*b^2*d^14*f^2 - 192*A*B*a^5*b^7*d^14*f^2 -
176*A*B*a^3*b^9*d^14*f^2 - 48*A*B*a^7*b^5*d^14*f^2 - 28*A*C*a^2*b^10*c^14*
f^2 + 2*A*C*a^4*b^8*c^14*f^2 - 16*A*B*a^3*b^9*c^14*f^2 + 2508*C^2*a^6*b^6*c
^6*d^8*f^2 + 2376*C^2*a^5*b^7*c^9*d^5*f^2 + 2357*C^2*a^8*b^4*c^6*d^8*f^2 -
2048*C^2*a^7*b^5*c^5*d^9*f^2 + 1304*C^2*a^3*b^9*c^9*d^5*f^2 + 1303*C^2*a^8*
b^4*c^4*d^10*f^2 + 1212*C^2*a^6*b^6*c^4*d^10*f^2 - 1203*C^2*a^4*b^8*c^8*d^6
*f^2 - 1192*C^2*a^9*b^3*c^5*d^9*f^2 + 1062*C^2*a^4*b^8*c^6*d^8*f^2 + 984*C^
2*a^7*b^5*c^9*d^5*f^2 - 952*C^2*a^6*b^6*c^8*d^6*f^2 + 768*C^2*a^5*b^7*c^7*d
^7*f^2 - 681*C^2*a^4*b^8*c^10*d^4*f^2 - 672*C^2*a^5*b^7*c^5*d^9*f^2 - 480*C
^2*a^6*b^6*c^10*d^4*f^2 + 458*C^2*a^10*b^2*c^6*d^8*f^2 - 448*C^2*a^7*b^5*c^
7*d^7*f^2 + 422*C^2*a^4*b^8*c^4*d^10*f^2 + 372*C^2*a^2*b^10*c^6*d^8*f^2 + 3
60*C^2*a^5*b^7*c^11*d^3*f^2 + 312*C^2*a^3*b^9*c^7*d^7*f^2 + 278*C^2*a^10*b^
2*c^4*d^10*f^2 - 232*C^2*a^9*b^3*c^7*d^7*f^2 + 194*C^2*a^2*b^10*c^12*d^2*f^
2 + 176*C^2*a^9*b^3*c^9*d^5*f^2 + 152*C^2*a^5*b^7*c^3*d^11*f^2 + 124*C^2*a^
2*b^10*c^4*d^10*f^2 - 120*C^2*a^7*b^5*c^3*d^11*f^2 - 114*C^2*a^10*b^2*c^2*d
^12*f^2 - 102*C^2*a^2*b^10*c^8*d^6*f^2 + 101*C^2*a^4*b^8*c^12*d^2*f^2 + 100
*C^2*a^6*b^6*c^2*d^12*f^2 - 88*C^2*a^3*b^9*c^5*d^9*f^2 + 77*C^2*a^8*b^4*c^2
*d^12*f^2 + 72*C^2*a^3*b^9*c^11*d^3*f^2 - 64*C^2*a^10*b^2*c^8*d^6*f^2 + 64*
C^2*a^3*b^9*c^3*d^11*f^2 - 58*C^2*a^2*b^10*c^10*d^4*f^2 + 56*C^2*a^7*b^5*c^
11*d^3*f^2 + 56*C^2*a^6*b^6*c^12*d^2*f^2 + 40*C^2*a^9*b^3*c^3*d^11*f^2 + 36
*C^2*a^8*b^4*c^12*d^2*f^2 + 32*C^2*a^4*b^8*c^2*d^12*f^2 + 26*C^2*a^8*b^4*c^
10*d^4*f^2 + 16*C^2*a^2*b^10*c^2*d^12*f^2 + 2*C^2*a^8*b^4*c^8*d^6*f^2 + 227
7*B^2*a^4*b^8*c^8*d^6*f^2 + 2144*B^2*a^7*b^5*c^5*d^9*f^2 - 2112*B^2*a^5*b^7

*c^9*d^5*f^2 + 2028*B^2*a^6*b^6*c^8*d^6*f^2 - 1671*B^2*a^8*b^4*c^6*d^8*f^2
+ 1275*B^2*a^4*b^8*c^10*d^4*f^2 + 1176*B^2*a^5*b^7*c^5*d^9*f^2 + 1096*B^2*a
^9*b^3*c^5*d^9*f^2 - 1044*B^2*a^6*b^6*c^6*d^8*f^2 + 984*B^2*a^6*b^6*c^10*d^
4*f^2 - 968*B^2*a^3*b^9*c^9*d^5*f^2 - 888*B^2*a^7*b^5*c^9*d^5*f^2 + 672*B^2
*a^7*b^5*c^7*d^7*f^2 + 664*B^2*a^3*b^9*c^5*d^9*f^2 - 649*B^2*a^8*b^4*c^4*d^
10*f^2 + 618*B^2*a^2*b^10*c^8*d^6*f^2 + 514*B^2*a^4*b^8*c^4*d^10*f^2 + 460*
B^2*a^6*b^6*c^2*d^12*f^2 + 422*B^2*a^8*b^4*c^8*d^6*f^2 + 406*B^2*a^2*b^10*c
^10*d^4*f^2 - 382*B^2*a^10*b^2*c^6*d^8*f^2 + 368*B^2*a^4*b^8*c^2*d^12*f^2 -
312*B^2*a^5*b^7*c^11*d^3*f^2 + 312*B^2*a^3*b^9*c^7*d^7*f^2 + 248*B^2*a^9*b
^3*c^7*d^7*f^2 + 245*B^2*a^8*b^4*c^2*d^12*f^2 - 192*B^2*a^5*b^7*c^7*d^7*f^2
- 184*B^2*a^9*b^3*c^3*d^11*f^2 + 182*B^2*a^10*b^2*c^2*d^12*f^2 + 176*B^2*a
^3*b^9*c^3*d^11*f^2 + 174*B^2*a^4*b^8*c^6*d^8*f^2 - 170*B^2*a^10*b^2*c^4*d^
10*f^2 - 152*B^2*a^9*b^3*c^9*d^5*f^2 + 152*B^2*a^2*b^10*c^4*d^10*f^2 + 142*
B^2*a^8*b^4*c^10*d^4*f^2 - 90*B^2*a^2*b^10*c^12*d^2*f^2 + 88*B^2*a^2*b^10*c
^2*d^12*f^2 + 84*B^2*a^10*b^2*c^8*d^6*f^2 + 84*B^2*a^2*b^10*c^6*d^8*f^2 + 6
0*B^2*a^6*b^6*c^12*d^2*f^2 - 56*B^2*a^7*b^5*c^11*d^3*f^2 + 53*B^2*a^4*b^8*c
^12*d^2*f^2 + 24*B^2*a^7*b^5*c^3*d^11*f^2 + 24*B^2*a^6*b^6*c^4*d^10*f^2 + 2
4*B^2*a^3*b^9*c^11*d^3*f^2 - 8*B^2*a^5*b^7*c^3*d^11*f^2 + 4566*A^2*a^4*b^8*
c^6*d^8*f^2 + 4284*A^2*a^6*b^6*c^6*d^8*f^2 - 3776*A^2*a^7*b^5*c^5*d^9*f^2 -
3624*A^2*a^5*b^7*c^5*d^9*f^2 + 3122*A^2*a^4*b^8*c^4*d^10*f^2 + 3108*A^2*a^
2*b^10*c^6*d^8*f^2 + 2741*A^2*a^8*b^4*c^6*d^8*f^2 + 2592*A^2*a^6*b^6*c^4*d^
10*f^2 - 2536*A^2*a^3*b^9*c^5*d^9*f^2 + 2224*A^2*a^2*b^10*c^4*d^10*f^2 - 21
84*A^2*a^3*b^9*c^7*d^7*f^2 - 2016*A^2*a^5*b^7*c^7*d^7*f^2 - 1984*A^2*a^7*b^
5*c^7*d^7*f^2 + 1626*A^2*a^2*b^10*c^8*d^6*f^2 - 1624*A^2*a^9*b^3*c^5*d^9*f^
2 + 1603*A^2*a^8*b^4*c^4*d^10*f^2 + 1296*A^2*a^5*b^7*c^9*d^5*f^2 - 1144*A^2
*a^5*b^7*c^3*d^11*f^2 - 992*A^2*a^3*b^9*c^3*d^11*f^2 + 968*A^2*a^4*b^8*c^2*
d^12*f^2 - 888*A^2*a^7*b^5*c^3*d^11*f^2 + 849*A^2*a^4*b^8*c^8*d^6*f^2 + 808
*A^2*a^2*b^10*c^2*d^12*f^2 - 616*A^2*a^9*b^3*c^7*d^7*f^2 + 554*A^2*a^10*b^2
*c^6*d^8*f^2 + 504*A^2*a^7*b^5*c^9*d^5*f^2 - 504*A^2*a^6*b^6*c^10*d^4*f^2 +
460*A^2*a^6*b^6*c^2*d^12*f^2 + 350*A^2*a^10*b^2*c^4*d^10*f^2 + 350*A^2*a^2
*b^10*c^10*d^4*f^2 - 321*A^2*a^4*b^8*c^10*d^4*f^2 + 216*A^2*a^5*b^7*c^11*d^
3*f^2 - 216*A^2*a^3*b^9*c^11*d^3*f^2 + 182*A^2*a^2*b^10*c^12*d^2*f^2 - 152*
A^2*a^9*b^3*c^3*d^11*f^2 - 124*A^2*a^6*b^6*c^8*d^6*f^2 - 114*A^2*a^10*b^2*c
^2*d^12*f^2 + 104*A^2*a^3*b^9*c^9*d^5*f^2 + 77*A^2*a^8*b^4*c^2*d^12*f^2 + 7
4*A^2*a^8*b^4*c^8*d^6*f^2 - 70*A^2*a^8*b^4*c^10*d^4*f^2 + 56*A^2*a^9*b^3*c^
9*d^5*f^2 + 56*A^2*a^7*b^5*c^11*d^3*f^2 + 41*A^2*a^4*b^8*c^12*d^2*f^2 - 28*
A^2*a^10*b^2*c^8*d^6*f^2 - 28*A^2*a^6*b^6*c^12*d^2*f^2 + 12*B*C*b^12*c^13*d
*f^2 + 24*B*C*a^12*c*d^13*f^2 - 24*A*B*b^12*c^13*d*f^2 - 24*A*B*b^12*c*d^13
*f^2 - 16*B*C*a^11*b*d^14*f^2 - 24*A*B*a^12*c*d^13*f^2 - 16*B*C*a*b^11*c^14
*f^2 - 48*A*B*a*b^11*d^14*f^2 + 16*A*B*a^11*b*d^14*f^2 + 16*A*B*a*b^11*c^14
*f^2 - 216*C^2*a^11*b*c^5*d^9*f^2 + 216*C^2*a*b^11*c^9*d^5*f^2 + 56*C^2*a^1
1*b*c^3*d^11*f^2 + 56*C^2*a^9*b^3*c*d^13*f^2 + 56*C^2*a^5*b^7*c*d^13*f^2 +
40*C^2*a^7*b^5*c*d^13*f^2 - 40*C^2*a*b^11*c^11*d^3*f^2 + 32*C^2*a^5*b^7*c^1
3*d*f^2 - 24*C^2*a*b^11*c^7*d^7*f^2 - 16*C^2*a^3*b^9*c^13*d*f^2 + 16*C^2*a^
3*b^9*c*d^13*f^2 + 8*C^2*a^11*b*c^7*d^7*f^2 - 8*C^2*a*b^11*c^5*d^9*f^2 + 26

4*B^2*a*b^11*c^7*d^7*f^2 + 224*B^2*a*b^11*c^5*d^9*f^2 + 168*B^2*a^11*b*c^5*
d^9*f^2 - 112*B^2*a^9*b^3*c*d^13*f^2 - 104*B^2*a^11*b*c^3*d^11*f^2 - 104*B^
2*a^7*b^5*c*d^13*f^2 + 96*B^2*a*b^11*c^3*d^11*f^2 + 88*B^2*a*b^11*c^11*d^3*
f^2 - 72*B^2*a*b^11*c^9*d^5*f^2 - 64*B^2*a^5*b^7*c*d^13*f^2 + 32*B^2*a^3*b^
9*c^13*d*f^2 - 24*B^2*a^11*b*c^7*d^7*f^2 - 24*B^2*a^5*b^7*c^13*d*f^2 + 16*B
^2*a^3*b^9*c*d^13*f^2 - 888*A^2*a*b^11*c^7*d^7*f^2 - 800*A^2*a*b^11*c^5*d^9
*f^2 - 336*A^2*a*b^11*c^3*d^11*f^2 - 264*A^2*a*b^11*c^9*d^5*f^2 - 216*A^2*a
^11*b*c^5*d^9*f^2 - 184*A^2*a*b^11*c^11*d^3*f^2 - 128*A^2*a^3*b^9*c*d^13*f^
2 - 112*A^2*a^5*b^7*c*d^13*f^2 - 64*A^2*a^3*b^9*c^13*d*f^2 + 56*A^2*a^11*b*
c^3*d^11*f^2 - 56*A^2*a^7*b^5*c*d^13*f^2 + 32*A^2*a^9*b^3*c*d^13*f^2 + 8*A^
2*a^11*b*c^7*d^7*f^2 + 8*A^2*a^5*b^7*c^13*d*f^2 + 24*C^2*a^11*b*c*d^13*f^2
- 16*C^2*a*b^11*c^13*d*f^2 - 40*B^2*a^11*b*c*d^13*f^2 + 24*B^2*a*b^11*c^13*
d*f^2 + 16*B^2*a*b^11*c*d^13*f^2 - 48*A^2*a*b^11*c*d^13*f^2 - 40*A^2*a*b^11
*c^13*d*f^2 + 24*A^2*a^11*b*c*d^13*f^2 - 6*A*C*a^12*d^14*f^2 + 2*A*C*b^12*c
^14*f^2 + 33*C^2*b^12*c^12*d^2*f^2 - 27*C^2*b^12*c^10*d^4*f^2 + 3*C^2*b^12*
c^8*d^6*f^2 + 117*B^2*b^12*c^10*d^4*f^2 + 111*B^2*b^12*c^8*d^6*f^2 + 72*B^2
*b^12*c^6*d^8*f^2 + 33*C^2*a^12*c^4*d^10*f^2 - 27*C^2*a^12*c^2*d^12*f^2 + 2
4*B^2*b^12*c^4*d^10*f^2 + 4*B^2*b^12*c^2*d^12*f^2 - 3*B^2*b^12*c^12*d^2*f^2
- C^2*a^12*c^6*d^8*f^2 + 720*A^2*b^12*c^6*d^8*f^2 + 552*A^2*b^12*c^4*d^10*
f^2 + 471*A^2*b^12*c^8*d^6*f^2 + 216*A^2*b^12*c^2*d^12*f^2 + 93*A^2*b^12*c^
10*d^4*f^2 + 33*B^2*a^12*c^2*d^12*f^2 + 33*A^2*b^12*c^12*d^2*f^2 + 31*C^2*a
^8*b^4*d^14*f^2 - 27*B^2*a^12*c^4*d^10*f^2 + 20*C^2*a^6*b^6*d^14*f^2 + 4*C^
2*a^4*b^8*d^14*f^2 + 3*B^2*a^12*c^6*d^8*f^2 + 2*C^2*a^10*b^2*d^14*f^2 + 80*
B^2*a^6*b^6*d^14*f^2 + 64*B^2*a^4*b^8*d^14*f^2 + 33*A^2*a^12*c^4*d^10*f^2 +
31*B^2*a^8*b^4*d^14*f^2 - 27*A^2*a^12*c^2*d^12*f^2 + 16*B^2*a^2*b^10*d^14*
f^2 + 14*C^2*a^2*b^10*c^14*f^2 + 14*B^2*a^10*b^2*d^14*f^2 - C^2*a^4*b^8*c^1
4*f^2 - A^2*a^12*c^6*d^8*f^2 + 120*A^2*a^2*b^10*d^14*f^2 + 112*A^2*a^4*b^8*
d^14*f^2 - 17*A^2*a^8*b^4*d^14*f^2 - 10*B^2*a^2*b^10*c^14*f^2 - 10*A^2*a^10
*b^2*d^14*f^2 + 8*A^2*a^6*b^6*d^14*f^2 + 3*B^2*a^4*b^8*c^14*f^2 + 14*A^2*a^
2*b^10*c^14*f^2 - A^2*a^4*b^8*c^14*f^2 + 3*C^2*a^12*d^14*f^2 - C^2*b^12*c^1
4*f^2 + 36*A^2*b^12*d^14*f^2 + 3*B^2*b^12*c^14*f^2 - B^2*a^12*d^14*f^2 + 3*
A^2*a^12*d^14*f^2 - A^2*b^12*c^14*f^2 - 44*A*B*C*a*b^9*c^10*d*f + 3816*A*B*
C*a^5*b^5*c^4*d^7*f + 2920*A*B*C*a^2*b^8*c^5*d^6*f - 2736*A*B*C*a^3*b^7*c^6
*d^5*f - 2672*A*B*C*a^4*b^6*c^3*d^8*f + 1996*A*B*C*a^4*b^6*c^7*d^4*f - 1412
*A*B*C*a^6*b^4*c^5*d^6*f + 1120*A*B*C*a^3*b^7*c^2*d^9*f + 1080*A*B*C*a^2*b^
8*c^7*d^4*f + 1040*A*B*C*a^5*b^5*c^2*d^9*f + 684*A*B*C*a^4*b^6*c^5*d^6*f +
592*A*B*C*a^3*b^7*c^4*d^7*f - 560*A*B*C*a^7*b^3*c^2*d^9*f - 448*A*B*C*a^2*b
^8*c^3*d^8*f - 400*A*B*C*a^5*b^5*c^8*d^3*f - 398*A*B*C*a^2*b^8*c^9*d^2*f -
312*A*B*C*a^6*b^4*c^3*d^8*f + 166*A*B*C*a^8*b^2*c^3*d^8*f + 136*A*B*C*a^5*b
^5*c^6*d^5*f + 128*A*B*C*a^7*b^3*c^6*d^5*f - 100*A*B*C*a^6*b^4*c^7*d^4*f +
64*A*B*C*a^7*b^3*c^4*d^7*f - 64*A*B*C*a^4*b^6*c^9*d^2*f - 32*A*B*C*a^3*b^7*
c^8*d^3*f - 16*A*B*C*a^8*b^2*c^5*d^6*f - 1312*A*B*C*a*b^9*c^4*d^7*f + 996*A
*B*C*a*b^9*c^8*d^3*f + 728*A*B*C*a^6*b^4*c*d^10*f - 624*A*B*C*a*b^9*c^6*d^5
*f - 584*A*B*C*a^2*b^8*c*d^10*f - 512*A*B*C*a^4*b^6*c*d^10*f - 320*A*B*C*a*
b^9*c^2*d^9*f - 98*A*B*C*a^8*b^2*c*d^10*f + 36*A*B*C*a^9*b*c^2*d^9*f + 32*A

*B*C*a^3*b^7*c^10*d*f - 16*A*B*C*a^9*b*c^4*d^7*f + 46*B*C^2*a*b^9*c^10*d*f
- 16*B^2*C*a*b^9*c*d^10*f - 2*B^2*C*a^9*b*c*d^10*f + 312*A^2*C*a*b^9*c*d^10
*f - 48*A*C^2*a*b^9*c*d^10*f - 6*A^2*C*a^9*b*c*d^10*f + 6*A*C^2*a^9*b*c*d^1
0*f + 208*A*B^2*a*b^9*c*d^10*f - 2*A^2*B*a*b^9*c^10*d*f + 2*A*B^2*a^9*b*c*d
^10*f - 480*A*B*C*b^10*c^7*d^4*f + 78*A*B*C*b^10*c^9*d^2*f - 64*A*B*C*b^10*
c^5*d^6*f + 2*A*B*C*a^10*c^3*d^8*f - 224*A*B*C*a^5*b^5*d^11*f + 80*A*B*C*a^
7*b^3*d^11*f - 32*A*B*C*a^3*b^7*d^11*f + 2*A*B*C*a^2*b^8*c^11*f - 1692*B*C^
2*a^5*b^5*c^4*d^7*f - 1500*B^2*C*a^5*b^5*c^5*d^6*f - 1464*B^2*C*a^3*b^7*c^5
*d^6*f + 1426*B*C^2*a^6*b^4*c^5*d^6*f - 1158*B^2*C*a^6*b^4*c^4*d^7*f + 1152
*B*C^2*a^3*b^7*c^6*d^5*f + 1026*B^2*C*a^4*b^6*c^6*d^5*f - 974*B*C^2*a^4*b^6
*c^7*d^4*f + 960*B^2*C*a^5*b^5*c^3*d^8*f - 884*B*C^2*a^2*b^8*c^5*d^6*f - 76
4*B^2*C*a^5*b^5*c^7*d^4*f + 752*B^2*C*a^2*b^8*c^4*d^7*f - 752*B*C^2*a^3*b^7
*c^4*d^7*f + 738*B^2*C*a^4*b^6*c^4*d^7*f - 688*B^2*C*a^6*b^4*c^2*d^9*f - 67
5*B^2*C*a^2*b^8*c^8*d^3*f + 560*B*C^2*a^5*b^5*c^8*d^3*f + 496*B*C^2*a^7*b^3
*c^2*d^9*f + 496*B*C^2*a^4*b^6*c^3*d^8*f - 468*B*C^2*a^2*b^8*c^7*d^4*f + 45
6*B^2*C*a^7*b^3*c^3*d^8*f - 452*B^2*C*a^4*b^6*c^8*d^3*f - 416*B*C^2*a^3*b^7
*c^2*d^9*f + 378*B*C^2*a^4*b^6*c^5*d^6*f + 376*B*C^2*a^3*b^7*c^8*d^3*f - 36
0*B^2*C*a^2*b^8*c^6*d^5*f + 355*B*C^2*a^2*b^8*c^9*d^2*f + 346*B^2*C*a^6*b^4
*c^6*d^5*f - 320*B^2*C*a^4*b^6*c^2*d^9*f + 268*B^2*C*a^2*b^8*c^2*d^9*f + 21
6*B^2*C*a^3*b^7*c^7*d^4*f - 203*B*C^2*a^8*b^2*c^3*d^8*f - 184*B*C^2*a^7*b^3
*c^6*d^5*f + 170*B*C^2*a^6*b^4*c^7*d^4*f + 160*B^2*C*a^7*b^3*c^5*d^6*f - 16
0*B*C^2*a^5*b^5*c^2*d^9*f - 140*B^2*C*a^8*b^2*c^4*d^7*f - 136*B*C^2*a^2*b^8
*c^3*d^8*f + 112*B^2*C*a^3*b^7*c^9*d^2*f + 91*B^2*C*a^8*b^2*c^2*d^9*f + 88*
B*C^2*a^7*b^3*c^4*d^7*f + 72*B^2*C*a^6*b^4*c^8*d^3*f - 64*B^2*C*a^3*b^7*c^3
*d^8*f - 60*B*C^2*a^6*b^4*c^3*d^8*f + 56*B*C^2*a^4*b^6*c^9*d^2*f + 52*B*C^2
*a^5*b^5*c^6*d^5*f - 48*B^2*C*a^7*b^3*c^7*d^4*f + 48*B^2*C*a^5*b^5*c^9*d^2*
f + 44*B*C^2*a^8*b^2*c^5*d^6*f - 36*B*C^2*a^6*b^4*c^9*d^2*f + 12*B^2*C*a^8*
b^2*c^6*d^5*f - 2958*A^2*C*a^4*b^6*c^4*d^7*f - 1932*A^2*C*a^2*b^8*c^4*d^7*f
 + 1848*A^2*C*a^3*b^7*c^5*d^6*f + 1728*A^2*C*a^3*b^7*c^3*d^8*f + 1524*A^2*C
*a^5*b^5*c^5*d^6*f + 1374*A*C^2*a^4*b^6*c^4*d^7*f - 1272*A*C^2*a^3*b^7*c^5*
d^6*f - 1236*A*C^2*a^5*b^5*c^5*d^6*f + 1116*A*C^2*a^2*b^8*c^4*d^7*f - 1110*
A^2*C*a^4*b^6*c^6*d^5*f + 1038*A*C^2*a^4*b^6*c^6*d^5*f - 768*A^2*C*a^2*b^8*
c^2*d^9*f - 696*A^2*C*a^3*b^7*c^7*d^4*f - 666*A*C^2*a^6*b^4*c^4*d^7*f + 564
*A^2*C*a^2*b^8*c^6*d^5*f - 564*A*C^2*a^5*b^5*c^7*d^4*f - 555*A*C^2*a^2*b^8*
c^8*d^3*f + 519*A^2*C*a^2*b^8*c^8*d^3*f - 480*A*C^2*a^3*b^7*c^3*d^8*f + 456
*A*C^2*a^5*b^5*c^3*d^8*f - 420*A*C^2*a^6*b^4*c^2*d^9*f + 408*A*C^2*a^3*b^7*
c^7*d^4*f + 408*A*C^2*a^2*b^8*c^2*d^9*f + 348*A^2*C*a^6*b^4*c^2*d^9*f - 348
*A*C^2*a^2*b^8*c^6*d^5*f + 342*A*C^2*a^6*b^4*c^6*d^5*f - 336*A*C^2*a^4*b^6*
c^8*d^3*f + 324*A^2*C*a^5*b^5*c^7*d^4*f - 312*A^2*C*a^4*b^6*c^2*d^9*f + 264
*A^2*C*a^4*b^6*c^8*d^3*f + 240*A*C^2*a^7*b^3*c^5*d^6*f + 195*A*C^2*a^8*b^2*
c^2*d^9*f - 174*A^2*C*a^6*b^4*c^6*d^5*f + 144*A*C^2*a^3*b^7*c^9*d^2*f - 123
*A^2*C*a^8*b^2*c^2*d^9*f + 120*A*C^2*a^7*b^3*c^3*d^8*f + 108*A*C^2*a^6*b^4*
c^8*d^3*f - 102*A^2*C*a^6*b^4*c^4*d^7*f - 96*A^2*C*a^8*b^2*c^4*d^7*f + 72*A
^2*C*a^7*b^3*c^3*d^8*f + 72*A*C^2*a^5*b^5*c^9*d^2*f + 48*A^2*C*a^7*b^3*c^5*
d^6*f - 48*A^2*C*a^3*b^7*c^9*d^2*f - 48*A*C^2*a^4*b^6*c^2*d^9*f - 24*A^2*C*

a^5*b^5*c^3*d^8*f - 12*A*C^2*a^8*b^2*c^4*d^7*f + 2736*A^2*B*a^3*b^7*c^6*d^5
*f + 2464*A^2*B*a^4*b^6*c^3*d^8*f - 2298*A*B^2*a^4*b^6*c^4*d^7*f - 2252*A^2
*B*a^2*b^8*c^5*d^6*f - 1692*A^2*B*a^5*b^5*c^4*d^7*f - 1592*A*B^2*a^2*b^8*c^
4*d^7*f - 1338*A*B^2*a^4*b^6*c^6*d^5*f + 1320*A*B^2*a^3*b^7*c^5*d^6*f + 121
2*A*B^2*a^5*b^5*c^5*d^6*f - 1056*A*B^2*a^5*b^5*c^3*d^8*f + 1024*A^2*B*a^3*b
^7*c^4*d^7*f - 1022*A^2*B*a^4*b^6*c^7*d^4*f - 880*A^2*B*a^5*b^5*c^2*d^9*f -
846*A^2*B*a^4*b^6*c^5*d^6*f - 840*A*B^2*a^3*b^7*c^7*d^4*f + 760*A*B^2*a^6*
b^4*c^2*d^9*f - 704*A^2*B*a^3*b^7*c^2*d^9*f + 688*A*B^2*a^3*b^7*c^3*d^8*f +
660*A^2*B*a^6*b^4*c^3*d^8*f - 612*A^2*B*a^2*b^8*c^7*d^4*f + 462*A*B^2*a^6*
b^4*c^4*d^7*f + 459*A*B^2*a^2*b^8*c^8*d^3*f - 412*A*B^2*a^2*b^8*c^2*d^9*f -
408*A*B^2*a^7*b^3*c^3*d^8*f + 388*A^2*B*a^5*b^5*c^6*d^5*f + 296*A^2*B*a^2*
b^8*c^3*d^8*f + 288*A*B^2*a^2*b^8*c^6*d^5*f + 284*A*B^2*a^5*b^5*c^7*d^4*f +
236*A*B^2*a^4*b^6*c^8*d^3*f - 226*A*B^2*a^6*b^4*c^6*d^5*f + 212*A*B^2*a^4*
b^6*c^2*d^9*f + 202*A^2*B*a^6*b^4*c^5*d^6*f - 152*A^2*B*a^7*b^3*c^4*d^7*f +
88*A^2*B*a^3*b^7*c^8*d^3*f + 79*A^2*B*a^2*b^8*c^9*d^2*f - 70*A^2*B*a^6*b^4
*c^7*d^4*f + 68*A*B^2*a^8*b^2*c^4*d^7*f + 64*A^2*B*a^7*b^3*c^2*d^9*f - 64*A
*B^2*a^3*b^7*c^9*d^2*f + 56*A^2*B*a^7*b^3*c^6*d^5*f + 56*A^2*B*a^5*b^5*c^8*
d^3*f + 37*A^2*B*a^8*b^2*c^3*d^8*f - 28*A^2*B*a^8*b^2*c^5*d^6*f - 28*A^2*B*
a^4*b^6*c^9*d^2*f + 17*A*B^2*a^8*b^2*c^2*d^9*f - 16*A*B^2*a^7*b^3*c^5*d^6*f
+ 24*A*B*C*b^10*c*d^10*f - 6*A*B*C*a^10*c*d^10*f + 48*A*B*C*a*b^9*d^11*f +
4*A*B*C*a^9*b*d^11*f + 432*B^2*C*a*b^9*c^7*d^4*f - 376*B*C^2*a^6*b^4*c*d^1
0*f - 354*B*C^2*a*b^9*c^8*d^3*f + 352*B^2*C*a^5*b^5*c*d^10*f + 320*B^2*C*a*
b^9*c^5*d^6*f + 256*B^2*C*a^3*b^7*c*d^10*f - 232*B^2*C*a^7*b^3*c*d^10*f - 2
10*B^2*C*a*b^9*c^9*d^2*f - 152*B*C^2*a^4*b^6*c*d^10*f + 85*B*C^2*a^8*b^2*c*
d^10*f + 72*B^2*C*a*b^9*c^3*d^8*f - 48*B*C^2*a*b^9*c^6*d^5*f - 40*B*C^2*a^3
*b^7*c^10*d*f + 40*B*C^2*a^2*b^8*c*d^10*f + 37*B^2*C*a^2*b^8*c^10*d*f + 22*
B^2*C*a^9*b*c^3*d^8*f - 18*B*C^2*a^9*b*c^2*d^9*f + 16*B*C^2*a*b^9*c^2*d^9*f
- 12*B^2*C*a^4*b^6*c^10*d*f + 8*B*C^2*a^9*b*c^4*d^7*f + 8*B*C^2*a*b^9*c^4*
d^7*f - 984*A^2*C*a*b^9*c^7*d^4*f + 672*A^2*C*a*b^9*c^3*d^8*f + 552*A*C^2*a
*b^9*c^7*d^4*f - 504*A^2*C*a^5*b^5*c*d^10*f - 408*A^2*C*a*b^9*c^5*d^6*f + 4
08*A*C^2*a*b^9*c^5*d^6*f + 336*A*C^2*a^5*b^5*c*d^10*f - 216*A*C^2*a^7*b^3*c
*d^10*f + 192*A*C^2*a^3*b^7*c*d^10*f - 162*A*C^2*a*b^9*c^9*d^2*f + 120*A^2*
C*a^7*b^3*c*d^10*f + 96*A^2*C*a^3*b^7*c*d^10*f + 90*A^2*C*a*b^9*c^9*d^2*f +
66*A^2*C*a^9*b*c^3*d^8*f - 66*A*C^2*a^9*b*c^3*d^8*f + 57*A*C^2*a^2*b^8*c^1
0*d*f - 48*A*C^2*a*b^9*c^3*d^8*f - 9*A^2*C*a^2*b^8*c^10*d*f + 1736*A^2*B*a*
b^9*c^4*d^7*f + 1248*A^2*B*a*b^9*c^6*d^5*f - 1008*A*B^2*a*b^9*c^7*d^4*f + 7
72*A^2*B*a^4*b^6*c*d^10*f - 688*A*B^2*a^5*b^5*c*d^10*f - 608*A*B^2*a*b^9*c^
5*d^6*f + 436*A^2*B*a^2*b^8*c*d^10*f - 426*A^2*B*a*b^9*c^8*d^3*f + 312*A*B^
2*a*b^9*c^3*d^8*f + 304*A^2*B*a*b^9*c^2*d^9*f - 244*A^2*B*a^6*b^4*c*d^10*f
- 160*A*B^2*a^3*b^7*c*d^10*f + 114*A*B^2*a*b^9*c^9*d^2*f + 88*A*B^2*a^7*b^3
*c*d^10*f - 22*A*B^2*a^9*b*c^3*d^8*f - 18*A^2*B*a^9*b*c^2*d^9*f + 13*A^2*B*
a^8*b^2*c*d^10*f - 13*A*B^2*a^2*b^8*c^10*d*f + 8*A^2*B*a^9*b*c^4*d^7*f + 8*
A^2*B*a^3*b^7*c^10*d*f + 111*B^2*C*b^10*c^8*d^3*f - 39*B*C^2*b^10*c^9*d^2*f
+ 24*B*C^2*b^10*c^7*d^4*f - 4*B^2*C*b^10*c^2*d^9*f - 4*B*C^2*b^10*c^5*d^6*
f + 432*A^2*C*b^10*c^6*d^5*f + 192*A^2*C*b^10*c^4*d^7*f - 111*A^2*C*b^10*c^

8*d^3*f + 111*A*C^2*b^10*c^8*d^3*f - 72*A*C^2*b^10*c^6*d^5*f + 12*A*C^2*b^1
0*c^4*d^7*f - 3*B^2*C*a^10*c^2*d^9*f - B*C^2*a^10*c^3*d^8*f + 456*A^2*B*b^1
0*c^7*d^4*f - 288*A^2*B*b^10*c^3*d^8*f + 252*A*B^2*b^10*c^6*d^5*f + 192*A*B
^2*b^10*c^4*d^7*f - 183*A*B^2*b^10*c^8*d^3*f - 148*A^2*B*b^10*c^5*d^6*f + 1
12*B^2*C*a^6*b^4*d^11*f + 76*A*B^2*b^10*c^2*d^9*f - 64*B*C^2*a^7*b^3*d^11*f
 + 16*B^2*C*a^4*b^6*d^11*f - 16*B^2*C*a^2*b^8*d^11*f + 16*B*C^2*a^5*b^5*d^1
1*f + 16*B*C^2*a^3*b^7*d^11*f - 9*A^2*C*a^10*c^2*d^9*f + 9*A*C^2*a^10*c^2*d
^9*f - 3*A^2*B*b^10*c^9*d^2*f - B^2*C*a^8*b^2*d^11*f + 96*A^2*C*a^4*b^6*d^1
1*f - 84*A^2*C*a^6*b^4*d^11*f + 72*A*C^2*a^6*b^4*d^11*f - 24*A*C^2*a^4*b^6*
d^11*f - 24*A*C^2*a^2*b^8*d^11*f - 21*A*C^2*a^8*b^2*d^11*f + 12*A^2*C*a^2*b
^8*d^11*f + 9*A^2*C*a^8*b^2*d^11*f + 3*A*B^2*a^10*c^2*d^9*f - A^2*B*a^10*c^
3*d^8*f - B*C^2*a^2*b^8*c^11*f + 176*A*B^2*a^4*b^6*d^11*f + 136*A^2*B*a^5*b
^5*d^11*f - 128*A^2*B*a^3*b^7*d^11*f + 112*A*B^2*a^2*b^8*d^11*f - 64*A*B^2*
a^6*b^4*d^11*f - 16*A^2*B*a^7*b^3*d^11*f - A^2*B*a^2*b^8*c^11*f - 2*C^3*a^9
*b*c*d^10*f - 2*B^3*a*b^9*c^10*d*f - 264*A^3*a*b^9*c*d^10*f + 2*A^3*a^9*b*c
*d^10*f - 9*B^2*C*b^10*c^10*d*f + 9*A^2*C*b^10*c^10*d*f - 9*A*C^2*b^10*c^10
*d*f + 3*B*C^2*a^10*c*d^10*f - 132*A^2*B*b^10*c*d^10*f - 3*A*B^2*b^10*c^10*
d*f - 2*B*C^2*a^9*b*d^11*f + 3*A^2*B*a^10*c*d^10*f - 2*B^2*C*a*b^9*c^11*f -
 120*A^2*B*a*b^9*d^11*f - 6*A^2*C*a*b^9*c^11*f + 6*A*C^2*a*b^9*c^11*f - 2*A
^2*B*a^9*b*d^11*f + 2*A*B^2*a*b^9*c^11*f + 520*C^3*a^3*b^7*c^5*d^6*f + 460*
C^3*a^5*b^5*c^5*d^6*f - 418*C^3*a^4*b^6*c^6*d^5*f + 406*C^3*a^6*b^4*c^4*d^7
*f + 268*C^3*a^5*b^5*c^7*d^4*f - 266*C^3*a^6*b^4*c^6*d^5*f + 233*C^3*a^2*b^
8*c^8*d^3*f - 176*C^3*a^7*b^3*c^5*d^6*f + 164*C^3*a^6*b^4*c^2*d^9*f + 140*C
^3*a^2*b^8*c^6*d^5*f + 136*C^3*a^4*b^6*c^2*d^9*f - 128*C^3*a^3*b^7*c^9*d^2*
f + 128*C^3*a^3*b^7*c^3*d^8*f - 108*C^3*a^6*b^4*c^8*d^3*f - 104*C^3*a^7*b^3
*c^3*d^8*f - 104*C^3*a^5*b^5*c^3*d^8*f + 100*C^3*a^4*b^6*c^8*d^3*f - 89*C^3
*a^8*b^2*c^2*d^9*f - 72*C^3*a^5*b^5*c^9*d^2*f + 40*C^3*a^8*b^2*c^4*d^7*f -
40*C^3*a^3*b^7*c^7*d^4*f - 28*C^3*a^2*b^8*c^4*d^7*f - 16*C^3*a^2*b^8*c^2*d^
9*f - 2*C^3*a^4*b^6*c^4*d^7*f + 828*B^3*a^5*b^5*c^4*d^7*f + 408*B^3*a^2*b^8
*c^5*d^6*f + 390*B^3*a^4*b^6*c^7*d^4*f - 372*B^3*a^4*b^6*c^3*d^8*f - 336*B^
3*a^3*b^7*c^6*d^5*f - 314*B^3*a^6*b^4*c^5*d^6*f + 288*B^3*a^3*b^7*c^4*d^7*f
 + 216*B^3*a^2*b^8*c^7*d^4*f - 176*B^3*a^7*b^3*c^2*d^9*f + 128*B^3*a^3*b^7*
c^2*d^9*f + 108*B^3*a^5*b^5*c^6*d^5*f + 88*B^3*a^7*b^3*c^4*d^7*f + 72*B^3*a
^5*b^5*c^2*d^9*f - 68*B^3*a^2*b^8*c^3*d^8*f - 65*B^3*a^2*b^8*c^9*d^2*f - 56
*B^3*a^5*b^5*c^8*d^3*f + 40*B^3*a^7*b^3*c^6*d^5*f + 37*B^3*a^8*b^2*c^3*d^8*
f + 30*B^3*a^4*b^6*c^5*d^6*f - 28*B^3*a^8*b^2*c^5*d^6*f + 24*B^3*a^3*b^7*c^
8*d^3*f - 4*B^3*a^4*b^6*c^9*d^2*f - 2*B^3*a^6*b^4*c^7*d^4*f + 1586*A^3*a^4*
b^6*c^4*d^7*f - 1376*A^3*a^3*b^7*c^3*d^8*f - 1096*A^3*a^3*b^7*c^5*d^6*f + 8
44*A^3*a^2*b^8*c^4*d^7*f - 748*A^3*a^5*b^5*c^5*d^6*f + 490*A^3*a^4*b^6*c^6*
d^5*f + 376*A^3*a^2*b^8*c^2*d^9*f + 362*A^3*a^6*b^4*c^4*d^7*f - 356*A^3*a^2
*b^8*c^6*d^5*f - 328*A^3*a^5*b^5*c^3*d^8*f + 328*A^3*a^3*b^7*c^7*d^4*f + 22
4*A^3*a^4*b^6*c^2*d^9*f - 197*A^3*a^2*b^8*c^8*d^3*f - 112*A^3*a^7*b^3*c^5*d
^6*f + 98*A^3*a^6*b^4*c^6*d^5*f - 92*A^3*a^6*b^4*c^2*d^9*f - 88*A^3*a^7*b^3
*c^3*d^8*f + 68*A^3*a^8*b^2*c^4*d^7*f + 32*A^3*a^3*b^7*c^9*d^2*f - 28*A^3*a
^5*b^5*c^7*d^4*f - 28*A^3*a^4*b^6*c^8*d^3*f + 17*A^3*a^8*b^2*c^2*d^9*f + 10

4*C^3*a^7*b^3*c*d^10*f + 54*C^3*a*b^9*c^9*d^2*f - 40*C^3*a*b^9*c^7*d^4*f -
35*C^3*a^2*b^8*c^10*d*f + 22*C^3*a^9*b*c^3*d^8*f + 16*C^3*a^5*b^5*c*d^10*f
- 16*C^3*a^3*b^7*c*d^10*f + 8*C^3*a*b^9*c^5*d^6*f - 2*A*B*C*b^10*c^11*f + 1
98*B^3*a*b^9*c^8*d^3*f + 192*B^3*a^6*b^4*c*d^10*f - 128*B^3*a*b^9*c^4*d^7*f
- 80*B^3*a^2*b^8*c*d^10*f - 56*B^3*a*b^9*c^2*d^9*f - 24*B^3*a*b^9*c^6*d^5*
f - 18*B^3*a^9*b*c^2*d^9*f - 16*B^3*a^4*b^6*c*d^10*f + 13*B^3*a^8*b^2*c*d^1
0*f + 8*B^3*a^9*b*c^4*d^7*f + 8*B^3*a^3*b^7*c^10*d*f - 624*A^3*a*b^9*c^3*d^
8*f + 472*A^3*a*b^9*c^7*d^4*f - 272*A^3*a^3*b^7*c*d^10*f + 152*A^3*a^5*b^5*
c*d^10*f - 22*A^3*a^9*b*c^3*d^8*f + 18*A^3*a*b^9*c^9*d^2*f - 13*A^3*a^2*b^8
*c^10*d*f - 8*A^3*a^7*b^3*c*d^10*f - 8*A^3*a*b^9*c^5*d^6*f + A*B^2*a^8*b^2*
d^11*f - C^3*b^10*c^8*d^3*f - 60*B^3*b^10*c^7*d^4*f - 32*B^3*b^10*c^5*d^6*f
+ 21*B^3*b^10*c^9*d^2*f - 12*B^3*b^10*c^3*d^8*f - 3*C^3*a^10*c^2*d^9*f - 3
60*A^3*b^10*c^6*d^5*f - 204*A^3*b^10*c^4*d^7*f + 11*C^3*a^8*b^2*d^11*f - 8*
C^3*a^6*b^4*d^11*f - 4*C^3*a^4*b^6*d^11*f - B^3*a^10*c^3*d^8*f - 64*B^3*a^5
*b^5*d^11*f - 32*B^3*a^3*b^7*d^11*f + 3*A^3*a^10*c^2*d^9*f - 68*A^3*a^4*b^6
*d^11*f + 20*A^3*a^6*b^4*d^11*f + 12*A^3*a^2*b^8*d^11*f - B^3*a^2*b^8*c^11*
f + 3*C^3*b^10*c^10*d*f + 3*B^3*a^10*c*d^10*f - 3*A^3*b^10*c^10*d*f - 2*C^3
*a*b^9*c^11*f - 2*B^3*a^9*b*d^11*f + 2*A^3*a*b^9*c^11*f - 36*A^2*C*b^10*d^1
1*f + 3*A^2*C*a^10*d^11*f - 3*A*C^2*a^10*d^11*f - A*B^2*a^10*d^11*f + 36*A^
3*b^10*d^11*f - A^3*a^10*d^11*f + A^3*b^10*c^8*d^3*f + A^3*a^8*b^2*d^11*f +
B^2*C*a^10*d^11*f + B*C^2*b^10*c^11*f + A^2*B*b^10*c^11*f + C^3*a^10*d^11*
f + B^3*b^10*c^11*f - 6*A*B^2*C*a*b^7*c^7*d + 4*A*B^2*C*a*b^7*c*d^7 + 168*A
^2*B*C*a^3*b^5*c^2*d^6 + 144*A*B*C^2*a^4*b^4*c^3*d^5 - 129*A^2*B*C*a^4*b^4*
c^3*d^5 - 96*A*B*C^2*a^3*b^5*c^2*d^6 + 84*A*B*C^2*a^2*b^6*c^3*d^5 + 72*A^2*
B*C*a^3*b^5*c^4*d^4 - 72*A^2*B*C*a^2*b^6*c^3*d^5 + 64*A*B^2*C*a^4*b^4*c^4*d
^4 - 60*A*B*C^2*a^3*b^5*c^4*d^4 + 57*A^2*B*C*a^2*b^6*c^5*d^3 - 56*A*B^2*C*a
^3*b^5*c^5*d^3 - 39*A*B^2*C*a^4*b^4*c^2*d^6 - 38*A*B^2*C*a^5*b^3*c^3*d^5 +
36*A*B^2*C*a^3*b^5*c^3*d^5 + 36*A*B*C^2*a^4*b^4*c^5*d^3 - 30*A*B*C^2*a^2*b^
6*c^5*d^3 + 27*A*B^2*C*a^2*b^6*c^6*d^2 - 24*A*B^2*C*a^2*b^6*c^2*d^6 - 24*A*
B*C^2*a^5*b^3*c^4*d^4 + 24*A*B*C^2*a^3*b^5*c^6*d^2 + 18*A^2*B*C*a^5*b^3*c^2
*d^6 - 18*A^2*B*C*a^4*b^4*c^5*d^3 - 15*A*B^2*C*a^2*b^6*c^4*d^4 + 12*A^2*B*C
*a^5*b^3*c^4*d^4 - 12*A^2*B*C*a^3*b^5*c^6*d^2 + 9*A*B^2*C*a^6*b^2*c^2*d^6 +
6*A*B*C^2*a^6*b^2*c^3*d^5 - 3*A^2*B*C*a^6*b^2*c^3*d^5 + 60*A^2*B*C*a*b^7*c
^2*d^6 - 51*A^2*B*C*a^4*b^4*c*d^7 + 48*A*B*C^2*a*b^7*c^6*d^2 - 42*A^2*B*C*a
^2*b^6*c*d^7 - 42*A^2*B*C*a*b^7*c^6*d^2 + 36*A*B*C^2*a^4*b^4*c*d^7 + 36*A*B
*C^2*a^2*b^6*c*d^7 + 36*A*B*C^2*a*b^7*c^4*d^4 - 30*A^2*B*C*a*b^7*c^4*d^4 +
24*A*B^2*C*a*b^7*c^3*d^5 - 24*A*B*C^2*a*b^7*c^2*d^6 + 18*A*B^2*C*a^5*b^3*c*
d^7 - 18*A*B*C^2*a^6*b^2*c*d^7 + 12*A*B^2*C*a^3*b^5*c*d^7 + 9*A^2*B*C*a^6*b
^2*c*d^7 + 6*A*B^2*C*a*b^7*c^5*d^3 - 6*A*B*C^2*a^2*b^6*c^7*d + 3*A^2*B*C*a^
2*b^6*c^7*d - 18*B^3*C*a*b^7*c^6*d^2 - 18*B*C^3*a*b^7*c^6*d^2 - 14*B^3*C*a*
b^7*c^4*d^4 - 14*B*C^3*a*b^7*c^4*d^4 - 10*B^3*C*a^2*b^6*c*d^7 - 10*B*C^3*a^
2*b^6*c*d^7 + 9*B^3*C*a^6*b^2*c*d^7 + 9*B*C^3*a^6*b^2*c*d^7 - 7*B^3*C*a^4*b
^4*c*d^7 - 7*B*C^3*a^4*b^4*c*d^7 + 6*B^2*C^2*a*b^7*c^7*d - 4*B^3*C*a*b^7*c^
2*d^6 + 4*B^2*C^2*a*b^7*c*d^7 - 4*B*C^3*a*b^7*c^2*d^6 + 3*B^3*C*a^2*b^6*c^7
*d + 3*B*C^3*a^2*b^6*c^7*d + 144*A^3*C*a*b^7*c^3*d^5 + 62*A^3*C*a*b^7*c^5*d

^3 + 48*A*C^3*a*b^7*c^3*d^5 - 36*A^2*C^2*a*b^7*c*d^7 + 26*A*C^3*a*b^7*c^5*d
^3 + 20*A^3*C*a^3*b^5*c*d^7 + 18*A^2*C^2*a*b^7*c^7*d - 18*A*C^3*a^5*b^3*c*d
^7 - 6*A^3*C*a^5*b^3*c*d^7 - 4*A*C^3*a^3*b^5*c*d^7 - 32*A^3*B*a*b^7*c^2*d^6
 - 32*A*B^3*a*b^7*c^2*d^6 + 22*A^3*B*a^4*b^4*c*d^7 + 22*A*B^3*a^4*b^4*c*d^7
 + 16*A^3*B*a^2*b^6*c*d^7 + 16*A*B^3*a^2*b^6*c*d^7 + 12*A^3*B*a*b^7*c^6*d^2
 + 12*A*B^3*a*b^7*c^6*d^2 + 8*A^3*B*a*b^7*c^4*d^4 - 8*A^2*B^2*a*b^7*c*d^7 +
 8*A*B^3*a*b^7*c^4*d^4 + 57*A^2*B*C*b^8*c^5*d^3 + 36*A^2*B*C*b^8*c^3*d^5 -
30*A*B*C^2*b^8*c^5*d^3 - 18*A*B*C^2*b^8*c^3*d^5 - 9*A*B^2*C*b^8*c^4*d^4 - 3
*A*B^2*C*b^8*c^6*d^2 - 2*A*B^2*C*b^8*c^2*d^6 + 36*A^2*B*C*a^3*b^5*d^8 + 24*
A*B*C^2*a^5*b^3*d^8 - 18*A^2*B*C*a^5*b^3*d^8 - 12*A*B*C^2*a^3*b^5*d^8 - 3*A
*B^2*C*a^6*b^2*d^8 - 3*A*B^2*C*a^4*b^4*d^8 - 2*A*B^2*C*a^2*b^6*d^8 + 34*B^2
*C^2*a^5*b^3*c^3*d^5 + 28*B^2*C^2*a^3*b^5*c^5*d^3 + 24*B^2*C^2*a^4*b^4*c^2*
d^6 - 20*B^2*C^2*a^4*b^4*c^4*d^4 + 12*B^2*C^2*a^3*b^5*c^3*d^5 + 12*B^2*C^2*
a^2*b^6*c^2*d^6 - 9*B^2*C^2*a^6*b^2*c^2*d^6 + 9*B^2*C^2*a^4*b^4*c^6*d^2 + 9
*B^2*C^2*a^2*b^6*c^4*d^4 - 3*B^2*C^2*a^2*b^6*c^6*d^2 + 159*A^2*C^2*a^2*b^6*
c^4*d^4 - 156*A^2*C^2*a^3*b^5*c^3*d^5 + 90*A^2*C^2*a^5*b^3*c^3*d^5 + 78*A^2
*C^2*a^2*b^6*c^2*d^6 - 63*A^2*C^2*a^4*b^4*c^4*d^4 - 27*A^2*C^2*a^6*b^2*c^2*
d^6 - 27*A^2*C^2*a^2*b^6*c^6*d^2 - 18*A^2*C^2*a^4*b^4*c^2*d^6 + 9*A^2*C^2*a
^4*b^4*c^6*d^2 + 66*A^2*B^2*a^2*b^6*c^2*d^6 + 60*A^2*B^2*a^2*b^6*c^4*d^4 -
48*A^2*B^2*a^3*b^5*c^3*d^5 + 42*A^2*B^2*a^4*b^4*c^2*d^6 + 28*A^2*B^2*a^3*b^
5*c^5*d^3 - 17*A^2*B^2*a^4*b^4*c^4*d^4 - 6*A^2*B^2*a^2*b^6*c^6*d^2 + 4*A^2*
B^2*a^5*b^3*c^3*d^5 + 36*A^3*C*a*b^7*c*d^7 - 18*A*C^3*a*b^7*c^7*d + 12*A*C^
3*a*b^7*c*d^7 - 6*A^3*C*a*b^7*c^7*d + 12*A^2*B*C*b^8*c*d^7 + 6*A*B*C^2*b^8*
c^7*d - 6*A*B*C^2*b^8*c*d^7 - 3*A^2*B*C*b^8*c^7*d + 24*A^2*B*C*a*b^7*d^8 -
12*A*B*C^2*a*b^7*d^8 - 53*B^3*C*a^4*b^4*c^3*d^5 - 53*B*C^3*a^4*b^4*c^3*d^5
- 32*B^3*C*a^2*b^6*c^3*d^5 - 32*B*C^3*a^2*b^6*c^3*d^5 - 18*B^3*C*a^4*b^4*c^
5*d^3 - 18*B*C^3*a^4*b^4*c^5*d^3 + 16*B^3*C*a^3*b^5*c^4*d^4 + 16*B*C^3*a^3*
b^5*c^4*d^4 + 12*B^3*C*a^5*b^3*c^4*d^4 - 12*B^3*C*a^3*b^5*c^6*d^2 + 12*B^2*
C^2*a*b^7*c^3*d^5 + 12*B*C^3*a^5*b^3*c^4*d^4 - 12*B*C^3*a^3*b^5*c^6*d^2 + 8
*B^3*C*a^3*b^5*c^2*d^6 + 8*B*C^3*a^3*b^5*c^2*d^6 - 6*B^3*C*a^5*b^3*c^2*d^6
- 6*B^2*C^2*a^5*b^3*c*d^7 + 6*B^2*C^2*a*b^7*c^5*d^3 - 6*B*C^3*a^5*b^3*c^2*d
^6 - 3*B^3*C*a^6*b^2*c^3*d^5 - 3*B*C^3*a^6*b^2*c^3*d^5 - 175*A^3*C*a^2*b^6*
c^4*d^4 + 164*A^3*C*a^3*b^5*c^3*d^5 - 144*A^2*C^2*a*b^7*c^3*d^5 - 124*A^3*C
*a^2*b^6*c^2*d^6 - 90*A*C^3*a^5*b^3*c^3*d^5 - 73*A*C^3*a^2*b^6*c^4*d^4 - 66
*A^2*C^2*a*b^7*c^5*d^3 + 44*A*C^3*a^3*b^5*c^3*d^5 + 36*A*C^3*a^4*b^4*c^4*d^
4 - 30*A^3*C*a^5*b^3*c^3*d^5 + 30*A^3*C*a^4*b^4*c^4*d^4 + 27*A*C^3*a^6*b^2*
c^2*d^6 + 21*A*C^3*a^4*b^4*c^2*d^6 + 18*A^2*C^2*a^5*b^3*c*d^7 - 18*A*C^3*a^
4*b^4*c^6*d^2 - 16*A*C^3*a^2*b^6*c^2*d^6 - 15*A^3*C*a^4*b^4*c^2*d^6 + 15*A^
3*C*a^2*b^6*c^6*d^2 - 12*A^2*C^2*a^3*b^5*c*d^7 + 9*A^3*C*a^6*b^2*c^2*d^6 +
9*A*C^3*a^2*b^6*c^6*d^2 - 80*A^3*B*a^3*b^5*c^2*d^6 - 80*A*B^3*a^3*b^5*c^2*d
^6 + 38*A^3*B*a^4*b^4*c^3*d^5 + 38*A*B^3*a^4*b^4*c^3*d^5 - 36*A^2*B^2*a*b^7
*c^3*d^5 - 28*A^3*B*a^3*b^5*c^4*d^4 - 28*A^3*B*a^2*b^6*c^5*d^3 - 28*A*B^3*a
^3*b^5*c^4*d^4 - 28*A*B^3*a^2*b^6*c^5*d^3 + 20*A^3*B*a^2*b^6*c^3*d^5 + 20*A
*B^3*a^2*b^6*c^3*d^5 - 12*A^3*B*a^5*b^3*c^2*d^6 - 12*A^2*B^2*a^5*b^3*c*d^7
- 12*A^2*B^2*a^3*b^5*c*d^7 - 12*A^2*B^2*a*b^7*c^5*d^3 - 12*A*B^3*a^5*b^3*c^

2*d^6 + 6*B^2*C^2*b^8*c^6*d^2 + 3*B^2*C^2*b^8*c^4*d^4 + 36*A^2*C^2*b^8*c^4*
d^4 + 27*A^2*C^2*b^8*c^2*d^6 - 18*A^2*C^2*b^8*c^6*d^2 + 33*A^2*B^2*b^8*c^4*
d^4 + 28*A^2*B^2*b^8*c^2*d^6 + 9*B^2*C^2*a^4*b^4*d^8 + 6*A^2*B^2*b^8*c^6*d^
2 + 4*B^2*C^2*a^2*b^6*d^8 + 3*B^2*C^2*a^6*b^2*d^8 - 30*A^2*C^2*a^4*b^4*d^8
+ 9*A^2*C^2*a^6*b^2*d^8 + 16*A^2*B^2*a^2*b^6*d^8 + 3*A^2*B^2*a^4*b^4*d^8 +
6*C^4*a^5*b^3*c*d^7 + 4*C^4*a^3*b^5*c*d^7 - 2*C^4*a*b^7*c^5*d^3 - 12*B^4*a^
5*b^3*c*d^7 + 12*B^4*a*b^7*c^3*d^5 + 8*B^4*a*b^7*c^5*d^3 - 4*B^4*a^3*b^5*c*
d^7 - 48*A^4*a*b^7*c^3*d^5 - 20*A^4*a*b^7*c^5*d^3 - 8*A^4*a^3*b^5*c*d^7 - 6
3*A^3*C*b^8*c^4*d^4 - 54*A^3*C*b^8*c^2*d^6 + 9*A^3*C*b^8*c^6*d^2 + 9*A*C^3*
b^8*c^6*d^2 - 3*A*C^3*b^8*c^4*d^4 - 28*A^3*B*b^8*c^5*d^3 - 28*A*B^3*b^8*c^5
*d^3 - 18*A^3*B*b^8*c^3*d^5 - 18*A*B^3*b^8*c^3*d^5 - 10*B^3*C*a^5*b^3*d^8 -
10*B*C^3*a^5*b^3*d^8 - 4*B^3*C*a^3*b^5*d^8 - 4*B*C^3*a^3*b^5*d^8 + 23*A^3*
C*a^4*b^4*d^8 - 18*A^3*C*a^2*b^6*d^8 + 11*A*C^3*a^4*b^4*d^8 - 9*A*C^3*a^6*b
^2*d^8 + 6*A*C^3*a^2*b^6*d^8 - 3*A^3*C*a^6*b^2*d^8 - 20*A^3*B*a^3*b^5*d^8 -
20*A*B^3*a^3*b^5*d^8 + 4*A^3*B*a^5*b^3*d^8 + 4*A*B^3*a^5*b^3*d^8 + B^3*C*a
^2*b^6*c^5*d^3 + B*C^3*a^2*b^6*c^5*d^3 + 6*C^4*a*b^7*c^7*d + 4*B^4*a*b^7*c*
d^7 - 12*A^4*a*b^7*c*d^7 - 3*B^3*C*b^8*c^7*d - 3*B*C^3*b^8*c^7*d - 6*A^3*B*
b^8*c*d^7 - 6*A*B^3*b^8*c*d^7 - 12*A^3*B*a*b^7*d^8 - 12*A*B^3*a*b^7*d^8 + 3
0*C^4*a^5*b^3*c^3*d^5 + 19*C^4*a^2*b^6*c^4*d^4 - 9*C^4*a^6*b^2*c^2*d^6 + 9*
C^4*a^4*b^4*c^6*d^2 + 4*C^4*a^3*b^5*c^3*d^5 + 4*C^4*a^2*b^6*c^2*d^6 - 3*C^4
*a^4*b^4*c^4*d^4 - 3*C^4*a^4*b^4*c^2*d^6 + 3*C^4*a^2*b^6*c^6*d^2 + 28*B^4*a
^3*b^5*c^5*d^3 + 27*B^4*a^4*b^4*c^2*d^6 - 17*B^4*a^4*b^4*c^4*d^4 - 10*B^4*a
^2*b^6*c^4*d^4 + 8*B^4*a^3*b^5*c^3*d^5 + 8*B^4*a^2*b^6*c^2*d^6 - 6*B^4*a^2*
b^6*c^6*d^2 + 4*B^4*a^5*b^3*c^3*d^5 + 70*A^4*a^2*b^6*c^4*d^4 + 58*A^4*a^2*b
^6*c^2*d^6 - 56*A^4*a^3*b^5*c^3*d^5 + 15*A^4*a^4*b^4*c^2*d^6 + B^2*C^2*b^8*
c^2*d^6 - 18*A^3*C*b^8*d^8 + B^3*C*b^8*c^5*d^3 + B*C^3*b^8*c^5*d^3 + 6*B^4*
b^8*c^6*d^2 + 3*B^4*b^8*c^4*d^4 + 30*A^4*b^8*c^4*d^4 + 27*A^4*b^8*c^2*d^6 +
3*C^4*a^6*b^2*d^8 + 8*B^4*a^4*b^4*d^8 + 4*B^4*a^2*b^6*d^8 + 12*A^4*a^2*b^6
*d^8 - 5*A^4*a^4*b^4*d^8 + 9*A^2*C^2*b^8*d^8 + 9*A^2*B^2*b^8*d^8 + 9*A^4*b^
8*d^8 + B^4*b^8*c^2*d^6 + C^4*a^4*b^4*d^8, f, k)*((4*a^7*b^8*d^19 + 4*a^9*b
^6*d^19 - 4*a^11*b^4*d^19 - 4*a^13*b^2*d^19 + 4*b^15*c^7*d^12 + 12*b^15*c^9
*d^10 + 8*b^15*c^11*d^8 - 8*b^15*c^13*d^6 - 12*b^15*c^15*d^4 - 4*b^15*c^17*
d^2 - 20*a*b^14*c^6*d^13 - 44*a*b^14*c^8*d^11 + 32*a*b^14*c^10*d^9 + 168*a*
b^14*c^12*d^7 + 172*a*b^14*c^14*d^5 + 68*a*b^14*c^16*d^3 + 16*a^3*b^12*c^18
*d + 8*a^5*b^10*c^18*d - 20*a^6*b^9*c*d^18 - 4*a^8*b^7*c*d^18 + 60*a^10*b^5
*c*d^18 + 52*a^12*b^3*c*d^18 + 32*a^14*b*c^3*d^16 + 48*a^14*b*c^5*d^14 + 32
*a^14*b*c^7*d^12 + 8*a^14*b*c^9*d^10 + 36*a^2*b^13*c^5*d^14 + 32*a^2*b^13*c
^7*d^12 - 292*a^2*b^13*c^9*d^10 - 768*a^2*b^13*c^11*d^8 - 772*a^2*b^13*c^13
*d^6 - 352*a^2*b^13*c^15*d^4 - 60*a^2*b^13*c^17*d^2 - 20*a^3*b^12*c^4*d^15
+ 64*a^3*b^12*c^6*d^13 + 668*a^3*b^12*c^8*d^11 + 1648*a^3*b^12*c^10*d^9 + 1
892*a^3*b^12*c^12*d^7 + 1088*a^3*b^12*c^14*d^5 + 276*a^3*b^12*c^16*d^3 - 20
*a^4*b^11*c^3*d^16 - 104*a^4*b^11*c^5*d^14 - 640*a^4*b^11*c^7*d^12 - 2028*a
^4*b^11*c^9*d^10 - 3092*a^4*b^11*c^11*d^8 - 2368*a^4*b^11*c^13*d^6 - 856*a^
4*b^11*c^15*d^4 - 108*a^4*b^11*c^17*d^2 + 36*a^5*b^10*c^2*d^17 + 8*a^5*b^10
*c^4*d^15 + 112*a^5*b^10*c^6*d^13 + 1404*a^5*b^10*c^8*d^11 + 3404*a^5*b^10*

c^10*d^9 + 3552*a^5*b^10*c^12*d^7 + 1752*a^5*b^10*c^14*d^5 + 348*a^5*b^10*c
^16*d^3 + 64*a^6*b^9*c^3*d^16 + 392*a^6*b^9*c^5*d^14 + 32*a^6*b^9*c^7*d^12
- 1864*a^6*b^9*c^9*d^10 - 3296*a^6*b^9*c^11*d^8 - 2360*a^6*b^9*c^13*d^6 - 7
04*a^6*b^9*c^15*d^4 - 52*a^6*b^9*c^17*d^2 - 32*a^7*b^8*c^2*d^17 - 568*a^7*b
^8*c^4*d^15 - 1504*a^7*b^8*c^6*d^13 - 976*a^7*b^8*c^8*d^11 + 1120*a^7*b^8*c
^10*d^9 + 1912*a^7*b^8*c^12*d^7 + 928*a^7*b^8*c^14*d^5 + 140*a^7*b^8*c^16*d
^3 + 472*a^8*b^7*c^3*d^16 + 2016*a^8*b^7*c^5*d^14 + 3076*a^8*b^7*c^7*d^12 +
1724*a^8*b^7*c^9*d^10 - 288*a^8*b^7*c^11*d^8 - 664*a^8*b^7*c^13*d^6 - 188*
a^8*b^7*c^15*d^4 - 240*a^9*b^6*c^2*d^17 - 1472*a^9*b^6*c^4*d^15 - 3316*a^9*
b^6*c^6*d^13 - 3484*a^9*b^6*c^8*d^11 - 1592*a^9*b^6*c^10*d^9 - 104*a^9*b^6*
c^12*d^7 + 92*a^9*b^6*c^14*d^5 + 704*a^10*b^5*c^3*d^16 + 2308*a^10*b^5*c^5*
d^14 + 3392*a^10*b^5*c^7*d^12 + 2468*a^10*b^5*c^9*d^10 + 832*a^10*b^5*c^11*
d^8 + 92*a^10*b^5*c^13*d^6 - 240*a^11*b^4*c^2*d^17 - 1108*a^11*b^4*c^4*d^15
- 2112*a^11*b^4*c^6*d^13 - 2028*a^11*b^4*c^8*d^11 - 976*a^11*b^4*c^10*d^9
- 188*a^11*b^4*c^12*d^7 + 348*a^12*b^3*c^3*d^16 + 872*a^12*b^3*c^5*d^14 + 1
048*a^12*b^3*c^7*d^12 + 612*a^12*b^3*c^9*d^10 + 140*a^12*b^3*c^11*d^8 - 68*
a^13*b^2*c^2*d^17 - 232*a^13*b^2*c^4*d^15 - 328*a^13*b^2*c^6*d^13 - 212*a^1
3*b^2*c^8*d^11 - 52*a^13*b^2*c^10*d^9 + 8*a*b^14*c^18*d + 8*a^14*b*c*d^18)/
(a^10*d^14 + b^10*c^14 + 2*a^2*b^8*c^14 + a^4*b^6*c^14 + a^6*b^4*d^14 + 2*a
^8*b^2*d^14 + 4*a^10*c^2*d^12 + 6*a^10*c^4*d^10 + 4*a^10*c^6*d^8 + a^10*c^8
*d^6 + b^10*c^6*d^8 + 4*b^10*c^8*d^6 + 6*b^10*c^10*d^4 + 4*b^10*c^12*d^2 -
6*a*b^9*c^5*d^9 - 24*a*b^9*c^7*d^7 - 36*a*b^9*c^9*d^5 - 24*a*b^9*c^11*d^3 -
12*a^3*b^7*c^13*d - 6*a^5*b^5*c*d^13 - 6*a^5*b^5*c^13*d - 12*a^7*b^3*c*d^1
3 - 24*a^9*b*c^3*d^11 - 36*a^9*b*c^5*d^9 - 24*a^9*b*c^7*d^7 - 6*a^9*b*c^9*d
^5 + 15*a^2*b^8*c^4*d^10 + 62*a^2*b^8*c^6*d^8 + 98*a^2*b^8*c^8*d^6 + 72*a^2
*b^8*c^10*d^4 + 23*a^2*b^8*c^12*d^2 - 20*a^3*b^7*c^3*d^11 - 92*a^3*b^7*c^5*
d^9 - 168*a^3*b^7*c^7*d^7 - 152*a^3*b^7*c^9*d^5 - 68*a^3*b^7*c^11*d^3 + 15*
a^4*b^6*c^2*d^12 + 90*a^4*b^6*c^4*d^10 + 211*a^4*b^6*c^6*d^8 + 244*a^4*b^6*
c^8*d^6 + 141*a^4*b^6*c^10*d^4 + 34*a^4*b^6*c^12*d^2 - 64*a^5*b^5*c^3*d^11
- 202*a^5*b^5*c^5*d^9 - 288*a^5*b^5*c^7*d^7 - 202*a^5*b^5*c^9*d^5 - 64*a^5*
b^5*c^11*d^3 + 34*a^6*b^4*c^2*d^12 + 141*a^6*b^4*c^4*d^10 + 244*a^6*b^4*c^6
*d^8 + 211*a^6*b^4*c^8*d^6 + 90*a^6*b^4*c^10*d^4 + 15*a^6*b^4*c^12*d^2 - 68
*a^7*b^3*c^3*d^11 - 152*a^7*b^3*c^5*d^9 - 168*a^7*b^3*c^7*d^7 - 92*a^7*b^3*
c^9*d^5 - 20*a^7*b^3*c^11*d^3 + 23*a^8*b^2*c^2*d^12 + 72*a^8*b^2*c^4*d^10 +
98*a^8*b^2*c^6*d^8 + 62*a^8*b^2*c^8*d^6 + 15*a^8*b^2*c^10*d^4 - 6*a*b^9*c^
13*d - 6*a^9*b*c*d^13) + (tan(e + f*x)*(6*a^14*b*d^19 + 6*b^15*c^18*d + 8*a
^6*b^9*d^19 + 22*a^8*b^7*d^19 + 26*a^10*b^5*d^19 + 18*a^12*b^3*d^19 + 8*b^1
5*c^6*d^13 + 38*b^15*c^8*d^11 + 78*b^15*c^10*d^9 + 92*b^15*c^12*d^7 + 68*b^
15*c^14*d^5 + 30*b^15*c^16*d^3 - 48*a*b^14*c^5*d^14 - 224*a*b^14*c^7*d^12 -
448*a*b^14*c^9*d^10 - 512*a*b^14*c^11*d^8 - 368*a*b^14*c^13*d^6 - 160*a*b^
14*c^15*d^4 - 32*a*b^14*c^17*d^2 + 10*a^2*b^13*c^18*d + 2*a^4*b^11*c^18*d -
48*a^5*b^10*c*d^18 - 2*a^6*b^9*c^18*d - 128*a^7*b^8*c*d^18 - 144*a^9*b^6*c
*d^18 - 96*a^11*b^4*c*d^18 - 32*a^13*b^2*c*d^18 + 22*a^14*b*c^2*d^17 + 28*a
^14*b*c^4*d^15 + 12*a^14*b*c^6*d^13 - 2*a^14*b*c^8*d^11 - 2*a^14*b*c^10*d^9
+ 120*a^2*b^13*c^4*d^15 + 568*a^2*b^13*c^6*d^13 + 1138*a^2*b^13*c^8*d^11 +

1282*a^2*b^13*c^10*d^9 + 908*a^2*b^13*c^12*d^7 + 412*a^2*b^13*c^14*d^5 + 1
06*a^2*b^13*c^16*d^3 - 160*a^3*b^12*c^3*d^16 - 832*a^3*b^12*c^5*d^14 - 1776
*a^3*b^12*c^7*d^12 - 2032*a^3*b^12*c^9*d^10 - 1408*a^3*b^12*c^11*d^8 - 672*
a^3*b^12*c^13*d^6 - 240*a^3*b^12*c^15*d^4 - 48*a^3*b^12*c^17*d^2 + 120*a^4*
b^11*c^2*d^17 + 820*a^4*b^11*c^4*d^15 + 2044*a^4*b^11*c^6*d^13 + 2434*a^4*b
^11*c^8*d^11 + 1498*a^4*b^11*c^10*d^9 + 552*a^4*b^11*c^12*d^7 + 208*a^4*b^1
1*c^14*d^5 + 66*a^4*b^11*c^16*d^3 - 608*a^5*b^10*c^3*d^16 - 1904*a^5*b^10*c
^5*d^14 - 2384*a^5*b^10*c^7*d^12 - 976*a^5*b^10*c^9*d^10 + 448*a^5*b^10*c^1
1*d^8 + 496*a^5*b^10*c^13*d^6 + 112*a^5*b^10*c^15*d^4 + 344*a^6*b^9*c^2*d^1
7 + 1428*a^6*b^9*c^4*d^15 + 1988*a^6*b^9*c^6*d^13 + 214*a^6*b^9*c^8*d^11 -
2058*a^6*b^9*c^10*d^9 - 2000*a^6*b^9*c^12*d^7 - 688*a^6*b^9*c^14*d^5 - 66*a
^6*b^9*c^16*d^3 - 848*a^7*b^8*c^3*d^16 - 1520*a^7*b^8*c^5*d^14 + 80*a^7*b^8
*c^7*d^12 + 3056*a^7*b^8*c^9*d^10 + 3536*a^7*b^8*c^11*d^8 + 1648*a^7*b^8*c^
13*d^6 + 304*a^7*b^8*c^15*d^4 + 16*a^7*b^8*c^17*d^2 + 406*a^8*b^7*c^2*d^17
+ 1072*a^8*b^7*c^4*d^15 + 200*a^8*b^7*c^6*d^13 - 2626*a^8*b^7*c^8*d^11 - 40
42*a^8*b^7*c^10*d^9 - 2540*a^8*b^7*c^12*d^7 - 692*a^8*b^7*c^14*d^5 - 56*a^8
*b^7*c^16*d^3 - 624*a^9*b^6*c^3*d^16 - 544*a^9*b^6*c^5*d^14 + 1296*a^9*b^6*
c^7*d^12 + 3184*a^9*b^6*c^9*d^10 + 2672*a^9*b^6*c^11*d^8 + 960*a^9*b^6*c^13
*d^6 + 112*a^9*b^6*c^15*d^4 + 282*a^10*b^5*c^2*d^17 + 568*a^10*b^5*c^4*d^15
 - 168*a^10*b^5*c^6*d^13 - 1622*a^10*b^5*c^8*d^11 - 1862*a^10*b^5*c^10*d^9
- 860*a^10*b^5*c^12*d^7 - 140*a^10*b^5*c^14*d^5 - 336*a^11*b^4*c^3*d^16 - 2
72*a^11*b^4*c^5*d^14 + 352*a^11*b^4*c^7*d^12 + 768*a^11*b^4*c^9*d^10 + 496*
a^11*b^4*c^11*d^8 + 112*a^11*b^4*c^13*d^6 + 122*a^12*b^3*c^2*d^17 + 252*a^1
2*b^3*c^4*d^15 + 148*a^12*b^3*c^6*d^13 - 118*a^12*b^3*c^8*d^11 - 174*a^12*b
^3*c^10*d^9 - 56*a^12*b^3*c^12*d^7 - 112*a^13*b^2*c^3*d^16 - 128*a^13*b^2*c
^5*d^14 - 32*a^13*b^2*c^7*d^12 + 32*a^13*b^2*c^9*d^10 + 16*a^13*b^2*c^11*d^
8))/(a^10*d^14 + b^10*c^14 + 2*a^2*b^8*c^14 + a^4*b^6*c^14 + a^6*b^4*d^14 +
 2*a^8*b^2*d^14 + 4*a^10*c^2*d^12 + 6*a^10*c^4*d^10 + 4*a^10*c^6*d^8 + a^10
*c^8*d^6 + b^10*c^6*d^8 + 4*b^10*c^8*d^6 + 6*b^10*c^10*d^4 + 4*b^10*c^12*d^
2 - 6*a*b^9*c^5*d^9 - 24*a*b^9*c^7*d^7 - 36*a*b^9*c^9*d^5 - 24*a*b^9*c^11*d
^3 - 12*a^3*b^7*c^13*d - 6*a^5*b^5*c*d^13 - 6*a^5*b^5*c^13*d - 12*a^7*b^3*c
*d^13 - 24*a^9*b*c^3*d^11 - 36*a^9*b*c^5*d^9 - 24*a^9*b*c^7*d^7 - 6*a^9*b*c
^9*d^5 + 15*a^2*b^8*c^4*d^10 + 62*a^2*b^8*c^6*d^8 + 98*a^2*b^8*c^8*d^6 + 72
*a^2*b^8*c^10*d^4 + 23*a^2*b^8*c^12*d^2 - 20*a^3*b^7*c^3*d^11 - 92*a^3*b^7*
c^5*d^9 - 168*a^3*b^7*c^7*d^7 - 152*a^3*b^7*c^9*d^5 - 68*a^3*b^7*c^11*d^3 +
 15*a^4*b^6*c^2*d^12 + 90*a^4*b^6*c^4*d^10 + 211*a^4*b^6*c^6*d^8 + 244*a^4*
b^6*c^8*d^6 + 141*a^4*b^6*c^10*d^4 + 34*a^4*b^6*c^12*d^2 - 64*a^5*b^5*c^3*d
^11 - 202*a^5*b^5*c^5*d^9 - 288*a^5*b^5*c^7*d^7 - 202*a^5*b^5*c^9*d^5 - 64*
a^5*b^5*c^11*d^3 + 34*a^6*b^4*c^2*d^12 + 141*a^6*b^4*c^4*d^10 + 244*a^6*b^4
*c^6*d^8 + 211*a^6*b^4*c^8*d^6 + 90*a^6*b^4*c^10*d^4 + 15*a^6*b^4*c^12*d^2
- 68*a^7*b^3*c^3*d^11 - 152*a^7*b^3*c^5*d^9 - 168*a^7*b^3*c^7*d^7 - 92*a^7*
b^3*c^9*d^5 - 20*a^7*b^3*c^11*d^3 + 23*a^8*b^2*c^2*d^12 + 72*a^8*b^2*c^4*d^
10 + 98*a^8*b^2*c^6*d^8 + 62*a^8*b^2*c^8*d^6 + 15*a^8*b^2*c^10*d^4 - 6*a*b^
9*c^13*d - 6*a^9*b*c*d^13)) - (C*b^13*c^15*d - A*b^13*c^15*d - B*a^12*b*d^1
6 + 12*A*a^3*b^10*d^16 + 20*A*a^5*b^8*d^16 - 4*A*a^9*b^4*d^16 + 4*A*a^11*b^

2*d^16 - 8*B*a^4*b^9*d^16 - 16*B*a^6*b^7*d^16 - B*a^8*b^5*d^16 + 6*B*a^10*b
^3*d^16 - 12*A*b^13*c^3*d^13 - 48*A*b^13*c^5*d^11 - 76*A*b^13*c^7*d^9 - 45*
A*b^13*c^9*d^7 + 5*A*b^13*c^11*d^5 + 9*A*b^13*c^13*d^3 + 4*C*a^5*b^8*d^16 +
12*C*a^7*b^6*d^16 + 4*C*a^9*b^4*d^16 - 4*C*a^11*b^2*d^16 + 4*B*b^13*c^4*d^
12 + 16*B*b^13*c^6*d^10 + 35*B*b^13*c^8*d^8 + 33*B*b^13*c^10*d^6 + 5*B*b^13
*c^12*d^4 - 5*B*b^13*c^14*d^2 + 4*C*b^13*c^7*d^9 - 3*C*b^13*c^9*d^7 - 17*C*
b^13*c^11*d^5 - 9*C*b^13*c^13*d^3 + 36*A*a*b^12*c^2*d^14 + 176*A*a*b^12*c^4
*d^12 + 380*A*a*b^12*c^6*d^10 + 396*A*a*b^12*c^8*d^8 + 176*A*a*b^12*c^10*d^
6 + 20*A*a*b^12*c^12*d^4 - 36*A*a^2*b^11*c*d^15 - 2*A*a^2*b^11*c^15*d - 92*
A*a^4*b^9*c*d^15 - A*a^4*b^9*c^15*d - 56*A*a^6*b^7*c*d^15 + 3*A*a^8*b^5*c*d
^15 + 2*A*a^10*b^3*c*d^15 - 3*A*a^12*b*c^3*d^13 - 3*A*a^12*b*c^5*d^11 - A*a
^12*b*c^7*d^9 - 4*B*a*b^12*c^3*d^13 - 24*B*a*b^12*c^5*d^11 - 116*B*a*b^12*c
^7*d^9 - 196*B*a*b^12*c^9*d^7 - 120*B*a*b^12*c^11*d^5 - 20*B*a*b^12*c^13*d^
3 + 20*B*a^3*b^10*c*d^15 + 68*B*a^5*b^8*c*d^15 + 56*B*a^7*b^6*c*d^15 + 4*B*
a^9*b^4*c*d^15 - 4*B*a^11*b^2*c*d^15 - 3*B*a^12*b*c^2*d^14 - 3*B*a^12*b*c^4
*d^12 - B*a^12*b*c^6*d^10 - 8*C*a*b^12*c^4*d^12 - 56*C*a*b^12*c^6*d^10 - 60
*C*a*b^12*c^8*d^8 + 28*C*a*b^12*c^10*d^6 + 52*C*a*b^12*c^12*d^4 + 12*C*a*b^
12*c^14*d^2 + 2*C*a^2*b^11*c^15*d - 4*C*a^4*b^9*c*d^15 + C*a^4*b^9*c^15*d -
40*C*a^6*b^7*c*d^15 - 51*C*a^8*b^5*c*d^15 - 14*C*a^10*b^3*c*d^15 + 3*C*a^1
2*b*c^3*d^13 + 3*C*a^12*b*c^5*d^11 + C*a^12*b*c^7*d^9 - 260*A*a^2*b^11*c^3*
d^13 - 780*A*a^2*b^11*c^5*d^11 - 1144*A*a^2*b^11*c^7*d^9 - 798*A*a^2*b^11*c
^9*d^7 - 202*A*a^2*b^11*c^11*d^5 + 6*A*a^2*b^11*c^13*d^3 + 204*A*a^3*b^10*c
^2*d^14 + 876*A*a^3*b^10*c^4*d^12 + 1812*A*a^3*b^10*c^6*d^10 + 1872*A*a^3*b
^10*c^8*d^8 + 872*A*a^3*b^10*c^10*d^6 + 136*A*a^3*b^10*c^12*d^4 + 8*A*a^3*b
^10*c^14*d^2 - 608*A*a^4*b^9*c^3*d^13 - 1866*A*a^4*b^9*c^5*d^11 - 2802*A*a^
4*b^9*c^7*d^9 - 2007*A*a^4*b^9*c^9*d^7 - 585*A*a^4*b^9*c^11*d^5 - 31*A*a^4*
b^9*c^13*d^3 + 264*A*a^5*b^8*c^2*d^14 + 1180*A*a^5*b^8*c^4*d^12 + 2528*A*a^
5*b^8*c^6*d^10 + 2628*A*a^5*b^8*c^8*d^8 + 1200*A*a^5*b^8*c^10*d^6 + 172*A*a
^5*b^8*c^12*d^4 + 8*A*a^5*b^8*c^14*d^2 - 356*A*a^6*b^7*c^3*d^13 - 1320*A*a^
6*b^7*c^5*d^11 - 2188*A*a^6*b^7*c^7*d^9 - 1588*A*a^6*b^7*c^9*d^7 - 448*A*a^
6*b^7*c^11*d^5 - 28*A*a^6*b^7*c^13*d^3 + 24*A*a^7*b^6*c^2*d^14 + 368*A*a^7*
b^6*c^4*d^12 + 1112*A*a^7*b^6*c^6*d^10 + 1272*A*a^7*b^6*c^8*d^8 + 560*A*a^7
*b^6*c^10*d^6 + 56*A*a^7*b^6*c^12*d^4 + 33*A*a^8*b^5*c^3*d^13 - 165*A*a^8*b
^5*c^5*d^11 - 487*A*a^8*b^5*c^7*d^9 - 362*A*a^8*b^5*c^9*d^7 - 70*A*a^8*b^5*
c^11*d^5 - 68*A*a^9*b^4*c^2*d^14 - 108*A*a^9*b^4*c^4*d^12 + 28*A*a^9*b^4*c^
6*d^10 + 128*A*a^9*b^4*c^8*d^8 + 56*A*a^9*b^4*c^10*d^6 + 26*A*a^10*b^3*c^3*
d^13 + 18*A*a^10*b^3*c^5*d^11 - 34*A*a^10*b^3*c^7*d^9 - 28*A*a^10*b^3*c^9*d
^7 + 4*A*a^11*b^2*c^2*d^14 + 4*A*a^11*b^2*c^4*d^12 + 12*A*a^11*b^2*c^6*d^10
+ 8*A*a^11*b^2*c^8*d^8 - 12*B*a^2*b^11*c^2*d^14 - 44*B*a^2*b^11*c^4*d^12 +
48*B*a^2*b^11*c^6*d^10 + 302*B*a^2*b^11*c^8*d^8 + 342*B*a^2*b^11*c^10*d^6
+ 118*B*a^2*b^11*c^12*d^4 - 2*B*a^2*b^11*c^14*d^2 + 132*B*a^3*b^10*c^3*d^13
+ 284*B*a^3*b^10*c^5*d^11 + 28*B*a^3*b^10*c^7*d^9 - 424*B*a^3*b^10*c^9*d^7
- 336*B*a^3*b^10*c^11*d^5 - 56*B*a^3*b^10*c^13*d^3 - 132*B*a^4*b^9*c^2*d^1
4 - 558*B*a^4*b^9*c^4*d^12 - 694*B*a^4*b^9*c^6*d^10 - 27*B*a^4*b^9*c^8*d^8
+ 411*B*a^4*b^9*c^10*d^6 + 181*B*a^4*b^9*c^12*d^4 + 3*B*a^4*b^9*c^14*d^2 +

496*B*a^5*b^8*c^3*d^13 + 1196*B*a^5*b^8*c^5*d^11 + 1032*B*a^5*b^8*c^7*d^9 + 84*B*a^5*b^8*c^9*d^7 - 216*B*a^5*b^8*c^11*d^5 - 36*B*a^5*b^8*c^13*d^3 - 24 4*B*a^6*b^7*c^2*d^14 - 1064*B*a^6*b^7*c^4*d^12 - 1596*B*a^6*b^7*c^6*d^10 - 828*B*a^6*b^7*c^8*d^8 + 68*B*a^6*b^7*c^12*d^4 + 488*B*a^7*b^6*c^3*d^13 + 12 24*B*a^7*b^6*c^5*d^11 + 1208*B*a^7*b^6*c^7*d^9 + 416*B*a^7*b^6*c^9*d^7 - 10 3*B*a^8*b^5*c^2*d^14 - 581*B*a^8*b^5*c^4*d^12 - 959*B*a^8*b^5*c^6*d^10 - 58 2*B*a^8*b^5*c^8*d^8 - 102*B*a^8*b^5*c^10*d^6 + 132*B*a^9*b^4*c^3*d^13 + 356 *B*a^9*b^4*c^5*d^11 + 332*B*a^9*b^4*c^7*d^9 + 104*B*a^9*b^4*c^9*d^7 + 18*B* a^10*b^3*c^2*d^14 - 30*B*a^10*b^3*c^4*d^12 - 90*B*a^10*b^3*c^6*d^10 - 48*B* a^10*b^3*c^8*d^8 + 4*B*a^11*b^2*c^3*d^13 + 20*B*a^11*b^2*c^5*d^11 + 12*B*a^ 11*b^2*c^7*d^9 + 20*C*a^2*b^11*c^3*d^13 + 156*C*a^2*b^11*c^5*d^11 + 328*C*a ^2*b^11*c^7*d^9 + 234*C*a^2*b^11*c^9*d^7 + 10*C*a^2*b^11*c^11*d^5 - 30*C*a^ 2*b^11*c^13*d^3 - 12*C*a^3*b^10*c^2*d^14 - 168*C*a^3*b^10*c^4*d^12 - 636*C* a^3*b^10*c^6*d^10 - 828*C*a^3*b^10*c^8*d^8 - 344*C*a^3*b^10*c^10*d^6 + 20*C *a^3*b^10*c^12*d^4 + 16*C*a^3*b^10*c^14*d^2 + 56*C*a^4*b^9*c^3*d^13 + 570*C *a^4*b^9*c^5*d^11 + 1218*C*a^4*b^9*c^7*d^9 + 951*C*a^4*b^9*c^9*d^7 + 225*C* a^4*b^9*c^11*d^5 - 17*C*a^4*b^9*c^13*d^3 + 36*C*a^5*b^8*c^2*d^14 - 172*C*a^ 5*b^8*c^4*d^12 - 1004*C*a^5*b^8*c^6*d^10 - 1452*C*a^5*b^8*c^8*d^8 - 732*C*a ^5*b^8*c^10*d^6 - 76*C*a^5*b^8*c^12*d^4 + 4*C*a^5*b^8*c^14*d^2 - 124*C*a^6* b^7*c^3*d^13 + 336*C*a^6*b^7*c^5*d^11 + 1132*C*a^6*b^7*c^7*d^9 + 964*C*a^6* b^7*c^9*d^7 + 256*C*a^6*b^7*c^11*d^5 + 4*C*a^6*b^7*c^13*d^3 + 144*C*a^7*b^6 *c^2*d^14 + 196*C*a^7*b^6*c^4*d^12 - 296*C*a^7*b^6*c^6*d^10 - 708*C*a^7*b^6 *c^8*d^8 - 392*C*a^7*b^6*c^10*d^6 - 44*C*a^7*b^6*c^12*d^4 - 237*C*a^8*b^5*c ^3*d^13 - 171*C*a^8*b^5*c^5*d^11 + 223*C*a^8*b^5*c^7*d^9 + 266*C*a^8*b^5*c^ 9*d^7 + 58*C*a^8*b^5*c^11*d^5 + 92*C*a^9*b^4*c^2*d^14 + 204*C*a^9*b^4*c^4*d ^12 + 116*C*a^9*b^4*c^6*d^10 - 32*C*a^9*b^4*c^8*d^8 - 32*C*a^9*b^4*c^10*d^6 - 74*C*a^10*b^3*c^3*d^13 - 90*C*a^10*b^3*c^5*d^11 - 14*C*a^10*b^3*c^7*d^9 + 16*C*a^10*b^3*c^9*d^7 - 4*C*a^11*b^2*c^2*d^14 - 4*C*a^11*b^2*c^4*d^12 - 1 2*C*a^11*b^2*c^6*d^10 - 8*C*a^11*b^2*c^8*d^8 - A*a^12*b*c*d^15 + C*a^12*b*c *d^15)/(a^10*d^14 + b^10*c^14 + 2*a^2*b^8*c^14 + a^4*b^6*c^14 + a^6*b^4*d^1 4 + 2*a^8*b^2*d^14 + 4*a^10*c^2*d^12 + 6*a^10*c^4*d^10 + 4*a^10*c^6*d^8 + a ^10*c^8*d^6 + b^10*c^6*d^8 + 4*b^10*c^8*d^6 + 6*b^10*c^10*d^4 + 4*b^10*c^12 *d^2 - 6*a*b^9*c^5*d^9 - 24*a*b^9*c^7*d^7 - 36*a*b^9*c^9*d^5 - 24*a*b^9*c^1 1*d^3 - 12*a^3*b^7*c^13*d - 6*a^5*b^5*c*d^13 - 6*a^5*b^5*c^13*d - 12*a^7*b^ 3*c*d^13 - 24*a^9*b*c^3*d^11 - 36*a^9*b*c^5*d^9 - 24*a^9*b*c^7*d^7 - 6*a^9* b*c^9*d^5 + 15*a^2*b^8*c^4*d^10 + 62*a^2*b^8*c^6*d^8 + 98*a^2*b^8*c^8*d^6 + 72*a^2*b^8*c^10*d^4 + 23*a^2*b^8*c^12*d^2 - 20*a^3*b^7*c^3*d^11 - 92*a^3*b ^7*c^5*d^9 - 168*a^3*b^7*c^7*d^7 - 152*a^3*b^7*c^9*d^5 - 68*a^3*b^7*c^11*d^ 3 + 15*a^4*b^6*c^2*d^12 + 90*a^4*b^6*c^4*d^10 + 211*a^4*b^6*c^6*d^8 + 244*a ^4*b^6*c^8*d^6 + 141*a^4*b^6*c^10*d^4 + 34*a^4*b^6*c^12*d^2 - 64*a^5*b^5*c^ 3*d^11 - 202*a^5*b^5*c^5*d^9 - 288*a^5*b^5*c^7*d^7 - 202*a^5*b^5*c^9*d^5 - 64*a^5*b^5*c^11*d^3 + 34*a^6*b^4*c^2*d^12 + 141*a^6*b^4*c^4*d^10 + 244*a^6* b^4*c^6*d^8 + 211*a^6*b^4*c^8*d^6 + 90*a^6*b^4*c^10*d^4 + 15*a^6*b^4*c^12*d ^2 - 68*a^7*b^3*c^3*d^11 - 152*a^7*b^3*c^5*d^9 - 168*a^7*b^3*c^7*d^7 - 92*a ^7*b^3*c^9*d^5 - 20*a^7*b^3*c^11*d^3 + 23*a^8*b^2*c^2*d^12 + 72*a^8*b^2*c^4 *d^10 + 98*a^8*b^2*c^6*d^8 + 62*a^8*b^2*c^8*d^6 + 15*a^8*b^2*c^10*d^4 - 6*a *b^9*c^13*d - 6*a^9*b*c*d^13) + (tan(e + f*x)*(3*C*a^12*b*d^16 - 3*A*a^12*b *d^16 + 3*B*b^13*c^15*d + 24*A*a^4*b^9*d^16 + 56*A*a^6*b^7*d^16 + 25*A*a^8* b^5*d^16 - 10*A*a^10*b^3*d^16 - 16*B*a^5*b^8*d^16 - 48*B*a^7*b^6*d^16 - 36* B*a^9*b^4*d^16 - 4*B*a^11*b^2*d^16 - 24*A*b^13*c^4*d^12 - 104*A*b^13*c^6*d^ 10 - 199*A*b^13*c^8*d^8 - 189*A*b^13*c^10*d^6 - 77*A*b^13*c^12*d^4 - 7*A*b^ 13*c^14*d^2 + 4*C*a^6*b^7*d^16 + 23*C*a^8*b^5*d^16 + 22*C*a^10*b^3*d^16 + 8 *B*b^13*c^5*d^11 + 24*B*b^13*c^7*d^9 + 51*B*b^13*c^9*d^7 + 65*B*b^13*c^11*d ^5 + 33*B*b^13*c^13*d^3 - 4*C*b^13*c^6*d^10 + 7*C*b^13*c^8*d^8 + 21*C*b^13* c^10*d^6 + 5*C*b^13*c^12*d^4 - 5*C*b^13*c^14*d^2 + 48*A*a*b^12*c^3*d^13 + 2 08*A*a*b^12*c^5*d^11 + 472*A*a*b^12*c^7*d^9 + 572*A*a*b^12*c^9*d^7 + 324*A* a*b^12*c^11*d^5 + 68*A*a*b^12*c^13*d^3 - 48*A*a^3*b^10*c*d^15 + 4*A*a^3*b^1 0*c^15*d - 144*A*a^5*b^8*c*d^15 - 104*A*a^7*b^6*c*d^15 + 4*A*a^9*b^4*c*d^15 + 12*A*a^11*b^2*c*d^15 - A*a^12*b*c^2*d^14 + 7*A*a^12*b*c^4*d^12 + 5*A*a^1 2*b*c^6*d^10 + 64*B*a*b^12*c^6*d^10 + 100*B*a*b^12*c^8*d^8 - 4*B*a*b^12*c^1 0*d^6 - 52*B*a*b^12*c^12*d^4 - 12*B*a*b^12*c^14*d^2 + 2*B*a^2*b^11*c^15*d + 24*B*a^4*b^9*c*d^15 - B*a^4*b^9*c^15*d + 120*B*a^6*b^7*c*d^15 + 147*B*a^8* b^5*c*d^15 + 58*B*a^10*b^3*c*d^15 + 13*B*a^12*b*c^3*d^13 + 5*B*a^12*b*c^5*d ^11 - B*a^12*b*c^7*d^9 + 8*C*a*b^12*c^5*d^11 - 88*C*a*b^12*c^7*d^9 - 236*C* a*b^12*c^9*d^7 - 180*C*a*b^12*c^11*d^5 - 44*C*a*b^12*c^13*d^3 - 4*C*a^3*b^1 0*c^15*d + 24*C*a^5*b^8*c*d^15 + 8*C*a^7*b^6*c*d^15 - 28*C*a^9*b^4*c*d^15 - 12*C*a^11*b^2*c*d^15 + C*a^12*b*c^2*d^14 - 7*C*a^12*b*c^4*d^12 - 5*C*a^12* b*c^6*d^10 - 24*A*a^2*b^11*c^4*d^12 - 316*A*a^2*b^11*c^6*d^10 - 838*A*a^2*b ^11*c^8*d^8 - 858*A*a^2*b^11*c^10*d^6 - 346*A*a^2*b^11*c^12*d^4 - 34*A*a^2* b^11*c^14*d^2 - 192*A*a^3*b^10*c^3*d^13 - 200*A*a^3*b^10*c^5*d^11 + 472*A*a ^3*b^10*c^7*d^9 + 1148*A*a^3*b^10*c^9*d^7 + 756*A*a^3*b^10*c^11*d^5 + 140*A *a^3*b^10*c^13*d^3 + 200*A*a^4*b^9*c^2*d^14 + 790*A*a^4*b^9*c^4*d^12 + 906* A*a^4*b^9*c^6*d^10 - 177*A*a^4*b^9*c^8*d^8 - 795*A*a^4*b^9*c^10*d^6 - 353*A *a^4*b^9*c^12*d^4 - 27*A*a^4*b^9*c^14*d^2 - 936*A*a^5*b^8*c^3*d^13 - 2016*A *a^5*b^8*c^5*d^11 - 1512*A*a^5*b^8*c^7*d^9 + 72*A*a^5*b^8*c^9*d^7 + 432*A*a ^5*b^8*c^11*d^5 + 72*A*a^5*b^8*c^13*d^3 + 468*A*a^6*b^7*c^2*d^14 + 1768*A*a ^6*b^7*c^4*d^12 + 2524*A*a^6*b^7*c^6*d^10 + 1252*A*a^6*b^7*c^8*d^8 - 84*A*a ^6*b^7*c^12*d^4 - 952*A*a^7*b^6*c^3*d^13 - 2264*A*a^7*b^6*c^5*d^11 - 2088*A *a^7*b^6*c^7*d^9 - 672*A*a^7*b^6*c^9*d^7 + 283*A*a^8*b^5*c^2*d^14 + 1137*A* a^8*b^5*c^4*d^12 + 1651*A*a^8*b^5*c^6*d^10 + 898*A*a^8*b^5*c^8*d^8 + 126*A* a^8*b^5*c^10*d^6 - 268*A*a^9*b^4*c^3*d^13 - 716*A*a^9*b^4*c^5*d^11 - 612*A* a^9*b^4*c^7*d^9 - 168*A*a^9*b^4*c^9*d^7 + 14*A*a^10*b^3*c^2*d^14 + 166*A*a^ 10*b^3*c^4*d^12 + 250*A*a^10*b^3*c^6*d^10 + 108*A*a^10*b^3*c^8*d^8 - 12*A*a ^11*b^2*c^3*d^13 - 60*A*a^11*b^2*c^5*d^11 - 36*A*a^11*b^2*c^7*d^9 - 32*B*a^ 2*b^11*c^3*d^13 - 280*B*a^2*b^11*c^5*d^11 - 612*B*a^2*b^11*c^7*d^9 - 474*B* a^2*b^11*c^9*d^7 - 70*B*a^2*b^11*c^11*d^5 + 42*B*a^2*b^11*c^13*d^3 + 16*B*a ^3*b^10*c^2*d^14 + 240*B*a^3*b^10*c^4*d^12 + 968*B*a^3*b^10*c^6*d^10 + 1348 *B*a^3*b^10*c^8*d^8 + 668*B*a^3*b^10*c^10*d^6 + 60*B*a^3*b^10*c^12*d^4 - 4* B*a^3*b^10*c^14*d^2 + 8*B*a^4*b^9*c^3*d^13 - 814*B*a^4*b^9*c^5*d^11 - 2034* B*a^4*b^9*c^7*d^9 - 1731*B*a^4*b^9*c^9*d^7 - 513*B*a^4*b^9*c^11*d^5 - 19*B*

a^4*b^9*c^13*d^3 - 128*B*a^5*b^8*c^2*d^14 + 144*B*a^5*b^8*c^4*d^12 + 1472*B
*a^5*b^8*c^6*d^10 + 2232*B*a^5*b^8*c^8*d^8 + 1176*B*a^5*b^8*c^10*d^6 + 168*
B*a^5*b^8*c^12*d^4 + 8*B*a^5*b^8*c^14*d^2 + 460*B*a^6*b^7*c^3*d^13 - 152*B*
a^6*b^7*c^5*d^11 - 1596*B*a^6*b^7*c^7*d^9 - 1524*B*a^6*b^7*c^9*d^7 - 448*B*
a^6*b^7*c^11*d^5 - 28*B*a^6*b^7*c^13*d^3 - 408*B*a^7*b^6*c^2*d^14 - 576*B*a
^7*b^6*c^4*d^12 + 328*B*a^7*b^6*c^6*d^10 + 1048*B*a^7*b^6*c^8*d^8 + 560*B*a
^7*b^6*c^10*d^6 + 56*B*a^7*b^6*c^12*d^4 + 617*B*a^8*b^5*c^3*d^13 + 587*B*a^
8*b^5*c^5*d^11 - 159*B*a^8*b^5*c^7*d^9 - 346*B*a^8*b^5*c^9*d^7 - 70*B*a^8*b
^5*c^11*d^5 - 316*B*a^9*b^4*c^2*d^14 - 564*B*a^9*b^4*c^4*d^12 - 268*B*a^9*b
^4*c^6*d^10 + 72*B*a^9*b^4*c^8*d^8 + 56*B*a^9*b^4*c^10*d^6 + 210*B*a^10*b^3
*c^3*d^13 + 218*B*a^10*b^3*c^5*d^11 + 38*B*a^10*b^3*c^7*d^9 - 28*B*a^10*b^3
*c^9*d^7 - 52*B*a^11*b^2*c^2*d^14 - 84*B*a^11*b^2*c^4*d^12 - 28*B*a^11*b^2*
c^6*d^10 + 8*B*a^11*b^2*c^8*d^8 - 36*C*a^2*b^11*c^4*d^12 + 52*C*a^2*b^11*c^
6*d^10 + 382*C*a^2*b^11*c^8*d^8 + 474*C*a^2*b^11*c^10*d^6 + 190*C*a^2*b^11*
c^12*d^4 + 10*C*a^2*b^11*c^14*d^2 + 96*C*a^3*b^10*c^3*d^13 + 344*C*a^3*b^10
*c^5*d^11 + 104*C*a^3*b^10*c^7*d^9 - 524*C*a^3*b^10*c^9*d^7 - 468*C*a^3*b^1
0*c^11*d^5 - 92*C*a^3*b^10*c^13*d^3 - 92*C*a^4*b^9*c^2*d^14 - 646*C*a^4*b^9
*c^4*d^12 - 942*C*a^4*b^9*c^6*d^10 - 87*C*a^4*b^9*c^8*d^8 + 543*C*a^4*b^9*c
^10*d^6 + 257*C*a^4*b^9*c^12*d^4 + 15*C*a^4*b^9*c^14*d^2 + 504*C*a^5*b^8*c^
3*d^13 + 1512*C*a^5*b^8*c^5*d^11 + 1416*C*a^5*b^8*c^7*d^9 + 144*C*a^5*b^8*c
^9*d^7 - 288*C*a^5*b^8*c^11*d^5 - 48*C*a^5*b^8*c^13*d^3 - 204*C*a^6*b^7*c^2
*d^14 - 1324*C*a^6*b^7*c^4*d^12 - 2188*C*a^6*b^7*c^6*d^10 - 1168*C*a^6*b^7*
c^8*d^8 - 24*C*a^6*b^7*c^10*d^6 + 72*C*a^6*b^7*c^12*d^4 + 568*C*a^7*b^6*c^3
*d^13 + 1688*C*a^7*b^6*c^5*d^11 + 1704*C*a^7*b^6*c^7*d^9 + 576*C*a^7*b^6*c^
9*d^7 - 79*C*a^8*b^5*c^2*d^14 - 801*C*a^8*b^5*c^4*d^12 - 1387*C*a^8*b^5*c^6
*d^10 - 802*C*a^8*b^5*c^8*d^8 - 114*C*a^8*b^5*c^10*d^6 + 172*C*a^9*b^4*c^3*
d^13 + 572*C*a^9*b^4*c^5*d^11 + 516*C*a^9*b^4*c^7*d^9 + 144*C*a^9*b^4*c^9*d
^7 + 34*C*a^10*b^3*c^2*d^14 - 94*C*a^10*b^3*c^4*d^12 - 202*C*a^10*b^3*c^6*d
^10 - 96*C*a^10*b^3*c^8*d^8 + 12*C*a^11*b^2*c^3*d^13 + 60*C*a^11*b^2*c^5*d^
11 + 36*C*a^11*b^2*c^7*d^9 + 4*A*a*b^12*c^15*d + 7*B*a^12*b*c*d^15 - 4*C*a*
b^12*c^15*d))/(a^10*d^14 + b^10*c^14 + 2*a^2*b^8*c^14 + a^4*b^6*c^14 + a^6*
b^4*d^14 + 2*a^8*b^2*d^14 + 4*a^10*c^2*d^12 + 6*a^10*c^4*d^10 + 4*a^10*c^6*
d^8 + a^10*c^8*d^6 + b^10*c^6*d^8 + 4*b^10*c^8*d^6 + 6*b^10*c^10*d^4 + 4*b^
10*c^12*d^2 - 6*a*b^9*c^5*d^9 - 24*a*b^9*c^7*d^7 - 36*a*b^9*c^9*d^5 - 24*a*
b^9*c^11*d^3 - 12*a^3*b^7*c^13*d - 6*a^5*b^5*c*d^13 - 6*a^5*b^5*c^13*d - 12
*a^7*b^3*c*d^13 - 24*a^9*b*c^3*d^11 - 36*a^9*b*c^5*d^9 - 24*a^9*b*c^7*d^7 -
 6*a^9*b*c^9*d^5 + 15*a^2*b^8*c^4*d^10 + 62*a^2*b^8*c^6*d^8 + 98*a^2*b^8*c^
8*d^6 + 72*a^2*b^8*c^10*d^4 + 23*a^2*b^8*c^12*d^2 - 20*a^3*b^7*c^3*d^11 - 9
2*a^3*b^7*c^5*d^9 - 168*a^3*b^7*c^7*d^7 - 152*a^3*b^7*c^9*d^5 - 68*a^3*b^7*
c^11*d^3 + 15*a^4*b^6*c^2*d^12 + 90*a^4*b^6*c^4*d^10 + 211*a^4*b^6*c^6*d^8
+ 244*a^4*b^6*c^8*d^6 + 141*a^4*b^6*c^10*d^4 + 34*a^4*b^6*c^12*d^2 - 64*a^5
*b^5*c^3*d^11 - 202*a^5*b^5*c^5*d^9 - 288*a^5*b^5*c^7*d^7 - 202*a^5*b^5*c^9
*d^5 - 64*a^5*b^5*c^11*d^3 + 34*a^6*b^4*c^2*d^12 + 141*a^6*b^4*c^4*d^10 + 2
44*a^6*b^4*c^6*d^8 + 211*a^6*b^4*c^8*d^6 + 90*a^6*b^4*c^10*d^4 + 15*a^6*b^4
*c^12*d^2 - 68*a^7*b^3*c^3*d^11 - 152*a^7*b^3*c^5*d^9 - 168*a^7*b^3*c^7*d^7

 - 92*a^7*b^3*c^9*d^5 - 20*a^7*b^3*c^11*d^3 + 23*a^8*b^2*c^2*d^12 + 72*a^8*
b^2*c^4*d^10 + 98*a^8*b^2*c^6*d^8 + 62*a^8*b^2*c^8*d^6 + 15*a^8*b^2*c^10*d^
4 - 6*a*b^9*c^13*d - 6*a^9*b*c*d^13)) + (36*A^2*a^3*b^8*d^13 - 4*A^2*a^5*b^
6*d^13 - 3*A^2*a^7*b^4*d^13 + 16*B^2*a^3*b^8*d^13 + 16*B^2*a^5*b^6*d^13 + B
^2*a^7*b^4*d^13 + 2*B^2*a^9*b^2*d^13 + 156*A^2*b^11*c^3*d^10 + 204*A^2*b^11
*c^5*d^8 + 85*A^2*b^11*c^7*d^6 + 3*A^2*b^11*c^11*d^2 + 8*C^2*a^5*b^6*d^13 +
 9*C^2*a^7*b^4*d^13 + 4*B^2*b^11*c^3*d^10 + 28*B^2*b^11*c^5*d^8 + 45*B^2*b^
11*c^7*d^6 + 24*B^2*b^11*c^9*d^4 - B^2*b^11*c^11*d^2 + C^2*b^11*c^7*d^6 + 3
*C^2*b^11*c^11*d^2 + 36*A^2*a*b^10*d^13 + 36*A^2*b^11*c*d^12 + 168*A^2*a^2*
b^9*c^3*d^10 + 393*A^2*a^2*b^9*c^5*d^8 + 43*A^2*a^2*b^9*c^7*d^6 + 5*A^2*a^2
*b^9*c^9*d^4 + 7*A^2*a^2*b^9*c^11*d^2 + 8*A^2*a^3*b^8*c^2*d^11 - 417*A^2*a^
3*b^8*c^4*d^9 - 411*A^2*a^3*b^8*c^6*d^7 - 7*A^2*a^3*b^8*c^8*d^5 - 17*A^2*a^
3*b^8*c^10*d^3 + 87*A^2*a^4*b^7*c^3*d^10 + 359*A^2*a^4*b^7*c^5*d^8 - 75*A^2
*a^4*b^7*c^7*d^6 + 9*A^2*a^4*b^7*c^9*d^4 + 17*A^2*a^5*b^6*c^2*d^11 - 205*A^
2*a^5*b^6*c^4*d^9 - 13*A^2*a^5*b^6*c^6*d^7 + 37*A^2*a^5*b^6*c^8*d^5 + A^2*a
^6*b^5*c^3*d^10 + 13*A^2*a^6*b^5*c^5*d^8 - 89*A^2*a^6*b^5*c^7*d^6 + 23*A^2*
a^7*b^4*c^2*d^11 + 23*A^2*a^7*b^4*c^4*d^9 + 93*A^2*a^7*b^4*c^6*d^7 - 53*A^2
*a^8*b^3*c^5*d^8 - 8*A^2*a^9*b^2*c^2*d^11 + 16*A^2*a^9*b^2*c^4*d^9 + 48*B^2
*a^2*b^9*c^3*d^10 - 47*B^2*a^2*b^9*c^5*d^8 + 131*B^2*a^2*b^9*c^7*d^6 + 9*B^
2*a^2*b^9*c^9*d^4 - 5*B^2*a^2*b^9*c^11*d^2 + 36*B^2*a^3*b^8*c^2*d^11 + 163*
B^2*a^3*b^8*c^4*d^9 - 31*B^2*a^3*b^8*c^6*d^7 - 199*B^2*a^3*b^8*c^8*d^5 + 7*
B^2*a^3*b^8*c^10*d^3 - 49*B^2*a^4*b^7*c^3*d^10 - 209*B^2*a^4*b^7*c^5*d^8 +
149*B^2*a^4*b^7*c^7*d^6 - 19*B^2*a^4*b^7*c^9*d^4 - 11*B^2*a^5*b^6*c^2*d^11
+ 91*B^2*a^5*b^6*c^4*d^9 - 185*B^2*a^5*b^6*c^6*d^7 - 127*B^2*a^5*b^6*c^8*d^
5 - 39*B^2*a^6*b^5*c^3*d^10 + 13*B^2*a^6*b^5*c^5*d^8 + 119*B^2*a^6*b^5*c^7*
d^6 - 13*B^2*a^7*b^4*c^2*d^11 + 3*B^2*a^7*b^4*c^4*d^9 - 79*B^2*a^7*b^4*c^6*
d^7 - 20*B^2*a^8*b^3*c^3*d^10 + 43*B^2*a^8*b^3*c^5*d^8 + 12*B^2*a^9*b^2*c^2
*d^11 - 14*B^2*a^9*b^2*c^4*d^9 + 36*C^2*a^2*b^9*c^3*d^10 + 141*C^2*a^2*b^9*
c^5*d^8 - 65*C^2*a^2*b^9*c^7*d^6 + 17*C^2*a^2*b^9*c^9*d^4 + 7*C^2*a^2*b^9*c
^11*d^2 + 20*C^2*a^3*b^8*c^2*d^11 - 69*C^2*a^3*b^8*c^4*d^9 + 57*C^2*a^3*b^8
*c^6*d^7 + 113*C^2*a^3*b^8*c^8*d^5 - 65*C^2*a^3*b^8*c^10*d^3 + 99*C^2*a^4*b
^7*c^3*d^10 + 179*C^2*a^4*b^7*c^5*d^8 - 231*C^2*a^4*b^7*c^7*d^6 - 15*C^2*a^
4*b^7*c^9*d^4 + 41*C^2*a^5*b^6*c^2*d^11 - 97*C^2*a^5*b^6*c^4*d^9 + 143*C^2*
a^5*b^6*c^6*d^7 + 61*C^2*a^5*b^6*c^8*d^5 - 36*C^2*a^5*b^6*c^10*d^3 - 11*C^2
*a^6*b^5*c^3*d^10 - 119*C^2*a^6*b^5*c^5*d^8 - 221*C^2*a^6*b^5*c^7*d^6 - 36*
C^2*a^6*b^5*c^9*d^4 + 11*C^2*a^7*b^4*c^2*d^11 - 37*C^2*a^7*b^4*c^4*d^9 + 57
*C^2*a^7*b^4*c^6*d^7 - 53*C^2*a^8*b^3*c^5*d^8 - 8*C^2*a^9*b^2*c^2*d^11 + 16
*C^2*a^9*b^2*c^4*d^9 - 48*A*B*a^2*b^9*d^13 - 48*A*B*a^4*b^7*d^13 - A*B*a^8*
b^3*d^13 + 36*A*C*a^3*b^8*d^13 + 32*A*C*a^5*b^6*d^13 - 6*A*C*a^7*b^4*d^13 -
 24*A*B*b^11*c^2*d^11 - 136*A*B*b^11*c^4*d^9 - 200*A*B*b^11*c^6*d^7 - 89*A*
B*b^11*c^8*d^5 + 6*A*B*b^11*c^10*d^3 - 24*B*C*a^4*b^7*d^13 - 24*B*C*a^6*b^5
*d^13 + B*C*a^8*b^3*d^13 - 12*A*C*b^11*c^3*d^10 + 12*A*C*b^11*c^5*d^8 + 58*
A*C*b^11*c^7*d^6 + 36*A*C*b^11*c^9*d^4 - 6*A*C*b^11*c^11*d^2 + 4*B*C*b^11*c
^4*d^9 - 4*B*C*b^11*c^6*d^7 - 19*B*C*b^11*c^8*d^5 - 18*B*C*b^11*c^10*d^3 -
A^2*a*b^10*c^12*d + 2*A^2*a^10*b*c*d^12 + B^2*a*b^10*c^12*d - 2*B^2*a^10*b*

c*d^12 - C^2*a*b^10*c^12*d + 2*C^2*a^10*b*c*d^12 + 24*A^2*a*b^10*c^2*d^11 -
188*A^2*a*b^10*c^4*d^9 - 277*A^2*a*b^10*c^6*d^7 - 27*A^2*a*b^10*c^8*d^5 -
15*A^2*a*b^10*c^10*d^3 - 44*A^2*a^4*b^7*c*d^12 - 29*A^2*a^6*b^5*c*d^12 + A^
2*a^8*b^3*c*d^12 - 2*A^2*a^10*b*c^3*d^10 + 20*B^2*a*b^10*c^2*d^11 + 72*B^2*
a*b^10*c^4*d^9 + 47*B^2*a*b^10*c^6*d^7 - 89*B^2*a*b^10*c^8*d^5 + 5*B^2*a*b^
10*c^10*d^3 + 32*B^2*a^2*b^9*c*d^12 + 16*B^2*a^4*b^7*c*d^12 - 5*B^2*a^6*b^5
*c*d^12 - 11*B^2*a^8*b^3*c*d^12 + 2*B^2*a^10*b*c^3*d^10 - 8*C^2*a*b^10*c^4*
d^9 - C^2*a*b^10*c^6*d^7 + 69*C^2*a*b^10*c^8*d^5 - 27*C^2*a*b^10*c^10*d^3 +
16*C^2*a^4*b^7*c*d^12 - 5*C^2*a^6*b^5*c*d^12 + C^2*a^8*b^3*c*d^12 - 2*C^2*
a^10*b*c^3*d^10 + A*B*a^10*b*d^13 - A*B*b^11*c^12*d - B*C*a^10*b*d^13 + B*C
*b^11*c^12*d - 72*A*B*a*b^10*c*d^12 + 2*A*C*a*b^10*c^12*d - 4*A*C*a^10*b*c*
d^12 - 160*A*B*a*b^10*c^3*d^10 + 56*A*B*a*b^10*c^5*d^8 + 312*A*B*a*b^10*c^7
*d^6 - 8*A*B*a*b^10*c^9*d^4 + A*B*a^2*b^9*c^12*d - 24*A*B*a^3*b^8*c*d^12 +
40*A*B*a^5*b^6*c*d^12 + 32*A*B*a^7*b^4*c*d^12 - 6*A*B*a^10*b*c^2*d^11 + A*B
*a^10*b*c^4*d^9 + 84*A*C*a*b^10*c^2*d^11 + 268*A*C*a*b^10*c^4*d^9 + 206*A*C
*a*b^10*c^6*d^7 - 150*A*C*a*b^10*c^8*d^5 + 6*A*C*a*b^10*c^10*d^3 + 36*A*C*a
^2*b^9*c*d^12 - 8*A*C*a^4*b^7*c*d^12 - 2*A*C*a^6*b^5*c*d^12 - 2*A*C*a^8*b^3
*c*d^12 + 4*A*C*a^10*b*c^3*d^10 - 20*B*C*a*b^10*c^3*d^10 - 116*B*C*a*b^10*c
^5*d^8 - 180*B*C*a*b^10*c^7*d^6 + 92*B*C*a*b^10*c^9*d^4 - B*C*a^2*b^9*c^12*
d - 36*B*C*a^3*b^8*c*d^12 + 8*B*C*a^5*b^6*c*d^12 + 4*B*C*a^7*b^4*c*d^12 + 6
*B*C*a^10*b*c^2*d^11 - B*C*a^10*b*c^4*d^9 - 64*A*B*a^2*b^9*c^2*d^11 - 112*A
*B*a^2*b^9*c^4*d^9 - 508*A*B*a^2*b^9*c^6*d^7 - 23*A*B*a^2*b^9*c^8*d^5 + 30*
A*B*a^2*b^9*c^10*d^3 - 112*A*B*a^3*b^8*c^3*d^10 + 480*A*B*a^3*b^8*c^5*d^8 +
584*A*B*a^3*b^8*c^7*d^6 - 56*A*B*a^3*b^8*c^9*d^4 - 8*A*B*a^3*b^8*c^11*d^2
+ 40*A*B*a^4*b^7*c^2*d^11 + 114*A*B*a^4*b^7*c^4*d^9 - 456*A*B*a^4*b^7*c^6*d
^7 + 170*A*B*a^4*b^7*c^8*d^5 + 28*A*B*a^4*b^7*c^10*d^3 - 104*A*B*a^5*b^6*c^
3*d^10 + 368*A*B*a^5*b^6*c^5*d^8 + 104*A*B*a^5*b^6*c^7*d^6 - 56*A*B*a^5*b^6
*c^9*d^4 + 52*A*B*a^6*b^5*c^2*d^11 - 50*A*B*a^6*b^5*c^4*d^9 - 176*A*B*a^6*b
^5*c^6*d^7 + 70*A*B*a^6*b^5*c^8*d^5 + 40*A*B*a^7*b^4*c^3*d^10 + 144*A*B*a^7
*b^4*c^5*d^8 - 56*A*B*a^7*b^4*c^7*d^6 - 30*A*B*a^8*b^3*c^2*d^11 - 105*A*B*a
^8*b^3*c^4*d^9 + 28*A*B*a^8*b^3*c^6*d^7 + 40*A*B*a^9*b^2*c^3*d^10 - 8*A*B*a
^9*b^2*c^5*d^8 - 60*A*C*a^2*b^9*c^3*d^10 - 318*A*C*a^2*b^9*c^5*d^8 + 166*A*
C*a^2*b^9*c^7*d^6 + 14*A*C*a^2*b^9*c^9*d^4 - 14*A*C*a^2*b^9*c^11*d^2 + 188*
A*C*a^3*b^8*c^2*d^11 + 630*A*C*a^3*b^8*c^4*d^9 + 210*A*C*a^3*b^8*c^6*d^7 -
322*A*C*a^3*b^8*c^8*d^5 + 10*A*C*a^3*b^8*c^10*d^3 - 330*A*C*a^4*b^7*c^3*d^1
0 - 754*A*C*a^4*b^7*c^5*d^8 + 162*A*C*a^4*b^7*c^7*d^6 - 30*A*C*a^4*b^7*c^9*
d^4 + 50*A*C*a^5*b^6*c^2*d^11 + 374*A*C*a^5*b^6*c^4*d^9 - 202*A*C*a^5*b^6*c
^6*d^7 - 206*A*C*a^5*b^6*c^8*d^5 - 134*A*C*a^6*b^5*c^3*d^10 - 110*A*C*a^6*b
^5*c^5*d^8 + 166*A*C*a^6*b^5*c^7*d^6 - 34*A*C*a^7*b^4*c^2*d^11 + 14*A*C*a^7
*b^4*c^4*d^9 - 150*A*C*a^7*b^4*c^6*d^7 + 106*A*C*a^8*b^3*c^5*d^8 + 16*A*C*a
^9*b^2*c^2*d^11 - 32*A*C*a^9*b^2*c^4*d^9 - 68*B*C*a^2*b^9*c^2*d^11 - 140*B*
C*a^2*b^9*c^4*d^9 + 208*B*C*a^2*b^9*c^6*d^7 - 109*B*C*a^2*b^9*c^8*d^5 - 30*
B*C*a^2*b^9*c^10*d^3 + 4*B*C*a^3*b^8*c^3*d^10 - 300*B*C*a^3*b^8*c^5*d^8 - 1
40*B*C*a^3*b^8*c^7*d^6 + 272*B*C*a^3*b^8*c^9*d^4 + 8*B*C*a^3*b^8*c^11*d^2 -
160*B*C*a^4*b^7*c^2*d^11 - 174*B*C*a^4*b^7*c^4*d^9 + 420*B*C*a^4*b^7*c^6*d

^7 - 182*B*C*a^4*b^7*c^8*d^5 - 16*B*C*a^4*b^7*c^10*d^3 + 236*B*C*a^5*b^6*c^3*d^10 - 116*B*C*a^5*b^6*c^5*d^8 + 196*B*C*a^5*b^6*c^7*d^6 + 188*B*C*a^5*b^6*c^9*d^4 - 64*B*C*a^6*b^5*c^2*d^11 + 110*B*C*a^6*b^5*c^4*d^9 + 236*B*C*a^6*b^5*c^6*d^7 - 58*B*C*a^6*b^5*c^8*d^5 + 20*B*C*a^7*b^4*c^3*d^10 - 132*B*C*a^7*b^4*c^5*d^8 + 44*B*C*a^7*b^4*c^7*d^6 + 30*B*C*a^8*b^3*c^2*d^11 + 105*B*C*a^8*b^3*c^4*d^9 - 28*B*C*a^8*b^3*c^6*d^7 - 40*B*C*a^9*b^2*c^3*d^10 + 8*B*C*a^9*b^2*c^5*d^8)/(a^10*d^14 + b^10*c^14 + 2*a^2*b^8*c^14 + a^4*b^6*c^14 + a^6*b^4*d^14 + 2*a^8*b^2*d^14 + 4*a^10*c^2*d^12 + 6*a^10*c^4*d^10 + 4*a^10*c^6*d^8 + a^10*c^8*d^6 + b^10*c^6*d^8 + 4*b^10*c^8*d^6 + 6*b^10*c^10*d^4 + 4*b^10*c^12*d^2 - 6*a*b^9*c^5*d^9 - 24*a*b^9*c^7*d^7 - 36*a*b^9*c^9*d^5 - 24*a*b^9*c^11*d^3 - 12*a^3*b^7*c^13*d - 6*a^5*b^5*c*d^13 - 6*a^5*b^5*c^13*d - 12*a^7*b^3*c*d^13 - 24*a^9*b*c^3*d^11 - 36*a^9*b*c^5*d^9 - 24*a^9*b*c^7*d^7 - 6*a^9*b*c^9*d^5 + 15*a^2*b^8*c^4*d^10 + 62*a^2*b^8*c^6*d^8 + 98*a^2*b^8*c^8*d^6 + 72*a^2*b^8*c^10*d^4 + 23*a^2*b^8*c^12*d^2 - 20*a^3*b^7*c^3*d^11 - 92*a^3*b^7*c^5*d^9 - 168*a^3*b^7*c^7*d^7 - 152*a^3*b^7*c^9*d^5 - 68*a^3*b^7*c^11*d^3 + 15*a^4*b^6*c^2*d^12 + 90*a^4*b^6*c^4*d^10 + 211*a^4*b^6*c^6*d^8 + 244*a^4*b^6*c^8*d^6 + 141*a^4*b^6*c^10*d^4 + 34*a^4*b^6*c^12*d^2 - 64*a^5*b^5*c^3*d^11 - 202*a^5*b^5*c^5*d^9 - 288*a^5*b^5*c^7*d^7 - 202*a^5*b^5*c^9*d^5 - 64*a^5*b^5*c^11*d^3 + 34*a^6*b^4*c^2*d^12 + 141*a^6*b^4*c^4*d^10 + 244*a^6*b^4*c^6*d^8 + 211*a^6*b^4*c^8*d^6 + 90*a^6*b^4*c^10*d^4 + 15*a^6*b^4*c^12*d^2 - 68*a^7*b^3*c^3*d^11 - 152*a^7*b^3*c^5*d^9 - 168*a^7*b^3*c^7*d^7 - 92*a^7*b^3*c^9*d^5 - 20*a^7*b^3*c^11*d^3 + 23*a^8*b^2*c^2*d^12 + 72*a^8*b^2*c^4*d^10 + 98*a^8*b^2*c^6*d^8 + 62*a^8*b^2*c^8*d^6 + 15*a^8*b^2*c^10*d^4 - 6*a*b^9*c^13*d - 6*a^9*b*c*d^13) - (tan(e + f*x)*(10*A^2*a^4*b^7*d^13 - 6*A^2*a^2*b^9*d^13 - 18*A^2*b^11*d^13 + 12*A^2*a^6*b^5*d^13 - 3*A^2*a^8*b^3*d^13 - 8*B^2*a^2*b^9*d^13 - 8*B^2*a^4*b^7*d^13 - 18*B^2*a^6*b^5*d^13 - 2*B^2*a^8*b^3*d^13 - 54*A^2*b^11*c^2*d^11 - 18*A^2*b^11*c^4*d^9 + 20*A^2*b^11*c^6*d^7 - 65*A^2*b^11*c^8*d^5 - 2*C^2*a^4*b^7*d^13 + 6*C^2*a^6*b^5*d^13 - 9*C^2*a^8*b^3*d^13 - 2*B^2*b^11*c^2*d^11 - 6*B^2*b^11*c^4*d^9 + 12*B^2*b^11*c^6*d^7 + 66*B^2*b^11*c^8*d^5 - 18*B^2*b^11*c^10*d^3 + 2*C^2*b^11*c^6*d^7 - 29*C^2*b^11*c^8*d^5 + 36*C^2*b^11*c^10*d^3 - B^2*a^10*b*d^13 - A^2*b^11*c^12*d - C^2*b^11*c^12*d - 158*A^2*a^2*b^9*c^2*d^11 - 224*A^2*a^2*b^9*c^4*d^9 - 252*A^2*a^2*b^9*c^6*d^7 - 194*A^2*a^2*b^9*c^8*d^5 - 2*A^2*a^2*b^9*c^10*d^3 + 504*A^2*a^3*b^8*c^3*d^10 + 580*A^2*a^3*b^8*c^5*d^8 + 464*A^2*a^3*b^8*c^7*d^6 + 28*A^2*a^3*b^8*c^9*d^4 - 232*A^2*a^4*b^7*c^2*d^11 - 446*A^2*a^4*b^7*c^4*d^9 - 452*A^2*a^4*b^7*c^6*d^7 - 128*A^2*a^4*b^7*c^8*d^5 + 248*A^2*a^5*b^6*c^3*d^10 + 332*A^2*a^5*b^6*c^5*d^8 + 152*A^2*a^5*b^6*c^7*d^6 - 96*A^2*a^6*b^5*c^2*d^11 - 244*A^2*a^6*b^5*c^4*d^9 - 144*A^2*a^6*b^5*c^6*d^7 + 120*A^2*a^7*b^4*c^3*d^10 + 132*A^2*a^7*b^4*c^5*d^8 - 34*A^2*a^8*b^3*c^2*d^11 - 83*A^2*a^8*b^3*c^4*d^9 + 28*A^2*a^9*b^2*c^3*d^10 + 18*B^2*a^2*b^9*c^2*d^11 + 36*B^2*a^2*b^9*c^4*d^9 + 208*B^2*a^2*b^9*c^6*d^7 + 179*B^2*a^2*b^9*c^8*d^5 - 32*B^2*a^2*b^9*c^10*d^3 + 128*B^2*a^3*b^8*c^3*d^10 + 180*B^2*a^3*b^8*c^5*d^8 - 96*B^2*a^3*b^8*c^7*d^6 + 36*B^2*a^3*b^8*c^9*d^4 + 8*B^2*a^3*b^8*c^11*d^2 + 4*B^2*a^4*b^7*c^2*d^11 - 36*B^2*a^4*b^7*c^4*d^9 + 164*B^2*a^4*b^7*c^6*d^7 + 76*B^2*a^4*b^7*c^8*d^5 - 16*B^2*a^4*b^7*c^10*d^3 + 208*B^2*a^

5*b^6*c^3*d^10 + 148*B^2*a^5*b^6*c^5*d^8 + 16*B^2*a^5*b^6*c^7*d^6 - 84*B^2*
a^6*b^5*c^2*d^11 - 134*B^2*a^6*b^5*c^4*d^9 - 96*B^2*a^6*b^5*c^6*d^7 - 36*B^
2*a^6*b^5*c^8*d^5 + 40*B^2*a^7*b^4*c^3*d^10 + 20*B^2*a^7*b^4*c^5*d^8 + 48*B
^2*a^7*b^4*c^7*d^6 - 4*B^2*a^8*b^3*c^2*d^11 + 22*B^2*a^8*b^3*c^4*d^9 - 28*B
^2*a^8*b^3*c^6*d^7 - 12*B^2*a^9*b^2*c^3*d^10 + 8*B^2*a^9*b^2*c^5*d^8 - 8*C^
2*a^2*b^9*c^2*d^11 + 16*C^2*a^2*b^9*c^4*d^9 - 132*C^2*a^2*b^9*c^6*d^7 - 104
*C^2*a^2*b^9*c^8*d^5 + 64*C^2*a^2*b^9*c^10*d^3 + 64*C^2*a^3*b^8*c^5*d^8 + 3
56*C^2*a^3*b^8*c^7*d^6 + 64*C^2*a^3*b^8*c^9*d^4 - 12*C^2*a^3*b^8*c^11*d^2 +
 44*C^2*a^4*b^7*c^2*d^11 + 178*C^2*a^4*b^7*c^4*d^9 - 68*C^2*a^4*b^7*c^6*d^7
 - 68*C^2*a^4*b^7*c^8*d^5 + 12*C^2*a^4*b^7*c^10*d^3 - 4*C^2*a^5*b^6*c^3*d^1
0 + 80*C^2*a^5*b^6*c^5*d^8 + 164*C^2*a^5*b^6*c^7*d^6 + 72*C^2*a^5*b^6*c^9*d
^4 + 90*C^2*a^6*b^5*c^2*d^11 + 188*C^2*a^6*b^5*c^4*d^9 + 120*C^2*a^6*b^5*c^
6*d^7 + 6*C^2*a^6*b^5*c^8*d^5 - 18*C^2*a^6*b^5*c^10*d^3 + 36*C^2*a^7*b^4*c^
3*d^10 - 60*C^2*a^7*b^4*c^7*d^6 - 28*C^2*a^8*b^3*c^2*d^11 - 53*C^2*a^8*b^3*
c^4*d^9 + 18*C^2*a^8*b^3*c^6*d^7 + 28*C^2*a^9*b^2*c^3*d^10 + 16*A*B*a^3*b^8
*d^13 + 16*A*B*a^5*b^6*d^13 - 8*A*B*a^7*b^4*d^13 + 2*A*B*a^9*b^2*d^13 - 12*
A*C*a^2*b^9*d^13 + 10*A*C*a^4*b^7*d^13 + 12*A*C*a^8*b^3*d^13 + 36*A*B*b^11*
c^3*d^10 - 36*A*B*b^11*c^5*d^8 - 132*A*B*b^11*c^7*d^6 + 60*A*B*b^11*c^9*d^4
 - 4*A*B*b^11*c^11*d^2 + 8*B*C*a^3*b^8*d^13 - 4*B*C*a^5*b^6*d^13 + 20*B*C*a
^7*b^4*d^13 - 2*B*C*a^9*b^2*d^13 - 18*A*C*b^11*c^4*d^9 + 14*A*C*b^11*c^6*d^
7 + 148*A*C*b^11*c^8*d^5 - 18*A*C*b^11*c^10*d^3 + 6*B*C*b^11*c^5*d^8 + 18*B
*C*b^11*c^7*d^6 - 114*B*C*b^11*c^9*d^4 + 10*B*C*b^11*c^11*d^2 + 96*A^2*a*b^
10*c*d^12 - 8*B^2*a*b^10*c*d^12 + 336*A^2*a*b^10*c^3*d^10 + 372*A^2*a*b^10*
c^5*d^8 + 320*A^2*a*b^10*c^7*d^6 + 40*A^2*a*b^10*c^9*d^4 + 4*A^2*a*b^10*c^1
1*d^2 + 136*A^2*a^3*b^8*c*d^12 + 52*A^2*a^5*b^6*c*d^12 + 20*A^2*a^7*b^4*c*d
^12 + 4*A^2*a^9*b^2*c*d^12 - 4*A^2*a^10*b*c^2*d^11 - 16*B^2*a*b^10*c^3*d^10
 + 52*B^2*a*b^10*c^5*d^8 - 72*B^2*a*b^10*c^7*d^6 + 24*B^2*a*b^10*c^9*d^4 +
4*B^2*a*b^10*c^11*d^2 - B^2*a^2*b^9*c^12*d + 48*B^2*a^3*b^8*c*d^12 + 92*B^2
*a^5*b^6*c*d^12 + 36*B^2*a^7*b^4*c*d^12 + 4*B^2*a^9*b^2*c*d^12 + 2*B^2*a^10
*b*c^2*d^11 - B^2*a^10*b*c^4*d^9 - 24*C^2*a*b^10*c^5*d^8 + 140*C^2*a*b^10*c
^7*d^6 + 4*C^2*a*b^10*c^9*d^4 - 8*C^2*a*b^10*c^11*d^2 - 8*C^2*a^3*b^8*c*d^1
2 - 8*C^2*a^5*b^6*c*d^12 + 8*C^2*a^7*b^4*c*d^12 + 4*C^2*a^9*b^2*c*d^12 - 4*
C^2*a^10*b*c^2*d^11 + 24*A*B*a*b^10*d^13 + 12*A*B*b^11*c*d^12 + 2*A*C*b^11*
c^12*d + 2*A*B*a*b^10*c^12*d - 4*A*B*a^10*b*c*d^12 - 24*A*C*a*b^10*c*d^12 -
 2*B*C*a*b^10*c^12*d + 4*B*C*a^10*b*c*d^12 + 16*A*B*a*b^10*c^2*d^11 - 136*A
*B*a*b^10*c^4*d^9 + 8*A*B*a*b^10*c^6*d^7 - 174*A*B*a*b^10*c^8*d^5 - 4*A*B*a
*b^10*c^10*d^3 - 140*A*B*a^2*b^9*c*d^12 - 220*A*B*a^4*b^7*c*d^12 - 68*A*B*a
^6*b^5*c*d^12 - 12*A*B*a^8*b^3*c*d^12 + 4*A*B*a^10*b*c^3*d^10 - 48*A*C*a*b^
10*c^3*d^10 + 84*A*C*a*b^10*c^5*d^8 - 172*A*C*a*b^10*c^7*d^6 + 28*A*C*a*b^1
0*c^9*d^4 + 4*A*C*a*b^10*c^11*d^2 + 16*A*C*a^3*b^8*c*d^12 + 28*A*C*a^5*b^6*
c*d^12 - 28*A*C*a^7*b^4*c*d^12 - 8*A*C*a^9*b^2*c*d^12 + 8*A*C*a^10*b*c^2*d^
11 + 8*B*C*a*b^10*c^2*d^11 + 28*B*C*a*b^10*c^4*d^9 - 188*B*C*a*b^10*c^6*d^7
 + 114*B*C*a*b^10*c^8*d^5 + 16*B*C*a*b^10*c^10*d^3 + 20*B*C*a^2*b^9*c*d^12
- 14*B*C*a^4*b^7*c*d^12 - 52*B*C*a^6*b^5*c*d^12 - 6*B*C*a^8*b^3*c*d^12 - 4*
B*C*a^10*b*c^3*d^10 - 300*A*B*a^2*b^9*c^3*d^10 - 580*A*B*a^2*b^9*c^5*d^8 -

340*A*B*a^2*b^9*c^7*d^6 + 92*A*B*a^2*b^9*c^9*d^4 - 12*A*B*a^2*b^9*c^11*d^2 + 64*A*B*a^3*b^8*c^2*d^11 + 8*A*B*a^3*b^8*c^4*d^9 + 208*A*B*a^3*b^8*c^6*d^7 - 200*A*B*a^3*b^8*c^8*d^5 - 420*A*B*a^4*b^7*c^3*d^10 - 596*A*B*a^4*b^7*c^5*d^8 - 100*A*B*a^4*b^7*c^7*d^6 + 56*A*B*a^4*b^7*c^9*d^4 + 184*A*B*a^5*b^6*c^2*d^11 + 292*A*B*a^5*b^6*c^4*d^9 + 128*A*B*a^5*b^6*c^6*d^7 - 28*A*B*a^5*b^6*c^8*d^5 - 84*A*B*a^6*b^5*c^3*d^10 + 60*A*B*a^6*b^5*c^5*d^8 + 92*A*B*a^6*b^5*c^7*d^6 + 32*A*B*a^7*b^4*c^2*d^11 - 40*A*B*a^7*b^4*c^4*d^9 - 144*A*B*a^7*b^4*c^6*d^7 - 20*A*B*a^8*b^3*c^3*d^10 + 96*A*B*a^8*b^3*c^5*d^8 + 20*A*B*a^9*b^2*c^2*d^11 - 30*A*B*a^9*b^2*c^4*d^9 + 112*A*C*a^2*b^9*c^2*d^11 + 172*A*C*a^2*b^9*c^4*d^9 + 420*A*C*a^2*b^9*c^6*d^7 + 352*A*C*a^2*b^9*c^8*d^5 - 44*A*C*a^2*b^9*c^10*d^3 + 72*A*C*a^3*b^8*c^3*d^10 + 220*A*C*a^3*b^8*c^5*d^8 - 244*A*C*a^3*b^8*c^7*d^6 + 52*A*C*a^3*b^8*c^9*d^4 + 12*A*C*a^3*b^8*c^11*d^2 + 242*A*C*a^4*b^7*c^2*d^11 + 304*A*C*a^4*b^7*c^4*d^9 + 484*A*C*a^4*b^7*c^6*d^7 + 142*A*C*a^4*b^7*c^8*d^5 - 30*A*C*a^4*b^7*c^10*d^3 + 44*A*C*a^5*b^6*c^3*d^10 + 20*A*C*a^5*b^6*c^5*d^8 - 28*A*C*a^5*b^6*c^7*d^6 + 60*A*C*a^6*b^5*c^2*d^11 + 92*A*C*a^6*b^5*c^4*d^9 - 12*A*C*a^6*b^5*c^6*d^7 - 60*A*C*a^6*b^5*c^8*d^5 - 156*A*C*a^7*b^4*c^3*d^10 - 132*A*C*a^7*b^4*c^5*d^8 + 60*A*C*a^7*b^4*c^7*d^6 + 62*A*C*a^8*b^3*c^2*d^11 + 136*A*C*a^8*b^3*c^4*d^9 - 18*A*C*a^8*b^3*c^6*d^7 - 56*A*C*a^9*b^2*c^3*d^10 + 160*B*C*a^2*b^9*c^5*d^8 - 80*B*C*a^2*b^9*c^7*d^6 - 272*B*C*a^2*b^9*c^9*d^4 + 12*B*C*a^2*b^9*c^11*d^2 - 88*B*C*a^3*b^8*c^2*d^11 - 332*B*C*a^3*b^8*c^4*d^9 - 652*B*C*a^3*b^8*c^6*d^7 + 68*B*C*a^3*b^8*c^8*d^5 + 36*B*C*a^3*b^8*c^10*d^3 - 66*B*C*a^4*b^7*c^3*d^10 + 248*B*C*a^4*b^7*c^5*d^8 - 80*B*C*a^4*b^7*c^7*d^6 - 146*B*C*a^4*b^7*c^9*d^4 - 6*B*C*a^4*b^7*c^11*d^2 - 172*B*C*a^5*b^6*c^2*d^11 - 448*B*C*a^5*b^6*c^4*d^9 - 404*B*C*a^5*b^6*c^6*d^7 - 68*B*C*a^5*b^6*c^8*d^5 + 24*B*C*a^5*b^6*c^10*d^3 - 96*B*C*a^6*b^5*c^3*d^10 - 24*B*C*a^6*b^5*c^5*d^8 + 40*B*C*a^6*b^5*c^7*d^6 + 36*B*C*a^6*b^5*c^9*d^4 + 28*B*C*a^7*b^4*c^2*d^11 + 100*B*C*a^7*b^4*c^4*d^9 + 132*B*C*a^7*b^4*c^6*d^7 - 24*B*C*a^7*b^4*c^8*d^5 - 10*B*C*a^8*b^3*c^3*d^10 - 102*B*C*a^8*b^3*c^5*d^8 + 6*B*C*a^8*b^3*c^7*d^6 - 20*B*C*a^9*b^2*c^2*d^11 + 30*B*C*a^9*b^2*c^4*d^9))/(a^10*d^14 + b^10*c^14 + 2*a^2*b^8*c^14 + a^4*b^6*c^14 + a^6*b^4*d^14 + 2*a^8*b^2*d^14 + 4*a^10*c^2*d^12 + 6*a^10*c^4*d^10 + 4*a^10*c^6*d^8 + a^10*c^8*d^6 + b^10*c^6*d^8 + 4*b^10*c^8*d^6 + 6*b^10*c^10*d^4 + 4*b^10*c^12*d^2 - 6*a*b^9*c^5*d^9 - 24*a*b^9*c^7*d^7 - 36*a*b^9*c^9*d^5 - 24*a*b^9*c^11*d^3 - 12*a^3*b^7*c^13*d - 6*a^5*b^5*c*d^13 - 6*a^5*b^5*c^13*d - 12*a^7*b^3*c*d^13 - 24*a^9*b*c^3*d^11 - 36*a^9*b*c^5*d^9 - 24*a^9*b*c^7*d^7 - 6*a^9*b*c^9*d^5 + 15*a^2*b^8*c^4*d^10 + 62*a^2*b^8*c^6*d^8 + 98*a^2*b^8*c^8*d^6 + 72*a^2*b^8*c^10*d^4 + 23*a^2*b^8*c^12*d^2 - 20*a^3*b^7*c^3*d^11 - 92*a^3*b^7*c^5*d^9 - 168*a^3*b^7*c^7*d^7 - 152*a^3*b^7*c^9*d^5 - 68*a^3*b^7*c^11*d^3 + 15*a^4*b^6*c^2*d^12 + 90*a^4*b^6*c^4*d^10 + 211*a^4*b^6*c^6*d^8 + 244*a^4*b^6*c^8*d^6 + 141*a^4*b^6*c^10*d^4 + 34*a^4*b^6*c^12*d^2 - 64*a^5*b^5*c^3*d^11 - 202*a^5*b^5*c^5*d^9 - 288*a^5*b^5*c^7*d^7 - 202*a^5*b^5*c^9*d^5 - 64*a^5*b^5*c^11*d^3 + 34*a^6*b^4*c^2*d^12 + 141*a^6*b^4*c^4*d^10 + 244*a^6*b^4*c^6*d^8 + 211*a^6*b^4*c^8*d^6 + 90*a^6*b^4*c^10*d^4 + 15*a^6*b^4*c^12*d^2 - 68*a^7*b^3*c^3*d^11 - 152*a^7*b^3*c^5*d^9 - 168*a^7*b^3*c^7*d^7 - 92*a^7*b^3*c^9*d^5 - 20*a^7*b^3*c^11*d^3 + 23*a^8 *b^2*c^2*d^12 + 72*a^8*b^2*c^4*d^10 + 98*a^8*b^2*c^6*d^8 + 62*a^8*b^2*c^8*d^6 + 15*a^8*b^2*c^10*d^4 - 6*a*b^9*c^13*d - 6*a^9*b*c*d^13)) + (tan(e + f*x)*(4*B^3*a^5*b^4*d^10 - 12*A^3*a^2*b^7*d^10 - A^3*a^4*b^5*d^10 - 9*A^3*b^9*d^10 - 27*A^3*b^9*c^2*d^8 - 24*A^3*b^9*c^4*d^6 + 10*A^3*b^9*c^6*d^4 + C^3*a^4*b^5*d^10 + B^3*b^9*c^3*d^7 + B^3*b^9*c^5*d^5 - C^3*b^9*c^6*d^4 + 3*C^3*b^9*c^8*d^2 + 9*A^2*C*b^9*d^10 - 58*A^3*a^2*b^7*c^2*d^8 - 46*A^3*a^2*b^7*c^4*d^6 + 52*A^3*a^3*b^6*c^3*d^7 - 17*A^3*a^4*b^5*c^2*d^8 + 16*B^3*a^2*b^7*c^3*d^7 - 26*B^3*a^2*b^7*c^5*d^5 - 6*B^3*a^2*b^7*c^7*d^3 - 8*B^3*a^3*b^6*c^2*d^8 + 20*B^3*a^3*b^6*c^4*d^6 + 28*B^3*a^3*b^6*c^6*d^4 + 17*B^3*a^4*b^5*c^3*d^7 - 17*B^3*a^4*b^5*c^5*d^5 - 8*B^3*a^5*b^4*c^2*d^8 + 4*B^3*a^5*b^4*c^4*d^6 + 4*C^3*a^2*b^7*c^2*d^8 - 2*C^3*a^2*b^7*c^4*d^6 + 6*C^3*a^2*b^7*c^6*d^4 + 20*C^3*a^3*b^6*c^3*d^7 - 10*C^3*a^4*b^5*c^2*d^8 - 6*C^3*a^4*b^5*c^4*d^6 + 9*C^3*a^4*b^5*c^6*d^4 + 36*C^3*a^5*b^4*c^3*d^7 - 12*C^3*a^6*b^3*c^2*d^8 + 12*A^2*B*a*b^8*d^10 + 15*A^2*B*b^9*c*d^9 + 12*A^3*a*b^8*c*d^9 - 4*A*B^2*a^2*b^7*d^10 - 14*A*B^2*a^4*b^5*d^10 + 20*A^2*B*a^3*b^6*d^10 + 6*A*C^2*a^2*b^7*d^10 + 6*A*C^2*a^4*b^5*d^10 + 6*A^2*C*a^2*b^7*d^10 - 6*A^2*C*a^4*b^5*d^10 - 7*A*B^2*b^9*c^2*d^8 - 15*A*B^2*b^9*c^4*d^6 - 24*A*B^2*b^9*c^6*d^4 - 4*B*C^2*a^3*b^6*d^10 - 6*B*C^2*a^5*b^4*d^10 + 45*A^2*B*b^9*c^3*d^7 + 56*A^2*B*b^9*c^5*d^5 - 6*A^2*B*b^9*c^7*d^3 + 4*B^2*C*a^2*b^7*d^10 + 8*B^2*C*a^4*b^5*d^10 - 3*B^2*C*a^6*b^3*d^10 + 3*A*C^2*b^9*c^4*d^6 + 21*A*C^2*b^9*c^6*d^4 - 6*A*C^2*b^9*c^8*d^2 + 27*A^2*C*b^9*c^2*d^8 + 21*A^2*C*b^9*c^4*d^6 - 30*A^2*C*b^9*c^6*d^4 + 3*A^2*C*b^9*c^8*d^2 - B*C^2*b^9*c^5*d^5 - 9*B*C^2*b^9*c^7*d^3 + B^2*C*b^9*c^2*d^8 + 3*B^2*C*b^9*c^4*d^6 + 6*B^2*C*b^9*c^6*d^4 + 36*A^3*a*b^8*c^3*d^7 - 8*A^3*a*b^8*c^5*d^5 + 20*A^3*a^3*b^6*c*d^9 + 4*B^3*a*b^8*c^2*d^8 + 12*B^3*a*b^8*c^4*d^6 + 24*B^3*a*b^8*c^6*d^4 + 4*B^3*a^2*b^7*c*d^9 + 2*B^3*a^4*b^5*c*d^9 + 8*C^3*a*b^8*c^5*d^5 + 4*C^3*a^3*b^6*c*d^9 + 12*C^3*a^5*b^4*c*d^9 - 12*A*B*C*a*b^8*d^10 - 6*A*B*C*b^9*c*d^9 + 8*A*B^2*a^2*b^7*c^2*d^8 + 54*A*B^2*a^2*b^7*c^4*d^6 - 22*A*B^2*a^2*b^7*c^6*d^4 - 92*A*B^2*a^3*b^6*c^3*d^7 - 56*A*B^2*a^3*b^6*c^5*d^5 - 7*A*B^2*a^4*b^5*c^2*d^8 + 55*A*B^2*a^4*b^5*c^4*d^6 - 16*A*B^2*a^5*b^4*c^3*d^7 + 46*A^2*B*a^2*b^7*c^3*d^7 + 82*A^2*B*a^2*b^7*c^5*d^5 + 68*A^2*B*a^3*b^6*c^2*d^8 - 16*A^2*B*a^3*b^6*c^4*d^6 - 33*A^2*B*a^4*b^5*c^3*d^7 + 16*A^2*B*a^5*b^4*c^2*d^8 - 12*A*C^2*a^2*b^7*c^2*d^8 + 12*A*C^2*a^2*b^7*c^4*d^6 + 6*A*C^2*a^2*b^7*c^6*d^4 + 12*A*C^2*a^3*b^6*c^3*d^7 + 30*A*C^2*a^4*b^5*c^2*d^8 + 39*A*C^2*a^4*b^5*c^4*d^6 - 9*A*C^2*a^4*b^5*c^6*d^4 - 72*A*C^2*a^5*b^4*c^3*d^7 + 24*A*C^2*a^6*b^3*c^2*d^8 + 66*A^2*C*a^2*b^7*c^2*d^8 + 36*A^2*C*a^2*b^7*c^4*d^6 - 12*A^2*C*a^2*b^7*c^6*d^4 - 84*A^2*C*a^3*b^6*c^3*d^7 - 3*A^2*C*a^4*b^5*c^2*d^8 - 33*A^2*C*a^4*b^5*c^4*d^6 + 36*A^2*C*a^5*b^4*c^3*d^7 - 12*A^2*C*a^6*b^3*c^2*d^8 - 20*B*C^2*a^2*b^7*c^3*d^7 + 4*B*C^2*a^2*b^7*c^5*d^5 + 6*B*C^2*a^2*b^7*c^7*d^3 + 8*B*C^2*a^3*b^6*c^2*d^8 + 32*B*C^2*a^3*b^6*c^4*d^6 - 12*B*C^2*a^3*b^6*c^6*d^4 - 66*B*C^2*a^4*b^5*c^3*d^7 - 21*B*C^2*a^4*b^5*c^5*d^5 + 9*B*C^2*a^4*b^5*c^7*d^3 + 4*B*C^2*a^5*b^4*c^2*d^8 + 42*B*C^2*a^5*b^4*c^4*d^6 - 12*B*C^2*a^6*b^3*c^3*d^7 - 2*B^2*C*a^2*b^7*c^2*d^8 - 63*B^2*C*a^2*b^7*c^4*d^6 - 2*B^2*C*a^2*b^7*c^6*d^4 + 3*B^2*C*a^2*b^7*c^8*d^2 + 32*B^2*C*a^3*b^6*c^3*d^7 + 44*B^2*C*a^3*b^6*c^5*d^5 - 12*B^2*C*a^3*b^6*c^7*d^3 + 13*B^2*C*a^4*b^5*c^2*d^8 - 73*B^2*

C*a^4*b^5*c^4*d^6 - 18*B^2*C*a^4*b^5*c^6*d^4 + 4*B^2*C*a^5*b^4*c^3*d^7 + 12
*B^2*C*a^5*b^4*c^5*d^5 + 6*B^2*C*a^6*b^3*c^2*d^8 - 3*B^2*C*a^6*b^3*c^4*d^6
- 16*A*B*C*a^3*b^6*d^10 + 6*A*B*C*a^5*b^4*d^10 - 18*A*B*C*b^9*c^3*d^7 - 28*
A*B*C*b^9*c^5*d^5 + 24*A*B*C*b^9*c^7*d^3 - 16*A*B^2*a*b^8*c*d^9 + 12*A*C^2*
a*b^8*c*d^9 - 24*A^2*C*a*b^8*c*d^9 + 4*B^2*C*a*b^8*c*d^9 - 56*A*B^2*a*b^8*c
^3*d^7 - 28*A*B^2*a*b^8*c^5*d^5 + 12*A*B^2*a*b^8*c^7*d^3 - 4*A*B^2*a^3*b^6*
c*d^9 + 16*A*B^2*a^5*b^4*c*d^9 + 20*A^2*B*a*b^8*c^2*d^8 - 56*A^2*B*a*b^8*c^
4*d^6 - 16*A^2*B*a*b^8*c^6*d^4 - 4*A^2*B*a^2*b^7*c*d^9 - 33*A^2*B*a^4*b^5*c
*d^9 + 36*A*C^2*a*b^8*c^3*d^7 - 24*A*C^2*a*b^8*c^5*d^5 + 12*A*C^2*a^3*b^6*c
*d^9 - 24*A*C^2*a^5*b^4*c*d^9 - 72*A^2*C*a*b^8*c^3*d^7 + 24*A^2*C*a*b^8*c^5
*d^5 - 36*A^2*C*a^3*b^6*c*d^9 + 12*A^2*C*a^5*b^4*c*d^9 - 4*B*C^2*a*b^8*c^2*
d^8 - 14*B*C^2*a*b^8*c^4*d^6 - 4*B*C^2*a*b^8*c^6*d^4 + 6*B*C^2*a*b^8*c^8*d^
2 - 10*B*C^2*a^2*b^7*c*d^9 - 12*B*C^2*a^4*b^5*c*d^9 + 12*B*C^2*a^6*b^3*c*d^
9 + 8*B^2*C*a*b^8*c^3*d^7 + 4*B^2*C*a*b^8*c^5*d^5 - 24*B^2*C*a*b^8*c^7*d^3
- 8*B^2*C*a^3*b^6*c*d^9 - 16*B^2*C*a^5*b^4*c*d^9 + 28*A*B*C*a^2*b^7*c^3*d^7
 - 32*A*B*C*a^2*b^7*c^5*d^5 + 12*A*B*C*a^2*b^7*c^7*d^3 - 76*A*B*C*a^3*b^6*c
^2*d^8 - 16*A*B*C*a^3*b^6*c^4*d^6 + 12*A*B*C*a^3*b^6*c^6*d^4 + 126*A*B*C*a^
4*b^5*c^3*d^7 + 48*A*B*C*a^4*b^5*c^5*d^5 - 20*A*B*C*a^5*b^4*c^2*d^8 - 42*A*
B*C*a^5*b^4*c^4*d^6 + 12*A*B*C*a^6*b^3*c^3*d^7 - 16*A*B*C*a*b^8*c^2*d^8 + 7
0*A*B*C*a*b^8*c^4*d^6 + 20*A*B*C*a*b^8*c^6*d^4 - 6*A*B*C*a*b^8*c^8*d^2 + 32
*A*B*C*a^2*b^7*c*d^9 + 54*A*B*C*a^4*b^5*c*d^9 - 12*A*B*C*a^6*b^3*c*d^9))/(a
^10*d^14 + b^10*c^14 + 2*a^2*b^8*c^14 + a^4*b^6*c^14 + a^6*b^4*d^14 + 2*a^8
*b^2*d^14 + 4*a^10*c^2*d^12 + 6*a^10*c^4*d^10 + 4*a^10*c^6*d^8 + a^10*c^8*d
^6 + b^10*c^6*d^8 + 4*b^10*c^8*d^6 + 6*b^10*c^10*d^4 + 4*b^10*c^12*d^2 - 6*
a*b^9*c^5*d^9 - 24*a*b^9*c^7*d^7 - 36*a*b^9*c^9*d^5 - 24*a*b^9*c^11*d^3 - 1
2*a^3*b^7*c^13*d - 6*a^5*b^5*c*d^13 - 6*a^5*b^5*c^13*d - 12*a^7*b^3*c*d^13
- 24*a^9*b*c^3*d^11 - 36*a^9*b*c^5*d^9 - 24*a^9*b*c^7*d^7 - 6*a^9*b*c^9*d^5
 + 15*a^2*b^8*c^4*d^10 + 62*a^2*b^8*c^6*d^8 + 98*a^2*b^8*c^8*d^6 + 72*a^2*b
^8*c^10*d^4 + 23*a^2*b^8*c^12*d^2 - 20*a^3*b^7*c^3*d^11 - 92*a^3*b^7*c^5*d^
9 - 168*a^3*b^7*c^7*d^7 - 152*a^3*b^7*c^9*d^5 - 68*a^3*b^7*c^11*d^3 + 15*a^
4*b^6*c^2*d^12 + 90*a^4*b^6*c^4*d^10 + 211*a^4*b^6*c^6*d^8 + 244*a^4*b^6*c^
8*d^6 + 141*a^4*b^6*c^10*d^4 + 34*a^4*b^6*c^12*d^2 - 64*a^5*b^5*c^3*d^11 -
202*a^5*b^5*c^5*d^9 - 288*a^5*b^5*c^7*d^7 - 202*a^5*b^5*c^9*d^5 - 64*a^5*b^
5*c^11*d^3 + 34*a^6*b^4*c^2*d^12 + 141*a^6*b^4*c^4*d^10 + 244*a^6*b^4*c^6*d
^8 + 211*a^6*b^4*c^8*d^6 + 90*a^6*b^4*c^10*d^4 + 15*a^6*b^4*c^12*d^2 - 68*a
^7*b^3*c^3*d^11 - 152*a^7*b^3*c^5*d^9 - 168*a^7*b^3*c^7*d^7 - 92*a^7*b^3*c^
9*d^5 - 20*a^7*b^3*c^11*d^3 + 23*a^8*b^2*c^2*d^12 + 72*a^8*b^2*c^4*d^10 + 9
8*a^8*b^2*c^6*d^8 + 62*a^8*b^2*c^8*d^6 + 15*a^8*b^2*c^10*d^4 - 6*a*b^9*c^13
*d - 6*a^9*b*c*d^13))*root(640*a^15*b*c^7*d^13*f^4 + 640*a*b^15*c^13*d^7*f^
4 + 480*a^15*b*c^9*d^11*f^4 + 480*a^15*b*c^5*d^15*f^4 + 480*a*b^15*c^15*d^5
*f^4 + 480*a*b^15*c^11*d^9*f^4 + 192*a^15*b*c^11*d^9*f^4 + 192*a^15*b*c^3*d
^17*f^4 + 192*a^11*b^5*c*d^19*f^4 + 192*a^5*b^11*c^19*d*f^4 + 192*a*b^15*c^
17*d^3*f^4 + 192*a*b^15*c^9*d^11*f^4 + 128*a^13*b^3*c*d^19*f^4 + 128*a^9*b^
7*c*d^19*f^4 + 128*a^7*b^9*c^19*d*f^4 + 128*a^3*b^13*c^19*d*f^4 + 32*a^15*b
*c^13*d^7*f^4 + 32*a^9*b^7*c^19*d*f^4 + 32*a^7*b^9*c*d^19*f^4 + 32*a*b^15*c

^7*d^13*f^4 + 32*a^15*b*c*d^19*f^4 + 32*a*b^15*c^19*d*f^4 - 47088*a^8*b^8*c
^10*d^10*f^4 + 42432*a^9*b^7*c^9*d^11*f^4 + 42432*a^7*b^9*c^11*d^9*f^4 + 39
328*a^9*b^7*c^11*d^9*f^4 + 39328*a^7*b^9*c^9*d^11*f^4 - 36912*a^8*b^8*c^12*
d^8*f^4 - 36912*a^8*b^8*c^8*d^12*f^4 - 34256*a^10*b^6*c^10*d^10*f^4 - 34256
*a^6*b^10*c^10*d^10*f^4 - 31152*a^10*b^6*c^8*d^12*f^4 - 31152*a^6*b^10*c^12
*d^8*f^4 + 28128*a^9*b^7*c^7*d^13*f^4 + 28128*a^7*b^9*c^13*d^7*f^4 + 24160*
a^11*b^5*c^9*d^11*f^4 + 24160*a^5*b^11*c^11*d^9*f^4 - 23088*a^10*b^6*c^12*d
^8*f^4 - 23088*a^6*b^10*c^8*d^12*f^4 + 22272*a^9*b^7*c^13*d^7*f^4 + 22272*a
^7*b^9*c^7*d^13*f^4 + 19072*a^11*b^5*c^11*d^9*f^4 + 19072*a^5*b^11*c^9*d^11
*f^4 + 18624*a^11*b^5*c^7*d^13*f^4 + 18624*a^5*b^11*c^13*d^7*f^4 - 17328*a^
8*b^8*c^14*d^6*f^4 - 17328*a^8*b^8*c^6*d^14*f^4 - 17232*a^10*b^6*c^6*d^14*f
^4 - 17232*a^6*b^10*c^14*d^6*f^4 - 13520*a^12*b^4*c^8*d^12*f^4 - 13520*a^4*
b^12*c^12*d^8*f^4 - 12464*a^12*b^4*c^10*d^10*f^4 - 12464*a^4*b^12*c^10*d^10
*f^4 + 10880*a^9*b^7*c^5*d^15*f^4 + 10880*a^7*b^9*c^15*d^5*f^4 - 9072*a^10*
b^6*c^14*d^6*f^4 - 9072*a^6*b^10*c^6*d^14*f^4 + 8928*a^11*b^5*c^13*d^7*f^4
+ 8928*a^5*b^11*c^7*d^13*f^4 - 8880*a^12*b^4*c^6*d^14*f^4 - 8880*a^4*b^12*c
^14*d^6*f^4 + 8480*a^11*b^5*c^5*d^15*f^4 + 8480*a^5*b^11*c^15*d^5*f^4 + 720
0*a^9*b^7*c^15*d^5*f^4 + 7200*a^7*b^9*c^5*d^15*f^4 - 6912*a^12*b^4*c^12*d^8
*f^4 - 6912*a^4*b^12*c^8*d^12*f^4 + 6400*a^13*b^3*c^9*d^11*f^4 + 6400*a^3*b
^13*c^11*d^9*f^4 + 5920*a^13*b^3*c^7*d^13*f^4 + 5920*a^3*b^13*c^13*d^7*f^4
- 5392*a^10*b^6*c^4*d^16*f^4 - 5392*a^6*b^10*c^16*d^4*f^4 - 4428*a^8*b^8*c^
16*d^4*f^4 - 4428*a^8*b^8*c^4*d^16*f^4 + 4128*a^13*b^3*c^11*d^9*f^4 + 4128*
a^3*b^13*c^9*d^11*f^4 - 3328*a^12*b^4*c^4*d^16*f^4 - 3328*a^4*b^12*c^16*d^4
*f^4 + 3264*a^13*b^3*c^5*d^15*f^4 + 3264*a^3*b^13*c^15*d^5*f^4 - 2480*a^14*
b^2*c^8*d^12*f^4 - 2480*a^2*b^14*c^12*d^8*f^4 + 2240*a^11*b^5*c^15*d^5*f^4
+ 2240*a^5*b^11*c^5*d^15*f^4 - 2128*a^12*b^4*c^14*d^6*f^4 - 2128*a^4*b^12*c
^6*d^14*f^4 + 2112*a^9*b^7*c^3*d^17*f^4 + 2112*a^7*b^9*c^17*d^3*f^4 + 2048*
a^11*b^5*c^3*d^17*f^4 + 2048*a^5*b^11*c^17*d^3*f^4 - 2000*a^14*b^2*c^6*d^14
*f^4 - 2000*a^2*b^14*c^14*d^6*f^4 - 1792*a^10*b^6*c^16*d^4*f^4 - 1792*a^6*b
^10*c^4*d^16*f^4 - 1776*a^14*b^2*c^10*d^10*f^4 - 1776*a^2*b^14*c^10*d^10*f^
4 + 1472*a^13*b^3*c^13*d^7*f^4 + 1472*a^3*b^13*c^7*d^13*f^4 + 1088*a^9*b^7*
c^17*d^3*f^4 + 1088*a^7*b^9*c^3*d^17*f^4 + 992*a^13*b^3*c^3*d^17*f^4 + 992*
a^3*b^13*c^17*d^3*f^4 - 912*a^14*b^2*c^4*d^16*f^4 - 912*a^2*b^14*c^16*d^4*f
^4 - 768*a^10*b^6*c^2*d^18*f^4 - 768*a^6*b^10*c^18*d^2*f^4 - 688*a^14*b^2*c
^12*d^8*f^4 - 688*a^2*b^14*c^8*d^12*f^4 - 592*a^12*b^4*c^2*d^18*f^4 - 592*a
^4*b^12*c^18*d^2*f^4 - 472*a^8*b^8*c^18*d^2*f^4 - 472*a^8*b^8*c^2*d^18*f^4
- 280*a^12*b^4*c^16*d^4*f^4 - 280*a^4*b^12*c^4*d^16*f^4 + 224*a^13*b^3*c^15
*d^5*f^4 + 224*a^11*b^5*c^17*d^3*f^4 + 224*a^5*b^11*c^3*d^17*f^4 + 224*a^3*
b^13*c^5*d^15*f^4 - 208*a^14*b^2*c^2*d^18*f^4 - 208*a^2*b^14*c^18*d^2*f^4 -
112*a^14*b^2*c^14*d^6*f^4 - 112*a^10*b^6*c^18*d^2*f^4 - 112*a^6*b^10*c^2*d
^18*f^4 - 112*a^2*b^14*c^6*d^14*f^4 - 80*b^16*c^14*d^6*f^4 - 60*b^16*c^16*d
^4*f^4 - 60*b^16*c^12*d^8*f^4 - 24*b^16*c^18*d^2*f^4 - 24*b^16*c^10*d^10*f^
4 - 4*b^16*c^8*d^12*f^4 - 80*a^16*c^6*d^14*f^4 - 60*a^16*c^8*d^12*f^4 - 60*
a^16*c^4*d^16*f^4 - 24*a^16*c^10*d^10*f^4 - 24*a^16*c^2*d^18*f^4 - 4*a^16*c
^12*d^8*f^4 - 24*a^12*b^4*d^20*f^4 - 16*a^14*b^2*d^20*f^4 - 16*a^10*b^6*d^2

0*f^4 - 4*a^8*b^8*d^20*f^4 - 24*a^4*b^12*c^20*f^4 - 16*a^6*b^10*c^20*f^4 -
16*a^2*b^14*c^20*f^4 - 4*a^8*b^8*c^20*f^4 - 4*b^16*c^20*f^4 - 4*a^16*d^20*f
^4 + 56*A*C*a*b^11*c^13*d*f^2 - 48*A*C*a^11*b*c*d^13*f^2 + 48*A*C*a*b^11*c*
d^13*f^2 + 5904*B*C*a^6*b^6*c^7*d^7*f^2 - 5016*B*C*a^5*b^7*c^8*d^6*f^2 - 46
08*B*C*a^7*b^5*c^6*d^8*f^2 - 4512*B*C*a^5*b^7*c^6*d^8*f^2 - 4384*B*C*a^7*b^
5*c^8*d^6*f^2 + 3056*B*C*a^8*b^4*c^7*d^7*f^2 + 2256*B*C*a^4*b^8*c^7*d^7*f^2
 - 1824*B*C*a^3*b^9*c^8*d^6*f^2 + 1632*B*C*a^9*b^3*c^4*d^10*f^2 - 1400*B*C*
a^8*b^4*c^3*d^11*f^2 - 1320*B*C*a^4*b^8*c^11*d^3*f^2 - 1248*B*C*a^3*b^9*c^6
*d^8*f^2 + 1152*B*C*a^3*b^9*c^10*d^4*f^2 - 1072*B*C*a^9*b^3*c^6*d^8*f^2 + 1
068*B*C*a^6*b^6*c^9*d^5*f^2 - 1004*B*C*a^4*b^8*c^5*d^9*f^2 - 968*B*C*a^6*b^
6*c^3*d^11*f^2 - 864*B*C*a^8*b^4*c^5*d^9*f^2 - 828*B*C*a^4*b^8*c^9*d^5*f^2
- 792*B*C*a^4*b^8*c^3*d^11*f^2 - 792*B*C*a^2*b^10*c^11*d^3*f^2 - 776*B*C*a^
9*b^3*c^8*d^6*f^2 + 688*B*C*a^7*b^5*c^4*d^10*f^2 - 672*B*C*a^10*b^2*c^3*d^1
1*f^2 - 592*B*C*a^2*b^10*c^9*d^5*f^2 + 544*B*C*a^10*b^2*c^7*d^7*f^2 - 492*B
*C*a^2*b^10*c^5*d^9*f^2 + 480*B*C*a^5*b^7*c^10*d^4*f^2 - 392*B*C*a^10*b^2*c
^5*d^9*f^2 + 332*B*C*a^8*b^4*c^9*d^5*f^2 - 328*B*C*a^6*b^6*c^11*d^3*f^2 + 3
20*B*C*a^9*b^3*c^2*d^12*f^2 + 272*B*C*a^3*b^9*c^12*d^2*f^2 - 248*B*C*a^5*b^
7*c^4*d^10*f^2 - 248*B*C*a^2*b^10*c^3*d^11*f^2 - 208*B*C*a^7*b^5*c^10*d^4*f
^2 - 192*B*C*a^5*b^7*c^2*d^12*f^2 + 144*B*C*a^2*b^10*c^7*d^7*f^2 - 96*B*C*a
^3*b^9*c^4*d^10*f^2 + 88*B*C*a^5*b^7*c^12*d^2*f^2 - 72*B*C*a^8*b^4*c^11*d^3
*f^2 + 48*B*C*a^9*b^3*c^10*d^4*f^2 - 48*B*C*a^7*b^5*c^12*d^2*f^2 - 48*B*C*a
^7*b^5*c^2*d^12*f^2 - 48*B*C*a^3*b^9*c^2*d^12*f^2 - 12*B*C*a^10*b^2*c^9*d^5
*f^2 + 4*B*C*a^6*b^6*c^5*d^9*f^2 + 5824*A*C*a^7*b^5*c^5*d^9*f^2 - 4378*A*C*
a^8*b^4*c^6*d^8*f^2 + 4296*A*C*a^5*b^7*c^5*d^9*f^2 - 3912*A*C*a^6*b^6*c^6*d
^8*f^2 - 3672*A*C*a^5*b^7*c^9*d^5*f^2 + 3594*A*C*a^4*b^8*c^8*d^6*f^2 + 3236
*A*C*a^6*b^6*c^8*d^6*f^2 + 2816*A*C*a^9*b^3*c^5*d^9*f^2 + 2624*A*C*a^3*b^9*
c^5*d^9*f^2 + 2432*A*C*a^7*b^5*c^7*d^7*f^2 - 2366*A*C*a^8*b^4*c^4*d^10*f^2
+ 2298*A*C*a^4*b^8*c^10*d^4*f^2 + 1872*A*C*a^3*b^9*c^7*d^7*f^2 + 1848*A*C*a
^6*b^6*c^10*d^4*f^2 - 1644*A*C*a^6*b^6*c^4*d^10*f^2 - 1488*A*C*a^7*b^5*c^9*
d^5*f^2 - 1408*A*C*a^3*b^9*c^9*d^5*f^2 - 1308*A*C*a^4*b^8*c^6*d^8*f^2 + 124
8*A*C*a^5*b^7*c^7*d^7*f^2 - 1012*A*C*a^10*b^2*c^6*d^8*f^2 + 1008*A*C*a^7*b^
5*c^3*d^11*f^2 + 992*A*C*a^5*b^7*c^3*d^11*f^2 + 928*A*C*a^3*b^9*c^3*d^11*f^
2 + 848*A*C*a^9*b^3*c^7*d^7*f^2 + 636*A*C*a^2*b^10*c^8*d^6*f^2 - 628*A*C*a^
10*b^2*c^4*d^10*f^2 - 600*A*C*a^2*b^10*c^6*d^8*f^2 - 576*A*C*a^5*b^7*c^11*d
^3*f^2 + 572*A*C*a^2*b^10*c^10*d^4*f^2 + 464*A*C*a^8*b^4*c^8*d^6*f^2 + 304*
A*C*a^6*b^6*c^2*d^12*f^2 - 304*A*C*a^4*b^8*c^4*d^10*f^2 + 296*A*C*a^4*b^8*c
^2*d^12*f^2 + 260*A*C*a^8*b^4*c^10*d^4*f^2 - 232*A*C*a^9*b^3*c^9*d^5*f^2 -
232*A*C*a^2*b^10*c^12*d^2*f^2 + 228*A*C*a^10*b^2*c^2*d^12*f^2 - 188*A*C*a^2
*b^10*c^4*d^10*f^2 + 144*A*C*a^3*b^9*c^11*d^3*f^2 + 116*A*C*a^6*b^6*c^12*d^
2*f^2 + 112*A*C*a^9*b^3*c^3*d^11*f^2 - 112*A*C*a^7*b^5*c^11*d^3*f^2 + 92*A*
C*a^10*b^2*c^8*d^6*f^2 + 74*A*C*a^4*b^8*c^12*d^2*f^2 + 62*A*C*a^8*b^4*c^2*d
^12*f^2 + 40*A*C*a^2*b^10*c^2*d^12*f^2 - 7008*A*B*a^6*b^6*c^7*d^7*f^2 - 403
2*A*B*a^4*b^8*c^7*d^7*f^2 + 3952*A*B*a^7*b^5*c^8*d^6*f^2 + 3648*A*B*a^5*b^7
*c^8*d^6*f^2 - 3392*A*B*a^8*b^4*c^7*d^7*f^2 + 3264*A*B*a^7*b^5*c^6*d^8*f^2
- 2992*A*B*a^5*b^7*c^4*d^10*f^2 - 2368*A*B*a^7*b^5*c^4*d^10*f^2 - 2304*A*B*

a^3*b^9*c^4*d^10*f^2 - 1968*A*B*a^6*b^6*c^9*d^5*f^2 - 1872*A*B*a^9*b^3*c^4*
d^10*f^2 - 1728*A*B*a^2*b^10*c^7*d^7*f^2 + 1712*A*B*a^8*b^4*c^3*d^11*f^2 +
1536*A*B*a^5*b^7*c^6*d^8*f^2 - 1536*A*B*a^3*b^9*c^10*d^4*f^2 - 1392*A*B*a^5
*b^7*c^2*d^12*f^2 + 1328*A*B*a^6*b^6*c^3*d^11*f^2 - 1104*A*B*a^3*b^9*c^2*d^
12*f^2 - 1056*A*B*a^3*b^9*c^6*d^8*f^2 + 976*A*B*a^9*b^3*c^6*d^8*f^2 + 960*A
*B*a^4*b^8*c^11*d^3*f^2 + 936*A*B*a^8*b^4*c^5*d^9*f^2 - 912*A*B*a^5*b^7*c^1
0*d^4*f^2 + 848*A*B*a^9*b^3*c^8*d^6*f^2 - 816*A*B*a^7*b^5*c^2*d^12*f^2 + 81
6*A*B*a^4*b^8*c^3*d^11*f^2 + 768*A*B*a^10*b^2*c^3*d^11*f^2 + 672*A*B*a^3*b^
9*c^8*d^6*f^2 - 632*A*B*a^8*b^4*c^9*d^5*f^2 - 608*A*B*a^2*b^10*c^9*d^5*f^2
- 552*A*B*a^4*b^8*c^9*d^5*f^2 - 544*A*B*a^10*b^2*c^7*d^7*f^2 - 480*A*B*a^2*
b^10*c^5*d^9*f^2 + 464*A*B*a^10*b^2*c^5*d^9*f^2 - 464*A*B*a^9*b^3*c^2*d^12*
f^2 + 432*A*B*a^2*b^10*c^11*d^3*f^2 - 368*A*B*a^3*b^9*c^12*d^2*f^2 - 256*A*
B*a^6*b^6*c^5*d^9*f^2 - 208*A*B*a^5*b^7*c^12*d^2*f^2 + 176*A*B*a^4*b^8*c^5*
d^9*f^2 + 112*A*B*a^7*b^5*c^10*d^4*f^2 + 112*A*B*a^6*b^6*c^11*d^3*f^2 - 16*
A*B*a^2*b^10*c^3*d^11*f^2 - 576*B*C*a*b^11*c^8*d^6*f^2 + 400*B*C*a^11*b*c^4
*d^10*f^2 - 288*B*C*a*b^11*c^6*d^8*f^2 - 176*B*C*a^11*b*c^6*d^8*f^2 + 128*B
*C*a*b^11*c^10*d^4*f^2 - 108*B*C*a^4*b^8*c*d^13*f^2 - 104*B*C*a*b^11*c^4*d^
10*f^2 - 92*B*C*a^4*b^8*c^13*d*f^2 - 60*B*C*a^8*b^4*c*d^13*f^2 - 60*B*C*a^6
*b^6*c*d^13*f^2 + 48*B*C*a^11*b*c^2*d^12*f^2 - 40*B*C*a^2*b^10*c*d^13*f^2 -
28*B*C*a^2*b^10*c^13*d*f^2 - 24*B*C*a*b^11*c^12*d^2*f^2 + 20*B*C*a^10*b^2*
c*d^13*f^2 - 16*B*C*a*b^11*c^2*d^12*f^2 + 12*B*C*a^6*b^6*c^13*d*f^2 + 912*A
*C*a*b^11*c^7*d^7*f^2 + 808*A*C*a*b^11*c^5*d^9*f^2 + 432*A*C*a^11*b*c^5*d^9
*f^2 + 336*A*C*a*b^11*c^3*d^11*f^2 + 224*A*C*a*b^11*c^11*d^3*f^2 - 112*A*C*
a^11*b*c^3*d^11*f^2 + 112*A*C*a^3*b^9*c*d^13*f^2 - 88*A*C*a^9*b^3*c*d^13*f^
2 + 80*A*C*a^3*b^9*c^13*d*f^2 + 56*A*C*a^5*b^7*c*d^13*f^2 + 48*A*C*a*b^11*c
^9*d^5*f^2 - 40*A*C*a^5*b^7*c^13*d*f^2 - 16*A*C*a^11*b*c^7*d^7*f^2 + 16*A*C
*a^7*b^5*c*d^13*f^2 - 496*A*B*a*b^11*c^4*d^10*f^2 - 400*A*B*a^11*b*c^4*d^10
*f^2 + 288*A*B*a*b^11*c^8*d^6*f^2 - 288*A*B*a*b^11*c^6*d^8*f^2 - 272*A*B*a*
b^11*c^2*d^12*f^2 + 240*A*B*a^6*b^6*c*d^13*f^2 - 224*A*B*a*b^11*c^10*d^4*f^
2 + 192*A*B*a^8*b^4*c*d^13*f^2 + 192*A*B*a^4*b^8*c*d^13*f^2 + 176*A*B*a^11*
b*c^6*d^8*f^2 + 104*A*B*a^4*b^8*c^13*d*f^2 - 48*A*B*a^11*b*c^2*d^12*f^2 + 1
6*A*B*a^10*b^2*c*d^13*f^2 + 16*A*B*a^2*b^10*c^13*d*f^2 + 16*A*B*a^2*b^10*c*
d^13*f^2 - 112*B*C*b^12*c^11*d^3*f^2 + 4*B*C*b^12*c^5*d^9*f^2 + 150*A*C*b^1
2*c^10*d^4*f^2 - 80*B*C*a^12*c^3*d^11*f^2 + 66*A*C*b^12*c^8*d^6*f^2 - 30*A*
C*b^12*c^12*d^2*f^2 + 24*B*C*a^12*c^5*d^9*f^2 - 12*A*C*b^12*c^4*d^10*f^2 -
576*A*B*b^12*c^7*d^7*f^2 - 432*A*B*b^12*c^9*d^5*f^2 - 400*A*B*b^12*c^5*d^9*
f^2 - 144*A*B*b^12*c^3*d^11*f^2 - 96*B*C*a^7*b^5*d^14*f^2 - 72*B*C*a^5*b^7*
d^14*f^2 - 66*A*C*a^12*c^4*d^10*f^2 + 54*A*C*a^12*c^2*d^12*f^2 - 32*A*B*b^1
2*c^11*d^3*f^2 - 24*B*C*a^9*b^3*d^14*f^2 - 16*B*C*a^3*b^9*d^14*f^2 + 2*A*C*
a^12*c^6*d^8*f^2 + 116*A*C*a^6*b^6*d^14*f^2 + 100*A*C*a^4*b^8*d^14*f^2 + 80
*A*B*a^12*c^3*d^11*f^2 + 24*A*C*a^2*b^10*d^14*f^2 - 24*A*B*a^12*c^5*d^9*f^2
+ 22*A*C*a^8*b^4*d^14*f^2 + 16*B*C*a^3*b^9*c^14*f^2 + 8*A*C*a^10*b^2*d^14*
f^2 - 192*A*B*a^5*b^7*d^14*f^2 - 176*A*B*a^3*b^9*d^14*f^2 - 48*A*B*a^7*b^5*
d^14*f^2 - 28*A*C*a^2*b^10*c^14*f^2 + 2*A*C*a^4*b^8*c^14*f^2 - 16*A*B*a^3*b
^9*c^14*f^2 + 2508*C^2*a^6*b^6*c^6*d^8*f^2 + 2376*C^2*a^5*b^7*c^9*d^5*f^2 +

2357*C^2*a^8*b^4*c^6*d^8*f^2 - 2048*C^2*a^7*b^5*c^5*d^9*f^2 + 1304*C^2*a^3*b^9*c^9*d^5*f^2 + 1303*C^2*a^8*b^4*c^4*d^10*f^2 + 1212*C^2*a^6*b^6*c^4*d^10*f^2 - 1203*C^2*a^4*b^8*c^8*d^6*f^2 - 1192*C^2*a^9*b^3*c^5*d^9*f^2 + 1062*C^2*a^4*b^8*c^6*d^8*f^2 + 984*C^2*a^7*b^5*c^9*d^5*f^2 - 952*C^2*a^6*b^6*c^8*d^6*f^2 + 768*C^2*a^5*b^7*c^7*d^7*f^2 - 681*C^2*a^4*b^8*c^10*d^4*f^2 - 672*C^2*a^5*b^7*c^5*d^9*f^2 - 480*C^2*a^6*b^6*c^10*d^4*f^2 + 458*C^2*a^10*b^2*c^6*d^8*f^2 - 448*C^2*a^7*b^5*c^7*d^7*f^2 + 422*C^2*a^4*b^8*c^4*d^10*f^2 + 372*C^2*a^2*b^10*c^6*d^8*f^2 + 360*C^2*a^5*b^7*c^11*d^3*f^2 + 312*C^2*a^3*b^9*c^7*d^7*f^2 + 278*C^2*a^10*b^2*c^4*d^10*f^2 - 232*C^2*a^9*b^3*c^7*d^7*f^2 + 194*C^2*a^2*b^10*c^12*d^2*f^2 + 176*C^2*a^9*b^3*c^9*d^5*f^2 + 152*C^2*a^5*b^7*c^3*d^11*f^2 + 124*C^2*a^2*b^10*c^4*d^10*f^2 - 120*C^2*a^7*b^5*c^3*d^11*f^2 - 114*C^2*a^10*b^2*c^2*d^12*f^2 - 102*C^2*a^2*b^10*c^8*d^6*f^2 + 101*C^2*a^4*b^8*c^12*d^2*f^2 + 100*C^2*a^6*b^6*c^2*d^12*f^2 - 88*C^2*a^3*b^9*c^5*d^9*f^2 + 77*C^2*a^8*b^4*c^2*d^12*f^2 + 72*C^2*a^3*b^9*c^11*d^3*f^2 - 64*C^2*a^10*b^2*c^8*d^6*f^2 + 64*C^2*a^3*b^9*c^3*d^11*f^2 - 58*C^2*a^2*b^10*c^10*d^4*f^2 + 56*C^2*a^7*b^5*c^11*d^3*f^2 + 56*C^2*a^6*b^6*c^12*d^2*f^2 + 40*C^2*a^9*b^3*c^3*d^11*f^2 + 36*C^2*a^8*b^4*c^12*d^2*f^2 + 32*C^2*a^4*b^8*c^2*d^12*f^2 + 26*C^2*a^8*b^4*c^10*d^4*f^2 + 16*C^2*a^2*b^10*c^2*d^12*f^2 + 2*C^2*a^8*b^4*c^8*d^6*f^2 + 2277*B^2*a^4*b^8*c^8*d^6*f^2 + 2144*B^2*a^7*b^5*c^5*d^9*f^2 - 2112*B^2*a^5*b^7*c^9*d^5*f^2 + 2028*B^2*a^6*b^6*c^8*d^6*f^2 - 1671*B^2*a^8*b^4*c^6*d^8*f^2 + 1275*B^2*a^4*b^8*c^10*d^4*f^2 + 1176*B^2*a^5*b^7*c^5*d^9*f^2 + 1096*B^2*a^9*b^3*c^5*d^9*f^2 - 1044*B^2*a^6*b^6*c^6*d^8*f^2 + 984*B^2*a^6*b^6*c^10*d^4*f^2 - 968*B^2*a^3*b^9*c^9*d^5*f^2 - 888*B^2*a^7*b^5*c^9*d^5*f^2 + 672*B^2*a^7*b^5*c^7*d^7*f^2 + 664*B^2*a^3*b^9*c^5*d^9*f^2 - 649*B^2*a^8*b^4*c^4*d^10*f^2 + 618*B^2*a^2*b^10*c^8*d^6*f^2 + 514*B^2*a^4*b^8*c^4*d^10*f^2 + 460*B^2*a^6*b^6*c^2*d^12*f^2 + 422*B^2*a^8*b^4*c^8*d^6*f^2 + 406*B^2*a^2*b^10*c^10*d^4*f^2 - 382*B^2*a^10*b^2*c^6*d^8*f^2 + 368*B^2*a^4*b^8*c^2*d^12*f^2 - 312*B^2*a^5*b^7*c^11*d^3*f^2 + 312*B^2*a^3*b^9*c^7*d^7*f^2 + 248*B^2*a^9*b^3*c^7*d^7*f^2 + 245*B^2*a^8*b^4*c^2*d^12*f^2 - 192*B^2*a^5*b^7*c^7*d^7*f^2 - 184*B^2*a^9*b^3*c^3*d^11*f^2 + 182*B^2*a^10*b^2*c^2*d^12*f^2 + 176*B^2*a^3*b^9*c^3*d^11*f^2 + 174*B^2*a^4*b^8*c^6*d^8*f^2 - 170*B^2*a^10*b^2*c^4*d^10*f^2 - 152*B^2*a^9*b^3*c^9*d^5*f^2 + 152*B^2*a^2*b^10*c^4*d^10*f^2 + 142*B^2*a^8*b^4*c^10*d^4*f^2 - 90*B^2*a^2*b^10*c^12*d^2*f^2 + 88*B^2*a^2*b^10*c^2*d^12*f^2 + 84*B^2*a^10*b^2*c^8*d^6*f^2 + 84*B^2*a^2*b^10*c^6*d^8*f^2 + 60*B^2*a^6*b^6*c^12*d^2*f^2 - 56*B^2*a^7*b^5*c^11*d^3*f^2 + 53*B^2*a^4*b^8*c^12*d^2*f^2 + 24*B^2*a^7*b^5*c^3*d^11*f^2 + 24*B^2*a^6*b^6*c^4*d^10*f^2 + 24*B^2*a^3*b^9*c^11*d^3*f^2 - 8*B^2*a^5*b^7*c^3*d^11*f^2 + 4566*A^2*a^4*b^8*c^6*d^8*f^2 + 4284*A^2*a^6*b^6*c^6*d^8*f^2 - 3776*A^2*a^7*b^5*c^5*d^9*f^2 - 3624*A^2*a^5*b^7*c^5*d^9*f^2 + 3122*A^2*a^4*b^8*c^4*d^10*f^2 + 3108*A^2*a^2*b^10*c^6*d^8*f^2 + 2741*A^2*a^8*b^4*c^6*d^8*f^2 + 2592*A^2*a^6*b^6*c^4*d^10*f^2 - 2536*A^2*a^3*b^9*c^5*d^9*f^2 + 2224*A^2*a^2*b^10*c^4*d^10*f^2 - 2184*A^2*a^3*b^9*c^7*d^7*f^2 - 2016*A^2*a^5*b^7*c^7*d^7*f^2 - 1984*A^2*a^7*b^5*c^7*d^7*f^2 + 1626*A^2*a^2*b^10*c^8*d^6*f^2 - 1624*A^2*a^9*b^3*c^5*d^9*f^2 + 1603*A^2*a^8*b^4*c^4*d^10*f^2 + 1296*A^2*a^5*b^7*c^9*d^5*f^2 - 1144*A^2*a^5*b^7*c^3*d^11*f^2 - 992*A^2*a^3*b^9*c^3

*d^11*f^2 + 968*A^2*a^4*b^8*c^2*d^12*f^2 - 888*A^2*a^7*b^5*c^3*d^11*f^2 + 8
49*A^2*a^4*b^8*c^8*d^6*f^2 + 808*A^2*a^2*b^10*c^2*d^12*f^2 - 616*A^2*a^9*b^
3*c^7*d^7*f^2 + 554*A^2*a^10*b^2*c^6*d^8*f^2 + 504*A^2*a^7*b^5*c^9*d^5*f^2
- 504*A^2*a^6*b^6*c^10*d^4*f^2 + 460*A^2*a^6*b^6*c^2*d^12*f^2 + 350*A^2*a^1
0*b^2*c^4*d^10*f^2 + 350*A^2*a^2*b^10*c^10*d^4*f^2 - 321*A^2*a^4*b^8*c^10*d
^4*f^2 + 216*A^2*a^5*b^7*c^11*d^3*f^2 - 216*A^2*a^3*b^9*c^11*d^3*f^2 + 182*
A^2*a^2*b^10*c^12*d^2*f^2 - 152*A^2*a^9*b^3*c^3*d^11*f^2 - 124*A^2*a^6*b^6*
c^8*d^6*f^2 - 114*A^2*a^10*b^2*c^2*d^12*f^2 + 104*A^2*a^3*b^9*c^9*d^5*f^2 +
77*A^2*a^8*b^4*c^2*d^12*f^2 + 74*A^2*a^8*b^4*c^8*d^6*f^2 - 70*A^2*a^8*b^4*
c^10*d^4*f^2 + 56*A^2*a^9*b^3*c^9*d^5*f^2 + 56*A^2*a^7*b^5*c^11*d^3*f^2 + 4
1*A^2*a^4*b^8*c^12*d^2*f^2 - 28*A^2*a^10*b^2*c^8*d^6*f^2 - 28*A^2*a^6*b^6*c
^12*d^2*f^2 + 12*B*C*b^12*c^13*d*f^2 + 24*B*C*a^12*c*d^13*f^2 - 24*A*B*b^12
*c^13*d*f^2 - 24*A*B*b^12*c*d^13*f^2 - 16*B*C*a^11*b*d^14*f^2 - 24*A*B*a^12
*c*d^13*f^2 - 16*B*C*a*b^11*c^14*f^2 - 48*A*B*a*b^11*d^14*f^2 + 16*A*B*a^11
*b*d^14*f^2 + 16*A*B*a*b^11*c^14*f^2 - 216*C^2*a^11*b*c^5*d^9*f^2 + 216*C^2
*a*b^11*c^9*d^5*f^2 + 56*C^2*a^11*b*c^3*d^11*f^2 + 56*C^2*a^9*b^3*c*d^13*f^
2 + 56*C^2*a^5*b^7*c*d^13*f^2 + 40*C^2*a^7*b^5*c*d^13*f^2 - 40*C^2*a*b^11*c
^11*d^3*f^2 + 32*C^2*a^5*b^7*c^13*d*f^2 - 24*C^2*a*b^11*c^7*d^7*f^2 - 16*C^
2*a^3*b^9*c^13*d*f^2 + 16*C^2*a^3*b^9*c*d^13*f^2 + 8*C^2*a^11*b*c^7*d^7*f^2
- 8*C^2*a*b^11*c^5*d^9*f^2 + 264*B^2*a*b^11*c^7*d^7*f^2 + 224*B^2*a*b^11*c
^5*d^9*f^2 + 168*B^2*a^11*b*c^5*d^9*f^2 - 112*B^2*a^9*b^3*c*d^13*f^2 - 104*
B^2*a^11*b*c^3*d^11*f^2 - 104*B^2*a^7*b^5*c*d^13*f^2 + 96*B^2*a*b^11*c^3*d^
11*f^2 + 88*B^2*a*b^11*c^11*d^3*f^2 - 72*B^2*a*b^11*c^9*d^5*f^2 - 64*B^2*a^
5*b^7*c*d^13*f^2 + 32*B^2*a^3*b^9*c^13*d*f^2 - 24*B^2*a^11*b*c^7*d^7*f^2 -
24*B^2*a^5*b^7*c^13*d*f^2 + 16*B^2*a^3*b^9*c*d^13*f^2 - 888*A^2*a*b^11*c^7*
d^7*f^2 - 800*A^2*a*b^11*c^5*d^9*f^2 - 336*A^2*a*b^11*c^3*d^11*f^2 - 264*A^
2*a*b^11*c^9*d^5*f^2 - 216*A^2*a^11*b*c^5*d^9*f^2 - 184*A^2*a*b^11*c^11*d^3
*f^2 - 128*A^2*a^3*b^9*c*d^13*f^2 - 112*A^2*a^5*b^7*c*d^13*f^2 - 64*A^2*a^3
*b^9*c^13*d*f^2 + 56*A^2*a^11*b*c^3*d^11*f^2 - 56*A^2*a^7*b^5*c*d^13*f^2 +
32*A^2*a^9*b^3*c*d^13*f^2 + 8*A^2*a^11*b*c^7*d^7*f^2 + 8*A^2*a^5*b^7*c^13*d
*f^2 + 24*C^2*a^11*b*c*d^13*f^2 - 16*C^2*a*b^11*c^13*d*f^2 - 40*B^2*a^11*b*
c*d^13*f^2 + 24*B^2*a*b^11*c^13*d*f^2 + 16*B^2*a*b^11*c*d^13*f^2 - 48*A^2*a
*b^11*c*d^13*f^2 - 40*A^2*a*b^11*c^13*d*f^2 + 24*A^2*a^11*b*c*d^13*f^2 - 6*
A*C*a^12*d^14*f^2 + 2*A*C*b^12*c^14*f^2 + 33*C^2*b^12*c^12*d^2*f^2 - 27*C^2
*b^12*c^10*d^4*f^2 + 3*C^2*b^12*c^8*d^6*f^2 + 117*B^2*b^12*c^10*d^4*f^2 + 1
11*B^2*b^12*c^8*d^6*f^2 + 72*B^2*b^12*c^6*d^8*f^2 + 33*C^2*a^12*c^4*d^10*f^
2 - 27*C^2*a^12*c^2*d^12*f^2 + 24*B^2*b^12*c^4*d^10*f^2 + 4*B^2*b^12*c^2*d^
12*f^2 - 3*B^2*b^12*c^12*d^2*f^2 - C^2*a^12*c^6*d^8*f^2 + 720*A^2*b^12*c^6*
d^8*f^2 + 552*A^2*b^12*c^4*d^10*f^2 + 471*A^2*b^12*c^8*d^6*f^2 + 216*A^2*b^
12*c^2*d^12*f^2 + 93*A^2*b^12*c^10*d^4*f^2 + 33*B^2*a^12*c^2*d^12*f^2 + 33*
A^2*b^12*c^12*d^2*f^2 + 31*C^2*a^8*b^4*d^14*f^2 - 27*B^2*a^12*c^4*d^10*f^2
+ 20*C^2*a^6*b^6*d^14*f^2 + 4*C^2*a^4*b^8*d^14*f^2 + 3*B^2*a^12*c^6*d^8*f^2
+ 2*C^2*a^10*b^2*d^14*f^2 + 80*B^2*a^6*b^6*d^14*f^2 + 64*B^2*a^4*b^8*d^14*
f^2 + 33*A^2*a^12*c^4*d^10*f^2 + 31*B^2*a^8*b^4*d^14*f^2 - 27*A^2*a^12*c^2*
d^12*f^2 + 16*B^2*a^2*b^10*d^14*f^2 + 14*C^2*a^2*b^10*c^14*f^2 + 14*B^2*a^1

0*b^2*d^14*f^2 - C^2*a^4*b^8*c^14*f^2 - A^2*a^12*c^6*d^8*f^2 + 120*A^2*a^2*
b^10*d^14*f^2 + 112*A^2*a^4*b^8*d^14*f^2 - 17*A^2*a^8*b^4*d^14*f^2 - 10*B^2
*a^2*b^10*c^14*f^2 - 10*A^2*a^10*b^2*d^14*f^2 + 8*A^2*a^6*b^6*d^14*f^2 + 3*
B^2*a^4*b^8*c^14*f^2 + 14*A^2*a^2*b^10*c^14*f^2 - A^2*a^4*b^8*c^14*f^2 + 3*
C^2*a^12*d^14*f^2 - C^2*b^12*c^14*f^2 + 36*A^2*b^12*d^14*f^2 + 3*B^2*b^12*c
^14*f^2 - B^2*a^12*d^14*f^2 + 3*A^2*a^12*d^14*f^2 - A^2*b^12*c^14*f^2 - 44*
A*B*C*a*b^9*c^10*d*f + 3816*A*B*C*a^5*b^5*c^4*d^7*f + 2920*A*B*C*a^2*b^8*c^
5*d^6*f - 2736*A*B*C*a^3*b^7*c^6*d^5*f - 2672*A*B*C*a^4*b^6*c^3*d^8*f + 199
6*A*B*C*a^4*b^6*c^7*d^4*f - 1412*A*B*C*a^6*b^4*c^5*d^6*f + 1120*A*B*C*a^3*b
^7*c^2*d^9*f + 1080*A*B*C*a^2*b^8*c^7*d^4*f + 1040*A*B*C*a^5*b^5*c^2*d^9*f
+ 684*A*B*C*a^4*b^6*c^5*d^6*f + 592*A*B*C*a^3*b^7*c^4*d^7*f - 560*A*B*C*a^7
*b^3*c^2*d^9*f - 448*A*B*C*a^2*b^8*c^3*d^8*f - 400*A*B*C*a^5*b^5*c^8*d^3*f
- 398*A*B*C*a^2*b^8*c^9*d^2*f - 312*A*B*C*a^6*b^4*c^3*d^8*f + 166*A*B*C*a^8
*b^2*c^3*d^8*f + 136*A*B*C*a^5*b^5*c^6*d^5*f + 128*A*B*C*a^7*b^3*c^6*d^5*f
- 100*A*B*C*a^6*b^4*c^7*d^4*f + 64*A*B*C*a^7*b^3*c^4*d^7*f - 64*A*B*C*a^4*b
^6*c^9*d^2*f - 32*A*B*C*a^3*b^7*c^8*d^3*f - 16*A*B*C*a^8*b^2*c^5*d^6*f - 13
12*A*B*C*a*b^9*c^4*d^7*f + 996*A*B*C*a*b^9*c^8*d^3*f + 728*A*B*C*a^6*b^4*c*
d^10*f - 624*A*B*C*a*b^9*c^6*d^5*f - 584*A*B*C*a^2*b^8*c*d^10*f - 512*A*B*C
*a^4*b^6*c*d^10*f - 320*A*B*C*a*b^9*c^2*d^9*f - 98*A*B*C*a^8*b^2*c*d^10*f +
36*A*B*C*a^9*b*c^2*d^9*f + 32*A*B*C*a^3*b^7*c^10*d*f - 16*A*B*C*a^9*b*c^4*
d^7*f + 46*B*C^2*a*b^9*c^10*d*f - 16*B^2*C*a*b^9*c*d^10*f - 2*B^2*C*a^9*b*c
*d^10*f + 312*A^2*C*a*b^9*c*d^10*f - 48*A*C^2*a*b^9*c*d^10*f - 6*A^2*C*a^9*
b*c*d^10*f + 6*A*C^2*a^9*b*c*d^10*f + 208*A*B^2*a*b^9*c*d^10*f - 2*A^2*B*a*
b^9*c^10*d*f + 2*A*B^2*a^9*b*c*d^10*f - 480*A*B*C*b^10*c^7*d^4*f + 78*A*B*C
*b^10*c^9*d^2*f - 64*A*B*C*b^10*c^5*d^6*f + 2*A*B*C*a^10*c^3*d^8*f - 224*A*
B*C*a^5*b^5*d^11*f + 80*A*B*C*a^7*b^3*d^11*f - 32*A*B*C*a^3*b^7*d^11*f + 2*
A*B*C*a^2*b^8*c^11*f - 1692*B*C^2*a^5*b^5*c^4*d^7*f - 1500*B^2*C*a^5*b^5*c^
5*d^6*f - 1464*B^2*C*a^3*b^7*c^5*d^6*f + 1426*B*C^2*a^6*b^4*c^5*d^6*f - 115
8*B^2*C*a^6*b^4*c^4*d^7*f + 1152*B*C^2*a^3*b^7*c^6*d^5*f + 1026*B^2*C*a^4*b
^6*c^6*d^5*f - 974*B*C^2*a^4*b^6*c^7*d^4*f + 960*B^2*C*a^5*b^5*c^3*d^8*f -
884*B*C^2*a^2*b^8*c^5*d^6*f - 764*B^2*C*a^5*b^5*c^7*d^4*f + 752*B^2*C*a^2*b
^8*c^4*d^7*f - 752*B*C^2*a^3*b^7*c^4*d^7*f + 738*B^2*C*a^4*b^6*c^4*d^7*f -
688*B^2*C*a^6*b^4*c^2*d^9*f - 675*B^2*C*a^2*b^8*c^8*d^3*f + 560*B*C^2*a^5*b
^5*c^8*d^3*f + 496*B*C^2*a^7*b^3*c^2*d^9*f + 496*B*C^2*a^4*b^6*c^3*d^8*f -
468*B*C^2*a^2*b^8*c^7*d^4*f + 456*B^2*C*a^7*b^3*c^3*d^8*f - 452*B^2*C*a^4*b
^6*c^8*d^3*f - 416*B*C^2*a^3*b^7*c^2*d^9*f + 378*B*C^2*a^4*b^6*c^5*d^6*f +
376*B*C^2*a^3*b^7*c^8*d^3*f - 360*B^2*C*a^2*b^8*c^6*d^5*f + 355*B*C^2*a^2*b
^8*c^9*d^2*f + 346*B^2*C*a^6*b^4*c^6*d^5*f - 320*B^2*C*a^4*b^6*c^2*d^9*f +
268*B^2*C*a^2*b^8*c^2*d^9*f + 216*B^2*C*a^3*b^7*c^7*d^4*f - 203*B*C^2*a^8*b
^2*c^3*d^8*f - 184*B*C^2*a^7*b^3*c^6*d^5*f + 170*B*C^2*a^6*b^4*c^7*d^4*f +
160*B^2*C*a^7*b^3*c^5*d^6*f - 160*B*C^2*a^5*b^5*c^2*d^9*f - 140*B^2*C*a^8*b
^2*c^4*d^7*f - 136*B*C^2*a^2*b^8*c^3*d^8*f + 112*B^2*C*a^3*b^7*c^9*d^2*f +
91*B^2*C*a^8*b^2*c^2*d^9*f + 88*B*C^2*a^7*b^3*c^4*d^7*f + 72*B^2*C*a^6*b^4*
c^8*d^3*f - 64*B^2*C*a^3*b^7*c^3*d^8*f - 60*B*C^2*a^6*b^4*c^3*d^8*f + 56*B*
C^2*a^4*b^6*c^9*d^2*f + 52*B*C^2*a^5*b^5*c^6*d^5*f - 48*B^2*C*a^7*b^3*c^7*d

^4*f + 48*B^2*C*a^5*b^5*c^9*d^2*f + 44*B*C^2*a^8*b^2*c^5*d^6*f - 36*B*C^2*a
^6*b^4*c^9*d^2*f + 12*B^2*C*a^8*b^2*c^6*d^5*f - 2958*A^2*C*a^4*b^6*c^4*d^7*
f - 1932*A^2*C*a^2*b^8*c^4*d^7*f + 1848*A^2*C*a^3*b^7*c^5*d^6*f + 1728*A^2*
C*a^3*b^7*c^3*d^8*f + 1524*A^2*C*a^5*b^5*c^5*d^6*f + 1374*A*C^2*a^4*b^6*c^4
*d^7*f - 1272*A*C^2*a^3*b^7*c^5*d^6*f - 1236*A*C^2*a^5*b^5*c^5*d^6*f + 1116
*A*C^2*a^2*b^8*c^4*d^7*f - 1110*A^2*C*a^4*b^6*c^6*d^5*f + 1038*A*C^2*a^4*b^
6*c^6*d^5*f - 768*A^2*C*a^2*b^8*c^2*d^9*f - 696*A^2*C*a^3*b^7*c^7*d^4*f - 6
66*A*C^2*a^6*b^4*c^4*d^7*f + 564*A^2*C*a^2*b^8*c^6*d^5*f - 564*A*C^2*a^5*b^
5*c^7*d^4*f - 555*A*C^2*a^2*b^8*c^8*d^3*f + 519*A^2*C*a^2*b^8*c^8*d^3*f - 4
80*A*C^2*a^3*b^7*c^3*d^8*f + 456*A*C^2*a^5*b^5*c^3*d^8*f - 420*A*C^2*a^6*b^
4*c^2*d^9*f + 408*A*C^2*a^3*b^7*c^7*d^4*f + 408*A*C^2*a^2*b^8*c^2*d^9*f + 3
48*A^2*C*a^6*b^4*c^2*d^9*f - 348*A*C^2*a^2*b^8*c^6*d^5*f + 342*A*C^2*a^6*b^
4*c^6*d^5*f - 336*A*C^2*a^4*b^6*c^8*d^3*f + 324*A^2*C*a^5*b^5*c^7*d^4*f - 3
12*A^2*C*a^4*b^6*c^2*d^9*f + 264*A^2*C*a^4*b^6*c^8*d^3*f + 240*A*C^2*a^7*b^
3*c^5*d^6*f + 195*A*C^2*a^8*b^2*c^2*d^9*f - 174*A^2*C*a^6*b^4*c^6*d^5*f + 1
44*A*C^2*a^3*b^7*c^9*d^2*f - 123*A^2*C*a^8*b^2*c^2*d^9*f + 120*A*C^2*a^7*b^
3*c^3*d^8*f + 108*A*C^2*a^6*b^4*c^8*d^3*f - 102*A^2*C*a^6*b^4*c^4*d^7*f - 9
6*A^2*C*a^8*b^2*c^4*d^7*f + 72*A^2*C*a^7*b^3*c^3*d^8*f + 72*A*C^2*a^5*b^5*c
^9*d^2*f + 48*A^2*C*a^7*b^3*c^5*d^6*f - 48*A^2*C*a^3*b^7*c^9*d^2*f - 48*A*C
^2*a^4*b^6*c^2*d^9*f - 24*A^2*C*a^5*b^5*c^3*d^8*f - 12*A*C^2*a^8*b^2*c^4*d^
7*f + 2736*A^2*B*a^3*b^7*c^6*d^5*f + 2464*A^2*B*a^4*b^6*c^3*d^8*f - 2298*A*
B^2*a^4*b^6*c^4*d^7*f - 2252*A^2*B*a^2*b^8*c^5*d^6*f - 1692*A^2*B*a^5*b^5*c
^4*d^7*f - 1592*A*B^2*a^2*b^8*c^4*d^7*f - 1338*A*B^2*a^4*b^6*c^6*d^5*f + 13
20*A*B^2*a^3*b^7*c^5*d^6*f + 1212*A*B^2*a^5*b^5*c^5*d^6*f - 1056*A*B^2*a^5*
b^5*c^3*d^8*f + 1024*A^2*B*a^3*b^7*c^4*d^7*f - 1022*A^2*B*a^4*b^6*c^7*d^4*f
 - 880*A^2*B*a^5*b^5*c^2*d^9*f - 846*A^2*B*a^4*b^6*c^5*d^6*f - 840*A*B^2*a^
3*b^7*c^7*d^4*f + 760*A*B^2*a^6*b^4*c^2*d^9*f - 704*A^2*B*a^3*b^7*c^2*d^9*f
 + 688*A*B^2*a^3*b^7*c^3*d^8*f + 660*A^2*B*a^6*b^4*c^3*d^8*f - 612*A^2*B*a^
2*b^8*c^7*d^4*f + 462*A*B^2*a^6*b^4*c^4*d^7*f + 459*A*B^2*a^2*b^8*c^8*d^3*f
 - 412*A*B^2*a^2*b^8*c^2*d^9*f - 408*A*B^2*a^7*b^3*c^3*d^8*f + 388*A^2*B*a^
5*b^5*c^6*d^5*f + 296*A^2*B*a^2*b^8*c^3*d^8*f + 288*A*B^2*a^2*b^8*c^6*d^5*f
 + 284*A*B^2*a^5*b^5*c^7*d^4*f + 236*A*B^2*a^4*b^6*c^8*d^3*f - 226*A*B^2*a^
6*b^4*c^6*d^5*f + 212*A*B^2*a^4*b^6*c^2*d^9*f + 202*A^2*B*a^6*b^4*c^5*d^6*f
 - 152*A^2*B*a^7*b^3*c^4*d^7*f + 88*A^2*B*a^3*b^7*c^8*d^3*f + 79*A^2*B*a^2*
b^8*c^9*d^2*f - 70*A^2*B*a^6*b^4*c^7*d^4*f + 68*A*B^2*a^8*b^2*c^4*d^7*f + 6
4*A^2*B*a^7*b^3*c^2*d^9*f - 64*A*B^2*a^3*b^7*c^9*d^2*f + 56*A^2*B*a^7*b^3*c
^6*d^5*f + 56*A^2*B*a^5*b^5*c^8*d^3*f + 37*A^2*B*a^8*b^2*c^3*d^8*f - 28*A^2
*B*a^8*b^2*c^5*d^6*f - 28*A^2*B*a^4*b^6*c^9*d^2*f + 17*A*B^2*a^8*b^2*c^2*d^
9*f - 16*A*B^2*a^7*b^3*c^5*d^6*f + 24*A*B*C*b^10*c*d^10*f - 6*A*B*C*a^10*c*
d^10*f + 48*A*B*C*a*b^9*d^11*f + 4*A*B*C*a^9*b*d^11*f + 432*B^2*C*a*b^9*c^7
*d^4*f - 376*B*C^2*a^6*b^4*c*d^10*f - 354*B*C^2*a*b^9*c^8*d^3*f + 352*B^2*C
*a^5*b^5*c*d^10*f + 320*B^2*C*a*b^9*c^5*d^6*f + 256*B^2*C*a^3*b^7*c*d^10*f
- 232*B^2*C*a^7*b^3*c*d^10*f - 210*B^2*C*a*b^9*c^9*d^2*f - 152*B*C^2*a^4*b^
6*c*d^10*f + 85*B*C^2*a^8*b^2*c*d^10*f + 72*B^2*C*a*b^9*c^3*d^8*f - 48*B*C^
2*a*b^9*c^6*d^5*f - 40*B*C^2*a^3*b^7*c^10*d*f + 40*B*C^2*a^2*b^8*c*d^10*f +

37*B^2*C*a^2*b^8*c^10*d*f + 22*B^2*C*a^9*b*c^3*d^8*f - 18*B*C^2*a^9*b*c^2*
d^9*f + 16*B*C^2*a*b^9*c^2*d^9*f - 12*B^2*C*a^4*b^6*c^10*d*f + 8*B*C^2*a^9*
b*c^4*d^7*f + 8*B*C^2*a*b^9*c^4*d^7*f - 984*A^2*C*a*b^9*c^7*d^4*f + 672*A^2
*C*a*b^9*c^3*d^8*f + 552*A*C^2*a*b^9*c^7*d^4*f - 504*A^2*C*a^5*b^5*c*d^10*f
 - 408*A^2*C*a*b^9*c^5*d^6*f + 408*A*C^2*a*b^9*c^5*d^6*f + 336*A*C^2*a^5*b^
5*c*d^10*f - 216*A*C^2*a^7*b^3*c*d^10*f + 192*A*C^2*a^3*b^7*c*d^10*f - 162*
A*C^2*a*b^9*c^9*d^2*f + 120*A^2*C*a^7*b^3*c*d^10*f + 96*A^2*C*a^3*b^7*c*d^1
0*f + 90*A^2*C*a*b^9*c^9*d^2*f + 66*A^2*C*a^9*b*c^3*d^8*f - 66*A*C^2*a^9*b*
c^3*d^8*f + 57*A*C^2*a^2*b^8*c^10*d*f - 48*A*C^2*a*b^9*c^3*d^8*f - 9*A^2*C*
a^2*b^8*c^10*d*f + 1736*A^2*B*a*b^9*c^4*d^7*f + 1248*A^2*B*a*b^9*c^6*d^5*f
 - 1008*A*B^2*a*b^9*c^7*d^4*f + 772*A^2*B*a^4*b^6*c*d^10*f - 688*A*B^2*a^5*b
^5*c*d^10*f - 608*A*B^2*a*b^9*c^5*d^6*f + 436*A^2*B*a^2*b^8*c*d^10*f - 426*
A^2*B*a*b^9*c^8*d^3*f + 312*A*B^2*a*b^9*c^3*d^8*f + 304*A^2*B*a*b^9*c^2*d^9
*f - 244*A^2*B*a^6*b^4*c*d^10*f - 160*A*B^2*a^3*b^7*c*d^10*f + 114*A*B^2*a*
b^9*c^9*d^2*f + 88*A*B^2*a^7*b^3*c*d^10*f - 22*A*B^2*a^9*b*c^3*d^8*f - 18*A
^2*B*a^9*b*c^2*d^9*f + 13*A^2*B*a^8*b^2*c*d^10*f - 13*A*B^2*a^2*b^8*c^10*d*
f + 8*A^2*B*a^9*b*c^4*d^7*f + 8*A^2*B*a^3*b^7*c^10*d*f + 111*B^2*C*b^10*c^8
*d^3*f - 39*B*C^2*b^10*c^9*d^2*f + 24*B*C^2*b^10*c^7*d^4*f - 4*B^2*C*b^10*c
^2*d^9*f - 4*B*C^2*b^10*c^5*d^6*f + 432*A^2*C*b^10*c^6*d^5*f + 192*A^2*C*b^
10*c^4*d^7*f - 111*A^2*C*b^10*c^8*d^3*f + 111*A*C^2*b^10*c^8*d^3*f - 72*A*C
^2*b^10*c^6*d^5*f + 12*A*C^2*b^10*c^4*d^7*f - 3*B^2*C*a^10*c^2*d^9*f - B*C^
2*a^10*c^3*d^8*f + 456*A^2*B*b^10*c^7*d^4*f - 288*A^2*B*b^10*c^3*d^8*f + 25
2*A*B^2*b^10*c^6*d^5*f + 192*A*B^2*b^10*c^4*d^7*f - 183*A*B^2*b^10*c^8*d^3*
f - 148*A^2*B*b^10*c^5*d^6*f + 112*B^2*C*a^6*b^4*d^11*f + 76*A*B^2*b^10*c^2
*d^9*f - 64*B*C^2*a^7*b^3*d^11*f + 16*B^2*C*a^4*b^6*d^11*f - 16*B^2*C*a^2*b
^8*d^11*f + 16*B*C^2*a^5*b^5*d^11*f + 16*B*C^2*a^3*b^7*d^11*f - 9*A^2*C*a^1
0*c^2*d^9*f + 9*A*C^2*a^10*c^2*d^9*f - 3*A^2*B*b^10*c^9*d^2*f - B^2*C*a^8*b
^2*d^11*f + 96*A^2*C*a^4*b^6*d^11*f - 84*A^2*C*a^6*b^4*d^11*f + 72*A*C^2*a^
6*b^4*d^11*f - 24*A*C^2*a^4*b^6*d^11*f - 24*A*C^2*a^2*b^8*d^11*f - 21*A*C^2
*a^8*b^2*d^11*f + 12*A^2*C*a^2*b^8*d^11*f + 9*A^2*C*a^8*b^2*d^11*f + 3*A*B^
2*a^10*c^2*d^9*f - A^2*B*a^10*c^3*d^8*f - B*C^2*a^2*b^8*c^11*f + 176*A*B^2*
a^4*b^6*d^11*f + 136*A^2*B*a^5*b^5*d^11*f - 128*A^2*B*a^3*b^7*d^11*f + 112*
A*B^2*a^2*b^8*d^11*f - 64*A*B^2*a^6*b^4*d^11*f - 16*A^2*B*a^7*b^3*d^11*f -
A^2*B*a^2*b^8*c^11*f - 2*C^3*a^9*b*c*d^10*f - 2*B^3*a*b^9*c^10*d*f - 264*A^
3*a*b^9*c*d^10*f + 2*A^3*a^9*b*c*d^10*f - 9*B^2*C*b^10*c^10*d*f + 9*A^2*C*b
^10*c^10*d*f - 9*A*C^2*b^10*c^10*d*f + 3*B*C^2*a^10*c*d^10*f - 132*A^2*B*b^
10*c*d^10*f - 3*A*B^2*b^10*c^10*d*f - 2*B*C^2*a^9*b*d^11*f + 3*A^2*B*a^10*c
*d^10*f - 2*B^2*C*a*b^9*c^11*f - 120*A^2*B*a*b^9*d^11*f - 6*A^2*C*a*b^9*c^1
1*f + 6*A*C^2*a*b^9*c^11*f - 2*A^2*B*a^9*b*d^11*f + 2*A*B^2*a*b^9*c^11*f +
520*C^3*a^3*b^7*c^5*d^6*f + 460*C^3*a^5*b^5*c^5*d^6*f - 418*C^3*a^4*b^6*c^6
*d^5*f + 406*C^3*a^6*b^4*c^4*d^7*f + 268*C^3*a^5*b^5*c^7*d^4*f - 266*C^3*a^
6*b^4*c^6*d^5*f + 233*C^3*a^2*b^8*c^8*d^3*f - 176*C^3*a^7*b^3*c^5*d^6*f + 1
64*C^3*a^6*b^4*c^2*d^9*f + 140*C^3*a^2*b^8*c^6*d^5*f + 136*C^3*a^4*b^6*c^2*
d^9*f - 128*C^3*a^3*b^7*c^9*d^2*f + 128*C^3*a^3*b^7*c^3*d^8*f - 108*C^3*a^6
*b^4*c^8*d^3*f - 104*C^3*a^7*b^3*c^3*d^8*f - 104*C^3*a^5*b^5*c^3*d^8*f + 10

0*C^3*a^4*b^6*c^8*d^3*f - 89*C^3*a^8*b^2*c^2*d^9*f - 72*C^3*a^5*b^5*c^9*d^2
*f + 40*C^3*a^8*b^2*c^4*d^7*f - 40*C^3*a^3*b^7*c^7*d^4*f - 28*C^3*a^2*b^8*c
^4*d^7*f - 16*C^3*a^2*b^8*c^2*d^9*f - 2*C^3*a^4*b^6*c^4*d^7*f + 828*B^3*a^5
*b^5*c^4*d^7*f + 408*B^3*a^2*b^8*c^5*d^6*f + 390*B^3*a^4*b^6*c^7*d^4*f - 37
2*B^3*a^4*b^6*c^3*d^8*f - 336*B^3*a^3*b^7*c^6*d^5*f - 314*B^3*a^6*b^4*c^5*d
^6*f + 288*B^3*a^3*b^7*c^4*d^7*f + 216*B^3*a^2*b^8*c^7*d^4*f - 176*B^3*a^7*
b^3*c^2*d^9*f + 128*B^3*a^3*b^7*c^2*d^9*f + 108*B^3*a^5*b^5*c^6*d^5*f + 88*
B^3*a^7*b^3*c^4*d^7*f + 72*B^3*a^5*b^5*c^2*d^9*f - 68*B^3*a^2*b^8*c^3*d^8*f
 - 65*B^3*a^2*b^8*c^9*d^2*f - 56*B^3*a^5*b^5*c^8*d^3*f + 40*B^3*a^7*b^3*c^6
*d^5*f + 37*B^3*a^8*b^2*c^3*d^8*f + 30*B^3*a^4*b^6*c^5*d^6*f - 28*B^3*a^8*b
^2*c^5*d^6*f + 24*B^3*a^3*b^7*c^8*d^3*f - 4*B^3*a^4*b^6*c^9*d^2*f - 2*B^3*a
^6*b^4*c^7*d^4*f + 1586*A^3*a^4*b^6*c^4*d^7*f - 1376*A^3*a^3*b^7*c^3*d^8*f
- 1096*A^3*a^3*b^7*c^5*d^6*f + 844*A^3*a^2*b^8*c^4*d^7*f - 748*A^3*a^5*b^5*
c^5*d^6*f + 490*A^3*a^4*b^6*c^6*d^5*f + 376*A^3*a^2*b^8*c^2*d^9*f + 362*A^3
*a^6*b^4*c^4*d^7*f - 356*A^3*a^2*b^8*c^6*d^5*f - 328*A^3*a^5*b^5*c^3*d^8*f
+ 328*A^3*a^3*b^7*c^7*d^4*f + 224*A^3*a^4*b^6*c^2*d^9*f - 197*A^3*a^2*b^8*c
^8*d^3*f - 112*A^3*a^7*b^3*c^5*d^6*f + 98*A^3*a^6*b^4*c^6*d^5*f - 92*A^3*a^
6*b^4*c^2*d^9*f - 88*A^3*a^7*b^3*c^3*d^8*f + 68*A^3*a^8*b^2*c^4*d^7*f + 32*
A^3*a^3*b^7*c^9*d^2*f - 28*A^3*a^5*b^5*c^7*d^4*f - 28*A^3*a^4*b^6*c^8*d^3*f
 + 17*A^3*a^8*b^2*c^2*d^9*f + 104*C^3*a^7*b^3*c*d^10*f + 54*C^3*a*b^9*c^9*d
^2*f - 40*C^3*a*b^9*c^7*d^4*f - 35*C^3*a^2*b^8*c^10*d*f + 22*C^3*a^9*b*c^3*
d^8*f + 16*C^3*a^5*b^5*c*d^10*f - 16*C^3*a^3*b^7*c*d^10*f + 8*C^3*a*b^9*c^5
*d^6*f - 2*A*B*C*b^10*c^11*f + 198*B^3*a*b^9*c^8*d^3*f + 192*B^3*a^6*b^4*c*
d^10*f - 128*B^3*a*b^9*c^4*d^7*f - 80*B^3*a^2*b^8*c*d^10*f - 56*B^3*a*b^9*c
^2*d^9*f - 24*B^3*a*b^9*c^6*d^5*f - 18*B^3*a^9*b*c^2*d^9*f - 16*B^3*a^4*b^6
*c*d^10*f + 13*B^3*a^8*b^2*c*d^10*f + 8*B^3*a^9*b*c^4*d^7*f + 8*B^3*a^3*b^7
*c^10*d*f - 624*A^3*a*b^9*c^3*d^8*f + 472*A^3*a*b^9*c^7*d^4*f - 272*A^3*a^3
*b^7*c*d^10*f + 152*A^3*a^5*b^5*c*d^10*f - 22*A^3*a^9*b*c^3*d^8*f + 18*A^3*
a*b^9*c^9*d^2*f - 13*A^3*a^2*b^8*c^10*d*f - 8*A^3*a^7*b^3*c*d^10*f - 8*A^3*
a*b^9*c^5*d^6*f + A*B^2*a^8*b^2*d^11*f - C^3*b^10*c^8*d^3*f - 60*B^3*b^10*c
^7*d^4*f - 32*B^3*b^10*c^5*d^6*f + 21*B^3*b^10*c^9*d^2*f - 12*B^3*b^10*c^3*
d^8*f - 3*C^3*a^10*c^2*d^9*f - 360*A^3*b^10*c^6*d^5*f - 204*A^3*b^10*c^4*d^
7*f + 11*C^3*a^8*b^2*d^11*f - 8*C^3*a^6*b^4*d^11*f - 4*C^3*a^4*b^6*d^11*f -
 B^3*a^10*c^3*d^8*f - 64*B^3*a^5*b^5*d^11*f - 32*B^3*a^3*b^7*d^11*f + 3*A^3
*a^10*c^2*d^9*f - 68*A^3*a^4*b^6*d^11*f + 20*A^3*a^6*b^4*d^11*f + 12*A^3*a^
2*b^8*d^11*f - B^3*a^2*b^8*c^11*f + 3*C^3*b^10*c^10*d*f + 3*B^3*a^10*c*d^10
*f - 3*A^3*b^10*c^10*d*f - 2*C^3*a*b^9*c^11*f - 2*B^3*a^9*b*d^11*f + 2*A^3*
a*b^9*c^11*f - 36*A^2*C*b^10*d^11*f + 3*A^2*C*a^10*d^11*f - 3*A*C^2*a^10*d^
11*f - A*B^2*a^10*d^11*f + 36*A^3*b^10*d^11*f - A^3*a^10*d^11*f + A^3*b^10*
c^8*d^3*f + A^3*a^8*b^2*d^11*f + B^2*C*a^10*d^11*f + B*C^2*b^10*c^11*f + A^
2*B*b^10*c^11*f + C^3*a^10*d^11*f + B^3*b^10*c^11*f - 6*A*B^2*C*a*b^7*c^7*d
 + 4*A*B^2*C*a*b^7*c*d^7 + 168*A^2*B*C*a^3*b^5*c^2*d^6 + 144*A*B*C^2*a^4*b^
4*c^3*d^5 - 129*A^2*B*C*a^4*b^4*c^3*d^5 - 96*A*B*C^2*a^3*b^5*c^2*d^6 + 84*A
*B*C^2*a^2*b^6*c^3*d^5 + 72*A^2*B*C*a^3*b^5*c^4*d^4 - 72*A^2*B*C*a^2*b^6*c^
3*d^5 + 64*A*A*B^2*C*a^4*b^4*c^4*d^4 - 60*A*B*C^2*a^3*b^5*c^4*d^4 + 57*A^2*B*

C*a^2*b^6*c^5*d^3 - 56*A*B^2*C*a^3*b^5*c^5*d^3 - 39*A*B^2*C*a^4*b^4*c^2*d^6 - 38*A*B^2*C*a^5*b^3*c^3*d^5 + 36*A*B^2*C*a^3*b^5*c^3*d^5 + 36*A*B*C^2*a^4*b^4*c^5*d^3 - 30*A*B*C^2*a^2*b^6*c^5*d^3 + 27*A*B^2*C*a^2*b^6*c^6*d^2 - 24*A*B^2*C*a^2*b^6*c^2*d^6 - 24*A*B*C^2*a^5*b^3*c^4*d^4 + 24*A*B*C^2*a^3*b^5*c^6*d^2 + 18*A^2*B*C*a^5*b^3*c^2*d^6 - 18*A^2*B*C*a^4*b^4*c^5*d^3 - 15*A*B^2*C*a^2*b^6*c^4*d^4 + 12*A^2*B*C*a^5*b^3*c^4*d^4 - 12*A^2*B*C*a^3*b^5*c^6*d^2 + 9*A*B^2*C*a^6*b^2*c^2*d^6 + 6*A*B*C^2*a^6*b^2*c^3*d^5 - 3*A^2*B*C*a^6*b^2*c^3*d^5 + 60*A^2*B*C*a*b^7*c^2*d^6 - 51*A^2*B*C*a^4*b^4*c*d^7 + 48*A*B*C^2*a*b^7*c^6*d^2 - 42*A^2*B*C*a^2*b^6*c*d^7 - 42*A^2*B*C*a*b^7*c^6*d^2 + 36*A*B*C^2*a^4*b^4*c*d^7 + 36*A*B*C^2*a^2*b^6*c*d^7 + 36*A*B*C^2*a*b^7*c^4*d^4 - 30*A^2*B*C*a*b^7*c^4*d^4 + 24*A*B^2*C*a*b^7*c^3*d^5 - 24*A*B*C^2*a*b^7*c^2*d^6 + 18*A*B^2*C*a^5*b^3*c*d^7 - 18*A*B*C^2*a^6*b^2*c*d^7 + 12*A*B^2*C*a^3*b^5*c*d^7 + 9*A^2*B*C*a^6*b^2*c*d^7 + 6*A*B^2*C*a*b^7*c^5*d^3 - 6*A*B*C^2*a^2*b^6*c^7*d + 3*A^2*B*C*a^2*b^6*c^7*d - 18*B^3*C*a*b^7*c^6*d^2 - 18*B*C^3*a*b^7*c^6*d^2 - 14*B^3*C*a*b^7*c^4*d^4 - 14*B*C^3*a*b^7*c^4*d^4 - 10*B^3*C*a^2*b^6*c*d^7 - 10*B*C^3*a^2*b^6*c*d^7 + 9*B^3*C*a^6*b^2*c*d^7 + 9*B*C^3*a^6*b^2*c*d^7 - 7*B^3*C*a^4*b^4*c*d^7 - 7*B*C^3*a^4*b^4*c*d^7 + 6*B^2*C^2*a*b^7*c^7*d - 4*B^3*C*a*b^7*c^2*d^6 + 4*B^2*C^2*a*b^7*c*d^7 - 4*B*C^3*a*b^7*c^2*d^6 + 3*B^3*C*a^2*b^6*c^7*d + 3*B*C^3*a^2*b^6*c^7*d + 144*A^3*C*a*b^7*c^3*d^5 + 62*A^3*C*a*b^7*c^5*d^3 + 48*A*C^3*a*b^7*c^3*d^5 - 36*A^2*C^2*a*b^7*c*d^7 + 26*A*C^3*a*b^7*c^5*d^3 + 20*A^3*C*a^3*b^5*c*d^7 + 18*A^2*C^2*a*b^7*c^7*d - 18*A*C^3*a^5*b^3*c*d^7 - 6*A^3*C*a^5*b^3*c*d^7 - 4*A*C^3*a^3*b^5*c*d^7 - 32*A^3*B*a*b^7*c^2*d^6 - 32*A*B^3*a*b^7*c^2*d^6 + 22*A^3*B*a^4*b^4*c*d^7 + 22*A*B^3*a^4*b^4*c*d^7 + 16*A^3*B*a^2*b^6*c*d^7 + 16*A*B^3*a^2*b^6*c*d^7 + 12*A^3*B*a*b^7*c^6*d^2 + 12*A*B^3*a*b^7*c^6*d^2 + 8*A^3*B*a*b^7*c^4*d^4 - 8*A^2*B^2*a*b^7*c*d^7 + 8*A*B^3*a*b^7*c^4*d^4 + 57*A^2*B*C*b^8*c^5*d^3 + 36*A^2*B*C*b^8*c^3*d^5 - 30*A*B*C^2*b^8*c^5*d^3 - 18*A*B*C^2*b^8*c^3*d^5 - 9*A*B^2*C*b^8*c^4*d^4 - 3*A*B^2*C*b^8*c^6*d^2 - 2*A*B^2*C*b^8*c^2*d^6 + 36*A^2*B*C*a^3*b^5*d^8 + 24*A*B*C^2*a^5*b^3*d^8 - 18*A^2*B*C*a^5*b^3*d^8 - 12*A*B*C^2*a^3*b^5*d^8 - 3*A*B^2*C*a^6*b^2*d^8 - 3*A*B^2*C*a^4*b^4*d^8 - 2*A*B^2*C*a^2*b^6*d^8 + 34*B^2*C^2*a^5*b^3*c^3*d^5 + 28*B^2*C^2*a^3*b^5*c^5*d^3 + 24*B^2*C^2*a^4*b^4*c^2*d^6 - 20*B^2*C^2*a^4*b^4*c^4*d^4 + 12*B^2*C^2*a^3*b^5*c^3*d^5 + 12*B^2*C^2*a^2*b^6*c^2*d^6 - 9*B^2*C^2*a^6*b^2*c^2*d^6 + 9*B^2*C^2*a^4*b^4*c^6*d^2 + 9*B^2*C^2*a^2*b^6*c^4*d^4 - 3*B^2*C^2*a^2*b^6*c^6*d^2 + 159*A^2*C^2*a^2*b^6*c^4*d^4 - 156*A^2*C^2*a^3*b^5*c^3*d^5 + 90*A^2*C^2*a^5*b^3*c^3*d^5 + 78*A^2*C^2*a^2*b^6*c^2*d^6 - 63*A^2*C^2*a^4*b^4*c^4*d^4 - 27*A^2*C^2*a^6*b^2*c^2*d^6 - 27*A^2*C^2*a^2*b^6*c^6*d^2 - 18*A^2*C^2*a^4*b^4*c^2*d^6 + 9*A^2*C^2*a^4*b^4*c^6*d^2 + 66*A^2*B^2*a^2*b^6*c^2*d^6 + 60*A^2*B^2*a^2*b^6*c^4*d^4 - 48*A^2*B^2*a^3*b^5*c^3*d^5 + 42*A^2*B^2*a^4*b^4*c^2*d^6 + 28*A^2*B^2*a^3*b^5*c^5*d^3 - 17*A^2*B^2*a^4*b^4*c^4*d^4 - 6*A^2*B^2*a^2*b^6*c^6*d^2 + 4*A^2*B^2*a^5*b^3*c^3*d^5 + 36*A^3*C*a*b^7*c*d^7 - 18*A*C^3*a*b^7*c^7*d + 12*A*C^3*a*b^7*c*d^7 - 6*A^3*C*a*b^7*c^7*d + 12*A^2*B*C*b^8*c*d^7 + 6*A*B*C^2*b^8*c^7*d - 6*A*B*C^2*b^8*c*d^7 - 3*A^2*B*C*b^8*c^7*d + 24*A^2*B*C*a*b^7*d^8 - 12*A*B*C^2*a*b^7*d^8 - 53*B^3*C*a^4*b^4*c^3*d^5 - 53*B*C^3*a^4*b^4*c^3*d^5 - 32*B^3*C*a^2*b^6*c^3*d^5 - 32*B*C^3*a^2*b

^6*c^3*d^5 - 18*B^3*C*a^4*b^4*c^5*d^3 - 18*B*C^3*a^4*b^4*c^5*d^3 + 16*B^3*C
*a^3*b^5*c^4*d^4 + 16*B*C^3*a^3*b^5*c^4*d^4 + 12*B^3*C*a^5*b^3*c^4*d^4 - 12
*B^3*C*a^3*b^5*c^6*d^2 + 12*B^2*C^2*a*b^7*c^3*d^5 + 12*B*C^3*a^5*b^3*c^4*d^
4 - 12*B*C^3*a^3*b^5*c^6*d^2 + 8*B^3*C*a^3*b^5*c^2*d^6 + 8*B*C^3*a^3*b^5*c^
2*d^6 - 6*B^3*C*a^5*b^3*c^2*d^6 - 6*B^2*C^2*a^5*b^3*c*d^7 + 6*B^2*C^2*a*b^7
*c^5*d^3 - 6*B*C^3*a^5*b^3*c^2*d^6 - 3*B^3*C*a^6*b^2*c^3*d^5 - 3*B*C^3*a^6*
b^2*c^3*d^5 - 175*A^3*C*a^2*b^6*c^4*d^4 + 164*A^3*C*a^3*b^5*c^3*d^5 - 144*A
^2*C^2*a*b^7*c^3*d^5 - 124*A^3*C*a^2*b^6*c^2*d^6 - 90*A*C^3*a^5*b^3*c^3*d^5
 - 73*A*C^3*a^2*b^6*c^4*d^4 - 66*A^2*C^2*a*b^7*c^5*d^3 + 44*A*C^3*a^3*b^5*c
^3*d^5 + 36*A*C^3*a^4*b^4*c^4*d^4 - 30*A^3*C*a^5*b^3*c^3*d^5 + 30*A^3*C*a^4
*b^4*c^4*d^4 + 27*A*C^3*a^6*b^2*c^2*d^6 + 21*A*C^3*a^4*b^4*c^2*d^6 + 18*A^2
*C^2*a^5*b^3*c*d^7 - 18*A*C^3*a^4*b^4*c^6*d^2 - 16*A*C^3*a^2*b^6*c^2*d^6 -
15*A^3*C*a^4*b^4*c^2*d^6 + 15*A^3*C*a^2*b^6*c^6*d^2 - 12*A^2*C^2*a^3*b^5*c*
d^7 + 9*A^3*C*a^6*b^2*c^2*d^6 + 9*A*C^3*a^2*b^6*c^6*d^2 - 80*A^3*B*a^3*b^5*
c^2*d^6 - 80*A*B^3*a^3*b^5*c^2*d^6 + 38*A^3*B*a^4*b^4*c^3*d^5 + 38*A*B^3*a^
4*b^4*c^3*d^5 - 36*A^2*B^2*a*b^7*c^3*d^5 - 28*A^3*B*a^3*b^5*c^4*d^4 - 28*A^
3*B*a^2*b^6*c^5*d^3 - 28*A*B^3*a^3*b^5*c^4*d^4 - 28*A*B^3*a^2*b^6*c^5*d^3 +
 20*A^3*B*a^2*b^6*c^3*d^5 + 20*A*B^3*a^2*b^6*c^3*d^5 - 12*A^3*B*a^5*b^3*c^2
*d^6 - 12*A^2*B^2*a^5*b^3*c*d^7 - 12*A^2*B^2*a^3*b^5*c*d^7 - 12*A^2*B^2*a*b
^7*c^5*d^3 - 12*A*B^3*a^5*b^3*c^2*d^6 + 6*B^2*C^2*b^8*c^6*d^2 + 3*B^2*C^2*b
^8*c^4*d^4 + 36*A^2*C^2*b^8*c^4*d^4 + 27*A^2*C^2*b^8*c^2*d^6 - 18*A^2*C^2*b
^8*c^6*d^2 + 33*A^2*B^2*b^8*c^4*d^4 + 28*A^2*B^2*b^8*c^2*d^6 + 9*B^2*C^2*a^
4*b^4*d^8 + 6*A^2*B^2*b^8*c^6*d^2 + 4*B^2*C^2*a^2*b^6*d^8 + 3*B^2*C^2*a^6*b
^2*d^8 - 30*A^2*C^2*a^4*b^4*d^8 + 9*A^2*C^2*a^6*b^2*d^8 + 16*A^2*B^2*a^2*b^
6*d^8 + 3*A^2*B^2*a^4*b^4*d^8 + 6*C^4*a^5*b^3*c*d^7 + 4*C^4*a^3*b^5*c*d^7 -
 2*C^4*a*b^7*c^5*d^3 - 12*B^4*a^5*b^3*c*d^7 + 12*B^4*a*b^7*c^3*d^5 + 8*B^4*
a*b^7*c^5*d^3 - 4*B^4*a^3*b^5*c*d^7 - 48*A^4*a*b^7*c^3*d^5 - 20*A^4*a*b^7*c
^5*d^3 - 8*A^4*a^3*b^5*c*d^7 - 63*A^3*C*b^8*c^4*d^4 - 54*A^3*C*b^8*c^2*d^6
+ 9*A^3*C*b^8*c^6*d^2 + 9*A*C^3*b^8*c^6*d^2 - 3*A*C^3*b^8*c^4*d^4 - 28*A^3*
B*b^8*c^5*d^3 - 28*A*B^3*b^8*c^5*d^3 - 18*A^3*B*b^8*c^3*d^5 - 18*A*B^3*b^8*
c^3*d^5 - 10*B^3*C*a^5*b^3*d^8 - 10*B*C^3*a^5*b^3*d^8 - 4*B^3*C*a^3*b^5*d^8
 - 4*B*C^3*a^3*b^5*d^8 + 23*A^3*C*a^4*b^4*d^8 - 18*A^3*C*a^2*b^6*d^8 + 11*A
*C^3*a^4*b^4*d^8 - 9*A*C^3*a^6*b^2*d^8 + 6*A*C^3*a^2*b^6*d^8 - 3*A^3*C*a^6*
b^2*d^8 - 20*A^3*B*a^3*b^5*d^8 - 20*A*B^3*a^3*b^5*d^8 + 4*A^3*B*a^5*b^3*d^8
 + 4*A*B^3*a^5*b^3*d^8 + B^3*C*a^2*b^6*c^5*d^3 + B*C^3*a^2*b^6*c^5*d^3 + 6*
C^4*a*b^7*c^7*d + 4*B^4*a*b^7*c*d^7 - 12*A^4*a*b^7*c*d^7 - 3*B^3*C*b^8*c^7*
d - 3*B*C^3*b^8*c^7*d - 6*A^3*B*b^8*c*d^7 - 6*A*B^3*b^8*c*d^7 - 12*A^3*B*a*
b^7*d^8 - 12*A*B^3*a*b^7*d^8 + 30*C^4*a^5*b^3*c^3*d^5 + 19*C^4*a^2*b^6*c^4*
d^4 - 9*C^4*a^6*b^2*c^2*d^6 + 9*C^4*a^4*b^4*c^6*d^2 + 4*C^4*a^3*b^5*c^3*d^5
 + 4*C^4*a^2*b^6*c^2*d^6 - 3*C^4*a^4*b^4*c^4*d^4 - 3*C^4*a^4*b^4*c^2*d^6 +
3*C^4*a^2*b^6*c^6*d^2 + 28*B^4*a^3*b^5*c^5*d^3 + 27*B^4*a^4*b^4*c^2*d^6 - 1
7*B^4*a^4*b^4*c^4*d^4 - 10*B^4*a^2*b^6*c^4*d^4 + 8*B^4*a^3*b^5*c^3*d^5 + 8*
B^4*a^2*b^6*c^2*d^6 - 6*B^4*a^2*b^6*c^6*d^2 + 4*B^4*a^5*b^3*c^3*d^5 + 70*A^
4*a^2*b^6*c^4*d^4 + 58*A^4*a^2*b^6*c^2*d^6 - 56*A^4*a^3*b^5*c^3*d^5 + 15*A^
4*a^4*b^4*c^2*d^6 + B^2*C^2*b^8*c^2*d^6 - 18*A^3*C*b^8*d^8 + B^3*C*b^8*c^5*

```
d^3 + B*C^3*b^8*c^5*d^3 + 6*B^4*b^8*c^6*d^2 + 3*B^4*b^8*c^4*d^4 + 30*A^4*b^
8*c^4*d^4 + 27*A^4*b^8*c^2*d^6 + 3*C^4*a^6*b^2*d^8 + 8*B^4*a^4*b^4*d^8 + 4*
B^4*a^2*b^6*d^8 + 12*A^4*a^2*b^6*d^8 - 5*A^4*a^4*b^4*d^8 + 9*A^2*C^2*b^8*d^
8 + 9*A^2*B^2*b^8*d^8 + 9*A^4*b^8*d^8 + B^4*b^8*c^2*d^6 + C^4*a^4*b^4*d^8,
f, k), k, 1, 4))/f
```

3.90 $\int (a+b\tan(e+fx))^3\sqrt{c+d\tan(e+fx)}(A+B\tan(e$

Optimal result 1107
Rubi [A] (verified) 1108
Mathematica [B] (verified) 1113
Maple [B] (verified) 1114
Fricas [B] (verification not implemented) 1117
Sympy [F] 1117
Maxima [F(-1)] 1118
Giac [F(-1)] 1118
Mupad [F(-1)] 1118

Optimal result

Integrand size = 47, antiderivative size = 464

$$\int (a+b\tan(e+fx))^3\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=-\frac{(a-ib)^3(iA+B-iC)\sqrt{c-id}\,\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f}$$
$$+\frac{(a+ib)^3(iA-B-iC)\sqrt{c+id}\,\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f}$$
$$+\frac{2(a^3B-3ab^2B+3a^2b(A-C)-b^3(A-C))\sqrt{c+d\tan(e+fx)}}{f}$$
$$+\frac{2(40a^3Cd^3-6a^2bd^2(16cC-45Bd)+9ab^2d(8c^2C-14Bcd+35(A-C)d^2)-b^3(16c^3C-24Bc^2d+4}{315d^4f}$$
$$+\frac{2b(21b(Ab+aB-bC)d^2+4(bc-ad)(2bcC-3bBd-2aCd))\tan(e+fx)(c+d\tan(e+fx))^{3/2}}{105d^3f}$$
$$-\frac{2(2bcC-3bBd-2aCd)(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}}{21d^2f}$$
$$+\frac{2C(a+b\tan(e+fx))^3(c+d\tan(e+fx))^{3/2}}{9df}$$

```
[Out] -(a-I*b)^3*(I*A+B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))*(c-I*d
)^(1/2)/f+(a+I*b)^3*(I*A-B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2
))*(c+I*d)^(1/2)/f+2*(B*a^3-3*B*a*b^2+3*a^2*b*(A-C)-b^3*(A-C))*(c+d*tan(f*x
+e))^(1/2)/f+2/315*(40*a^3*C*d^3-6*a^2*b*d^2*(-45*B*d+16*C*c)+9*a*b^2*d*(8*
c^2*C-14*B*c*d+35*(A-C)*d^2)-b^3*(16*c^3*C-24*B*c^2*d+42*c*(A-C)*d^2+105*B*
d^3))*(c+d*tan(f*x+e))^(3/2)/d^4/f+2/105*b*(21*b*(A*b+B*a-C*b)*d^2+4*(-a*d+
b*c)*(-3*B*b*d-2*C*a*d+2*C*b*c))*tan(f*x+e)*(c+d*tan(f*x+e))^(3/2)/d^3/f-2/
21*(-3*B*b*d-2*C*a*d+2*C*b*c)*(a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^(3/2)/d^2
/f+2/9*C*(a+b*tan(f*x+e))^3*(c+d*tan(f*x+e))^(3/2)/d/f
```

Rubi [A] (verified)

Time = 2.70 (sec) , antiderivative size = 464, normalized size of antiderivative = 1.00, number of steps used = 12, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}} = 0.170$, Rules used = {3728, 3718, 3711, 3609, 3620, 3618, 65, 214}

$$\int (a+b\tan(e+fx))^3\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$

$$= \frac{2(c+d\tan(e+fx))^{3/2}\left(40a^3Cd^3-6a^2bd^2(16cC-45Bd)+9ab^2d(35d^2(A-C)-14Bcd+8c^2C)-(b^3(4\right.}{315d^4f}$$

$$+\frac{2(a^3B+3a^2b(A-C)-3ab^2B-b^3(A-C))\sqrt{c+d\tan(e+fx)}}{f}$$

$$-\frac{(a-ib)^3\sqrt{c-id}(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f}$$

$$+\frac{(a+ib)^3\sqrt{c+id}(iA-B-iC)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f}$$

$$+\frac{2b\tan(e+fx)(c+d\tan(e+fx))^{3/2}\left(21bd^2(aB+Ab-bC)+4(bc-ad)(-2aCd-3bBd+2bcC)\right)}{105d^3f}$$

$$-\frac{2(-2aCd-3bBd+2bcC)(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}}{21d^2f}$$

$$+\frac{2C(a+b\tan(e+fx))^3(c+d\tan(e+fx))^{3/2}}{9df}$$

```
[In] Int[(a + b*Tan[e + f*x])^3*Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C
*Tan[e + f*x]^2),x]
```

```
[Out] -(((a - I*b)^3*(I*A + B - I*C)*Sqrt[c - I*d]*ArcTanh[Sqrt[c + d*Tan[e + f*x
]]/Sqrt[c - I*d]])/f) + ((a + I*b)^3*(I*A - B - I*C)*Sqrt[c + I*d]*ArcTanh[
Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]])/f + (2*(a^3*B - 3*a*b^2*B + 3*a^2*
b*(A - C) - b^3*(A - C))*Sqrt[c + d*Tan[e + f*x]])/f + (2*(40*a^3*C*d^3 - 6
*a^2*b*d^2*(16*c*C - 45*B*d) + 9*a*b^2*d*(8*c^2*C - 14*B*c*d + 35*(A - C)*d
^2) - b^3*(16*c^3*C - 24*B*c^2*d + 42*c*(A - C)*d^2 + 105*B*d^3))*(c + d*Ta
n[e + f*x])^(3/2))/(315*d^4*f) + (2*b*(21*b*(A*b + a*B - b*C)*d^2 + 4*(b*c
- a*d)*(2*b*c*C - 3*b*B*d - 2*a*C*d))*Tan[e + f*x]*(c + d*Tan[e + f*x])^(3/
2))/(105*d^3*f) - (2*(2*b*c*C - 3*b*B*d - 2*a*C*d)*(a + b*Tan[e + f*x])^2*(
c + d*Tan[e + f*x])^(3/2))/(21*d^2*f) + (2*C*(a + b*Tan[e + f*x])^3*(c + d*
Tan[e + f*x])^(3/2))/(9*d*f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
```

```
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3609

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[d*((a + b*Tan[e + f*x])^m/(f*m)), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
```

```
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\text{integral} = \frac{2C(a+b\tan(e+fx))^3(c+d\tan(e+fx))^{3/2}}{9df} + \frac{2\int (a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}\left(-\frac{3}{2}(2bcC-a(3A-C)d)+\frac{9}{2}(Ab+aB-bC)d\tan(e+fx)\right.}{9d}$$

$$= -\frac{2(2bcC-3bBd-2aCd)(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}}{21d^2f} + \frac{2C(a+b\tan(e+fx))^3(c+d\tan(e+fx))^{3/2}}{9df} + \frac{4\int (a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}\left(\frac{3}{4}(a^2(21A-13C)d^2+4b^2c(2cC-3Bd)-abd(16cC\right.}{}$$

$$= \frac{2b(21b(Ab+aB-bC)d^2+4(bc-ad)(2bcC-3bBd-2aCd))\tan(e+fx)(c+d\tan(e+fx))^{3/2}}{105d^3f} - \frac{2(2bcC-3bBd-2aCd)(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}}{21d^2f} + \frac{2C(a+b\tan(e+fx))^3(c+d\tan(e+fx))^{3/2}}{9df} - \frac{8\int \sqrt{c+d\tan(e+fx)}\left(-\frac{3}{8}(5a^3(21A-13C)d^3+18ab^2cd(4cC-7Bd)-3a^2bd^2(32cC+15Bd)\right.}{}$$

$$= \frac{2(40a^3Cd^3 - 6a^2bd^2(16cC - 45Bd) + 9ab^2d(8c^2C - 14Bcd + 35(A-C)d^2) - b^3(16c^3C - 24Bc^2}{315d^4f}$$
$$+ \frac{2b(21b(Ab + aB - bC)d^2 + 4(bc - ad)(2bcC - 3bBd - 2aCd))\tan(e+fx)(c + d\tan(e+fx))}{105d^3f}$$
$$- \frac{2(2bcC - 3bBd - 2aCd)(a + b\tan(e+fx))^2(c + d\tan(e+fx))^{3/2}}{21d^2f}$$
$$+ \frac{2C(a + b\tan(e+fx))^3(c + d\tan(e+fx))^{3/2}}{9df}$$
$$- \frac{8\int \sqrt{c + d\tan(e+fx)}\left(\frac{315}{8}(3a^2bB - b^3B - a^3(A-C) + 3ab^2(A-C))\,d^3 - \frac{315}{8}(a^3B - 3ab^2B\right.}{315d^3}$$

$$= \frac{2(a^3B - 3ab^2B + 3a^2b(A-C) - b^3(A-C))\sqrt{c + d\tan(e+fx)}}{f}$$
$$+ \frac{2(40a^3Cd^3 - 6a^2bd^2(16cC - 45Bd) + 9ab^2d(8c^2C - 14Bcd + 35(A-C)d^2) - b^3(16c^3C - 24B}{315d^4f}$$
$$+ \frac{2b(21b(Ab + aB - bC)d^2 + 4(bc - ad)(2bcC - 3bBd - 2aCd))\tan(e+fx)(c + d\tan(e+fx))}{105d^3f}$$
$$- \frac{2(2bcC - 3bBd - 2aCd)(a + b\tan(e+fx))^2(c + d\tan(e+fx))^{3/2}}{21d^2f}$$
$$+ \frac{2C(a + b\tan(e+fx))^3(c + d\tan(e+fx))^{3/2}}{9df}$$
$$- \frac{8\int \frac{-\frac{315}{8}d^3(a^3(Ac - cC - Bd) - 3ab^2(Ac - cC - Bd) - 3a^2b(Bc + (A-C)d) + b^3(Bc + (A-C)d)) - \frac{315}{8}d^3(3a^2b(Ac - cC - Bd) - b^3(Ac - c}{\sqrt{c + d\tan(e+fx)}}}{315d^3}$$

$$= \frac{2(a^3B - 3ab^2B + 3a^2b(A-C) - b^3(A-C))\sqrt{c + d\tan(e+fx)}}{f}$$
$$+ \frac{2(40a^3Cd^3 - 6a^2bd^2(16cC - 45Bd) + 9ab^2d(8c^2C - 14Bcd + 35(A-C)d^2) - b^3(16c^3C - 24B}{315d^4f}$$
$$+ \frac{2b(21b(Ab + aB - bC)d^2 + 4(bc - ad)(2bcC - 3bBd - 2aCd))\tan(e+fx)(c + d\tan(e+fx))}{105d^3f}$$
$$- \frac{2(2bcC - 3bBd - 2aCd)(a + b\tan(e+fx))^2(c + d\tan(e+fx))^{3/2}}{21d^2f}$$
$$+ \frac{2C(a + b\tan(e+fx))^3(c + d\tan(e+fx))^{3/2}}{9df}$$
$$+ \frac{1}{2}\left((a-ib)^3(A-iB-C)(c-id)\right)\int \frac{1 + i\tan(e+fx)}{\sqrt{c + d\tan(e+fx)}}\,dx + \frac{1}{2}\left((a+ib)^3(A+iB-C)(c+id)\right)\int$$

$$= \frac{2(a^3B - 3ab^2B + 3a^2b(A-C) - b^3(A-C))\sqrt{c+d\tan(e+fx)}}{f}$$
$$+ \frac{2(40a^3Cd^3 - 6a^2bd^2(16cC - 45Bd) + 9ab^2d(8c^2C - 14Bcd + 35(A-C)d^2) - b^3(16c^3C - 24Bc}{315d^4f}$$
$$+ \frac{2b(21b(Ab + aB - bC)d^2 + 4(bc-ad)(2bcC - 3bBd - 2aCd))\tan(e+fx)(c+d\tan(e+fx))^{3/}}{105d^3f}$$
$$- \frac{2(2bcC - 3bBd - 2aCd)(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}}{21d^2f}$$
$$+ \frac{2C(a+b\tan(e+fx))^3(c+d\tan(e+fx))^{3/2}}{9df}$$
$$- \frac{(i(a+ib)^3(A+iB-C)(c+id))\operatorname{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c+idx}}\,dx, x, -i\tan(e+fx)\right)}{2f}$$
$$+ \frac{((a-ib)^3(A-iB-C)(ic+d))\operatorname{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c-idx}}\,dx, x, i\tan(e+fx)\right)}{2f}$$

$$= \frac{2(a^3B - 3ab^2B + 3a^2b(A-C) - b^3(A-C))\sqrt{c+d\tan(e+fx)}}{f}$$
$$+ \frac{2(40a^3Cd^3 - 6a^2bd^2(16cC - 45Bd) + 9ab^2d(8c^2C - 14Bcd + 35(A-C)d^2) - b^3(16c^3C - 24Bc}{315d^4f}$$
$$+ \frac{2b(21b(Ab + aB - bC)d^2 + 4(bc-ad)(2bcC - 3bBd - 2aCd))\tan(e+fx)(c+d\tan(e+fx))^{3/}}{105d^3f}$$
$$- \frac{2(2bcC - 3bBd - 2aCd)(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}}{21d^2f}$$
$$+ \frac{2C(a+b\tan(e+fx))^3(c+d\tan(e+fx))^{3/2}}{9df}$$
$$- \frac{((a+ib)^3(A+iB-C)(c+id))\operatorname{Subst}\left(\int \frac{1}{-1+\frac{ic}{d}-\frac{ix^2}{d}}\,dx, x, \sqrt{c+d\tan(e+fx)}\right)}{df}$$
$$- \frac{((ia+b)^3(A-iB-C)(ic+d))\operatorname{Subst}\left(\int \frac{1}{-1-\frac{ic}{d}+\frac{ix^2}{d}}\,dx, x, \sqrt{c+d\tan(e+fx)}\right)}{df}$$

$$= -\frac{(a-ib)^3(iA+B-iC)\sqrt{c-id}\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f}$$
$$-\frac{(ia-b)^3(A+iB-C)\sqrt{c+id}\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f}$$
$$+\frac{2(a^3B-3ab^2B+3a^2b(A-C)-b^3(A-C))\sqrt{c+d\tan(e+fx)}}{f}$$
$$+\frac{2(40a^3Cd^3-6a^2bd^2(16cC-45Bd)+9ab^2d(8c^2C-14Bcd+35(A-C)d^2)-b^3(16c^3C-24B}{315d^4f}$$
$$+\frac{2b(21b(Ab+aB-bC)d^2+4(bc-ad)(2bcC-3bBd-2aCd))\tan(e+fx)(c+d\tan(e+fx))}{105d^3f}$$
$$-\frac{2(2bcC-3bBd-2aCd)(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}}{21d^2f}$$
$$+\frac{2C(a+b\tan(e+fx))^3(c+d\tan(e+fx))^{3/2}}{9df}$$

Mathematica [B] (verified)

Both result and optimal contain complex but leaf count is larger than twice the leaf count of optimal. 1232 vs. 2(464) = 928.

Time = 6.55 (sec) , antiderivative size = 1232, normalized size of antiderivative = 2.66

$$\int (a+b\tan(e+fx))^3\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\frac{2C(a+b\tan(e+fx))^3(c+d\tan(e+fx))^{3/2}}{9df}$$
$$+\frac{2\left(-\frac{3(2bcC-3bBd-2aCd)(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}}{7df}+\frac{2\left(\frac{3b\left(21b(Ab+aB-bC)d^2+4(bc-ad)(2bcC-3bBd-2aCd)\right)\tan(e+fx)(c+d}{10df}\right.}{}\right.}{}$$

```
[In] Integrate[(a + b*Tan[e + f*x])^3*Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*
x] + C*Tan[e + f*x]^2),x]
```

```
[Out] (2*C*(a + b*Tan[e + f*x])^3*(c + d*Tan[e + f*x])^(3/2))/(9*d*f) + (2*((-3*(
2*b*c*C - 3*b*B*d - 2*a*C*d)*(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])^(3
/2))/(7*d*f) + (2*((3*b*(21*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(2*b*c*
C - 3*b*B*d - 2*a*C*d))*Tan[e + f*x]*(c + d*Tan[e + f*x])^(3/2))/(10*d*f) -
 (2*((2*((-15*a*d*(21*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(2*b*c*C - 3*
b*B*d - 2*a*C*d)))/8 + b*((-315*(a^2*B - b^2*B + 2*a*b*(A - C))*d^3)/8 + (3
*c*(21*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(2*b*c*C - 3*b*B*d - 2*a*C*d
)))/4))*(c + d*Tan[e + f*x])^(3/2))/(3*d*f) + ((I/2)*((-15*a*d*(a^2*(21*A -
 13*C)*d^2 + 4*b^2*c*(2*c*C - 3*B*d) - a*b*d*(16*c*C + 9*B*d)))/8 + (3*b*c*
(21*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(2*b*c*C - 3*b*B*d - 2*a*C*d)))
/4 + (15*a*d*(21*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(2*b*c*C - 3*b*B*d
 - 2*a*C*d)))/8 + ((5*I)/2)*d*((63*a*(a^2*B - b^2*B + 2*a*b*(A - C))*d^2)/4
 + (3*b*(a^2*(21*A - 13*C)*d^2 + 4*b^2*c*(2*c*C - 3*B*d) - a*b*d*(16*c*C +
9*B*d)))/4 - (3*b*(21*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(2*b*c*C - 3*
b*B*d - 2*a*C*d)))/4) - b*((-315*(a^2*B - b^2*B + 2*a*b*(A - C))*d^3)/8 + (
3*c*(21*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(2*b*c*C - 3*b*B*d - 2*a*C*
d)))/4))*((2*(c - I*d)^(3/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]
])/(-c + I*d) + 2*Sqrt[c + d*Tan[e + f*x]]))/f - ((I/2)*((-15*a*d*(a^2*(21*
A - 13*C)*d^2 + 4*b^2*c*(2*c*C - 3*B*d) - a*b*d*(16*c*C + 9*B*d)))/8 + (3*b
*c*(21*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(2*b*c*C - 3*b*B*d - 2*a*C*d
)))/4 + (15*a*d*(21*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(2*b*c*C - 3*b*
B*d - 2*a*C*d)))/8 - ((5*I)/2)*d*((63*a*(a^2*B - b^2*B + 2*a*b*(A - C))*d^2
)/4 + (3*b*(a^2*(21*A - 13*C)*d^2 + 4*b^2*c*(2*c*C - 3*B*d) - a*b*d*(16*c*C
 + 9*B*d)))/4 - (3*b*(21*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(2*b*c*C -
 3*b*B*d - 2*a*C*d)))/4) - b*((-315*(a^2*B - b^2*B + 2*a*b*(A - C))*d^3)/8
+ (3*c*(21*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(2*b*c*C - 3*b*B*d - 2*a
*C*d)))/4))*((2*(c + I*d)^(3/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I
*d]])/(-c - I*d) + 2*Sqrt[c + d*Tan[e + f*x]]))/f))/(5*d)))/(7*d)))/(9*d)
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 4472 vs. $2(424) = 848$.

Time = 0.53 (sec) , antiderivative size = 4473, normalized size of antiderivative = 9.64

method	result	size
parts	Expression too large to display	4473
derivativedivides	Expression too large to display	6661
default	Expression too large to display	6661

```
[In] int((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x,method=_RETURNVERBOSE)
```

[Out] 1/f*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*A*a^3+1/f*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*b^3-1/f*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*A*a^3-1/f*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*b^3+1/f*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*C*a^3-1/f*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*C*a^3+2/f/d*A*a*b^2*(c+d*tan(f*x+e))^(3/2)+2/f/d*B*a^2*b*(c+d*tan(f*x+e))^(3/2)-12/5/f/d^3*C*a*b^2*c*(c+d*tan(f*x+e))^(5/2)-1/4/f/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*a^3+1/4/f/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*a^3*c-1/4/f/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*b^3+1/4/f/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*b^3*c+1/4/f/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*a^3-1/4/f/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*a^3*c+2/f/d^3*C*a*b^2*c^2*(c+d*tan(f*x+e))^(3/2)+3/f*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*C*a*b^2+3/f*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*A*a*b^2+3/f*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*a^2*b-3/f*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*C*a*b^2-3/f*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*A*a*b^2-3/f*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*a^2*b+1/4/f/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*a^3-1/4/f/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*a^3*c+1/4/f/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*b^3-1/4/f/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*b^3*c-1/4/f/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*a^3+

1/4/f/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e
)-c-(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*a^3*c+1/f*a^2*(3*A*b+B
a)(2*(c+d*tan(f*x+e))^(1/2)+1/4*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*ln((c+d*tan
(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2)
)+((c^2+d^2)^(1/2)-c)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1
/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))-1/4
(2(c^2+d^2)^(1/2)+2*c)^(1/2)*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*
(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))+(-(c^2+d^2)^(1/2)+c)/(2*(c^2+d^
2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c
)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)))+2*C*b^3/f/d^4*(1/9*(c+d*tan(f*x+e)
)^(9/2)-3/7*c*(c+d*tan(f*x+e))^(7/2)+3/5*c^2*(c+d*tan(f*x+e))^(5/2)-1/5*(c+
d*tan(f*x+e))^(5/2)*d^2-1/3*c^3*(c+d*tan(f*x+e))^(3/2)+1/3*c*d^2*(c+d*tan(f
*x+e))^(3/2)+(c+d*tan(f*x+e))^(1/2)*d^4+d^4*(1/8*(2*(c^2+d^2)^(1/2)+2*c)^(1
/2)*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-
(c^2+d^2)^(1/2))+1/2*((c^2+d^2)^(1/2)-c)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arct
an(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1
/2)-2*c)^(1/2))-1/8*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*ln(d*tan(f*x+e)+c+(c+d*ta
n(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))+1/2*(-(c^2+d
^2)^(1/2)+c)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)
+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))))-3/4/f/d*ln
(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d
^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*a*b^2*c+3/4/f/d*ln(d*tan(f*x+e)+
c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*B*(
2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*a^2*b-3/4/f/d*ln(d*tan(f*x+e)+
c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*B*(
2*(c^2+d^2)^(1/2)+2*c)^(1/2)*a^2*b*c-3/4/f/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x
+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1
/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*a*b^2+3/4/f/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x
+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1
/2)+2*c)^(1/2)*a*b^2*c-3/4/f/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)
+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)
*(c^2+d^2)^(1/2)*a*b^2+3/4/f/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)
+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)
*a*b^2*c-3/4/f/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*
tan(f*x+e)-c-(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/
2)*a^2*b+3/4/f/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*
tan(f*x+e)-c-(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*a^2*b*c+3/4/f
/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(
c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*a*b^2-3/4/f
/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(
c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*a*b^2*c+3/4/f/d*ln(d*tan(f*
x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2)
)*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*a*b^2+2/3/f/d^3*B*b^3*c^2
*(c+d*tan(f*x+e))^(3/2)+6/7/f/d^3*C*a*b^2*(c+d*tan(f*x+e))^(7/2)-4/5/f/d^3*
B*b^3*c*(c+d*tan(f*x+e))^(5/2)-2/f/d*C*a*b^2*(c+d*tan(f*x+e))^(3/2)+2/f*b*(

```
A*b^2+3*B*a*b+3*C*a^2)/d^2*(1/5*(c+d*tan(f*x+e))^(5/2)-1/3*(c+d*tan(f*x+e))
^(3/2)*c-(c+d*tan(f*x+e))^(1/2)*d^2-d^2*(1/8*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*
ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2
+d^2)^(1/2))+1/2*((c^2+d^2)^(1/2)-c)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((
(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-
2*c)^(1/2))-1/8*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*ln(d*tan(f*x+e)+c+(c+d*tan(f*
x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))+1/2*(-(c^2+d^2)^
(1/2)+c)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*
(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))))+2/7/f/d^3*B*b^
3*(c+d*tan(f*x+e))^(7/2)-2/3/f/d*B*b^3*(c+d*tan(f*x+e))^(3/2)+2/3/f/d*C*a^3
*(c+d*tan(f*x+e))^(3/2)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 35153 vs. $2(414) = 828$.

Time = 10.87 (sec) , antiderivative size = 35153, normalized size of antiderivative = 75.76

$$\int (a+b\tan(e+fx))^3\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= \text{Too large to display}$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f
*x+e)^2),x, algorithm="fricas")
```

```
[Out] Too large to include
```

Sympy [F]

$$\int (a+b\tan(e+fx))^3\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= \int (a+b\tan{(e+fx)})^3\sqrt{c+d\tan{(e+fx)}}\left(A+B\tan{(e+fx)}+C\tan^2{(e+fx)}\right)\,dx$$

```
[In] integrate((c+d*tan(f*x+e))**(1/2)*(a+b*tan(f*x+e))**3*(A+B*tan(f*x+e)+C*tan
(f*x+e)**2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**3*sqrt(c + d*tan(e + f*x))*(A + B*tan(e + f*
x) + C*tan(e + f*x)**2), x)
```

Maxima [F(-1)]

Timed out.

$$\int (a+b\tan(e+fx))^3\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f
*x+e)^2),x, algorithm="maxima")
```

```
[Out] Timed out
```

Giac [F(-1)]

Timed out.

$$\int (a+b\tan(e+fx))^3\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f
*x+e)^2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int (a+b\tan(e+fx))^3\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\text{Hanged}$$

```
[In] int((a + b*tan(e + f*x))^3*(c + d*tan(e + f*x))^(1/2)*(A + B*tan(e + f*x) +
 C*tan(e + f*x)^2),x)
```

```
[Out] \text{Hanged}
```

3.91 $\int (a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}(A+B\tan(e$

Optimal result 1119
Rubi [A] (verified) 1120
Mathematica [A] (verified) 1124
Maple [B] (verified) 1125
Fricas [B] (verification not implemented) 1127
Sympy [F] 1127
Maxima [F] 1127
Giac [F(-1)] 1128
Mupad [F(-1)] 1128

Optimal result

Integrand size = 47, antiderivative size = 325

$$\begin{aligned}
&\int (a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx\\
&=-\frac{(a-ib)^2(B+i(A-C))\sqrt{c-id}\,\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f}\\
&\quad-\frac{(a+ib)^2(B-i(A-C))\sqrt{c+id}\,\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f}\\
&\quad+\frac{2(a^2B-b^2B+2ab(A-C))\sqrt{c+d\tan(e+fx)}}{f}\\
&\quad+\frac{2(20a^2Cd^2-14abd(2cC-5Bd)+b^2(8c^2C-14Bcd+35(A-C)d^2))(c+d\tan(e+fx))^{3/2}}{105d^3f}\\
&\quad-\frac{2b(4bcC-7bBd-4aCd)\tan(e+fx)(c+d\tan(e+fx))^{3/2}}{35d^2f}\\
&\quad+\frac{2C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}}{7df}
\end{aligned}$$

```
[Out] -(a-I*b)^2*(B+I*(A-C))*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))*(c-I*d
)^(1/2)/f-(a+I*b)^2*(B-I*(A-C))*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2
))*(c+I*d)^(1/2)/f+2*(B*a^2-B*b^2+2*a*b*(A-C))*(c+d*tan(f*x+e))^(1/2)/f+2/1
05*(20*a^2*C*d^2-14*a*b*d*(-5*B*d+2*C*c)+b^2*(8*c^2*C-14*B*c*d+35*(A-C)*d^2
))*(c+d*tan(f*x+e))^(3/2)/d^3/f-2/35*b*(-7*B*b*d-4*C*a*d+4*C*b*c)*tan(f*x+e
)*(c+d*tan(f*x+e))^(3/2)/d^2/f+2/7*C*(a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^(3
/2)/d/f
```

Rubi [A] (verified)

Time = 1.46 (sec) , antiderivative size = 325, normalized size of antiderivative = 1.00, number of steps used = 11, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.170, Rules used = {3728, 3718, 3711, 3609, 3620, 3618, 65, 214}

$$\int (a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\frac{2(c+d\tan(e+fx))^{3/2}\left(20a^2Cd^2-14abd(2cC-5Bd)+b^2(35d^2(A-C)-14Bcd+8c^2C)\right)}{105d^3f}$$
$$+\frac{2(a^2B+2ab(A-C)-b^2B)\sqrt{c+d\tan(e+fx)}}{f}$$
$$-\frac{(a-ib)^2\sqrt{c-id}(B+i(A-C))\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f}$$
$$-\frac{(a+ib)^2\sqrt{c+id}(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f}$$
$$-\frac{2b\tan(e+fx)(-4aCd-7bBd+4bcC)(c+d\tan(e+fx))^{3/2}}{35d^2f}$$
$$+\frac{2C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}}{7df}$$

```
[In] Int[(a + b*Tan[e + f*x])^2*Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C
*Tan[e + f*x]^2),x]
```

```
[Out] -(((a - I*b)^2*(B + I*(A - C))*Sqrt[c - I*d]*ArcTanh[Sqrt[c + d*Tan[e + f*x
]]/Sqrt[c - I*d]])/f) - ((a + I*b)^2*(B - I*(A - C))*Sqrt[c + I*d]*ArcTanh[
Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]])/f + (2*(a^2*B - b^2*B + 2*a*b*(A -
 C))*Sqrt[c + d*Tan[e + f*x]])/f + (2*(20*a^2*C*d^2 - 14*a*b*d*(2*c*C - 5*B
*d) + b^2*(8*c^2*C - 14*B*c*d + 35*(A - C)*d^2))*(c + d*Tan[e + f*x])^(3/2)
)/(105*d^3*f) - (2*b*(4*b*c*C - 7*b*B*d - 4*a*C*d)*Tan[e + f*x]*(c + d*Tan[
e + f*x])^(3/2))/(35*d^2*f) + (2*C*(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*
x])^(3/2))/(7*d*f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3609

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[d*((a + b*Tan[e + f*x])^m/(f*m)), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\text{integral} = \frac{2C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}}{7df} + \frac{2\int(a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}\left(\frac{1}{2}(-4bcC+a(7A-3C)d)+\frac{7}{2}(Ab+aB-bC)d\tan(e+fx)\right.}{7d}$$

$$= -\frac{2b(4bcC-7bBd-4aCd)\tan(e+fx)(c+d\tan(e+fx))^{3/2}}{35d^2f} + \frac{2C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}}{7df} - \frac{4\int\sqrt{c+d\tan(e+fx)}\left(\frac{1}{4}\left(28abcCd-5a^2(7A-3C)d^2-4b^2\left(2c^2C-\frac{7Bcd}{2}\right)\right)-\frac{35}{4}(a^2B-b^2B+\right.}{}$$

$$= \frac{2(20a^2Cd^2-14abd(2cC-5Bd)+b^2(8c^2C-14Bcd+35(A-C)d^2))(c+d\tan(e+fx))^{3/2}}{105d^3f} - \frac{2b(4bcC-7bBd-4aCd)\tan(e+fx)(c+d\tan(e+fx))^{3/2}}{35d^2f} + \frac{2C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}}{7df} - \frac{4\int\sqrt{c+d\tan(e+fx)}\left(\frac{35}{4}(2abB-a^2(A-C)+b^2(A-C))d^2-\frac{35}{4}(a^2B-b^2B+2ab(A-C))\right.}{35d^2}$$

$$= \frac{2(a^2B-b^2B+2ab(A-C))\sqrt{c+d\tan(e+fx)}}{f} + \frac{2(20a^2Cd^2-14abd(2cC-5Bd)+b^2(8c^2C-14Bcd+35(A-C)d^2))(c+d\tan(e+fx))^{3/2}}{105d^3f} - \frac{2b(4bcC-7bBd-4aCd)\tan(e+fx)(c+d\tan(e+fx))^{3/2}}{35d^2f} + \frac{2C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}}{7df} - \frac{4\int\frac{-\frac{35}{4}d^2(a^2(Ac-cC-Bd)-b^2(Ac-cC-Bd)-2ab(Bc+(A-C)d))-\frac{35}{4}d^2(2ab(Ac-cC-Bd)+a^2(Bc+(A-C)d)-b^2(Bc+(A-C)d}{\sqrt{c+d\tan(e+fx)}}}{35d^2}$$

$$= \frac{2(a^2B - b^2B + 2ab(A - C))\sqrt{c + d\tan(e + fx)}}{f} + \frac{2(20a^2Cd^2 - 14abd(2cC - 5Bd) + b^2(8c^2C - 14Bcd + 35(A - C)d^2))(c + d\tan(e + fx))^{3/2}}{105d^3f} - \frac{2b(4bcC - 7bBd - 4aCd)\tan(e + fx)(c + d\tan(e + fx))^{3/2}}{35d^2f} + \frac{2C(a + b\tan(e + fx))^2(c + d\tan(e + fx))^{3/2}}{7df} + \frac{1}{2}((a-ib)^2(A-iB-C)(c-id))\int\frac{1 + i\tan(e + fx)}{\sqrt{c + d\tan(e + fx)}}dx + \frac{1}{2}((a+ib)^2(A+iB-C)(c+id))\int$$

$$= \frac{2(a^2B - b^2B + 2ab(A - C))\sqrt{c + d\tan(e + fx)}}{f} + \frac{2(20a^2Cd^2 - 14abd(2cC - 5Bd) + b^2(8c^2C - 14Bcd + 35(A - C)d^2))(c + d\tan(e + fx))^{3/2}}{105d^3f} - \frac{2b(4bcC - 7bBd - 4aCd)\tan(e + fx)(c + d\tan(e + fx))^{3/2}}{35d^2f} + \frac{2C(a + b\tan(e + fx))^2(c + d\tan(e + fx))^{3/2}}{7df} - \frac{(i(a + ib)^2(A + iB - C)(c + id))\,\text{Subst}\left(\int\frac{1}{(-1+x)\sqrt{c+idx}}dx, x, -i\tan(e + fx)\right)}{2f} + \frac{((a - ib)^2(A - iB - C)(ic + d))\,\text{Subst}\left(\int\frac{1}{(-1+x)\sqrt{c-idx}}dx, x, i\tan(e + fx)\right)}{2f}$$

$$= \frac{2(a^2B - b^2B + 2ab(A - C))\sqrt{c + d\tan(e + fx)}}{f} + \frac{2(20a^2Cd^2 - 14abd(2cC - 5Bd) + b^2(8c^2C - 14Bcd + 35(A - C)d^2))(c + d\tan(e + fx))^{3/2}}{105d^3f} - \frac{2b(4bcC - 7bBd - 4aCd)\tan(e + fx)(c + d\tan(e + fx))^{3/2}}{35d^2f} + \frac{2C(a + b\tan(e + fx))^2(c + d\tan(e + fx))^{3/2}}{7df} - \frac{((a + ib)^2(A + iB - C)(c + id))\,\text{Subst}\left(\int\frac{1}{-1+\frac{ic}{d}-\frac{ix^2}{d}}dx, x, \sqrt{c + d\tan(e + fx)}\right)}{df} + \frac{((a - ib)^2(iA + B - iC)(ic + d))\,\text{Subst}\left(\int\frac{1}{-1-\frac{ic}{d}+\frac{ix^2}{d}}dx, x, \sqrt{c + d\tan(e + fx)}\right)}{df}$$

$$= -\frac{(a-ib)^2(B+i(A-C))\sqrt{c-id}\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f} - \frac{(a+ib)^2(B-i(A-C))\sqrt{c+id}\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f} + \frac{2(a^2B-b^2B+2ab(A-C))\sqrt{c+d\tan(e+fx)}}{f} + \frac{2(20a^2Cd^2-14abd(2cC-5Bd)+b^2(8c^2C-14Bcd+35(A-C)d^2))(c+d\tan(e+fx))^{3/2}}{105d^3f} - \frac{2b(4bcC-7bBd-4aCd)\tan(e+fx)(c+d\tan(e+fx))^{3/2}}{35d^2f} + \frac{2C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}}{7df}$$

Mathematica [A] (verified)

Time = 5.24 (sec) , antiderivative size = 314, normalized size of antiderivative = 0.97

$$\int (a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$

$$= \frac{2\Big((20a^2Cd^2+14abd(-2cC+5Bd)+b^2(8c^2C-14Bcd+35(A-C)d^2))(c+d\tan(e+fx))^{3/2}+3bd(-4}{}$$

```
[In] Integrate[(a + b*Tan[e + f*x])^2*Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*
x] + C*Tan[e + f*x]^2),x]
```

```
[Out] (2*((20*a^2*C*d^2 + 14*a*b*d*(-2*c*C + 5*B*d) + b^2*(8*c^2*C - 14*B*c*d + 3
5*(A - C)*d^2))*(c + d*Tan[e + f*x])^(3/2) + 3*b*d*(-4*b*c*C + 7*b*B*d + 4*
a*C*d)*Tan[e + f*x]*(c + d*Tan[e + f*x])^(3/2) + 15*C*d^2*(a + b*Tan[e + f*
x])^2*(c + d*Tan[e + f*x])^(3/2) + (105*(a - I*b)^2*(I*A + B - I*C)*d^3*(-(
Sqrt[c - I*d]*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]]) + Sqrt[c + d
*Tan[e + f*x]]))/2 + (105*(a + I*b)^2*((-I)*A + B + I*C)*d^3*(-(Sqrt[c + I*
d]*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]]) + Sqrt[c + d*Tan[e + f*
x]]))/2))/(105*d^3*f)
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 3352 vs. 2(291) = 582.

Time = 0.16 (sec) , antiderivative size = 3353, normalized size of antiderivative = 10.32

method	result	size
parts	Expression too large to display	3353
derivativedivides	Expression too large to display	4775
default	Expression too large to display	4775

```
[In] int((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x,method=_RETURNVERBOSE)
```

```
[Out] 4/3/f/d*B*a*b*(c+d*tan(f*x+e))^(3/2)-1/f*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*ar
ctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^
(1/2)-2*c)^(1/2))*A*a^2+1/f*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2
+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1
/2))*A*b^2+1/f*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2
*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*C*a^2+1/
f*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+
d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*A*a^2-1/f*d/(2*(c^2+d
^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*
c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*A*b^2-1/f*d/(2*(c^2+d^2)^(1/2)-2*c
)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*
(c^2+d^2)^(1/2)-2*c)^(1/2))*C*a^2+1/2/f/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e)
)^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)
+2*c)^(1/2)*(c^2+d^2)^(1/2)*a*b-1/2/f/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^
(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)+2
*c)^(1/2)*a*b*c-1/2/f/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(
1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d
^2)^(1/2)*a*b+1/2/f/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/
2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*a*b*c+2/
f*b*(B*b+2*C*a)/d^2*(1/5*(c+d*tan(f*x+e))^(5/2)-1/3*(c+d*tan(f*x+e))^(3/2)*
c-(c+d*tan(f*x+e))^(1/2)*d^2-d^2*(1/8*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*ln((c+d
*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(
1/2))+1/2*((c^2+d^2)^(1/2)-c)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2
+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1
/2))-1/8*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(
1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))+1/2*(-(c^2+d^2)^(1/2)+c
)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^
2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))))-1/4/f/d*ln(d*tan(f*x+
e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*
A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*a^2+1/4/f/d*ln(d*tan(f*x+e)
+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*A*
(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*b^2+1/4/f/d*ln(d*tan(f*x+e)+c
```

```
+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*A*(2
*(c^2+d^2)^(1/2)+2*c)^(1/2)*a^2*c-1/4/f/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e)
)^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)
+2*c)^(1/2)*b^2*c+1/4/f/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+
d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2
+d^2)^(1/2)*a^2-1/4/f/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^
2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*a^2*c-
2/f*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^
2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*a*b+2/f*d/(2*(c^2
+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x
+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*a*b+1/f*a*(2*A*b+B*a)*(2*(c+d*
tan(f*x+e))^(1/2)+1/4*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*ln((c+d*tan(f*x+e))^(1/
2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))+((c^2+d^2)
^(1/2)-c)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/
2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))-1/4*(2*(c^2+d^2
)^(1/2)+2*c)^(1/2)*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1
/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))+(-(c^2+d^2)^(1/2)+c)/(2*(c^2+d^2)^(1/2)-2*c
)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*
(c^2+d^2)^(1/2)-2*c)^(1/2)))+1/4/f/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)
^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)
^(1/2)*(c^2+d^2)^(1/2)*a^2-1/4/f/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(
1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(
1/2)*(c^2+d^2)^(1/2)*b^2-1/4/f/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/
2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/
2)*a^2*c+1/4/f/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*
tan(f*x+e)-c-(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*b^2*c-1/4/f/d
*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^
2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*a^2+1/4/f/d*l
n((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+
d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*a^2*c+2/3/f/d*A*b^2*(c+d*tan(f*
x+e))^(3/2)+2/3/f/d*C*a^2*(c+d*tan(f*x+e))^(3/2)+C*b^2*(2/7/f/d^3*(c+d*tan(
f*x+e))^(7/2)-4/5/f/d^3*c*(c+d*tan(f*x+e))^(5/2)+2/3/f/d^3*(c+d*tan(f*x+e))
^(3/2)*c^2-2/3/f/d*(c+d*tan(f*x+e))^(3/2)+1/4/f/d*(2*(c^2+d^2)^(1/2)+2*c)^(
1/2)*(c^2+d^2)^(1/2)*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2
)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))-1/f*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arcta
n(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/
2)-2*c)^(1/2))-1/4/f/d*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c*ln((c+d*tan(f*x+e))^
(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))-1/4/f/d
*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*ln(d*tan(f*x+e)+c+(c+d*tan(f
*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))+1/f*d/(2*(c^2+d
^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*
c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))+1/4/f/d*(2*(c^2+d^2)^(1/2)+2*c)^(1
/2)*c*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2
)+(c^2+d^2)^(1/2)))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 23984 vs. 2(281) = 562.

Time = 4.69 (sec) , antiderivative size = 23984, normalized size of antiderivative = 73.80

$$\int (a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= \text{Too large to display}$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f
*x+e)^2),x, algorithm="fricas")
```

```
[Out] Too large to include
```

Sympy [F]

$$\int (a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= \int (a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$

```
[In] integrate((c+d*tan(f*x+e))**(1/2)*(a+b*tan(f*x+e))**2*(A+B*tan(f*x+e)+C*tan
(f*x+e)**2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**2*sqrt(c + d*tan(e + f*x))*(A + B*tan(e + f*
x) + C*tan(e + f*x)**2), x)
```

Maxima [F]

$$\int (a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= \int (C\tan(fx+e)^2+B\tan(fx+e)+A)(b\tan(fx+e)+a)^2\sqrt{d\tan(fx+e)+c}\,dx$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f
*x+e)^2),x, algorithm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)^2*sq
rt(d*tan(f*x + e) + c), x)
```

Giac [F(-1)]

Timed out.

$$\int (a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f
*x+e)^2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int (a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \text{Hanged}$$

```
[In] int((a + b*tan(e + f*x))^2*(c + d*tan(e + f*x))^(1/2)*(A + B*tan(e + f*x) +
 C*tan(e + f*x)^2),x)
```

```
[Out] \text{Hanged}
```

3.92 $\int (a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}(A+B\tan(e+$

Optimal result 1129
Rubi [A] (verified) 1130
Mathematica [A] (verified) 1133
Maple [B] (verified) 1134
Fricas [B] (verification not implemented) 1135
Sympy [F] 1135
Maxima [F] 1136
Giac [F(-1)] 1136
Mupad [B] (verification not implemented) 1136

Optimal result

Integrand size = 45, antiderivative size = 224

$$\int (a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= -\frac{(ia+b)(A-iB-C)\sqrt{c-id}\,\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f} + \frac{(ia-b)(A+iB-C)\sqrt{c+id}\,\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f} + \frac{2(Ab+aB-bC)\sqrt{c+d\tan(e+fx)}}{f} - \frac{2(2bcC-5bBd-5aCd)(c+d\tan(e+fx))^{3/2}}{15d^2f} + \frac{2bC\tan(e+fx)(c+d\tan(e+fx))^{3/2}}{5df}$$

```
[Out] -(I*a+b)*(A-I*B-C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))*(c-I*d)^(1
/2)/f+(I*a-b)*(A+I*B-C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))*(c+I*
d)^(1/2)/f+2*(A*b+B*a-C*b)*(c+d*tan(f*x+e))^(1/2)/f-2/15*(-5*B*b*d-5*C*a*d+
2*C*b*c)*(c+d*tan(f*x+e))^(3/2)/d^2/f+2/5*b*C*tan(f*x+e)*(c+d*tan(f*x+e))^(
3/2)/d/f
```

Rubi [A] (verified)

Time = 0.70 (sec) , antiderivative size = 224, normalized size of antiderivative = 1.00, number of steps used = 10, number of rules used = 7, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.156, Rules used = {3718, 3711, 3609, 3620, 3618, 65, 214}

$$\int (a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= -\frac{(b+ia)\sqrt{c-id}(A-iB-C)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f} + \frac{(-b+ia)\sqrt{c+id}(A+iB-C)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f} + \frac{2(aB+Ab-bC)\sqrt{c+d\tan(e+fx)}}{f} - \frac{2(-5aCd-5bBd+2bcC)(c+d\tan(e+fx))^{3/2}}{15d^2f} + \frac{2bC\tan(e+fx)(c+d\tan(e+fx))^{3/2}}{5df}$$

```
[In] Int[(a + b*Tan[e + f*x])*Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*T
an[e + f*x]^2),x]
```

```
[Out] -(((I*a + b)*(A - I*B - C)*Sqrt[c - I*d]*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/S
qrt[c - I*d]])/f) + ((I*a - b)*(A + I*B - C)*Sqrt[c + I*d]*ArcTanh[Sqrt[c +
 d*Tan[e + f*x]]/Sqrt[c + I*d]])/f + (2*(A*b + a*B - b*C)*Sqrt[c + d*Tan[e
+ f*x]])/f - (2*(2*b*c*C - 5*b*B*d - 5*a*C*d)*(c + d*Tan[e + f*x])^(3/2))/(
15*d^2*f) + (2*b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(3/2))/(5*d*f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3609

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[d*((a + b*Tan[e + f*x])^m/(f*m)), x] + Int
```

```
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rubi steps

$$\text{integral} = \frac{2bC\tan(e+fx)(c+d\tan(e+fx))^{3/2}}{5df} - \frac{2\int \sqrt{c+d\tan(e+fx)}\left(\frac{1}{2}(2bcC-5aAd)-\frac{5}{2}(Ab+aB-bC)d\tan(e+fx)+\frac{1}{2}(2bcC-5bBd-5aCd$$

$$=-\frac{2(2bcC-5bBd-5aCd)(c+d\tan(e+fx))^{3/2}}{15d^2f}+\frac{2bC\tan(e+fx)(c+d\tan(e+fx))^{3/2}}{5df}-\frac{2\int\sqrt{c+d\tan(e+fx)}\left(\frac{5}{2}(bB-a(A-C))d-\frac{5}{2}(Ab+aB-bC)d\tan(e+fx)\right)\,dx}{5d}$$

$$=\frac{2(Ab+aB-bC)\sqrt{c+d\tan(e+fx)}}{f}-\frac{2(2bcC-5bBd-5aCd)(c+d\tan(e+fx))^{3/2}}{15d^2f}+\frac{2bC\tan(e+fx)(c+d\tan(e+fx))^{3/2}}{5df}-\frac{2\int\frac{\frac{5}{2}d(bBc+b(A-C)d-a(Ac-cC-Bd))-\frac{5}{2}d(Abc+aBc-bcC+aAd-bBd-aCd)\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{5d}$$

$$=\frac{2(Ab+aB-bC)\sqrt{c+d\tan(e+fx)}}{f}-\frac{2(2bcC-5bBd-5aCd)(c+d\tan(e+fx))^{3/2}}{15d^2f}+\frac{2bC\tan(e+fx)(c+d\tan(e+fx))^{3/2}}{5df}+\frac{1}{2}((a-ib)(A-iB-C)(c-id))\int\frac{1+i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx+\frac{1}{2}((a+ib)(A+iB-C)(c+id))\int\frac{1-i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx$$

$$=\frac{2(Ab+aB-bC)\sqrt{c+d\tan(e+fx)}}{f}-\frac{2(2bcC-5bBd-5aCd)(c+d\tan(e+fx))^{3/2}}{15d^2f}+\frac{2bC\tan(e+fx)(c+d\tan(e+fx))^{3/2}}{5df}-\frac{((ia-b)(A+iB-C)(c+id))\text{Subst}\left(\int\frac{1}{(-1+x)\sqrt{c+idx}}\,dx,x,-i\tan(e+fx)\right)}{2f}+\frac{((a-ib)(A-iB-C)(ic+d))\text{Subst}\left(\int\frac{1}{(-1+x)\sqrt{c-idx}}\,dx,x,i\tan(e+fx)\right)}{2f}$$

$$
\begin{aligned}
&= \frac{2(Ab+aB-bC)\sqrt{c+d\tan(e+fx)}}{f} \\
&\quad - \frac{2(2bcC-5bBd-5aCd)(c+d\tan(e+fx))^{3/2}}{15d^2f} \\
&\quad + \frac{2bC\tan(e+fx)(c+d\tan(e+fx))^{3/2}}{5df} \\
&\quad - \frac{((a+ib)(A+iB-C)(c+id))\text{Subst}\left(\int \frac{1}{-1+\frac{ic}{d}-\frac{ix^2}{d}}\,dx, x, \sqrt{c+d\tan(e+fx)}\right)}{df} \\
&\quad + \frac{((ia+b)(A-iB-C)(ic+d))\text{Subst}\left(\int \frac{1}{-1-\frac{ic}{d}+\frac{ix^2}{d}}\,dx, x, \sqrt{c+d\tan(e+fx)}\right)}{df} \\
&= -\frac{(ia+b)(A-iB-C)\sqrt{c-id}\,\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f} \\
&\quad + \frac{(ia-b)(A+iB-C)\sqrt{c+id}\,\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f} \\
&\quad + \frac{2(Ab+aB-bC)\sqrt{c+d\tan(e+fx)}}{f} \\
&\quad - \frac{2(2bcC-5bBd-5aCd)(c+d\tan(e+fx))^{3/2}}{15d^2f} \\
&\quad + \frac{2bC\tan(e+fx)(c+d\tan(e+fx))^{3/2}}{5df}
\end{aligned}
$$

Mathematica [A] (verified)

Time = 2.16 (sec) , antiderivative size = 220, normalized size of antiderivative = 0.98

$$
\int (a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx
$$

$$
= \frac{\frac{2(-2bcC+5bBd+5aCd)(c+d\tan(e+fx))^{3/2}}{d} + 6bC\tan(e+fx)(c+d\tan(e+fx))^{3/2} + 15(ia+b)(A-iB-C)d\Big(-
$$

```
[In] Integrate[(a + b*Tan[e + f*x])*Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x]
 + C*Tan[e + f*x]^2),x]
```

```
[Out] ((2*(-2*b*c*C + 5*b*B*d + 5*a*C*d)*(c + d*Tan[e + f*x])^(3/2))/d + 6*b*C*Ta
n[e + f*x]*(c + d*Tan[e + f*x])^(3/2) + 15*(I*a + b)*(A - I*B - C)*d*(-(Sqr
t[c - I*d]*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]]) + Sqrt[c + d*Ta
n[e + f*x]]) + 15*((-I)*a + b)*(A + I*B - C)*d*(-(Sqrt[c + I*d]*ArcTanh[Sqr
t[c + d*Tan[e + f*x]]/Sqrt[c + I*d]]) + Sqrt[c + d*Tan[e + f*x]]))/(15*d*f)
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 2217 vs. 2(194) = 388.

Time = 0.18 (sec) , antiderivative size = 2218, normalized size of antiderivative = 9.90

method	result	size
parts	Expression too large to display	2218
derivativedivides	Expression too large to display	3028
default	Expression too large to display	3028

```
[In] int((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)
,x,method=_RETURNVERBOSE)
```

```
[Out] 1/f*(A*b+B*a)*(2*(c+d*tan(f*x+e))^(1/2)+1/4*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*l
n((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+
d^2)^(1/2))+((c^2+d^2)^(1/2)-c)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c
^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^
(1/2))-1/4*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))
^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))+(-(c^2+d^2)^(1/2)+c)/
(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)
^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)))+2/3/f/d*B*b*(c+d*tan(f*x
+e))^(3/2)+2/3/f/d*C*a*(c+d*tan(f*x+e))^(3/2)-1/4/f/d*ln((c+d*tan(f*x+e))^(
1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*B*(2*(c^
2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*b-1/4/f/d*ln((c+d*tan(f*x+e))^(1/2)
*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*C*(2*(c^2+d^
2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*a+1/f*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*a
rctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)
^(1/2)-2*c)^(1/2))*B*b+1/f*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+
d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/
2))*C*a+1/4/f/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*t
an(f*x+e)-c-(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*b*c+1/4/f/d*ln
((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d
^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*a*c+1/4/f/d*ln(d*tan(f*x+e)+c+(c
+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*B*(2*(c
^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*b+1/4/f/d*ln(d*tan(f*x+e)+c+(c+d*t
an(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*C*(2*(c^2+d
^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*a-1/f*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*
arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2
)^(1/2)-2*c)^(1/2))*B*b-1/f*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*
tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1
/2))*C*a-1/4/f/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2
)+2*c)^(1/2)+(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*b*c-1/4/f/d*l
n(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+
d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*a*c+1/4/f/d*ln((c+d*tan(f*x+e))
^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*A*(2*(
```

```
c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*a-1/f*d/(2*(c^2+d^2)^(1/2)-2*c)^(
1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^
2+d^2)^(1/2)-2*c)^(1/2))*A*a-1/4/f/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)
^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)
^(1/2)*a*c-1/4/f/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1
/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(
1/2)*a+1/f*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)
+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*A*a+1/4/f/d*
ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2
+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*a*c+2*C*b/f/d^2*(1/5*(c+d*tan(
f*x+e))^(5/2)-1/3*(c+d*tan(f*x+e))^(3/2)*c-(c+d*tan(f*x+e))^(1/2)*d^2-d^2*(
1/8*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1
/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))+1/2*((c^2+d^2)^(1/2)-c)/(2*(
c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(
f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))-1/8*(2*(c^2+d^2)^(1/2)+2*c)^(
1/2)*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)
+(c^2+d^2)^(1/2))+1/2*(-(c^2+d^2)^(1/2)+c)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*ar
ctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^
(1/2)-2*c)^(1/2))))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 12410 vs. $2(187) = 374$.

Time = 1.66 (sec) , antiderivative size = 12410, normalized size of antiderivative = 55.40

$$\int (a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= \text{Too large to display}$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x, algorithm="fricas")
[Out] Too large to include
```

Sympy [F]

$$\int (a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= \int (a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$

```
[In] integrate((c+d*tan(f*x+e))**(1/2)*(a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*
x+e)**2),x)
[Out] Integral((a + b*tan(e + f*x))*sqrt(c + d*tan(e + f*x))*(A + B*tan(e + f*x)
+ C*tan(e + f*x)**2), x)
```

Maxima [F]

$$\int (a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\int (C\tan(fx+e)^2+B\tan(fx+e)+A)(b\tan(fx+e)+a)\sqrt{d\tan(fx+e)+c}\,dx$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x, algorithm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)*sqrt
(d*tan(f*x + e) + c), x)
```

Giac [F(-1)]

Timed out.

$$\int (a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [B] (verification not implemented)

Time = 57.65 (sec) , antiderivative size = 22955, normalized size of antiderivative = 102.48

$$\int (a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\text{Too large to display}$$

```
[In] int((a + b*tan(e + f*x))*(c + d*tan(e + f*x))^(1/2)*(A + B*tan(e + f*x) + C
*tan(e + f*x)^2),x)
```

```
[Out] ((2*B*a*d - 4*C*a*c)/(d*f) + (4*C*a*c)/(d*f))*(c + d*tan(e + f*x))^(1/2) +
((2*B*b*d - 6*C*b*c)/(3*d^2*f) + (4*C*b*c)/(3*d^2*f))*(c + d*tan(e + f*x))^
(3/2) + (c + d*tan(e + f*x))^(1/2)*(2*c*((2*B*b*d - 6*C*b*c)/(d^2*f) + (4*C
*b*c)/(d^2*f)) + (2*A*b*d^2 + 6*C*b*c^2 - 4*B*b*c*d)/(d^2*f) - (2*C*b*(d^4*
f + c^2*d^2*f))/(d^4*f^2)) - atan(((((8*(4*A*b*d^4*f^2 - 4*C*b*d^4*f^2 + 4*
A*b*c^2*d^2*f^2 - 4*C*b*c^2*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(
1/2)*((A^2*b^2*c)/(4*f^2) - (4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^
```

4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2))*((A^2*b^2*c)/(4*f^2) - (4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(A^2*b^2*d^4 - B^2*b^2*d^4 + C^2*b^2*d^4 - A^2*b^2*c^2*d^2 + B^2*b^2*c^2*d^2 - C^2*b^2*c^2*d^2 - 2*A*C*b^2*d^4 + 2*A*C*b^2*c^2*d^2 + 4*A*B*b^2*c*d^3 - 4*B*C*b^2*c*d^3))/f^2)*((A^2*b^2*c)/(4*f^2) - (4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2)*1i - (((8*(4*A*b*d^4*f^2 - 4*C*b*d^4*f^2 + 4*A*b*c^2*d^2*f^2 - 4*C*b*c^2*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((A^2*b^2*c)/(4*f^2) - (4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2))*((A^2*b^2*c)/(4*f^2) - (4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(A^2*b^2*d^4 - B^2*b^2*d^4 + C^2*b^2*d^4 - A^2*b^2*c^2*d^2 + B^2*b^2*c^2*d^2 - C^2*b^2*c^2*d^2 - 2*A*C*b^2*d^4 + 2*A*C*b^2*c^2*d^2 + 4*A*B*b^2*c*d^3 - 4*B*C*b^2*c*d^3))/f^2)*((A^2*b^2*c)/(4*f^2) - (4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^

2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f
^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 -
4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C
*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*
b^4*c*d*f^4)^(1/2)/(4*f^4) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A
*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2)*1i)/((
16*(B^3*b^3*d^5 - A^3*b^3*c^3*d^2 + B^3*b^3*c^2*d^3 + C^3*b^3*c^3*d^2 + A^2
*B*b^3*d^5 + B*C^2*b^3*d^5 - A^3*b^3*c*d^4 + C^3*b^3*c*d^4 - A*B^2*b^3*c*d^
4 - 3*A*C^2*b^3*c*d^4 + 3*A^2*C*b^3*c*d^4 + B^2*C*b^3*c*d^4 - A*B^2*b^3*c^3
*d^2 + A^2*B*b^3*c^2*d^3 - 3*A*C^2*b^3*c^3*d^2 + 3*A^2*C*b^3*c^3*d^2 + B*C^
2*b^3*c^2*d^3 + B^2*C*b^3*c^3*d^2 - 2*A*B*C*b^3*d^5 - 2*A*B*C*b^3*c^2*d^3))
/f^3 + (((8*(4*A*b*d^4*f^2 - 4*C*b*d^4*f^2 + 4*A*b*c^2*d^2*f^2 - 4*C*b*c^2*
d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((A^2*b^2*c)/(4*f^2) -
(4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4
+ 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A
^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^
3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d
*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f
^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2
*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^
2))^(1/2))*((A^2*b^2*c)/(4*f^2) - (4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 -
C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2
*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^
4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B
*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*
b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4
) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*
b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2
)*(A^2*b^2*d^4 - B^2*b^2*d^4 + C^2*b^2*d^4 - A^2*b^2*c^2*d^2 + B^2*b^2*c^2*
d^2 - C^2*b^2*c^2*d^2 - 2*A*C*b^2*d^4 + 2*A*C*b^2*c^2*d^2 + 4*A*B*b^2*c*d^3
 - 4*B*C*b^2*c*d^3))/f^2)*((A^2*b^2*c)/(4*f^2) - (4*A*C^3*b^4*d^2*f^4 - B^4
*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*
A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2
*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^
4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f
^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^
4)^(1/2)/(4*f^4) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/
(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2) + (((8*(4*A*b*d^
4*f^2 - 4*C*b*d^4*f^2 + 4*A*b*c^2*d^2*f^2 - 4*C*b*c^2*d^2*f^2))/f^3 + 64*c*
d^2*(c + d*tan(e + f*x))^(1/2)*((A^2*b^2*c)/(4*f^2) - (4*A*C^3*b^4*d^2*f^4
- B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4
 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 -
4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3
*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*
c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c

*d*f^4)^(1/2)/(4*f^4) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2))*((A^2*b^2*c)/(4*f^2) - (4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(A^2*b^2*d^4 - B^2*b^2*d^4 + C^2*b^2*d^4 - A^2*b^2*c^2*d^2 + B^2*b^2*c^2*d^2 - C^2*b^2*c^2*d^2 - 2*A*C*b^2*d^4 + 2*A*C*b^2*c^2*d^2 + 4*A*B*b^2*c*d^3 - 4*B*C*b^2*c*d^3))/f^2)*((A^2*b^2*c)/(4*f^2) - (4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2)))*((A^2*b^2*c)/(4*f^2) - (4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2)*2i - atan(((((8*(4*A*b*d^4*f^2 - 4*C*b*d^4*f^2 + 4*A*b*c^2*d^2*f^2 - 4*C*b*c^2*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) + (A^2*b^2*c)/(4*f^2) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2))*((4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) + (A^2*b^2*c)/(4*f^2) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(A^2*b^2*d^4 - B^2*b^2*d^4 + C^2*b^2*d^4 - A^2*b^2*c^2*d^2 + B^2*b^2*c^

2*d^2 - C^2*b^2*c^2*d^2 - 2*A*C*b^2*d^4 + 2*A*C*b^2*c^2*d^2 + 4*A*B*b^2*c*d^3 - 4*B*C*b^2*c*d^3))/f^2)*((4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) + (A^2*b^2*c)/(4*f^2) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2)*1i - (((8*(4*A*b*d^4*f^2 - 4*C*b*d^4*f^2 + 4*A*b*c^2*d^2*f^2 - 4*C*b*c^2*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) + (A^2*b^2*c)/(4*f^2) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2))*((4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) + (A^2*b^2*c)/(4*f^2) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(A^2*b^2*d^4 - B^2*b^2*d^4 + C^2*b^2*d^4 - A^2*b^2*c^2*d^2 + B^2*b^2*c^2*d^2 - C^2*b^2*c^2*d^2 - 2*A*C*b^2*d^4 + 2*A*C*b^2*c^2*d^2 + 4*A*B*b^2*c*d^3 - 4*B*C*b^2*c*d^3))/f^2)*((4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) + (A^2*b^2*c)/(4*f^2) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2)*1i)/((16*(B^3*b^3*d^5 - A^3*b^3*c^3*d^2 + B^3*b^3*c^2*d^3 + C^3*b^3*c^3*d^2 + A^2*B*b^3*d^5 + B*C^2*b^3*d^5 - A^3*b^3*c*d^4 + C^3*b^3*c*d^4 - A*B^2*b^3*c*d^4 - 3*A*C^2*b^3*c*d^4 + 3*A^2*C*b^3*c*d^4 + B^2*C*b^3*c*d^4 - A*B^2*b^3*c^3*d^2 + A^2*B*b^3*c^2*d^3 - 3*A*C^2*b^3*c^3*d^2 + 3*A^2*C*b^3*c^3*d^2 + B*C^2*b^3*c^2*d^3 + B^2*C*b^3*c^3*d^2 - 2*A*B*C*b^3*d^5 - 2*A*B*C*b^3*c^2*d^3))/f^3 + (((8*(4*A*b*d^4*f^2 - 4*C*b*d^4*f^2 + 4*A*b*c^2*d^2*f^2 - 4*C*b*c^2*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B

^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) + (A^2*b^2*c)/(4*f^2) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2))*((4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) + (A^2*b^2*c)/(4*f^2) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(A^2*b^2*d^4 - B^2*b^2*d^4 + C^2*b^2*d^4 - A^2*b^2*c^2*d^2 + B^2*b^2*c^2*d^2 - C^2*b^2*c^2*d^2 - 2*A*C*b^2*d^4 + 2*A*C*b^2*c^2*d^2 + 4*A*B*b^2*c*d^3 - 4*B*C*b^2*c*d^3))/f^2)*((4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) + (A^2*b^2*c)/(4*f^2) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2) + (((8*(4*A*b*d^4*f^2 - 4*C*b*d^4*f^2 + 4*A*b*c^2*d^2*f^2 - 4*C*b*c^2*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) + (A^2*b^2*c)/(4*f^2) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2))*((4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) + (A^2*b^2*c)/(4*f^2) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(A^2*b^2*d^4 - B^2*b^2*d^4 + C^2*b^2*d^4 - A^2*b^2*c^2*d^2 + B^2*b^2*c^2*d^2 - C^2*b^2*c^2*d^2 - 2*A*C*b^2*d^4 + 2*A*C*b^2*c^2*d^2 + 4*A*B*b^2*c*d^3 - 4*B*C*b^2*c*d^3))/f^2)*((4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^

3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4 *c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4* c*d*f^4)^(1/2)/(4*f^4) + (A^2*b^2*c)/(4*f^2) - (B^2*b^2*c)/(4*f^2) + (C^2*b ^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2* f^2))^(1/2)))*((4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A ^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^ 4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d ^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A* B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) + (A^2*b^2*c)/(4* f^2) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A *C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2)*2i - atan(((((8*(4*B*a*d^4*f ^2 + 4*B*a*c^2*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((B^2*a^ 2*c)/(4*f^2) - (A^2*a^2*c)/(4*f^2) - (4*A*C^3*a^4*d^2*f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*d^2*f^4 - A^4*a^4*d^2*f^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B^2*a^4* c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^4 - 6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*a^4*c^2 *f^4 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 - 4*A*B^2 *C*a^4*d^2*f^4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1/2)/(4* f^4) - (C^2*a^2*c)/(4*f^2) + (A*B*a^2*d)/(2*f^2) + (A*C*a^2*c)/(2*f^2) - (B *C*a^2*d)/(2*f^2))^(1/2))*((B^2*a^2*c)/(4*f^2) - (A^2*a^2*c)/(4*f^2) - (4*A *C^3*a^4*d^2*f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*d^2*f^4 - A^4*a^4*d^2*f^4 + 4* A^3*C*a^4*d^2*f^4 - 4*A^2*B^2*a^4*c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^4 - 6*A^2*C ^2*a^4*d^2*f^4 - 4*B^2*C^2*a^4*c^2*f^4 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A*B^3*a^ 4*c*d*f^4 - 4*A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 - 4*A*B^2*C*a^4*d^2*f^4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1/2)/(4*f^4) - (C^2*a^2*c)/(4*f^2) + (A*B*a^2*d)/ (2*f^2) + (A*C*a^2*c)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2) + (16*(c + d*tan (e + f*x))^(1/2)*(A^2*a^2*d^4 - B^2*a^2*d^4 + C^2*a^2*d^4 - A^2*a^2*c^2*d^2 + B^2*a^2*c^2*d^2 - C^2*a^2*c^2*d^2 - 2*A*C*a^2*d^4 + 2*A*C*a^2*c^2*d^2 + 4*A*B*a^2*c*d^3 - 4*B*C*a^2*c*d^3))/f^2)*((B^2*a^2*c)/(4*f^2) - (A^2*a^2*c) /(4*f^2) - (4*A*C^3*a^4*d^2*f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*d^2*f^4 - A^4*a ^4*d^2*f^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B^2*a^4*c^2*f^4 + 2*A^2*B^2*a^4*d^ 2*f^4 - 6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*a^4*c^2*f^4 + 2*B^2*C^2*a^4*d^2*f ^4 + 4*A*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^ 3*C*a^4*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 - 4*A*B^2*C*a^4*d^2*f^4 - 12*A*B*C^ 2*a^4*c*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1/2)/(4*f^4) - (C^2*a^2*c)/(4*f^2) + (A*B*a^2*d)/(2*f^2) + (A*C*a^2*c)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2)*1 i - (((8*(4*B*a*d^4*f^2 + 4*B*a*c^2*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((B^2*a^2*c)/(4*f^2) - (A^2*a^2*c)/(4*f^2) - (4*A*C^3*a^4*d^2* f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*d^2*f^4 - A^4*a^4*d^2*f^4 + 4*A^3*C*a^4*d^2 *f^4 - 4*A^2*B^2*a^4*c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^4 - 6*A^2*C^2*a^4*d^2*f^ 4 - 4*B^2*C^2*a^4*c^2*f^4 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A*B^3*a^4*c*d*f^4 - 4 *A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8*A*B^2*C* a^4*c^2*f^4 - 4*A*B^2*C*a^4*d^2*f^4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*A^2*B*C*a

^4*c*d*f^4)^(1/2)/(4*f^4) - (C^2*a^2*c)/(4*f^2) + (A*B*a^2*d)/(2*f^2) + (A*C*a^2*c)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2))*((B^2*a^2*c)/(4*f^2) - (A^2*a^2*c)/(4*f^2) - (4*A*C^3*a^4*d^2*f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*d^2*f^4 - A^4*a^4*d^2*f^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B^2*a^4*c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^4 - 6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*a^4*c^2*f^4 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 - 4*A*B^2*C*a^4*d^2*f^4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1/2)/(4*f^4) - (C^2*a^2*c)/(4*f^2) + (A*B*a^2*d)/(2*f^2) + (A*C*a^2*c)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^2*d^4 - B^2*a^2*d^4 + C^2*a^2*d^4 - A^2*a^2*c^2*d^2 + B^2*a^2*c^2*d^2 - C^2*a^2*c^2*d^2 - 2*A*C*a^2*d^4 + 2*A*C*a^2*c^2*d^2 + 4*A*B*a^2*c*d^3 - 4*B*C*a^2*c*d^3))/f^2)*((B^2*a^2*c)/(4*f^2) - (A^2*a^2*c)/(4*f^2) - (4*A*C^3*a^4*d^2*f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*d^2*f^4 - A^4*a^4*d^2*f^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B^2*a^4*c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^4 - 6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*a^4*c^2*f^4 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 - 4*A*B^2*C*a^4*d^2*f^4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1/2)/(4*f^4) - (C^2*a^2*c)/(4*f^2) + (A*B*a^2*d)/(2*f^2) + (A*C*a^2*c)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2)*1i)/((((8*(4*B*a*d^4*f^2 + 4*B*a*c^2*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((B^2*a^2*c)/(4*f^2) - (A^2*a^2*c)/(4*f^2) - (4*A*C^3*a^4*d^2*f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*d^2*f^4 - A^4*a^4*d^2*f^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B^2*a^4*c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^4 - 6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*a^4*c^2*f^4 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 - 4*A*B^2*C*a^4*d^2*f^4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1/2)/(4*f^4) - (C^2*a^2*c)/(4*f^2) + (A*B*a^2*d)/(2*f^2) + (A*C*a^2*c)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2))*((B^2*a^2*c)/(4*f^2) - (A^2*a^2*c)/(4*f^2) - (4*A*C^3*a^4*d^2*f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*d^2*f^4 - A^4*a^4*d^2*f^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B^2*a^4*c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^4 - 6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*a^4*c^2*f^4 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 - 4*A*B^2*C*a^4*d^2*f^4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1/2)/(4*f^4) - (C^2*a^2*c)/(4*f^2) + (A*B*a^2*d)/(2*f^2) + (A*C*a^2*c)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^2*d^4 - B^2*a^2*d^4 + C^2*a^2*d^4 - A^2*a^2*c^2*d^2 + B^2*a^2*c^2*d^2 - C^2*a^2*c^2*d^2 - 2*A*C*a^2*d^4 + 2*A*C*a^2*c^2*d^2 + 4*A*B*a^2*c*d^3 - 4*B*C*a^2*c*d^3))/f^2)*((B^2*a^2*c)/(4*f^2) - (A^2*a^2*c)/(4*f^2) - (4*A*C^3*a^4*d^2*f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*d^2*f^4 - A^4*a^4*d^2*f^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B^2*a^4*c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^4 - 6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*a^4*c^2*f^4 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 - 4*A*B^2*C*a^4*d^2*f^4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1/2)/(4*f^4) - (C^2*a^2*c)/(4*f^2) + (A*B*a^2*d)/(2*f^2) + (A*C*a^2 *c)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2) - (16*(A^3*a^3*d^5 - C^3*a^3*d^5 + A^3*a^3*c^2*d^3 + B^3*a^3*c^3*d^2 - C^3*a^3*c^2*d^3 + A*B^2*a^3*d^5 + 3*A*C^2*a^3*d^5 - 3*A^2*C*a^3*d^5 - B^2*C*a^3*d^5 + B^3*a^3*c*d^4 + A^2*B*a^3*c*d^4 + B*C^2*a^3*c*d^4 + A*B^2*a^3*c^2*d^3 + A^2*B*a^3*c^3*d^2 + 3*A*C^2*a^3*c^2*d^3 - 3*A^2*C*a^3*c^2*d^3 + B*C^2*a^3*c^3*d^2 - B^2*C*a^3*c^2*d^3 - 2*A*B*C*a^3*c*d^4 - 2*A*B*C*a^3*c^3*d^2))/f^3 + (((8*(4*B*a*d^4*f^2 + 4*B*a*c^2*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((B^2*a^2*c)/(4*f^2) - (A^2*a^2*c)/(4*f^2) - (4*A*C^3*a^4*d^2*f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*d^2*f^4 - A^4*a^4*d^2*f^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B^2*a^4*c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^4 - 6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*a^4*c^2*f^4 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 - 4*A*B^2*C*a^4*d^2*f^4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1/2)/(4*f^4) - (C^2*a^2*c)/(4*f^2) + (A*B*a^2*d)/(2*f^2) + (A*C*a^2*c)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2))*((B^2*a^2*c)/(4*f^2) - (A^2*a^2*c)/(4*f^2) - (4*A*C^3*a^4*d^2*f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*d^2*f^4 - A^4*a^4*d^2*f^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B^2*a^4*c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^4 - 6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*a^4*c^2*f^4 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 - 4*A*B^2*C*a^4*d^2*f^4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1/2)/(4*f^4) - (C^2*a^2*c)/(4*f^2) + (A*B*a^2*d)/(2*f^2) + (A*C*a^2*c)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^2*d^4 - B^2*a^2*d^4 + C^2*a^2*d^4 - A^2*a^2*c^2*d^2 + B^2*a^2*c^2*d^2 - C^2*a^2*c^2*d^2 - 2*A*C*a^2*d^4 + 2*A*C*a^2*c^2*d^2 + 4*A*B*a^2*c*d^3 - 4*B*C*a^2*c*d^3))/f^2)*((B^2*a^2*c)/(4*f^2) - (A^2*a^2*c)/(4*f^2) - (4*A*C^3*a^4*d^2*f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*d^2*f^4 - A^4*a^4*d^2*f^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B^2*a^4*c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^4 - 6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*a^4*c^2*f^4 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 - 4*A*B^2*C*a^4*d^2*f^4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1/2)/(4*f^4) - (C^2*a^2*c)/(4*f^2) + (A*B*a^2*d)/(2*f^2) + (A*C*a^2*c)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2)))*((B^2*a^2*c)/(4*f^2) - (A^2*a^2*c)/(4*f^2) - (4*A*C^3*a^4*d^2*f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*d^2*f^4 - A^4*a^4*d^2*f^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B^2*a^4*c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^4 - 6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*a^4*c^2*f^4 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 - 4*A*B^2*C*a^4*d^2*f^4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1/2)/(4*f^4) - (C^2*a^2*c)/(4*f^2) + (A*B*a^2*d)/(2*f^2) + (A*C*a^2*c)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2)*2i - atan(((((8*(4*B*a*d^4*f^2 + 4*B*a*c^2*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((4*A*C^3*a^4*d^2*f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*d^2*f^4 - A^4*a^4*d^2*f^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B^2*a^4*c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^4 - 6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*a^4*c^2*f^4 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 -

4*A*B^2*C*a^4*d^2*f^4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1
/2)/(4*f^4) - (A^2*a^2*c)/(4*f^2) + (B^2*a^2*c)/(4*f^2) - (C^2*a^2*c)/(4*f^
2) + (A*B*a^2*d)/(2*f^2) + (A*C*a^2*c)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2)
)*((4*A*C^3*a^4*d^2*f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*d^2*f^4 - A^4*a^4*d^2*f
^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B^2*a^4*c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^4 -
6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*a^4*c^2*f^4 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A
*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*
c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 - 4*A*B^2*C*a^4*d^2*f^4 - 12*A*B*C^2*a^4*c*
d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1/2)/(4*f^4) - (A^2*a^2*c)/(4*f^2) + (B^2*
a^2*c)/(4*f^2) - (C^2*a^2*c)/(4*f^2) + (A*B*a^2*d)/(2*f^2) + (A*C*a^2*c)/(2
*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^
2*d^4 - B^2*a^2*d^4 + C^2*a^2*d^4 - A^2*a^2*c^2*d^2 + B^2*a^2*c^2*d^2 - C^2
*a^2*c^2*d^2 - 2*A*C*a^2*d^4 + 2*A*C*a^2*c^2*d^2 + 4*A*B*a^2*c*d^3 - 4*B*C*
a^2*c*d^3))/f^2)*((4*A*C^3*a^4*d^2*f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*d^2*f^4
- A^4*a^4*d^2*f^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B^2*a^4*c^2*f^4 + 2*A^2*B^2
*a^4*d^2*f^4 - 6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*a^4*c^2*f^4 + 2*B^2*C^2*a^
4*d^2*f^4 + 4*A*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4*c*d*f^4
 - 4*B^3*C*a^4*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 - 4*A*B^2*C*a^4*d^2*f^4 - 12
*A*B*C^2*a^4*c*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1/2)/(4*f^4) - (A^2*a^2*c)/
(4*f^2) + (B^2*a^2*c)/(4*f^2) - (C^2*a^2*c)/(4*f^2) + (A*B*a^2*d)/(2*f^2) +
 (A*C*a^2*c)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2)*1i - (((8*(4*B*a*d^4*f^2
+ 4*B*a*c^2*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((4*A*C^3*a
^4*d^2*f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*d^2*f^4 - A^4*a^4*d^2*f^4 + 4*A^3*C*
a^4*d^2*f^4 - 4*A^2*B^2*a^4*c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^4 - 6*A^2*C^2*a^4
*d^2*f^4 - 4*B^2*C^2*a^4*c^2*f^4 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A*B^3*a^4*c*d*
f^4 - 4*A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8*A
*B^2*C*a^4*c^2*f^4 - 4*A*B^2*C*a^4*d^2*f^4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*A^
2*B*C*a^4*c*d*f^4)^(1/2)/(4*f^4) - (A^2*a^2*c)/(4*f^2) + (B^2*a^2*c)/(4*f^2
) - (C^2*a^2*c)/(4*f^2) + (A*B*a^2*d)/(2*f^2) + (A*C*a^2*c)/(2*f^2) - (B*C*
a^2*d)/(2*f^2))^(1/2))*((4*A*C^3*a^4*d^2*f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*d^
2*f^4 - A^4*a^4*d^2*f^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B^2*a^4*c^2*f^4 + 2*A
^2*B^2*a^4*d^2*f^4 - 6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*a^4*c^2*f^4 + 2*B^2*
C^2*a^4*d^2*f^4 + 4*A*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4*c
*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 - 4*A*B^2*C*a^4*d^2*f^
4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1/2)/(4*f^4) - (A^2*a
^2*c)/(4*f^2) + (B^2*a^2*c)/(4*f^2) - (C^2*a^2*c)/(4*f^2) + (A*B*a^2*d)/(2*
f^2) + (A*C*a^2*c)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2) - (16*(c + d*tan(e
+ f*x))^(1/2)*(A^2*a^2*d^4 - B^2*a^2*d^4 + C^2*a^2*d^4 - A^2*a^2*c^2*d^2 +
B^2*a^2*c^2*d^2 - C^2*a^2*c^2*d^2 - 2*A*C*a^2*d^4 + 2*A*C*a^2*c^2*d^2 + 4*A
*B*a^2*c*d^3 - 4*B*C*a^2*c*d^3))/f^2)*((4*A*C^3*a^4*d^2*f^4 - B^4*a^4*d^2*f
^4 - C^4*a^4*d^2*f^4 - A^4*a^4*d^2*f^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B^2*a^
4*c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^4 - 6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*a^4*c
^2*f^4 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d*f^4
+ 4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 - 4*A*B
^2*C*a^4*d^2*f^4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1/2)/(

4*f^4) - (A^2*a^2*c)/(4*f^2) + (B^2*a^2*c)/(4*f^2) - (C^2*a^2*c)/(4*f^2) +
(A*B*a^2*d)/(2*f^2) + (A*C*a^2*c)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2)*1i)/
((((8*(4*B*a*d^4*f^2 + 4*B*a*c^2*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*
x))^(1/2)*((4*A*C^3*a^4*d^2*f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*d^2*f^4 - A^4*a
^4*d^2*f^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B^2*a^4*c^2*f^4 + 2*A^2*B^2*a^4*d^
2*f^4 - 6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*a^4*c^2*f^4 + 2*B^2*C^2*a^4*d^2*f
^4 + 4*A*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^
3*C*a^4*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 - 4*A*B^2*C*a^4*d^2*f^4 - 12*A*B*C^
2*a^4*c*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1/2)/(4*f^4) - (A^2*a^2*c)/(4*f^2)
 + (B^2*a^2*c)/(4*f^2) - (C^2*a^2*c)/(4*f^2) + (A*B*a^2*d)/(2*f^2) + (A*C*a
^2*c)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2))*((4*A*C^3*a^4*d^2*f^4 - B^4*a^4
*d^2*f^4 - C^4*a^4*d^2*f^4 - A^4*a^4*d^2*f^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*
B^2*a^4*c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^4 - 6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2
*a^4*c^2*f^4 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*
d*f^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 -
 4*A*B^2*C*a^4*d^2*f^4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(
1/2)/(4*f^4) - (A^2*a^2*c)/(4*f^2) + (B^2*a^2*c)/(4*f^2) - (C^2*a^2*c)/(4*f
^2) + (A*B*a^2*d)/(2*f^2) + (A*C*a^2*c)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2
) + (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^2*d^4 - B^2*a^2*d^4 + C^2*a^2*d^4
 - A^2*a^2*c^2*d^2 + B^2*a^2*c^2*d^2 - C^2*a^2*c^2*d^2 - 2*A*C*a^2*d^4 + 2*
A*C*a^2*c^2*d^2 + 4*A*B*a^2*c*d^3 - 4*B*C*a^2*c*d^3))/f^2)*((4*A*C^3*a^4*d^
2*f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*d^2*f^4 - A^4*a^4*d^2*f^4 + 4*A^3*C*a^4*d
^2*f^4 - 4*A^2*B^2*a^4*c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^4 - 6*A^2*C^2*a^4*d^2*
f^4 - 4*B^2*C^2*a^4*c^2*f^4 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A*B^3*a^4*c*d*f^4 -
 4*A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8*A*B^2*
C*a^4*c^2*f^4 - 4*A*B^2*C*a^4*d^2*f^4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*A^2*B*C
*a^4*c*d*f^4)^(1/2)/(4*f^4) - (A^2*a^2*c)/(4*f^2) + (B^2*a^2*c)/(4*f^2) - (
C^2*a^2*c)/(4*f^2) + (A*B*a^2*d)/(2*f^2) + (A*C*a^2*c)/(2*f^2) - (B*C*a^2*d
)/(2*f^2))^(1/2) - (16*(A^3*a^3*d^5 - C^3*a^3*d^5 + A^3*a^3*c^2*d^3 + B^3*a
^3*c^3*d^2 - C^3*a^3*c^2*d^3 + A*B^2*a^3*d^5 + 3*A*C^2*a^3*d^5 - 3*A^2*C*a^
3*d^5 - B^2*C*a^3*d^5 + B^3*a^3*c*d^4 + A^2*B*a^3*c*d^4 + B*C^2*a^3*c*d^4 +
 A*B^2*a^3*c^2*d^3 + A^2*B*a^3*c^3*d^2 + 3*A*C^2*a^3*c^2*d^3 - 3*A^2*C*a^3*
c^2*d^3 + B*C^2*a^3*c^3*d^2 - B^2*C*a^3*c^2*d^3 - 2*A*B*C*a^3*c*d^4 - 2*A*B
*C*a^3*c^3*d^2))/f^3 + (((8*(4*B*a*d^4*f^2 + 4*B*a*c^2*d^2*f^2))/f^3 + 64*c
d^2(c + d*tan(e + f*x))^(1/2)*((4*A*C^3*a^4*d^2*f^4 - B^4*a^4*d^2*f^4 - C
^4*a^4*d^2*f^4 - A^4*a^4*d^2*f^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B^2*a^4*c^2*
f^4 + 2*A^2*B^2*a^4*d^2*f^4 - 6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*a^4*c^2*f^4
 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d*f^4 + 4*B*
C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 - 4*A*B^2*C*a
^4*d^2*f^4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1/2)/(4*f^4)
 - (A^2*a^2*c)/(4*f^2) + (B^2*a^2*c)/(4*f^2) - (C^2*a^2*c)/(4*f^2) + (A*B*a
^2*d)/(2*f^2) + (A*C*a^2*c)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2))*((4*A*C^3
*a^4*d^2*f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*d^2*f^4 - A^4*a^4*d^2*f^4 + 4*A^3*
C*a^4*d^2*f^4 - 4*A^2*B^2*a^4*c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^4 - 6*A^2*C^2*a
^4*d^2*f^4 - 4*B^2*C^2*a^4*c^2*f^4 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A*B^3*a^4*c*

```
d*f^4 - 4*A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8
*A*B^2*C*a^4*c^2*f^4 - 4*A*B^2*C*a^4*d^2*f^4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*
A^2*B*C*a^4*c*d*f^4)^(1/2)/(4*f^4) - (A^2*a^2*c)/(4*f^2) + (B^2*a^2*c)/(4*f
^2) - (C^2*a^2*c)/(4*f^2) + (A*B*a^2*d)/(2*f^2) + (A*C*a^2*c)/(2*f^2) - (B*
C*a^2*d)/(2*f^2))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^2*d^4 - B^2
*a^2*d^4 + C^2*a^2*d^4 - A^2*a^2*c^2*d^2 + B^2*a^2*c^2*d^2 - C^2*a^2*c^2*d^
2 - 2*A*C*a^2*d^4 + 2*A*C*a^2*c^2*d^2 + 4*A*B*a^2*c*d^3 - 4*B*C*a^2*c*d^3))
/f^2)*((4*A*C^3*a^4*d^2*f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*d^2*f^4 - A^4*a^4*d
^2*f^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B^2*a^4*c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^
4 - 6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*a^4*c^2*f^4 + 2*B^2*C^2*a^4*d^2*f^4 +
 4*A*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*
a^4*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 - 4*A*B^2*C*a^4*d^2*f^4 - 12*A*B*C^2*a^
4*c*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1/2)/(4*f^4) - (A^2*a^2*c)/(4*f^2) + (
B^2*a^2*c)/(4*f^2) - (C^2*a^2*c)/(4*f^2) + (A*B*a^2*d)/(2*f^2) + (A*C*a^2*c
)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2)))*((4*A*C^3*a^4*d^2*f^4 - B^4*a^4*d^
2*f^4 - C^4*a^4*d^2*f^4 - A^4*a^4*d^2*f^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B^2
*a^4*c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^4 - 6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*a^
4*c^2*f^4 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d*f
^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 - 4*
A*B^2*C*a^4*d^2*f^4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1/2
)/(4*f^4) - (A^2*a^2*c)/(4*f^2) + (B^2*a^2*c)/(4*f^2) - (C^2*a^2*c)/(4*f^2)
 + (A*B*a^2*d)/(2*f^2) + (A*C*a^2*c)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2)*2
i + (2*C*a*(c + d*tan(e + f*x))^(3/2))/(3*d*f) + (2*C*b*(c + d*tan(e + f*x)
)^(5/2))/(5*d^2*f)
```

3.93 $\int \sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e$

Optimal result 1148
Rubi [A] (verified) 1148
Mathematica [A] (verified) 1151
Maple [B] (verified) 1151
Fricas [B] (verification not implemented) 1153
Sympy [F] 1154
Maxima [F] 1154
Giac [F(-1)] 1155
Mupad [B] (verification not implemented) 1156

Optimal result

Integrand size = 35, antiderivative size = 155

$$
\begin{aligned}
\int \sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx &= -\frac{(iA+B-iC)\sqrt{c-id}\,\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f} \\
&\quad -\frac{(B-i(A-C))\sqrt{c+id}\,\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f} \\
&\quad +\frac{2B\sqrt{c+d\tan(e+fx)}}{f}+\frac{2C(c+d\tan(e+fx))^{3/2}}{3df}
\end{aligned}
$$

```
[Out] -(I*A+B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))*(c-I*d)^(1/2)/f-
(B-I*(A-C))*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))*(c+I*d)^(1/2)/f+2
*B*(c+d*tan(f*x+e))^(1/2)/f+2/3*C*(c+d*tan(f*x+e))^(3/2)/d/f
```

Rubi [A] (verified)

Time = 0.33 (sec) , antiderivative size = 155, normalized size of antiderivative = 1.00, number of steps used = 9, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.171, Rules used

= {3711, 3609, 3620, 3618, 65, 214}

$$\int \sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=-\frac{\sqrt{c-id}(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f}$$
$$-\frac{\sqrt{c+id}(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f}$$
$$+\frac{2B\sqrt{c+d\tan(e+fx)}}{f}+\frac{2C(c+d\tan(e+fx))^{3/2}}{3df}$$

```
[In] Int[Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] -(((I*A + B - I*C)*Sqrt[c - I*d]*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c -
I*d]])/f) - ((B - I*(A - C))*Sqrt[c + I*d]*ArcTanh[Sqrt[c + d*Tan[e + f*x]]
/Sqrt[c + I*d]])/f + (2*B*Sqrt[c + d*Tan[e + f*x]])/f + (2*C*(c + d*Tan[e +
 f*x])^(3/2))/(3*d*f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3609

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[d*((a + b*Tan[e + f*x])^m/(f*m)), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \frac{2C(c+d\tan(e+fx))^{3/2}}{3df}+\int (A-C+B\tan(e+fx))\sqrt{c+d\tan(e+fx)}\,dx\\
&= \frac{2B\sqrt{c+d\tan(e+fx)}}{f}+\frac{2C(c+d\tan(e+fx))^{3/2}}{3df}\\
&\quad+\int \frac{Ac-cC-Bd+(Bc+(A-C)d)\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx\\
&= \frac{2B\sqrt{c+d\tan(e+fx)}}{f}+\frac{2C(c+d\tan(e+fx))^{3/2}}{3df}\\
&\quad+\frac{1}{2}((A-iB-C)(c-id))\int \frac{1+i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx\\
&\quad+\frac{1}{2}((A+iB-C)(c+id))\int \frac{1-i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx\\
&= \frac{2B\sqrt{c+d\tan(e+fx)}}{f}+\frac{2C(c+d\tan(e+fx))^{3/2}}{3df}\\
&\quad-\frac{(i(A+iB-C)(c+id))\text{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c+idx}}\,dx,x,-i\tan(e+fx)\right)}{2f}\\
&\quad+\frac{((A-iB-C)(ic+d))\text{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c-idx}}\,dx,x,i\tan(e+fx)\right)}{2f}
\end{aligned}$$

$$= \frac{2B\sqrt{c+d\tan(e+fx)}}{f} + \frac{2C(c+d\tan(e+fx))^{3/2}}{3df} - \frac{((A+iB-C)(c+id))\text{Subst}\left(\int \frac{1}{-1+\frac{ic}{d}-\frac{ix^2}{d}}\,dx, x, \sqrt{c+d\tan(e+fx)}\right)}{df} + \frac{((iA+B-iC)(ic+d))\text{Subst}\left(\int \frac{1}{-1-\frac{ic}{d}+\frac{ix^2}{d}}\,dx, x, \sqrt{c+d\tan(e+fx)}\right)}{df}$$

$$= -\frac{(B+i(A-C))\sqrt{c-id}\,\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f} - \frac{(B-i(A-C))\sqrt{c+id}\,\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f} + \frac{2B\sqrt{c+d\tan(e+fx)}}{f} + \frac{2C(c+d\tan(e+fx))^{3/2}}{3df}$$

Mathematica [A] (verified)

Time = 0.59 (sec) , antiderivative size = 150, normalized size of antiderivative = 0.97

$$\int \sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \frac{-3i(A-iB-C)\sqrt{c-id}\,d\,\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)+3i(A+iB-C)\sqrt{c+id}\,d\,\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)+}{3df}$$

```
[In] Integrate[Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2),
x]
[Out] ((-3*I)*(A - I*B - C)*Sqrt[c - I*d]*d*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt
[c - I*d]] + (3*I)*(A + I*B - C)*Sqrt[c + I*d]*d*ArcTanh[Sqrt[c + d*Tan[e +
 f*x]]/Sqrt[c + I*d]] + 2*Sqrt[c + d*Tan[e + f*x]]*(c*C + 3*B*d + C*d*Tan[e
 + f*x]))/(3*d*f)
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 1311 vs. 2(130) = 260.

Time = 0.13 (sec) , antiderivative size = 1312, normalized size of antiderivative = 8.46

method	result	size
parts	Expression too large to display	1312
derivativedivides	Expression too large to display	1472
default	Expression too large to display	1472

```
[In] int((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x,method=_RETURN
VERBOSE)
```

```
[Out] 1/4/f/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e
)-c-(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)-1/f*d/
(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*
tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*A-1/4/f/d*ln((c+d*tan(f*x
+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*A*
(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c-1/4/f/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^
(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2
*c)^(1/2)*(c^2+d^2)^(1/2)+1/f*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+
d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^
(1/2))*A+1/4/f/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2
)+2*c)^(1/2)+(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c+B/f*(2*(c+d
*tan(f*x+e))^(1/2)+1/4*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*ln((c+d*tan(f*x+e))^(1
/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))+((c^2+d^2
)^(1/2)-c)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1
/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))-1/4*(2*(c^2+d^
2)^(1/2)+2*c)^(1/2)*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(
1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))+(-(c^2+d^2)^(1/2)+c)/(2*(c^2+d^2)^(1/2)-2*
c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2
*(c^2+d^2)^(1/2)-2*c)^(1/2)))+C*(2/3/f/d*(c+d*tan(f*x+e))^(3/2)-1/4/f/d*(2*
(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^
2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))+1/f*d/(2*(c^2+d^2)^
(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(
1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))+1/4/f/d*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*
c*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c
^2+d^2)^(1/2))+1/4/f/d*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*ln(d*t
an(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^
(1/2))-1/f*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)
+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))-1/4/f/d*(2*(
c^2+d^2)^(1/2)+2*c)^(1/2)*c*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^
2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2)))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 2588 vs. $2(123) = 246$.

Time = 0.35 (sec) , antiderivative size = 2588, normalized size of antiderivative = 16.70

$$\int \sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \text{Too large to display}$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorit
hm="fricas")
```

```
[Out] -1/6*(3*d*f*sqrt(-(f^2*sqrt(-(4*(A^2*B^2 - 2*A*B^2*C + B^2*C^2)*c^2 + 4*(A^
3*B - A*B^3 + 3*A*B*C^2 - B*C^3 - (3*A^2*B - B^3)*C)*c*d + (A^4 - 2*A^2*B^2
 + B^4 - 4*A*C^3 + C^4 + 2*(3*A^2 - B^2)*C^2 - 4*(A^3 - A*B^2)*C)*d^2)/f^4)
 + (A^2 - B^2 - 2*A*C + C^2)*c - 2*(A*B - B*C)*d)/f^2)*log((2*(A^3*B + A*B^
3 + 3*A*B*C^2 - B*C^3 - (3*A^2*B + B^3)*C)*c + (A^4 - B^4 - 4*A^3*C + 6*A^2
*C^2 - 4*A*C^3 + C^4)*d)*sqrt(d*tan(f*x + e) + c) + ((A - C)*f^3*sqrt(-(4*(
A^2*B^2 - 2*A*B^2*C + B^2*C^2)*c^2 + 4*(A^3*B - A*B^3 + 3*A*B*C^2 - B*C^3 -
 (3*A^2*B - B^3)*C)*c*d + (A^4 - 2*A^2*B^2 + B^4 - 4*A*C^3 + C^4 + 2*(3*A^2
 - B^2)*C^2 - 4*(A^3 - A*B^2)*C)*d^2)/f^4) + (2*(A*B^2 - B^2*C)*c + (A^2*B
- B^3 - 2*A*B*C + B*C^2)*d)*f)*sqrt(-(f^2*sqrt(-(4*(A^2*B^2 - 2*A*B^2*C + B
^2*C^2)*c^2 + 4*(A^3*B - A*B^3 + 3*A*B*C^2 - B*C^3 - (3*A^2*B - B^3)*C)*c*d
 + (A^4 - 2*A^2*B^2 + B^4 - 4*A*C^3 + C^4 + 2*(3*A^2 - B^2)*C^2 - 4*(A^3 -
A*B^2)*C)*d^2)/f^4) + (A^2 - B^2 - 2*A*C + C^2)*c - 2*(A*B - B*C)*d)/f^2))
- 3*d*f*sqrt(-(f^2*sqrt(-(4*(A^2*B^2 - 2*A*B^2*C + B^2*C^2)*c^2 + 4*(A^3*B
- A*B^3 + 3*A*B*C^2 - B*C^3 - (3*A^2*B - B^3)*C)*c*d + (A^4 - 2*A^2*B^2 + B
^4 - 4*A*C^3 + C^4 + 2*(3*A^2 - B^2)*C^2 - 4*(A^3 - A*B^2)*C)*d^2)/f^4) + (
A^2 - B^2 - 2*A*C + C^2)*c - 2*(A*B - B*C)*d)/f^2)*log((2*(A^3*B + A*B^3 +
3*A*B*C^2 - B*C^3 - (3*A^2*B + B^3)*C)*c + (A^4 - B^4 - 4*A^3*C + 6*A^2*C^2
 - 4*A*C^3 + C^4)*d)*sqrt(d*tan(f*x + e) + c) - ((A - C)*f^3*sqrt(-(4*(A^2*
B^2 - 2*A*B^2*C + B^2*C^2)*c^2 + 4*(A^3*B - A*B^3 + 3*A*B*C^2 - B*C^3 - (3*
A^2*B - B^3)*C)*c*d + (A^4 - 2*A^2*B^2 + B^4 - 4*A*C^3 + C^4 + 2*(3*A^2 - B
^2)*C^2 - 4*(A^3 - A*B^2)*C)*d^2)/f^4) + (2*(A*B^2 - B^2*C)*c + (A^2*B - B^
3 - 2*A*B*C + B*C^2)*d)*f)*sqrt(-(f^2*sqrt(-(4*(A^2*B^2 - 2*A*B^2*C + B^2*C
^2)*c^2 + 4*(A^3*B - A*B^3 + 3*A*B*C^2 - B*C^3 - (3*A^2*B - B^3)*C)*c*d + (
A^4 - 2*A^2*B^2 + B^4 - 4*A*C^3 + C^4 + 2*(3*A^2 - B^2)*C^2 - 4*(A^3 - A*B^
2)*C)*d^2)/f^4) + (A^2 - B^2 - 2*A*C + C^2)*c - 2*(A*B - B*C)*d)/f^2)) - 3*
d*f*sqrt((f^2*sqrt(-(4*(A^2*B^2 - 2*A*B^2*C + B^2*C^2)*c^2 + 4*(A^3*B - A*B
^3 + 3*A*B*C^2 - B*C^3 - (3*A^2*B - B^3)*C)*c*d + (A^4 - 2*A^2*B^2 + B^4 -
4*A*C^3 + C^4 + 2*(3*A^2 - B^2)*C^2 - 4*(A^3 - A*B^2)*C)*d^2)/f^4) - (A^2 -
 B^2 - 2*A*C + C^2)*c + 2*(A*B - B*C)*d)/f^2)*log((2*(A^3*B + A*B^3 + 3*A*B
*C^2 - B*C^3 - (3*A^2*B + B^3)*C)*c + (A^4 - B^4 - 4*A^3*C + 6*A^2*C^2 - 4*
A*C^3 + C^4)*d)*sqrt(d*tan(f*x + e) + c) + ((A - C)*f^3*sqrt(-(4*(A^2*B^2 -
 2*A*B^2*C + B^2*C^2)*c^2 + 4*(A^3*B - A*B^3 + 3*A*B*C^2 - B*C^3 - (3*A^2*B
 - B^3)*C)*c*d + (A^4 - 2*A^2*B^2 + B^4 - 4*A*C^3 + C^4 + 2*(3*A^2 - B^2)*C
```

```
^2 - 4*(A^3 - A*B^2)*C)*d^2)/f^4) - (2*(A*B^2 - B^2*C)*c + (A^2*B - B^3 - 2
*A*B*C + B*C^2)*d)*f)*sqrt((f^2*sqrt(-(4*(A^2*B^2 - 2*A*B^2*C + B^2*C^2)*c^
2 + 4*(A^3*B - A*B^3 + 3*A*B*C^2 - B*C^3 - (3*A^2*B - B^3)*C)*c*d + (A^4 -
2*A^2*B^2 + B^4 - 4*A*C^3 + C^4 + 2*(3*A^2 - B^2)*C^2 - 4*(A^3 - A*B^2)*C)*
d^2)/f^4) - (A^2 - B^2 - 2*A*C + C^2)*c + 2*(A*B - B*C)*d)/f^2)) + 3*d*f*sq
rt((f^2*sqrt(-(4*(A^2*B^2 - 2*A*B^2*C + B^2*C^2)*c^2 + 4*(A^3*B - A*B^3 + 3
*A*B*C^2 - B*C^3 - (3*A^2*B - B^3)*C)*c*d + (A^4 - 2*A^2*B^2 + B^4 - 4*A*C^
3 + C^4 + 2*(3*A^2 - B^2)*C^2 - 4*(A^3 - A*B^2)*C)*d^2)/f^4) - (A^2 - B^2 -
 2*A*C + C^2)*c + 2*(A*B - B*C)*d)/f^2)*log((2*(A^3*B + A*B^3 + 3*A*B*C^2 -
 B*C^3 - (3*A^2*B + B^3)*C)*c + (A^4 - B^4 - 4*A^3*C + 6*A^2*C^2 - 4*A*C^3
+ C^4)*d)*sqrt(d*tan(f*x + e) + c) - ((A - C)*f^3*sqrt(-(4*(A^2*B^2 - 2*A*B
^2*C + B^2*C^2)*c^2 + 4*(A^3*B - A*B^3 + 3*A*B*C^2 - B*C^3 - (3*A^2*B - B^3
)*C)*c*d + (A^4 - 2*A^2*B^2 + B^4 - 4*A*C^3 + C^4 + 2*(3*A^2 - B^2)*C^2 - 4
*(A^3 - A*B^2)*C)*d^2)/f^4) - (2*(A*B^2 - B^2*C)*c + (A^2*B - B^3 - 2*A*B*C
 + B*C^2)*d)*f)*sqrt((f^2*sqrt(-(4*(A^2*B^2 - 2*A*B^2*C + B^2*C^2)*c^2 + 4*
(A^3*B - A*B^3 + 3*A*B*C^2 - B*C^3 - (3*A^2*B - B^3)*C)*c*d + (A^4 - 2*A^2*
B^2 + B^4 - 4*A*C^3 + C^4 + 2*(3*A^2 - B^2)*C^2 - 4*(A^3 - A*B^2)*C)*d^2)/f
^4) - (A^2 - B^2 - 2*A*C + C^2)*c + 2*(A*B - B*C)*d)/f^2)) - 4*(C*d*tan(f*x
 + e) + C*c + 3*B*d)*sqrt(d*tan(f*x + e) + c))/(d*f)
```

Sympy [F]

$$\int \sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\int \sqrt{c+d\tan\left(e+fx\right)}\left(A+B\tan\left(e+fx\right)+C\tan^2\left(e+fx\right)\right)\,dx$$

```
[In] integrate((c+d*tan(f*x+e))**(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2),x)
[Out] Integral(sqrt(c + d*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)**2),
 x)
```

Maxima [F]

$$\int \sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\int \left(C\tan\left(fx+e\right)^2+B\tan\left(fx+e\right)+A\right)\sqrt{d\tan\left(fx+e\right)+c}\,dx$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorit
hm="maxima")
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*sqrt(d*tan(f*x + e) + c),
 x)
```

Giac [F(-1)]

Timed out.

$$\int \sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorit
hm="giac")
```

```
[Out] Timed out
```

Mupad [B] (verification not implemented)

Time = 16.04 (sec) , antiderivative size = 1199, normalized size of antiderivative = 7.74

$$\int \sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= 2\,\mathrm{atanh}\left(\frac{32\,B^2\,d^4\,\sqrt{\frac{B^2\,c}{4\,f^2}-\frac{\sqrt{-B^4\,d^2\,f^4}}{4\,f^4}}\,\sqrt{c+d\tan\left(e+f\,x\right)}}{\frac{16\,B\,d^4\,\sqrt{-B^4\,d^2\,f^4}}{f^3}+\frac{16\,B\,c^2\,d^2\,\sqrt{-B^4\,d^2\,f^4}}{f^3}} - \frac{32\,c\,d^2\,\sqrt{\frac{B^2\,c}{4\,f^2}-\frac{\sqrt{-B^4\,d^2\,f^4}}{4\,f^4}}\,\sqrt{c+d\tan\left(e+f\,x\right)}\,\sqrt{-B^4\,d^2\,f^4}}{\frac{16\,B\,d^4\,\sqrt{-B^4\,d^2\,f^4}}{f}+\frac{16\,B\,c^2\,d^2\,\sqrt{-B^4\,d^2\,f^4}}{f}}\right)\sqrt{-\frac{\sqrt{-B^4\,d^2\,f^4}-B^2\,c\,f^2}{4\,f^4}}$$
$$- 2\,\mathrm{atanh}\left(\frac{32\,B^2\,d^4\,\sqrt{\frac{\sqrt{-B^4\,d^2\,f^4}}{4\,f^4}+\frac{B^2\,c}{4\,f^2}}\,\sqrt{c+d\tan\left(e+f\,x\right)}}{\frac{16\,B\,d^4\,\sqrt{-B^4\,d^2\,f^4}}{f^3}+\frac{16\,B\,c^2\,d^2\,\sqrt{-B^4\,d^2\,f^4}}{f^3}} + \frac{32\,c\,d^2\,\sqrt{\frac{\sqrt{-B^4\,d^2\,f^4}}{4\,f^4}+\frac{B^2\,c}{4\,f^2}}\,\sqrt{c+d\tan\left(e+f\,x\right)}\,\sqrt{-B^4\,d^2\,f^4}}{\frac{16\,B\,d^4\,\sqrt{-B^4\,d^2\,f^4}}{f}+\frac{16\,B\,c^2\,d^2\,\sqrt{-B^4\,d^2\,f^4}}{f}}\right)\sqrt{\frac{\sqrt{-B^4\,d^2\,f^4}+B^2\,c\,f^2}{4\,f^4}}$$
$$-\mathrm{atanh}\left(\frac{f^3\,\sqrt{-\frac{\sqrt{-A^4\,d^2\,f^4}+A^2\,c\,f^2}{f^4}}\,\left(\frac{16\,\left(A^2\,d^4-A^2\,c^2\,d^2\right)\,\sqrt{c+d\tan(e+f\,x)}}{f^2}+\frac{16\,c\,d^2\,\left(\sqrt{-A^4\,d^2\,f^4}+A^2\,c\,f^2\right)\,\sqrt{c+d\tan(e+f\,x)}}{f^4}\right)}{16\,\left(A^3\,c^2\,d^3+A^3\,d^5\right)}\right)$$
$$-\mathrm{atanh}\left(\frac{f^3\,\sqrt{\frac{\sqrt{-A^4\,d^2\,f^4}-A^2\,c\,f^2}{f^4}}\,\left(\frac{16\,\left(A^2\,d^4-A^2\,c^2\,d^2\right)\,\sqrt{c+d\tan(e+f\,x)}}{f^2}-\frac{16\,c\,d^2\,\left(\sqrt{-A^4\,d^2\,f^4}-A^2\,c\,f^2\right)\,\sqrt{c+d\tan(e+f\,x)}}{f^4}\right)}{16\,\left(A^3\,c^2\,d^3+A^3\,d^5\right)}\right)$$
$$+\mathrm{atanh}\left(\frac{f^3\,\sqrt{-\frac{\sqrt{-C^4\,d^2\,f^4}+C^2\,c\,f^2}{f^4}}\,\left(\frac{16\,\left(C^2\,d^4-C^2\,c^2\,d^2\right)\,\sqrt{c+d\tan(e+f\,x)}}{f^2}+\frac{16\,c\,d^2\,\left(\sqrt{-C^4\,d^2\,f^4}+C^2\,c\,f^2\right)\,\sqrt{c+d\tan(e+f\,x)}}{f^4}\right)}{16\,\left(C^3\,c^2\,d^3+C^3\,d^5\right)}\right)$$
$$+\mathrm{atanh}\left(\frac{f^3\,\sqrt{\frac{\sqrt{-C^4\,d^2\,f^4}-C^2\,c\,f^2}{f^4}}\,\left(\frac{16\,\left(C^2\,d^4-C^2\,c^2\,d^2\right)\,\sqrt{c+d\tan(e+f\,x)}}{f^2}-\frac{16\,c\,d^2\,\left(\sqrt{-C^4\,d^2\,f^4}-C^2\,c\,f^2\right)\,\sqrt{c+d\tan(e+f\,x)}}{f^4}\right)}{16\,\left(C^3\,c^2\,d^3+C^3\,d^5\right)}\right)$$
$$+\frac{2\,B\,\sqrt{c+d\tan\left(e+f\,x\right)}}{f}+\frac{2\,C\,\left(c+d\tan(e+f\,x)\right)^{3/2}}{3\,d\,f}$$

```
[In] int((c + d*tan(e + f*x))^(1/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2),x)
```

```
[Out] 2*atanh((32*B^2*d^4*((B^2*c)/(4*f^2) - (-B^4*d^2*f^4)^(1/2)/(4*f^4))^(1/2)*
(c + d*tan(e + f*x))^(1/2))/((16*B*d^4*(-B^4*d^2*f^4)^(1/2))/f^3 + (16*B*c^
2*d^2*(-B^4*d^2*f^4)^(1/2))/f^3) - (32*c*d^2*((B^2*c)/(4*f^2) - (-B^4*d^2*f
```

^4)^(1/2)/(4*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(-B^4*d^2*f^4)^(1/2))/(
(16*B*d^4*(-B^4*d^2*f^4)^(1/2))/f + (16*B*c^2*d^2*(-B^4*d^2*f^4)^(1/2))/f))
*(-((-B^4*d^2*f^4)^(1/2) - B^2*c*f^2)/(4*f^4))^(1/2) - 2*atanh((32*B^2*d^4*
((-B^4*d^2*f^4)^(1/2)/(4*f^4) + (B^2*c)/(4*f^2))^(1/2)*(c + d*tan(e + f*x))
^(1/2))/((16*B*d^4*(-B^4*d^2*f^4)^(1/2))/f^3 + (16*B*c^2*d^2*(-B^4*d^2*f^4)
^(1/2))/f^3) + (32*c*d^2*((-B^4*d^2*f^4)^(1/2)/(4*f^4) + (B^2*c)/(4*f^2))^(
1/2)*(c + d*tan(e + f*x))^(1/2)*(-B^4*d^2*f^4)^(1/2))/((16*B*d^4*(-B^4*d^2*
f^4)^(1/2))/f + (16*B*c^2*d^2*(-B^4*d^2*f^4)^(1/2))/f))*(((-B^4*d^2*f^4)^(1
/2) + B^2*c*f^2)/(4*f^4))^(1/2) - atanh((f^3*(-((-A^4*d^2*f^4)^(1/2) + A^2*
c*f^2)/f^4)^(1/2)*((16*(A^2*d^4 - A^2*c^2*d^2)*(c + d*tan(e + f*x))^(1/2))/
f^2 + (16*c*d^2*((-A^4*d^2*f^4)^(1/2) + A^2*c*f^2)*(c + d*tan(e + f*x))^(1/
2))/f^4))/(16*(A^3*d^5 + A^3*c^2*d^3)))*(-((-A^4*d^2*f^4)^(1/2) + A^2*c*f^2
)/f^4)^(1/2) - atanh((f^3*(((-A^4*d^2*f^4)^(1/2) - A^2*c*f^2)/f^4)^(1/2)*((
16*(A^2*d^4 - A^2*c^2*d^2)*(c + d*tan(e + f*x))^(1/2))/f^2 - (16*c*d^2*((-A
^4*d^2*f^4)^(1/2) - A^2*c*f^2)*(c + d*tan(e + f*x))^(1/2))/f^4))/(16*(A^3*d
^5 + A^3*c^2*d^3)))*(((-A^4*d^2*f^4)^(1/2) - A^2*c*f^2)/f^4)^(1/2) + atanh(
(f^3*(-((-C^4*d^2*f^4)^(1/2) + C^2*c*f^2)/f^4)^(1/2)*((16*(C^2*d^4 - C^2*c^
2*d^2)*(c + d*tan(e + f*x))^(1/2))/f^2 + (16*c*d^2*((-C^4*d^2*f^4)^(1/2) +
C^2*c*f^2)*(c + d*tan(e + f*x))^(1/2))/f^4))/(16*(C^3*d^5 + C^3*c^2*d^3)))*
(-((-C^4*d^2*f^4)^(1/2) + C^2*c*f^2)/f^4)^(1/2) + atanh((f^3*(((-C^4*d^2*f^
4)^(1/2) - C^2*c*f^2)/f^4)^(1/2)*((16*(C^2*d^4 - C^2*c^2*d^2)*(c + d*tan(e
+ f*x))^(1/2))/f^2 - (16*c*d^2*((-C^4*d^2*f^4)^(1/2) - C^2*c*f^2)*(c + d*ta
n(e + f*x))^(1/2))/f^4))/(16*(C^3*d^5 + C^3*c^2*d^3)))*(((-C^4*d^2*f^4)^(1/
2) - C^2*c*f^2)/f^4)^(1/2) + (2*B*(c + d*tan(e + f*x))^(1/2))/f + (2*C*(c +
d*tan(e + f*x))^(3/2))/(3*d*f)

3.94 $\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx$

Optimal result 1158
Rubi [A] (verified) 1159
Mathematica [A] (verified) 1162
Maple [B] (verified) 1162
Fricas [F(-1)] 1164
Sympy [F] 1165
Maxima [F(-2)] 1165
Giac [F(-1)] 1165
Mupad [B] (verification not implemented) 1166

Optimal result

Integrand size = 47, antiderivative size = 234

$$\begin{aligned}\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx &= -\frac{(iA+B-iC)\sqrt{c-id}\,\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(a-ib)f} \\ &\quad+\frac{(iA-B-iC)\sqrt{c+id}\,\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(a+ib)f} \\ &\quad-\frac{2(Ab^2-a(bB-aC))\sqrt{bc-ad}\,\text{arctanh}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{b^{3/2}(a^2+b^2)f}+\frac{2C\sqrt{c+d\tan(e+fx)}}{bf}\end{aligned}$$

```
[Out] -(I*A+B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))*(c-I*d)^(1/2)/(a
-I*b)/f+(I*A-B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))*(c+I*d)^(
1/2)/(a+I*b)/f-2*(A*b^2-a*(B*b-C*a))*arctanh(b^(1/2)*(c+d*tan(f*x+e))^(1/2)
/(-a*d+b*c)^(1/2))*(-a*d+b*c)^(1/2)/b^(3/2)/(a^2+b^2)/f+2*C*(c+d*tan(f*x+e)
)^(1/2)/b/f
```

Rubi [A] (verified)

Time = 1.19 (sec) , antiderivative size = 234, normalized size of antiderivative = 1.00, number of steps used = 12, number of rules used = 7, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.149, Rules used = {3728, 3734, 3620, 3618, 65, 214, 3715}

$$\begin{aligned}
&\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx\\
&=-\frac{2\sqrt{bc-ad}(Ab^2-a(bB-aC))\,\text{arctanh}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{b^{3/2}f\,(a^2+b^2)}\\
&\quad-\frac{\sqrt{c-id}(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f(a-ib)}\\
&\quad+\frac{\sqrt{c+id}(iA-B-iC)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f(a+ib)}+\frac{2C\sqrt{c+d\tan(e+fx)}}{bf}
\end{aligned}$$

```
[In] Int[(Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a +
 b*Tan[e + f*x]),x]
```

```
[Out] -(((I*A + B - I*C)*Sqrt[c - I*d]*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c -
I*d]])/((a - I*b)*f)) + ((I*A - B - I*C)*Sqrt[c + I*d]*ArcTanh[Sqrt[c + d*T
an[e + f*x]]/Sqrt[c + I*d]])/((a + I*b)*f) - (2*(A*b^2 - a*(b*B - a*C))*Sqr
t[b*c - a*d]*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*d]])/(
b^(3/2)*(a^2 + b^2)*f) + (2*C*Sqrt[c + d*Tan[e + f*x]])/(b*f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3715

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_) + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :>
 Dist[A/f, Subst[Int[(a + b*x)^m*(c + d*x)^n, x], x, Tan[e + f*x]], x] /; F
reeQ[{a, b, c, d, e, f, A, C, m, n}, x] && EqQ[A, C]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3734

```
Int[(((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2))/((a_.) + (b_.)*tan[(e_.)
+ (f_.)*(x_)]), x_Symbol] :> Dist[1/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n
*Simp[b*B + a*(A - C) + (a*B - b*(A - C))*Tan[e + f*x], x], x], x] + Dist[(
A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n*((1 + Tan[e
+ f*x]^2)/(a + b*Tan[e + f*x])), x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C
, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&
 !GtQ[n, 0] &&  !LeQ[n, -1]
```

Rubi steps

$$\text{integral} = \frac{2C\sqrt{c+d\tan(e+fx)}}{bf} + \frac{2\int \frac{\frac{1}{2}(Abc-aCd)+\frac{1}{2}b(Bc+(A-C)d)\tan(e+fx)+\frac{1}{2}(bcC+bBd-aCd)\tan^2(e+fx)}{(a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}}\,dx}{b}$$

$$= \frac{2C\sqrt{c+d\tan(e+fx)}}{bf} + \frac{2\int \frac{\frac{1}{2}b(bBc+b(A-C)d+a(Ac-cC-Bd))-\frac{1}{2}b(Abc-aBc-bcC-aAd-bBd+aCd)\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{b\left(a^2+b^2\right)} + \frac{\left(\left(Ab^2-a(bB-aC)\right)(bc-ad)\right)\int \frac{1+\tan^2(e+fx)}{(a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}}\,dx}{b\left(a^2+b^2\right)}$$

$$= \frac{2C\sqrt{c+d\tan(e+fx)}}{bf} + \frac{((A-iB-C)(c-id))\int \frac{1+i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(a-ib)} + \frac{((A+iB-C)(c+id))\int \frac{1-i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(a+ib)} + \frac{\left(\left(Ab^2-a(bB-aC)\right)(bc-ad)\right)\text{Subst}\left(\int \frac{1}{(a+bx)\sqrt{c+dx}}\,dx, x, \tan(e+fx)\right)}{b\left(a^2+b^2\right)f}$$

$$= \frac{2C\sqrt{c+d\tan(e+fx)}}{bf} - \frac{(i(A+iB-C)(c+id))\text{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c+idx}}\,dx, x, -i\tan(e+fx)\right)}{2(a+ib)f} + \frac{((A-iB-C)(ic+d))\text{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c-idx}}\,dx, x, i\tan(e+fx)\right)}{2(a-ib)f} + \frac{\left(2\left(Ab^2-a(bB-aC)\right)(bc-ad)\right)\text{Subst}\left(\int \frac{1}{a-\frac{bc}{d}+\frac{bx^2}{d}}\,dx, x, \sqrt{c+d\tan(e+fx)}\right)}{b\left(a^2+b^2\right)df}$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)\sqrt{bc-ad}\,\text{arctanh}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{b^{3/2}\left(a^2+b^2\right)f} + \frac{2C\sqrt{c+d\tan(e+fx)}}{bf} - \frac{((A+iB-C)(c+id))\text{Subst}\left(\int \frac{1}{-1+\frac{ic}{d}-\frac{ix^2}{d}}\,dx, x, \sqrt{c+d\tan(e+fx)}\right)}{(a+ib)df} + \frac{((iA+B-iC)(ic+d))\text{Subst}\left(\int \frac{1}{-1-\frac{ic}{d}+\frac{ix^2}{d}}\,dx, x, \sqrt{c+d\tan(e+fx)}\right)}{(a-ib)df}$$

$$= \frac{(A-iB-C)\sqrt{c-id}\,\mathrm{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(ia+b)f} - \frac{(A+iB-C)\sqrt{c+id}\,\mathrm{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(ia-b)f} - \frac{2(Ab^2-a(bB-aC))\sqrt{bc-ad}\,\mathrm{arctanh}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{b^{3/2}(a^2+b^2)f} + \frac{2C\sqrt{c+d\tan(e+fx)}}{bf}$$

Mathematica [A] (verified)

Time = 0.76 (sec) , antiderivative size = 233, normalized size of antiderivative = 1.00

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx$$
$$= \frac{b^{3/2}(-ia+b)(A-iB-C)\sqrt{c-id}\,\mathrm{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right) + b^{3/2}(ia+b)(A+iB-C)\sqrt{c+id}\,\mathrm{arctanh}\left(\frac{\sqrt{}}{}\right.}{b^{3/2}}$$

```
[In] Integrate[(Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)
)/(a + b*Tan[e + f*x]),x]
```

```
[Out] (b^(3/2)*((-I)*a + b)*(A - I*B - C)*Sqrt[c - I*d]*ArcTanh[Sqrt[c + d*Tan[e
+ f*x]]/Sqrt[c - I*d]] + b^(3/2)*(I*a + b)*(A + I*B - C)*Sqrt[c + I*d]*ArcT
anh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]] - 2*(A*b^2 + a*(-(b*B) + a*C))*
Sqrt[b*c - a*d]*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*d]]
 + 2*Sqrt[b]*(a^2 + b^2)*C*Sqrt[c + d*Tan[e + f*x]])/(b^(3/2)*(a^2 + b^2)*f
)
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 3575 vs. 2(200) = 400.

Time = 0.15 (sec) , antiderivative size = 3576, normalized size of antiderivative = 15.28

method	result	size
derivativedivides	Expression too large to display	3576
default	Expression too large to display	3576

```
[In] int((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
,x,method=_RETURNVERBOSE)
```

```
[Out] 2*C*(c+d*tan(f*x+e))^(1/2)/b/f-1/4/f/(a^2+b^2)/d*ln((c+d*tan(f*x+e))^(1/2)*
(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*A*(2*(c^2+d^2
)^(1/2)+2*c)^(1/2)*a*c+1/4/f/(a^2+b^2)/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+
d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)+
2*c)^(1/2)*(c^2+d^2)^(1/2)*b-2/f*b/(a^2+b^2)/((a*d-b*c)*b)^(1/2)*arctan(b*(
c+d*tan(f*x+e))^(1/2)/((a*d-b*c)*b)^(1/2))*B*a*c-1/4/f/(a^2+b^2)/d*ln(d*tan
(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1
/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*a+1/4/f/(a^2+b^2)/d*ln
(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d
^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*a*c-1/4/f/(a^2+b^2)/d*ln(d*tan(f
*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2
))*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*b-1/4/f/(a^2+b^2)/d*ln(d
*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2
)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*a*c+1/4/f/(a^2+b^2)/d*ln((c+d*tan(
f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))
*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*a-2/f*b/(a^2+b^2)/((a*d-b*
c)*b)^(1/2)*arctan(b*(c+d*tan(f*x+e))^(1/2)/((a*d-b*c)*b)^(1/2))*A*a*d-1/4/
f/(a^2+b^2)/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan
(f*x+e)-c-(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*b*c-1/4/f/(a^2+b
^2)/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-
c-(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*a-2/f/b/
(a^2+b^2)/((a*d-b*c)*b)^(1/2)*arctan(b*(c+d*tan(f*x+e))^(1/2)/((a*d-b*c)*b)
^(1/2))*a^3*C*d+1/4/f/(a^2+b^2)/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1
/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1
/2)*a*c+1/4/f/(a^2+b^2)/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+
d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*b*c+
1/4/f/(a^2+b^2)/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/
2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1
/2)*a-1/f/(a^2+b^2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e)
)^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*(c^
2+d^2)^(1/2)*a-1/f/(a^2+b^2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*t
an(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/
2))*C*(c^2+d^2)^(1/2)*b+1/f/(a^2+b^2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(
(2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)
-2*c)^(1/2))*B*a*c+1/f/(a^2+b^2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c
+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)
^(1/2))*C*b*c-1/f/(a^2+b^2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*ta
n(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2
))*A*b*c+1/f/(a^2+b^2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(
1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*C*
(c^2+d^2)^(1/2)*b-1/f/(a^2+b^2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c
^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^
(1/2))*C*b*c+1/f/(a^2+b^2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^
2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)
)*A*b*c-1/f/(a^2+b^2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1
```

```
/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*a
*c+2/f/(a^2+b^2)/((a*d-b*c)*b)^(1/2)*arctan(b*(c+d*tan(f*x+e))^(1/2)/((a*d-
b*c)*b)^(1/2))*B*a^2*d+2/f*b^2/(a^2+b^2)/((a*d-b*c)*b)^(1/2)*arctan(b*(c+d*
tan(f*x+e))^(1/2)/((a*d-b*c)*b)^(1/2))*A*c-1/f/(a^2+b^2)/(2*(c^2+d^2)^(1/2)
-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))
/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*A*(c^2+d^2)^(1/2)*b+1/f/(a^2+b^2)/(2*(c^2+d
^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e
))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*(c^2+d^2)^(1/2)*a-1/f/(a^2+b^2)*
d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+
d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*A*a-1/f/(a^2+b^2)*d/(2*
(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan
(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*b+1/f/(a^2+b^2)*d/(2*(c^2+
d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+
e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*C*a+1/f/(a^2+b^2)*d/(2*(c^2+d^2)^
(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(
1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*A*a+1/f/(a^2+b^2)*d/(2*(c^2+d^2)^(1/2)
-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))
/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*b-1/f/(a^2+b^2)*d/(2*(c^2+d^2)^(1/2)-2*c)
^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(
c^2+d^2)^(1/2)-2*c)^(1/2))*C*a+2/f/(a^2+b^2)/((a*d-b*c)*b)^(1/2)*arctan(b*(
c+d*tan(f*x+e))^(1/2)/((a*d-b*c)*b)^(1/2))*C*a^2*c+1/f/(a^2+b^2)/(2*(c^2+d^
2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c
)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*A*(c^2+d^2)^(1/2)*b-1/4/f/(a^2+b^2)
*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^
2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*b-1/4/f/(a^2+b^2)*ln((c+d*tan
(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2)
)*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*b+1/4/f/(a^2+b^2)*ln((c+d*tan(f*x+e))^(1/
2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*B*(2*(c^2+
d^2)^(1/2)+2*c)^(1/2)*a+1/4/f/(a^2+b^2)*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d
^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2
*c)^(1/2)*b+1/4/f/(a^2+b^2)*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^
2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*b-
1/4/f/(a^2+b^2)*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)
+2*c)^(1/2)+(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*a
```

Fricas [F(-1)]

Timed out.

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx=\text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e)),x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx$$
$$=\int \frac{\sqrt{c+d\tan{(e+fx)}}(A+B\tan{(e+fx)}+C\tan^2{(e+fx)})}{a+b\tan{(e+fx)}}\,dx$$

```
[In] integrate((c+d*tan(f*x+e))**(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e)),x)
```

```
[Out] Integral(sqrt(c + d*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/
(a + b*tan(e + f*x)), x)
```

Maxima [F(-2)]

Exception generated.

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx$$
$$=\text{Exception raised: ValueError}$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e)),x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 detail
```

Giac [F(-1)]

Timed out.

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx=\text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e)),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [B] (verification not implemented)

Time = 33.93 (sec) , antiderivative size = 62245, normalized size of antiderivative = 266.00

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx = \text{Too large to display}$$

```
[In] int(((c + d*tan(e + f*x))^(1/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x)),x)
```

```
[Out] atan((((((((32*(4*C*a*b^8*d^11*f^4 - 4*C*b^9*c*d^10*f^4 + 8*C*a^3*b^6*d^11*f
^4 + 4*C*a^5*b^4*d^11*f^4 - 4*C*b^9*c^3*d^8*f^4 + 4*C*a*b^8*c^2*d^9*f^4 - 8
*C*a^2*b^7*c*d^10*f^4 - 4*C*a^4*b^5*c*d^10*f^4 - 8*C*a^2*b^7*c^3*d^8*f^4 +
8*C*a^3*b^6*c^2*d^9*f^4 - 4*C*a^4*b^5*c^3*d^8*f^4 + 4*C*a^5*b^4*c^2*d^9*f^4
))/(b*f^5) - (32*(c + d*tan(e + f*x))^(1/2)*((((8*C^2*a^2*c*f^2 - 8*C^2*b^2
*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f
^4 + 32*a^2*b^2*f^4))^(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 - 8*C^2*a*b
*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^10*d^10*f^4 +
 16*a^2*b^8*d^10*f^4 - 16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*d^10*f^4 + 24*b^10*
c^2*d^8*f^4 + 40*a^2*b^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^
2*d^8*f^4 + 8*a*b^9*c*d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 24*a^5*b^5*c*d^9*f^4
 + 8*a^7*b^3*c*d^9*f^4))/(b*f^4))*((((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 1
6*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^
2*b^2*f^4))^(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(1
6*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1
/2)*(14*C^2*a*b^7*d^11*f^2 - 2*C^2*a^5*b^3*d^11*f^2 - 10*C^2*b^8*c^3*d^8*f^
2 - 4*C^2*a^3*b^5*d^11*f^2 - 16*C^2*a^7*b*d^11*f^2 + 8*C^2*a^8*c*d^10*f^2 -
 6*C^2*b^8*c*d^10*f^2 + 18*C^2*a*b^7*c^2*d^9*f^2 + 12*C^2*a^2*b^6*c*d^10*f^
2 + 2*C^2*a^4*b^4*c*d^10*f^2 + 24*C^2*a^6*b^2*c*d^10*f^2 - 16*C^2*a^7*b*c^2
*d^9*f^2 + 4*C^2*a^2*b^6*c^3*d^8*f^2 + 4*C^2*a^3*b^5*c^2*d^9*f^2 - 10*C^2*a
^4*b^4*c^3*d^8*f^2 + 2*C^2*a^5*b^3*c^2*d^9*f^2 + 8*C^2*a^6*b^2*c^3*d^8*f^2)
)/(b*f^4))*((((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 -
(C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*C
^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 +
2*a^2*b^2*f^4)))^(1/2) + (32*(15*C^3*a^4*b^3*d^12*f^2 - C^3*a^2*b^5*d^12*f^
2 + C^3*b^7*c^2*d^10*f^2 + C^3*b^7*c^4*d^8*f^2 - 12*C^3*a^6*b*d^12*f^2 - 24
*C^3*a^3*b^4*c*d^11*f^2 + 24*C^3*a^5*b^2*c*d^11*f^2 - 12*C^3*a^6*b*c^2*d^10
*f^2 + 8*C^3*a^2*b^5*c^2*d^10*f^2 + 9*C^3*a^2*b^5*c^4*d^8*f^2 - 24*C^3*a^3*
b^4*c^3*d^9*f^2 + 3*C^3*a^4*b^3*c^2*d^10*f^2 - 12*C^3*a^4*b^3*c^4*d^8*f^2 +
 24*C^3*a^5*b^2*c^3*d^9*f^2))/(b*f^5))*((((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^
2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 +
32*a^2*b^2*f^4))^(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^
2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x
))^(1/2)*(C^4*b^6*d^12 - 2*C^4*a^6*d^12 + 2*C^4*a^6*c^2*d^10 + 2*C^4*b^6*c^
2*d^10 + C^4*b^6*c^4*d^8 - 2*C^4*a^4*b^2*c^2*d^10 + 2*C^4*a^4*b^2*c^4*d^8 +
```

4*C^4*a^5*b*c*d^11 - 4*C^4*a^5*b*c^3*d^9))/(b*f^4))*((((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*1i - (((((32*(4*C*a*b^8*d^11*f^4 - 4*C*b^9*c*d^10*f^4 + 8*C*a^3*b^6*d^11*f^4 + 4*C*a^5*b^4*d^11*f^4 - 4*C*b^9*c^3*d^8*f^4 + 4*C*a*b^8*c^2*d^9*f^4 - 8*C*a^2*b^7*c*d^10*f^4 - 4*C*a^4*b^5*c*d^10*f^4 - 8*C*a^2*b^7*c^3*d^8*f^4 + 8*C*a^3*b^6*c^2*d^9*f^4 - 4*C*a^4*b^5*c^3*d^8*f^4 + 4*C*a^5*b^4*c^2*d^9*f^4))/(b*f^5) + (32*(c + d*tan(e + f*x))^(1/2)*((((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^10*d^10*f^4 + 16*a^2*b^8*d^10*f^4 - 16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*d^10*f^4 + 24*b^10*c^2*d^8*f^4 + 40*a^2*b^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^9*c*d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 24*a^5*b^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9*f^4))/(b*f^4))*((((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(14*C^2*a*b^7*d^11*f^2 - 2*C^2*a^5*b^3*d^11*f^2 - 10*C^2*b^8*c^3*d^8*f^2 - 4*C^2*a^3*b^5*d^11*f^2 - 16*C^2*a^7*b*d^11*f^2 + 8*C^2*a^8*c*d^10*f^2 - 6*C^2*b^8*c*d^10*f^2 + 18*C^2*a*b^7*c^2*d^9*f^2 + 12*C^2*a^2*b^6*c*d^10*f^2 + 2*C^2*a^4*b^4*c*d^10*f^2 + 24*C^2*a^6*b^2*c*d^10*f^2 - 16*C^2*a^7*b*c^2*d^9*f^2 + 4*C^2*a^2*b^6*c^3*d^8*f^2 + 4*C^2*a^3*b^5*c^2*d^9*f^2 - 10*C^2*a^4*b^4*c^3*d^8*f^2 + 2*C^2*a^5*b^3*c^2*d^9*f^2 + 8*C^2*a^6*b^2*c^3*d^8*f^2))/(b*f^4))*((((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(15*C^3*a^4*b^3*d^12*f^2 - C^3*a^2*b^5*d^12*f^2 + C^3*b^7*c^2*d^10*f^2 + C^3*b^7*c^4*d^8*f^2 - 12*C^3*a^6*b*d^12*f^2 - 24*C^3*a^3*b^4*c*d^11*f^2 + 24*C^3*a^5*b^2*c*d^11*f^2 - 12*C^3*a^6*b*c^2*d^10*f^2 + 8*C^3*a^2*b^5*c^2*d^10*f^2 + 9*C^3*a^2*b^5*c^4*d^8*f^2 - 24*C^3*a^3*b^4*c^3*d^9*f^2 + 3*C^3*a^4*b^3*c^2*d^10*f^2 - 12*C^3*a^4*b^3*c^4*d^8*f^2 + 24*C^3*a^5*b^2*c^3*d^9*f^2))/(b*f^5))*((((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(C^4*b^6*d^12 - 2*C^4*a^6*d^12 + 2*C^4*a^6*c^2*d^10 + 2*C^4*b^6*c^2*d^10 + C^4*b^6*c^4*d^8 - 2*C^4*a^4*b^2*c^2*d^10 + 2*C^4*a^4*b^2*c^4*d^8 + 4*C^4*a^5*b*c*d^11 - 4*C^4*a^5*b*c^3*d^9))/(b*f^4))*((((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*1i)/((((((32*(4*C*a*b^8*d^11*f^4 - 4*C*b^9*c*d^10*f^4 + 8*C*a^3*b^6*d^11*f^4 + 4*C*a^5*b^4*d^11*f^4 - 4*C*b^9*c^3*d^8*f^4 + 4*C*a*b^8*c^2*d^9*f^4 - 8*C*a^2*b^7*c*d^10*f^4

- 4*C*a^4*b^5*c*d^10*f^4 - 8*C*a^2*b^7*c^3*d^8*f^4 + 8*C*a^3*b^6*c^2*d^9*f^
4 - 4*C*a^4*b^5*c^3*d^8*f^4 + 4*C*a^5*b^4*c^2*d^9*f^4))/(b*f^5) - (32*(c +
d*tan(e + f*x))^(1/2)*((((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*
f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^
(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 +
b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^10*d^10*f^4 + 16*a^2*b^8*d^10*f^4 -
16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*d^10*f^4 + 24*b^10*c^2*d^8*f^4 + 40*a^2*b
^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^9*c*
d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 24*a^5*b^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9*f^4
))/(b*f^4))*((((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 -
(C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*
C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 +
2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(14*C^2*a*b^7*d^11
*f^2 - 2*C^2*a^5*b^3*d^11*f^2 - 10*C^2*b^8*c^3*d^8*f^2 - 4*C^2*a^3*b^5*d^11
*f^2 - 16*C^2*a^7*b*d^11*f^2 + 8*C^2*a^8*c*d^10*f^2 - 6*C^2*b^8*c*d^10*f^2
+ 18*C^2*a*b^7*c^2*d^9*f^2 + 12*C^2*a^2*b^6*c*d^10*f^2 + 2*C^2*a^4*b^4*c*d^
10*f^2 + 24*C^2*a^6*b^2*c*d^10*f^2 - 16*C^2*a^7*b*c^2*d^9*f^2 + 4*C^2*a^2*b
^6*c^3*d^8*f^2 + 4*C^2*a^3*b^5*c^2*d^9*f^2 - 10*C^2*a^4*b^4*c^3*d^8*f^2 + 2
*C^2*a^5*b^3*c^2*d^9*f^2 + 8*C^2*a^6*b^2*c^3*d^8*f^2))/(b*f^4))*((((8*C^2*a
^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(1
6*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b
^2*c*f^2 - 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)
+ (32*(15*C^3*a^4*b^3*d^12*f^2 - C^3*a^2*b^5*d^12*f^2 + C^3*b^7*c^2*d^10*f
^2 + C^3*b^7*c^4*d^8*f^2 - 12*C^3*a^6*b*d^12*f^2 - 24*C^3*a^3*b^4*c*d^11*f^
2 + 24*C^3*a^5*b^2*c*d^11*f^2 - 12*C^3*a^6*b*c^2*d^10*f^2 + 8*C^3*a^2*b^5*c
^2*d^10*f^2 + 9*C^3*a^2*b^5*c^4*d^8*f^2 - 24*C^3*a^3*b^4*c^3*d^9*f^2 + 3*C^
3*a^4*b^3*c^2*d^10*f^2 - 12*C^3*a^4*b^3*c^4*d^8*f^2 + 24*C^3*a^5*b^2*c^3*d^
9*f^2))/(b*f^5))*((((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^
2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2)
- 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*
f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(C^4*b^6*d^12
- 2*C^4*a^6*d^12 + 2*C^4*a^6*c^2*d^10 + 2*C^4*b^6*c^2*d^10 + C^4*b^6*c^4*d
^8 - 2*C^4*a^4*b^2*c^2*d^10 + 2*C^4*a^4*b^2*c^4*d^8 + 4*C^4*a^5*b*c*d^11 -
4*C^4*a^5*b*c^3*d^9))/(b*f^4))*((((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C
^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b
^2*f^4))^(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(16*(
a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (((((32*(4*C*a*b^8*d^11*f^4 -
4*C*b^9*c*d^10*f^4 + 8*C*a^3*b^6*d^11*f^4 + 4*C*a^5*b^4*d^11*f^4 - 4*C*b^9*
c^3*d^8*f^4 + 4*C*a*b^8*c^2*d^9*f^4 - 8*C*a^2*b^7*c*d^10*f^4 - 4*C*a^4*b^5*
c*d^10*f^4 - 8*C*a^2*b^7*c^3*d^8*f^4 + 8*C*a^3*b^6*c^2*d^9*f^4 - 4*C*a^4*b^
5*c^3*d^8*f^4 + 4*C*a^5*b^4*c^2*d^9*f^4))/(b*f^5) + (32*(c + d*tan(e + f*x)
)^(1/2)*((((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^
4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*C^2*
a^2*c*f^2 + 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a
^2*b^2*f^4)))^(1/2)*(16*b^10*d^10*f^4 + 16*a^2*b^8*d^10*f^4 - 16*a^4*b^6*d^

10*f^4 - 16*a^6*b^4*d^10*f^4 + 24*b^10*c^2*d^8*f^4 + 40*a^2*b^8*c^2*d^8*f^4
 + 8*a^4*b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^9*c*d^9*f^4 + 24*a
^3*b^7*c*d^9*f^4 + 24*a^5*b^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9*f^4))/(b*f^4))*((
((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^
4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*C^2*a^2*c*f^2
 + 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4
)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(14*C^2*a*b^7*d^11*f^2 - 2*C^2*a
^5*b^3*d^11*f^2 - 10*C^2*b^8*c^3*d^8*f^2 - 4*C^2*a^3*b^5*d^11*f^2 - 16*C^2*
a^7*b*d^11*f^2 + 8*C^2*a^8*c*d^10*f^2 - 6*C^2*b^8*c*d^10*f^2 + 18*C^2*a*b^7
*c^2*d^9*f^2 + 12*C^2*a^2*b^6*c*d^10*f^2 + 2*C^2*a^4*b^4*c*d^10*f^2 + 24*C^
2*a^6*b^2*c*d^10*f^2 - 16*C^2*a^7*b*c^2*d^9*f^2 + 4*C^2*a^2*b^6*c^3*d^8*f^2
 + 4*C^2*a^3*b^5*c^2*d^9*f^2 - 10*C^2*a^4*b^4*c^3*d^8*f^2 + 2*C^2*a^5*b^3*c
^2*d^9*f^2 + 8*C^2*a^6*b^2*c^3*d^8*f^2))/(b*f^4))*((((8*C^2*a^2*c*f^2 - 8*C
^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16
*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 - 8*C
^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(15*C^3
*a^4*b^3*d^12*f^2 - C^3*a^2*b^5*d^12*f^2 + C^3*b^7*c^2*d^10*f^2 + C^3*b^7*c
^4*d^8*f^2 - 12*C^3*a^6*b*d^12*f^2 - 24*C^3*a^3*b^4*c*d^11*f^2 + 24*C^3*a^5
*b^2*c*d^11*f^2 - 12*C^3*a^6*b*c^2*d^10*f^2 + 8*C^3*a^2*b^5*c^2*d^10*f^2 +
9*C^3*a^2*b^5*c^4*d^8*f^2 - 24*C^3*a^3*b^4*c^3*d^9*f^2 + 3*C^3*a^4*b^3*c^2*
d^10*f^2 - 12*C^3*a^4*b^3*c^4*d^8*f^2 + 24*C^3*a^5*b^2*c^3*d^9*f^2))/(b*f^5
))*((((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2
 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*C^2*a^2*c
*f^2 + 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^
2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(C^4*b^6*d^12 - 2*C^4*a^6*d
^12 + 2*C^4*a^6*c^2*d^10 + 2*C^4*b^6*c^2*d^10 + C^4*b^6*c^4*d^8 - 2*C^4*a^4
*b^2*c^2*d^10 + 2*C^4*a^4*b^2*c^4*d^8 + 4*C^4*a^5*b*c*d^11 - 4*C^4*a^5*b*c^
3*d^9))/(b*f^4))*((((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^
2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2)
 - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*
f^4 + 2*a^2*b^2*f^4)))^(1/2) - (64*(C^5*a^5*d^13 - C^5*a^3*b^2*d^13 + C^5*a
^5*c^2*d^11 + 2*C^5*a^2*b^3*c^3*d^10 + C^5*a^2*b^3*c^5*d^8 - 2*C^5*a^3*b^2*
c^2*d^11 - C^5*a^3*b^2*c^4*d^9 - C^5*a^4*b*c*d^12 + C^5*a^2*b^3*c*d^12 - C^
5*a^4*b*c^3*d^10))/(b*f^5)))*((((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2
*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2
*f^4))^(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(16*(a^
4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*2i - atan(((((32*(15*B^3*a^3*b^3*d
^12*f^2 + B^3*b^6*c^3*d^9*f^2 - B^3*a*b^5*d^12*f^2 - 4*B^3*a^5*b*d^12*f^2 +
 B^3*b^6*c*d^11*f^2 + 6*B^3*a*b^5*c^2*d^10*f^2 + 7*B^3*a*b^5*c^4*d^8*f^2 -
22*B^3*a^2*b^4*c*d^11*f^2 + 9*B^3*a^4*b^2*c*d^11*f^2 - 4*B^3*a^5*b*c^2*d^10
*f^2 - 22*B^3*a^2*b^4*c^3*d^9*f^2 + 10*B^3*a^3*b^3*c^2*d^10*f^2 - 5*B^3*a^3
*b^3*c^4*d^8*f^2 + 9*B^3*a^4*b^2*c^3*d^9*f^2))/f^5 - (((32*(4*B*a^2*b^6*d^1
1*f^4 + 8*B*a^4*b^4*d^11*f^4 + 4*B*a^6*b^2*d^11*f^4 - 4*B*a*b^7*c^3*d^8*f^4
 - 8*B*a^3*b^5*c*d^10*f^4 - 4*B*a^5*b^3*c*d^10*f^4 + 4*B*a^2*b^6*c^2*d^9*f^
4 - 8*B*a^3*b^5*c^3*d^8*f^4 + 8*B*a^4*b^4*c^2*d^9*f^4 - 4*B*a^5*b^3*c^3*d^8

*f^4 + 4*B*a^6*b^2*c^2*d^9*f^4 - 4*B*a*b^7*c*d^10*f^4))/f^5 - (32*(c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/f^4)*((((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(14*B^2*a^5*b^2*d^11*f^2 - 4*B^2*a^3*b^4*d^11*f^2 - 10*B^2*b^7*c^3*d^8*f^2 + 14*B^2*a*b^6*d^11*f^2 - 6*B^2*b^7*c*d^10*f^2 - 8*B^2*a^6*b*c*d^10*f^2 + 18*B^2*a*b^6*c^2*d^9*f^2 + 12*B^2*a^2*b^5*c*d^10*f^2 - 22*B^2*a^4*b^3*c*d^10*f^2 + 12*B^2*a^2*b^5*c^3*d^8*f^2 + 4*B^2*a^3*b^4*c^2*d^9*f^2 - 10*B^2*a^4*b^3*c^3*d^8*f^2 + 18*B^2*a^5*b^2*c^2*d^9*f^2))/f^4)*((((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2))*((((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(B^4*b^5*d^12 + 2*B^4*b^5*c^2*d^10 + B^4*b^5*c^4*d^8 + 2*B^4*a^4*b*d^12 + 2*B^4*a^2*b^3*c^2*d^10 - 2*B^4*a^2*b^3*c^4*d^8 + 4*B^4*a^3*b^2*c^3*d^9 - 4*B^4*a^3*b^2*c*d^11 - 2*B^4*a^4*b*c^2*d^10))/f^4)*((((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*1i - (((32*(15*B^3*a^3*b^3*d^12*f^2 + B^3*b^6*c^3*d^9*f^2 - B^3*a*b^5*d^12*f^2 - 4*B^3*a^5*b*d^12*f^2 + B^3*b^6*c*d^11*f^2 + 6*B^3*a*b^5*c^2*d^10*f^2 + 7*B^3*a*b^5*c^4*d^8*f^2 - 22*B^3*a^2*b^4*c*d^11*f^2 + 9*B^3*a^4*b^2*c*d^11*f^2 - 4*B^3*a^5*b*c^2*d^10*f^2 - 22*B^3*a^2*b^4*c^3*d^9*f^2 + 10*B^3*a^3*b^3*c^2*d^10*f^2 - 5*B^3*a^3*b^3*c^4*d^8*f^2 + 9*B^3*a^4*b^2*c^3*d^9*f^2))/f^5 - (((32*(4*B*a^2*b^6*d^11*f^4 + 8*B*a^4*b^4*d^11*f^4 + 4*B*a^6*b^2*d^11*f^4 - 4*B*a*b^7*c^3*d^8*f^4 - 8*B*a^3*b^5*c*d^10*f^4 - 4*B*a^5*b^3*c*d^10*f^4 + 4*B*a^2*b^6*c^2*d^9*f^4 - 8*B*a^3*b^5*c^3*d^8*f^4 + 8*B*a^4*b^4*c^2*d^9*f^4 - 4*B*a^5*b^3*c^3*d^8*f^4 + 4*B*a^6*b^2*c^2*d^9*f^4 - 4*B*a*b^7*c*d^10*f^4))/f^5 + (32*(c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7 *b^2*c*d^9*f^4))/f^4)*((((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(14*B^2*a^5*b^2*d^11*f^2 - 4*B^2*a^3*b^4*d^11*f^2 - 10*B^2*b^7*c^3*d^8*f^2 + 14*B^2*a*b^6*d^11*f^2 - 6*B^2*b^7*c*d^10*f^2 - 8*B^2*a^6*b*c*d^10*f^2 + 18*B^2*a*b^6*c^2*d^9*f^2 + 12*B^2*a^2*b^5*c*d^10*f^2 - 22*B^2*a^4*b^3*c*d^10*f^2 + 12*B^2*a^2*b^5*c^3*d^8*f^2 + 4*B^2*a^3*b^4*c^2*d^9*f^2 - 10*B^2*a^4*b^3*c^3*d^8*f^2 + 18*B^2*a^5*b^2*c^2*d^9*f^2))/f^4)*((((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2))*((((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(B^4*b^5*d^12 + 2*B^4*b^5*c^2*d^10 + B^4*b^5*c^4*d^8 + 2*B^4*a^4*b*d^12 + 2*B^4*a^2*b^3*c^2*d^10 - 2*B^4*a^2*b^3*c^4*d^8 + 4*B^4*a^3*b^2*c^3*d^9 - 4*B^4*a^3*b^2*c*d^11 - 2*B^4*a^4*b*c^2*d^10))/f^4)*((((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*1i)/((((32*(15*B^3*a^3*b^3*d^12*f^2 + B^3*b^6*c^3*d^9*f^2 - B^3*a*b^5*d^12*f^2 - 4*B^3*a^5*b*d^12*f^2 + B^3*b^6*c*d^11*f^2 + 6*B^3*a*b^5*c^2*d^10*f^2 + 7*B^3*a*b^5*c^4*d^8*f^2 - 22*B^3*a^2*b^4*c*d^11*f^2 + 9*B^3*a^4*b^2*c*d^11*f^2 - 4*B^3*a^5*b*c^2*d^10*f^2 - 22*B^3*a^2*b^4*c^3*d^9*f^2 + 10*B^3*a^3*b^3*c^2*d^10*f^2 - 5*B^3*a^3*b^3*c^4*d^8*f^2 + 9*B^3*a^4*b^2*c^3*d^9*f^2))/f^5 - (((32*(4*B*a^2*b^6*d^11*f^4 + 8*B*a^4*b^4*d^11*f^4 + 4*B*a^6*b^2*d^11*f^4 - 4*B*a*b^7*c^3*d^8*f^4 - 8*B*a^3*b^5*c*d^10*f^4 - 4*B*a^5*b^3*c*d^10*f^4 + 4*B*a^2*b^6*c^2*d^9*f^4 - 8*B*a^3*b^5*c^3*d^8*f^4 + 8*B*a^4*b^4*c^2*d^9*f^4 - 4*B*a^5*b^3*c^3*d^8*f^4 + 4*B*a^6*b^2*c^2*d^9*f^4 - 4*B*a*b^7*c*d^10*f^4))/f^5 - (32*(c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/f^4)*((((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(14*B^2*a^5*b^2*d^11*f^2 - 4*B^2*a^3*b^4*d^11*f^2 - 10*B^2*b^7*c^3*d^8*f^2 + 14*B^2*a*b^6*d^11*f^2 - 6*B^2*b^7*c*d^10*f^2 - 8*B^2*a^6*b*c*d^10*f^2 + 18*B^2*a*b^6*c^2*d^9*f^2 + 12*B^2*a^2*b^5*c*d^10*f^2 - 22*B^2*a^4*b^3*c*d^10*f^2 + 12*B^2*a^2*b^5*c^3*d^8*f^2 + 4*B^2*a^3*b^4*c^2*d^9*f^2 -

10*B^2*a^4*b^3*c^3*d^8*f^2 + 18*B^2*a^5*b^2*c^2*d^9*f^2))/f^4)*((((8*B^2*a
^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(1
6*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b
^2*c*f^2 + 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)
)*((((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2
+ B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*B^2*a^2*c*
f^2 - 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2
f^4)))^(1/2) + (32(c + d*tan(e + f*x))^(1/2)*(B^4*b^5*d^12 + 2*B^4*b^5*c^
2*d^10 + B^4*b^5*c^4*d^8 + 2*B^4*a^4*b*d^12 + 2*B^4*a^2*b^3*c^2*d^10 - 2*B^
4*a^2*b^3*c^4*d^8 + 4*B^4*a^3*b^2*c^3*d^9 - 4*B^4*a^3*b^2*c*d^11 - 2*B^4*a^
4*b*c^2*d^10))/f^4)*((((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^
2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1
/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b
^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (((32*(15*B^3*a^3*b^3*d^12*f^2 + B^3*b^6*
c^3*d^9*f^2 - B^3*a*b^5*d^12*f^2 - 4*B^3*a^5*b*d^12*f^2 + B^3*b^6*c*d^11*f^
2 + 6*B^3*a*b^5*c^2*d^10*f^2 + 7*B^3*a*b^5*c^4*d^8*f^2 - 22*B^3*a^2*b^4*c*d
^11*f^2 + 9*B^3*a^4*b^2*c*d^11*f^2 - 4*B^3*a^5*b*c^2*d^10*f^2 - 22*B^3*a^2*
b^4*c^3*d^9*f^2 + 10*B^3*a^3*b^3*c^2*d^10*f^2 - 5*B^3*a^3*b^3*c^4*d^8*f^2 +
9*B^3*a^4*b^2*c^3*d^9*f^2))/f^5 - (((32*(4*B*a^2*b^6*d^11*f^4 + 8*B*a^4*b^
4*d^11*f^4 + 4*B*a^6*b^2*d^11*f^4 - 4*B*a*b^7*c^3*d^8*f^4 - 8*B*a^3*b^5*c*d
^10*f^4 - 4*B*a^5*b^3*c*d^10*f^4 + 4*B*a^2*b^6*c^2*d^9*f^4 - 8*B*a^3*b^5*c^
3*d^8*f^4 + 8*B*a^4*b^4*c^2*d^9*f^4 - 4*B*a^5*b^3*c^3*d^8*f^4 + 4*B*a^6*b^2
*c^2*d^9*f^4 - 4*B*a*b^7*c*d^10*f^4))/f^5 + (32*(c + d*tan(e + f*x))^(1/2)*
((((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 +
B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*B^2*a^2*c*f^
2 - 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f
^4)))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 -
16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b
^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d
^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/f^4)*((((8*B^2*a^2*c*
f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4
*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*
f^2 + 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (3
2*(c + d*tan(e + f*x))^(1/2)*(14*B^2*a^5*b^2*d^11*f^2 - 4*B^2*a^3*b^4*d^11*
f^2 - 10*B^2*b^7*c^3*d^8*f^2 + 14*B^2*a*b^6*d^11*f^2 - 6*B^2*b^7*c*d^10*f^2
- 8*B^2*a^6*b*c*d^10*f^2 + 18*B^2*a*b^6*c^2*d^9*f^2 + 12*B^2*a^2*b^5*c*d^1
0*f^2 - 22*B^2*a^4*b^3*c*d^10*f^2 + 12*B^2*a^2*b^5*c^3*d^8*f^2 + 4*B^2*a^3*
b^4*c^2*d^9*f^2 - 10*B^2*a^4*b^3*c^3*d^8*f^2 + 18*B^2*a^5*b^2*c^2*d^9*f^2))
/f^4)*((((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*
c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*B^2*a^
2*c*f^2 - 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2
*b^2*f^4)))^(1/2))*((((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2
)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/
2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^
4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(B^4*b^5*d^

12 + 2*B^4*b^5*c^2*d^10 + B^4*b^5*c^4*d^8 + 2*B^4*a^4*b*d^12 + 2*B^4*a^2*b^
3*c^2*d^10 - 2*B^4*a^2*b^3*c^4*d^8 + 4*B^4*a^3*b^2*c^3*d^9 - 4*B^4*a^3*b^2*
c*d^11 - 2*B^4*a^4*b*c^2*d^10))/f^4)*((((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2
+ 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32
*a^2*b^2*f^4))^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)
/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (64*(B^5*a*b^3*c*d^12 -
3*B^5*a^2*b^2*c^2*d^11 - 2*B^5*a^2*b^2*c^4*d^9 - B^5*a^2*b^2*d^13 + B^5*a^3
*b*c*d^12 + 2*B^5*a*b^3*c^3*d^10 + B^5*a*b^3*c^5*d^8 + B^5*a^3*b*c^3*d^10))
/f^5))*((((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4
*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*B^2*a
^2*c*f^2 - 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^
2*b^2*f^4)))^(1/2)*2i - atan(((((32*(15*B^3*a^3*b^3*d^12*f^2 + B^3*b^6*c^3*
d^9*f^2 - B^3*a*b^5*d^12*f^2 - 4*B^3*a^5*b*d^12*f^2 + B^3*b^6*c*d^11*f^2 +
6*B^3*a*b^5*c^2*d^10*f^2 + 7*B^3*a*b^5*c^4*d^8*f^2 - 22*B^3*a^2*b^4*c*d^11*
f^2 + 9*B^3*a^4*b^2*c*d^11*f^2 - 4*B^3*a^5*b*c^2*d^10*f^2 - 22*B^3*a^2*b^4*
c^3*d^9*f^2 + 10*B^3*a^3*b^3*c^2*d^10*f^2 - 5*B^3*a^3*b^3*c^4*d^8*f^2 + 9*B
^3*a^4*b^2*c^3*d^9*f^2))/f^5 - (((32*(4*B*a^2*b^6*d^11*f^4 + 8*B*a^4*b^4*d^
11*f^4 + 4*B*a^6*b^2*d^11*f^4 - 4*B*a*b^7*c^3*d^8*f^4 - 8*B*a^3*b^5*c*d^10*
f^4 - 4*B*a^5*b^3*c*d^10*f^4 + 4*B*a^2*b^6*c^2*d^9*f^4 - 8*B*a^3*b^5*c^3*d^
8*f^4 + 8*B*a^4*b^4*c^2*d^9*f^4 - 4*B*a^5*b^3*c^3*d^8*f^4 + 4*B*a^6*b^2*c^2
*d^9*f^4 - 4*B*a*b^7*c*d^10*f^4))/f^5 - (32*(c + d*tan(e + f*x))^(1/2)*(-((
(8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4
d^2)(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*B^2*a^2*c*f^2 +
4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)
))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*
a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*
c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*
f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/f^4)*(-(((8*B^2*a^2*c*f^
2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f
^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^
2 - 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*
(c + d*tan(e + f*x))^(1/2)*(14*B^2*a^5*b^2*d^11*f^2 - 4*B^2*a^3*b^4*d^11*f^
2 - 10*B^2*b^7*c^3*d^8*f^2 + 14*B^2*a*b^6*d^11*f^2 - 6*B^2*b^7*c*d^10*f^2 -
8*B^2*a^6*b*c*d^10*f^2 + 18*B^2*a*b^6*c^2*d^9*f^2 + 12*B^2*a^2*b^5*c*d^10*
f^2 - 22*B^2*a^4*b^3*c*d^10*f^2 + 12*B^2*a^2*b^5*c^3*d^8*f^2 + 4*B^2*a^3*b^
4*c^2*d^9*f^2 - 10*B^2*a^4*b^3*c^3*d^8*f^2 + 18*B^2*a^5*b^2*c^2*d^9*f^2))/f
^4)*(-(((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c
^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*B^2*a^2
*c*f^2 + 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*
b^2*f^4)))^(1/2))*(-(((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2
)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/
2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^
4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(B^4*b^5*d^
12 + 2*B^4*b^5*c^2*d^10 + B^4*b^5*c^4*d^8 + 2*B^4*a^4*b*d^12 + 2*B^4*a^2*b^
3*c^2*d^10 - 2*B^4*a^2*b^3*c^4*d^8 + 4*B^4*a^3*b^2*c^3*d^9 - 4*B^4*a^3*b^2*

c*d^11 - 2*B^4*a^4*b*c^2*d^10))/f^4)*(-(((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*1i - (((32*(15*B^3*a^3*b^3*d^12*f^2 + B^3*b^6*c^3*d^9*f^2 - B^3*a*b^5*d^12*f^2 - 4*B^3*a^5*b*d^12*f^2 + B^3*b^6*c*d^11*f^2 + 6*B^3*a*b^5*c^2*d^10*f^2 + 7*B^3*a*b^5*c^4*d^8*f^2 - 22*B^3*a^2*b^4*c*d^11*f^2 + 9*B^3*a^4*b^2*c*d^11*f^2 - 4*B^3*a^5*b*c^2*d^10*f^2 - 22*B^3*a^2*b^4*c^3*d^9*f^2 + 10*B^3*a^3*b^3*c^2*d^10*f^2 - 5*B^3*a^3*b^3*c^4*d^8*f^2 + 9*B^3*a^4*b^2*c^3*d^9*f^2))/f^5 - (((32*(4*B*a^2*b^6*d^11*f^4 + 8*B*a^4*b^4*d^11*f^4 + 4*B*a^6*b^2*d^11*f^4 - 4*B*a*b^7*c^3*d^8*f^4 - 8*B*a^3*b^5*c*d^10*f^4 - 4*B*a^5*b^3*c*d^10*f^4 + 4*B*a^2*b^6*c^2*d^9*f^4 - 8*B*a^3*b^5*c^3*d^8*f^4 + 8*B*a^4*b^4*c^2*d^9*f^4 - 4*B*a^5*b^3*c^3*d^8*f^4 + 4*B*a^6*b^2*c^2*d^9*f^4 - 4*B*a*b^7*c*d^10*f^4))/f^5 + (32*(c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/f^4)*(-(((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(14*B^2*a^5*b^2*d^11*f^2 - 4*B^2*a^3*b^4*d^11*f^2 - 10*B^2*b^7*c^3*d^8*f^2 + 14*B^2*a*b^6*d^11*f^2 - 6*B^2*b^7*c*d^10*f^2 - 8*B^2*a^6*b*c*d^10*f^2 + 18*B^2*a*b^6*c^2*d^9*f^2 + 12*B^2*a^2*b^5*c*d^10*f^2 - 22*B^2*a^4*b^3*c*d^10*f^2 + 12*B^2*a^2*b^5*c^3*d^8*f^2 + 4*B^2*a^3*b^4*c^2*d^9*f^2 - 10*B^2*a^4*b^3*c^3*d^8*f^2 + 18*B^2*a^5*b^2*c^2*d^9*f^2))/f^4)*(-(((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2))*(-(((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(B^4*b^5*d^12 + 2*B^4*b^5*c^2*d^10 + B^4*b^5*c^4*d^8 + 2*B^4*a^4*b*d^12 + 2*B^4*a^2*b^3*c^2*d^10 - 2*B^4*a^2*b^3*c^4*d^8 + 4*B^4*a^3*b^2*c^3*d^9 - 4*B^4*a^3*b^2*c*d^11 - 2*B^4*a^4*b*c^2*d^10))/f^4)*(-(((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*1i)/((((32*(15*B^3*a^3*b^3*d^12*f^2 + B^3*b^6*c^3*d^9*f^2 - B^3*a*b^5*d^12*f^2 - 4*B^3*a^5*b*d^12*f^2 + B^3*b^6*c*d^11*f^2 + 6*B^3*a*b^5*c^2*d^10*f^2 + 7*B^3*a*b^5*c^4*d^8*f^2 - 22*B^3*a^2*b^4*c*d^11*f^2 + 9*B^3*a^4*b^2*c*d^11*f^2 - 4*B^3*a^5*b*c^2*d^10*f^2 - 22*B^3*a^2*b^4*c^3*d^9*f^2 + 10*B^3*a^3*

b^3*c^2*d^10*f^2 - 5*B^3*a^3*b^3*c^4*d^8*f^2 + 9*B^3*a^4*b^2*c^3*d^9*f^2))/
f^5 - (((32*(4*B*a^2*b^6*d^11*f^4 + 8*B*a^4*b^4*d^11*f^4 + 4*B*a^6*b^2*d^11
*f^4 - 4*B*a*b^7*c^3*d^8*f^4 - 8*B*a^3*b^5*c*d^10*f^4 - 4*B*a^5*b^3*c*d^10*
f^4 + 4*B*a^2*b^6*c^2*d^9*f^4 - 8*B*a^3*b^5*c^3*d^8*f^4 + 8*B*a^4*b^4*c^2*d
^9*f^4 - 4*B*a^5*b^3*c^3*d^8*f^4 + 4*B*a^6*b^2*c^2*d^9*f^4 - 4*B*a*b^7*c*d^
10*f^4))/f^5 - (32*(c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^2*c*f^2 - 8*B^2*
b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^
4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 - 8*B^2*
a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^9*d^10*f^4
 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9
*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c
^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^
4 + 8*a^7*b^2*c*d^9*f^4))/f^4)*(-(((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*
B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*
b^2*f^4))^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(16*
(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2
)*(14*B^2*a^5*b^2*d^11*f^2 - 4*B^2*a^3*b^4*d^11*f^2 - 10*B^2*b^7*c^3*d^8*f^
2 + 14*B^2*a*b^6*d^11*f^2 - 6*B^2*b^7*c*d^10*f^2 - 8*B^2*a^6*b*c*d^10*f^2 +
 18*B^2*a*b^6*c^2*d^9*f^2 + 12*B^2*a^2*b^5*c*d^10*f^2 - 22*B^2*a^4*b^3*c*d^
10*f^2 + 12*B^2*a^2*b^5*c^3*d^8*f^2 + 4*B^2*a^3*b^4*c^2*d^9*f^2 - 10*B^2*a^
4*b^3*c^3*d^8*f^2 + 18*B^2*a^5*b^2*c^2*d^9*f^2))/f^4)*(-(((8*B^2*a^2*c*f^2
- 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4
 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2
- 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2))*(-(((8*
B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^
2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*B^2*a^2*c*f^2 + 4*
B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^
(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(B^4*b^5*d^12 + 2*B^4*b^5*c^2*d^10 +
 B^4*b^5*c^4*d^8 + 2*B^4*a^4*b*d^12 + 2*B^4*a^2*b^3*c^2*d^10 - 2*B^4*a^2*b^
3*c^4*d^8 + 4*B^4*a^3*b^2*c^3*d^9 - 4*B^4*a^3*b^2*c*d^11 - 2*B^4*a^4*b*c^2*
d^10))/f^4)*(-(((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4
- (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4
*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4
+ 2*a^2*b^2*f^4)))^(1/2) + (((32*(15*B^3*a^3*b^3*d^12*f^2 + B^3*b^6*c^3*d^9
*f^2 - B^3*a*b^5*d^12*f^2 - 4*B^3*a^5*b*d^12*f^2 + B^3*b^6*c*d^11*f^2 + 6*B
^3*a*b^5*c^2*d^10*f^2 + 7*B^3*a*b^5*c^4*d^8*f^2 - 22*B^3*a^2*b^4*c*d^11*f^2
 + 9*B^3*a^4*b^2*c*d^11*f^2 - 4*B^3*a^5*b*c^2*d^10*f^2 - 22*B^3*a^2*b^4*c^3
*d^9*f^2 + 10*B^3*a^3*b^3*c^2*d^10*f^2 - 5*B^3*a^3*b^3*c^4*d^8*f^2 + 9*B^3*
a^4*b^2*c^3*d^9*f^2))/f^5 - (((32*(4*B*a^2*b^6*d^11*f^4 + 8*B*a^4*b^4*d^11*
f^4 + 4*B*a^6*b^2*d^11*f^4 - 4*B*a*b^7*c^3*d^8*f^4 - 8*B*a^3*b^5*c*d^10*f^4
 - 4*B*a^5*b^3*c*d^10*f^4 + 4*B*a^2*b^6*c^2*d^9*f^4 - 8*B*a^3*b^5*c^3*d^8*f
^4 + 8*B*a^4*b^4*c^2*d^9*f^4 - 4*B*a^5*b^3*c^3*d^8*f^4 + 4*B*a^6*b^2*c^2*d^
9*f^4 - 4*B*a*b^7*c*d^10*f^4))/f^5 + (32*(c + d*tan(e + f*x))^(1/2)*(-(((8*
B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^
2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*B^2*a^2*c*f^2 + 4*

B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/f^4)*(-(((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(14*B^2*a^5*b^2*d^11*f^2 - 4*B^2*a^3*b^4*d^11*f^2 - 10*B^2*b^7*c^3*d^8*f^2 + 14*B^2*a*b^6*d^11*f^2 - 6*B^2*b^7*c*d^10*f^2 - 8*B^2*a^6*b*c*d^10*f^2 + 18*B^2*a*b^6*c^2*d^9*f^2 + 12*B^2*a^2*b^5*c*d^10*f^2 - 22*B^2*a^4*b^3*c*d^10*f^2 + 12*B^2*a^2*b^5*c^3*d^8*f^2 + 4*B^2*a^3*b^4*c^2*d^9*f^2 - 10*B^2*a^4*b^3*c^3*d^8*f^2 + 18*B^2*a^5*b^2*c^2*d^9*f^2))/f^4)*(-(((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2))*(-(((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(B^4*b^5*d^12 + 2*B^4*b^5*c^2*d^10 + B^4*b^5*c^4*d^8 + 2*B^4*a^4*b*d^12 + 2*B^4*a^2*b^3*c^2*d^10 - 2*B^4*a^2*b^3*c^4*d^8 + 4*B^4*a^3*b^2*c^3*d^9 - 4*B^4*a^3*b^2*c*d^11 - 2*B^4*a^4*b*c^2*d^10))/f^4)*(-(((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (64*(B^5*a*b^3*c*d^12 - 3*B^5*a^2*b^2*c^2*d^11 - 2*B^5*a^2*b^2*c^4*d^9 - B^5*a^2*b^2*d^13 + B^5*a^3*b*c*d^12 + 2*B^5*a*b^3*c^3*d^10 + B^5*a*b^3*c^5*d^8 + B^5*a^3*b*c^3*d^10))/f^5))*(-(((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*2i + atan((((((32*(4*C*a*b^8*d^11*f^4 - 4*C*b^9*c*d^10*f^4 + 8*C*a^3*b^6*d^11*f^4 + 4*C*a^5*b^4*d^11*f^4 - 4*C*b^9*c^3*d^8*f^4 + 4*C*a*b^8*c^2*d^9*f^4 - 8*C*a^2*b^7*c*d^10*f^4 - 4*C*a^4*b^5*c*d^10*f^4 - 8*C*a^2*b^7*c^3*d^8*f^4 + 8*C*a^3*b^6*c^2*d^9*f^4 - 4*C*a^4*b^5*c^3*d^8*f^4 + 4*C*a^5*b^4*c^2*d^9*f^4))/(b*f^5) - (32*(c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^10*d^10*f^4 + 16*a^2*b^8*d^10*f^4 - 16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*d^10*f^4 + 24*b^10*c^2*d^8*f^4 + 40*a^2*b^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^9*c*d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 24*a^5*b^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9*f^4))/(b*f^4))*(-(((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*

f^2 + 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (3
2*(c + d*tan(e + f*x))^(1/2)*(14*C^2*a*b^7*d^11*f^2 - 2*C^2*a^5*b^3*d^11*f^
2 - 10*C^2*b^8*c^3*d^8*f^2 - 4*C^2*a^3*b^5*d^11*f^2 - 16*C^2*a^7*b*d^11*f^2
 + 8*C^2*a^8*c*d^10*f^2 - 6*C^2*b^8*c*d^10*f^2 + 18*C^2*a*b^7*c^2*d^9*f^2 +
 12*C^2*a^2*b^6*c*d^10*f^2 + 2*C^2*a^4*b^4*c*d^10*f^2 + 24*C^2*a^6*b^2*c*d^
10*f^2 - 16*C^2*a^7*b*c^2*d^9*f^2 + 4*C^2*a^2*b^6*c^3*d^8*f^2 + 4*C^2*a^3*b
^5*c^2*d^9*f^2 - 10*C^2*a^4*b^4*c^3*d^8*f^2 + 2*C^2*a^5*b^3*c^2*d^9*f^2 + 8
*C^2*a^6*b^2*c^3*d^8*f^2))/(b*f^4))*(-(((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2
+ 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32
*a^2*b^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)
/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(15*C^3*a^4*b^3*d^12
*f^2 - C^3*a^2*b^5*d^12*f^2 + C^3*b^7*c^2*d^10*f^2 + C^3*b^7*c^4*d^8*f^2 -
12*C^3*a^6*b*d^12*f^2 - 24*C^3*a^3*b^4*c*d^11*f^2 + 24*C^3*a^5*b^2*c*d^11*f
^2 - 12*C^3*a^6*b*c^2*d^10*f^2 + 8*C^3*a^2*b^5*c^2*d^10*f^2 + 9*C^3*a^2*b^5
*c^4*d^8*f^2 - 24*C^3*a^3*b^4*c^3*d^9*f^2 + 3*C^3*a^4*b^3*c^2*d^10*f^2 - 12
*C^3*a^4*b^3*c^4*d^8*f^2 + 24*C^3*a^5*b^2*c^3*d^9*f^2))/(b*f^5))*(-(((8*C^2
*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*
(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2
*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/
2) - (32*(c + d*tan(e + f*x))^(1/2)*(C^4*b^6*d^12 - 2*C^4*a^6*d^12 + 2*C^4*
a^6*c^2*d^10 + 2*C^4*b^6*c^2*d^10 + C^4*b^6*c^4*d^8 - 2*C^4*a^4*b^2*c^2*d^1
0 + 2*C^4*a^4*b^2*c^4*d^8 + 4*C^4*a^5*b*c*d^11 - 4*C^4*a^5*b*c^3*d^9))/(b*f
^4))*(-(((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*
c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^
2*c*f^2 - 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2
*b^2*f^4)))^(1/2)*1i - (((((32*(4*C*a*b^8*d^11*f^4 - 4*C*b^9*c*d^10*f^4 + 8
*C*a^3*b^6*d^11*f^4 + 4*C*a^5*b^4*d^11*f^4 - 4*C*b^9*c^3*d^8*f^4 + 4*C*a*b^
8*c^2*d^9*f^4 - 8*C*a^2*b^7*c*d^10*f^4 - 4*C*a^4*b^5*c*d^10*f^4 - 8*C*a^2*b
^7*c^3*d^8*f^4 + 8*C*a^3*b^6*c^2*d^9*f^4 - 4*C*a^4*b^5*c^3*d^8*f^4 + 4*C*a^
5*b^4*c^2*d^9*f^4))/(b*f^5) + (32*(c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^2
*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*
a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2
*c*f^2 + 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(
16*b^10*d^10*f^4 + 16*a^2*b^8*d^10*f^4 - 16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*d
^10*f^4 + 24*b^10*c^2*d^8*f^4 + 40*a^2*b^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8*
f^4 - 8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^9*c*d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 24
*a^5*b^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9*f^4))/(b*f^4))*(-(((8*C^2*a^2*c*f^2 -
8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 +
 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 +
8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c +
 d*tan(e + f*x))^(1/2)*(14*C^2*a*b^7*d^11*f^2 - 2*C^2*a^5*b^3*d^11*f^2 - 10
*C^2*b^8*c^3*d^8*f^2 - 4*C^2*a^3*b^5*d^11*f^2 - 16*C^2*a^7*b*d^11*f^2 + 8*C
^2*a^8*c*d^10*f^2 - 6*C^2*b^8*c*d^10*f^2 + 18*C^2*a*b^7*c^2*d^9*f^2 + 12*C^
2*a^2*b^6*c*d^10*f^2 + 2*C^2*a^4*b^4*c*d^10*f^2 + 24*C^2*a^6*b^2*c*d^10*f^2
 - 16*C^2*a^7*b*c^2*d^9*f^2 + 4*C^2*a^2*b^6*c^3*d^8*f^2 + 4*C^2*a^3*b^5*c^2

*d^9*f^2 - 10*C^2*a^4*b^4*c^3*d^8*f^2 + 2*C^2*a^5*b^3*c^2*d^9*f^2 + 8*C^2*a
^6*b^2*c^3*d^8*f^2))/(b*f^4))*(-(((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C
^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b
^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(16*(
a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(15*C^3*a^4*b^3*d^12*f^2 -
C^3*a^2*b^5*d^12*f^2 + C^3*b^7*c^2*d^10*f^2 + C^3*b^7*c^4*d^8*f^2 - 12*C^3
*a^6*b*d^12*f^2 - 24*C^3*a^3*b^4*c*d^11*f^2 + 24*C^3*a^5*b^2*c*d^11*f^2 - 1
2*C^3*a^6*b*c^2*d^10*f^2 + 8*C^3*a^2*b^5*c^2*d^10*f^2 + 9*C^3*a^2*b^5*c^4*d
^8*f^2 - 24*C^3*a^3*b^4*c^3*d^9*f^2 + 3*C^3*a^4*b^3*c^2*d^10*f^2 - 12*C^3*a
^4*b^3*c^4*d^8*f^2 + 24*C^3*a^5*b^2*c^3*d^9*f^2))/(b*f^5))*(-(((8*C^2*a^2*c
*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^
4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c
*f^2 + 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (
32*(c + d*tan(e + f*x))^(1/2)*(C^4*b^6*d^12 - 2*C^4*a^6*d^12 + 2*C^4*a^6*c^
2*d^10 + 2*C^4*b^6*c^2*d^10 + C^4*b^6*c^4*d^8 - 2*C^4*a^4*b^2*c^2*d^10 + 2*
C^4*a^4*b^2*c^4*d^8 + 4*C^4*a^5*b*c*d^11 - 4*C^4*a^5*b*c^3*d^9))/(b*f^4))*(
-(((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 +
C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^2*c*f^
2 - 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f
^4)))^(1/2)*1i)/((((((32*(4*C*a*b^8*d^11*f^4 - 4*C*b^9*c*d^10*f^4 + 8*C*a^3
*b^6*d^11*f^4 + 4*C*a^5*b^4*d^11*f^4 - 4*C*b^9*c^3*d^8*f^4 + 4*C*a*b^8*c^2*
d^9*f^4 - 8*C*a^2*b^7*c*d^10*f^4 - 4*C*a^4*b^5*c*d^10*f^4 - 8*C*a^2*b^7*c^3
*d^8*f^4 + 8*C*a^3*b^6*c^2*d^9*f^4 - 4*C*a^4*b^5*c^3*d^8*f^4 + 4*C*a^5*b^4*
c^2*d^9*f^4))/(b*f^5) - (32*(c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^2*c*f^2
- 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^
4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2
+ 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^1
0*d^10*f^4 + 16*a^2*b^8*d^10*f^4 - 16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*d^10*f^
4 + 24*b^10*c^2*d^8*f^4 + 40*a^2*b^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8*f^4 -
8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^9*c*d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 24*a^5*b
^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9*f^4))/(b*f^4))*(-(((8*C^2*a^2*c*f^2 - 8*C^2*
b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^
4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 + 8*C^2*
a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan
(e + f*x))^(1/2)*(14*C^2*a*b^7*d^11*f^2 - 2*C^2*a^5*b^3*d^11*f^2 - 10*C^2*b
^8*c^3*d^8*f^2 - 4*C^2*a^3*b^5*d^11*f^2 - 16*C^2*a^7*b*d^11*f^2 + 8*C^2*a^8
*c*d^10*f^2 - 6*C^2*b^8*c*d^10*f^2 + 18*C^2*a*b^7*c^2*d^9*f^2 + 12*C^2*a^2*
b^6*c*d^10*f^2 + 2*C^2*a^4*b^4*c*d^10*f^2 + 24*C^2*a^6*b^2*c*d^10*f^2 - 16*
C^2*a^7*b*c^2*d^9*f^2 + 4*C^2*a^2*b^6*c^3*d^8*f^2 + 4*C^2*a^3*b^5*c^2*d^9*f
^2 - 10*C^2*a^4*b^4*c^3*d^8*f^2 + 2*C^2*a^5*b^3*c^2*d^9*f^2 + 8*C^2*a^6*b^2
*c^3*d^8*f^2))/(b*f^4))*(-(((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b
*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4
))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(16*(a^4*f^
4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(15*C^3*a^4*b^3*d^12*f^2 - C^3*a
^2*b^5*d^12*f^2 + C^3*b^7*c^2*d^10*f^2 + C^3*b^7*c^4*d^8*f^2 - 12*C^3*a^6*b

*d^12*f^2 - 24*C^3*a^3*b^4*c*d^11*f^2 + 24*C^3*a^5*b^2*c*d^11*f^2 - 12*C^3*a^6*b*c^2*d^10*f^2 + 8*C^3*a^2*b^5*c^2*d^10*f^2 + 9*C^3*a^2*b^5*c^4*d^8*f^2 - 24*C^3*a^3*b^4*c^3*d^9*f^2 + 3*C^3*a^4*b^3*c^2*d^10*f^2 - 12*C^3*a^4*b^3*c^4*d^8*f^2 + 24*C^3*a^5*b^2*c^3*d^9*f^2))/(b*f^5))*(-(((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(C^4*b^6*d^12 - 2*C^4*a^6*d^12 + 2*C^4*a^6*c^2*d^10 + 2*C^4*b^6*c^2*d^10 + C^4*b^6*c^4*d^8 - 2*C^4*a^4*b^2*c^2*d^10 + 2*C^4*a^4*b^2*c^4*d^8 + 4*C^4*a^5*b*c*d^11 - 4*C^4*a^5*b*c^3*d^9))/(b*f^4))*(-(((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (((((32*(4*C*a*b^8*d^11*f^4 - 4*C*b^9*c*d^10*f^4 + 8*C*a^3*b^6*d^11*f^4 + 4*C*a^5*b^4*d^11*f^4 - 4*C*b^9*c^3*d^8*f^4 + 4*C*a*b^8*c^2*d^9*f^4 - 8*C*a^2*b^7*c*d^10*f^4 - 4*C*a^4*b^5*c*d^10*f^4 - 8*C*a^2*b^7*c^3*d^8*f^4 + 8*C*a^3*b^6*c^2*d^9*f^4 - 4*C*a^4*b^5*c^3*d^8*f^4 + 4*C*a^5*b^4*c^2*d^9*f^4))/(b*f^5) + (32*(c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^10*d^10*f^4 + 16*a^2*b^8*d^10*f^4 - 16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*d^10*f^4 + 24*b^10*c^2*d^8*f^4 + 40*a^2*b^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^9*c*d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 24*a^5*b^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9*f^4))/(b*f^4))*(-(((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(14*C^2*a*b^7*d^11*f^2 - 2*C^2*a^5*b^3*d^11*f^2 - 10*C^2*b^8*c^3*d^8*f^2 - 4*C^2*a^3*b^5*d^11*f^2 - 16*C^2*a^7*b*d^11*f^2 + 8*C^2*a^8*c*d^10*f^2 - 6*C^2*b^8*c*d^10*f^2 + 18*C^2*a*b^7*c^2*d^9*f^2 + 12*C^2*a^2*b^6*c*d^10*f^2 + 2*C^2*a^4*b^4*c*d^10*f^2 + 24*C^2*a^6*b^2*c*d^10*f^2 - 16*C^2*a^7*b*c^2*d^9*f^2 + 4*C^2*a^2*b^6*c^3*d^8*f^2 + 4*C^2*a^3*b^5*c^2*d^9*f^2 - 10*C^2*a^4*b^4*c^3*d^8*f^2 + 2*C^2*a^5*b^3*c^2*d^9*f^2 + 8*C^2*a^6*b^2*c^3*d^8*f^2))/(b*f^4))*(-(((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(15*C^3*a^4*b^3*d^12*f^2 - C^3*a^2*b^5*d^12*f^2 + C^3*b^7*c^2*d^10*f^2 + C^3*b^7*c^4*d^8*f^2 - 12*C^3*a^6*b*d^12*f^2 - 24*C^3*a^3*b^4*c*d^11*f^2 + 24*C^3*a^5*b^2*c*d^11*f^2 - 12*C^3*a^6*b*c^2*d^10*f^2 + 8*C^3*a^2*b^5*c^2*d^10*f^2 + 9*C^3*a^2*b^5*c^4*d^8*f^2 - 24*C^3*a^3*b^4*c^3*d^9*f^2 + 3*C^3*a^4*b^3*c^2*d^10*f^2 - 12*C^3*a^4*b^3*c^4*d^8*f^2 + 24*C^3*a^5*b^2*c^3*d^9*f^2))/(b*f^5))*(-(((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 + 8*C^2*a* b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(C^4*b^6*d^12 - 2*C^4*a^6*d^12 + 2*C^4*a^6*c^2*d^10 + 2*C^4*b^6*c^2*d^10 + C^4*b^6*c^4*d^8 - 2*C^4*a^4*b^2*c^2*d^10 + 2*C^4*a^4*b^2*c^4*d^8 + 4*C^4*a^5*b*c*d^11 - 4*C^4*a^5*b*c^3*d^9))/(b*f^4))*(-(((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (64*(C^5*a^5*d^13 - C^5*a^3*b^2*d^13 + C^5*a^5*c^2*d^11 + 2*C^5*a^2*b^3*c^3*d^10 + C^5*a^2*b^3*c^5*d^8 - 2*C^5*a^3*b^2*c^2*d^11 - C^5*a^3*b^2*c^4*d^9 - C^5*a^4*b*c*d^12 + C^5*a^2*b^3*c*d^12 - C^5*a^4*b*c^3*d^10))/(b*f^5)))*(-(((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*2i + atan((((((32*(12*A*a*b^7*d^11*f^4 - 12*A*b^8*c*d^10*f^4 + 24*A*a^3*b^5*d^11*f^4 + 12*A*a^5*b^3*d^11*f^4 - 12*A*b^8*c^3*d^8*f^4 + 12*A*a*b^7*c^2*d^9*f^4 - 24*A*a^2*b^6*c*d^10*f^4 - 12*A*a^4*b^4*c*d^10*f^4 - 24*A*a^2*b^6*c^3*d^8*f^4 + 24*A*a^3*b^5*c^2*d^9*f^4 - 12*A*a^4*b^4*c^3*d^8*f^4 + 12*A*a^5*b^3*c^2*d^9*f^4))/f^5 - (32*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/f^4)*((((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^4*d^11*f^2 + 2*A^2*a^5*b^2*d^11*f^2 + 18*A^2*b^7*c^3*d^8*f^2 - 14*A^2*a*b^6*d^11*f^2 + 6*A^2*b^7*c*d^10*f^2 - 18*A^2*a*b^6*c^2*d^9*f^2 - 36*A^2*a^2*b^5*c*d^10*f^2 - 10*A^2*a^4*b^3*c*d^10*f^2 - 12*A^2*a^2*b^5*c^3*d^8*f^2 + 12*A^2*a^3*b^4*c^2*d^9*f^2 + 2*A^2*a^4*b^3*c^3*d^8*f^2 - 2*A^2*a^5*b^2*c^2*d^9*f^2))/f^4)*((((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(13*A^3*a^2*b^4*d^12*f^2 + A^3*a^4*b^2*d^12*f^2 + 3*A^3*b^6*c^2*d^10*f^2 + 3*A^3*b^6*c^4*d^8*f^2 - 16*A^3*a*b^5*c*d^11*f^2 - 16*A^3*a*b^5*c^3*d^9*f^2 + 12*A^3*a^2*b^4*c^2*d^10*f^2 - A^3*a^2*b^4*c^4*d^8*f^2 + A^3*a^4*b^2*c^2*d^10*f^2))/f^5)*((((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*d^12 - 2*A^4*a^2*b^3*d^12 + 3*A^4*b^5*c^4*d^8 + 2*A^4*a^2*b^3*c^2*d^10 + 4*A^4*a*b^4*c*d^11 - 4*A^4*a*b^4*c^3*d^9))/f^4)*((((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^

2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*1i - (((((32*(12*A*a*b^7*d^11*f^4 - 12*A*b^8*c*d^10*f^4 + 24*A*a^3*b^5*d^11*f^4 + 12*A*a^5*b^3*d^11*f^4 - 12*A*b^8*c^3*d^8*f^4 + 12*A*a*b^7*c^2*d^9*f^4 - 24*A*a^2*b^6*c*d^10*f^4 - 12*A*a^4*b^4*c*d^10*f^4 - 24*A*a^2*b^6*c^3*d^8*f^4 + 24*A*a^3*b^5*c^2*d^9*f^4 - 12*A*a^4*b^4*c^3*d^8*f^4 + 12*A*a^5*b^3*c^2*d^9*f^4))/f^5 + (32*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/f^4)*((((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^4*d^11*f^2 + 2*A^2*a^5*b^2*d^11*f^2 + 18*A^2*b^7*c^3*d^8*f^2 - 14*A^2*a*b^6*d^11*f^2 + 6*A^2*b^7*c*d^10*f^2 - 18*A^2*a*b^6*c^2*d^9*f^2 - 36*A^2*a^2*b^5*c*d^10*f^2 - 10*A^2*a^4*b^3*c*d^10*f^2 - 12*A^2*a^2*b^5*c^3*d^8*f^2 + 12*A^2*a^3*b^4*c^2*d^9*f^2 + 2*A^2*a^4*b^3*c^3*d^8*f^2 - 2*A^2*a^5*b^2*c^2*d^9*f^2))/f^4)*((((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(13*A^3*a^2*b^4*d^12*f^2 + A^3*a^4*b^2*d^12*f^2 + 3*A^3*b^6*c^2*d^10*f^2 + 3*A^3*b^6*c^4*d^8*f^2 - 16*A^3*a*b^5*c*d^11*f^2 - 16*A^3*a*b^5*c^3*d^9*f^2 + 12*A^3*a^2*b^4*c^2*d^10*f^2 - A^3*a^2*b^4*c^4*d^8*f^2 + A^3*a^4*b^2*c^2*d^10*f^2))/f^5)*((((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*d^12 - 2*A^4*a^2*b^3*d^12 + 3*A^4*b^5*c^4*d^8 + 2*A^4*a^2*b^3*c^2*d^10 + 4*A^4*a*b^4*c*d^11 - 4*A^4*a*b^4*c^3*d^9))/f^4)*((((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*1i)/((((((32*(12*A*a*b^7*d^11*f^4 - 12*A*b^8*c*d^10*f^4 + 24*A*a^3*b^5*d^11*f^4 + 12*A*a^5*b^3*d^11*f^4 - 12*A*b^8*c^3*d^8*f^4 + 12*A*a*b^7*c^2*d^9*f^4 - 24*A*a^2*b^6*c*d^10*f^4 - 12*A*a^4*b^4*c*d^10*f^4 - 24*A*a^2*b^6*c^3*d^8*f^4 + 24*A*a^3*b^5*c^2*d^9*f^4 - 12*A*a^4*b^4*c^3*d^8*f^4 + 12*A*a^5*b^3*c^2*d^9*f^4))/f^5 - (32*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^9*d^10*f^4 +

16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/f^4)*((((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^4*d^11*f^2 + 2*A^2*a^5*b^2*d^11*f^2 + 18*A^2*b^7*c^3*d^8*f^2 - 14*A^2*a*b^6*d^11*f^2 + 6*A^2*b^7*c*d^10*f^2 - 18*A^2*a*b^6*c^2*d^9*f^2 - 36*A^2*a^2*b^5*c*d^10*f^2 - 10*A^2*a^4*b^3*c*d^10*f^2 - 12*A^2*a^2*b^5*c^3*d^8*f^2 + 12*A^2*a^3*b^4*c^2*d^9*f^2 + 2*A^2*a^4*b^3*c^3*d^8*f^2 - 2*A^2*a^5*b^2*c^2*d^9*f^2))/f^4)*((((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(13*A^3*a^2*b^4*d^12*f^2 + A^3*a^4*b^2*d^12*f^2 + 3*A^3*b^6*c^2*d^10*f^2 + 3*A^3*b^6*c^4*d^8*f^2 - 16*A^3*a*b^5*c*d^11*f^2 - 16*A^3*a*b^5*c^3*d^9*f^2 + 12*A^3*a^2*b^4*c^2*d^10*f^2 - A^3*a^2*b^4*c^4*d^8*f^2 + A^3*a^4*b^2*c^2*d^10*f^2))/f^5)*((((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*d^12 - 2*A^4*a^2*b^3*d^12 + 3*A^4*b^5*c^4*d^8 + 2*A^4*a^2*b^3*c^2*d^10 + 4*A^4*a*b^4*c*d^11 - 4*A^4*a*b^4*c^3*d^9))/f^4)*((((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (64*(A^5*b^4*c^3*d^10 - A^5*a*b^3*d^13 + A^5*b^4*c*d^12 - A^5*a*b^3*c^2*d^11))/f^5 + (((((32*(12*A*a*b^7*d^11*f^4 - 12*A*b^8*c*d^10*f^4 + 24*A*a^3*b^5*d^11*f^4 + 12*A*a^5*b^3*d^11*f^4 - 12*A*b^8*c^3*d^8*f^4 + 12*A*a*b^7*c^2*d^9*f^4 - 24*A*a^2*b^6*c*d^10*f^4 - 12*A*a^4*b^4*c*d^10*f^4 - 24*A*a^2*b^6*c^3*d^8*f^4 + 24*A*a^3*b^5*c^2*d^9*f^4 - 12*A*a^4*b^4*c^3*d^8*f^4 + 12*A*a^5*b^3*c^2*d^9*f^4))/f^5 + (32*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/f^4)*((((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^4*d^11*f^2 + 2*A^2*a^5*b^2*d^11*f^2 + 18*A^2*b^7*c^3*d^8*f^2 - 14*A^2*a*b^6*d^11*f^2 + 6*A^2*b^7*c*d^10*f^2 - 18*A^2*a*b^6*c^2*d^9*f^2 - 36*A^2*a^2*b^5*c*d^10*f^2 - 10*A^2*a^4*b^3* c*d^10*f^2 - 12*A^2*a^2*b^5*c^3*d^8*f^2 + 12*A^2*a^3*b^4*c^2*d^9*f^2 + 2*A^2*a^4*b^3*c^3*d^8*f^2 - 2*A^2*a^5*b^2*c^2*d^9*f^2))/f^4)*((((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(13*A^3*a^2*b^4*d^12*f^2 + A^3*a^4*b^2*d^12*f^2 + 3*A^3*b^6*c^2*d^10*f^2 + 3*A^3*b^6*c^4*d^8*f^2 - 16*A^3*a*b^5*c*d^11*f^2 - 16*A^3*a*b^5*c^3*d^9*f^2 + 12*A^3*a^2*b^4*c^2*d^10*f^2 - A^3*a^2*b^4*c^4*d^8*f^2 + A^3*a^4*b^2*c^2*d^10*f^2))/f^5)*((((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*d^12 - 2*A^4*a^2*b^3*d^12 + 3*A^4*b^5*c^4*d^8 + 2*A^4*a^2*b^3*c^2*d^10 + 4*A^4*a*b^4*c*d^11 - 4*A^4*a*b^4*c^3*d^9))/f^4)*((((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)))*((((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*2i + atan(((((((32*(12*A*a*b^7*d^11*f^4 - 12*A*b^8*c*d^10*f^4 + 24*A*a^3*b^5*d^11*f^4 + 12*A*a^5*b^3*d^11*f^4 - 12*A*b^8*c^3*d^8*f^4 + 12*A*a*b^7*c^2*d^9*f^4 - 24*A*a^2*b^6*c*d^10*f^4 - 12*A*a^4*b^4*c*d^10*f^4 - 24*A*a^2*b^6*c^3*d^8*f^4 + 24*A*a^3*b^5*c^2*d^9*f^4 - 12*A*a^4*b^4*c^3*d^8*f^4 + 12*A*a^5*b^3*c^2*d^9*f^4))/f^5 - (32*(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/f^4)*(-(((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^4*d^11*f^2 + 2*A^2*a^5*b^2*d^11*f^2 + 18*A^2*b^7*c^3*d^8*f^2 - 14*A^2*a*b^6*d^11*f^2 + 6*A^2*b^7*c*d^10*f^2 - 18*A^2*a*b^6*c^2*d^9*f^2 - 36*A^2*a^2*b^5*c*d^10*f^2 - 10*A^2*a^4*b^3*c*d^10*f^2 - 12*A^2*a^2*b^5*c^3*d^8*f^2 + 12*A^2*a^3*b^4*c^2*d^9*f^2 + 2*A^2*a^4*b^3*c^3*d^8*f^2 - 2*A^2*a^5*b^2*c^2*d^9*f^2))/f^4)*(-(((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(13*A^3*a^2*b^4*d^12*f^2 + A^3*a^4*b^2*d^12*f^2 + 3*A^3*b^6*c^2*d^10*f^2 + 3*A^3*b^6*c^4*d^8*f^2 - 16*A^3*a*b^5*c*d^11*f^2 - 16*A^3*a*b^5*c^3*d^9*f^2 + 12*A^3*a^2*b^4*c^2*d^10*

f^2 - A^3*a^2*b^4*c^4*d^8*f^2 + A^3*a^4*b^2*c^2*d^10*f^2))/f^5)*(-(((8*A^2*
a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(
16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*
b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2
) - (32*(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*d^12 - 2*A^4*a^2*b^3*d^12 + 3*A
^4*b^5*c^4*d^8 + 2*A^4*a^2*b^3*c^2*d^10 + 4*A^4*a*b^4*c*d^11 - 4*A^4*a*b^4*
c^3*d^9))/f^4)*(-(((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2
/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2)
+ 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f
^4 + 2*a^2*b^2*f^4)))^(1/2)*1i - (((((32*(12*A*a*b^7*d^11*f^4 - 12*A*b^8*c*
d^10*f^4 + 24*A*a^3*b^5*d^11*f^4 + 12*A*a^5*b^3*d^11*f^4 - 12*A*b^8*c^3*d^8
*f^4 + 12*A*a*b^7*c^2*d^9*f^4 - 24*A*a^2*b^6*c*d^10*f^4 - 12*A*a^4*b^4*c*d^
10*f^4 - 24*A*a^2*b^6*c^3*d^8*f^4 + 24*A*a^3*b^5*c^2*d^9*f^4 - 12*A*a^4*b^4
*c^3*d^8*f^4 + 12*A*a^5*b^3*c^2*d^9*f^4))/f^5 + (32*(c + d*tan(e + f*x))^(1
/2)*(-(((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c
^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*A^2*a^2
*c*f^2 - 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*
b^2*f^4)))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f
^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*
a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^
6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/f^4)*(-(((8*A^2*
a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(
16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*
b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2
) - (32*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^4*d^11*f^2 + 2*A^2*a^5*b^2
*d^11*f^2 + 18*A^2*b^7*c^3*d^8*f^2 - 14*A^2*a*b^6*d^11*f^2 + 6*A^2*b^7*c*d^
10*f^2 - 18*A^2*a*b^6*c^2*d^9*f^2 - 36*A^2*a^2*b^5*c*d^10*f^2 - 10*A^2*a^4*
b^3*c*d^10*f^2 - 12*A^2*a^2*b^5*c^3*d^8*f^2 + 12*A^2*a^3*b^4*c^2*d^9*f^2 +
2*A^2*a^4*b^3*c^3*d^8*f^2 - 2*A^2*a^5*b^2*c^2*d^9*f^2))/f^4)*(-(((8*A^2*a^2
*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*
a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2
*c*f^2 + 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) +
 (32*(13*A^3*a^2*b^4*d^12*f^2 + A^3*a^4*b^2*d^12*f^2 + 3*A^3*b^6*c^2*d^10*f
^2 + 3*A^3*b^6*c^4*d^8*f^2 - 16*A^3*a*b^5*c*d^11*f^2 - 16*A^3*a*b^5*c^3*d^9
*f^2 + 12*A^3*a^2*b^4*c^2*d^10*f^2 - A^3*a^2*b^4*c^4*d^8*f^2 + A^3*a^4*b^2*
c^2*d^10*f^2))/f^5)*(-(((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f
^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(
1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 +
b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*
d^12 - 2*A^4*a^2*b^3*d^12 + 3*A^4*b^5*c^4*d^8 + 2*A^4*a^2*b^3*c^2*d^10 + 4*
A^4*a*b^4*c*d^11 - 4*A^4*a*b^4*c^3*d^9))/f^4)*(-(((8*A^2*a^2*c*f^2 - 8*A^2*
b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^
4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 + 8*A^2*
a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*1i)/((((((32*(12
*A*a*b^7*d^11*f^4 - 12*A*b^8*c*d^10*f^4 + 24*A*a^3*b^5*d^11*f^4 + 12*A*a^5*

b^3*d^11*f^4 - 12*A*b^8*c^3*d^8*f^4 + 12*A*a*b^7*c^2*d^9*f^4 - 24*A*a^2*b^6 *c*d^10*f^4 - 12*A*a^4*b^4*c*d^10*f^4 - 24*A*a^2*b^6*c^3*d^8*f^4 + 24*A*a^3 *b^5*c^2*d^9*f^4 - 12*A*a^4*b^4*c^3*d^8*f^4 + 12*A*a^5*b^3*c^2*d^9*f^4))/f^ 5 - (32*(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32* a^2*b^2*f^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/ (16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b ^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^ 4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b ^2*c*d^9*f^4))/f^4)*(-(((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f ^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a ^3*b^4*d^11*f^2 + 2*A^2*a^5*b^2*d^11*f^2 + 18*A^2*b^7*c^3*d^8*f^2 - 14*A^2* a*b^6*d^11*f^2 + 6*A^2*b^7*c*d^10*f^2 - 18*A^2*a*b^6*c^2*d^9*f^2 - 36*A^2*a ^2*b^5*c*d^10*f^2 - 10*A^2*a^4*b^3*c*d^10*f^2 - 12*A^2*a^2*b^5*c^3*d^8*f^2 + 12*A^2*a^3*b^4*c^2*d^9*f^2 + 2*A^2*a^4*b^3*c^3*d^8*f^2 - 2*A^2*a^5*b^2*c^ 2*d^9*f^2))/f^4)*(-(((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2) ^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4 *f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(13*A^3*a^2*b^4*d^12*f^2 + A^3*a^4*b^2* d^12*f^2 + 3*A^3*b^6*c^2*d^10*f^2 + 3*A^3*b^6*c^4*d^8*f^2 - 16*A^3*a*b^5*c* d^11*f^2 - 16*A^3*a*b^5*c^3*d^9*f^2 + 12*A^3*a^2*b^4*c^2*d^10*f^2 - A^3*a^2 *b^4*c^4*d^8*f^2 + A^3*a^4*b^2*c^2*d^10*f^2))/f^5)*(-(((8*A^2*a^2*c*f^2 - 8 *A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 + 8 *A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*d^12 - 2*A^4*a^2*b^3*d^12 + 3*A^4*b^5*c^4*d^ 8 + 2*A^4*a^2*b^3*c^2*d^10 + 4*A^4*a*b^4*c*d^11 - 4*A^4*a*b^4*c^3*d^9))/f^4)*(-(((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*A^2*a^2*c *f^2 - 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^ 2*f^4)))^(1/2) - (64*(A^5*b^4*c^3*d^10 - A^5*a*b^3*d^13 + A^5*b^4*c*d^12 - A^5*a*b^3*c^2*d^11))/f^5 + (((((32*(12*A*a*b^7*d^11*f^4 - 12*A*b^8*c*d^10*f ^4 + 24*A*a^3*b^5*d^11*f^4 + 12*A*a^5*b^3*d^11*f^4 - 12*A*b^8*c^3*d^8*f^4 + 12*A*a*b^7*c^2*d^9*f^4 - 24*A*a^2*b^6*c*d^10*f^4 - 12*A*a^4*b^4*c*d^10*f^4 - 24*A*a^2*b^6*c^3*d^8*f^4 + 24*A*a^3*b^5*c^2*d^9*f^4 - 12*A*a^4*b^4*c^3*d ^8*f^4 + 12*A*a^5*b^3*c^2*d^9*f^4))/f^5 + (32*(c + d*tan(e + f*x))^(1/2)*(- (((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A ^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^ 4)))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 1 6*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^ 5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^

9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/f^4)*(-(((8*A^2*a^2*c*
f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4
*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*
f^2 + 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (3
2*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^4*d^11*f^2 + 2*A^2*a^5*b^2*d^11*
f^2 + 18*A^2*b^7*c^3*d^8*f^2 - 14*A^2*a*b^6*d^11*f^2 + 6*A^2*b^7*c*d^10*f^2
 - 18*A^2*a*b^6*c^2*d^9*f^2 - 36*A^2*a^2*b^5*c*d^10*f^2 - 10*A^2*a^4*b^3*c*
d^10*f^2 - 12*A^2*a^2*b^5*c^3*d^8*f^2 + 12*A^2*a^3*b^4*c^2*d^9*f^2 + 2*A^2*
a^4*b^3*c^3*d^8*f^2 - 2*A^2*a^5*b^2*c^2*d^9*f^2))/f^4)*(-(((8*A^2*a^2*c*f^2
 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^
4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2
 + 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(
13*A^3*a^2*b^4*d^12*f^2 + A^3*a^4*b^2*d^12*f^2 + 3*A^3*b^6*c^2*d^10*f^2 + 3
*A^3*b^6*c^4*d^8*f^2 - 16*A^3*a*b^5*c*d^11*f^2 - 16*A^3*a*b^5*c^3*d^9*f^2 +
 12*A^3*a^2*b^4*c^2*d^10*f^2 - A^3*a^2*b^4*c^4*d^8*f^2 + A^3*a^4*b^2*c^2*d^
10*f^2))/f^5)*(-(((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/
4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) +
 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^
4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*d^12 -
 2*A^4*a^2*b^3*d^12 + 3*A^4*b^5*c^4*d^8 + 2*A^4*a^2*b^3*c^2*d^10 + 4*A^4*a*
b^4*c*d^11 - 4*A^4*a*b^4*c^3*d^9))/f^4)*(-(((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*
f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4
+ 32*a^2*b^2*f^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*
f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)))*(-(((8*A^2*a^2*c*f^2
 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^
4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2
 + 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*2i + (a
tan((((C^2*a^5*d - C^2*a^4*b*c)*(((C^2*a^5*d - C^2*a^4*b*c)*((32*(15*C^3*a^
4*b^3*d^12*f^2 - C^3*a^2*b^5*d^12*f^2 + C^3*b^7*c^2*d^10*f^2 + C^3*b^7*c^4*
d^8*f^2 - 12*C^3*a^6*b*d^12*f^2 - 24*C^3*a^3*b^4*c*d^11*f^2 + 24*C^3*a^5*b^
2*c*d^11*f^2 - 12*C^3*a^6*b*c^2*d^10*f^2 + 8*C^3*a^2*b^5*c^2*d^10*f^2 + 9*C
^3*a^2*b^5*c^4*d^8*f^2 - 24*C^3*a^3*b^4*c^3*d^9*f^2 + 3*C^3*a^4*b^3*c^2*d^1
0*f^2 - 12*C^3*a^4*b^3*c^4*d^8*f^2 + 24*C^3*a^5*b^2*c^3*d^9*f^2))/(b*f^5) +
 ((C^2*a^5*d - C^2*a^4*b*c)*(((C^2*a^5*d - C^2*a^4*b*c)*((32*(4*C*a*b^8*d^1
1*f^4 - 4*C*b^9*c*d^10*f^4 + 8*C*a^3*b^6*d^11*f^4 + 4*C*a^5*b^4*d^11*f^4 -
4*C*b^9*c^3*d^8*f^4 + 4*C*a*b^8*c^2*d^9*f^4 - 8*C*a^2*b^7*c*d^10*f^4 - 4*C*
a^4*b^5*c*d^10*f^4 - 8*C*a^2*b^7*c^3*d^8*f^4 + 8*C*a^3*b^6*c^2*d^9*f^4 - 4*
C*a^4*b^5*c^3*d^8*f^4 + 4*C*a^5*b^4*c^2*d^9*f^4))/(b*f^5) - (32*(C^2*a^5*d
- C^2*a^4*b*c)*(c + d*tan(e + f*x))^(1/2)*(16*b^10*d^10*f^4 + 16*a^2*b^8*d^
10*f^4 - 16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*d^10*f^4 + 24*b^10*c^2*d^8*f^4 +
40*a^2*b^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^2*d^8*f^4 + 8*
a*b^9*c*d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 24*a^5*b^5*c*d^9*f^4 + 8*a^7*b^3*c
*d^9*f^4))/(b*f^4*(-(C^2*a^5*d - C^2*a^4*b*c)*(b^7*f^2 + 2*a^2*b^5*f^2 + a^
4*b^3*f^2))^(1/2))))/(-(C^2*a^5*d - C^2*a^4*b*c)*(b^7*f^2 + 2*a^2*b^5*f^2 +
 a^4*b^3*f^2))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(14*C^2*a*b^7*d^11*f^

2 - 2*C^2*a^5*b^3*d^11*f^2 - 10*C^2*b^8*c^3*d^8*f^2 - 4*C^2*a^3*b^5*d^11*f^
2 - 16*C^2*a^7*b*d^11*f^2 + 8*C^2*a^8*c*d^10*f^2 - 6*C^2*b^8*c*d^10*f^2 + 1
8*C^2*a*b^7*c^2*d^9*f^2 + 12*C^2*a^2*b^6*c*d^10*f^2 + 2*C^2*a^4*b^4*c*d^10*
f^2 + 24*C^2*a^6*b^2*c*d^10*f^2 - 16*C^2*a^7*b*c^2*d^9*f^2 + 4*C^2*a^2*b^6*
c^3*d^8*f^2 + 4*C^2*a^3*b^5*c^2*d^9*f^2 - 10*C^2*a^4*b^4*c^3*d^8*f^2 + 2*C^
2*a^5*b^3*c^2*d^9*f^2 + 8*C^2*a^6*b^2*c^3*d^8*f^2))/(b*f^4)))/(-(C^2*a^5*d
- C^2*a^4*b*c)*(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2))^(1/2)))/(-(C^2*a^5*
d - C^2*a^4*b*c)*(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2))^(1/2) - (32*(c +
d*tan(e + f*x))^(1/2)*(C^4*b^6*d^12 - 2*C^4*a^6*d^12 + 2*C^4*a^6*c^2*d^10 +
2*C^4*b^6*c^2*d^10 + C^4*b^6*c^4*d^8 - 2*C^4*a^4*b^2*c^2*d^10 + 2*C^4*a^4*
b^2*c^4*d^8 + 4*C^4*a^5*b*c*d^11 - 4*C^4*a^5*b*c^3*d^9))/(b*f^4))*1i)/(-(C^
2*a^5*d - C^2*a^4*b*c)*(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2))^(1/2) - ((C
^2*a^5*d - C^2*a^4*b*c)*(((C^2*a^5*d - C^2*a^4*b*c)*((32*(15*C^3*a^4*b^3*d^
12*f^2 - C^3*a^2*b^5*d^12*f^2 + C^3*b^7*c^2*d^10*f^2 + C^3*b^7*c^4*d^8*f^2
- 12*C^3*a^6*b*d^12*f^2 - 24*C^3*a^3*b^4*c*d^11*f^2 + 24*C^3*a^5*b^2*c*d^11
*f^2 - 12*C^3*a^6*b*c^2*d^10*f^2 + 8*C^3*a^2*b^5*c^2*d^10*f^2 + 9*C^3*a^2*b
^5*c^4*d^8*f^2 - 24*C^3*a^3*b^4*c^3*d^9*f^2 + 3*C^3*a^4*b^3*c^2*d^10*f^2 -
12*C^3*a^4*b^3*c^4*d^8*f^2 + 24*C^3*a^5*b^2*c^3*d^9*f^2))/(b*f^5) + ((C^2*a
^5*d - C^2*a^4*b*c)*(((C^2*a^5*d - C^2*a^4*b*c)*((32*(4*C*a*b^8*d^11*f^4 -
4*C*b^9*c*d^10*f^4 + 8*C*a^3*b^6*d^11*f^4 + 4*C*a^5*b^4*d^11*f^4 - 4*C*b^9*
c^3*d^8*f^4 + 4*C*a*b^8*c^2*d^9*f^4 - 8*C*a^2*b^7*c*d^10*f^4 - 4*C*a^4*b^5*
c*d^10*f^4 - 8*C*a^2*b^7*c^3*d^8*f^4 + 8*C*a^3*b^6*c^2*d^9*f^4 - 4*C*a^4*b^
5*c^3*d^8*f^4 + 4*C*a^5*b^4*c^2*d^9*f^4))/(b*f^5) + (32*(C^2*a^5*d - C^2*a^
4*b*c)*(c + d*tan(e + f*x))^(1/2)*(16*b^10*d^10*f^4 + 16*a^2*b^8*d^10*f^4 -
16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*d^10*f^4 + 24*b^10*c^2*d^8*f^4 + 40*a^2*b
^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^9*c*
d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 24*a^5*b^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9*f^4
))/(b*f^4*(-(C^2*a^5*d - C^2*a^4*b*c)*(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^
2))^(1/2))))/(-(C^2*a^5*d - C^2*a^4*b*c)*(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3
f^2))^(1/2) + (32(c + d*tan(e + f*x))^(1/2)*(14*C^2*a*b^7*d^11*f^2 - 2*C^
2*a^5*b^3*d^11*f^2 - 10*C^2*b^8*c^3*d^8*f^2 - 4*C^2*a^3*b^5*d^11*f^2 - 16*C
^2*a^7*b*d^11*f^2 + 8*C^2*a^8*c*d^10*f^2 - 6*C^2*b^8*c*d^10*f^2 + 18*C^2*a*
b^7*c^2*d^9*f^2 + 12*C^2*a^2*b^6*c*d^10*f^2 + 2*C^2*a^4*b^4*c*d^10*f^2 + 24
*C^2*a^6*b^2*c*d^10*f^2 - 16*C^2*a^7*b*c^2*d^9*f^2 + 4*C^2*a^2*b^6*c^3*d^8*
f^2 + 4*C^2*a^3*b^5*c^2*d^9*f^2 - 10*C^2*a^4*b^4*c^3*d^8*f^2 + 2*C^2*a^5*b^
3*c^2*d^9*f^2 + 8*C^2*a^6*b^2*c^3*d^8*f^2))/(b*f^4)))/(-(C^2*a^5*d - C^2*a^
4*b*c)*(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2))^(1/2)))/(-(C^2*a^5*d - C^2*
a^4*b*c)*(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2))^(1/2) + (32*(c + d*tan(e
+ f*x))^(1/2)*(C^4*b^6*d^12 - 2*C^4*a^6*d^12 + 2*C^4*a^6*c^2*d^10 + 2*C^4*b
^6*c^2*d^10 + C^4*b^6*c^4*d^8 - 2*C^4*a^4*b^2*c^2*d^10 + 2*C^4*a^4*b^2*c^4*
d^8 + 4*C^4*a^5*b*c*d^11 - 4*C^4*a^5*b*c^3*d^9))/(b*f^4))*1i)/(-(C^2*a^5*d
- C^2*a^4*b*c)*(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2))^(1/2))/(((C^2*a^5*d
- C^2*a^4*b*c)*(((C^2*a^5*d - C^2*a^4*b*c)*((32*(15*C^3*a^4*b^3*d^12*f^2 -
C^3*a^2*b^5*d^12*f^2 + C^3*b^7*c^2*d^10*f^2 + C^3*b^7*c^4*d^8*f^2 - 12*C^3
*a^6*b*d^12*f^2 - 24*C^3*a^3*b^4*c*d^11*f^2 + 24*C^3*a^5*b^2*c*d^11*f^2 - 1

2*C^3*a^6*b*c^2*d^10*f^2 + 8*C^3*a^2*b^5*c^2*d^10*f^2 + 9*C^3*a^2*b^5*c^4*d
^8*f^2 - 24*C^3*a^3*b^4*c^3*d^9*f^2 + 3*C^3*a^4*b^3*c^2*d^10*f^2 - 12*C^3*a
^4*b^3*c^4*d^8*f^2 + 24*C^3*a^5*b^2*c^3*d^9*f^2))/(b*f^5) + ((C^2*a^5*d - C
^2*a^4*b*c)*(((C^2*a^5*d - C^2*a^4*b*c)*((32*(4*C*a*b^8*d^11*f^4 - 4*C*b^9*
c*d^10*f^4 + 8*C*a^3*b^6*d^11*f^4 + 4*C*a^5*b^4*d^11*f^4 - 4*C*b^9*c^3*d^8*
f^4 + 4*C*a*b^8*c^2*d^9*f^4 - 8*C*a^2*b^7*c*d^10*f^4 - 4*C*a^4*b^5*c*d^10*f
^4 - 8*C*a^2*b^7*c^3*d^8*f^4 + 8*C*a^3*b^6*c^2*d^9*f^4 - 4*C*a^4*b^5*c^3*d^
8*f^4 + 4*C*a^5*b^4*c^2*d^9*f^4))/(b*f^5) - (32*(C^2*a^5*d - C^2*a^4*b*c)*(
c + d*tan(e + f*x))^(1/2)*(16*b^10*d^10*f^4 + 16*a^2*b^8*d^10*f^4 - 16*a^4*
b^6*d^10*f^4 - 16*a^6*b^4*d^10*f^4 + 24*b^10*c^2*d^8*f^4 + 40*a^2*b^8*c^2*d
^8*f^4 + 8*a^4*b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^9*c*d^9*f^4
+ 24*a^3*b^7*c*d^9*f^4 + 24*a^5*b^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9*f^4))/(b*f^
4*(-(C^2*a^5*d - C^2*a^4*b*c)*(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2))^(1/2
))))/(-(C^2*a^5*d - C^2*a^4*b*c)*(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2))^(
1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(14*C^2*a*b^7*d^11*f^2 - 2*C^2*a^5*b^
3*d^11*f^2 - 10*C^2*b^8*c^3*d^8*f^2 - 4*C^2*a^3*b^5*d^11*f^2 - 16*C^2*a^7*b
*d^11*f^2 + 8*C^2*a^8*c*d^10*f^2 - 6*C^2*b^8*c*d^10*f^2 + 18*C^2*a*b^7*c^2*
d^9*f^2 + 12*C^2*a^2*b^6*c*d^10*f^2 + 2*C^2*a^4*b^4*c*d^10*f^2 + 24*C^2*a^6
*b^2*c*d^10*f^2 - 16*C^2*a^7*b*c^2*d^9*f^2 + 4*C^2*a^2*b^6*c^3*d^8*f^2 + 4*
C^2*a^3*b^5*c^2*d^9*f^2 - 10*C^2*a^4*b^4*c^3*d^8*f^2 + 2*C^2*a^5*b^3*c^2*d^
9*f^2 + 8*C^2*a^6*b^2*c^3*d^8*f^2))/(b*f^4)))/(-(C^2*a^5*d - C^2*a^4*b*c)*(
b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2))^(1/2)))/(-(C^2*a^5*d - C^2*a^4*b*c)
*(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2))^(1/2) - (32*(c + d*tan(e + f*x))^
(1/2)*(C^4*b^6*d^12 - 2*C^4*a^6*d^12 + 2*C^4*a^6*c^2*d^10 + 2*C^4*b^6*c^2*d
^10 + C^4*b^6*c^4*d^8 - 2*C^4*a^4*b^2*c^2*d^10 + 2*C^4*a^4*b^2*c^4*d^8 + 4*
C^4*a^5*b*c*d^11 - 4*C^4*a^5*b*c^3*d^9))/(b*f^4)))/(-(C^2*a^5*d - C^2*a^4*b
c)(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2))^(1/2) - (64*(C^5*a^5*d^13 - C^
5*a^3*b^2*d^13 + C^5*a^5*c^2*d^11 + 2*C^5*a^2*b^3*c^3*d^10 + C^5*a^2*b^3*c^
5*d^8 - 2*C^5*a^3*b^2*c^2*d^11 - C^5*a^3*b^2*c^4*d^9 - C^5*a^4*b*c*d^12 + C
^5*a^2*b^3*c*d^12 - C^5*a^4*b*c^3*d^10))/(b*f^5) + ((C^2*a^5*d - C^2*a^4*b*
c)*(((C^2*a^5*d - C^2*a^4*b*c)*((32*(15*C^3*a^4*b^3*d^12*f^2 - C^3*a^2*b^5*
d^12*f^2 + C^3*b^7*c^2*d^10*f^2 + C^3*b^7*c^4*d^8*f^2 - 12*C^3*a^6*b*d^12*f
^2 - 24*C^3*a^3*b^4*c*d^11*f^2 + 24*C^3*a^5*b^2*c*d^11*f^2 - 12*C^3*a^6*b*c
^2*d^10*f^2 + 8*C^3*a^2*b^5*c^2*d^10*f^2 + 9*C^3*a^2*b^5*c^4*d^8*f^2 - 24*C
^3*a^3*b^4*c^3*d^9*f^2 + 3*C^3*a^4*b^3*c^2*d^10*f^2 - 12*C^3*a^4*b^3*c^4*d^
8*f^2 + 24*C^3*a^5*b^2*c^3*d^9*f^2))/(b*f^5) + ((C^2*a^5*d - C^2*a^4*b*c)*(
((C^2*a^5*d - C^2*a^4*b*c)*((32*(4*C*a*b^8*d^11*f^4 - 4*C*b^9*c*d^10*f^4 +
8*C*a^3*b^6*d^11*f^4 + 4*C*a^5*b^4*d^11*f^4 - 4*C*b^9*c^3*d^8*f^4 + 4*C*a*b
^8*c^2*d^9*f^4 - 8*C*a^2*b^7*c*d^10*f^4 - 4*C*a^4*b^5*c*d^10*f^4 - 8*C*a^2*
b^7*c^3*d^8*f^4 + 8*C*a^3*b^6*c^2*d^9*f^4 - 4*C*a^4*b^5*c^3*d^8*f^4 + 4*C*a
^5*b^4*c^2*d^9*f^4))/(b*f^5) + (32*(C^2*a^5*d - C^2*a^4*b*c)*(c + d*tan(e +
f*x))^(1/2)*(16*b^10*d^10*f^4 + 16*a^2*b^8*d^10*f^4 - 16*a^4*b^6*d^10*f^4
- 16*a^6*b^4*d^10*f^4 + 24*b^10*c^2*d^8*f^4 + 40*a^2*b^8*c^2*d^8*f^4 + 8*a^
4*b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^9*c*d^9*f^4 + 24*a^3*b^7*
c*d^9*f^4 + 24*a^5*b^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9*f^4))/(b*f^4*(-(C^2*a^5*

d - C^2*a^4*b*c)*(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2))^(1/2))))/(-(C^2*a ^5*d - C^2*a^4*b*c)*(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(14*C^2*a*b^7*d^11*f^2 - 2*C^2*a^5*b^3*d^11*f^2 - 10*C^2*b^8*c^3*d^8*f^2 - 4*C^2*a^3*b^5*d^11*f^2 - 16*C^2*a^7*b*d^11*f^2 + 8 *C^2*a^8*c*d^10*f^2 - 6*C^2*b^8*c*d^10*f^2 + 18*C^2*a*b^7*c^2*d^9*f^2 + 12* C^2*a^2*b^6*c*d^10*f^2 + 2*C^2*a^4*b^4*c*d^10*f^2 + 24*C^2*a^6*b^2*c*d^10*f ^2 - 16*C^2*a^7*b*c^2*d^9*f^2 + 4*C^2*a^2*b^6*c^3*d^8*f^2 + 4*C^2*a^3*b^5*c ^2*d^9*f^2 - 10*C^2*a^4*b^4*c^3*d^8*f^2 + 2*C^2*a^5*b^3*c^2*d^9*f^2 + 8*C^2 *a^6*b^2*c^3*d^8*f^2))/(b*f^4)))/(-(C^2*a^5*d - C^2*a^4*b*c)*(b^7*f^2 + 2*a ^2*b^5*f^2 + a^4*b^3*f^2))^(1/2)))/(-(C^2*a^5*d - C^2*a^4*b*c)*(b^7*f^2 + 2 *a^2*b^5*f^2 + a^4*b^3*f^2))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(C^4*b^ 6*d^12 - 2*C^4*a^6*d^12 + 2*C^4*a^6*c^2*d^10 + 2*C^4*b^6*c^2*d^10 + C^4*b^6 *c^4*d^8 - 2*C^4*a^4*b^2*c^2*d^10 + 2*C^4*a^4*b^2*c^4*d^8 + 4*C^4*a^5*b*c*d ^11 - 4*C^4*a^5*b*c^3*d^9))/(b*f^4)))/(-(C^2*a^5*d - C^2*a^4*b*c)*(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2))^(1/2)))*(C^2*a^5*d - C^2*a^4*b*c)*2i)/(-(C^ 2*a^5*d - C^2*a^4*b*c)*(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2))^(1/2) + (at an((((((32*(13*A^3*a^2*b^4*d^12*f^2 + A^3*a^4*b^2*d^12*f^2 + 3*A^3*b^6*c^2 *d^10*f^2 + 3*A^3*b^6*c^4*d^8*f^2 - 16*A^3*a*b^5*c*d^11*f^2 - 16*A^3*a*b^5* c^3*d^9*f^2 + 12*A^3*a^2*b^4*c^2*d^10*f^2 - A^3*a^2*b^4*c^4*d^8*f^2 + A^3*a ^4*b^2*c^2*d^10*f^2))/f^5 + ((((A^2*b^2*c - A^2*a*b*d)*((32*(12*A*a*b^7*d^1 1*f^4 - 12*A*b^8*c*d^10*f^4 + 24*A*a^3*b^5*d^11*f^4 + 12*A*a^5*b^3*d^11*f^4 - 12*A*b^8*c^3*d^8*f^4 + 12*A*a*b^7*c^2*d^9*f^4 - 24*A*a^2*b^6*c*d^10*f^4 - 12*A*a^4*b^4*c*d^10*f^4 - 24*A*a^2*b^6*c^3*d^8*f^4 + 24*A*a^3*b^5*c^2*d^9 *f^4 - 12*A*a^4*b^4*c^3*d^8*f^4 + 12*A*a^5*b^3*c^2*d^9*f^4))/f^5 - (32*(A^2 *b^2*c - A^2*a*b*d)*(c + d*tan(e + f*x))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^ 7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^ 2*c*d^9*f^4))/(f^4*((A^2*b^2*c - A^2*a*b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2* f^2))^(1/2))))/((A^2*b^2*c - A^2*a*b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2*f^2))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^4*d^11*f^2 + 2*A^2*a ^5*b^2*d^11*f^2 + 18*A^2*b^7*c^3*d^8*f^2 - 14*A^2*a*b^6*d^11*f^2 + 6*A^2*b^ 7*c*d^10*f^2 - 18*A^2*a*b^6*c^2*d^9*f^2 - 36*A^2*a^2*b^5*c*d^10*f^2 - 10*A^ 2*a^4*b^3*c*d^10*f^2 - 12*A^2*a^2*b^5*c^3*d^8*f^2 + 12*A^2*a^3*b^4*c^2*d^9* f^2 + 2*A^2*a^4*b^3*c^3*d^8*f^2 - 2*A^2*a^5*b^2*c^2*d^9*f^2))/f^4)*(A^2*b^2 *c - A^2*a*b*d))/((A^2*b^2*c - A^2*a*b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2*f^ 2))^(1/2))*(A^2*b^2*c - A^2*a*b*d))/((A^2*b^2*c - A^2*a*b*d)*(a^4*f^2 + b^4 *f^2 + 2*a^2*b^2*f^2))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*d^12 - 2*A^4*a^2*b^3*d^12 + 3*A^4*b^5*c^4*d^8 + 2*A^4*a^2*b^3*c^2*d^10 + 4*A^4* a*b^4*c*d^11 - 4*A^4*a*b^4*c^3*d^9))/f^4)*(A^2*b^2*c - A^2*a*b*d)*1i)/((A^2 *b^2*c - A^2*a*b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2*f^2))^(1/2) - (((((32*(1 3*A^3*a^2*b^4*d^12*f^2 + A^3*a^4*b^2*d^12*f^2 + 3*A^3*b^6*c^2*d^10*f^2 + 3* A^3*b^6*c^4*d^8*f^2 - 16*A^3*a*b^5*c*d^11*f^2 - 16*A^3*a*b^5*c^3*d^9*f^2 + 12*A^3*a^2*b^4*c^2*d^10*f^2 - A^3*a^2*b^4*c^4*d^8*f^2 + A^3*a^4*b^2*c^2*d^1 0*f^2))/f^5 + ((((A^2*b^2*c - A^2*a*b*d)*((32*(12*A*a*b^7*d^11*f^4 - 12*A*b

^8*c*d^10*f^4 + 24*A*a^3*b^5*d^11*f^4 + 12*A*a^5*b^3*d^11*f^4 - 12*A*b^8*c^3*d^8*f^4 + 12*A*a*b^7*c^2*d^9*f^4 - 24*A*a^2*b^6*c*d^10*f^4 - 12*A*a^4*b^4*c*d^10*f^4 - 24*A*a^2*b^6*c^3*d^8*f^4 + 24*A*a^3*b^5*c^2*d^9*f^4 - 12*A*a^4*b^4*c^3*d^8*f^4 + 12*A*a^5*b^3*c^2*d^9*f^4))/f^5 + (32*(A^2*b^2*c - A^2*a*b*d)*(c + d*tan(e + f*x))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/(f^4*((A^2*b^2*c - A^2*a*b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2*f^2))^(1/2))))/((A^2*b^2*c - A^2*a*b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2*f^2))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^4*d^11*f^2 + 2*A^2*a^5*b^2*d^11*f^2 + 18*A^2*b^7*c^3*d^8*f^2 - 14*A^2*a*b^6*d^11*f^2 + 6*A^2*b^7*c*d^10*f^2 - 18*A^2*a*b^6*c^2*d^9*f^2 - 36*A^2*a^2*b^5*c*d^10*f^2 - 10*A^2*a^4*b^3*c*d^10*f^2 - 12*A^2*a^2*b^5*c^3*d^8*f^2 + 12*A^2*a^3*b^4*c^2*d^9*f^2 + 2*A^2*a^4*b^3*c^3*d^8*f^2 - 2*A^2*a^5*b^2*c^2*d^9*f^2))/f^4)*(A^2*b^2*c - A^2*a*b*d))/((A^2*b^2*c - A^2*a*b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2*f^2))^(1/2))*(A^2*b^2*c - A^2*a*b*d))/((A^2*b^2*c - A^2*a*b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2*f^2))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*d^12 - 2*A^4*a^2*b^3*d^12 + 3*A^4*b^5*c^4*d^8 + 2*A^4*a^2*b^3*c^2*d^10 + 4*A^4*a*b^4*c*d^11 - 4*A^4*a*b^4*c^3*d^9))/f^4)*(A^2*b^2*c - A^2*a*b*d)*1i)/((A^2*b^2*c - A^2*a*b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2*f^2))^(1/2))/((((((32*(13*A^3*a^2*b^4*d^12*f^2 + A^3*a^4*b^2*d^12*f^2 + 3*A^3*b^6*c^2*d^10*f^2 + 3*A^3*b^6*c^4*d^8*f^2 - 16*A^3*a*b^5*c*d^11*f^2 - 16*A^3*a*b^5*c^3*d^9*f^2 + 12*A^3*a^2*b^4*c^2*d^10*f^2 - A^3*a^2*b^4*c^4*d^8*f^2 + A^3*a^4*b^2*c^2*d^10*f^2))/f^5 + ((((A^2*b^2*c - A^2*a*b*d)*((32*(12*A*a*b^7*d^11*f^4 - 12*A*b^8*c*d^10*f^4 + 24*A*a^3*b^5*d^11*f^4 + 12*A*a^5*b^3*d^11*f^4 - 12*A*b^8*c^3*d^8*f^4 + 12*A*a*b^7*c^2*d^9*f^4 - 24*A*a^2*b^6*c*d^10*f^4 - 12*A*a^4*b^4*c*d^10*f^4 - 24*A*a^2*b^6*c^3*d^8*f^4 + 24*A*a^3*b^5*c^2*d^9*f^4 - 12*A*a^4*b^4*c^3*d^8*f^4 + 12*A*a^5*b^3*c^2*d^9*f^4))/f^5 - (32*(A^2*b^2*c - A^2*a*b*d)*(c + d*tan(e + f*x))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/(f^4*((A^2*b^2*c - A^2*a*b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2*f^2))^(1/2))))/((A^2*b^2*c - A^2*a*b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2*f^2))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^4*d^11*f^2 + 2*A^2*a^5*b^2*d^11*f^2 + 18*A^2*b^7*c^3*d^8*f^2 - 14*A^2*a*b^6*d^11*f^2 + 6*A^2*b^7*c*d^10*f^2 - 18*A^2*a*b^6*c^2*d^9*f^2 - 36*A^2*a^2*b^5*c*d^10*f^2 - 10*A^2*a^4*b^3*c*d^10*f^2 - 12*A^2*a^2*b^5*c^3*d^8*f^2 + 12*A^2*a^3*b^4*c^2*d^9*f^2 + 2*A^2*a^4*b^3*c^3*d^8*f^2 - 2*A^2*a^5*b^2*c^2*d^9*f^2))/f^4)*(A^2*b^2*c - A^2*a*b*d))/((A^2*b^2*c - A^2*a*b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2*f^2))^(1/2))*(A^2*b^2*c - A^2*a*b*d))/((A^2*b^2*c - A^2*a*b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2*f^2))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*d^12 - 2*A^4*a^2*b^3*d^12 + 3*A^4*b^5*c^4*d^8 + 2*A^4*a^2*b^3*c^2*d^10 + 4*A^4*a*b^4*c*d^11 - 4*A^4*a*b^4*c^3*d^9))/f^4)*(A^2*b^2*c - A^2*a*b*d))/((A^2*b^2*c - A^2*a*b*d)*(a^4*f^2 +

b^4*f^2 + 2*a^2*b^2*f^2))^(1/2) - (64*(A^5*b^4*c^3*d^10 - A^5*a*b^3*d^13 + A^5*b^4*c*d^12 - A^5*a*b^3*c^2*d^11))/f^5 + (((((32*(13*A^3*a^2*b^4*d^12*f^2 + A^3*a^4*b^2*d^12*f^2 + 3*A^3*b^6*c^2*d^10*f^2 + 3*A^3*b^6*c^4*d^8*f^2 - 16*A^3*a*b^5*c*d^11*f^2 - 16*A^3*a*b^5*c^3*d^9*f^2 + 12*A^3*a^2*b^4*c^2*d^10*f^2 - A^3*a^2*b^4*c^4*d^8*f^2 + A^3*a^4*b^2*c^2*d^10*f^2))/f^5 + ((((A^2*b^2*c - A^2*a*b*d)*((32*(12*A*a*b^7*d^11*f^4 - 12*A*b^8*c*d^10*f^4 + 24*A*a^3*b^5*d^11*f^4 + 12*A*a^5*b^3*d^11*f^4 - 12*A*b^8*c^3*d^8*f^4 + 12*A*a*b^7*c^2*d^9*f^4 - 24*A*a^2*b^6*c*d^10*f^4 - 12*A*a^4*b^4*c*d^10*f^4 - 24*A*a^2*b^6*c^3*d^8*f^4 + 24*A*a^3*b^5*c^2*d^9*f^4 - 12*A*a^4*b^4*c^3*d^8*f^4 + 12*A*a^5*b^3*c^2*d^9*f^4))/f^5 + (32*(A^2*b^2*c - A^2*a*b*d)*(c + d*tan(e + f*x))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/(f^4*((A^2*b^2*c - A^2*a*b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2*f^2))^(1/2))))/((A^2*b^2*c - A^2*a*b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2*f^2))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^4*d^11*f^2 + 2*A^2*a^5*b^2*d^11*f^2 + 18*A^2*b^7*c^3*d^8*f^2 - 14*A^2*a*b^6*d^11*f^2 + 6*A^2*b^7*c*d^10*f^2 - 18*A^2*a*b^6*c^2*d^9*f^2 - 36*A^2*a^2*b^5*c*d^10*f^2 - 10*A^2*a^4*b^3*c*d^10*f^2 - 12*A^2*a^2*b^5*c^3*d^8*f^2 + 12*A^2*a^3*b^4*c^2*d^9*f^2 + 2*A^2*a^4*b^3*c^3*d^8*f^2 - 2*A^2*a^5*b^2*c^2*d^9*f^2))/f^4)*(A^2*b^2*c - A^2*a*b*d))/((A^2*b^2*c - A^2*a*b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2*f^2))^(1/2))*(A^2*b^2*c - A^2*a*b*d))/((A^2*b^2*c - A^2*a*b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2*f^2))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*d^12 - 2*A^4*a^2*b^3*d^12 + 3*A^4*b^5*c^4*d^8 + 2*A^4*a^2*b^3*c^2*d^10 + 4*A^4*a*b^4*c*d^11 - 4*A^4*a*b^4*c^3*d^9))/f^4)*(A^2*b^2*c - A^2*a*b*d))/((A^2*b^2*c - A^2*a*b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2*f^2))^(1/2)))*(A^2*b^2*c - A^2*a*b*d)*2i)/((A^2*b^2*c - A^2*a*b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2*f^2))^(1/2) + (2*C*(c + d*tan(e + f*x))^(1/2))/(b*f) - (atan(((((32*(c + d*tan(e + f*x))^(1/2)*(B^4*b^5*d^12 + 2*B^4*b^5*c^2*d^10 + B^4*b^5*c^4*d^8 + 2*B^4*a^4*b*d^12 + 2*B^4*a^2*b^3*c^2*d^10 - 2*B^4*a^2*b^3*c^4*d^8 + 4*B^4*a^3*b^2*c^3*d^9 - 4*B^4*a^3*b^2*c*d^11 - 2*B^4*a^4*b*c^2*d^10))/f^4 + (((32*(15*B^3*a^3*b^3*d^12*f^2 + B^3*b^6*c^3*d^9*f^2 - B^3*a*b^5*d^12*f^2 - 4*B^3*a^5*b*d^12*f^2 + B^3*b^6*c*d^11*f^2 + 6*B^3*a*b^5*c^2*d^10*f^2 + 7*B^3*a*b^5*c^4*d^8*f^2 - 22*B^3*a^2*b^4*c*d^11*f^2 + 9*B^3*a^4*b^2*c*d^11*f^2 - 4*B^3*a^5*b*c^2*d^10*f^2 - 22*B^3*a^2*b^4*c^3*d^9*f^2 + 10*B^3*a^3*b^3*c^2*d^10*f^2 - 5*B^3*a^3*b^3*c^4*d^8*f^2 + 9*B^3*a^4*b^2*c^3*d^9*f^2))/f^5 - ((-(B^2*a^3*d - B^2*a^2*b*c)*(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(14*B^2*a^5*b^2*d^11*f^2 - 4*B^2*a^3*b^4*d^11*f^2 - 10*B^2*b^7*c^3*d^8*f^2 + 14*B^2*a*b^6*d^11*f^2 - 6*B^2*b^7*c*d^10*f^2 - 8*B^2*a^6*b*c*d^10*f^2 + 18*B^2*a*b^6*c^2*d^9*f^2 + 12*B^2*a^2*b^5*c*d^10*f^2 - 22*B^2*a^4*b^3*c*d^10*f^2 + 12*B^2*a^2*b^5*c^3*d^8*f^2 + 4*B^2*a^3*b^4*c^2*d^9*f^2 - 10*B^2*a^4*b^3*c^3*d^8*f^2 + 18*B^2*a^5*b^2*c^2*d^9*f^2))/f^4 + (((32*(4*B*a^2*b^6*d^11*f^4 + 8*B*a^4*b^4*d^11*f^4 + 4*B*a^6*b^2*d^11*f^4 - 4*B*a*b^7*c^3*d^8*f^4 - 8*B*a^3*b^5*c*d^10*f^4 - 4*B*a^5*b^3*c*d^10*f^4 + 4*B*a^2*b^6*c^2*d^9*f^4 - 8*B*a^3*b

^5*c^3*d^8*f^4 + 8*B*a^4*b^4*c^2*d^9*f^4 - 4*B*a^5*b^3*c^3*d^8*f^4 + 4*B*a^6*b^2*c^2*d^9*f^4 - 4*B*a*b^7*c*d^10*f^4))/f^5 - (32*(-(B^2*a^3*d - B^2*a^2*b*c)*(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/(b*f^6*(a^2 + b^2)^2))*(-(B^2*a^3*d - B^2*a^2*b*c)*(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2))/(b*f^2*(a^2 + b^2)^2)))/(b*f^2*(a^2 + b^2)^2))*(-(B^2*a^3*d - B^2*a^2*b*c)*(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2))/(b*f^2*(a^2 + b^2)^2))*(-(B^2*a^3*d - B^2*a^2*b*c)*(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2)*1i)/(b*f^2*(a^2 + b^2)^2) + (((32*(c + d*tan(e + f*x))^(1/2)*(B^4*b^5*d^12 + 2*B^4*b^5*c^2*d^10 + B^4*b^5*c^4*d^8 + 2*B^4*a^4*b*d^12 + 2*B^4*a^2*b^3*c^2*d^10 - 2*B^4*a^2*b^3*c^4*d^8 + 4*B^4*a^3*b^2*c^3*d^9 - 4*B^4*a^3*b^2*c*d^11 - 2*B^4*a^4*b*c^2*d^10))/f^4 - (((32*(15*B^3*a^3*b^3*d^12*f^2 + B^3*b^6*c^3*d^9*f^2 - B^3*a*b^5*d^12*f^2 - 4*B^3*a^5*b*d^12*f^2 + B^3*b^6*c*d^11*f^2 + 6*B^3*a*b^5*c^2*d^10*f^2 + 7*B^3*a*b^5*c^4*d^8*f^2 - 22*B^3*a^2*b^4*c*d^11*f^2 + 9*B^3*a^4*b^2*c*d^11*f^2 - 4*B^3*a^5*b*c^2*d^10*f^2 - 22*B^3*a^2*b^4*c^3*d^9*f^2 + 10*B^3*a^3*b^3*c^2*d^10*f^2 - 5*B^3*a^3*b^3*c^4*d^8*f^2 + 9*B^3*a^4*b^2*c^3*d^9*f^2))/f^5 + ((-(B^2*a^3*d - B^2*a^2*b*c)*(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(14*B^2*a^5*b^2*d^11*f^2 - 4*B^2*a^3*b^4*d^11*f^2 - 10*B^2*b^7*c^3*d^8*f^2 + 14*B^2*a*b^6*d^11*f^2 - 6*B^2*b^7*c*d^10*f^2 - 8*B^2*a^6*b*c*d^10*f^2 + 18*B^2*a*b^6*c^2*d^9*f^2 + 12*B^2*a^2*b^5*c*d^10*f^2 - 22*B^2*a^4*b^3*c*d^10*f^2 + 12*B^2*a^2*b^5*c^3*d^8*f^2 + 4*B^2*a^3*b^4*c^2*d^9*f^2 - 10*B^2*a^4*b^3*c^3*d^8*f^2 + 18*B^2*a^5*b^2*c^2*d^9*f^2))/f^4 - (((32*(4*B*a^2*b^6*d^11*f^4 + 8*B*a^4*b^4*d^11*f^4 + 4*B*a^6*b^2*d^11*f^4 - 4*B*a*b^7*c^3*d^8*f^4 - 8*B*a^3*b^5*c*d^10*f^4 - 4*B*a^5*b^3*c*d^10*f^4 + 4*B*a^2*b^6*c^2*d^9*f^4 - 8*B*a^3*b^5*c^3*d^8*f^4 + 8*B*a^4*b^4*c^2*d^9*f^4 - 4*B*a^5*b^3*c^3*d^8*f^4 + 4*B*a^6*b^2*c^2*d^9*f^4 - 4*B*a*b^7*c*d^10*f^4))/f^5 + (32*(-(B^2*a^3*d - B^2*a^2*b*c)*(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/(b*f^6*(a^2 + b^2)^2))*(-(B^2*a^3*d - B^2*a^2*b*c)*(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2))/(b*f^2*(a^2 + b^2)^2)))/(b*f^2*(a^2 + b^2)^2))*(-(B^2*a^3*d - B^2*a^2*b*c)*(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2))/(b*f^2*(a^2 + b^2)^2))*(-(B^2*a^3*d - B^2*a^2*b*c)*(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2)*1i)/(b*f^2*(a^2 + b^2)^2))/((64*(B^5*a*b^3*c*d^12 - 3*B^5*a^2*b^2*c^2*d^11 - 2*B^5*a^2*b^2*c^4*d^9 - B^5*a^2*b^2*d^13 + B^5*a^3*b*c*d^12 + 2*B^5*a*b^3*c^3*d^10 + B^5*a*b^3*c^5*d^8 + B^5*a^3*b*c^3*d^10))/f^5 + (((32*(c + d*tan(e + f*x))^(1/2)*(B^4*b^5*d^12 + 2*B^4*b^5*c^2*d^10 + B^4*b^5*c^4*d^8 + 2*B^4*a^4*b*d^12 + 2*B^4*a^2*b^3*c^2*d^10 - 2*B^4*a^2*b^3*c^4*d^8 + 4*B^4*a^3*b^2*c^3*d^9 - 4*B^4*a^3*b^2*c*d^11 - 2*B^4*a^4*b*c^2*d^10))/f^4 + (((32*(15*B^3*a^3*b^3*d^12*f^2 + B^3

*b^6*c^3*d^9*f^2 - B^3*a*b^5*d^12*f^2 - 4*B^3*a^5*b*d^12*f^2 + B^3*b^6*c*d^
11*f^2 + 6*B^3*a*b^5*c^2*d^10*f^2 + 7*B^3*a*b^5*c^4*d^8*f^2 - 22*B^3*a^2*b^
4*c*d^11*f^2 + 9*B^3*a^4*b^2*c*d^11*f^2 - 4*B^3*a^5*b*c^2*d^10*f^2 - 22*B^3
*a^2*b^4*c^3*d^9*f^2 + 10*B^3*a^3*b^3*c^2*d^10*f^2 - 5*B^3*a^3*b^3*c^4*d^8*
f^2 + 9*B^3*a^4*b^2*c^3*d^9*f^2))/f^5 - ((-(B^2*a^3*d - B^2*a^2*b*c)*(b^5*f
^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(14*
B^2*a^5*b^2*d^11*f^2 - 4*B^2*a^3*b^4*d^11*f^2 - 10*B^2*b^7*c^3*d^8*f^2 + 14
*B^2*a*b^6*d^11*f^2 - 6*B^2*b^7*c*d^10*f^2 - 8*B^2*a^6*b*c*d^10*f^2 + 18*B^
2*a*b^6*c^2*d^9*f^2 + 12*B^2*a^2*b^5*c*d^10*f^2 - 22*B^2*a^4*b^3*c*d^10*f^2
 + 12*B^2*a^2*b^5*c^3*d^8*f^2 + 4*B^2*a^3*b^4*c^2*d^9*f^2 - 10*B^2*a^4*b^3*
c^3*d^8*f^2 + 18*B^2*a^5*b^2*c^2*d^9*f^2))/f^4 + (((32*(4*B*a^2*b^6*d^11*f^
4 + 8*B*a^4*b^4*d^11*f^4 + 4*B*a^6*b^2*d^11*f^4 - 4*B*a*b^7*c^3*d^8*f^4 - 8
*B*a^3*b^5*c*d^10*f^4 - 4*B*a^5*b^3*c*d^10*f^4 + 4*B*a^2*b^6*c^2*d^9*f^4 -
8*B*a^3*b^5*c^3*d^8*f^4 + 8*B*a^4*b^4*c^2*d^9*f^4 - 4*B*a^5*b^3*c^3*d^8*f^4
 + 4*B*a^6*b^2*c^2*d^9*f^4 - 4*B*a*b^7*c*d^10*f^4))/f^5 - (32*(-(B^2*a^3*d
- B^2*a^2*b*c)*(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2)*(c + d*tan(e +
f*x))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 -
16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b
^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d
^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/(b*f^6*(a^2 + b^2)^2)
)*(-(B^2*a^3*d - B^2*a^2*b*c)*(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2))
/(b*f^2*(a^2 + b^2)^2)))/(b*f^2*(a^2 + b^2)^2))*(-(B^2*a^3*d - B^2*a^2*b*c)
*(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2))/(b*f^2*(a^2 + b^2)^2))*(-(B^
2*a^3*d - B^2*a^2*b*c)*(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2))/(b*f^2
(a^2 + b^2)^2) - (((32(c + d*tan(e + f*x))^(1/2)*(B^4*b^5*d^12 + 2*B^4*b^
5*c^2*d^10 + B^4*b^5*c^4*d^8 + 2*B^4*a^4*b*d^12 + 2*B^4*a^2*b^3*c^2*d^10 -
2*B^4*a^2*b^3*c^4*d^8 + 4*B^4*a^3*b^2*c^3*d^9 - 4*B^4*a^3*b^2*c*d^11 - 2*B^
4*a^4*b*c^2*d^10))/f^4 - (((32*(15*B^3*a^3*b^3*d^12*f^2 + B^3*b^6*c^3*d^9*f
^2 - B^3*a*b^5*d^12*f^2 - 4*B^3*a^5*b*d^12*f^2 + B^3*b^6*c*d^11*f^2 + 6*B^3
*a*b^5*c^2*d^10*f^2 + 7*B^3*a*b^5*c^4*d^8*f^2 - 22*B^3*a^2*b^4*c*d^11*f^2 +
 9*B^3*a^4*b^2*c*d^11*f^2 - 4*B^3*a^5*b*c^2*d^10*f^2 - 22*B^3*a^2*b^4*c^3*d
^9*f^2 + 10*B^3*a^3*b^3*c^2*d^10*f^2 - 5*B^3*a^3*b^3*c^4*d^8*f^2 + 9*B^3*a^
4*b^2*c^3*d^9*f^2))/f^5 + ((-(B^2*a^3*d - B^2*a^2*b*c)*(b^5*f^2 + a^4*b*f^2
 + 2*a^2*b^3*f^2))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(14*B^2*a^5*b^2*d^
11*f^2 - 4*B^2*a^3*b^4*d^11*f^2 - 10*B^2*b^7*c^3*d^8*f^2 + 14*B^2*a*b^6*d^1
1*f^2 - 6*B^2*b^7*c*d^10*f^2 - 8*B^2*a^6*b*c*d^10*f^2 + 18*B^2*a*b^6*c^2*d^
9*f^2 + 12*B^2*a^2*b^5*c*d^10*f^2 - 22*B^2*a^4*b^3*c*d^10*f^2 + 12*B^2*a^2*
b^5*c^3*d^8*f^2 + 4*B^2*a^3*b^4*c^2*d^9*f^2 - 10*B^2*a^4*b^3*c^3*d^8*f^2 +
18*B^2*a^5*b^2*c^2*d^9*f^2))/f^4 - (((32*(4*B*a^2*b^6*d^11*f^4 + 8*B*a^4*b^
4*d^11*f^4 + 4*B*a^6*b^2*d^11*f^4 - 4*B*a*b^7*c^3*d^8*f^4 - 8*B*a^3*b^5*c*d
^10*f^4 - 4*B*a^5*b^3*c*d^10*f^4 + 4*B*a^2*b^6*c^2*d^9*f^4 - 8*B*a^3*b^5*c^
3*d^8*f^4 + 8*B*a^4*b^4*c^2*d^9*f^4 - 4*B*a^5*b^3*c^3*d^8*f^4 + 4*B*a^6*b^2
*c^2*d^9*f^4 - 4*B*a*b^7*c*d^10*f^4))/f^5 + (32*(-(B^2*a^3*d - B^2*a^2*b*c)
*(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(1
6*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^1

```
0*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4
 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^
5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/(b*f^6*(a^2 + b^2)^2))*(-(B^2*a^3*d
 - B^2*a^2*b*c)*(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2))/(b*f^2*(a^2 +
 b^2)^2)))/(b*f^2*(a^2 + b^2)^2))*(-(B^2*a^3*d - B^2*a^2*b*c)*(b^5*f^2 + a^
4*b*f^2 + 2*a^2*b^3*f^2))^(1/2))/(b*f^2*(a^2 + b^2)^2))*(-(B^2*a^3*d - B^2*
a^2*b*c)*(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2))/(b*f^2*(a^2 + b^2)^2
)))*(-(B^2*a^3*d - B^2*a^2*b*c)*(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2
)*2i)/(b*f^2*(a^2 + b^2)^2)
```

3.95 $\int \frac{\sqrt{c+d\tan(e+fx)}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx$

Optimal result 1195
Rubi [A] (verified) 1196
Mathematica [B] (verified) 1199
Maple [B] (verified) 1200
Fricas [F(-1)] 1200
Sympy [F] 1201
Maxima [F(-2)] 1201
Giac [F(-1)] 1201
Mupad [B] (verification not implemented) 1202

Optimal result

Integrand size = 47, antiderivative size = 317

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx$$
$$= -\frac{(iA+B-iC)\sqrt{c-id}\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(a-ib)^2f}$$
$$-\frac{(B-i(A-C))\sqrt{c+id}\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(a+ib)^2f}$$
$$-\frac{(a^3bBd+a^4Cd+b^4(2Bc+Ad)+ab^3(4Ac-4cC-3Bd)-a^2b^2(2Bc+3Ad-5Cd))\operatorname{arctanh}\left(\frac{\sqrt{b}\sqrt{c+}}{\sqrt{}}\right.}{b^{3/2}(a^2+b^2)^2\sqrt{bc-ad}f}$$
$$-\frac{(Ab^2-a(bB-aC))\sqrt{c+d\tan(e+fx)}}{b(a^2+b^2)f(a+b\tan(e+fx))}$$

```
[Out] -(I*A+B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))*(c-I*d)^(1/2)/(a
-I*b)^2/f-(B-I*(A-C))*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))*(c+I*d)
^(1/2)/(a+I*b)^2/f-(a^3*b*B*d+a^4*C*d+b^4*(A*d+2*B*c)+a*b^3*(4*A*c-3*B*d-4*
C*c)-a^2*b^2*(3*A*d+2*B*c-5*C*d))*arctanh(b^(1/2)*(c+d*tan(f*x+e))^(1/2)/(-
a*d+b*c)^(1/2))/b^(3/2)/(a^2+b^2)^2/f/(-a*d+b*c)^(1/2)-(A*b^2-a*(B*b-C*a))*
(c+d*tan(f*x+e))^(1/2)/b/(a^2+b^2)/f/(a+b*tan(f*x+e))
```

Rubi [A] (verified)

Time = 1.54 (sec) , antiderivative size = 317, normalized size of antiderivative = 1.00, number of steps used = 12, number of rules used = 7, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.149, Rules used = {3726, 3734, 3620, 3618, 65, 214, 3715}

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx$$
$$= -\frac{(Ab^2-a(bB-aC))\sqrt{c+d\tan(e+fx)}}{bf(a^2+b^2)(a+b\tan(e+fx))}$$
$$-\frac{(a^4Cd+a^3bBd-a^2b^2(3Ad+2Bc-5Cd)+ab^3(4Ac-3Bd-4cC)+b^4(Ad+2Bc))\operatorname{arctanh}\left(\frac{\sqrt{b}\sqrt{c+d\,t\cdots}}{\sqrt{bc\cdots}}\right.}{b^{3/2}f(a^2+b^2)^2\sqrt{bc-ad}}$$
$$-\frac{\sqrt{c-id}(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f(a-ib)^2}$$
$$-\frac{\sqrt{c+id}(B-i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f(a+ib)^2}$$

```
[In] Int[(Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a +
 b*Tan[e + f*x])^2,x]
```

```
[Out] -(((I*A + B - I*C)*Sqrt[c - I*d]*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c -
I*d]])/((a - I*b)^2*f)) - ((B - I*(A - C))*Sqrt[c + I*d]*ArcTanh[Sqrt[c + d
*Tan[e + f*x]]/Sqrt[c + I*d]])/((a + I*b)^2*f) - ((a^3*b*B*d + a^4*C*d + b^
4*(2*B*c + A*d) + a*b^3*(4*A*c - 4*c*C - 3*B*d) - a^2*b^2*(2*B*c + 3*A*d -
5*C*d))*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*d]])/(b^(3/
2)*(a^2 + b^2)^2*Sqrt[b*c - a*d]*f) - ((A*b^2 - a*(b*B - a*C))*Sqrt[c + d*T
an[e + f*x]])/(b*(a^2 + b^2)*f*(a + b*Tan[e + f*x]))
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
```

```
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3715

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_) + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :>
 Dist[A/f, Subst[Int[(a + b*x)^m*(c + d*x)^n, x], x, Tan[e + f*x]], x] /; F
reeQ[{a, b, c, d, e, f, A, C, m, n}, x] && EqQ[A, C]
```

Rule 3726

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*d^2 + c*(c*C - B*d))*(a + b*Tan[e +
 f*x])^m*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3734

```
Int[(((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2))/((a_.) + (b_.)*tan[(e_.)
+ (f_.)*(x_)]), x_Symbol] :> Dist[1/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n
*Simp[b*B + a*(A - C) + (a*B - b*(A - C))*Tan[e + f*x], x], x], x] + Dist[(
A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n*((1 + Tan[e
+ f*x]^2)/(a + b*Tan[e + f*x])), x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C
, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&
 !GtQ[n, 0] &&  !LeQ[n, -1]
```

Rubi steps

$$\text{integral} = -\frac{(Ab^2 - a(bB - aC))\sqrt{c + d\tan(e + fx)}}{b(a^2 + b^2) f(a + b\tan(e + fx))} + \frac{\int \frac{\frac{1}{2}\left(2(bB-aC)\left(bc-\frac{ad}{2}\right)+2Ab\left(ac+\frac{bd}{2}\right)\right)-b((A-C)(bc-ad)-B(ac+bd))\tan(e+fx)-\frac{1}{2}\left(Ab^2-abB-a^2C-2b^2C\right)d\tan^2(e+fx)}{(a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}}\,dx}{b(a^2 + b^2)}$$

$$= -\frac{(Ab^2 - a(bB - aC))\sqrt{c + d\tan(e + fx)}}{b(a^2 + b^2) f(a + b\tan(e + fx))} + \frac{\int \frac{b(a^2(Ac-cC-Bd)-b^2(Ac-cC-Bd)+2ab(Bc+(A-C)d))-b(2ab(Ac-cC-Bd)-a^2(Bc+(A-C)d)+b^2(Bc+(A-C)d))\tan(e+fx}{\sqrt{c+d\tan(e+fx)}}}{b(a^2 + b^2)^2} + \frac{(a^3bBd + a^4Cd + b^4(2Bc + Ad) + ab^3(4Ac - 4cC - 3Bd) - a^2b^2(2Bc + 3Ad - 5Cd))\int \frac{}{(a+b\tan(}}{2b(a^2 + b^2)^2}$$

$$= -\frac{(Ab^2 - a(bB - aC))\sqrt{c + d\tan(e + fx)}}{b(a^2 + b^2) f(a + b\tan(e + fx))} + \frac{((A - iB - C)(c - id))\int \frac{1+i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(a - ib)^2} + \frac{((A + iB - C)(c + id))\int \frac{1-i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(a + ib)^2} + \frac{(a^3bBd + a^4Cd + b^4(2Bc + Ad) + ab^3(4Ac - 4cC - 3Bd) - a^2b^2(2Bc + 3Ad - 5Cd))\,\text{Subst}\left(\int\right.}{2b(a^2 + b^2)^2 f}$$

$$= -\frac{(Ab^2 - a(bB - aC))\sqrt{c + d\tan(e + fx)}}{b(a^2 + b^2) f(a + b\tan(e + fx))} - \frac{(i(A + iB - C)(c + id))\text{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c+idx}}\,dx, x, -i\tan(e + fx)\right)}{2(a + ib)^2 f} + \frac{((A - iB - C)(ic + d))\text{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c-idx}}\,dx, x, i\tan(e + fx)\right)}{2(a - ib)^2 f} + \frac{(a^3bBd + a^4Cd + b^4(2Bc + Ad) + ab^3(4Ac - 4cC - 3Bd) - a^2b^2(2Bc + 3Ad - 5Cd))\,\text{Subst}\left(\int\right.}{b(a^2 + b^2)^2 df}$$

$$
\begin{aligned}
&= -\frac{(a^3bBd + a^4Cd + b^4(2Bc + Ad) + ab^3(4Ac - 4cC - 3Bd) - a^2b^2(2Bc + 3Ad - 5Cd))\operatorname{arctanh}}{b^{3/2}\left(a^2+b^2\right)^2\sqrt{bc-ad}f} \\
&\quad - \frac{(Ab^2 - a(bB - aC))\sqrt{c + d\tan(e + fx)}}{b\left(a^2+b^2\right)f(a + b\tan(e + fx))} \\
&\quad - \frac{((A + iB - C)(c + id))\text{Subst}\left(\int \frac{1}{-1+\frac{ic}{d}-\frac{ix^2}{d}}\,dx, x, \sqrt{c + d\tan(e + fx)}\right)}{(a + ib)^2 df} \\
&\quad + \frac{((iA + B - iC)(ic + d))\text{Subst}\left(\int \frac{1}{-1-\frac{ic}{d}+\frac{ix^2}{d}}\,dx, x, \sqrt{c + d\tan(e + fx)}\right)}{(a - ib)^2 df} \\
&= -\frac{(B + i(A - C))\sqrt{c - id}\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(a - ib)^2 f} \\
&\quad - \frac{(B - i(A - C))\sqrt{c + id}\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(a + ib)^2 f} \\
&\quad - \frac{(a^3bBd + a^4Cd + b^4(2Bc + Ad) + ab^3(4Ac - 4cC - 3Bd) - a^2b^2(2Bc + 3Ad - 5Cd))\operatorname{arctanh}}{b^{3/2}\left(a^2+b^2\right)^2\sqrt{bc-ad}f} \\
&\quad - \frac{(Ab^2 - a(bB - aC))\sqrt{c + d\tan(e + fx)}}{b\left(a^2+b^2\right)f(a + b\tan(e + fx))}
\end{aligned}
$$

Mathematica [B] (verified)

Both result and optimal contain complex but leaf count is larger than twice the leaf count of optimal. 764 vs. 2(317) = 634.

Time = 6.46 (sec) , antiderivative size = 764, normalized size of antiderivative = 2.41

$$
\int \frac{\sqrt{c + d\tan(e + fx)}(A + B\tan(e + fx) + C\tan^2(e + fx))}{(a + b\tan(e + fx))^2}\,dx = -\frac{2C\sqrt{c + d\tan(e + fx)}}{bf(a + b\tan(e + fx))}
$$

$$
-\frac{2\left(-\frac{\frac{i\sqrt{c-id}\left(\frac{1}{2}b(bc-ad)\left(a^2(Ac-cC-Bd)-b^2(Ac-cC-Bd)+2ab(Bc+(A-C)d)\right)+\frac{1}{2}ib(bc-ad)\left(2ab(Ac-cC-Bd)-a^2(Bc+(A-C)d)+b^2(Bc+(A-C)d)\right)\right)}{(-c+id)f}}{}\right.}{}
$$

```
[In] Integrate[(Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)
)/(a + b*Tan[e + f*x])^2,x]
```

```
[Out] (-2*C*Sqrt[c + d*Tan[e + f*x]])/(b*f*(a + b*Tan[e + f*x])) - (2*(-((((I*Sqr
t[c - I*d]*((b*(b*c - a*d)*(a^2*(A*c - c*C - B*d) - b^2*(A*c - c*C - B*d) +
```

```
2*a*b*(B*c + (A - C)*d)))/2 + (I/2)*b*(b*c - a*d)*(2*a*b*(A*c - c*C - B*d)
 - a^2*(B*c + (A - C)*d) + b^2*(B*c + (A - C)*d)))*ArcTanh[Sqrt[c + d*Tan[e
 + f*x]]/Sqrt[c - I*d]])/((-c + I*d)*f) - (I*Sqrt[c + I*d]*((b*(b*c - a*d)*
(a^2*(A*c - c*C - B*d) - b^2*(A*c - c*C - B*d) + 2*a*b*(B*c + (A - C)*d)))/
2 - (I/2)*b*(b*c - a*d)*(2*a*b*(A*c - c*C - B*d) - a^2*(B*c + (A - C)*d) +
b^2*(B*c + (A - C)*d)))*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]])/((
-c - I*d)*f))/(a^2 + b^2) + (2*Sqrt[b*c - a*d]*(-1/4*(a^2*(A*b^2 - a*b*B -
a^2*C - 2*b^2*C)*d*(b*c - a*d)) + (a*b^2*(b*c - a*d)*(A*b*c - a*B*c - b*c*C
 - a*A*d - b*B*d + a*C*d))/2 + (b^2*(b*c - a*d)*(a^2*C*d + b^2*(2*B*c + A*d
) + a*b*(2*A*c - 2*c*C - B*d)))/4)*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]
])/Sqrt[b*c - a*d]])/(Sqrt[b]*(a^2 + b^2)*(-(b*c) + a*d)*f))/((a^2 + b^2)*(
b*c - a*d))) - (((b^2*(-(A*b*c) + 2*b*c*C - a*C*d))/2 - a*(-1/2*(b^2*(B*c +
 (A - C)*d)) - (a*(b*c*C - b*B*d - a*C*d))/2))*Sqrt[c + d*Tan[e + f*x]])/((
a^2 + b^2)*(b*c - a*d)*f*(a + b*Tan[e + f*x]))))/b
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 5777 vs. $2(284) = 568$.

Time = 0.13 (sec) , antiderivative size = 5778, normalized size of antiderivative = 18.23

method	result	size
derivativedivides	Expression too large to display	5778
default	Expression too large to display	5778

```
[In] int((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^2,x,method=_RETURNVERBOSE)
```

```
[Out] result too large to display
```

Fricas [F(-1)]

Timed out.

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^2,x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx = \int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx$$

```
[In] integrate((c+d*tan(f*x+e))**(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e))**2,x)
```

```
[Out] Integral(sqrt(c + d*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/
(a + b*tan(e + f*x))**2, x)
```

Maxima [F(-2)]

Exception generated.

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx = \text{Exception raised: ValueError}$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^2,x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 detail
```

Giac [F(-1)]

Timed out.

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^2,x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [B] (verification not implemented)

Time = 42.93 (sec) , antiderivative size = 138318, normalized size of antiderivative = 436.33

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx = \text{Too large to display}$$

[In] int(((c + d*tan(e + f*x))^(1/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a + b*tan(e + f*x))^2,x)

[Out] atan(((((8*(156*B^3*a^2*b^9*d^12*f^2 - 16*B^3*a^4*b^7*d^12*f^2 - 120*B^3*a^6*b^5*d^12*f^2 + 48*B^3*a^8*b^3*d^12*f^2 + 12*B^3*b^11*c^2*d^10*f^2 + 12*B^3*b^11*c^4*d^8*f^2 - 4*B^3*a^10*b*d^12*f^2 - 124*B^3*a*b^10*c*d^11*f^2 - 124*B^3*a*b^10*c^3*d^9*f^2 + 224*B^3*a^3*b^8*c*d^11*f^2 + 200*B^3*a^5*b^6*c*d^11*f^2 - 128*B^3*a^7*b^4*c*d^11*f^2 + 20*B^3*a^9*b^2*c*d^11*f^2 - 4*B^3*a^10*b*c^2*d^10*f^2 + 44*B^3*a^2*b^9*c^2*d^10*f^2 - 112*B^3*a^2*b^9*c^4*d^8*f^2 + 224*B^3*a^3*b^8*c^3*d^9*f^2 - 40*B^3*a^4*b^7*c^2*d^10*f^2 - 24*B^3*a^4*b^7*c^4*d^8*f^2 + 200*B^3*a^5*b^6*c^3*d^9*f^2 - 40*B^3*a^6*b^5*c^2*d^10*f^2 + 80*B^3*a^6*b^5*c^4*d^8*f^2 - 128*B^3*a^7*b^4*c^3*d^9*f^2 + 28*B^3*a^8*b^3*c^2*d^10*f^2 - 20*B^3*a^8*b^3*c^4*d^8*f^2 + 20*B^3*a^9*b^2*c^3*d^9*f^2))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((8*(80*B*a*b^14*d^11*f^4 - 48*B*b^15*c*d^10*f^4 + 384*B*a^3*b^12*d^11*f^4 + 720*B*a^5*b^10*d^11*f^4 + 640*B*a^7*b^8*d^11*f^4 + 240*B*a^9*b^6*d^11*f^4 - 16*B*a^13*b^2*d^11*f^4 - 48*B*b^15*c^3*d^8*f^4 + 80*B*a*b^14*c^2*d^9*f^4 - 224*B*a^2*b^13*c*d^10*f^4 - 400*B*a^4*b^11*c*d^10*f^4 - 320*B*a^6*b^9*c*d^10*f^4 - 80*B*a^8*b^7*c*d^10*f^4 + 32*B*a^10*b^5*c*d^10*f^4 + 16*B*a^12*b^3*c*d^10*f^4 - 224*B*a^2*b^13*c^3*d^8*f^4 + 384*B*a^3*b^12*c^2*d^9*f^4 - 400*B*a^4*b^11*c^3*d^8*f^4 + 720*B*a^5*b^10*c^2*d^9*f^4 - 320*B*a^6*b^9*c^3*d^8*f^4 + 640*B*a^7*b^8*c^2*d^9*f^4 - 80*B*a^8*b^7*c^3*d^8*f^4 + 240*B*a^9*b^6*c^2*d^9*f^4 + 32*B*a^10*b^5*c^3*d^8*f^4 + 16*B*a^12*b^3*c^3*d^8*f^4 - 16*B*a^13*b^2*c^2*d^9*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) - (16*(c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*(32*b^17*d^10*f^4 + 160*a^2*b^15*d^10*f^4 + 288*a^4*b^13*d^10*f^4 + 160*a^6*b^11*d^10*f^4 - 160*a^8*b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b^5*d^10*f^4 - 32*a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^9*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14*b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336*a^5*b^12*c*d^9*f^4 + 560*a^7*b^10*c*d^9*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*a^11*b^6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2*c*d^9*f^4))/(a^8*f^4 + b^8* f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(44*B^2*a^9*b^4*d^11*f^2 - 168*B^2*a^5*b^8*d^11*f^2 - 40*B^2*a^7*b^6*d^11*f^2 - 20*B^2*a^3*b^10*d^11*f^2 - 4*B^2*a^11*b^2*d^11*f^2 - 36*B^2*b^13*c^3*d^8*f^2 + 60*B^2*a*b^12*d^11*f^2 - 12*B^2*b^13*c*d^10*f^2 + 4*B^2*a^12*b*c*d^10*f^2 + 100*B^2*a*b^12*c^2*d^9*f^2 + 120*B^2*a^2*b^11*c*d^10*f^2 + 156*B^2*a^4*b^9*c*d^10*f^2 - 112*B^2*a^6*b^7*c*d^10*f^2 - 148*B^2*a^8*b^5*c*d^10*f^2 - 8*B^2*a^10*b^3*c*d^10*f^2 + 68*B^2*a^2*b^11*c^3*d^8*f^2 + 124*B^2*a^3*b^10*c^2*d^9*f^2 + 184*B^2*a^4*b^9*c^3*d^8*f^2 + 8*B^2*a^5*b^8*c^2*d^9*f^2 + 40*B^2*a^6*b^7*c^3*d^8*f^2 + 24*B^2*a^7*b^6*c^2*d^9*f^2 - 20*B^2*a^8*b^5*c^3*d^8*f^2 + 20*B^2*a^9*b^4*c^2*d^9*f^2 + 20*B^2*a^10*b^3*c^3*d^8*f^2 - 20*B^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2))*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(2*B^4*b^9*d^12 - 5*B^4*a^2*b^7*d^12 + 17*B^4*a^4*b^5*d^12 - 7*B^4*a^6*b^3*d^12 + 6*B^4*b^9*c^4*d^8 + B^4*a^8*b*d^12 + 77*B^4*a^2*b^7*c^2*d^10 - 8*B^4*a^2*b^7*c^4*d^8 + 60*B^4*a^3*b^6*c^3*d^9 - 87*B^4*a^4*b^5*c^2*d^10 + 14*B^4*a^4*b^5*c^4*d^8 - 36*B^4*a^5*b^4*c^3*d^9 + 27*B^4*a^6*b^3*c^2*d^10 - 4*B^4*a^6*b^3*c^4*d^8 + 4*B^4*a^7*b^2*c^3*d^9 + 12*B^4*a*b^8*c*d^11 - 28*B^4*a*b^8*c^3*d^9 - 64*B^4*a^3*b^6*c*d^11 + 44*B^4*a^5*b^4*c*d^11 - 8*B^4*a^7*b^2*c*d^11 - B^4*a^8*b*c^2*d^10))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*1i - (((8*(156*B^3*a^2*b^9*d^12*f^2 - 16*B^3*a^4*b^7*d^12*f^2 - 120*B^3*a^6*b^5*d^12*f^2 + 48*B^3*a^8*b^3*d^12*f^2 + 12*B^3*b^11*c^2*d^10*f^2 + 12*B^3*b^11*c^4*d^8*f^2 - 4*B^3*a^10*b*d^12*f^2 - 124*B^3*a*b^10*c*d^11*f^2 - 124*B^3*a*b^10*c^3*d^9*f^2 + 224*B^3*a^3*b^8*c*d^11*f^2 + 200*B^3*a^5*b^6*c*d^11*f^2 - 128*B^3*a^7*b^4*c*d^11*f^2 + 20*B^

3*a^9*b^2*c*d^11*f^2 - 4*B^3*a^10*b*c^2*d^10*f^2 + 44*B^3*a^2*b^9*c^2*d^10*f^2 - 112*B^3*a^2*b^9*c^4*d^8*f^2 + 224*B^3*a^3*b^8*c^3*d^9*f^2 - 40*B^3*a^4*b^7*c^2*d^10*f^2 - 24*B^3*a^4*b^7*c^4*d^8*f^2 + 200*B^3*a^5*b^6*c^3*d^9*f^2 - 40*B^3*a^6*b^5*c^2*d^10*f^2 + 80*B^3*a^6*b^5*c^4*d^8*f^2 - 128*B^3*a^7*b^4*c^3*d^9*f^2 + 28*B^3*a^8*b^3*c^2*d^10*f^2 - 20*B^3*a^8*b^3*c^4*d^8*f^2 + 20*B^3*a^9*b^2*c^3*d^9*f^2))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((8*(80*B*a*b^14*d^11*f^4 - 48*B*b^15*c*d^10*f^4 + 384*B*a^3*b^12*d^11*f^4 + 720*B*a^5*b^10*d^11*f^4 + 640*B*a^7*b^8*d^11*f^4 + 240*B*a^9*b^6*d^11*f^4 - 16*B*a^13*b^2*d^11*f^4 - 48*B*b^15*c^3*d^8*f^4 + 80*B*a*b^14*c^2*d^9*f^4 - 224*B*a^2*b^13*c*d^10*f^4 - 400*B*a^4*b^11*c*d^10*f^4 - 320*B*a^6*b^9*c*d^10*f^4 - 80*B*a^8*b^7*c*d^10*f^4 + 32*B*a^10*b^5*c*d^10*f^4 + 16*B*a^12*b^3*c*d^10*f^4 - 224*B*a^2*b^13*c^3*d^8*f^4 + 384*B*a^3*b^12*c^2*d^9*f^4 - 400*B*a^4*b^11*c^3*d^8*f^4 + 720*B*a^5*b^10*c^2*d^9*f^4 - 320*B*a^6*b^9*c^3*d^8*f^4 + 640*B*a^7*b^8*c^2*d^9*f^4 - 80*B*a^8*b^7*c^3*d^8*f^4 + 240*B*a^9*b^6*c^2*d^9*f^4 + 32*B*a^10*b^5*c^3*d^8*f^4 + 16*B*a^12*b^3*c^3*d^8*f^4 - 16*B*a^13*b^2*c^2*d^9*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (16*(c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*(32*b^17*d^10*f^4 + 160*a^2*b^15*d^10*f^4 + 288*a^4*b^13*d^10*f^4 + 160*a^6*b^11*d^10*f^4 - 160*a^8*b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b^5*d^10*f^4 - 32*a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^9*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14*b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336*a^5*b^12*c*d^9*f^4 + 560*a^7*b^10*c*d^9*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*a^11*b^6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2*c*d^9*f^4))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(44*B^2*a^9*b^4*d^11*f^2 - 168*B^2*a^5*b^8*d^11*f^2 - 40*B^2*a^7*b^6*d^11*f^2 - 20*B^2*a^3*b^10*d^11*f^2 - 4*B^2*a^11*b^2*d^11*f^2 - 36*B^2*b^13*c^3*d^8*f^2 + 60*B^2*a*b^12*d^11*f^2 - 12*B^2*b^13*c*d^10*f^2 + 4*B^2*a^12*b*c*d^10*f^2 + 100*B^2*a*b^12*c^2*d^9*f^2 + 120*B^2*a^2*b^11*c*d^10*f^2 + 156*B^2*a^4*b^9*c*d^10*f^2 - 112*B^2*a^6*b^7*c*d^10*f^2 - 148*B^2*a^8*b^5*c*d^10*f^2 - 8*B^2*a^10*b^3*c*d^10*f^2 + 68*B^2*a^2*b^11*c^3*d^8*f^2 + 124*B^2*a^3*b^10*c^2*d^9*f^2 + 184*B^2*a^4*b^9*c^3*d^8*f^2 + 8*B^2*a^5*b^8*c^2*d^9*f^2 + 40*B^2*a^6*b^7*c^3*d^8*f^2 + 24*B^2*a^7*b^6*c^2*d^9*f^2 - 20*B^2*a^8*b^5*c^3*d^8*f

^2 + 20*B^2*a^9*b^4*c^2*d^9*f^2 + 20*B^2*a^10*b^3*c^3*d^8*f^2 - 20*B^2*a^11
*b^2*c^2*d^9*f^2))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a
^6*b^2*f^4))*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 +
32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^
8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/
2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*
d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^
4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2))*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2
- 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^
4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4
 + 64*a^6*b^2*f^4))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^
3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4
 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) + (16*(c + d*tan(
e + f*x))^(1/2)*(2*B^4*b^9*d^12 - 5*B^4*a^2*b^7*d^12 + 17*B^4*a^4*b^5*d^12
- 7*B^4*a^6*b^3*d^12 + 6*B^4*b^9*c^4*d^8 + B^4*a^8*b*d^12 + 77*B^4*a^2*b^7*
c^2*d^10 - 8*B^4*a^2*b^7*c^4*d^8 + 60*B^4*a^3*b^6*c^3*d^9 - 87*B^4*a^4*b^5*
c^2*d^10 + 14*B^4*a^4*b^5*c^4*d^8 - 36*B^4*a^5*b^4*c^3*d^9 + 27*B^4*a^6*b^3
*c^2*d^10 - 4*B^4*a^6*b^3*c^4*d^8 + 4*B^4*a^7*b^2*c^3*d^9 + 12*B^4*a*b^8*c*
d^11 - 28*B^4*a*b^8*c^3*d^9 - 64*B^4*a^3*b^6*c*d^11 + 44*B^4*a^5*b^4*c*d^11
 - 8*B^4*a^7*b^2*c*d^11 - B^4*a^8*b*c^2*d^10))/(a^8*f^4 + b^8*f^4 + 4*a^2*b
^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c
*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4
- (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^
4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2
*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^
8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*1i)/((((8*(1
56*B^3*a^2*b^9*d^12*f^2 - 16*B^3*a^4*b^7*d^12*f^2 - 120*B^3*a^6*b^5*d^12*f^
2 + 48*B^3*a^8*b^3*d^12*f^2 + 12*B^3*b^11*c^2*d^10*f^2 + 12*B^3*b^11*c^4*d^
8*f^2 - 4*B^3*a^10*b*d^12*f^2 - 124*B^3*a*b^10*c*d^11*f^2 - 124*B^3*a*b^10*
c^3*d^9*f^2 + 224*B^3*a^3*b^8*c*d^11*f^2 + 200*B^3*a^5*b^6*c*d^11*f^2 - 128
*B^3*a^7*b^4*c*d^11*f^2 + 20*B^3*a^9*b^2*c*d^11*f^2 - 4*B^3*a^10*b*c^2*d^10
*f^2 + 44*B^3*a^2*b^9*c^2*d^10*f^2 - 112*B^3*a^2*b^9*c^4*d^8*f^2 + 224*B^3*
a^3*b^8*c^3*d^9*f^2 - 40*B^3*a^4*b^7*c^2*d^10*f^2 - 24*B^3*a^4*b^7*c^4*d^8*
f^2 + 200*B^3*a^5*b^6*c^3*d^9*f^2 - 40*B^3*a^6*b^5*c^2*d^10*f^2 + 80*B^3*a^
6*b^5*c^4*d^8*f^2 - 128*B^3*a^7*b^4*c^3*d^9*f^2 + 28*B^3*a^8*b^3*c^2*d^10*f
^2 - 20*B^3*a^8*b^3*c^4*d^8*f^2 + 20*B^3*a^9*b^2*c^3*d^9*f^2))/(a^8*f^5 + b
^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((8*(80*B*a*b^14
*d^11*f^4 - 48*B*b^15*c*d^10*f^4 + 384*B*a^3*b^12*d^11*f^4 + 720*B*a^5*b^10
*d^11*f^4 + 640*B*a^7*b^8*d^11*f^4 + 240*B*a^9*b^6*d^11*f^4 - 16*B*a^13*b^2
*d^11*f^4 - 48*B*b^15*c^3*d^8*f^4 + 80*B*a*b^14*c^2*d^9*f^4 - 224*B*a^2*b^1
3*c*d^10*f^4 - 400*B*a^4*b^11*c*d^10*f^4 - 320*B*a^6*b^9*c*d^10*f^4 - 80*B*
a^8*b^7*c*d^10*f^4 + 32*B*a^10*b^5*c*d^10*f^4 + 16*B*a^12*b^3*c*d^10*f^4 -
224*B*a^2*b^13*c^3*d^8*f^4 + 384*B*a^3*b^12*c^2*d^9*f^4 - 400*B*a^4*b^11*c^
3*d^8*f^4 + 720*B*a^5*b^10*c^2*d^9*f^4 - 320*B*a^6*b^9*c^3*d^8*f^4 + 640*B*
a^7*b^8*c^2*d^9*f^4 - 80*B*a^8*b^7*c^3*d^8*f^4 + 240*B*a^9*b^6*c^2*d^9*f^4

+ 32*B*a^10*b^5*c^3*d^8*f^4 + 16*B*a^12*b^3*c^3*d^8*f^4 - 16*B*a^13*b^2*c^2
*d^9*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f
^5) - (16*(c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2
 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B
^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^
4 + 64*a^6*b^2*f^4))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b
^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^
4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*(32*b^17*d^10*f^
4 + 160*a^2*b^15*d^10*f^4 + 288*a^4*b^13*d^10*f^4 + 160*a^6*b^11*d^10*f^4 -
 160*a^8*b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b^5*d^10*f^4 - 32*
a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*d^8*f^4 + 624*a^
4*b^13*c^2*d^8*f^4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^9*c^2*d^8*f^4 + 4
8*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14*b^3*c^2*d^8*f^4
+ 16*a*b^16*c*d^9*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336*a^5*b^12*c*d^9*f^4 + 5
60*a^7*b^10*c*d^9*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*a^11*b^6*c*d^9*f^4 + 11
2*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2*c*d^9*f^4))/(a^8*f^4 + b^8*f^4 + 4*a^2*b
^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c
*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4
- (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^
4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2
*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^
8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(c + d
*tan(e + f*x))^(1/2)*(44*B^2*a^9*b^4*d^11*f^2 - 168*B^2*a^5*b^8*d^11*f^2 -
40*B^2*a^7*b^6*d^11*f^2 - 20*B^2*a^3*b^10*d^11*f^2 - 4*B^2*a^11*b^2*d^11*f^
2 - 36*B^2*b^13*c^3*d^8*f^2 + 60*B^2*a*b^12*d^11*f^2 - 12*B^2*b^13*c*d^10*f
^2 + 4*B^2*a^12*b*c*d^10*f^2 + 100*B^2*a*b^12*c^2*d^9*f^2 + 120*B^2*a^2*b^1
1*c*d^10*f^2 + 156*B^2*a^4*b^9*c*d^10*f^2 - 112*B^2*a^6*b^7*c*d^10*f^2 - 14
8*B^2*a^8*b^5*c*d^10*f^2 - 8*B^2*a^10*b^3*c*d^10*f^2 + 68*B^2*a^2*b^11*c^3*
d^8*f^2 + 124*B^2*a^3*b^10*c^2*d^9*f^2 + 184*B^2*a^4*b^9*c^3*d^8*f^2 + 8*B^
2*a^5*b^8*c^2*d^9*f^2 + 40*B^2*a^6*b^7*c^3*d^8*f^2 + 24*B^2*a^7*b^6*c^2*d^9
*f^2 - 20*B^2*a^8*b^5*c^3*d^8*f^2 + 20*B^2*a^9*b^4*c^2*d^9*f^2 + 20*B^2*a^1
0*b^3*c^3*d^8*f^2 - 20*B^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 + b^8*f^4 + 4*a^
2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^
4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2
/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4
*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*
B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 +
 b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2))*(-(((8*B
^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 -
48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 +
64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*B^2*a^4*c*f^2
- 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^
2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2
f^4)))^(1/2) - (16(c + d*tan(e + f*x))^(1/2)*(2*B^4*b^9*d^12 - 5*B^4*a^2*
b^7*d^12 + 17*B^4*a^4*b^5*d^12 - 7*B^4*a^6*b^3*d^12 + 6*B^4*b^9*c^4*d^8 + B

^4*a^8*b*d^12 + 77*B^4*a^2*b^7*c^2*d^10 - 8*B^4*a^2*b^7*c^4*d^8 + 60*B^4*a^3*b^6*c^3*d^9 - 87*B^4*a^4*b^5*c^2*d^10 + 14*B^4*a^4*b^5*c^4*d^8 - 36*B^4*a^5*b^4*c^3*d^9 + 27*B^4*a^6*b^3*c^2*d^10 - 4*B^4*a^6*b^3*c^4*d^8 + 4*B^4*a^7*b^2*c^3*d^9 + 12*B^4*a*b^8*c*d^11 - 28*B^4*a*b^8*c^3*d^9 - 64*B^4*a^3*b^6*c*d^11 + 44*B^4*a^5*b^4*c*d^11 - 8*B^4*a^7*b^2*c*d^11 - B^4*a^8*b*c^2*d^10))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) + (((8*(156*B^3*a^2*b^9*d^12*f^2 - 16*B^3*a^4*b^7*d^12*f^2 - 120*B^3*a^6*b^5*d^12*f^2 + 48*B^3*a^8*b^3*d^12*f^2 + 12*B^3*b^11*c^2*d^10*f^2 + 12*B^3*b^11*c^4*d^8*f^2 - 4*B^3*a^10*b*d^12*f^2 - 124*B^3*a*b^10*c*d^11*f^2 - 124*B^3*a*b^10*c^3*d^9*f^2 + 224*B^3*a^3*b^8*c*d^11*f^2 + 200*B^3*a^5*b^6*c*d^11*f^2 - 128*B^3*a^7*b^4*c*d^11*f^2 + 20*B^3*a^9*b^2*c*d^11*f^2 - 4*B^3*a^10*b*c^2*d^10*f^2 + 44*B^3*a^2*b^9*c^2*d^10*f^2 - 112*B^3*a^2*b^9*c^4*d^8*f^2 + 224*B^3*a^3*b^8*c^3*d^9*f^2 - 40*B^3*a^4*b^7*c^2*d^10*f^2 - 24*B^3*a^4*b^7*c^4*d^8*f^2 + 200*B^3*a^5*b^6*c^3*d^9*f^2 - 40*B^3*a^6*b^5*c^2*d^10*f^2 + 80*B^3*a^6*b^5*c^4*d^8*f^2 - 128*B^3*a^7*b^4*c^3*d^9*f^2 + 28*B^3*a^8*b^3*c^2*d^10*f^2 - 20*B^3*a^8*b^3*c^4*d^8*f^2 + 20*B^3*a^9*b^2*c^3*d^9*f^2))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((8*(80*B*a*b^14*d^11*f^4 - 48*B*b^15*c*d^10*f^4 + 384*B*a^3*b^12*d^11*f^4 + 720*B*a^5*b^10*d^11*f^4 + 640*B*a^7*b^8*d^11*f^4 + 240*B*a^9*b^6*d^11*f^4 - 16*B*a^13*b^2*d^11*f^4 - 48*B*b^15*c^3*d^8*f^4 + 80*B*a*b^14*c^2*d^9*f^4 - 224*B*a^2*b^13*c*d^10*f^4 - 400*B*a^4*b^11*c*d^10*f^4 - 320*B*a^6*b^9*c*d^10*f^4 - 80*B*a^8*b^7*c*d^10*f^4 + 32*B*a^10*b^5*c*d^10*f^4 + 16*B*a^12*b^3*c*d^10*f^4 - 224*B*a^2*b^13*c^3*d^8*f^4 + 384*B*a^3*b^12*c^2*d^9*f^4 - 400*B*a^4*b^11*c^3*d^8*f^4 + 720*B*a^5*b^10*c^2*d^9*f^4 - 320*B*a^6*b^9*c^3*d^8*f^4 + 640*B*a^7*b^8*c^2*d^9*f^4 - 80*B*a^8*b^7*c^3*d^8*f^4 + 240*B*a^9*b^6*c^2*d^9*f^4 + 32*B*a^10*b^5*c^3*d^8*f^4 + 16*B*a^12*b^3*c^3*d^8*f^4 - 16*B*a^13*b^2*c^2*d^9*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (16*(c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*(32*b^17*d^10*f^4 + 160*a^2*b^15*d^10*f^4 + 288*a^4*b^13*d^10*f^4 + 160*a^6*b^11*d^10*f^4 - 160*a^8*b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b^5*d^10*f^4 - 32*a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^9*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14*b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336*a^5*b^12*c*d^9*f^4 + 560*a^7*b^10*c*d^9*f^4 + 560*a^9*b^8*c*d^9*f^4 +

336*a^11*b^6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2*c*d^9*f^4))/
(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8
*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2
- 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4
+ 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*B^2*a^4*c*f^
2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*
b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b
^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(44*B^2*a^9*b^4*d^11*f^2 -
168*B^2*a^5*b^8*d^11*f^2 - 40*B^2*a^7*b^6*d^11*f^2 - 20*B^2*a^3*b^10*d^11*
f^2 - 4*B^2*a^11*b^2*d^11*f^2 - 36*B^2*b^13*c^3*d^8*f^2 + 60*B^2*a*b^12*d^1
1*f^2 - 12*B^2*b^13*c*d^10*f^2 + 4*B^2*a^12*b*c*d^10*f^2 + 100*B^2*a*b^12*c
^2*d^9*f^2 + 120*B^2*a^2*b^11*c*d^10*f^2 + 156*B^2*a^4*b^9*c*d^10*f^2 - 112
*B^2*a^6*b^7*c*d^10*f^2 - 148*B^2*a^8*b^5*c*d^10*f^2 - 8*B^2*a^10*b^3*c*d^1
0*f^2 + 68*B^2*a^2*b^11*c^3*d^8*f^2 + 124*B^2*a^3*b^10*c^2*d^9*f^2 + 184*B^
2*a^4*b^9*c^3*d^8*f^2 + 8*B^2*a^5*b^8*c^2*d^9*f^2 + 40*B^2*a^6*b^7*c^3*d^8*
f^2 + 24*B^2*a^7*b^6*c^2*d^9*f^2 - 20*B^2*a^8*b^5*c^3*d^8*f^2 + 20*B^2*a^9*
b^4*c^2*d^9*f^2 + 20*B^2*a^10*b^3*c^3*d^8*f^2 - 20*B^2*a^11*b^2*c^2*d^9*f^2
))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(
((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f
^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f
^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*B^2*a^4*c
*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a
^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^
6*b^2*f^4)))^(1/2))*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d
*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)
*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^
4))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2
*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4
+ 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*
(2*B^4*b^9*d^12 - 5*B^4*a^2*b^7*d^12 + 17*B^4*a^4*b^5*d^12 - 7*B^4*a^6*b^3*
d^12 + 6*B^4*b^9*c^4*d^8 + B^4*a^8*b*d^12 + 77*B^4*a^2*b^7*c^2*d^10 - 8*B^4
*a^2*b^7*c^4*d^8 + 60*B^4*a^3*b^6*c^3*d^9 - 87*B^4*a^4*b^5*c^2*d^10 + 14*B^
4*a^4*b^5*c^4*d^8 - 36*B^4*a^5*b^4*c^3*d^9 + 27*B^4*a^6*b^3*c^2*d^10 - 4*B^
4*a^6*b^3*c^4*d^8 + 4*B^4*a^7*b^2*c^3*d^9 + 12*B^4*a*b^8*c*d^11 - 28*B^4*a*
b^8*c^3*d^9 - 64*B^4*a^3*b^6*c*d^11 + 44*B^4*a^5*b^4*c*d^11 - 8*B^4*a^7*b^2
*c*d^11 - B^4*a^8*b*c^2*d^10))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b
^4*f^4 + 4*a^6*b^2*f^4))*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*
b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4
d^2)(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b
^2*f^4))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 1
6*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^
6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(2*B^5*a^3*b^4*d^13 +
4*B^5*b^7*c^3*d^10 - 6*B^5*a*b^6*d^13 + 4*B^5*b^7*c*d^12 - 9*B^5*a^2*b^5*c^
3*d^10 + 4*B^5*a^2*b^5*c^5*d^8 - 12*B^5*a^3*b^4*c^2*d^11 - 14*B^5*a^3*b^4*c
^4*d^9 + 2*B^5*a^4*b^3*c^3*d^10 - 4*B^5*a^4*b^3*c^5*d^8 + 4*B^5*a^5*b^2*c^2

*d^11 + 4*B^5*a^5*b^2*c^4*d^9 - B^5*a^6*b*c*d^12 + 6*B^5*a*b^6*c^4*d^9 - 13
*B^5*a^2*b^5*c*d^12 + 6*B^5*a^4*b^3*c*d^12 - B^5*a^6*b*c^3*d^10))/(a^8*f^5
+ b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5)))*(-(((8*B^2*a^4
*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2
*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2
*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^
2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^
2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))
)^(1/2)*2i - atan(((((8*(304*C^3*a^3*b^9*d^12*f^2 + 120*C^3*a^5*b^7*d^12*f^
2 - 320*C^3*a^7*b^5*d^12*f^2 - 148*C^3*a^9*b^3*d^12*f^2 + 4*C^3*b^12*c^3*d^
9*f^2 - 4*C^3*a*b^11*d^12*f^2 - 16*C^3*a^11*b*d^12*f^2 + 4*C^3*b^12*c*d^11*
f^2 + 60*C^3*a*b^11*c^2*d^10*f^2 + 64*C^3*a*b^11*c^4*d^8*f^2 - 320*C^3*a^2*
b^10*c*d^11*f^2 + 104*C^3*a^4*b^8*c*d^11*f^2 + 544*C^3*a^6*b^6*c*d^11*f^2 +
116*C^3*a^8*b^4*c*d^11*f^2 - 16*C^3*a^11*b*c^2*d^10*f^2 - 320*C^3*a^2*b^10
*c^3*d^9*f^2 + 176*C^3*a^3*b^9*c^2*d^10*f^2 - 128*C^3*a^3*b^9*c^4*d^8*f^2 +
104*C^3*a^4*b^8*c^3*d^9*f^2 - 72*C^3*a^5*b^7*c^2*d^10*f^2 - 192*C^3*a^5*b^
7*c^4*d^8*f^2 + 544*C^3*a^6*b^6*c^3*d^9*f^2 - 320*C^3*a^7*b^5*c^2*d^10*f^2
+ 116*C^3*a^8*b^4*c^3*d^9*f^2 - 148*C^3*a^9*b^3*c^2*d^10*f^2))/(b^9*f^5 + a
^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^5) - (((8*(96*C*a^2*
b^14*d^11*f^4 + 480*C*a^4*b^12*d^11*f^4 + 960*C*a^6*b^10*d^11*f^4 + 960*C*a
^8*b^8*d^11*f^4 + 480*C*a^10*b^6*d^11*f^4 + 96*C*a^12*b^4*d^11*f^4 - 64*C*a
*b^15*c^3*d^8*f^4 - 320*C*a^3*b^13*c*d^10*f^4 - 640*C*a^5*b^11*c*d^10*f^4 -
640*C*a^7*b^9*c*d^10*f^4 - 320*C*a^9*b^7*c*d^10*f^4 - 64*C*a^11*b^5*c*d^10
*f^4 + 96*C*a^2*b^14*c^2*d^9*f^4 - 320*C*a^3*b^13*c^3*d^8*f^4 + 480*C*a^4*b
^12*c^2*d^9*f^4 - 640*C*a^5*b^11*c^3*d^8*f^4 + 960*C*a^6*b^10*c^2*d^9*f^4 -
640*C*a^7*b^9*c^3*d^8*f^4 + 960*C*a^8*b^8*c^2*d^9*f^4 - 320*C*a^9*b^7*c^3*
d^8*f^4 + 480*C*a^10*b^6*c^2*d^9*f^4 - 64*C*a^11*b^5*c^3*d^8*f^4 + 96*C*a^1
2*b^4*c^2*d^9*f^4 - 64*C*a*b^15*c*d^10*f^4))/(b^9*f^5 + a^8*b*f^5 + 4*a^2*b
^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^5) - (16*(c + d*tan(e + f*x))^(1/2)*(-
(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*
f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*
f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*C^2*a^4*
c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*
a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a
^6*b^2*f^4)))^(1/2)*(32*b^18*d^10*f^4 + 160*a^2*b^16*d^10*f^4 + 288*a^4*b^1
4*d^10*f^4 + 160*a^6*b^12*d^10*f^4 - 160*a^8*b^10*d^10*f^4 - 288*a^10*b^8*d
^10*f^4 - 160*a^12*b^6*d^10*f^4 - 32*a^14*b^4*d^10*f^4 + 48*b^18*c^2*d^8*f^
4 + 272*a^2*b^16*c^2*d^8*f^4 + 624*a^4*b^14*c^2*d^8*f^4 + 720*a^6*b^12*c^2*
d^8*f^4 + 400*a^8*b^10*c^2*d^8*f^4 + 48*a^10*b^8*c^2*d^8*f^4 - 48*a^12*b^6*
c^2*d^8*f^4 - 16*a^14*b^4*c^2*d^8*f^4 + 16*a*b^17*c*d^9*f^4 + 112*a^3*b^15*
c*d^9*f^4 + 336*a^5*b^13*c*d^9*f^4 + 560*a^7*b^11*c*d^9*f^4 + 560*a^9*b^9*c
*d^9*f^4 + 336*a^11*b^7*c*d^9*f^4 + 112*a^13*b^5*c*d^9*f^4 + 16*a^15*b^3*c*
d^9*f^4))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*
f^4))*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*
a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 +

16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*
C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 -
24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f
^4 + 4*a^6*b^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(52*C^2*a^3*b^
11*d^11*f^2 + 128*C^2*a^5*b^9*d^11*f^2 + 424*C^2*a^7*b^7*d^11*f^2 + 380*C^2
*a^9*b^5*d^11*f^2 + 100*C^2*a^11*b^3*d^11*f^2 - 20*C^2*b^14*c^3*d^8*f^2 + 6
0*C^2*a*b^13*d^11*f^2 + 8*C^2*a^13*b*d^11*f^2 - 4*C^2*a^14*c*d^10*f^2 - 12*
C^2*b^14*c*d^10*f^2 + 84*C^2*a*b^13*c^2*d^9*f^2 + 60*C^2*a^2*b^12*c*d^10*f^
2 - 116*C^2*a^4*b^10*c*d^10*f^2 - 604*C^2*a^6*b^8*c*d^10*f^2 - 596*C^2*a^8*
b^6*c*d^10*f^2 - 220*C^2*a^10*b^4*c*d^10*f^2 - 44*C^2*a^12*b^2*c*d^10*f^2 +
116*C^2*a^2*b^12*c^3*d^8*f^2 + 108*C^2*a^3*b^11*c^2*d^9*f^2 + 216*C^2*a^4*
b^10*c^3*d^8*f^2 + 104*C^2*a^5*b^9*c^2*d^9*f^2 + 8*C^2*a^6*b^8*c^3*d^8*f^2
+ 248*C^2*a^7*b^7*c^2*d^9*f^2 - 68*C^2*a^8*b^6*c^3*d^8*f^2 + 196*C^2*a^9*b^
5*c^2*d^9*f^2 + 4*C^2*a^10*b^4*c^3*d^8*f^2 + 28*C^2*a^11*b^3*c^2*d^9*f^2))/
(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*(-((
(8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^
2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^
4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*C^2*a^4*c*
f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^
2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6
*b^2*f^4)))^(1/2))*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*
f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*
(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4
))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*
a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4
+ 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(
2*C^4*b^10*d^12 - C^4*a^10*d^12 + 4*C^4*a^2*b^8*d^12 + 27*C^4*a^4*b^6*d^12
- 15*C^4*a^6*b^4*d^12 - 9*C^4*a^8*b^2*d^12 + C^4*a^10*c^2*d^10 + 4*C^4*b^10
*c^2*d^10 + 2*C^4*b^10*c^4*d^8 + 24*C^4*a^2*b^8*c^2*d^10 - 12*C^4*a^2*b^8*c
^4*d^8 + 104*C^4*a^3*b^7*c^3*d^9 - 197*C^4*a^4*b^6*c^2*d^10 + 18*C^4*a^4*b^
6*c^4*d^8 - 32*C^4*a^5*b^5*c^3*d^9 - 17*C^4*a^6*b^4*c^2*d^10 - 8*C^4*a^7*b^
3*c^3*d^9 + 9*C^4*a^8*b^2*c^2*d^10 + 4*C^4*a^9*b*c*d^11 - 40*C^4*a^3*b^7*c*
d^11 + 132*C^4*a^5*b^5*c*d^11 + 48*C^4*a^7*b^3*c*d^11))/(b^9*f^4 + a^8*b*f^
4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*(-(((8*C^2*a^4*c*f^2 +
8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*
c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4
+ 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f
^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(a
^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*1
i - (((8*(304*C^3*a^3*b^9*d^12*f^2 + 120*C^3*a^5*b^7*d^12*f^2 - 320*C^3*a^7
*b^5*d^12*f^2 - 148*C^3*a^9*b^3*d^12*f^2 + 4*C^3*b^12*c^3*d^9*f^2 - 4*C^3*a
*b^11*d^12*f^2 - 16*C^3*a^11*b*d^12*f^2 + 4*C^3*b^12*c*d^11*f^2 + 60*C^3*a*
b^11*c^2*d^10*f^2 + 64*C^3*a*b^11*c^4*d^8*f^2 - 320*C^3*a^2*b^10*c*d^11*f^2
+ 104*C^3*a^4*b^8*c*d^11*f^2 + 544*C^3*a^6*b^6*c*d^11*f^2 + 116*C^3*a^8*b^
4*c*d^11*f^2 - 16*C^3*a^11*b*c^2*d^10*f^2 - 320*C^3*a^2*b^10*c^3*d^9*f^2 +

176*C^3*a^3*b^9*c^2*d^10*f^2 - 128*C^3*a^3*b^9*c^4*d^8*f^2 + 104*C^3*a^4*b^8*c^3*d^9*f^2 - 72*C^3*a^5*b^7*c^2*d^10*f^2 - 192*C^3*a^5*b^7*c^4*d^8*f^2 + 544*C^3*a^6*b^6*c^3*d^9*f^2 - 320*C^3*a^7*b^5*c^2*d^10*f^2 + 116*C^3*a^8*b^4*c^3*d^9*f^2 - 148*C^3*a^9*b^3*c^2*d^10*f^2))/(b^9*f^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^5) - (((8*(96*C*a^2*b^14*d^11*f^4 + 480*C*a^4*b^12*d^11*f^4 + 960*C*a^6*b^10*d^11*f^4 + 960*C*a^8*b^8*d^11*f^4 + 480*C*a^10*b^6*d^11*f^4 + 96*C*a^12*b^4*d^11*f^4 - 64*C*a*b^15*c^3*d^8*f^4 - 320*C*a^3*b^13*c*d^10*f^4 - 640*C*a^5*b^11*c*d^10*f^4 - 640*C*a^7*b^9*c*d^10*f^4 - 320*C*a^9*b^7*c*d^10*f^4 - 64*C*a^11*b^5*c*d^10*f^4 + 96*C*a^2*b^14*c^2*d^9*f^4 - 320*C*a^3*b^13*c^3*d^8*f^4 + 480*C*a^4*b^12*c^2*d^9*f^4 - 640*C*a^5*b^11*c^3*d^8*f^4 + 960*C*a^6*b^10*c^2*d^9*f^4 - 640*C*a^7*b^9*c^3*d^8*f^4 + 960*C*a^8*b^8*c^2*d^9*f^4 - 320*C*a^9*b^7*c^3*d^8*f^4 + 480*C*a^10*b^6*c^2*d^9*f^4 - 64*C*a^11*b^5*c^3*d^8*f^4 + 96*C*a^12*b^4*c^2*d^9*f^4 - 64*C*a*b^15*c*d^10*f^4))/(b^9*f^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^5) + (16*(c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*(32*b^18*d^10*f^4 + 160*a^2*b^16*d^10*f^4 + 288*a^4*b^14*d^10*f^4 + 160*a^6*b^12*d^10*f^4 - 160*a^8*b^10*d^10*f^4 - 288*a^10*b^8*d^10*f^4 - 160*a^12*b^6*d^10*f^4 - 32*a^14*b^4*d^10*f^4 + 48*b^18*c^2*d^8*f^4 + 272*a^2*b^16*c^2*d^8*f^4 + 624*a^4*b^14*c^2*d^8*f^4 + 720*a^6*b^12*c^2*d^8*f^4 + 400*a^8*b^10*c^2*d^8*f^4 + 48*a^10*b^8*c^2*d^8*f^4 - 48*a^12*b^6*c^2*d^8*f^4 - 16*a^14*b^4*c^2*d^8*f^4 + 16*a*b^17*c*d^9*f^4 + 112*a^3*b^15*c*d^9*f^4 + 336*a^5*b^13*c*d^9*f^4 + 560*a^7*b^11*c*d^9*f^4 + 560*a^9*b^9*c*d^9*f^4 + 336*a^11*b^7*c*d^9*f^4 + 112*a^13*b^5*c*d^9*f^4 + 16*a^15*b^3*c*d^9*f^4))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(52*C^2*a^3*b^11*d^11*f^2 + 128*C^2*a^5*b^9*d^11*f^2 + 424*C^2*a^7*b^7*d^11*f^2 + 380*C^2*a^9*b^5*d^11*f^2 + 100*C^2*a^11*b^3*d^11*f^2 - 20*C^2*b^14*c^3*d^8*f^2 + 60*C^2*a*b^13*d^11*f^2 + 8*C^2*a^13*b*d^11*f^2 - 4*C^2*a^14*c*d^10*f^2 - 12*C^2*b^14*c*d^10*f^2 + 84*C^2*a*b^13*c^2*d^9*f^2 + 60*C^2*a^2*b^12*c*d^10*f^2 - 116*C^2*a^4*b^10*c*d^10*f^2 - 604*C^2*a^6*b^8*c*d^10*f^2 - 596*C^2*a^8*b^6*c*d^10*f^2 - 220*C^2*a^10*b^4*c*d^10*f^2 - 44*C^2*a^12*b^2*c*d^10*f^2 + 116*C^2*a^2*b^12*c^3*d^8*f^2 + 108*C^2*a^3*b^11*c^2*d^9*f^2 + 216*C^2*a^4*b^10*c^3*d^8*f^2 + 104*C^2*a^5*b^9*c^2*d^9*f^2 + 8*C^2*a^6*b^8*c^3*d^8*f^2 + 248*C^2*a^7*b^7*c^2*d^9*f^2 - 68*C^2*a^8*b^6*c^3*d^8*f^2 + 196*C^2*a^9*b^5*c^2*d^9*f^2 + 4*C^2*a^10*b^4*c^3*d^8*f^2 + 28*C^2*a^11*b^3*c^2*d^9*f^2))/(b^9*f^4 + a^8*

b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*(-(((8*C^2*a^4*c*f^
2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*
b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*
f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4
*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(1
6*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/
2))*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^
3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 1
6*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*C^
2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 2
4*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4
 + 4*a^6*b^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(2*C^4*b^10*d^12
 - C^4*a^10*d^12 + 4*C^4*a^2*b^8*d^12 + 27*C^4*a^4*b^6*d^12 - 15*C^4*a^6*b^
4*d^12 - 9*C^4*a^8*b^2*d^12 + C^4*a^10*c^2*d^10 + 4*C^4*b^10*c^2*d^10 + 2*C
^4*b^10*c^4*d^8 + 24*C^4*a^2*b^8*c^2*d^10 - 12*C^4*a^2*b^8*c^4*d^8 + 104*C^
4*a^3*b^7*c^3*d^9 - 197*C^4*a^4*b^6*c^2*d^10 + 18*C^4*a^4*b^6*c^4*d^8 - 32*
C^4*a^5*b^5*c^3*d^9 - 17*C^4*a^6*b^4*c^2*d^10 - 8*C^4*a^7*b^3*c^3*d^9 + 9*C
^4*a^8*b^2*c^2*d^10 + 4*C^4*a^9*b*c*d^11 - 40*C^4*a^3*b^7*c*d^11 + 132*C^4*
a^5*b^5*c*d^11 + 48*C^4*a^7*b^3*c*d^11))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f
^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2
 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C
^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^
4 + 64*a^6*b^2*f^4))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b
^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^
4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*1i)/((((8*(304*C
^3*a^3*b^9*d^12*f^2 + 120*C^3*a^5*b^7*d^12*f^2 - 320*C^3*a^7*b^5*d^12*f^2 -
 148*C^3*a^9*b^3*d^12*f^2 + 4*C^3*b^12*c^3*d^9*f^2 - 4*C^3*a*b^11*d^12*f^2
- 16*C^3*a^11*b*d^12*f^2 + 4*C^3*b^12*c*d^11*f^2 + 60*C^3*a*b^11*c^2*d^10*f
^2 + 64*C^3*a*b^11*c^4*d^8*f^2 - 320*C^3*a^2*b^10*c*d^11*f^2 + 104*C^3*a^4*
b^8*c*d^11*f^2 + 544*C^3*a^6*b^6*c*d^11*f^2 + 116*C^3*a^8*b^4*c*d^11*f^2 -
16*C^3*a^11*b*c^2*d^10*f^2 - 320*C^3*a^2*b^10*c^3*d^9*f^2 + 176*C^3*a^3*b^9
*c^2*d^10*f^2 - 128*C^3*a^3*b^9*c^4*d^8*f^2 + 104*C^3*a^4*b^8*c^3*d^9*f^2 -
 72*C^3*a^5*b^7*c^2*d^10*f^2 - 192*C^3*a^5*b^7*c^4*d^8*f^2 + 544*C^3*a^6*b^
6*c^3*d^9*f^2 - 320*C^3*a^7*b^5*c^2*d^10*f^2 + 116*C^3*a^8*b^4*c^3*d^9*f^2
- 148*C^3*a^9*b^3*c^2*d^10*f^2))/(b^9*f^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a
^4*b^5*f^5 + 4*a^6*b^3*f^5) - (((8*(96*C*a^2*b^14*d^11*f^4 + 480*C*a^4*b^12
*d^11*f^4 + 960*C*a^6*b^10*d^11*f^4 + 960*C*a^8*b^8*d^11*f^4 + 480*C*a^10*b
^6*d^11*f^4 + 96*C*a^12*b^4*d^11*f^4 - 64*C*a*b^15*c^3*d^8*f^4 - 320*C*a^3*
b^13*c*d^10*f^4 - 640*C*a^5*b^11*c*d^10*f^4 - 640*C*a^7*b^9*c*d^10*f^4 - 32
0*C*a^9*b^7*c*d^10*f^4 - 64*C*a^11*b^5*c*d^10*f^4 + 96*C*a^2*b^14*c^2*d^9*f
^4 - 320*C*a^3*b^13*c^3*d^8*f^4 + 480*C*a^4*b^12*c^2*d^9*f^4 - 640*C*a^5*b^
11*c^3*d^8*f^4 + 960*C*a^6*b^10*c^2*d^9*f^4 - 640*C*a^7*b^9*c^3*d^8*f^4 + 9
60*C*a^8*b^8*c^2*d^9*f^4 - 320*C*a^9*b^7*c^3*d^8*f^4 + 480*C*a^10*b^6*c^2*d
^9*f^4 - 64*C*a^11*b^5*c^3*d^8*f^4 + 96*C*a^12*b^4*c^2*d^9*f^4 - 64*C*a*b^1
5*c*d^10*f^4))/(b^9*f^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6

*b^3*f^5) - (16*(c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4
*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/
4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*
b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C
^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 +
b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*(32*b^18*d
^10*f^4 + 160*a^2*b^16*d^10*f^4 + 288*a^4*b^14*d^10*f^4 + 160*a^6*b^12*d^10
*f^4 - 160*a^8*b^10*d^10*f^4 - 288*a^10*b^8*d^10*f^4 - 160*a^12*b^6*d^10*f^
4 - 32*a^14*b^4*d^10*f^4 + 48*b^18*c^2*d^8*f^4 + 272*a^2*b^16*c^2*d^8*f^4 +
624*a^4*b^14*c^2*d^8*f^4 + 720*a^6*b^12*c^2*d^8*f^4 + 400*a^8*b^10*c^2*d^8
*f^4 + 48*a^10*b^8*c^2*d^8*f^4 - 48*a^12*b^6*c^2*d^8*f^4 - 16*a^14*b^4*c^2*
d^8*f^4 + 16*a*b^17*c*d^9*f^4 + 112*a^3*b^15*c*d^9*f^4 + 336*a^5*b^13*c*d^9
*f^4 + 560*a^7*b^11*c*d^9*f^4 + 560*a^9*b^9*c*d^9*f^4 + 336*a^11*b^7*c*d^9*
f^4 + 112*a^13*b^5*c*d^9*f^4 + 16*a^15*b^3*c*d^9*f^4))/(b^9*f^4 + a^8*b*f^4
+ 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*(-(((8*C^2*a^4*c*f^2 + 8
*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c
*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 +
96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^
2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(a^
8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) +
(16*(c + d*tan(e + f*x))^(1/2)*(52*C^2*a^3*b^11*d^11*f^2 + 128*C^2*a^5*b^9*
d^11*f^2 + 424*C^2*a^7*b^7*d^11*f^2 + 380*C^2*a^9*b^5*d^11*f^2 + 100*C^2*a^
11*b^3*d^11*f^2 - 20*C^2*b^14*c^3*d^8*f^2 + 60*C^2*a*b^13*d^11*f^2 + 8*C^2*
a^13*b*d^11*f^2 - 4*C^2*a^14*c*d^10*f^2 - 12*C^2*b^14*c*d^10*f^2 + 84*C^2*a
*b^13*c^2*d^9*f^2 + 60*C^2*a^2*b^12*c*d^10*f^2 - 116*C^2*a^4*b^10*c*d^10*f^
2 - 604*C^2*a^6*b^8*c*d^10*f^2 - 596*C^2*a^8*b^6*c*d^10*f^2 - 220*C^2*a^10*
b^4*c*d^10*f^2 - 44*C^2*a^12*b^2*c*d^10*f^2 + 116*C^2*a^2*b^12*c^3*d^8*f^2
+ 108*C^2*a^3*b^11*c^2*d^9*f^2 + 216*C^2*a^4*b^10*c^3*d^8*f^2 + 104*C^2*a^5
*b^9*c^2*d^9*f^2 + 8*C^2*a^6*b^8*c^3*d^8*f^2 + 248*C^2*a^7*b^7*c^2*d^9*f^2
- 68*C^2*a^8*b^6*c^3*d^8*f^2 + 196*C^2*a^9*b^5*c^2*d^9*f^2 + 4*C^2*a^10*b^4
*c^3*d^8*f^2 + 28*C^2*a^11*b^3*c^2*d^9*f^2))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b
^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c
*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4
- (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^
4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2
*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^
8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2))*(-(((8*C^2*
a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*
C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*
a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*C^2*a^4*c*f^2 + 4
*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c
f^2)/(16(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^
4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(2*C^4*b^10*d^12 - C^4*a^10*d^1
2 + 4*C^4*a^2*b^8*d^12 + 27*C^4*a^4*b^6*d^12 - 15*C^4*a^6*b^4*d^12 - 9*C^4*
a^8*b^2*d^12 + C^4*a^10*c^2*d^10 + 4*C^4*b^10*c^2*d^10 + 2*C^4*b^10*c^4*d^8

+ 24*C^4*a^2*b^8*c^2*d^10 - 12*C^4*a^2*b^8*c^4*d^8 + 104*C^4*a^3*b^7*c^3*d^9 - 197*C^4*a^4*b^6*c^2*d^10 + 18*C^4*a^4*b^6*c^4*d^8 - 32*C^4*a^5*b^5*c^3*d^9 - 17*C^4*a^6*b^4*c^2*d^10 - 8*C^4*a^7*b^3*c^3*d^9 + 9*C^4*a^8*b^2*c^2*d^10 + 4*C^4*a^9*b*c*d^11 - 40*C^4*a^3*b^7*c*d^11 + 132*C^4*a^5*b^5*c*d^11 + 48*C^4*a^7*b^3*c*d^11))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) + (((8*(304*C^3*a^3*b^9*d^12*f^2 + 120*C^3*a^5*b^7*d^12*f^2 - 320*C^3*a^7*b^5*d^12*f^2 - 148*C^3*a^9*b^3*d^12*f^2 + 4*C^3*b^12*c^3*d^9*f^2 - 4*C^3*a*b^11*d^12*f^2 - 16*C^3*a^11*b*d^12*f^2 + 4*C^3*b^12*c*d^11*f^2 + 60*C^3*a*b^11*c^2*d^10*f^2 + 64*C^3*a*b^11*c^4*d^8*f^2 - 320*C^3*a^2*b^10*c*d^11*f^2 + 104*C^3*a^4*b^8*c*d^11*f^2 + 544*C^3*a^6*b^6*c*d^11*f^2 + 116*C^3*a^8*b^4*c*d^11*f^2 - 16*C^3*a^11*b*c^2*d^10*f^2 - 320*C^3*a^2*b^10*c^3*d^9*f^2 + 176*C^3*a^3*b^9*c^2*d^10*f^2 - 128*C^3*a^3*b^9*c^4*d^8*f^2 + 104*C^3*a^4*b^8*c^3*d^9*f^2 - 72*C^3*a^5*b^7*c^2*d^10*f^2 - 192*C^3*a^5*b^7*c^4*d^8*f^2 + 544*C^3*a^6*b^6*c^3*d^9*f^2 - 320*C^3*a^7*b^5*c^2*d^10*f^2 + 116*C^3*a^8*b^4*c^3*d^9*f^2 - 148*C^3*a^9*b^3*c^2*d^10*f^2))/(b^9*f^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^5) - (((8*(96*C*a^2*b^14*d^11*f^4 + 480*C*a^4*b^12*d^11*f^4 + 960*C*a^6*b^10*d^11*f^4 + 960*C*a^8*b^8*d^11*f^4 + 480*C*a^10*b^6*d^11*f^4 + 96*C*a^12*b^4*d^11*f^4 - 64*C*a*b^15*c^3*d^8*f^4 - 320*C*a^3*b^13*c*d^10*f^4 - 640*C*a^5*b^11*c*d^10*f^4 - 640*C*a^7*b^9*c*d^10*f^4 - 320*C*a^9*b^7*c*d^10*f^4 - 64*C*a^11*b^5*c*d^10*f^4 + 96*C*a^2*b^14*c^2*d^9*f^4 - 320*C*a^3*b^13*c^3*d^8*f^4 + 480*C*a^4*b^12*c^2*d^9*f^4 - 640*C*a^5*b^11*c^3*d^8*f^4 + 960*C*a^6*b^10*c^2*d^9*f^4 - 640*C*a^7*b^9*c^3*d^8*f^4 + 960*C*a^8*b^8*c^2*d^9*f^4 - 320*C*a^9*b^7*c^3*d^8*f^4 + 480*C*a^10*b^6*c^2*d^9*f^4 - 64*C*a^11*b^5*c^3*d^8*f^4 + 96*C*a^12*b^4*c^2*d^9*f^4 - 64*C*a*b^15*c*d^10*f^4))/(b^9*f^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^5) + (16*(c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*(32*b^18*d^10*f^4 + 160*a^2*b^16*d^10*f^4 + 288*a^4*b^14*d^10*f^4 + 160*a^6*b^12*d^10*f^4 - 160*a^8*b^10*d^10*f^4 - 288*a^10*b^8*d^10*f^4 - 160*a^12*b^6*d^10*f^4 - 32*a^14*b^4*d^10*f^4 + 48*b^18*c^2*d^8*f^4 + 272*a^2*b^16*c^2*d^8*f^4 + 624*a^4*b^14*c^2*d^8*f^4 + 720*a^6*b^12*c^2*d^8*f^4 + 400*a^8*b^10*c^2*d^8*f^4 + 48*a^10*b^8*c^2*d^8*f^4 - 48*a^12*b^6*c^2*d^8*f^4 - 16*a^14*b^4*c^2*d^8*f^4 + 16*a*b^17*c*d^9*f^4 + 112*a^3*b^15*c*d^9*f^4 + 336*a^5*b^13*c*d^9*f^4 + 560*a^7*b^11*c*d^9*f^4 + 560*a^9*b^9*c*d^9*f^4 + 336*a^11*b^7*c*d^9*f^4 + 112*a^13*b^5*c*d^9*f^4 + 16*a^15*b^3*c*d^9*f^4))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 +

6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 3
2*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c
^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 +
64*a^6*b^2*f^4))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d
*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 +
4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(c + d*tan(e +
f*x))^(1/2)*(52*C^2*a^3*b^11*d^11*f^2 + 128*C^2*a^5*b^9*d^11*f^2 + 424*C^2
*a^7*b^7*d^11*f^2 + 380*C^2*a^9*b^5*d^11*f^2 + 100*C^2*a^11*b^3*d^11*f^2 -
20*C^2*b^14*c^3*d^8*f^2 + 60*C^2*a*b^13*d^11*f^2 + 8*C^2*a^13*b*d^11*f^2 -
4*C^2*a^14*c*d^10*f^2 - 12*C^2*b^14*c*d^10*f^2 + 84*C^2*a*b^13*c^2*d^9*f^2
+ 60*C^2*a^2*b^12*c*d^10*f^2 - 116*C^2*a^4*b^10*c*d^10*f^2 - 604*C^2*a^6*b^
8*c*d^10*f^2 - 596*C^2*a^8*b^6*c*d^10*f^2 - 220*C^2*a^10*b^4*c*d^10*f^2 - 4
4*C^2*a^12*b^2*c*d^10*f^2 + 116*C^2*a^2*b^12*c^3*d^8*f^2 + 108*C^2*a^3*b^11
*c^2*d^9*f^2 + 216*C^2*a^4*b^10*c^3*d^8*f^2 + 104*C^2*a^5*b^9*c^2*d^9*f^2 +
8*C^2*a^6*b^8*c^3*d^8*f^2 + 248*C^2*a^7*b^7*c^2*d^9*f^2 - 68*C^2*a^8*b^6*c
^3*d^8*f^2 + 196*C^2*a^9*b^5*c^2*d^9*f^2 + 4*C^2*a^10*b^4*c^3*d^8*f^2 + 28*
C^2*a^11*b^3*c^2*d^9*f^2))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5
*f^4 + 4*a^6*b^3*f^4))*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^
3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d
^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2
*f^4))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*
C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*
f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2))*(-(((8*C^2*a^4*c*f^2 + 8*C^2*
b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)
^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a
^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 1
6*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4
+ b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(
c + d*tan(e + f*x))^(1/2)*(2*C^4*b^10*d^12 - C^4*a^10*d^12 + 4*C^4*a^2*b^8*
d^12 + 27*C^4*a^4*b^6*d^12 - 15*C^4*a^6*b^4*d^12 - 9*C^4*a^8*b^2*d^12 + C^4
*a^10*c^2*d^10 + 4*C^4*b^10*c^2*d^10 + 2*C^4*b^10*c^4*d^8 + 24*C^4*a^2*b^8*
c^2*d^10 - 12*C^4*a^2*b^8*c^4*d^8 + 104*C^4*a^3*b^7*c^3*d^9 - 197*C^4*a^4*b
^6*c^2*d^10 + 18*C^4*a^4*b^6*c^4*d^8 - 32*C^4*a^5*b^5*c^3*d^9 - 17*C^4*a^6*
b^4*c^2*d^10 - 8*C^4*a^7*b^3*c^3*d^9 + 9*C^4*a^8*b^2*c^2*d^10 + 4*C^4*a^9*b
*c*d^11 - 40*C^4*a^3*b^7*c*d^11 + 132*C^4*a^5*b^5*c*d^11 + 48*C^4*a^7*b^3*c
*d^11))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^
4))*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^
3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 1
6*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*C^
2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 2
4*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4
+ 4*a^6*b^2*f^4)))^(1/2) - (16*(C^5*a^8*d^13 + 10*C^5*a^2*b^6*d^13 + 27*C^
5*a^4*b^4*d^13 + 10*C^5*a^6*b^2*d^13 + C^5*a^8*c^2*d^11 + 36*C^5*a^2*b^6*c^
2*d^11 + 26*C^5*a^2*b^6*c^4*d^9 - 40*C^5*a^3*b^5*c^3*d^10 + 29*C^5*a^4*b^4*
c^2*d^11 + 2*C^5*a^4*b^4*c^4*d^9 - 8*C^5*a^5*b^3*c^3*d^10 + 10*C^5*a^6*b^2*

c^2*d^11 - 8*C^5*a*b^7*c*d^12 - 16*C^5*a*b^7*c^3*d^10 - 8*C^5*a*b^7*c^5*d^8
 - 40*C^5*a^3*b^5*c*d^12 - 8*C^5*a^5*b^3*c*d^12))/(b^9*f^5 + a^8*b*f^5 + 4*
a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^5)))*(-(((8*C^2*a^4*c*f^2 + 8*C^2
*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2
)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*
a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 -
16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^
4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*2i - a
tan(((((8*(128*A^3*a^3*b^8*d^12*f^2 + 24*A^3*a^5*b^6*d^12*f^2 - 160*A^3*a^7
*b^4*d^12*f^2 - 4*A^3*a^9*b^2*d^12*f^2 + 20*A^3*b^11*c^3*d^9*f^2 - 52*A^3*a
*b^10*d^12*f^2 + 20*A^3*b^11*c*d^11*f^2 + 12*A^3*a*b^10*c^2*d^10*f^2 + 64*A
^3*a*b^10*c^4*d^8*f^2 - 256*A^3*a^2*b^9*c*d^11*f^2 + 72*A^3*a^4*b^7*c*d^11*
f^2 + 352*A^3*a^6*b^5*c*d^11*f^2 + 4*A^3*a^8*b^3*c*d^11*f^2 - 256*A^3*a^2*b
^9*c^3*d^9*f^2 - 128*A^3*a^3*b^8*c^4*d^8*f^2 + 72*A^3*a^4*b^7*c^3*d^9*f^2 -
 168*A^3*a^5*b^6*c^2*d^10*f^2 - 192*A^3*a^5*b^6*c^4*d^8*f^2 + 352*A^3*a^6*b
^5*c^3*d^9*f^2 - 160*A^3*a^7*b^4*c^2*d^10*f^2 + 4*A^3*a^8*b^3*c^3*d^9*f^2 -
 4*A^3*a^9*b^2*c^2*d^10*f^2))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^
4*f^5 + 4*a^6*b^2*f^5) + (((8*(32*A*b^15*d^11*f^4 + 96*A*a^2*b^13*d^11*f^4
- 320*A*a^6*b^9*d^11*f^4 - 480*A*a^8*b^7*d^11*f^4 - 288*A*a^10*b^5*d^11*f^4
 - 64*A*a^12*b^3*d^11*f^4 + 32*A*b^15*c^2*d^9*f^4 + 64*A*a*b^14*c^3*d^8*f^4
 + 320*A*a^3*b^12*c*d^10*f^4 + 640*A*a^5*b^10*c*d^10*f^4 + 640*A*a^7*b^8*c*
d^10*f^4 + 320*A*a^9*b^6*c*d^10*f^4 + 64*A*a^11*b^4*c*d^10*f^4 + 96*A*a^2*b
^13*c^2*d^9*f^4 + 320*A*a^3*b^12*c^3*d^8*f^4 + 640*A*a^5*b^10*c^3*d^8*f^4 -
 320*A*a^6*b^9*c^2*d^9*f^4 + 640*A*a^7*b^8*c^3*d^8*f^4 - 480*A*a^8*b^7*c^2*
d^9*f^4 + 320*A*a^9*b^6*c^3*d^8*f^4 - 288*A*a^10*b^5*c^2*d^9*f^4 + 64*A*a^1
1*b^4*c^3*d^8*f^4 - 64*A*a^12*b^3*c^2*d^9*f^4 + 64*A*a*b^14*c*d^10*f^4))/(a
^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) - (16*(c
+ d*tan(e + f*x))^(1/2)*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^
3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d
^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2
*f^4))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*
A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*
f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*(32*b^17*d^10*f^4 + 160*a^2*b^
15*d^10*f^4 + 288*a^4*b^13*d^10*f^4 + 160*a^6*b^11*d^10*f^4 - 160*a^8*b^9*d
^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b^5*d^10*f^4 - 32*a^14*b^3*d^10*
f^4 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*d^8*f^4 + 624*a^4*b^13*c^2*d^8
*f^4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^9*c^2*d^8*f^4 + 48*a^10*b^7*c^2
*d^8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14*b^3*c^2*d^8*f^4 + 16*a*b^16*c*
d^9*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336*a^5*b^12*c*d^9*f^4 + 560*a^7*b^10*c*
d^9*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*a^11*b^6*c*d^9*f^4 + 112*a^13*b^4*c*d
^9*f^4 + 16*a^15*b^2*c*d^9*f^4))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4
*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a
*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^
4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*
b^2*f^4))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 -

16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b ^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^ (1/2)*(20*A^2*a^3*b^10*d^11*f^2 - 88*A^2*a^5*b^8*d^11*f^2 + 40*A^2*a^7*b^6* d^11*f^2 + 84*A^2*a^9*b^4*d^11*f^2 + 4*A^2*a^11*b^2*d^11*f^2 - 20*A^2*b^13* c^3*d^8*f^2 + 68*A^2*a*b^12*d^11*f^2 - 8*A^2*b^13*c*d^10*f^2 + 116*A^2*a*b^ 12*c^2*d^9*f^2 + 104*A^2*a^2*b^11*c*d^10*f^2 + 48*A^2*a^4*b^9*c*d^10*f^2 - 304*A^2*a^6*b^7*c*d^10*f^2 - 296*A^2*a^8*b^5*c*d^10*f^2 - 56*A^2*a^10*b^3*c *d^10*f^2 + 116*A^2*a^2*b^11*c^3*d^8*f^2 + 204*A^2*a^3*b^10*c^2*d^9*f^2 + 2 16*A^2*a^4*b^9*c^3*d^8*f^2 + 168*A^2*a^5*b^8*c^2*d^9*f^2 + 8*A^2*a^6*b^7*c^ 3*d^8*f^2 + 184*A^2*a^7*b^6*c^2*d^9*f^2 - 68*A^2*a^8*b^5*c^3*d^8*f^2 + 100* A^2*a^9*b^4*c^2*d^9*f^2 + 4*A^2*a^10*b^3*c^3*d^8*f^2 - 4*A^2*a^11*b^2*c^2*d ^9*f^2))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3* b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16* b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*A^2* a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24* A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2))*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b ^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4* d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^ 2*f^4))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16 *A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6 *f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1 /2)*(3*A^4*b^9*d^12 - 3*A^4*a^2*b^7*d^12 + 17*A^4*a^4*b^5*d^12 - 9*A^4*a^6* b^3*d^12 + 3*A^4*b^9*c^2*d^10 + 2*A^4*b^9*c^4*d^8 + 63*A^4*a^2*b^7*c^2*d^10 - 12*A^4*a^2*b^7*c^4*d^8 + 96*A^4*a^3*b^6*c^3*d^9 - 123*A^4*a^4*b^5*c^2*d^ 10 + 18*A^4*a^4*b^5*c^4*d^8 - 24*A^4*a^5*b^4*c^3*d^9 + 9*A^4*a^6*b^3*c^2*d^ 10 + 12*A^4*a*b^8*c*d^11 - 8*A^4*a*b^8*c^3*d^9 - 56*A^4*a^3*b^6*c*d^11 + 60 *A^4*a^5*b^4*c*d^11))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16* a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3* b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6* a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*1i - (((8*(128*A^3*a^3*b^8*d^12*f^2 + 24*A^3*a^5*b^6*d^12*f^2 - 160*A^3*a^7*b^4*d^12*f^2 - 4*A^3*a^9*b^2*d^12*f^2 + 20*A^3*b^11*c^3*d^9*f^2 - 52*A^3*a*b^10*d^12*f^2 + 20*A^3*b^11*c*d^11*f^ 2 + 12*A^3*a*b^10*c^2*d^10*f^2 + 64*A^3*a*b^10*c^4*d^8*f^2 - 256*A^3*a^2*b^ 9*c*d^11*f^2 + 72*A^3*a^4*b^7*c*d^11*f^2 + 352*A^3*a^6*b^5*c*d^11*f^2 + 4*A ^3*a^8*b^3*c*d^11*f^2 - 256*A^3*a^2*b^9*c^3*d^9*f^2 - 128*A^3*a^3*b^8*c^4*d ^8*f^2 + 72*A^3*a^4*b^7*c^3*d^9*f^2 - 168*A^3*a^5*b^6*c^2*d^10*f^2 - 192*A^ 3*a^5*b^6*c^4*d^8*f^2 + 352*A^3*a^6*b^5*c^3*d^9*f^2 - 160*A^3*a^7*b^4*c^2*d ^10*f^2 + 4*A^3*a^8*b^3*c^3*d^9*f^2 - 4*A^3*a^9*b^2*c^2*d^10*f^2))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((8*(32*A*b^ 15*d^11*f^4 + 96*A*a^2*b^13*d^11*f^4 - 320*A*a^6*b^9*d^11*f^4 - 480*A*a^8*b

^7*d^11*f^4 - 288*A*a^10*b^5*d^11*f^4 - 64*A*a^12*b^3*d^11*f^4 + 32*A*b^15*
c^2*d^9*f^4 + 64*A*a*b^14*c^3*d^8*f^4 + 320*A*a^3*b^12*c*d^10*f^4 + 640*A*a
^5*b^10*c*d^10*f^4 + 640*A*a^7*b^8*c*d^10*f^4 + 320*A*a^9*b^6*c*d^10*f^4 +
64*A*a^11*b^4*c*d^10*f^4 + 96*A*a^2*b^13*c^2*d^9*f^4 + 320*A*a^3*b^12*c^3*d
^8*f^4 + 640*A*a^5*b^10*c^3*d^8*f^4 - 320*A*a^6*b^9*c^2*d^9*f^4 + 640*A*a^7
*b^8*c^3*d^8*f^4 - 480*A*a^8*b^7*c^2*d^9*f^4 + 320*A*a^9*b^6*c^3*d^8*f^4 -
288*A*a^10*b^5*c^2*d^9*f^4 + 64*A*a^11*b^4*c^3*d^8*f^4 - 64*A*a^12*b^3*c^2*
d^9*f^4 + 64*A*a*b^14*c*d^10*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a
^4*b^4*f^5 + 4*a^6*b^2*f^5) + (16*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^4*
c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*
a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*
b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2
*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2
)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))
^(1/2)*(32*b^17*d^10*f^4 + 160*a^2*b^15*d^10*f^4 + 288*a^4*b^13*d^10*f^4 +
160*a^6*b^11*d^10*f^4 - 160*a^8*b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*
a^12*b^5*d^10*f^4 - 32*a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^
15*c^2*d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*
a^8*b^9*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 1
6*a^14*b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336
*a^5*b^12*c*d^9*f^4 + 560*a^7*b^10*c*d^9*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*
a^11*b^6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2*c*d^9*f^4))/(a^8*
f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*A^2*a
^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A
^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a
^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*A^2*a^4*c*f^2 - 4*
A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*
f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4
)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^10*d^11*f^2 - 88*A
^2*a^5*b^8*d^11*f^2 + 40*A^2*a^7*b^6*d^11*f^2 + 84*A^2*a^9*b^4*d^11*f^2 + 4
*A^2*a^11*b^2*d^11*f^2 - 20*A^2*b^13*c^3*d^8*f^2 + 68*A^2*a*b^12*d^11*f^2 -
8*A^2*b^13*c*d^10*f^2 + 116*A^2*a*b^12*c^2*d^9*f^2 + 104*A^2*a^2*b^11*c*d^
10*f^2 + 48*A^2*a^4*b^9*c*d^10*f^2 - 304*A^2*a^6*b^7*c*d^10*f^2 - 296*A^2*a
^8*b^5*c*d^10*f^2 - 56*A^2*a^10*b^3*c*d^10*f^2 + 116*A^2*a^2*b^11*c^3*d^8*f
^2 + 204*A^2*a^3*b^10*c^2*d^9*f^2 + 216*A^2*a^4*b^9*c^3*d^8*f^2 + 168*A^2*a
^5*b^8*c^2*d^9*f^2 + 8*A^2*a^6*b^7*c^3*d^8*f^2 + 184*A^2*a^7*b^6*c^2*d^9*f^
2 - 68*A^2*a^8*b^5*c^3*d^8*f^2 + 100*A^2*a^9*b^4*c^2*d^9*f^2 + 4*A^2*a^10*b
^3*c^3*d^8*f^2 - 4*A^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^
6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f
^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 -
(A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*
f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a
*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*
f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2))*((((8*A^2*a^4
*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2

```
*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2
*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^
2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^
2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))
)^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(3*A^4*b^9*d^12 - 3*A^4*a^2*b^7*d^
12 + 17*A^4*a^4*b^5*d^12 - 9*A^4*a^6*b^3*d^12 + 3*A^4*b^9*c^2*d^10 + 2*A^4*
b^9*c^4*d^8 + 63*A^4*a^2*b^7*c^2*d^10 - 12*A^4*a^2*b^7*c^4*d^8 + 96*A^4*a^3
*b^6*c^3*d^9 - 123*A^4*a^4*b^5*c^2*d^10 + 18*A^4*a^4*b^5*c^4*d^8 - 24*A^4*a
^5*b^4*c^3*d^9 + 9*A^4*a^6*b^3*c^2*d^10 + 12*A^4*a*b^8*c*d^11 - 8*A^4*a*b^8
*c^3*d^9 - 56*A^4*a^3*b^6*c*d^11 + 60*A^4*a^5*b^4*c*d^11))/(a^8*f^4 + b^8*f
^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*A^2*a^4*c*f^2 +
8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*
c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4
+ 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f
^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(a
^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*1
i)/((16*(A^5*b^7*d^13 - 9*A^5*a^4*b^3*d^13 + 3*A^5*b^7*c^2*d^11 + 2*A^5*b^7
*c^4*d^9 - 22*A^5*a^2*b^5*c^2*d^11 - 22*A^5*a^2*b^5*c^4*d^9 + 24*A^5*a^3*b^
4*c^3*d^10 - 9*A^5*a^4*b^3*c^2*d^11 + 8*A^5*a*b^6*c^3*d^10 + 8*A^5*a*b^6*c^
5*d^8 + 24*A^5*a^3*b^4*c*d^12))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*
b^4*f^5 + 4*a^6*b^2*f^5) + (((8*(128*A^3*a^3*b^8*d^12*f^2 + 24*A^3*a^5*b^6*
d^12*f^2 - 160*A^3*a^7*b^4*d^12*f^2 - 4*A^3*a^9*b^2*d^12*f^2 + 20*A^3*b^11*
c^3*d^9*f^2 - 52*A^3*a*b^10*d^12*f^2 + 20*A^3*b^11*c*d^11*f^2 + 12*A^3*a*b^
10*c^2*d^10*f^2 + 64*A^3*a*b^10*c^4*d^8*f^2 - 256*A^3*a^2*b^9*c*d^11*f^2 +
72*A^3*a^4*b^7*c*d^11*f^2 + 352*A^3*a^6*b^5*c*d^11*f^2 + 4*A^3*a^8*b^3*c*d^
11*f^2 - 256*A^3*a^2*b^9*c^3*d^9*f^2 - 128*A^3*a^3*b^8*c^4*d^8*f^2 + 72*A^3
*a^4*b^7*c^3*d^9*f^2 - 168*A^3*a^5*b^6*c^2*d^10*f^2 - 192*A^3*a^5*b^6*c^4*d
^8*f^2 + 352*A^3*a^6*b^5*c^3*d^9*f^2 - 160*A^3*a^7*b^4*c^2*d^10*f^2 + 4*A^3
*a^8*b^3*c^3*d^9*f^2 - 4*A^3*a^9*b^2*c^2*d^10*f^2))/(a^8*f^5 + b^8*f^5 + 4*
a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((8*(32*A*b^15*d^11*f^4 + 9
6*A*a^2*b^13*d^11*f^4 - 320*A*a^6*b^9*d^11*f^4 - 480*A*a^8*b^7*d^11*f^4 - 2
88*A*a^10*b^5*d^11*f^4 - 64*A*a^12*b^3*d^11*f^4 + 32*A*b^15*c^2*d^9*f^4 + 6
4*A*a*b^14*c^3*d^8*f^4 + 320*A*a^3*b^12*c*d^10*f^4 + 640*A*a^5*b^10*c*d^10*
f^4 + 640*A*a^7*b^8*c*d^10*f^4 + 320*A*a^9*b^6*c*d^10*f^4 + 64*A*a^11*b^4*c
*d^10*f^4 + 96*A*a^2*b^13*c^2*d^9*f^4 + 320*A*a^3*b^12*c^3*d^8*f^4 + 640*A*
a^5*b^10*c^3*d^8*f^4 - 320*A*a^6*b^9*c^2*d^9*f^4 + 640*A*a^7*b^8*c^3*d^8*f^
4 - 480*A*a^8*b^7*c^2*d^9*f^4 + 320*A*a^9*b^6*c^3*d^8*f^4 - 288*A*a^10*b^5*
c^2*d^9*f^4 + 64*A*a^11*b^4*c^3*d^8*f^4 - 64*A*a^12*b^3*c^2*d^9*f^4 + 64*A*
a*b^14*c*d^10*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*
a^6*b^2*f^5) - (16*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^4*c*f^2 + 8*A^2*b
^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^
2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^
4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16
*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4
+ b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*(32*b^17
```

*d^10*f^4 + 160*a^2*b^15*d^10*f^4 + 288*a^4*b^13*d^10*f^4 + 160*a^6*b^11*d^10*f^4 - 160*a^8*b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b^5*d^10*f^4 - 32*a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^9*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14*b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336*a^5*b^12*c*d^9*f^4 + 560*a^7*b^10*c*d^9*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*a^11*b^6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2*c*d^9*f^4))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^10*d^11*f^2 - 88*A^2*a^5*b^8*d^11*f^2 + 40*A^2*a^7*b^6*d^11*f^2 + 84*A^2*a^9*b^4*d^11*f^2 + 4*A^2*a^11*b^2*d^11*f^2 - 20*A^2*b^13*c^3*d^8*f^2 + 68*A^2*a*b^12*d^11*f^2 - 8*A^2*b^13*c*d^10*f^2 + 116*A^2*a*b^12*c^2*d^9*f^2 + 104*A^2*a^2*b^11*c*d^10*f^2 + 48*A^2*a^4*b^9*c*d^10*f^2 - 304*A^2*a^6*b^7*c*d^10*f^2 - 296*A^2*a^8*b^5*c*d^10*f^2 - 56*A^2*a^10*b^3*c*d^10*f^2 + 116*A^2*a^2*b^11*c^3*d^8*f^2 + 204*A^2*a^3*b^10*c^2*d^9*f^2 + 216*A^2*a^4*b^9*c^3*d^8*f^2 + 168*A^2*a^5*b^8*c^2*d^9*f^2 + 8*A^2*a^6*b^7*c^3*d^8*f^2 + 184*A^2*a^7*b^6*c^2*d^9*f^2 - 68*A^2*a^8*b^5*c^3*d^8*f^2 + 100*A^2*a^9*b^4*c^2*d^9*f^2 + 4*A^2*a^10*b^3*c^3*d^8*f^2 - 4*A^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2))*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(3*A^4*b^9*d^12 - 3*A^4*a^2*b^7*d^12 + 17*A^4*a^4*b^5*d^12 - 9*A^4*a^6*b^3*d^12 + 3*A^4*b^9*c^2*d^10 + 2*A^4*b^9*c^4*d^8 + 63*A^4*a^2*b^7*c^2*d^10 - 12*A^4*a^2*b^7*c^4*d^8 + 96*A^4*a^3*b^6*c^3*d^9 - 123*A^4*a^4*b^5*c^2*d^10 + 18*A^4*a^4*b^5*c^4*d^8 - 24*A^4*a^5*b^4*c^3*d^9 + 9*A^4*a^6*b^3*c^2*d^10 + 12*A^4*a*b^8*c*d^11 - 8*A^4*a*b^8*c^3*d^9 - 56*A^4*a^3*b^6*c*d^11 + 60*A^4*a^5*b^4*c*d^11))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b

^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) + (((8*(128*A^3*a^3*b^8*d^12*f^2 + 24*A^3*a^5*b^6*d^12*f^2 - 160*A^3*a^7*b^4*d^12*f^2 - 4*A^3*a^9*b^2*d^12*f^2 + 20*A^3*b^11*c^3*d^9*f^2 - 52*A^3*a*b^10*d^12*f^2 + 20*A^3*b^11*c*d^11*f^2 + 12*A^3*a*b^10*c^2*d^10*f^2 + 64*A^3*a*b^10*c^4*d^8*f^2 - 256*A^3*a^2*b^9*c*d^11*f^2 + 72*A^3*a^4*b^7*c*d^11*f^2 + 352*A^3*a^6*b^5*c*d^11*f^2 + 4*A^3*a^8*b^3*c*d^11*f^2 - 256*A^3*a^2*b^9*c^3*d^9*f^2 - 128*A^3*a^3*b^8*c^4*d^8*f^2 + 72*A^3*a^4*b^7*c^3*d^9*f^2 - 168*A^3*a^5*b^6*c^2*d^10*f^2 - 192*A^3*a^5*b^6*c^4*d^8*f^2 + 352*A^3*a^6*b^5*c^3*d^9*f^2 - 160*A^3*a^7*b^4*c^2*d^10*f^2 + 4*A^3*a^8*b^3*c^3*d^9*f^2 - 4*A^3*a^9*b^2*c^2*d^10*f^2))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((8*(32*A*b^15*d^11*f^4 + 96*A*a^2*b^13*d^11*f^4 - 320*A*a^6*b^9*d^11*f^4 - 480*A*a^8*b^7*d^11*f^4 - 288*A*a^10*b^5*d^11*f^4 - 64*A*a^12*b^3*d^11*f^4 + 32*A*b^15*c^2*d^9*f^4 + 64*A*a*b^14*c^3*d^8*f^4 + 320*A*a^3*b^12*c*d^10*f^4 + 640*A*a^5*b^10*c*d^10*f^4 + 640*A*a^7*b^8*c*d^10*f^4 + 320*A*a^9*b^6*c*d^10*f^4 + 64*A*a^11*b^4*c*d^10*f^4 + 96*A*a^2*b^13*c^2*d^9*f^4 + 320*A*a^3*b^12*c^3*d^8*f^4 + 640*A*a^5*b^10*c^3*d^8*f^4 - 320*A*a^6*b^9*c^2*d^9*f^4 + 640*A*a^7*b^8*c^3*d^8*f^4 - 480*A*a^8*b^7*c^2*d^9*f^4 + 320*A*a^9*b^6*c^3*d^8*f^4 - 288*A*a^10*b^5*c^2*d^9*f^4 + 64*A*a^11*b^4*c^3*d^8*f^4 - 64*A*a^12*b^3*c^2*d^9*f^4 + 64*A*a*b^14*c*d^10*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (16*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*(32*b^17*d^10*f^4 + 160*a^2*b^15*d^10*f^4 + 288*a^4*b^13*d^10*f^4 + 160*a^6*b^11*d^10*f^4 - 160*a^8*b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b^5*d^10*f^4 - 32*a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^9*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14*b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336*a^5*b^12*c*d^9*f^4 + 560*a^7*b^10*c*d^9*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*a^11*b^6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2*c*d^9*f^4))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^10*d^11*f^2 - 88*A^2*a^5*b^8*d^11*f^2 + 40*A^2*a^7*b^6*d^11*f^2 + 84*A^2*a^9*b^4*d^11*f^2 + 4*A^2*a^11*b^2*d^11*f^2 - 20*A^2*b^13*c^3*d^8*f^2 + 68*A^2*a*b^12*d^11*f^2 - 8*A^2*b^13*c*d^10*f^2 + 116*A^2*a*b^12*c^2*d^9*f^2 + 104*A^2*a^2*b^11*c*d^10*f^2 + 48*A^2*a^4*b^9*c*d^10*f^2 - 304*A^2*a^6*b^7*c*d

^10*f^2 - 296*A^2*a^8*b^5*c*d^10*f^2 - 56*A^2*a^10*b^3*c*d^10*f^2 + 116*A^2
*a^2*b^11*c^3*d^8*f^2 + 204*A^2*a^3*b^10*c^2*d^9*f^2 + 216*A^2*a^4*b^9*c^3*
d^8*f^2 + 168*A^2*a^5*b^8*c^2*d^9*f^2 + 8*A^2*a^6*b^7*c^3*d^8*f^2 + 184*A^2
*a^7*b^6*c^2*d^9*f^2 - 68*A^2*a^8*b^5*c^3*d^8*f^2 + 100*A^2*a^9*b^4*c^2*d^9
*f^2 + 4*A^2*a^10*b^3*c^3*d^8*f^2 - 4*A^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 +
b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*A^2*a^4*c*
f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^
2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^
6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b
^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/
(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(
1/2))*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a
^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 +
16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*A
^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 +
24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^
4 + 4*a^6*b^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(3*A^4*b^9*d^12
- 3*A^4*a^2*b^7*d^12 + 17*A^4*a^4*b^5*d^12 - 9*A^4*a^6*b^3*d^12 + 3*A^4*b^
9*c^2*d^10 + 2*A^4*b^9*c^4*d^8 + 63*A^4*a^2*b^7*c^2*d^10 - 12*A^4*a^2*b^7*c
^4*d^8 + 96*A^4*a^3*b^6*c^3*d^9 - 123*A^4*a^4*b^5*c^2*d^10 + 18*A^4*a^4*b^5
*c^4*d^8 - 24*A^4*a^5*b^4*c^3*d^9 + 9*A^4*a^6*b^3*c^2*d^10 + 12*A^4*a*b^8*c
*d^11 - 8*A^4*a*b^8*c^3*d^9 - 56*A^4*a^3*b^6*c*d^11 + 60*A^4*a^5*b^4*c*d^11
))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(((
(8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^
2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^
4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*A^2*a^4*c*
f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^
2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6
*b^2*f^4)))^(1/2)))*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*
f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*
(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4
))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*
a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4
+ 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*2i - atan(((((8*(128*A^3*a^3*b^8*d
^12*f^2 + 24*A^3*a^5*b^6*d^12*f^2 - 160*A^3*a^7*b^4*d^12*f^2 - 4*A^3*a^9*b^
2*d^12*f^2 + 20*A^3*b^11*c^3*d^9*f^2 - 52*A^3*a*b^10*d^12*f^2 + 20*A^3*b^11
*c*d^11*f^2 + 12*A^3*a*b^10*c^2*d^10*f^2 + 64*A^3*a*b^10*c^4*d^8*f^2 - 256*
A^3*a^2*b^9*c*d^11*f^2 + 72*A^3*a^4*b^7*c*d^11*f^2 + 352*A^3*a^6*b^5*c*d^11
*f^2 + 4*A^3*a^8*b^3*c*d^11*f^2 - 256*A^3*a^2*b^9*c^3*d^9*f^2 - 128*A^3*a^3
*b^8*c^4*d^8*f^2 + 72*A^3*a^4*b^7*c^3*d^9*f^2 - 168*A^3*a^5*b^6*c^2*d^10*f^
2 - 192*A^3*a^5*b^6*c^4*d^8*f^2 + 352*A^3*a^6*b^5*c^3*d^9*f^2 - 160*A^3*a^7
*b^4*c^2*d^10*f^2 + 4*A^3*a^8*b^3*c^3*d^9*f^2 - 4*A^3*a^9*b^2*c^2*d^10*f^2)
)/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((
8*(32*A*b^15*d^11*f^4 + 96*A*a^2*b^13*d^11*f^4 - 320*A*a^6*b^9*d^11*f^4 - 4
80*A*a^8*b^7*d^11*f^4 - 288*A*a^10*b^5*d^11*f^4 - 64*A*a^12*b^3*d^11*f^4 +

32*A*b^15*c^2*d^9*f^4 + 64*A*a*b^14*c^3*d^8*f^4 + 320*A*a^3*b^12*c*d^10*f^4 + 640*A*a^5*b^10*c*d^10*f^4 + 640*A*a^7*b^8*c*d^10*f^4 + 320*A*a^9*b^6*c*d^10*f^4 + 64*A*a^11*b^4*c*d^10*f^4 + 96*A*a^2*b^13*c^2*d^9*f^4 + 320*A*a^3*b^12*c^3*d^8*f^4 + 640*A*a^5*b^10*c^3*d^8*f^4 - 320*A*a^6*b^9*c^2*d^9*f^4 + 640*A*a^7*b^8*c^3*d^8*f^4 - 480*A*a^8*b^7*c^2*d^9*f^4 + 320*A*a^9*b^6*c^3*d^8*f^4 - 288*A*a^10*b^5*c^2*d^9*f^4 + 64*A*a^11*b^4*c^3*d^8*f^4 - 64*A*a^12*b^3*c^2*d^9*f^4 + 64*A*a*b^14*c*d^10*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) - (16*(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*(32*b^17*d^10*f^4 + 160*a^2*b^15*d^10*f^4 + 288*a^4*b^13*d^10*f^4 + 160*a^6*b^11*d^10*f^4 - 160*a^8*b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b^5*d^10*f^4 - 32*a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^9*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14*b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336*a^5*b^12*c*d^9*f^4 + 560*a^7*b^10*c*d^9*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*a^11*b^6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2*c*d^9*f^4))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^10*d^11*f^2 - 88*A^2*a^5*b^8*d^11*f^2 + 40*A^2*a^7*b^6*d^11*f^2 + 84*A^2*a^9*b^4*d^11*f^2 + 4*A^2*a^11*b^2*d^11*f^2 - 20*A^2*b^13*c^3*d^8*f^2 + 68*A^2*a*b^12*d^11*f^2 - 8*A^2*b^13*c*d^10*f^2 + 116*A^2*a*b^12*c^2*d^9*f^2 + 104*A^2*a^2*b^11*c*d^10*f^2 + 48*A^2*a^4*b^9*c*d^10*f^2 - 304*A^2*a^6*b^7*c*d^10*f^2 - 296*A^2*a^8*b^5*c*d^10*f^2 - 56*A^2*a^10*b^3*c*d^10*f^2 + 116*A^2*a^2*b^11*c^3*d^8*f^2 + 204*A^2*a^3*b^10*c^2*d^9*f^2 + 216*A^2*a^4*b^9*c^3*d^8*f^2 + 168*A^2*a^5*b^8*c^2*d^9*f^2 + 8*A^2*a^6*b^7*c^3*d^8*f^2 + 184*A^2*a^7*b^6*c^2*d^9*f^2 - 68*A^2*a^8*b^5*c^3*d^8*f^2 + 100*A^2*a^9*b^4*c^2*d^9*f^2 + 4*A^2*a^10*b^3*c^3*d^8*f^2 - 4*A^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2))*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^

8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^
4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^
2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4
*a^6*b^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(3*A^4*b^9*d^12 - 3*
A^4*a^2*b^7*d^12 + 17*A^4*a^4*b^5*d^12 - 9*A^4*a^6*b^3*d^12 + 3*A^4*b^9*c^2
*d^10 + 2*A^4*b^9*c^4*d^8 + 63*A^4*a^2*b^7*c^2*d^10 - 12*A^4*a^2*b^7*c^4*d^
8 + 96*A^4*a^3*b^6*c^3*d^9 - 123*A^4*a^4*b^5*c^2*d^10 + 18*A^4*a^4*b^5*c^4*
d^8 - 24*A^4*a^5*b^4*c^3*d^9 + 9*A^4*a^6*b^3*c^2*d^10 + 12*A^4*a*b^8*c*d^11
 - 8*A^4*a*b^8*c^3*d^9 - 56*A^4*a^3*b^6*c*d^11 + 60*A^4*a^5*b^4*c*d^11))/(a
^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8*A
^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 -
48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 +
64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^4*c*f^2
+ 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^
2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2
*f^4)))^(1/2)*1i - (((8*(128*A^3*a^3*b^8*d^12*f^2 + 24*A^3*a^5*b^6*d^12*f^2
 - 160*A^3*a^7*b^4*d^12*f^2 - 4*A^3*a^9*b^2*d^12*f^2 + 20*A^3*b^11*c^3*d^9*
f^2 - 52*A^3*a*b^10*d^12*f^2 + 20*A^3*b^11*c*d^11*f^2 + 12*A^3*a*b^10*c^2*d
^10*f^2 + 64*A^3*a*b^10*c^4*d^8*f^2 - 256*A^3*a^2*b^9*c*d^11*f^2 + 72*A^3*a
^4*b^7*c*d^11*f^2 + 352*A^3*a^6*b^5*c*d^11*f^2 + 4*A^3*a^8*b^3*c*d^11*f^2 -
 256*A^3*a^2*b^9*c^3*d^9*f^2 - 128*A^3*a^3*b^8*c^4*d^8*f^2 + 72*A^3*a^4*b^7
*c^3*d^9*f^2 - 168*A^3*a^5*b^6*c^2*d^10*f^2 - 192*A^3*a^5*b^6*c^4*d^8*f^2 +
 352*A^3*a^6*b^5*c^3*d^9*f^2 - 160*A^3*a^7*b^4*c^2*d^10*f^2 + 4*A^3*a^8*b^3
*c^3*d^9*f^2 - 4*A^3*a^9*b^2*c^2*d^10*f^2))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*
f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((8*(32*A*b^15*d^11*f^4 + 96*A*a^2*
b^13*d^11*f^4 - 320*A*a^6*b^9*d^11*f^4 - 480*A*a^8*b^7*d^11*f^4 - 288*A*a^1
0*b^5*d^11*f^4 - 64*A*a^12*b^3*d^11*f^4 + 32*A*b^15*c^2*d^9*f^4 + 64*A*a*b^
14*c^3*d^8*f^4 + 320*A*a^3*b^12*c*d^10*f^4 + 640*A*a^5*b^10*c*d^10*f^4 + 64
0*A*a^7*b^8*c*d^10*f^4 + 320*A*a^9*b^6*c*d^10*f^4 + 64*A*a^11*b^4*c*d^10*f^
4 + 96*A*a^2*b^13*c^2*d^9*f^4 + 320*A*a^3*b^12*c^3*d^8*f^4 + 640*A*a^5*b^10
*c^3*d^8*f^4 - 320*A*a^6*b^9*c^2*d^9*f^4 + 640*A*a^7*b^8*c^3*d^8*f^4 - 480*
A*a^8*b^7*c^2*d^9*f^4 + 320*A*a^9*b^6*c^3*d^8*f^4 - 288*A*a^10*b^5*c^2*d^9*
f^4 + 64*A*a^11*b^4*c^3*d^8*f^4 - 64*A*a^12*b^3*c^2*d^9*f^4 + 64*A*a*b^14*c
*d^10*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*
f^5) + (16*(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^
2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (
A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f
^4 + 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*
b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f
^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*(32*b^17*d^10*f
^4 + 160*a^2*b^15*d^10*f^4 + 288*a^4*b^13*d^10*f^4 + 160*a^6*b^11*d^10*f^4
- 160*a^8*b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b^5*d^10*f^4 - 32
*a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*d^8*f^4 + 624*a
^4*b^13*c^2*d^8*f^4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^9*c^2*d^8*f^4 +
48*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14*b^3*c^2*d^8*f^4

+ 16*a*b^16*c*d^9*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336*a^5*b^12*c*d^9*f^4 +
560*a^7*b^10*c*d^9*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*a^11*b^6*c*d^9*f^4 + 1
12*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2*c*d^9*f^4))/(a^8*f^4 + b^8*f^4 + 4*a^2*
b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*
c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4
 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b
^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^
2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b
^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(c +
d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^10*d^11*f^2 - 88*A^2*a^5*b^8*d^11*f^2 +
 40*A^2*a^7*b^6*d^11*f^2 + 84*A^2*a^9*b^4*d^11*f^2 + 4*A^2*a^11*b^2*d^11*f^
2 - 20*A^2*b^13*c^3*d^8*f^2 + 68*A^2*a*b^12*d^11*f^2 - 8*A^2*b^13*c*d^10*f^
2 + 116*A^2*a*b^12*c^2*d^9*f^2 + 104*A^2*a^2*b^11*c*d^10*f^2 + 48*A^2*a^4*b
^9*c*d^10*f^2 - 304*A^2*a^6*b^7*c*d^10*f^2 - 296*A^2*a^8*b^5*c*d^10*f^2 - 5
6*A^2*a^10*b^3*c*d^10*f^2 + 116*A^2*a^2*b^11*c^3*d^8*f^2 + 204*A^2*a^3*b^10
*c^2*d^9*f^2 + 216*A^2*a^4*b^9*c^3*d^8*f^2 + 168*A^2*a^5*b^8*c^2*d^9*f^2 +
8*A^2*a^6*b^7*c^3*d^8*f^2 + 184*A^2*a^7*b^6*c^2*d^9*f^2 - 68*A^2*a^8*b^5*c^
3*d^8*f^2 + 100*A^2*a^9*b^4*c^2*d^9*f^2 + 4*A^2*a^10*b^3*c^3*d^8*f^2 - 4*A^
2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4
 + 4*a^6*b^2*f^4))*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*
f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*
(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4
))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*
a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4
+ 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2))*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*
c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4
 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b
^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^
2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b
^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) + (16*(c +
d*tan(e + f*x))^(1/2)*(3*A^4*b^9*d^12 - 3*A^4*a^2*b^7*d^12 + 17*A^4*a^4*b^5
*d^12 - 9*A^4*a^6*b^3*d^12 + 3*A^4*b^9*c^2*d^10 + 2*A^4*b^9*c^4*d^8 + 63*A^
4*a^2*b^7*c^2*d^10 - 12*A^4*a^2*b^7*c^4*d^8 + 96*A^4*a^3*b^6*c^3*d^9 - 123*
A^4*a^4*b^5*c^2*d^10 + 18*A^4*a^4*b^5*c^4*d^8 - 24*A^4*a^5*b^4*c^3*d^9 + 9*
A^4*a^6*b^3*c^2*d^10 + 12*A^4*a*b^8*c*d^11 - 8*A^4*a*b^8*c^3*d^9 - 56*A^4*a
^3*b^6*c*d^11 + 60*A^4*a^5*b^4*c*d^11))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4
+ 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 -
32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*
c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 +
 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*
d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 +
 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*1i)/((16*(A^5*b^7*d
^13 - 9*A^5*a^4*b^3*d^13 + 3*A^5*b^7*c^2*d^11 + 2*A^5*b^7*c^4*d^9 - 22*A^5*
a^2*b^5*c^2*d^11 - 22*A^5*a^2*b^5*c^4*d^9 + 24*A^5*a^3*b^4*c^3*d^10 - 9*A^5
*a^4*b^3*c^2*d^11 + 8*A^5*a*b^6*c^3*d^10 + 8*A^5*a*b^6*c^5*d^8 + 24*A^5*a^3

*b^4*c*d^12))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((8*(128*A^3*a^3*b^8*d^12*f^2 + 24*A^3*a^5*b^6*d^12*f^2 - 160*A^3*a^7*b^4*d^12*f^2 - 4*A^3*a^9*b^2*d^12*f^2 + 20*A^3*b^11*c^3*d^9*f^2 - 52*A^3*a*b^10*d^12*f^2 + 20*A^3*b^11*c*d^11*f^2 + 12*A^3*a*b^10*c^2*d^10*f^2 + 64*A^3*a*b^10*c^4*d^8*f^2 - 256*A^3*a^2*b^9*c*d^11*f^2 + 72*A^3*a^4*b^7*c*d^11*f^2 + 352*A^3*a^6*b^5*c*d^11*f^2 + 4*A^3*a^8*b^3*c*d^11*f^2 - 256*A^3*a^2*b^9*c^3*d^9*f^2 - 128*A^3*a^3*b^8*c^4*d^8*f^2 + 72*A^3*a^4*b^7*c^3*d^9*f^2 - 168*A^3*a^5*b^6*c^2*d^10*f^2 - 192*A^3*a^5*b^6*c^4*d^8*f^2 + 352*A^3*a^6*b^5*c^3*d^9*f^2 - 160*A^3*a^7*b^4*c^2*d^10*f^2 + 4*A^3*a^8*b^3*c^3*d^9*f^2 - 4*A^3*a^9*b^2*c^2*d^10*f^2))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((8*(32*A*b^15*d^11*f^4 + 96*A*a^2*b^13*d^11*f^4 - 320*A*a^6*b^9*d^11*f^4 - 480*A*a^8*b^7*d^11*f^4 - 288*A*a^10*b^5*d^11*f^4 - 64*A*a^12*b^3*d^11*f^4 + 32*A*b^15*c^2*d^9*f^4 + 64*A*a*b^14*c^3*d^8*f^4 + 320*A*a^3*b^12*c*d^10*f^4 + 640*A*a^5*b^10*c*d^10*f^4 + 640*A*a^7*b^8*c*d^10*f^4 + 320*A*a^9*b^6*c*d^10*f^4 + 64*A*a^11*b^4*c*d^10*f^4 + 96*A*a^2*b^13*c^2*d^9*f^4 + 320*A*a^3*b^12*c^3*d^8*f^4 + 640*A*a^5*b^10*c^3*d^8*f^4 - 320*A*a^6*b^9*c^2*d^9*f^4 + 640*A*a^7*b^8*c^3*d^8*f^4 - 480*A*a^8*b^7*c^2*d^9*f^4 + 320*A*a^9*b^6*c^3*d^8*f^4 - 288*A*a^10*b^5*c^2*d^9*f^4 + 64*A*a^11*b^4*c^3*d^8*f^4 - 64*A*a^12*b^3*c^2*d^9*f^4 + 64*A*a*b^14*c*d^10*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) - (16*(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*(32*b^17*d^10*f^4 + 160*a^2*b^15*d^10*f^4 + 288*a^4*b^13*d^10*f^4 + 160*a^6*b^11*d^10*f^4 - 160*a^8*b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b^5*d^10*f^4 - 32*a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^9*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14*b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336*a^5*b^12*c*d^9*f^4 + 560*a^7*b^10*c*d^9*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*a^11*b^6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2*c*d^9*f^4))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^10*d^11*f^2 - 88*A^2*a^5*b^8*d^11*f^2 + 40*A^2*a^7*b^6*d^11*f^2 + 84*A^2*a^9*b^4*d^11*f^2 + 4*A^2*a^11*b^2*d^11*f^2 - 20*A^2*b^13*c^3*d^8*f^2 + 68*A^2*a*b^12*d^11*f^2 - 8*A^2*b^13*c*d^10*f^2 + 116*A^2*a*b^12*c^2*d^9*f^2 + 104*A^2*a^2*b^11*c*d^10*f^2 + 48*A^2*a^4*b^9*c*d^10*f^2 - 304*A^2*a^6*b^7*c*d^10*f^2 - 296*A^2*a^8*b^5*c*d^10*f^2 - 56*A^2*a^10 *b^3*c*d^10*f^2 + 116*A^2*a^2*b^11*c^3*d^8*f^2 + 204*A^2*a^3*b^10*c^2*d^9*f^2 + 216*A^2*a^4*b^9*c^3*d^8*f^2 + 168*A^2*a^5*b^8*c^2*d^9*f^2 + 8*A^2*a^6*b^7*c^3*d^8*f^2 + 184*A^2*a^7*b^6*c^2*d^9*f^2 - 68*A^2*a^8*b^5*c^3*d^8*f^2 + 100*A^2*a^9*b^4*c^2*d^9*f^2 + 4*A^2*a^10*b^3*c^3*d^8*f^2 - 4*A^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2))*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(3*A^4*b^9*d^12 - 3*A^4*a^2*b^7*d^12 + 17*A^4*a^4*b^5*d^12 - 9*A^4*a^6*b^3*d^12 + 3*A^4*b^9*c^2*d^10 + 2*A^4*b^9*c^4*d^8 + 63*A^4*a^2*b^7*c^2*d^10 - 12*A^4*a^2*b^7*c^4*d^8 + 96*A^4*a^3*b^6*c^3*d^9 - 123*A^4*a^4*b^5*c^2*d^10 + 18*A^4*a^4*b^5*c^4*d^8 - 24*A^4*a^5*b^4*c^3*d^9 + 9*A^4*a^6*b^3*c^2*d^10 + 12*A^4*a*b^8*c*d^11 - 8*A^4*a*b^8*c^3*d^9 - 56*A^4*a^3*b^6*c*d^11 + 60*A^4*a^5*b^4*c*d^11))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) + (((8*(128*A^3*a^3*b^8*d^12*f^2 + 24*A^3*a^5*b^6*d^12*f^2 - 160*A^3*a^7*b^4*d^12*f^2 - 4*A^3*a^9*b^2*d^12*f^2 + 20*A^3*b^11*c^3*d^9*f^2 - 52*A^3*a*b^10*d^12*f^2 + 20*A^3*b^11*c*d^11*f^2 + 12*A^3*a*b^10*c^2*d^10*f^2 + 64*A^3*a*b^10*c^4*d^8*f^2 - 256*A^3*a^2*b^9*c*d^11*f^2 + 72*A^3*a^4*b^7*c*d^11*f^2 + 352*A^3*a^6*b^5*c*d^11*f^2 + 4*A^3*a^8*b^3*c*d^11*f^2 - 256*A^3*a^2*b^9*c^3*d^9*f^2 - 128*A^3*a^3*b^8*c^4*d^8*f^2 + 72*A^3*a^4*b^7*c^3*d^9*f^2 - 168*A^3*a^5*b^6*c^2*d^10*f^2 - 192*A^3*a^5*b^6*c^4*d^8*f^2 + 352*A^3*a^6*b^5*c^3*d^9*f^2 - 160*A^3*a^7*b^4*c^2*d^10*f^2 + 4*A^3*a^8*b^3*c^3*d^9*f^2 - 4*A^3*a^9*b^2*c^2*d^10*f^2))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((8*(32*A*b^15*d^11*f^4 + 96*A*a^2*b^13*d^11*f^4 - 320*A*a^6*b^9*d^11*f^4 - 480*A*a^8*b^7*d^11*f^4 - 288*A*a^10*b^5*d^11*f^4 - 64*A*a^12*b^3*d^11*f^4 + 32*A*b^15*c^2*d^9*f^4 + 64*A*a*b^14*c^3*d^8*f^4 + 320*A*a^3*b^12*c*d^10*f^4 + 640*A*a^5*b^10*c*d^10*f^4 + 640*A*a^7*b^8*c*d^10*f^4 + 320*A*a^9*b^6*c*d^10*f^4 + 64*A*a^11*b^4*c*d^10*f^4 + 96*A*a^2*b^13*c^2*d^9*f^4 + 320*A*a^3*b^12*c^3*d^8*f^4 + 640*A*a^5*b^10*c^3*d^8*f^4 - 320*A*a^6*b^9*c^2*d^9*f^4 + 640*A*a^7*b^8*c^3*d^8*f^4 - 480*A*a^8*b^7*c^2*d^9*f^4 + 320*A*a^9*b^6*c^3*d^8*f^4 - 288*A*a^10*b^5*c^2*d^9*f^4 + 64*A*a^11*b^4*c^3*d^8*f^4 - 64*A*a^12*b^

3*c^2*d^9*f^4 + 64*A*a*b^14*c*d^10*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (16*(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*(32*b^17*d^10*f^4 + 160*a^2*b^15*d^10*f^4 + 288*a^4*b^13*d^10*f^4 + 160*a^6*b^11*d^10*f^4 - 160*a^8*b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b^5*d^10*f^4 - 32*a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^9*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14*b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336*a^5*b^12*c*d^9*f^4 + 560*a^7*b^10*c*d^9*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*a^11*b^6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2*c*d^9*f^4))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^10*d^11*f^2 - 88*A^2*a^5*b^8*d^11*f^2 + 40*A^2*a^7*b^6*d^11*f^2 + 84*A^2*a^9*b^4*d^11*f^2 + 4*A^2*a^11*b^2*d^11*f^2 - 20*A^2*b^13*c^3*d^8*f^2 + 68*A^2*a*b^12*d^11*f^2 - 8*A^2*b^13*c*d^10*f^2 + 116*A^2*a*b^12*c^2*d^9*f^2 + 104*A^2*a^2*b^11*c*d^10*f^2 + 48*A^2*a^4*b^9*c*d^10*f^2 - 304*A^2*a^6*b^7*c*d^10*f^2 - 296*A^2*a^8*b^5*c*d^10*f^2 - 56*A^2*a^10*b^3*c*d^10*f^2 + 116*A^2*a^2*b^11*c^3*d^8*f^2 + 204*A^2*a^3*b^10*c^2*d^9*f^2 + 216*A^2*a^4*b^9*c^3*d^8*f^2 + 168*A^2*a^5*b^8*c^2*d^9*f^2 + 8*A^2*a^6*b^7*c^3*d^8*f^2 + 184*A^2*a^7*b^6*c^2*d^9*f^2 - 68*A^2*a^8*b^5*c^3*d^8*f^2 + 100*A^2*a^9*b^4*c^2*d^9*f^2 + 4*A^2*a^10*b^3*c^3*d^8*f^2 - 4*A^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2))*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(3*A^4*b^9*d^12 - 3*A^4*a^2*b^7*d^12 + 17*A^4*a^4*b^5*d^12 - 9*A^4*a^6*b^3*d^12 + 3*A^4*b^9*c^2*d^10 + 2*A^4*b^9*c^4*d^8 + 63*A^4*a^2*b^7*c^2*d^10 - 12*A^4*a^2*b^7*c^4*d^8 +

96*A^4*a^3*b^6*c^3*d^9 - 123*A^4*a^4*b^5*c^2*d^10 + 18*A^4*a^4*b^5*c^4*d^8
- 24*A^4*a^5*b^4*c^3*d^9 + 9*A^4*a^6*b^3*c^2*d^10 + 12*A^4*a*b^8*c*d^11 - 8
*A^4*a*b^8*c^3*d^9 - 56*A^4*a^3*b^6*c*d^11 + 60*A^4*a^5*b^4*c*d^11))/(a^8*f
^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8*A^2*a
^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A
^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a
^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^4*c*f^2 + 4*
A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*
f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4
)))^(1/2)))*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 3
2*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8
*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2
) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d
*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4
*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*2i - atan(((((8*(304*C^3*a^3*b^9*d^12*f^2
 + 120*C^3*a^5*b^7*d^12*f^2 - 320*C^3*a^7*b^5*d^12*f^2 - 148*C^3*a^9*b^3*d^
12*f^2 + 4*C^3*b^12*c^3*d^9*f^2 - 4*C^3*a*b^11*d^12*f^2 - 16*C^3*a^11*b*d^1
2*f^2 + 4*C^3*b^12*c*d^11*f^2 + 60*C^3*a*b^11*c^2*d^10*f^2 + 64*C^3*a*b^11*
c^4*d^8*f^2 - 320*C^3*a^2*b^10*c*d^11*f^2 + 104*C^3*a^4*b^8*c*d^11*f^2 + 54
4*C^3*a^6*b^6*c*d^11*f^2 + 116*C^3*a^8*b^4*c*d^11*f^2 - 16*C^3*a^11*b*c^2*d
^10*f^2 - 320*C^3*a^2*b^10*c^3*d^9*f^2 + 176*C^3*a^3*b^9*c^2*d^10*f^2 - 128
*C^3*a^3*b^9*c^4*d^8*f^2 + 104*C^3*a^4*b^8*c^3*d^9*f^2 - 72*C^3*a^5*b^7*c^2
*d^10*f^2 - 192*C^3*a^5*b^7*c^4*d^8*f^2 + 544*C^3*a^6*b^6*c^3*d^9*f^2 - 320
*C^3*a^7*b^5*c^2*d^10*f^2 + 116*C^3*a^8*b^4*c^3*d^9*f^2 - 148*C^3*a^9*b^3*c
^2*d^10*f^2))/(b^9*f^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*
b^3*f^5) - (((8*(96*C*a^2*b^14*d^11*f^4 + 480*C*a^4*b^12*d^11*f^4 + 960*C*a
^6*b^10*d^11*f^4 + 960*C*a^8*b^8*d^11*f^4 + 480*C*a^10*b^6*d^11*f^4 + 96*C*
a^12*b^4*d^11*f^4 - 64*C*a*b^15*c^3*d^8*f^4 - 320*C*a^3*b^13*c*d^10*f^4 - 6
40*C*a^5*b^11*c*d^10*f^4 - 640*C*a^7*b^9*c*d^10*f^4 - 320*C*a^9*b^7*c*d^10*
f^4 - 64*C*a^11*b^5*c*d^10*f^4 + 96*C*a^2*b^14*c^2*d^9*f^4 - 320*C*a^3*b^13
*c^3*d^8*f^4 + 480*C*a^4*b^12*c^2*d^9*f^4 - 640*C*a^5*b^11*c^3*d^8*f^4 + 96
0*C*a^6*b^10*c^2*d^9*f^4 - 640*C*a^7*b^9*c^3*d^8*f^4 + 960*C*a^8*b^8*c^2*d^
9*f^4 - 320*C*a^9*b^7*c^3*d^8*f^4 + 480*C*a^10*b^6*c^2*d^9*f^4 - 64*C*a^11*
b^5*c^3*d^8*f^4 + 96*C*a^12*b^4*c^2*d^9*f^4 - 64*C*a*b^15*c*d^10*f^4))/(b^9
*f^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^5) - (16*(c
+ d*tan(e + f*x))^(1/2)*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^
3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d
^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2
*f^4))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*
C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*
f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*(32*b^18*d^10*f^4 + 160*a^2*b^
16*d^10*f^4 + 288*a^4*b^14*d^10*f^4 + 160*a^6*b^12*d^10*f^4 - 160*a^8*b^10*
d^10*f^4 - 288*a^10*b^8*d^10*f^4 - 160*a^12*b^6*d^10*f^4 - 32*a^14*b^4*d^10
*f^4 + 48*b^18*c^2*d^8*f^4 + 272*a^2*b^16*c^2*d^8*f^4 + 624*a^4*b^14*c^2*d^
8*f^4 + 720*a^6*b^12*c^2*d^8*f^4 + 400*a^8*b^10*c^2*d^8*f^4 + 48*a^10*b^8*c

^2*d^8*f^4 - 48*a^12*b^6*c^2*d^8*f^4 - 16*a^14*b^4*c^2*d^8*f^4 + 16*a*b^17*c*d^9*f^4 + 112*a^3*b^15*c*d^9*f^4 + 336*a^5*b^13*c*d^9*f^4 + 560*a^7*b^11*c*d^9*f^4 + 560*a^9*b^9*c*d^9*f^4 + 336*a^11*b^7*c*d^9*f^4 + 112*a^13*b^5*c*d^9*f^4 + 16*a^15*b^3*c*d^9*f^4))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(52*C^2*a^3*b^11*d^11*f^2 + 128*C^2*a^5*b^9*d^11*f^2 + 424*C^2*a^7*b^7*d^11*f^2 + 380*C^2*a^9*b^5*d^11*f^2 + 100*C^2*a^11*b^3*d^11*f^2 - 20*C^2*b^14*c^3*d^8*f^2 + 60*C^2*a*b^13*d^11*f^2 + 8*C^2*a^13*b*d^11*f^2 - 4*C^2*a^14*c*d^10*f^2 - 12*C^2*b^14*c*d^10*f^2 + 84*C^2*a*b^13*c^2*d^9*f^2 + 60*C^2*a^2*b^12*c*d^10*f^2 - 116*C^2*a^4*b^10*c*d^10*f^2 - 604*C^2*a^6*b^8*c*d^10*f^2 - 596*C^2*a^8*b^6*c*d^10*f^2 - 220*C^2*a^10*b^4*c*d^10*f^2 - 44*C^2*a^12*b^2*c*d^10*f^2 + 116*C^2*a^2*b^12*c^3*d^8*f^2 + 108*C^2*a^3*b^11*c^2*d^9*f^2 + 216*C^2*a^4*b^10*c^3*d^8*f^2 + 104*C^2*a^5*b^9*c^2*d^9*f^2 + 8*C^2*a^6*b^8*c^3*d^8*f^2 + 248*C^2*a^7*b^7*c^2*d^9*f^2 - 68*C^2*a^8*b^6*c^3*d^8*f^2 + 196*C^2*a^9*b^5*c^2*d^9*f^2 + 4*C^2*a^10*b^4*c^3*d^8*f^2 + 28*C^2*a^11*b^3*c^2*d^9*f^2))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2))*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(2*C^4*b^10*d^12 - C^4*a^10*d^12 + 4*C^4*a^2*b^8*d^12 + 27*C^4*a^4*b^6*d^12 - 15*C^4*a^6*b^4*d^12 - 9*C^4*a^8*b^2*d^12 + C^4*a^10*c^2*d^10 + 4*C^4*b^10*c^2*d^10 + 2*C^4*b^10*c^4*d^8 + 24*C^4*a^2*b^8*c^2*d^10 - 12*C^4*a^2*b^8*c^4*d^8 + 104*C^4*a^3*b^7*c^3*d^9 - 197*C^4*a^4*b^6*c^2*d^10 + 18*C^4*a^4*b^6*c^4*d^8 - 32*C^4*a^5*b^5*c^3*d^9 - 17*C^4*a^6*b^4*c^2*d^10 - 8*C^4*a^7*b^3*c^3*d^9 + 9*C^4*a^8*b^2*c^2*d^10 + 4*C^4*a^9*b*c*d^11 - 40*C^4*a^3*b^7*c*d^11 + 132*C^4*a^5*b^5*c*d^11 + 48*C^4*a^7*b^3*c*d^11))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a

^6*b^2*f^4)))^(1/2)*1i - (((8*(304*C^3*a^3*b^9*d^12*f^2 + 120*C^3*a^5*b^7*d
^12*f^2 - 320*C^3*a^7*b^5*d^12*f^2 - 148*C^3*a^9*b^3*d^12*f^2 + 4*C^3*b^12*
c^3*d^9*f^2 - 4*C^3*a*b^11*d^12*f^2 - 16*C^3*a^11*b*d^12*f^2 + 4*C^3*b^12*c
*d^11*f^2 + 60*C^3*a*b^11*c^2*d^10*f^2 + 64*C^3*a*b^11*c^4*d^8*f^2 - 320*C^
3*a^2*b^10*c*d^11*f^2 + 104*C^3*a^4*b^8*c*d^11*f^2 + 544*C^3*a^6*b^6*c*d^11
*f^2 + 116*C^3*a^8*b^4*c*d^11*f^2 - 16*C^3*a^11*b*c^2*d^10*f^2 - 320*C^3*a^
2*b^10*c^3*d^9*f^2 + 176*C^3*a^3*b^9*c^2*d^10*f^2 - 128*C^3*a^3*b^9*c^4*d^8
*f^2 + 104*C^3*a^4*b^8*c^3*d^9*f^2 - 72*C^3*a^5*b^7*c^2*d^10*f^2 - 192*C^3*
a^5*b^7*c^4*d^8*f^2 + 544*C^3*a^6*b^6*c^3*d^9*f^2 - 320*C^3*a^7*b^5*c^2*d^1
0*f^2 + 116*C^3*a^8*b^4*c^3*d^9*f^2 - 148*C^3*a^9*b^3*c^2*d^10*f^2))/(b^9*f
^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^5) - (((8*(96*
C*a^2*b^14*d^11*f^4 + 480*C*a^4*b^12*d^11*f^4 + 960*C*a^6*b^10*d^11*f^4 + 9
60*C*a^8*b^8*d^11*f^4 + 480*C*a^10*b^6*d^11*f^4 + 96*C*a^12*b^4*d^11*f^4 -
64*C*a*b^15*c^3*d^8*f^4 - 320*C*a^3*b^13*c*d^10*f^4 - 640*C*a^5*b^11*c*d^10
*f^4 - 640*C*a^7*b^9*c*d^10*f^4 - 320*C*a^9*b^7*c*d^10*f^4 - 64*C*a^11*b^5*
c*d^10*f^4 + 96*C*a^2*b^14*c^2*d^9*f^4 - 320*C*a^3*b^13*c^3*d^8*f^4 + 480*C
*a^4*b^12*c^2*d^9*f^4 - 640*C*a^5*b^11*c^3*d^8*f^4 + 960*C*a^6*b^10*c^2*d^9
*f^4 - 640*C*a^7*b^9*c^3*d^8*f^4 + 960*C*a^8*b^8*c^2*d^9*f^4 - 320*C*a^9*b^
7*c^3*d^8*f^4 + 480*C*a^10*b^6*c^2*d^9*f^4 - 64*C*a^11*b^5*c^3*d^8*f^4 + 96
*C*a^12*b^4*c^2*d^9*f^4 - 64*C*a*b^15*c*d^10*f^4))/(b^9*f^5 + a^8*b*f^5 + 4
*a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^5) + (16*(c + d*tan(e + f*x))^(1
/2)*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3
*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16
*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*C^2
*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24
*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4
+ 4*a^6*b^2*f^4)))^(1/2)*(32*b^18*d^10*f^4 + 160*a^2*b^16*d^10*f^4 + 288*a^
4*b^14*d^10*f^4 + 160*a^6*b^12*d^10*f^4 - 160*a^8*b^10*d^10*f^4 - 288*a^10*
b^8*d^10*f^4 - 160*a^12*b^6*d^10*f^4 - 32*a^14*b^4*d^10*f^4 + 48*b^18*c^2*d
^8*f^4 + 272*a^2*b^16*c^2*d^8*f^4 + 624*a^4*b^14*c^2*d^8*f^4 + 720*a^6*b^12
*c^2*d^8*f^4 + 400*a^8*b^10*c^2*d^8*f^4 + 48*a^10*b^8*c^2*d^8*f^4 - 48*a^12
*b^6*c^2*d^8*f^4 - 16*a^14*b^4*c^2*d^8*f^4 + 16*a*b^17*c*d^9*f^4 + 112*a^3*
b^15*c*d^9*f^4 + 336*a^5*b^13*c*d^9*f^4 + 560*a^7*b^11*c*d^9*f^4 + 560*a^9*
b^9*c*d^9*f^4 + 336*a^11*b^7*c*d^9*f^4 + 112*a^13*b^5*c*d^9*f^4 + 16*a^15*b
^3*c*d^9*f^4))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6
*b^3*f^4))*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*
C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f
^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2)
- 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f
^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b
^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(52*C^2*a^
3*b^11*d^11*f^2 + 128*C^2*a^5*b^9*d^11*f^2 + 424*C^2*a^7*b^7*d^11*f^2 + 380
*C^2*a^9*b^5*d^11*f^2 + 100*C^2*a^11*b^3*d^11*f^2 - 20*C^2*b^14*c^3*d^8*f^2
+ 60*C^2*a*b^13*d^11*f^2 + 8*C^2*a^13*b*d^11*f^2 - 4*C^2*a^14*c*d^10*f^2 -
12*C^2*b^14*c*d^10*f^2 + 84*C^2*a*b^13*c^2*d^9*f^2 + 60*C^2*a^2*b^12*c*d^1

0*f^2 - 116*C^2*a^4*b^10*c*d^10*f^2 - 604*C^2*a^6*b^8*c*d^10*f^2 - 596*C^2*
a^8*b^6*c*d^10*f^2 - 220*C^2*a^10*b^4*c*d^10*f^2 - 44*C^2*a^12*b^2*c*d^10*f
^2 + 116*C^2*a^2*b^12*c^3*d^8*f^2 + 108*C^2*a^3*b^11*c^2*d^9*f^2 + 216*C^2*
a^4*b^10*c^3*d^8*f^2 + 104*C^2*a^5*b^9*c^2*d^9*f^2 + 8*C^2*a^6*b^8*c^3*d^8*
f^2 + 248*C^2*a^7*b^7*c^2*d^9*f^2 - 68*C^2*a^8*b^6*c^3*d^8*f^2 + 196*C^2*a^
9*b^5*c^2*d^9*f^2 + 4*C^2*a^10*b^4*c^3*d^8*f^2 + 28*C^2*a^11*b^3*c^2*d^9*f^
2))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*
((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d
*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8
*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*C^2*a^4
*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2
*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*
a^6*b^2*f^4)))^(1/2))*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*
d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2
)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f
^4))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^
2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^
4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)
*(2*C^4*b^10*d^12 - C^4*a^10*d^12 + 4*C^4*a^2*b^8*d^12 + 27*C^4*a^4*b^6*d^1
2 - 15*C^4*a^6*b^4*d^12 - 9*C^4*a^8*b^2*d^12 + C^4*a^10*c^2*d^10 + 4*C^4*b^
10*c^2*d^10 + 2*C^4*b^10*c^4*d^8 + 24*C^4*a^2*b^8*c^2*d^10 - 12*C^4*a^2*b^8
*c^4*d^8 + 104*C^4*a^3*b^7*c^3*d^9 - 197*C^4*a^4*b^6*c^2*d^10 + 18*C^4*a^4*
b^6*c^4*d^8 - 32*C^4*a^5*b^5*c^3*d^9 - 17*C^4*a^6*b^4*c^2*d^10 - 8*C^4*a^7*
b^3*c^3*d^9 + 9*C^4*a^8*b^2*c^2*d^10 + 4*C^4*a^9*b*c*d^11 - 40*C^4*a^3*b^7*
c*d^11 + 132*C^4*a^5*b^5*c*d^11 + 48*C^4*a^7*b^3*c*d^11))/(b^9*f^4 + a^8*b*
f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*((((8*C^2*a^4*c*f^2 +
 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2
*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4
 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*
f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(
a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*
1i)/((((8*(304*C^3*a^3*b^9*d^12*f^2 + 120*C^3*a^5*b^7*d^12*f^2 - 320*C^3*a^
7*b^5*d^12*f^2 - 148*C^3*a^9*b^3*d^12*f^2 + 4*C^3*b^12*c^3*d^9*f^2 - 4*C^3*
a*b^11*d^12*f^2 - 16*C^3*a^11*b*d^12*f^2 + 4*C^3*b^12*c*d^11*f^2 + 60*C^3*a
*b^11*c^2*d^10*f^2 + 64*C^3*a*b^11*c^4*d^8*f^2 - 320*C^3*a^2*b^10*c*d^11*f^
2 + 104*C^3*a^4*b^8*c*d^11*f^2 + 544*C^3*a^6*b^6*c*d^11*f^2 + 116*C^3*a^8*b
^4*c*d^11*f^2 - 16*C^3*a^11*b*c^2*d^10*f^2 - 320*C^3*a^2*b^10*c^3*d^9*f^2 +
 176*C^3*a^3*b^9*c^2*d^10*f^2 - 128*C^3*a^3*b^9*c^4*d^8*f^2 + 104*C^3*a^4*b
^8*c^3*d^9*f^2 - 72*C^3*a^5*b^7*c^2*d^10*f^2 - 192*C^3*a^5*b^7*c^4*d^8*f^2
+ 544*C^3*a^6*b^6*c^3*d^9*f^2 - 320*C^3*a^7*b^5*c^2*d^10*f^2 + 116*C^3*a^8*
b^4*c^3*d^9*f^2 - 148*C^3*a^9*b^3*c^2*d^10*f^2))/(b^9*f^5 + a^8*b*f^5 + 4*a
^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^5) - (((8*(96*C*a^2*b^14*d^11*f^4
+ 480*C*a^4*b^12*d^11*f^4 + 960*C*a^6*b^10*d^11*f^4 + 960*C*a^8*b^8*d^11*f^
4 + 480*C*a^10*b^6*d^11*f^4 + 96*C*a^12*b^4*d^11*f^4 - 64*C*a*b^15*c^3*d^8*
f^4 - 320*C*a^3*b^13*c*d^10*f^4 - 640*C*a^5*b^11*c*d^10*f^4 - 640*C*a^7*b^9

*c*d^10*f^4 - 320*C*a^9*b^7*c*d^10*f^4 - 64*C*a^11*b^5*c*d^10*f^4 + 96*C*a^2*b^14*c^2*d^9*f^4 - 320*C*a^3*b^13*c^3*d^8*f^4 + 480*C*a^4*b^12*c^2*d^9*f^4 - 640*C*a^5*b^11*c^3*d^8*f^4 + 960*C*a^6*b^10*c^2*d^9*f^4 - 640*C*a^7*b^9*c^3*d^8*f^4 + 960*C*a^8*b^8*c^2*d^9*f^4 - 320*C*a^9*b^7*c^3*d^8*f^4 + 480*C*a^10*b^6*c^2*d^9*f^4 - 64*C*a^11*b^5*c^3*d^8*f^4 + 96*C*a^12*b^4*c^2*d^9*f^4 - 64*C*a*b^15*c*d^10*f^4))/(b^9*f^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^5) - (16*(c + d*tan(e + f*x))^(1/2)*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*(32*b^18*d^10*f^4 + 160*a^2*b^16*d^10*f^4 + 288*a^4*b^14*d^10*f^4 + 160*a^6*b^12*d^10*f^4 - 160*a^8*b^10*d^10*f^4 - 288*a^10*b^8*d^10*f^4 - 160*a^12*b^6*d^10*f^4 - 32*a^14*b^4*d^10*f^4 + 48*b^18*c^2*d^8*f^4 + 272*a^2*b^16*c^2*d^8*f^4 + 624*a^4*b^14*c^2*d^8*f^4 + 720*a^6*b^12*c^2*d^8*f^4 + 400*a^8*b^10*c^2*d^8*f^4 + 48*a^10*b^8*c^2*d^8*f^4 - 48*a^12*b^6*c^2*d^8*f^4 - 16*a^14*b^4*c^2*d^8*f^4 + 16*a*b^17*c*d^9*f^4 + 112*a^3*b^15*c*d^9*f^4 + 336*a^5*b^13*c*d^9*f^4 + 560*a^7*b^11*c*d^9*f^4 + 560*a^9*b^9*c*d^9*f^4 + 336*a^11*b^7*c*d^9*f^4 + 112*a^13*b^5*c*d^9*f^4 + 16*a^15*b^3*c*d^9*f^4))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(52*C^2*a^3*b^11*d^11*f^2 + 128*C^2*a^5*b^9*d^11*f^2 + 424*C^2*a^7*b^7*d^11*f^2 + 380*C^2*a^9*b^5*d^11*f^2 + 100*C^2*a^11*b^3*d^11*f^2 - 20*C^2*b^14*c^3*d^8*f^2 + 60*C^2*a*b^13*d^11*f^2 + 8*C^2*a^13*b*d^11*f^2 - 4*C^2*a^14*c*d^10*f^2 - 12*C^2*b^14*c*d^10*f^2 + 84*C^2*a*b^13*c^2*d^9*f^2 + 60*C^2*a^2*b^12*c*d^10*f^2 - 116*C^2*a^4*b^10*c*d^10*f^2 - 604*C^2*a^6*b^8*c*d^10*f^2 - 596*C^2*a^8*b^6*c*d^10*f^2 - 220*C^2*a^10*b^4*c*d^10*f^2 - 44*C^2*a^12*b^2*c*d^10*f^2 + 116*C^2*a^2*b^12*c^3*d^8*f^2 + 108*C^2*a^3*b^11*c^2*d^9*f^2 + 216*C^2*a^4*b^10*c^3*d^8*f^2 + 104*C^2*a^5*b^9*c^2*d^9*f^2 + 8*C^2*a^6*b^8*c^3*d^8*f^2 + 248*C^2*a^7*b^7*c^2*d^9*f^2 - 68*C^2*a^8*b^6*c^3*d^8*f^2 + 196*C^2*a^9*b^5*c^2*d^9*f^2 + 4*C^2*a^10*b^4*c^3*d^8*f^2 + 28*C^2*a^11*b^3*c^2*d^9*f^2))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2))*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b

^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*C^2*a
^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C
^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 +
4*a^6*b^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(2*C^4*b^10*d^12 -
C^4*a^10*d^12 + 4*C^4*a^2*b^8*d^12 + 27*C^4*a^4*b^6*d^12 - 15*C^4*a^6*b^4*d
^12 - 9*C^4*a^8*b^2*d^12 + C^4*a^10*c^2*d^10 + 4*C^4*b^10*c^2*d^10 + 2*C^4*
b^10*c^4*d^8 + 24*C^4*a^2*b^8*c^2*d^10 - 12*C^4*a^2*b^8*c^4*d^8 + 104*C^4*a
^3*b^7*c^3*d^9 - 197*C^4*a^4*b^6*c^2*d^10 + 18*C^4*a^4*b^6*c^4*d^8 - 32*C^4
*a^5*b^5*c^3*d^9 - 17*C^4*a^6*b^4*c^2*d^10 - 8*C^4*a^7*b^3*c^3*d^9 + 9*C^4*
a^8*b^2*c^2*d^10 + 4*C^4*a^9*b*c*d^11 - 40*C^4*a^3*b^7*c*d^11 + 132*C^4*a^5
*b^5*c*d^11 + 48*C^4*a^7*b^3*c*d^11))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4
+ 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 3
2*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c
^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 +
64*a^6*b^2*f^4))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d
*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 +
4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) + (((8*(304*C^3*a^3*
b^9*d^12*f^2 + 120*C^3*a^5*b^7*d^12*f^2 - 320*C^3*a^7*b^5*d^12*f^2 - 148*C^
3*a^9*b^3*d^12*f^2 + 4*C^3*b^12*c^3*d^9*f^2 - 4*C^3*a*b^11*d^12*f^2 - 16*C^
3*a^11*b*d^12*f^2 + 4*C^3*b^12*c*d^11*f^2 + 60*C^3*a*b^11*c^2*d^10*f^2 + 64
*C^3*a*b^11*c^4*d^8*f^2 - 320*C^3*a^2*b^10*c*d^11*f^2 + 104*C^3*a^4*b^8*c*d
^11*f^2 + 544*C^3*a^6*b^6*c*d^11*f^2 + 116*C^3*a^8*b^4*c*d^11*f^2 - 16*C^3*
a^11*b*c^2*d^10*f^2 - 320*C^3*a^2*b^10*c^3*d^9*f^2 + 176*C^3*a^3*b^9*c^2*d^
10*f^2 - 128*C^3*a^3*b^9*c^4*d^8*f^2 + 104*C^3*a^4*b^8*c^3*d^9*f^2 - 72*C^3
*a^5*b^7*c^2*d^10*f^2 - 192*C^3*a^5*b^7*c^4*d^8*f^2 + 544*C^3*a^6*b^6*c^3*d
^9*f^2 - 320*C^3*a^7*b^5*c^2*d^10*f^2 + 116*C^3*a^8*b^4*c^3*d^9*f^2 - 148*C
^3*a^9*b^3*c^2*d^10*f^2))/(b^9*f^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a^4*b^5*
f^5 + 4*a^6*b^3*f^5) - (((8*(96*C*a^2*b^14*d^11*f^4 + 480*C*a^4*b^12*d^11*f
^4 + 960*C*a^6*b^10*d^11*f^4 + 960*C*a^8*b^8*d^11*f^4 + 480*C*a^10*b^6*d^11
*f^4 + 96*C*a^12*b^4*d^11*f^4 - 64*C*a*b^15*c^3*d^8*f^4 - 320*C*a^3*b^13*c*
d^10*f^4 - 640*C*a^5*b^11*c*d^10*f^4 - 640*C*a^7*b^9*c*d^10*f^4 - 320*C*a^9
*b^7*c*d^10*f^4 - 64*C*a^11*b^5*c*d^10*f^4 + 96*C*a^2*b^14*c^2*d^9*f^4 - 32
0*C*a^3*b^13*c^3*d^8*f^4 + 480*C*a^4*b^12*c^2*d^9*f^4 - 640*C*a^5*b^11*c^3*
d^8*f^4 + 960*C*a^6*b^10*c^2*d^9*f^4 - 640*C*a^7*b^9*c^3*d^8*f^4 + 960*C*a^
8*b^8*c^2*d^9*f^4 - 320*C*a^9*b^7*c^3*d^8*f^4 + 480*C*a^10*b^6*c^2*d^9*f^4
- 64*C*a^11*b^5*c^3*d^8*f^4 + 96*C*a^12*b^4*c^2*d^9*f^4 - 64*C*a*b^15*c*d^1
0*f^4))/(b^9*f^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^
5) + (16*(c + d*tan(e + f*x))^(1/2)*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 -
32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4
*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4
+ 64*a^6*b^2*f^4))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3
*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4
+ 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*(32*b^18*d^10*f^4
+ 160*a^2*b^16*d^10*f^4 + 288*a^4*b^14*d^10*f^4 + 160*a^6*b^12*d^10*f^4 - 1
60*a^8*b^10*d^10*f^4 - 288*a^10*b^8*d^10*f^4 - 160*a^12*b^6*d^10*f^4 - 32*a

^14*b^4*d^10*f^4 + 48*b^18*c^2*d^8*f^4 + 272*a^2*b^16*c^2*d^8*f^4 + 624*a^4
*b^14*c^2*d^8*f^4 + 720*a^6*b^12*c^2*d^8*f^4 + 400*a^8*b^10*c^2*d^8*f^4 + 4
8*a^10*b^8*c^2*d^8*f^4 - 48*a^12*b^6*c^2*d^8*f^4 - 16*a^14*b^4*c^2*d^8*f^4
+ 16*a*b^17*c*d^9*f^4 + 112*a^3*b^15*c*d^9*f^4 + 336*a^5*b^13*c*d^9*f^4 + 5
60*a^7*b^11*c*d^9*f^4 + 560*a^9*b^9*c*d^9*f^4 + 336*a^11*b^7*c*d^9*f^4 + 11
2*a^13*b^5*c*d^9*f^4 + 16*a^15*b^3*c*d^9*f^4))/(b^9*f^4 + a^8*b*f^4 + 4*a^2
*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*
c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4
 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b
^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^
2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b
^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(c +
d*tan(e + f*x))^(1/2)*(52*C^2*a^3*b^11*d^11*f^2 + 128*C^2*a^5*b^9*d^11*f^2
+ 424*C^2*a^7*b^7*d^11*f^2 + 380*C^2*a^9*b^5*d^11*f^2 + 100*C^2*a^11*b^3*d^
11*f^2 - 20*C^2*b^14*c^3*d^8*f^2 + 60*C^2*a*b^13*d^11*f^2 + 8*C^2*a^13*b*d^
11*f^2 - 4*C^2*a^14*c*d^10*f^2 - 12*C^2*b^14*c*d^10*f^2 + 84*C^2*a*b^13*c^2
*d^9*f^2 + 60*C^2*a^2*b^12*c*d^10*f^2 - 116*C^2*a^4*b^10*c*d^10*f^2 - 604*C
^2*a^6*b^8*c*d^10*f^2 - 596*C^2*a^8*b^6*c*d^10*f^2 - 220*C^2*a^10*b^4*c*d^1
0*f^2 - 44*C^2*a^12*b^2*c*d^10*f^2 + 116*C^2*a^2*b^12*c^3*d^8*f^2 + 108*C^2
*a^3*b^11*c^2*d^9*f^2 + 216*C^2*a^4*b^10*c^3*d^8*f^2 + 104*C^2*a^5*b^9*c^2*
d^9*f^2 + 8*C^2*a^6*b^8*c^3*d^8*f^2 + 248*C^2*a^7*b^7*c^2*d^9*f^2 - 68*C^2*
a^8*b^6*c^3*d^8*f^2 + 196*C^2*a^9*b^5*c^2*d^9*f^2 + 4*C^2*a^10*b^4*c^3*d^8*
f^2 + 28*C^2*a^11*b^3*c^2*d^9*f^2))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 +
6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*
C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2
 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64
*a^6*b^2*f^4))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f
^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*
a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2))*((((8*C^2*a^4*c*f^2 +
 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2
*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4
 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*
f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(
a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)
- (16*(c + d*tan(e + f*x))^(1/2)*(2*C^4*b^10*d^12 - C^4*a^10*d^12 + 4*C^4*a
^2*b^8*d^12 + 27*C^4*a^4*b^6*d^12 - 15*C^4*a^6*b^4*d^12 - 9*C^4*a^8*b^2*d^1
2 + C^4*a^10*c^2*d^10 + 4*C^4*b^10*c^2*d^10 + 2*C^4*b^10*c^4*d^8 + 24*C^4*a
^2*b^8*c^2*d^10 - 12*C^4*a^2*b^8*c^4*d^8 + 104*C^4*a^3*b^7*c^3*d^9 - 197*C^
4*a^4*b^6*c^2*d^10 + 18*C^4*a^4*b^6*c^4*d^8 - 32*C^4*a^5*b^5*c^3*d^9 - 17*C
^4*a^6*b^4*c^2*d^10 - 8*C^4*a^7*b^3*c^3*d^9 + 9*C^4*a^8*b^2*c^2*d^10 + 4*C^
4*a^9*b*c*d^11 - 40*C^4*a^3*b^7*c*d^11 + 132*C^4*a^5*b^5*c*d^11 + 48*C^4*a^
7*b^3*c*d^11))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6
*b^3*f^4))*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*
C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f
^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2)

- 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f
^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b
^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(C^5*a^8*d^13 + 10*C^5*a^2*b^6*d^13 +
27*C^5*a^4*b^4*d^13 + 10*C^5*a^6*b^2*d^13 + C^5*a^8*c^2*d^11 + 36*C^5*a^2*
b^6*c^2*d^11 + 26*C^5*a^2*b^6*c^4*d^9 - 40*C^5*a^3*b^5*c^3*d^10 + 29*C^5*a^
4*b^4*c^2*d^11 + 2*C^5*a^4*b^4*c^4*d^9 - 8*C^5*a^5*b^3*c^3*d^10 + 10*C^5*a^
6*b^2*c^2*d^11 - 8*C^5*a*b^7*c*d^12 - 16*C^5*a*b^7*c^3*d^10 - 8*C^5*a*b^7*c
^5*d^8 - 40*C^5*a^3*b^5*c*d^12 - 8*C^5*a^5*b^3*c*d^12))/(b^9*f^5 + a^8*b*f^
5 + 4*a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^5)))*((((8*C^2*a^4*c*f^2 +
8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*
c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4
+ 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f
^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(a
^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*2
i + atan(((((8*(156*B^3*a^2*b^9*d^12*f^2 - 16*B^3*a^4*b^7*d^12*f^2 - 120*B^
3*a^6*b^5*d^12*f^2 + 48*B^3*a^8*b^3*d^12*f^2 + 12*B^3*b^11*c^2*d^10*f^2 + 1
2*B^3*b^11*c^4*d^8*f^2 - 4*B^3*a^10*b*d^12*f^2 - 124*B^3*a*b^10*c*d^11*f^2
- 124*B^3*a*b^10*c^3*d^9*f^2 + 224*B^3*a^3*b^8*c*d^11*f^2 + 200*B^3*a^5*b^6
*c*d^11*f^2 - 128*B^3*a^7*b^4*c*d^11*f^2 + 20*B^3*a^9*b^2*c*d^11*f^2 - 4*B^
3*a^10*b*c^2*d^10*f^2 + 44*B^3*a^2*b^9*c^2*d^10*f^2 - 112*B^3*a^2*b^9*c^4*d
^8*f^2 + 224*B^3*a^3*b^8*c^3*d^9*f^2 - 40*B^3*a^4*b^7*c^2*d^10*f^2 - 24*B^3
*a^4*b^7*c^4*d^8*f^2 + 200*B^3*a^5*b^6*c^3*d^9*f^2 - 40*B^3*a^6*b^5*c^2*d^1
0*f^2 + 80*B^3*a^6*b^5*c^4*d^8*f^2 - 128*B^3*a^7*b^4*c^3*d^9*f^2 + 28*B^3*a
^8*b^3*c^2*d^10*f^2 - 20*B^3*a^8*b^3*c^4*d^8*f^2 + 20*B^3*a^9*b^2*c^3*d^9*f
^2))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) +
(((8*(80*B*a*b^14*d^11*f^4 - 48*B*b^15*c*d^10*f^4 + 384*B*a^3*b^12*d^11*f^4
+ 720*B*a^5*b^10*d^11*f^4 + 640*B*a^7*b^8*d^11*f^4 + 240*B*a^9*b^6*d^11*f^
4 - 16*B*a^13*b^2*d^11*f^4 - 48*B*b^15*c^3*d^8*f^4 + 80*B*a*b^14*c^2*d^9*f^
4 - 224*B*a^2*b^13*c*d^10*f^4 - 400*B*a^4*b^11*c*d^10*f^4 - 320*B*a^6*b^9*c
*d^10*f^4 - 80*B*a^8*b^7*c*d^10*f^4 + 32*B*a^10*b^5*c*d^10*f^4 + 16*B*a^12*
b^3*c*d^10*f^4 - 224*B*a^2*b^13*c^3*d^8*f^4 + 384*B*a^3*b^12*c^2*d^9*f^4 -
400*B*a^4*b^11*c^3*d^8*f^4 + 720*B*a^5*b^10*c^2*d^9*f^4 - 320*B*a^6*b^9*c^3
*d^8*f^4 + 640*B*a^7*b^8*c^2*d^9*f^4 - 80*B*a^8*b^7*c^3*d^8*f^4 + 240*B*a^9
*b^6*c^2*d^9*f^4 + 32*B*a^10*b^5*c^3*d^8*f^4 + 16*B*a^12*b^3*c^3*d^8*f^4 -
16*B*a^13*b^2*c^2*d^9*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*
f^5 + 4*a^6*b^2*f^5) - (16*(c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^4*c*f^2 +
8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2
*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4
+ 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*
f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(
a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*
(32*b^17*d^10*f^4 + 160*a^2*b^15*d^10*f^4 + 288*a^4*b^13*d^10*f^4 + 160*a^6
*b^11*d^10*f^4 - 160*a^8*b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b^
5*d^10*f^4 - 32*a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*
d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^9

*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14*
b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336*a^5*b^
12*c*d^9*f^4 + 560*a^7*b^10*c*d^9*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*a^11*b^
6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2*c*d^9*f^4))/(a^8*f^4 + b
^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*B^2*a^4*c*f^
2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*
b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*
f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4
*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(1
6*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/
2) - (16*(c + d*tan(e + f*x))^(1/2)*(44*B^2*a^9*b^4*d^11*f^2 - 168*B^2*a^5*
b^8*d^11*f^2 - 40*B^2*a^7*b^6*d^11*f^2 - 20*B^2*a^3*b^10*d^11*f^2 - 4*B^2*a
^11*b^2*d^11*f^2 - 36*B^2*b^13*c^3*d^8*f^2 + 60*B^2*a*b^12*d^11*f^2 - 12*B^
2*b^13*c*d^10*f^2 + 4*B^2*a^12*b*c*d^10*f^2 + 100*B^2*a*b^12*c^2*d^9*f^2 +
120*B^2*a^2*b^11*c*d^10*f^2 + 156*B^2*a^4*b^9*c*d^10*f^2 - 112*B^2*a^6*b^7*
c*d^10*f^2 - 148*B^2*a^8*b^5*c*d^10*f^2 - 8*B^2*a^10*b^3*c*d^10*f^2 + 68*B^
2*a^2*b^11*c^3*d^8*f^2 + 124*B^2*a^3*b^10*c^2*d^9*f^2 + 184*B^2*a^4*b^9*c^3
*d^8*f^2 + 8*B^2*a^5*b^8*c^2*d^9*f^2 + 40*B^2*a^6*b^7*c^3*d^8*f^2 + 24*B^2*
a^7*b^6*c^2*d^9*f^2 - 20*B^2*a^8*b^5*c^3*d^8*f^2 + 20*B^2*a^9*b^4*c^2*d^9*f
^2 + 20*B^2*a^10*b^3*c^3*d^8*f^2 - 20*B^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 +
b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*B^2*a^4*c*
f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^
2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^
6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b
^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/
(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(
1/2))*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a
^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 +
16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*B
^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 -
24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^
4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(2*B^4*b^9*d^12
- 5*B^4*a^2*b^7*d^12 + 17*B^4*a^4*b^5*d^12 - 7*B^4*a^6*b^3*d^12 + 6*B^4*b^
9*c^4*d^8 + B^4*a^8*b*d^12 + 77*B^4*a^2*b^7*c^2*d^10 - 8*B^4*a^2*b^7*c^4*d^
8 + 60*B^4*a^3*b^6*c^3*d^9 - 87*B^4*a^4*b^5*c^2*d^10 + 14*B^4*a^4*b^5*c^4*d
^8 - 36*B^4*a^5*b^4*c^3*d^9 + 27*B^4*a^6*b^3*c^2*d^10 - 4*B^4*a^6*b^3*c^4*d
^8 + 4*B^4*a^7*b^2*c^3*d^9 + 12*B^4*a*b^8*c*d^11 - 28*B^4*a*b^8*c^3*d^9 - 6
4*B^4*a^3*b^6*c*d^11 + 44*B^4*a^5*b^4*c*d^11 - 8*B^4*a^7*b^2*c*d^11 - B^4*a
^8*b*c^2*d^10))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*
b^2*f^4))*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B
^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^
4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) +
4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^
2 - 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^
4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*1i - (((8*(156*B^3*a^2*b^9*d^12*f^2 - 16*B^3

*a^4*b^7*d^12*f^2 - 120*B^3*a^6*b^5*d^12*f^2 + 48*B^3*a^8*b^3*d^12*f^2 + 12 *B^3*b^11*c^2*d^10*f^2 + 12*B^3*b^11*c^4*d^8*f^2 - 4*B^3*a^10*b*d^12*f^2 - 124*B^3*a*b^10*c*d^11*f^2 - 124*B^3*a*b^10*c^3*d^9*f^2 + 224*B^3*a^3*b^8*c* d^11*f^2 + 200*B^3*a^5*b^6*c*d^11*f^2 - 128*B^3*a^7*b^4*c*d^11*f^2 + 20*B^3 *a^9*b^2*c*d^11*f^2 - 4*B^3*a^10*b*c^2*d^10*f^2 + 44*B^3*a^2*b^9*c^2*d^10*f ^2 - 112*B^3*a^2*b^9*c^4*d^8*f^2 + 224*B^3*a^3*b^8*c^3*d^9*f^2 - 40*B^3*a^4 *b^7*c^2*d^10*f^2 - 24*B^3*a^4*b^7*c^4*d^8*f^2 + 200*B^3*a^5*b^6*c^3*d^9*f^ 2 - 40*B^3*a^6*b^5*c^2*d^10*f^2 + 80*B^3*a^6*b^5*c^4*d^8*f^2 - 128*B^3*a^7* b^4*c^3*d^9*f^2 + 28*B^3*a^8*b^3*c^2*d^10*f^2 - 20*B^3*a^8*b^3*c^4*d^8*f^2 + 20*B^3*a^9*b^2*c^3*d^9*f^2))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b ^4*f^5 + 4*a^6*b^2*f^5) + (((8*(80*B*a*b^14*d^11*f^4 - 48*B*b^15*c*d^10*f^4 + 384*B*a^3*b^12*d^11*f^4 + 720*B*a^5*b^10*d^11*f^4 + 640*B*a^7*b^8*d^11*f ^4 + 240*B*a^9*b^6*d^11*f^4 - 16*B*a^13*b^2*d^11*f^4 - 48*B*b^15*c^3*d^8*f^ 4 + 80*B*a*b^14*c^2*d^9*f^4 - 224*B*a^2*b^13*c*d^10*f^4 - 400*B*a^4*b^11*c* d^10*f^4 - 320*B*a^6*b^9*c*d^10*f^4 - 80*B*a^8*b^7*c*d^10*f^4 + 32*B*a^10*b ^5*c*d^10*f^4 + 16*B*a^12*b^3*c*d^10*f^4 - 224*B*a^2*b^13*c^3*d^8*f^4 + 384 *B*a^3*b^12*c^2*d^9*f^4 - 400*B*a^4*b^11*c^3*d^8*f^4 + 720*B*a^5*b^10*c^2*d ^9*f^4 - 320*B*a^6*b^9*c^3*d^8*f^4 + 640*B*a^7*b^8*c^2*d^9*f^4 - 80*B*a^8*b ^7*c^3*d^8*f^4 + 240*B*a^9*b^6*c^2*d^9*f^4 + 32*B*a^10*b^5*c^3*d^8*f^4 + 16 *B*a^12*b^3*c^3*d^8*f^4 - 16*B*a^13*b^2*c^2*d^9*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (16*(c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^ 3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 1 6*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*B^ 2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 2 4*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*(32*b^17*d^10*f^4 + 160*a^2*b^15*d^10*f^4 + 288*a ^4*b^13*d^10*f^4 + 160*a^6*b^11*d^10*f^4 - 160*a^8*b^9*d^10*f^4 - 288*a^10* b^7*d^10*f^4 - 160*a^12*b^5*d^10*f^4 - 32*a^14*b^3*d^10*f^4 + 48*b^17*c^2*d ^8*f^4 + 272*a^2*b^15*c^2*d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^4 + 720*a^6*b^11 *c^2*d^8*f^4 + 400*a^8*b^9*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^8*f^4 - 48*a^12* b^5*c^2*d^8*f^4 - 16*a^14*b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9*f^4 + 112*a^3*b ^14*c*d^9*f^4 + 336*a^5*b^12*c*d^9*f^4 + 560*a^7*b^10*c*d^9*f^4 + 560*a^9*b ^8*c*d^9*f^4 + 336*a^11*b^6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*f^4 + 16*a^15*b^ 2*c*d^9*f^4))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^ 2*f^4))*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2 *a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4 *B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4* f^4 + 4*a^6*b^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(44*B^2*a^9*b ^4*d^11*f^2 - 168*B^2*a^5*b^8*d^11*f^2 - 40*B^2*a^7*b^6*d^11*f^2 - 20*B^2*a ^3*b^10*d^11*f^2 - 4*B^2*a^11*b^2*d^11*f^2 - 36*B^2*b^13*c^3*d^8*f^2 + 60*B ^2*a*b^12*d^11*f^2 - 12*B^2*b^13*c*d^10*f^2 + 4*B^2*a^12*b*c*d^10*f^2 + 100 *B^2*a*b^12*c^2*d^9*f^2 + 120*B^2*a^2*b^11*c*d^10*f^2 + 156*B^2*a^4*b^9*c*d

^10*f^2 - 112*B^2*a^6*b^7*c*d^10*f^2 - 148*B^2*a^8*b^5*c*d^10*f^2 - 8*B^2*a^10*b^3*c*d^10*f^2 + 68*B^2*a^2*b^11*c^3*d^8*f^2 + 124*B^2*a^3*b^10*c^2*d^9*f^2 + 184*B^2*a^4*b^9*c^3*d^8*f^2 + 8*B^2*a^5*b^8*c^2*d^9*f^2 + 40*B^2*a^6*b^7*c^3*d^8*f^2 + 24*B^2*a^7*b^6*c^2*d^9*f^2 - 20*B^2*a^8*b^5*c^3*d^8*f^2 + 20*B^2*a^9*b^4*c^2*d^9*f^2 + 20*B^2*a^10*b^3*c^3*d^8*f^2 - 20*B^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2))*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(2*B^4*b^9*d^12 - 5*B^4*a^2*b^7*d^12 + 17*B^4*a^4*b^5*d^12 - 7*B^4*a^6*b^3*d^12 + 6*B^4*b^9*c^4*d^8 + B^4*a^8*b*d^12 + 77*B^4*a^2*b^7*c^2*d^10 - 8*B^4*a^2*b^7*c^4*d^8 + 60*B^4*a^3*b^6*c^3*d^9 - 87*B^4*a^4*b^5*c^2*d^10 + 14*B^4*a^4*b^5*c^4*d^8 - 36*B^4*a^5*b^4*c^3*d^9 + 27*B^4*a^6*b^3*c^2*d^10 - 4*B^4*a^6*b^3*c^4*d^8 + 4*B^4*a^7*b^2*c^3*d^9 + 12*B^4*a*b^8*c*d^11 - 28*B^4*a*b^8*c^3*d^9 - 64*B^4*a^3*b^6*c*d^11 + 44*B^4*a^5*b^4*c*d^11 - 8*B^4*a^7*b^2*c*d^11 - B^4*a^8*b*c^2*d^10))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*1i)/((((8*(156*B^3*a^2*b^9*d^12*f^2 - 16*B^3*a^4*b^7*d^12*f^2 - 120*B^3*a^6*b^5*d^12*f^2 + 48*B^3*a^8*b^3*d^12*f^2 + 12*B^3*b^11*c^2*d^10*f^2 + 12*B^3*b^11*c^4*d^8*f^2 - 4*B^3*a^10*b*d^12*f^2 - 124*B^3*a*b^10*c*d^11*f^2 - 124*B^3*a*b^10*c^3*d^9*f^2 + 224*B^3*a^3*b^8*c*d^11*f^2 + 200*B^3*a^5*b^6*c*d^11*f^2 - 128*B^3*a^7*b^4*c*d^11*f^2 + 20*B^3*a^9*b^2*c*d^11*f^2 - 4*B^3*a^10*b*c^2*d^10*f^2 + 44*B^3*a^2*b^9*c^2*d^10*f^2 - 112*B^3*a^2*b^9*c^4*d^8*f^2 + 224*B^3*a^3*b^8*c^3*d^9*f^2 - 40*B^3*a^4*b^7*c^2*d^10*f^2 - 24*B^3*a^4*b^7*c^4*d^8*f^2 + 200*B^3*a^5*b^6*c^3*d^9*f^2 - 40*B^3*a^6*b^5*c^2*d^10*f^2 + 80*B^3*a^6*b^5*c^4*d^8*f^2 - 128*B^3*a^7*b^4*c^3*d^9*f^2 + 28*B^3*a^8*b^3*c^2*d^10*f^2 - 20*B^3*a^8*b^3*c^4*d^8*f^2 + 20*B^3*a^9*b^2*c^3*d^9*f^2))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((8*(80*B*a*b^14*d^11*f^4 - 48*B*b^15*c*d^10*f^4 + 384*B*a^3*b^12*d^11*f^4 + 720*B*a^5*b^10*d^11*f^4 + 640*B*a^7*b^8*d^11*f^4 + 240*B*a^9*b^6*d^11*f^4 - 16*B*a^13*b^2*d^11*f^4 - 48*B*b^15*c^3*d^8*f^4 + 80*B*a*b^14*c^2*d^9*f^4 - 224*B*a^2*b^13*c*d^10*f^4 - 400*B*a^4*b^11*c*d^10*f^4 - 320*B*a^6*b^9*c*d^10*f^4 - 80*B*a^8*b^

7*c*d^10*f^4 + 32*B*a^10*b^5*c*d^10*f^4 + 16*B*a^12*b^3*c*d^10*f^4 - 224*B*
a^2*b^13*c^3*d^8*f^4 + 384*B*a^3*b^12*c^2*d^9*f^4 - 400*B*a^4*b^11*c^3*d^8*
f^4 + 720*B*a^5*b^10*c^2*d^9*f^4 - 320*B*a^6*b^9*c^3*d^8*f^4 + 640*B*a^7*b^
8*c^2*d^9*f^4 - 80*B*a^8*b^7*c^3*d^8*f^4 + 240*B*a^9*b^6*c^2*d^9*f^4 + 32*B
*a^10*b^5*c^3*d^8*f^4 + 16*B*a^12*b^3*c^3*d^8*f^4 - 16*B*a^13*b^2*c^2*d^9*f
^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) -
(16*(c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B
^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2
+ B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*
a^6*b^2*f^4))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^
2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a
^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*(32*b^17*d^10*f^4 + 160
*a^2*b^15*d^10*f^4 + 288*a^4*b^13*d^10*f^4 + 160*a^6*b^11*d^10*f^4 - 160*a^
8*b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b^5*d^10*f^4 - 32*a^14*b^
3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*d^8*f^4 + 624*a^4*b^13*
c^2*d^8*f^4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^9*c^2*d^8*f^4 + 48*a^10*
b^7*c^2*d^8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14*b^3*c^2*d^8*f^4 + 16*a*
b^16*c*d^9*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336*a^5*b^12*c*d^9*f^4 + 560*a^7*
b^10*c*d^9*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*a^11*b^6*c*d^9*f^4 + 112*a^13*
b^4*c*d^9*f^4 + 16*a^15*b^2*c*d^9*f^4))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4
+ 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 3
2*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c
^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 +
64*a^6*b^2*f^4))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d
*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 +
4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(c + d*tan(e +
f*x))^(1/2)*(44*B^2*a^9*b^4*d^11*f^2 - 168*B^2*a^5*b^8*d^11*f^2 - 40*B^2*a
^7*b^6*d^11*f^2 - 20*B^2*a^3*b^10*d^11*f^2 - 4*B^2*a^11*b^2*d^11*f^2 - 36*B
^2*b^13*c^3*d^8*f^2 + 60*B^2*a*b^12*d^11*f^2 - 12*B^2*b^13*c*d^10*f^2 + 4*B
^2*a^12*b*c*d^10*f^2 + 100*B^2*a*b^12*c^2*d^9*f^2 + 120*B^2*a^2*b^11*c*d^10
*f^2 + 156*B^2*a^4*b^9*c*d^10*f^2 - 112*B^2*a^6*b^7*c*d^10*f^2 - 148*B^2*a^
8*b^5*c*d^10*f^2 - 8*B^2*a^10*b^3*c*d^10*f^2 + 68*B^2*a^2*b^11*c^3*d^8*f^2
+ 124*B^2*a^3*b^10*c^2*d^9*f^2 + 184*B^2*a^4*b^9*c^3*d^8*f^2 + 8*B^2*a^5*b^
8*c^2*d^9*f^2 + 40*B^2*a^6*b^7*c^3*d^8*f^2 + 24*B^2*a^7*b^6*c^2*d^9*f^2 - 2
0*B^2*a^8*b^5*c^3*d^8*f^2 + 20*B^2*a^9*b^4*c^2*d^9*f^2 + 20*B^2*a^10*b^3*c^
3*d^8*f^2 - 20*B^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^
4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 -
32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4
*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4
+ 64*a^6*b^2*f^4))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3
*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4
+ 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2))*((((8*B^2*a^4*c*f
^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2
*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6
*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^

4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(
16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1
/2) - (16*(c + d*tan(e + f*x))^(1/2)*(2*B^4*b^9*d^12 - 5*B^4*a^2*b^7*d^12 +
17*B^4*a^4*b^5*d^12 - 7*B^4*a^6*b^3*d^12 + 6*B^4*b^9*c^4*d^8 + B^4*a^8*b*d
^12 + 77*B^4*a^2*b^7*c^2*d^10 - 8*B^4*a^2*b^7*c^4*d^8 + 60*B^4*a^3*b^6*c^3*
d^9 - 87*B^4*a^4*b^5*c^2*d^10 + 14*B^4*a^4*b^5*c^4*d^8 - 36*B^4*a^5*b^4*c^3
*d^9 + 27*B^4*a^6*b^3*c^2*d^10 - 4*B^4*a^6*b^3*c^4*d^8 + 4*B^4*a^7*b^2*c^3*
d^9 + 12*B^4*a*b^8*c*d^11 - 28*B^4*a*b^8*c^3*d^9 - 64*B^4*a^3*b^6*c*d^11 +
44*B^4*a^5*b^4*c*d^11 - 8*B^4*a^7*b^2*c*d^11 - B^4*a^8*b*c^2*d^10))/(a^8*f^
4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*B^2*a^4
*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2
*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2
*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^
2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^
2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))
)^(1/2) + (((8*(156*B^3*a^2*b^9*d^12*f^2 - 16*B^3*a^4*b^7*d^12*f^2 - 120*B^
3*a^6*b^5*d^12*f^2 + 48*B^3*a^8*b^3*d^12*f^2 + 12*B^3*b^11*c^2*d^10*f^2 + 1
2*B^3*b^11*c^4*d^8*f^2 - 4*B^3*a^10*b*d^12*f^2 - 124*B^3*a*b^10*c*d^11*f^2
- 124*B^3*a*b^10*c^3*d^9*f^2 + 224*B^3*a^3*b^8*c*d^11*f^2 + 200*B^3*a^5*b^6
*c*d^11*f^2 - 128*B^3*a^7*b^4*c*d^11*f^2 + 20*B^3*a^9*b^2*c*d^11*f^2 - 4*B^
3*a^10*b*c^2*d^10*f^2 + 44*B^3*a^2*b^9*c^2*d^10*f^2 - 112*B^3*a^2*b^9*c^4*d
^8*f^2 + 224*B^3*a^3*b^8*c^3*d^9*f^2 - 40*B^3*a^4*b^7*c^2*d^10*f^2 - 24*B^3
*a^4*b^7*c^4*d^8*f^2 + 200*B^3*a^5*b^6*c^3*d^9*f^2 - 40*B^3*a^6*b^5*c^2*d^1
0*f^2 + 80*B^3*a^6*b^5*c^4*d^8*f^2 - 128*B^3*a^7*b^4*c^3*d^9*f^2 + 28*B^3*a
^8*b^3*c^2*d^10*f^2 - 20*B^3*a^8*b^3*c^4*d^8*f^2 + 20*B^3*a^9*b^2*c^3*d^9*f
^2))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) +
(((8*(80*B*a*b^14*d^11*f^4 - 48*B*b^15*c*d^10*f^4 + 384*B*a^3*b^12*d^11*f^4
+ 720*B*a^5*b^10*d^11*f^4 + 640*B*a^7*b^8*d^11*f^4 + 240*B*a^9*b^6*d^11*f^
4 - 16*B*a^13*b^2*d^11*f^4 - 48*B*b^15*c^3*d^8*f^4 + 80*B*a*b^14*c^2*d^9*f^
4 - 224*B*a^2*b^13*c*d^10*f^4 - 400*B*a^4*b^11*c*d^10*f^4 - 320*B*a^6*b^9*c
*d^10*f^4 - 80*B*a^8*b^7*c*d^10*f^4 + 32*B*a^10*b^5*c*d^10*f^4 + 16*B*a^12*
b^3*c*d^10*f^4 - 224*B*a^2*b^13*c^3*d^8*f^4 + 384*B*a^3*b^12*c^2*d^9*f^4 -
400*B*a^4*b^11*c^3*d^8*f^4 + 720*B*a^5*b^10*c^2*d^9*f^4 - 320*B*a^6*b^9*c^3
*d^8*f^4 + 640*B*a^7*b^8*c^2*d^9*f^4 - 80*B*a^8*b^7*c^3*d^8*f^4 + 240*B*a^9
*b^6*c^2*d^9*f^4 + 32*B*a^10*b^5*c^3*d^8*f^4 + 16*B*a^12*b^3*c^3*d^8*f^4 -
16*B*a^13*b^2*c^2*d^9*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*
f^5 + 4*a^6*b^2*f^5) + (16*(c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^4*c*f^2 +
8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2
*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4
+ 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*
f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(
a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*
(32*b^17*d^10*f^4 + 160*a^2*b^15*d^10*f^4 + 288*a^4*b^13*d^10*f^4 + 160*a^6
*b^11*d^10*f^4 - 160*a^8*b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b^
5*d^10*f^4 - 32*a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*

d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^9
*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14*
b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336*a^5*b^
12*c*d^9*f^4 + 560*a^7*b^10*c*d^9*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*a^11*b^
6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2*c*d^9*f^4))/(a^8*f^4 + b
^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*B^2*a^4*c*f^
2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*
b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*
f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4
*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(1
6*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/
2) + (16*(c + d*tan(e + f*x))^(1/2)*(44*B^2*a^9*b^4*d^11*f^2 - 168*B^2*a^5*
b^8*d^11*f^2 - 40*B^2*a^7*b^6*d^11*f^2 - 20*B^2*a^3*b^10*d^11*f^2 - 4*B^2*a
^11*b^2*d^11*f^2 - 36*B^2*b^13*c^3*d^8*f^2 + 60*B^2*a*b^12*d^11*f^2 - 12*B^
2*b^13*c*d^10*f^2 + 4*B^2*a^12*b*c*d^10*f^2 + 100*B^2*a*b^12*c^2*d^9*f^2 +
120*B^2*a^2*b^11*c*d^10*f^2 + 156*B^2*a^4*b^9*c*d^10*f^2 - 112*B^2*a^6*b^7*
c*d^10*f^2 - 148*B^2*a^8*b^5*c*d^10*f^2 - 8*B^2*a^10*b^3*c*d^10*f^2 + 68*B^
2*a^2*b^11*c^3*d^8*f^2 + 124*B^2*a^3*b^10*c^2*d^9*f^2 + 184*B^2*a^4*b^9*c^3
*d^8*f^2 + 8*B^2*a^5*b^8*c^2*d^9*f^2 + 40*B^2*a^6*b^7*c^3*d^8*f^2 + 24*B^2*
a^7*b^6*c^2*d^9*f^2 - 20*B^2*a^8*b^5*c^3*d^8*f^2 + 20*B^2*a^9*b^4*c^2*d^9*f
^2 + 20*B^2*a^10*b^3*c^3*d^8*f^2 - 20*B^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 +
b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*B^2*a^4*c*
f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^
2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^
6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b
^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/
(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(
1/2))*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a
^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 +
16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*B
^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 -
24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^
4 + 4*a^6*b^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(2*B^4*b^9*d^12
 - 5*B^4*a^2*b^7*d^12 + 17*B^4*a^4*b^5*d^12 - 7*B^4*a^6*b^3*d^12 + 6*B^4*b^
9*c^4*d^8 + B^4*a^8*b*d^12 + 77*B^4*a^2*b^7*c^2*d^10 - 8*B^4*a^2*b^7*c^4*d^
8 + 60*B^4*a^3*b^6*c^3*d^9 - 87*B^4*a^4*b^5*c^2*d^10 + 14*B^4*a^4*b^5*c^4*d
^8 - 36*B^4*a^5*b^4*c^3*d^9 + 27*B^4*a^6*b^3*c^2*d^10 - 4*B^4*a^6*b^3*c^4*d
^8 + 4*B^4*a^7*b^2*c^3*d^9 + 12*B^4*a*b^8*c*d^11 - 28*B^4*a*b^8*c^3*d^9 - 6
4*B^4*a^3*b^6*c*d^11 + 44*B^4*a^5*b^4*c*d^11 - 8*B^4*a^7*b^2*c*d^11 - B^4*a
^8*b*c^2*d^10))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*
b^2*f^4))*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B
^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^
4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) +
 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^
2 - 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^

4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(2*B^5*a^3*b^4*d^13 + 4*B^5*b^7*c^3*d^
10 - 6*B^5*a*b^6*d^13 + 4*B^5*b^7*c*d^12 - 9*B^5*a^2*b^5*c^3*d^10 + 4*B^5*a
^2*b^5*c^5*d^8 - 12*B^5*a^3*b^4*c^2*d^11 - 14*B^5*a^3*b^4*c^4*d^9 + 2*B^5*a
^4*b^3*c^3*d^10 - 4*B^5*a^4*b^3*c^5*d^8 + 4*B^5*a^5*b^2*c^2*d^11 + 4*B^5*a^
5*b^2*c^4*d^9 - B^5*a^6*b*c*d^12 + 6*B^5*a*b^6*c^4*d^9 - 13*B^5*a^2*b^5*c*d
^12 + 6*B^5*a^4*b^3*c*d^12 - B^5*a^6*b*c^3*d^10))/(a^8*f^5 + b^8*f^5 + 4*a^
2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5)))*((((8*B^2*a^4*c*f^2 + 8*B^2*b^
4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2
/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4
*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*
B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 +
b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*2i + (ata
n(-((((((8*(304*C^3*a^3*b^9*d^12*f^2 + 120*C^3*a^5*b^7*d^12*f^2 - 320*C^3*a
^7*b^5*d^12*f^2 - 148*C^3*a^9*b^3*d^12*f^2 + 4*C^3*b^12*c^3*d^9*f^2 - 4*C^3
*a*b^11*d^12*f^2 - 16*C^3*a^11*b*d^12*f^2 + 4*C^3*b^12*c*d^11*f^2 + 60*C^3*
a*b^11*c^2*d^10*f^2 + 64*C^3*a*b^11*c^4*d^8*f^2 - 320*C^3*a^2*b^10*c*d^11*f
^2 + 104*C^3*a^4*b^8*c*d^11*f^2 + 544*C^3*a^6*b^6*c*d^11*f^2 + 116*C^3*a^8*
b^4*c*d^11*f^2 - 16*C^3*a^11*b*c^2*d^10*f^2 - 320*C^3*a^2*b^10*c^3*d^9*f^2
+ 176*C^3*a^3*b^9*c^2*d^10*f^2 - 128*C^3*a^3*b^9*c^4*d^8*f^2 + 104*C^3*a^4*
b^8*c^3*d^9*f^2 - 72*C^3*a^5*b^7*c^2*d^10*f^2 - 192*C^3*a^5*b^7*c^4*d^8*f^2
+ 544*C^3*a^6*b^6*c^3*d^9*f^2 - 320*C^3*a^7*b^5*c^2*d^10*f^2 + 116*C^3*a^8
*b^4*c^3*d^9*f^2 - 148*C^3*a^9*b^3*c^2*d^10*f^2))/(b^9*f^5 + a^8*b*f^5 + 4*
a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^5) - (((16*(c + d*tan(e + f*x))^(
1/2)*(52*C^2*a^3*b^11*d^11*f^2 + 128*C^2*a^5*b^9*d^11*f^2 + 424*C^2*a^7*b^7
*d^11*f^2 + 380*C^2*a^9*b^5*d^11*f^2 + 100*C^2*a^11*b^3*d^11*f^2 - 20*C^2*b
^14*c^3*d^8*f^2 + 60*C^2*a*b^13*d^11*f^2 + 8*C^2*a^13*b*d^11*f^2 - 4*C^2*a^
14*c*d^10*f^2 - 12*C^2*b^14*c*d^10*f^2 + 84*C^2*a*b^13*c^2*d^9*f^2 + 60*C^2
*a^2*b^12*c*d^10*f^2 - 116*C^2*a^4*b^10*c*d^10*f^2 - 604*C^2*a^6*b^8*c*d^10
*f^2 - 596*C^2*a^8*b^6*c*d^10*f^2 - 220*C^2*a^10*b^4*c*d^10*f^2 - 44*C^2*a^
12*b^2*c*d^10*f^2 + 116*C^2*a^2*b^12*c^3*d^8*f^2 + 108*C^2*a^3*b^11*c^2*d^9
*f^2 + 216*C^2*a^4*b^10*c^3*d^8*f^2 + 104*C^2*a^5*b^9*c^2*d^9*f^2 + 8*C^2*a
^6*b^8*c^3*d^8*f^2 + 248*C^2*a^7*b^7*c^2*d^9*f^2 - 68*C^2*a^8*b^6*c^3*d^8*f
^2 + 196*C^2*a^9*b^5*c^2*d^9*f^2 + 4*C^2*a^10*b^4*c^3*d^8*f^2 + 28*C^2*a^11
*b^3*c^2*d^9*f^2))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4
*a^6*b^3*f^4) + (((8*(96*C*a^2*b^14*d^11*f^4 + 480*C*a^4*b^12*d^11*f^4 + 96
0*C*a^6*b^10*d^11*f^4 + 960*C*a^8*b^8*d^11*f^4 + 480*C*a^10*b^6*d^11*f^4 +
96*C*a^12*b^4*d^11*f^4 - 64*C*a*b^15*c^3*d^8*f^4 - 320*C*a^3*b^13*c*d^10*f^
4 - 640*C*a^5*b^11*c*d^10*f^4 - 640*C*a^7*b^9*c*d^10*f^4 - 320*C*a^9*b^7*c*
d^10*f^4 - 64*C*a^11*b^5*c*d^10*f^4 + 96*C*a^2*b^14*c^2*d^9*f^4 - 320*C*a^3
*b^13*c^3*d^8*f^4 + 480*C*a^4*b^12*c^2*d^9*f^4 - 640*C*a^5*b^11*c^3*d^8*f^4
+ 960*C*a^6*b^10*c^2*d^9*f^4 - 640*C*a^7*b^9*c^3*d^8*f^4 + 960*C*a^8*b^8*c
^2*d^9*f^4 - 320*C*a^9*b^7*c^3*d^8*f^4 + 480*C*a^10*b^6*c^2*d^9*f^4 - 64*C*
a^11*b^5*c^3*d^8*f^4 + 96*C*a^12*b^4*c^2*d^9*f^4 - 64*C*a*b^15*c*d^10*f^4))
/(b^9*f^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^5) - (4
*(c + d*tan(e + f*x))^(1/2)*(4*(C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 25*C^2*a

^4*b^4*d^2 + 10*C^2*a^6*b^2*d^2 - 40*C^2*a^3*b^5*c*d - 8*C^2*a^5*b^3*c*d)*(
b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4
*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^
2 - a*b^11*d*f^2))^(1/2)*(32*b^18*d^10*f^4 + 160*a^2*b^16*d^10*f^4 + 288*a^
4*b^14*d^10*f^4 + 160*a^6*b^12*d^10*f^4 - 160*a^8*b^10*d^10*f^4 - 288*a^10*
b^8*d^10*f^4 - 160*a^12*b^6*d^10*f^4 - 32*a^14*b^4*d^10*f^4 + 48*b^18*c^2*d
^8*f^4 + 272*a^2*b^16*c^2*d^8*f^4 + 624*a^4*b^14*c^2*d^8*f^4 + 720*a^6*b^12
*c^2*d^8*f^4 + 400*a^8*b^10*c^2*d^8*f^4 + 48*a^10*b^8*c^2*d^8*f^4 - 48*a^12
*b^6*c^2*d^8*f^4 - 16*a^14*b^4*c^2*d^8*f^4 + 16*a*b^17*c*d^9*f^4 + 112*a^3*
b^15*c*d^9*f^4 + 336*a^5*b^13*c*d^9*f^4 + 560*a^7*b^11*c*d^9*f^4 + 560*a^9*
b^9*c*d^9*f^4 + 336*a^11*b^7*c*d^9*f^4 + 112*a^13*b^5*c*d^9*f^4 + 16*a^15*b
^3*c*d^9*f^4))/((b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^
6*b^3*f^4)*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f
^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 -
a^9*b^3*d*f^2 - a*b^11*d*f^2)))*(4*(C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 25*C
^2*a^4*b^4*d^2 + 10*C^2*a^6*b^2*d^2 - 40*C^2*a^3*b^5*c*d - 8*C^2*a^5*b^3*c*
d)*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8
*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*
d*f^2 - a*b^11*d*f^2))^(1/2))/(4*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8
*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^
2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2)))*(4*(C^2*a^8*d^2 + 16*
C^2*a^2*b^6*c^2 + 25*C^2*a^4*b^4*d^2 + 10*C^2*a^6*b^2*d^2 - 40*C^2*a^3*b^5*
c*d - 8*C^2*a^5*b^3*c*d)*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 +
4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^
7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2))^(1/2))/(4*(b^12*c*f^2 + 4*a^2*
b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*
d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2)))
(4(C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 25*C^2*a^4*b^4*d^2 + 10*C^2*a^6*b^2
*d^2 - 40*C^2*a^3*b^5*c*d - 8*C^2*a^5*b^3*c*d)*(b^12*c*f^2 + 4*a^2*b^10*c*f
^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 -
6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2))^(1/2))/(
4*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*
b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d
*f^2 - a*b^11*d*f^2)) + (16*(c + d*tan(e + f*x))^(1/2)*(2*C^4*b^10*d^12 - C
^4*a^10*d^12 + 4*C^4*a^2*b^8*d^12 + 27*C^4*a^4*b^6*d^12 - 15*C^4*a^6*b^4*d^
12 - 9*C^4*a^8*b^2*d^12 + C^4*a^10*c^2*d^10 + 4*C^4*b^10*c^2*d^10 + 2*C^4*b
^10*c^4*d^8 + 24*C^4*a^2*b^8*c^2*d^10 - 12*C^4*a^2*b^8*c^4*d^8 + 104*C^4*a^
3*b^7*c^3*d^9 - 197*C^4*a^4*b^6*c^2*d^10 + 18*C^4*a^4*b^6*c^4*d^8 - 32*C^4*
a^5*b^5*c^3*d^9 - 17*C^4*a^6*b^4*c^2*d^10 - 8*C^4*a^7*b^3*c^3*d^9 + 9*C^4*a
^8*b^2*c^2*d^10 + 4*C^4*a^9*b*c*d^11 - 40*C^4*a^3*b^7*c*d^11 + 132*C^4*a^5*
b^5*c*d^11 + 48*C^4*a^7*b^3*c*d^11))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 +
6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*(4*(C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 25*
C^2*a^4*b^4*d^2 + 10*C^2*a^6*b^2*d^2 - 40*C^2*a^3*b^5*c*d - 8*C^2*a^5*b^3*c
d)(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^
8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3

*d*f^2 - a*b^11*d*f^2))^(1/2)*1i)/(4*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4
*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*
d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2)) - (((((8*(304*C^3*
a^3*b^9*d^12*f^2 + 120*C^3*a^5*b^7*d^12*f^2 - 320*C^3*a^7*b^5*d^12*f^2 - 14
8*C^3*a^9*b^3*d^12*f^2 + 4*C^3*b^12*c^3*d^9*f^2 - 4*C^3*a*b^11*d^12*f^2 - 1
6*C^3*a^11*b*d^12*f^2 + 4*C^3*b^12*c*d^11*f^2 + 60*C^3*a*b^11*c^2*d^10*f^2
+ 64*C^3*a*b^11*c^4*d^8*f^2 - 320*C^3*a^2*b^10*c*d^11*f^2 + 104*C^3*a^4*b^8
*c*d^11*f^2 + 544*C^3*a^6*b^6*c*d^11*f^2 + 116*C^3*a^8*b^4*c*d^11*f^2 - 16*
C^3*a^11*b*c^2*d^10*f^2 - 320*C^3*a^2*b^10*c^3*d^9*f^2 + 176*C^3*a^3*b^9*c^
2*d^10*f^2 - 128*C^3*a^3*b^9*c^4*d^8*f^2 + 104*C^3*a^4*b^8*c^3*d^9*f^2 - 72
*C^3*a^5*b^7*c^2*d^10*f^2 - 192*C^3*a^5*b^7*c^4*d^8*f^2 + 544*C^3*a^6*b^6*c
^3*d^9*f^2 - 320*C^3*a^7*b^5*c^2*d^10*f^2 + 116*C^3*a^8*b^4*c^3*d^9*f^2 - 1
48*C^3*a^9*b^3*c^2*d^10*f^2))/(b^9*f^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a^4*
b^5*f^5 + 4*a^6*b^3*f^5) + (((16*(c + d*tan(e + f*x))^(1/2)*(52*C^2*a^3*b^1
1*d^11*f^2 + 128*C^2*a^5*b^9*d^11*f^2 + 424*C^2*a^7*b^7*d^11*f^2 + 380*C^2*
a^9*b^5*d^11*f^2 + 100*C^2*a^11*b^3*d^11*f^2 - 20*C^2*b^14*c^3*d^8*f^2 + 60
*C^2*a*b^13*d^11*f^2 + 8*C^2*a^13*b*d^11*f^2 - 4*C^2*a^14*c*d^10*f^2 - 12*C
^2*b^14*c*d^10*f^2 + 84*C^2*a*b^13*c^2*d^9*f^2 + 60*C^2*a^2*b^12*c*d^10*f^2
 - 116*C^2*a^4*b^10*c*d^10*f^2 - 604*C^2*a^6*b^8*c*d^10*f^2 - 596*C^2*a^8*b
^6*c*d^10*f^2 - 220*C^2*a^10*b^4*c*d^10*f^2 - 44*C^2*a^12*b^2*c*d^10*f^2 +
116*C^2*a^2*b^12*c^3*d^8*f^2 + 108*C^2*a^3*b^11*c^2*d^9*f^2 + 216*C^2*a^4*b
^10*c^3*d^8*f^2 + 104*C^2*a^5*b^9*c^2*d^9*f^2 + 8*C^2*a^6*b^8*c^3*d^8*f^2 +
 248*C^2*a^7*b^7*c^2*d^9*f^2 - 68*C^2*a^8*b^6*c^3*d^8*f^2 + 196*C^2*a^9*b^5
*c^2*d^9*f^2 + 4*C^2*a^10*b^4*c^3*d^8*f^2 + 28*C^2*a^11*b^3*c^2*d^9*f^2))/(
b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4) - (((8
*(96*C*a^2*b^14*d^11*f^4 + 480*C*a^4*b^12*d^11*f^4 + 960*C*a^6*b^10*d^11*f^
4 + 960*C*a^8*b^8*d^11*f^4 + 480*C*a^10*b^6*d^11*f^4 + 96*C*a^12*b^4*d^11*f
^4 - 64*C*a*b^15*c^3*d^8*f^4 - 320*C*a^3*b^13*c*d^10*f^4 - 640*C*a^5*b^11*c
*d^10*f^4 - 640*C*a^7*b^9*c*d^10*f^4 - 320*C*a^9*b^7*c*d^10*f^4 - 64*C*a^11
*b^5*c*d^10*f^4 + 96*C*a^2*b^14*c^2*d^9*f^4 - 320*C*a^3*b^13*c^3*d^8*f^4 +
480*C*a^4*b^12*c^2*d^9*f^4 - 640*C*a^5*b^11*c^3*d^8*f^4 + 960*C*a^6*b^10*c^
2*d^9*f^4 - 640*C*a^7*b^9*c^3*d^8*f^4 + 960*C*a^8*b^8*c^2*d^9*f^4 - 320*C*a
^9*b^7*c^3*d^8*f^4 + 480*C*a^10*b^6*c^2*d^9*f^4 - 64*C*a^11*b^5*c^3*d^8*f^4
 + 96*C*a^12*b^4*c^2*d^9*f^4 - 64*C*a*b^15*c*d^10*f^4))/(b^9*f^5 + a^8*b*f^
5 + 4*a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^5) + (4*(c + d*tan(e + f*x)
)^(1/2)*(4*(C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 25*C^2*a^4*b^4*d^2 + 10*C^2*
a^6*b^2*d^2 - 40*C^2*a^3*b^5*c*d - 8*C^2*a^5*b^3*c*d)*(b^12*c*f^2 + 4*a^2*b
^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d
*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2))^(
1/2)*(32*b^18*d^10*f^4 + 160*a^2*b^16*d^10*f^4 + 288*a^4*b^14*d^10*f^4 + 16
0*a^6*b^12*d^10*f^4 - 160*a^8*b^10*d^10*f^4 - 288*a^10*b^8*d^10*f^4 - 160*a
^12*b^6*d^10*f^4 - 32*a^14*b^4*d^10*f^4 + 48*b^18*c^2*d^8*f^4 + 272*a^2*b^1
6*c^2*d^8*f^4 + 624*a^4*b^14*c^2*d^8*f^4 + 720*a^6*b^12*c^2*d^8*f^4 + 400*a
^8*b^10*c^2*d^8*f^4 + 48*a^10*b^8*c^2*d^8*f^4 - 48*a^12*b^6*c^2*d^8*f^4 - 1
6*a^14*b^4*c^2*d^8*f^4 + 16*a*b^17*c*d^9*f^4 + 112*a^3*b^15*c*d^9*f^4 + 336

*a^5*b^13*c*d^9*f^4 + 560*a^7*b^11*c*d^9*f^4 + 560*a^9*b^9*c*d^9*f^4 + 336*
a^11*b^7*c*d^9*f^4 + 112*a^13*b^5*c*d^9*f^4 + 16*a^15*b^3*c*d^9*f^4))/((b^9
*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4)*(b^12*c*f
^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 -
 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^
11*d*f^2)))*(4*(C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 25*C^2*a^4*b^4*d^2 + 10*
C^2*a^6*b^2*d^2 - 40*C^2*a^3*b^5*c*d - 8*C^2*a^5*b^3*c*d)*(b^12*c*f^2 + 4*a
^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b
^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2
))^(1/2))/(4*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c
*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2
 - a^9*b^3*d*f^2 - a*b^11*d*f^2)))*(4*(C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 25
*C^2*a^4*b^4*d^2 + 10*C^2*a^6*b^2*d^2 - 40*C^2*a^3*b^5*c*d - 8*C^2*a^5*b^3*
c*d)*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a
^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^
3*d*f^2 - a*b^11*d*f^2))^(1/2))/(4*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b
^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*
f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2)))*(4*(C^2*a^8*d^2 + 1
6*C^2*a^2*b^6*c^2 + 25*C^2*a^4*b^4*d^2 + 10*C^2*a^6*b^2*d^2 - 40*C^2*a^3*b^
5*c*d - 8*C^2*a^5*b^3*c*d)*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2
 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*
a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2))^(1/2))/(4*(b^12*c*f^2 + 4*a^
2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^
9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2)
) - (16*(c + d*tan(e + f*x))^(1/2)*(2*C^4*b^10*d^12 - C^4*a^10*d^12 + 4*C^4
*a^2*b^8*d^12 + 27*C^4*a^4*b^6*d^12 - 15*C^4*a^6*b^4*d^12 - 9*C^4*a^8*b^2*d
^12 + C^4*a^10*c^2*d^10 + 4*C^4*b^10*c^2*d^10 + 2*C^4*b^10*c^4*d^8 + 24*C^4
*a^2*b^8*c^2*d^10 - 12*C^4*a^2*b^8*c^4*d^8 + 104*C^4*a^3*b^7*c^3*d^9 - 197*
C^4*a^4*b^6*c^2*d^10 + 18*C^4*a^4*b^6*c^4*d^8 - 32*C^4*a^5*b^5*c^3*d^9 - 17
*C^4*a^6*b^4*c^2*d^10 - 8*C^4*a^7*b^3*c^3*d^9 + 9*C^4*a^8*b^2*c^2*d^10 + 4*
C^4*a^9*b*c*d^11 - 40*C^4*a^3*b^7*c*d^11 + 132*C^4*a^5*b^5*c*d^11 + 48*C^4*
a^7*b^3*c*d^11))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a
^6*b^3*f^4))*(4*(C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 25*C^2*a^4*b^4*d^2 + 10
*C^2*a^6*b^2*d^2 - 40*C^2*a^3*b^5*c*d - 8*C^2*a^5*b^3*c*d)*(b^12*c*f^2 + 4*
a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*
b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^
2))^(1/2)*1i)/(4*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b
^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*
f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2)))/((((((8*(304*C^3*a^3*b^9*d^12*f^2 + 1
20*C^3*a^5*b^7*d^12*f^2 - 320*C^3*a^7*b^5*d^12*f^2 - 148*C^3*a^9*b^3*d^12*f
^2 + 4*C^3*b^12*c^3*d^9*f^2 - 4*C^3*a*b^11*d^12*f^2 - 16*C^3*a^11*b*d^12*f^
2 + 4*C^3*b^12*c*d^11*f^2 + 60*C^3*a*b^11*c^2*d^10*f^2 + 64*C^3*a*b^11*c^4*
d^8*f^2 - 320*C^3*a^2*b^10*c*d^11*f^2 + 104*C^3*a^4*b^8*c*d^11*f^2 + 544*C^
3*a^6*b^6*c*d^11*f^2 + 116*C^3*a^8*b^4*c*d^11*f^2 - 16*C^3*a^11*b*c^2*d^10*
f^2 - 320*C^3*a^2*b^10*c^3*d^9*f^2 + 176*C^3*a^3*b^9*c^2*d^10*f^2 - 128*C^3

*a^3*b^9*c^4*d^8*f^2 + 104*C^3*a^4*b^8*c^3*d^9*f^2 - 72*C^3*a^5*b^7*c^2*d^10*f^2 - 192*C^3*a^5*b^7*c^4*d^8*f^2 + 544*C^3*a^6*b^6*c^3*d^9*f^2 - 320*C^3*a^7*b^5*c^2*d^10*f^2 + 116*C^3*a^8*b^4*c^3*d^9*f^2 - 148*C^3*a^9*b^3*c^2*d^10*f^2))/(b^9*f^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^5) - (((16*(c + d*tan(e + f*x))^(1/2)*(52*C^2*a^3*b^11*d^11*f^2 + 128*C^2*a^5*b^9*d^11*f^2 + 424*C^2*a^7*b^7*d^11*f^2 + 380*C^2*a^9*b^5*d^11*f^2 + 100*C^2*a^11*b^3*d^11*f^2 - 20*C^2*b^14*c^3*d^8*f^2 + 60*C^2*a*b^13*d^11*f^2 + 8*C^2*a^13*b*d^11*f^2 - 4*C^2*a^14*c*d^10*f^2 - 12*C^2*b^14*c*d^10*f^2 + 84*C^2*a*b^13*c^2*d^9*f^2 + 60*C^2*a^2*b^12*c*d^10*f^2 - 116*C^2*a^4*b^10*c*d^10*f^2 - 604*C^2*a^6*b^8*c*d^10*f^2 - 596*C^2*a^8*b^6*c*d^10*f^2 - 220*C^2*a^10*b^4*c*d^10*f^2 - 44*C^2*a^12*b^2*c*d^10*f^2 + 116*C^2*a^2*b^12*c^3*d^8*f^2 + 108*C^2*a^3*b^11*c^2*d^9*f^2 + 216*C^2*a^4*b^10*c^3*d^8*f^2 + 104*C^2*a^5*b^9*c^2*d^9*f^2 + 8*C^2*a^6*b^8*c^3*d^8*f^2 + 248*C^2*a^7*b^7*c^2*d^9*f^2 - 68*C^2*a^8*b^6*c^3*d^8*f^2 + 196*C^2*a^9*b^5*c^2*d^9*f^2 + 4*C^2*a^10*b^4*c^3*d^8*f^2 + 28*C^2*a^11*b^3*c^2*d^9*f^2))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4) + (((8*(96*C*a^2*b^14*d^11*f^4 + 480*C*a^4*b^12*d^11*f^4 + 960*C*a^6*b^10*d^11*f^4 + 960*C*a^8*b^8*d^11*f^4 + 480*C*a^10*b^6*d^11*f^4 + 96*C*a^12*b^4*d^11*f^4 - 64*C*a*b^15*c^3*d^8*f^4 - 320*C*a^3*b^13*c*d^10*f^4 - 640*C*a^5*b^11*c*d^10*f^4 - 640*C*a^7*b^9*c*d^10*f^4 - 320*C*a^9*b^7*c*d^10*f^4 - 64*C*a^11*b^5*c*d^10*f^4 + 96*C*a^2*b^14*c^2*d^9*f^4 - 320*C*a^3*b^13*c^3*d^8*f^4 + 480*C*a^4*b^12*c^2*d^9*f^4 - 640*C*a^5*b^11*c^3*d^8*f^4 + 960*C*a^6*b^10*c^2*d^9*f^4 - 640*C*a^7*b^9*c^3*d^8*f^4 + 960*C*a^8*b^8*c^2*d^9*f^4 - 320*C*a^9*b^7*c^3*d^8*f^4 + 480*C*a^10*b^6*c^2*d^9*f^4 - 64*C*a^11*b^5*c^3*d^8*f^4 + 96*C*a^12*b^4*c^2*d^9*f^4 - 64*C*a*b^15*c*d^10*f^4))/(b^9*f^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^5) - (4*(c + d*tan(e + f*x))^(1/2)*(4*(C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 25*C^2*a^4*b^4*d^2 + 10*C^2*a^6*b^2*d^2 - 40*C^2*a^3*b^5*c*d - 8*C^2*a^5*b^3*c*d)*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2))^(1/2)*(32*b^18*d^10*f^4 + 160*a^2*b^16*d^10*f^4 + 288*a^4*b^14*d^10*f^4 + 160*a^6*b^12*d^10*f^4 - 160*a^8*b^10*d^10*f^4 - 288*a^10*b^8*d^10*f^4 - 160*a^12*b^6*d^10*f^4 - 32*a^14*b^4*d^10*f^4 + 48*b^18*c^2*d^8*f^4 + 272*a^2*b^16*c^2*d^8*f^4 + 624*a^4*b^14*c^2*d^8*f^4 + 720*a^6*b^12*c^2*d^8*f^4 + 400*a^8*b^10*c^2*d^8*f^4 + 48*a^10*b^8*c^2*d^8*f^4 - 48*a^12*b^6*c^2*d^8*f^4 - 16*a^14*b^4*c^2*d^8*f^4 + 16*a*b^17*c*d^9*f^4 + 112*a^3*b^15*c*d^9*f^4 + 336*a^5*b^13*c*d^9*f^4 + 560*a^7*b^11*c*d^9*f^4 + 560*a^9*b^9*c*d^9*f^4 + 336*a^11*b^7*c*d^9*f^4 + 112*a^13*b^5*c*d^9*f^4 + 16*a^15*b^3*c*d^9*f^4))/((b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4)*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2)))*(4*(C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 25*C^2*a^4*b^4*d^2 + 10*C^2*a^6*b^2*d^2 - 40*C^2*a^3*b^5*c*d - 8*C^2*a^5*b^3*c*d)*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2))^(1/2))/(4*(b^12*c *f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2)))*(4*(C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 25*C^2*a^4*b^4*d^2 + 10*C^2*a^6*b^2*d^2 - 40*C^2*a^3*b^5*c*d - 8*C^2*a^5*b^3*c*d)*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2))^(1/2))/(4*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2)))*(4*(C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 25*C^2*a^4*b^4*d^2 + 10*C^2*a^6*b^2*d^2 - 40*C^2*a^3*b^5*c*d - 8*C^2*a^5*b^3*c*d)*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2))^(1/2))/(4*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2)) + (16*(c + d*tan(e + f*x))^(1/2)*(2*C^4*b^10*d^12 - C^4*a^10*d^12 + 4*C^4*a^2*b^8*d^12 + 27*C^4*a^4*b^6*d^12 - 15*C^4*a^6*b^4*d^12 - 9*C^4*a^8*b^2*d^12 + C^4*a^10*c^2*d^10 + 4*C^4*b^10*c^2*d^10 + 2*C^4*b^10*c^4*d^8 + 24*C^4*a^2*b^8*c^2*d^10 - 12*C^4*a^2*b^8*c^4*d^8 + 104*C^4*a^3*b^7*c^3*d^9 - 197*C^4*a^4*b^6*c^2*d^10 + 18*C^4*a^4*b^6*c^4*d^8 - 32*C^4*a^5*b^5*c^3*d^9 - 17*C^4*a^6*b^4*c^2*d^10 - 8*C^4*a^7*b^3*c^3*d^9 + 9*C^4*a^8*b^2*c^2*d^10 + 4*C^4*a^9*b*c*d^11 - 40*C^4*a^3*b^7*c*d^11 + 132*C^4*a^5*b^5*c*d^11 + 48*C^4*a^7*b^3*c*d^11))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*(4*(C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 25*C^2*a^4*b^4*d^2 + 10*C^2*a^6*b^2*d^2 - 40*C^2*a^3*b^5*c*d - 8*C^2*a^5*b^3*c*d)*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2))^(1/2))/(4*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2)) - (16*(C^5*a^8*d^13 + 10*C^5*a^2*b^6*d^13 + 27*C^5*a^4*b^4*d^13 + 10*C^5*a^6*b^2*d^13 + C^5*a^8*c^2*d^11 + 36*C^5*a^2*b^6*c^2*d^11 + 26*C^5*a^2*b^6*c^4*d^9 - 40*C^5*a^3*b^5*c^3*d^10 + 29*C^5*a^4*b^4*c^2*d^11 + 2*C^5*a^4*b^4*c^4*d^9 - 8*C^5*a^5*b^3*c^3*d^10 + 10*C^5*a^6*b^2*c^2*d^11 - 8*C^5*a*b^7*c*d^12 - 16*C^5*a*b^7*c^3*d^10 - 8*C^5*a*b^7*c^5*d^8 - 40*C^5*a^3*b^5*c*d^12 - 8*C^5*a^5*b^3*c*d^12))/(b^9*f^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^5) + (((((8*(304*C^3*a^3*b^9*d^12*f^2 + 120*C^3*a^5*b^7*d^12*f^2 - 320*C^3*a^7*b^5*d^12*f^2 - 148*C^3*a^9*b^3*d^12*f^2 + 4*C^3*b^12*c^3*d^9*f^2 - 4*C^3*a*b^11*d^12*f^2 - 16*C^3*a^11*b*d^12*f^2 + 4*C^3*b^12*c*d^11*f^2 + 60*C^3*a*b^11*c^2*d^10*f^2 + 64*C^3*a*b^11*c^4*d^8*f^2 - 320*C^3*a^2*b^10*c*d^11*f^2 + 104*C^3*a^4*b^8*c*d^11*f^2 + 544*C^3*a^6*b^6*c*d^11*f^2 + 116*C^3*a^8*b^4*c*d^11*f^2 - 16*C^3*a^11*b*c^2*d^10*f^2 - 320*C^3*a^2*b^10*c^3*d^9*f^2 + 176*C^3*a^3*b^9*c^2*d^10*f^2 - 128*C^3*a^3*b^9*c^4*d^8*f^2 + 104*C^3*a^4*b^8*c^3*d^9*f^2 - 72*C^3*a^5*b^7*c^2*d^10*f^2 - 192*C^3*a^5*b^7*c^4*d^8*f^2 + 544*C^3*a^6*b^6*c^3*d^9*f^2 - 320*C^3*a^7*b^5*c^2*d^10*f^2 + 116*C^3*a^8*b^4*c^3*d^9*f^2 - 148*C^3*a^9*b^3*c^2*d^10*

f^2))/(b^9*f^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^5)
 + (((16*(c + d*tan(e + f*x))^(1/2)*(52*C^2*a^3*b^11*d^11*f^2 + 128*C^2*a^5
*b^9*d^11*f^2 + 424*C^2*a^7*b^7*d^11*f^2 + 380*C^2*a^9*b^5*d^11*f^2 + 100*C
^2*a^11*b^3*d^11*f^2 - 20*C^2*b^14*c^3*d^8*f^2 + 60*C^2*a*b^13*d^11*f^2 + 8
*C^2*a^13*b*d^11*f^2 - 4*C^2*a^14*c*d^10*f^2 - 12*C^2*b^14*c*d^10*f^2 + 84*
C^2*a*b^13*c^2*d^9*f^2 + 60*C^2*a^2*b^12*c*d^10*f^2 - 116*C^2*a^4*b^10*c*d^
10*f^2 - 604*C^2*a^6*b^8*c*d^10*f^2 - 596*C^2*a^8*b^6*c*d^10*f^2 - 220*C^2*
a^10*b^4*c*d^10*f^2 - 44*C^2*a^12*b^2*c*d^10*f^2 + 116*C^2*a^2*b^12*c^3*d^8
*f^2 + 108*C^2*a^3*b^11*c^2*d^9*f^2 + 216*C^2*a^4*b^10*c^3*d^8*f^2 + 104*C^
2*a^5*b^9*c^2*d^9*f^2 + 8*C^2*a^6*b^8*c^3*d^8*f^2 + 248*C^2*a^7*b^7*c^2*d^9
*f^2 - 68*C^2*a^8*b^6*c^3*d^8*f^2 + 196*C^2*a^9*b^5*c^2*d^9*f^2 + 4*C^2*a^1
0*b^4*c^3*d^8*f^2 + 28*C^2*a^11*b^3*c^2*d^9*f^2))/(b^9*f^4 + a^8*b*f^4 + 4*
a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4) - (((8*(96*C*a^2*b^14*d^11*f^4
 + 480*C*a^4*b^12*d^11*f^4 + 960*C*a^6*b^10*d^11*f^4 + 960*C*a^8*b^8*d^11*f
^4 + 480*C*a^10*b^6*d^11*f^4 + 96*C*a^12*b^4*d^11*f^4 - 64*C*a*b^15*c^3*d^8
*f^4 - 320*C*a^3*b^13*c*d^10*f^4 - 640*C*a^5*b^11*c*d^10*f^4 - 640*C*a^7*b^
9*c*d^10*f^4 - 320*C*a^9*b^7*c*d^10*f^4 - 64*C*a^11*b^5*c*d^10*f^4 + 96*C*a
^2*b^14*c^2*d^9*f^4 - 320*C*a^3*b^13*c^3*d^8*f^4 + 480*C*a^4*b^12*c^2*d^9*f
^4 - 640*C*a^5*b^11*c^3*d^8*f^4 + 960*C*a^6*b^10*c^2*d^9*f^4 - 640*C*a^7*b^
9*c^3*d^8*f^4 + 960*C*a^8*b^8*c^2*d^9*f^4 - 320*C*a^9*b^7*c^3*d^8*f^4 + 480
*C*a^10*b^6*c^2*d^9*f^4 - 64*C*a^11*b^5*c^3*d^8*f^4 + 96*C*a^12*b^4*c^2*d^9
*f^4 - 64*C*a*b^15*c*d^10*f^4))/(b^9*f^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a^
4*b^5*f^5 + 4*a^6*b^3*f^5) + (4*(c + d*tan(e + f*x))^(1/2)*(4*(C^2*a^8*d^2
+ 16*C^2*a^2*b^6*c^2 + 25*C^2*a^4*b^4*d^2 + 10*C^2*a^6*b^2*d^2 - 40*C^2*a^3
*b^5*c*d - 8*C^2*a^5*b^3*c*d)*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*
f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 -
 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2))^(1/2)*(32*b^18*d^10*f^4 +
 160*a^2*b^16*d^10*f^4 + 288*a^4*b^14*d^10*f^4 + 160*a^6*b^12*d^10*f^4 - 16
0*a^8*b^10*d^10*f^4 - 288*a^10*b^8*d^10*f^4 - 160*a^12*b^6*d^10*f^4 - 32*a^
14*b^4*d^10*f^4 + 48*b^18*c^2*d^8*f^4 + 272*a^2*b^16*c^2*d^8*f^4 + 624*a^4*
b^14*c^2*d^8*f^4 + 720*a^6*b^12*c^2*d^8*f^4 + 400*a^8*b^10*c^2*d^8*f^4 + 48
*a^10*b^8*c^2*d^8*f^4 - 48*a^12*b^6*c^2*d^8*f^4 - 16*a^14*b^4*c^2*d^8*f^4 +
 16*a*b^17*c*d^9*f^4 + 112*a^3*b^15*c*d^9*f^4 + 336*a^5*b^13*c*d^9*f^4 + 56
0*a^7*b^11*c*d^9*f^4 + 560*a^9*b^9*c*d^9*f^4 + 336*a^11*b^7*c*d^9*f^4 + 112
*a^13*b^5*c*d^9*f^4 + 16*a^15*b^3*c*d^9*f^4))/((b^9*f^4 + a^8*b*f^4 + 4*a^2
*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4)*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 +
6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5
*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2)))*(4*(C^2*a^8*
d^2 + 16*C^2*a^2*b^6*c^2 + 25*C^2*a^4*b^4*d^2 + 10*C^2*a^6*b^2*d^2 - 40*C^2
*a^3*b^5*c*d - 8*C^2*a^5*b^3*c*d)*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^
8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f
^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2))^(1/2))/(4*(b^12*c*f^2
 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4
*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11
*d*f^2)))*(4*(C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 25*C^2*a^4*b^4*d^2 + 10*C^

2*a^6*b^2*d^2 - 40*C^2*a^3*b^5*c*d - 8*C^2*a^5*b^3*c*d)*(b^12*c*f^2 + 4*a^2
*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9
*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2))
^(1/2))/(4*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f
^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 -
a^9*b^3*d*f^2 - a*b^11*d*f^2)))*(4*(C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 25*C
^2*a^4*b^4*d^2 + 10*C^2*a^6*b^2*d^2 - 40*C^2*a^3*b^5*c*d - 8*C^2*a^5*b^3*c*
d)*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8
*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*
d*f^2 - a*b^11*d*f^2))^(1/2))/(4*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8
*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^
2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2)) - (16*(c + d*tan(e + f
x))^(1/2)(2*C^4*b^10*d^12 - C^4*a^10*d^12 + 4*C^4*a^2*b^8*d^12 + 27*C^4*a
^4*b^6*d^12 - 15*C^4*a^6*b^4*d^12 - 9*C^4*a^8*b^2*d^12 + C^4*a^10*c^2*d^10
+ 4*C^4*b^10*c^2*d^10 + 2*C^4*b^10*c^4*d^8 + 24*C^4*a^2*b^8*c^2*d^10 - 12*C
^4*a^2*b^8*c^4*d^8 + 104*C^4*a^3*b^7*c^3*d^9 - 197*C^4*a^4*b^6*c^2*d^10 + 1
8*C^4*a^4*b^6*c^4*d^8 - 32*C^4*a^5*b^5*c^3*d^9 - 17*C^4*a^6*b^4*c^2*d^10 -
8*C^4*a^7*b^3*c^3*d^9 + 9*C^4*a^8*b^2*c^2*d^10 + 4*C^4*a^9*b*c*d^11 - 40*C^
4*a^3*b^7*c*d^11 + 132*C^4*a^5*b^5*c*d^11 + 48*C^4*a^7*b^3*c*d^11))/(b^9*f^
4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*(4*(C^2*a^8
*d^2 + 16*C^2*a^2*b^6*c^2 + 25*C^2*a^4*b^4*d^2 + 10*C^2*a^6*b^2*d^2 - 40*C^
2*a^3*b^5*c*d - 8*C^2*a^5*b^3*c*d)*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b
^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*
f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2))^(1/2))/(4*(b^12*c*f^
2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 -
4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^1
1*d*f^2))))*(4*(C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 25*C^2*a^4*b^4*d^2 + 10*
C^2*a^6*b^2*d^2 - 40*C^2*a^3*b^5*c*d - 8*C^2*a^5*b^3*c*d)*(b^12*c*f^2 + 4*a
^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b
^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2
))^(1/2)*1i)/(2*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^
6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f
^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2)) + (atan(((((16*(c + d*tan(e + f*x))^(1/
2)*(2*B^4*b^9*d^12 - 5*B^4*a^2*b^7*d^12 + 17*B^4*a^4*b^5*d^12 - 7*B^4*a^6*b
^3*d^12 + 6*B^4*b^9*c^4*d^8 + B^4*a^8*b*d^12 + 77*B^4*a^2*b^7*c^2*d^10 - 8*
B^4*a^2*b^7*c^4*d^8 + 60*B^4*a^3*b^6*c^3*d^9 - 87*B^4*a^4*b^5*c^2*d^10 + 14
*B^4*a^4*b^5*c^4*d^8 - 36*B^4*a^5*b^4*c^3*d^9 + 27*B^4*a^6*b^3*c^2*d^10 - 4
*B^4*a^6*b^3*c^4*d^8 + 4*B^4*a^7*b^2*c^3*d^9 + 12*B^4*a*b^8*c*d^11 - 28*B^4
*a*b^8*c^3*d^9 - 64*B^4*a^3*b^6*c*d^11 + 44*B^4*a^5*b^4*c*d^11 - 8*B^4*a^7*
b^2*c*d^11 - B^4*a^8*b*c^2*d^10))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^
4*b^4*f^4 + 4*a^6*b^2*f^4) - (((8*(156*B^3*a^2*b^9*d^12*f^2 - 16*B^3*a^4*b^
7*d^12*f^2 - 120*B^3*a^6*b^5*d^12*f^2 + 48*B^3*a^8*b^3*d^12*f^2 + 12*B^3*b^
11*c^2*d^10*f^2 + 12*B^3*b^11*c^4*d^8*f^2 - 4*B^3*a^10*b*d^12*f^2 - 124*B^3
*a*b^10*c*d^11*f^2 - 124*B^3*a*b^10*c^3*d^9*f^2 + 224*B^3*a^3*b^8*c*d^11*f^
2 + 200*B^3*a^5*b^6*c*d^11*f^2 - 128*B^3*a^7*b^4*c*d^11*f^2 + 20*B^3*a^9*b^

2*c*d^11*f^2 - 4*B^3*a^10*b*c^2*d^10*f^2 + 44*B^3*a^2*b^9*c^2*d^10*f^2 - 11
2*B^3*a^2*b^9*c^4*d^8*f^2 + 224*B^3*a^3*b^8*c^3*d^9*f^2 - 40*B^3*a^4*b^7*c^
2*d^10*f^2 - 24*B^3*a^4*b^7*c^4*d^8*f^2 + 200*B^3*a^5*b^6*c^3*d^9*f^2 - 40*
B^3*a^6*b^5*c^2*d^10*f^2 + 80*B^3*a^6*b^5*c^4*d^8*f^2 - 128*B^3*a^7*b^4*c^3
*d^9*f^2 + 28*B^3*a^8*b^3*c^2*d^10*f^2 - 20*B^3*a^8*b^3*c^4*d^8*f^2 + 20*B^
3*a^9*b^2*c^3*d^9*f^2))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5
+ 4*a^6*b^2*f^5) + (((((8*(80*B*a*b^14*d^11*f^4 - 48*B*b^15*c*d^10*f^4 + 38
4*B*a^3*b^12*d^11*f^4 + 720*B*a^5*b^10*d^11*f^4 + 640*B*a^7*b^8*d^11*f^4 +
240*B*a^9*b^6*d^11*f^4 - 16*B*a^13*b^2*d^11*f^4 - 48*B*b^15*c^3*d^8*f^4 + 8
0*B*a*b^14*c^2*d^9*f^4 - 224*B*a^2*b^13*c*d^10*f^4 - 400*B*a^4*b^11*c*d^10*
f^4 - 320*B*a^6*b^9*c*d^10*f^4 - 80*B*a^8*b^7*c*d^10*f^4 + 32*B*a^10*b^5*c*
d^10*f^4 + 16*B*a^12*b^3*c*d^10*f^4 - 224*B*a^2*b^13*c^3*d^8*f^4 + 384*B*a^
3*b^12*c^2*d^9*f^4 - 400*B*a^4*b^11*c^3*d^8*f^4 + 720*B*a^5*b^10*c^2*d^9*f^
4 - 320*B*a^6*b^9*c^3*d^8*f^4 + 640*B*a^7*b^8*c^2*d^9*f^4 - 80*B*a^8*b^7*c^
3*d^8*f^4 + 240*B*a^9*b^6*c^2*d^9*f^4 + 32*B*a^10*b^5*c^3*d^8*f^4 + 16*B*a^
12*b^3*c^3*d^8*f^4 - 16*B*a^13*b^2*c^2*d^9*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2
*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) - (4*(c + d*tan(e + f*x))^(1/2)*(
4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9*
B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^2*d^2 + 16*B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*d
 - 4*B^2*a^5*b*c*d)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6
*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*
d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))^(1/2)*(32*b^17*d^10*f^4 + 160*a^2*b^15*
d^10*f^4 + 288*a^4*b^13*d^10*f^4 + 160*a^6*b^11*d^10*f^4 - 160*a^8*b^9*d^10
*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b^5*d^10*f^4 - 32*a^14*b^3*d^10*f^4
 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^
4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^9*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^
8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14*b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9
*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336*a^5*b^12*c*d^9*f^4 + 560*a^7*b^10*c*d^9
*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*a^11*b^6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*
f^4 + 16*a^15*b^2*c*d^9*f^4))/((a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b
^4*f^4 + 4*a^6*b^2*f^4)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4
*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*
b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)))*(4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 -
8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9*B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^2*d
^2 + 16*B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*d - 4*B^2*a^5*b*c*d)*(b^10*c*f^2 +
 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^
3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2
))^(1/2))/(4*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*
f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 -
 a*b^9*d*f^2 - a^9*b*d*f^2)) - (16*(c + d*tan(e + f*x))^(1/2)*(44*B^2*a^9*b
^4*d^11*f^2 - 168*B^2*a^5*b^8*d^11*f^2 - 40*B^2*a^7*b^6*d^11*f^2 - 20*B^2*a
^3*b^10*d^11*f^2 - 4*B^2*a^11*b^2*d^11*f^2 - 36*B^2*b^13*c^3*d^8*f^2 + 60*B
^2*a*b^12*d^11*f^2 - 12*B^2*b^13*c*d^10*f^2 + 4*B^2*a^12*b*c*d^10*f^2 + 100
*B^2*a*b^12*c^2*d^9*f^2 + 120*B^2*a^2*b^11*c*d^10*f^2 + 156*B^2*a^4*b^9*c*d
^10*f^2 - 112*B^2*a^6*b^7*c*d^10*f^2 - 148*B^2*a^8*b^5*c*d^10*f^2 - 8*B^2*a

^10*b^3*c*d^10*f^2 + 68*B^2*a^2*b^11*c^3*d^8*f^2 + 124*B^2*a^3*b^10*c^2*d^9
*f^2 + 184*B^2*a^4*b^9*c^3*d^8*f^2 + 8*B^2*a^5*b^8*c^2*d^9*f^2 + 40*B^2*a^6
*b^7*c^3*d^8*f^2 + 24*B^2*a^7*b^6*c^2*d^9*f^2 - 20*B^2*a^8*b^5*c^3*d^8*f^2
+ 20*B^2*a^9*b^4*c^2*d^9*f^2 + 20*B^2*a^10*b^3*c^3*d^8*f^2 - 20*B^2*a^11*b^
2*c^2*d^9*f^2))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*
b^2*f^4))*(4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b
^2*c^2 + 9*B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^2*d^2 + 16*B^2*a^3*b^3*c*d - 12*B^
2*a*b^5*c*d - 4*B^2*a^5*b*c*d)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*
f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 -
4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))^(1/2))/(4*(b^10*c*f^2 + 4*a^
2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7
*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)))*(
4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9*
B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^2*d^2 + 16*B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*d
- 4*B^2*a^5*b*c*d)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6
*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*
d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))^(1/2))/(4*(b^10*c*f^2 + 4*a^2*b^8*c*f^2
+ 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*
a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)))*(4*(B^2*a^6*
d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9*B^2*a^2*b^4
*d^2 - 6*B^2*a^4*b^2*d^2 + 16*B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*d - 4*B^2*a^
5*b*c*d)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2
+ a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b
^9*d*f^2 - a^9*b*d*f^2))^(1/2)*1i)/(4*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4
*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*
d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)) + (((16*(c + d*tan(e
+ f*x))^(1/2)*(2*B^4*b^9*d^12 - 5*B^4*a^2*b^7*d^12 + 17*B^4*a^4*b^5*d^12 -
7*B^4*a^6*b^3*d^12 + 6*B^4*b^9*c^4*d^8 + B^4*a^8*b*d^12 + 77*B^4*a^2*b^7*c^
2*d^10 - 8*B^4*a^2*b^7*c^4*d^8 + 60*B^4*a^3*b^6*c^3*d^9 - 87*B^4*a^4*b^5*c^
2*d^10 + 14*B^4*a^4*b^5*c^4*d^8 - 36*B^4*a^5*b^4*c^3*d^9 + 27*B^4*a^6*b^3*c
^2*d^10 - 4*B^4*a^6*b^3*c^4*d^8 + 4*B^4*a^7*b^2*c^3*d^9 + 12*B^4*a*b^8*c*d^
11 - 28*B^4*a*b^8*c^3*d^9 - 64*B^4*a^3*b^6*c*d^11 + 44*B^4*a^5*b^4*c*d^11 -
8*B^4*a^7*b^2*c*d^11 - B^4*a^8*b*c^2*d^10))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6
*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4) + (((8*(156*B^3*a^2*b^9*d^12*f^2 - 16
*B^3*a^4*b^7*d^12*f^2 - 120*B^3*a^6*b^5*d^12*f^2 + 48*B^3*a^8*b^3*d^12*f^2
+ 12*B^3*b^11*c^2*d^10*f^2 + 12*B^3*b^11*c^4*d^8*f^2 - 4*B^3*a^10*b*d^12*f^
2 - 124*B^3*a*b^10*c*d^11*f^2 - 124*B^3*a*b^10*c^3*d^9*f^2 + 224*B^3*a^3*b^
8*c*d^11*f^2 + 200*B^3*a^5*b^6*c*d^11*f^2 - 128*B^3*a^7*b^4*c*d^11*f^2 + 20
*B^3*a^9*b^2*c*d^11*f^2 - 4*B^3*a^10*b*c^2*d^10*f^2 + 44*B^3*a^2*b^9*c^2*d^
10*f^2 - 112*B^3*a^2*b^9*c^4*d^8*f^2 + 224*B^3*a^3*b^8*c^3*d^9*f^2 - 40*B^3
*a^4*b^7*c^2*d^10*f^2 - 24*B^3*a^4*b^7*c^4*d^8*f^2 + 200*B^3*a^5*b^6*c^3*d^
9*f^2 - 40*B^3*a^6*b^5*c^2*d^10*f^2 + 80*B^3*a^6*b^5*c^4*d^8*f^2 - 128*B^3*
a^7*b^4*c^3*d^9*f^2 + 28*B^3*a^8*b^3*c^2*d^10*f^2 - 20*B^3*a^8*b^3*c^4*d^8*
f^2 + 20*B^3*a^9*b^2*c^3*d^9*f^2))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a
^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((((8*(80*B*a*b^14*d^11*f^4 - 48*B*b^15*c*d^

10*f^4 + 384*B*a^3*b^12*d^11*f^4 + 720*B*a^5*b^10*d^11*f^4 + 640*B*a^7*b^8*
d^11*f^4 + 240*B*a^9*b^6*d^11*f^4 - 16*B*a^13*b^2*d^11*f^4 - 48*B*b^15*c^3*
d^8*f^4 + 80*B*a*b^14*c^2*d^9*f^4 - 224*B*a^2*b^13*c*d^10*f^4 - 400*B*a^4*b
^11*c*d^10*f^4 - 320*B*a^6*b^9*c*d^10*f^4 - 80*B*a^8*b^7*c*d^10*f^4 + 32*B*
a^10*b^5*c*d^10*f^4 + 16*B*a^12*b^3*c*d^10*f^4 - 224*B*a^2*b^13*c^3*d^8*f^4
 + 384*B*a^3*b^12*c^2*d^9*f^4 - 400*B*a^4*b^11*c^3*d^8*f^4 + 720*B*a^5*b^10
*c^2*d^9*f^4 - 320*B*a^6*b^9*c^3*d^8*f^4 + 640*B*a^7*b^8*c^2*d^9*f^4 - 80*B
*a^8*b^7*c^3*d^8*f^4 + 240*B*a^9*b^6*c^2*d^9*f^4 + 32*B*a^10*b^5*c^3*d^8*f^
4 + 16*B*a^12*b^3*c^3*d^8*f^4 - 16*B*a^13*b^2*c^2*d^9*f^4))/(a^8*f^5 + b^8*
f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (4*(c + d*tan(e + f*
x))^(1/2)*(4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b
^2*c^2 + 9*B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^2*d^2 + 16*B^2*a^3*b^3*c*d - 12*B^
2*a*b^5*c*d - 4*B^2*a^5*b*c*d)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*
f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 -
 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))^(1/2)*(32*b^17*d^10*f^4 + 16
0*a^2*b^15*d^10*f^4 + 288*a^4*b^13*d^10*f^4 + 160*a^6*b^11*d^10*f^4 - 160*a
^8*b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b^5*d^10*f^4 - 32*a^14*b
^3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*d^8*f^4 + 624*a^4*b^13
*c^2*d^8*f^4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^9*c^2*d^8*f^4 + 48*a^10
*b^7*c^2*d^8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14*b^3*c^2*d^8*f^4 + 16*a
*b^16*c*d^9*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336*a^5*b^12*c*d^9*f^4 + 560*a^7
*b^10*c*d^9*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*a^11*b^6*c*d^9*f^4 + 112*a^13
*b^4*c*d^9*f^4 + 16*a^15*b^2*c*d^9*f^4))/((a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^
4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^
6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f
^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)))*(4*(B^2*a^6*d^2 + 4*B^2
*b^6*c^2 - 8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9*B^2*a^2*b^4*d^2 - 6*B^
2*a^4*b^2*d^2 + 16*B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*d - 4*B^2*a^5*b*c*d)*(b
^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c
*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 -
a^9*b*d*f^2))^(1/2))/(4*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4
*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*
b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)) + (16*(c + d*tan(e + f*x))^(1/2)*(4
4*B^2*a^9*b^4*d^11*f^2 - 168*B^2*a^5*b^8*d^11*f^2 - 40*B^2*a^7*b^6*d^11*f^2
 - 20*B^2*a^3*b^10*d^11*f^2 - 4*B^2*a^11*b^2*d^11*f^2 - 36*B^2*b^13*c^3*d^8
*f^2 + 60*B^2*a*b^12*d^11*f^2 - 12*B^2*b^13*c*d^10*f^2 + 4*B^2*a^12*b*c*d^1
0*f^2 + 100*B^2*a*b^12*c^2*d^9*f^2 + 120*B^2*a^2*b^11*c*d^10*f^2 + 156*B^2*
a^4*b^9*c*d^10*f^2 - 112*B^2*a^6*b^7*c*d^10*f^2 - 148*B^2*a^8*b^5*c*d^10*f^
2 - 8*B^2*a^10*b^3*c*d^10*f^2 + 68*B^2*a^2*b^11*c^3*d^8*f^2 + 124*B^2*a^3*b
^10*c^2*d^9*f^2 + 184*B^2*a^4*b^9*c^3*d^8*f^2 + 8*B^2*a^5*b^8*c^2*d^9*f^2 +
 40*B^2*a^6*b^7*c^3*d^8*f^2 + 24*B^2*a^7*b^6*c^2*d^9*f^2 - 20*B^2*a^8*b^5*c
^3*d^8*f^2 + 20*B^2*a^9*b^4*c^2*d^9*f^2 + 20*B^2*a^10*b^3*c^3*d^8*f^2 - 20*
B^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f
^4 + 4*a^6*b^2*f^4))*(4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^2*b^4*c^2 +
4*B^2*a^4*b^2*c^2 + 9*B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^2*d^2 + 16*B^2*a^3*b^3*

c*d - 12*B^2*a*b^5*c*d - 4*B^2*a^5*b*c*d)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))^(1/2))/(4*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)))*(4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9*B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^2*d^2 + 16*B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*d - 4*B^2*a^5*b*c*d)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))^(1/2))/(4*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)))*(4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9*B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^2*d^2 + 16*B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*d - 4*B^2*a^5*b*c*d)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))^(1/2)*1i)/(4*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)))/((16*(2*B^5*a^3*b^4*d^13 + 4*B^5*b^7*c^3*d^10 - 6*B^5*a*b^6*d^13 + 4*B^5*b^7*c*d^12 - 9*B^5*a^2*b^5*c^3*d^10 + 4*B^5*a^2*b^5*c^5*d^8 - 12*B^5*a^3*b^4*c^2*d^11 - 14*B^5*a^3*b^4*c^4*d^9 + 2*B^5*a^4*b^3*c^3*d^10 - 4*B^5*a^4*b^3*c^5*d^8 + 4*B^5*a^5*b^2*c^2*d^11 + 4*B^5*a^5*b^2*c^4*d^9 - B^5*a^6*b*c*d^12 + 6*B^5*a*b^6*c^4*d^9 - 13*B^5*a^2*b^5*c*d^12 + 6*B^5*a^4*b^3*c*d^12 - B^5*a^6*b*c^3*d^10))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((16*(c + d*tan(e + f*x))^(1/2)*(2*B^4*b^9*d^12 - 5*B^4*a^2*b^7*d^12 + 17*B^4*a^4*b^5*d^12 - 7*B^4*a^6*b^3*d^12 + 6*B^4*b^9*c^4*d^8 + B^4*a^8*b*d^12 + 77*B^4*a^2*b^7*c^2*d^10 - 8*B^4*a^2*b^7*c^4*d^8 + 60*B^4*a^3*b^6*c^3*d^9 - 87*B^4*a^4*b^5*c^2*d^10 + 14*B^4*a^4*b^5*c^4*d^8 - 36*B^4*a^5*b^4*c^3*d^9 + 27*B^4*a^6*b^3*c^2*d^10 - 4*B^4*a^6*b^3*c^4*d^8 + 4*B^4*a^7*b^2*c^3*d^9 + 12*B^4*a*b^8*c*d^11 - 28*B^4*a*b^8*c^3*d^9 - 64*B^4*a^3*b^6*c*d^11 + 44*B^4*a^5*b^4*c*d^11 - 8*B^4*a^7*b^2*c*d^11 - B^4*a^8*b*c^2*d^10))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4) - (((8*(156*B^3*a^2*b^9*d^12*f^2 - 16*B^3*a^4*b^7*d^12*f^2 - 120*B^3*a^6*b^5*d^12*f^2 + 48*B^3*a^8*b^3*d^12*f^2 + 12*B^3*b^11*c^2*d^10*f^2 + 12*B^3*b^11*c^4*d^8*f^2 - 4*B^3*a^10*b*d^12*f^2 - 124*B^3*a*b^10*c*d^11*f^2 - 124*B^3*a*b^10*c^3*d^9*f^2 + 224*B^3*a^3*b^8*c*d^11*f^2 + 200*B^3*a^5*b^6*c*d^11*f^2 - 128*B^3*a^7*b^4*c*d^11*f^2 + 20*B^3*a^9*b^2*c*d^11*f^2 - 4*B^3*a^10*b*c^2*d^10*f^2 + 44*B^3*a^2*b^9*c^2*d^10*f^2 - 112*B^3*a^2*b^9*c^4*d^8*f^2 + 224*B^3*a^3*b^8*c^3*d^9*f^2 - 40*B^3*a^4*b^7*c^2*d^10*f^2 - 24*B^3*a^4*b^7*c^4*d^8*f^2 + 200*B^3*a^5*b^6*c^3*d^9*f^2 - 40*B^3*a^6*b^5*c^2*d^10*f^2 + 80*B^3*a^6*b^5*c^4*d^8*f^2 - 128*B^3*a^7*b^4*c^3*d^9*f^2 + 28*B^3*a^8*b^3*c^2*d^10*f^2 - 20*B^3*a^8*b^3*c^4*d^8*f^2 + 20*B^3*a^9*b^2*c^3*d^9*f^2))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((((8*(80*B*a*b^14*d^11*f^4 - 48*B*b^15*c*d^10*f^4 + 384*B*a^3*b^12*d^11*f^4 + 720*B*a^5*b^10*d^

11*f^4 + 640*B*a^7*b^8*d^11*f^4 + 240*B*a^9*b^6*d^11*f^4 - 16*B*a^13*b^2*d^
11*f^4 - 48*B*b^15*c^3*d^8*f^4 + 80*B*a*b^14*c^2*d^9*f^4 - 224*B*a^2*b^13*c
*d^10*f^4 - 400*B*a^4*b^11*c*d^10*f^4 - 320*B*a^6*b^9*c*d^10*f^4 - 80*B*a^8
*b^7*c*d^10*f^4 + 32*B*a^10*b^5*c*d^10*f^4 + 16*B*a^12*b^3*c*d^10*f^4 - 224
*B*a^2*b^13*c^3*d^8*f^4 + 384*B*a^3*b^12*c^2*d^9*f^4 - 400*B*a^4*b^11*c^3*d
^8*f^4 + 720*B*a^5*b^10*c^2*d^9*f^4 - 320*B*a^6*b^9*c^3*d^8*f^4 + 640*B*a^7
*b^8*c^2*d^9*f^4 - 80*B*a^8*b^7*c^3*d^8*f^4 + 240*B*a^9*b^6*c^2*d^9*f^4 + 3
2*B*a^10*b^5*c^3*d^8*f^4 + 16*B*a^12*b^3*c^3*d^8*f^4 - 16*B*a^13*b^2*c^2*d^
9*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5)
 - (4*(c + d*tan(e + f*x))^(1/2)*(4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^
2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9*B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^2*d^2 + 16*
B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*d - 4*B^2*a^5*b*c*d)*(b^10*c*f^2 + 4*a^2*b
^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*
f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))^(1/2)
*(32*b^17*d^10*f^4 + 160*a^2*b^15*d^10*f^4 + 288*a^4*b^13*d^10*f^4 + 160*a^
6*b^11*d^10*f^4 - 160*a^8*b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b
^5*d^10*f^4 - 32*a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2
*d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^
9*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14
*b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336*a^5*b
^12*c*d^9*f^4 + 560*a^7*b^10*c*d^9*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*a^11*b
^6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2*c*d^9*f^4))/((a^8*f^4 +
 b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)*(b^10*c*f^2 + 4*a
^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^
7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)))*
(4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9
*B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^2*d^2 + 16*B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*
d - 4*B^2*a^5*b*c*d)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^
6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3
*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))^(1/2))/(4*(b^10*c*f^2 + 4*a^2*b^8*c*f^
2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6
*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)) - (16*(c + d
*tan(e + f*x))^(1/2)*(44*B^2*a^9*b^4*d^11*f^2 - 168*B^2*a^5*b^8*d^11*f^2 -
40*B^2*a^7*b^6*d^11*f^2 - 20*B^2*a^3*b^10*d^11*f^2 - 4*B^2*a^11*b^2*d^11*f^
2 - 36*B^2*b^13*c^3*d^8*f^2 + 60*B^2*a*b^12*d^11*f^2 - 12*B^2*b^13*c*d^10*f
^2 + 4*B^2*a^12*b*c*d^10*f^2 + 100*B^2*a*b^12*c^2*d^9*f^2 + 120*B^2*a^2*b^1
1*c*d^10*f^2 + 156*B^2*a^4*b^9*c*d^10*f^2 - 112*B^2*a^6*b^7*c*d^10*f^2 - 14
8*B^2*a^8*b^5*c*d^10*f^2 - 8*B^2*a^10*b^3*c*d^10*f^2 + 68*B^2*a^2*b^11*c^3*
d^8*f^2 + 124*B^2*a^3*b^10*c^2*d^9*f^2 + 184*B^2*a^4*b^9*c^3*d^8*f^2 + 8*B^
2*a^5*b^8*c^2*d^9*f^2 + 40*B^2*a^6*b^7*c^3*d^8*f^2 + 24*B^2*a^7*b^6*c^2*d^9
*f^2 - 20*B^2*a^8*b^5*c^3*d^8*f^2 + 20*B^2*a^9*b^4*c^2*d^9*f^2 + 20*B^2*a^1
0*b^3*c^3*d^8*f^2 - 20*B^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 + b^8*f^4 + 4*a^
2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2
 - 8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9*B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^
2*d^2 + 16*B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*d - 4*B^2*a^5*b*c*d)*(b^10*c*f^

2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))^(1/2))/(4*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)))*(4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9*B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^2*d^2 + 16*B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*d - 4*B^2*a^5*b*c*d)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))^(1/2))/(4*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)))*(4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9*B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^2*d^2 + 16*B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*d - 4*B^2*a^5*b*c*d)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))^(1/2))/(4*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)) - (((16*(c + d*tan(e + f*x))^(1/2)*(2*B^4*b^9*d^12 - 5*B^4*a^2*b^7*d^12 + 17*B^4*a^4*b^5*d^12 - 7*B^4*a^6*b^3*d^12 + 6*B^4*b^9*c^4*d^8 + B^4*a^8*b*d^12 + 77*B^4*a^2*b^7*c^2*d^10 - 8*B^4*a^2*b^7*c^4*d^8 + 60*B^4*a^3*b^6*c^3*d^9 - 87*B^4*a^4*b^5*c^2*d^10 + 14*B^4*a^4*b^5*c^4*d^8 - 36*B^4*a^5*b^4*c^3*d^9 + 27*B^4*a^6*b^3*c^2*d^10 - 4*B^4*a^6*b^3*c^4*d^8 + 4*B^4*a^7*b^2*c^3*d^9 + 12*B^4*a*b^8*c*d^11 - 28*B^4*a*b^8*c^3*d^9 - 64*B^4*a^3*b^6*c*d^11 + 44*B^4*a^5*b^4*c*d^11 - 8*B^4*a^7*b^2*c*d^11 - B^4*a^8*b*c^2*d^10))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4) + (((8*(156*B^3*a^2*b^9*d^12*f^2 - 16*B^3*a^4*b^7*d^12*f^2 - 120*B^3*a^6*b^5*d^12*f^2 + 48*B^3*a^8*b^3*d^12*f^2 + 12*B^3*b^11*c^2*d^10*f^2 + 12*B^3*b^11*c^4*d^8*f^2 - 4*B^3*a^10*b*d^12*f^2 - 124*B^3*a*b^10*c*d^11*f^2 - 124*B^3*a*b^10*c^3*d^9*f^2 + 224*B^3*a^3*b^8*c*d^11*f^2 + 200*B^3*a^5*b^6*c*d^11*f^2 - 128*B^3*a^7*b^4*c*d^11*f^2 + 20*B^3*a^9*b^2*c*d^11*f^2 - 4*B^3*a^10*b*c^2*d^10*f^2 + 44*B^3*a^2*b^9*c^2*d^10*f^2 - 112*B^3*a^2*b^9*c^4*d^8*f^2 + 224*B^3*a^3*b^8*c^3*d^9*f^2 - 40*B^3*a^4*b^7*c^2*d^10*f^2 - 24*B^3*a^4*b^7*c^4*d^8*f^2 + 200*B^3*a^5*b^6*c^3*d^9*f^2 - 40*B^3*a^6*b^5*c^2*d^10*f^2 + 80*B^3*a^6*b^5*c^4*d^8*f^2 - 128*B^3*a^7*b^4*c^3*d^9*f^2 + 28*B^3*a^8*b^3*c^2*d^10*f^2 - 20*B^3*a^8*b^3*c^4*d^8*f^2 + 20*B^3*a^9*b^2*c^3*d^9*f^2))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((((8*(80*B*a*b^14*d^11*f^4 - 48*B*b^15*c*d^10*f^4 + 384*B*a^3*b^12*d^11*f^4 + 720*B*a^5*b^10*d^11*f^4 + 640*B*a^7*b^8*d^11*f^4 + 240*B*a^9*b^6*d^11*f^4 - 16*B*a^13*b^2*d^11*f^4 - 48*B*b^15*c^3*d^8*f^4 + 80*B*a*b^14*c^2*d^9*f^4 - 224*B*a^2*b^13*c*d^10*f^4 - 400*B*a^4*b^11*c*d^10*f^4 - 320*B*a^6*b^9*c*d^10*f^4 - 80*B*a^8*b^7*c*d^10*f^4 + 32*B*a^10*b^5*c*d^10*f^4 + 16*B*a^12*b^3*c*d^10*f^4 - 224*B*a^2*b^13*c^3*d^8*f^4 + 384*B*a^3*b^12*c^2*d^9*f^4 - 400*B*a^4*b^11*c^3*d^8*f^4 + 720*B*a^5*b^10*c^2*d^9*f^4 - 320*B*a^6*b^9*c^3*d^8*f^4 + 640*B*a^7*b^8*c^2*d^9*f^4 - 80*B*a^8*b^7*c^3*d^8*f^4 + 240*B*a^9*b^6*c^2*d^9

*f^4 + 32*B*a^10*b^5*c^3*d^8*f^4 + 16*B*a^12*b^3*c^3*d^8*f^4 - 16*B*a^13*b^2*c^2*d^9*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (4*(c + d*tan(e + f*x))^(1/2)*(4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9*B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^2*d^2 + 16*B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*d - 4*B^2*a^5*b*c*d)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))^(1/2)*(32*b^17*d^10*f^4 + 160*a^2*b^15*d^10*f^4 + 288*a^4*b^13*d^10*f^4 + 160*a^6*b^11*d^10*f^4 - 160*a^8*b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b^5*d^10*f^4 - 32*a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^9*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14*b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336*a^5*b^12*c*d^9*f^4 + 560*a^7*b^10*c*d^9*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*a^11*b^6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2*c*d^9*f^4))/((a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)))*(4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9*B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^2*d^2 + 16*B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*d - 4*B^2*a^5*b*c*d)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))^(1/2))/(4*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)) + (16*(c + d*tan(e + f*x))^(1/2)*(44*B^2*a^9*b^4*d^11*f^2 - 168*B^2*a^5*b^8*d^11*f^2 - 40*B^2*a^7*b^6*d^11*f^2 - 20*B^2*a^3*b^10*d^11*f^2 - 4*B^2*a^11*b^2*d^11*f^2 - 36*B^2*b^13*c^3*d^8*f^2 + 60*B^2*a*b^12*d^11*f^2 - 12*B^2*b^13*c*d^10*f^2 + 4*B^2*a^12*b*c*d^10*f^2 + 100*B^2*a*b^12*c^2*d^9*f^2 + 120*B^2*a^2*b^11*c*d^10*f^2 + 156*B^2*a^4*b^9*c*d^10*f^2 - 112*B^2*a^6*b^7*c*d^10*f^2 - 148*B^2*a^8*b^5*c*d^10*f^2 - 8*B^2*a^10*b^3*c*d^10*f^2 + 68*B^2*a^2*b^11*c^3*d^8*f^2 + 124*B^2*a^3*b^10*c^2*d^9*f^2 + 184*B^2*a^4*b^9*c^3*d^8*f^2 + 8*B^2*a^5*b^8*c^2*d^9*f^2 + 40*B^2*a^6*b^7*c^3*d^8*f^2 + 24*B^2*a^7*b^6*c^2*d^9*f^2 - 20*B^2*a^8*b^5*c^3*d^8*f^2 + 20*B^2*a^9*b^4*c^2*d^9*f^2 + 20*B^2*a^10*b^3*c^3*d^8*f^2 - 20*B^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9*B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^2*d^2 + 16*B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*d - 4*B^2*a^5*b*c*d)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))^(1/2))/(4*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)))*(4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9*B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^2*d^2 + 16*B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*d - 4*B^2*a^5*b*c*d)*(b^10*c*f^2 +

4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))^(1/2))/(4*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)))*(4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9*B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^2*d^2 + 16*B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*d - 4*B^2*a^5*b*c*d)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))^(1/2))/(4*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))))*(4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9*B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^2*d^2 + 16*B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*d - 4*B^2*a^5*b*c*d)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))^(1/2)*1i)/(2*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)) - (atan(((((((8*(128*A^3*a^3*b^8*d^12*f^2 + 24*A^3*a^5*b^6*d^12*f^2 - 160*A^3*a^7*b^4*d^12*f^2 - 4*A^3*a^9*b^2*d^12*f^2 + 20*A^3*b^11*c^3*d^9*f^2 - 52*A^3*a*b^10*d^12*f^2 + 20*A^3*b^11*c*d^11*f^2 + 12*A^3*a*b^10*c^2*d^10*f^2 + 64*A^3*a*b^10*c^4*d^8*f^2 - 256*A^3*a^2*b^9*c*d^11*f^2 + 72*A^3*a^4*b^7*c*d^11*f^2 + 352*A^3*a^6*b^5*c*d^11*f^2 + 4*A^3*a^8*b^3*c*d^11*f^2 - 256*A^3*a^2*b^9*c^3*d^9*f^2 - 128*A^3*a^3*b^8*c^4*d^8*f^2 + 72*A^3*a^4*b^7*c^3*d^9*f^2 - 168*A^3*a^5*b^6*c^2*d^10*f^2 - 192*A^3*a^5*b^6*c^4*d^8*f^2 + 352*A^3*a^6*b^5*c^3*d^9*f^2 - 160*A^3*a^7*b^4*c^2*d^10*f^2 + 4*A^3*a^8*b^3*c^3*d^9*f^2 - 4*A^3*a^9*b^2*c^2*d^10*f^2))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((((8*(32*A*b^15*d^11*f^4 + 96*A*a^2*b^13*d^11*f^4 - 320*A*a^6*b^9*d^11*f^4 - 480*A*a^8*b^7*d^11*f^4 - 288*A*a^10*b^5*d^11*f^4 - 64*A*a^12*b^3*d^11*f^4 + 32*A*b^15*c^2*d^9*f^4 + 64*A*a*b^14*c^3*d^8*f^4 + 320*A*a^3*b^12*c*d^10*f^4 + 640*A*a^5*b^10*c*d^10*f^4 + 640*A*a^7*b^8*c*d^10*f^4 + 320*A*a^9*b^6*c*d^10*f^4 + 64*A*a^11*b^4*c*d^10*f^4 + 96*A*a^2*b^13*c^2*d^9*f^4 + 320*A*a^3*b^12*c^3*d^8*f^4 + 640*A*a^5*b^10*c^3*d^8*f^4 - 320*A*a^6*b^9*c^2*d^9*f^4 + 640*A*a^7*b^8*c^3*d^8*f^4 - 480*A*a^8*b^7*c^2*d^9*f^4 + 320*A*a^9*b^6*c^3*d^8*f^4 - 288*A*a^10*b^5*c^2*d^9*f^4 + 64*A*a^11*b^4*c^3*d^8*f^4 - 64*A*a^12*b^3*c^2*d^9*f^4 + 64*A*a*b^14*c*d^10*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) - (4*(c + d*tan(e + f*x))^(1/2)*(-4*(A^2*b^5*d^2 + 16*A^2*a^2*b^3*c^2 - 6*A^2*a^2*b^3*d^2 + 9*A^2*a^4*b*d^2 - 24*A^2*a^3*b^2*c*d + 8*A^2*a*b^4*c*d)*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2))^(1/2)*(32*b^17*d^10*f^4 + 160*a^2*b^15*d^10*f^4 + 288*a^4*b^13*d^10*f^4 + 160*a^6*b^11*d^10*f^4 - 160*a^8*b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b^5*d^10*f^4 - 32*a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^4 + 720*a^6*b^11*

c^2*d^8*f^4 + 400*a^8*b^9*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b
^5*c^2*d^8*f^4 - 16*a^14*b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9*f^4 + 112*a^3*b^
14*c*d^9*f^4 + 336*a^5*b^12*c*d^9*f^4 + 560*a^7*b^10*c*d^9*f^4 + 560*a^9*b^
8*c*d^9*f^4 + 336*a^11*b^6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2
*c*d^9*f^4))/((a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^
2*f^4)*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b
^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^
2 + a*b^8*d*f^2)))*(-4*(A^2*b^5*d^2 + 16*A^2*a^2*b^3*c^2 - 6*A^2*a^2*b^3*d^
2 + 9*A^2*a^4*b*d^2 - 24*A^2*a^3*b^2*c*d + 8*A^2*a*b^4*c*d)*(a^9*d*f^2 - b^
9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d
*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2))^(1/2
))/(4*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^
3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2
 + a*b^8*d*f^2)) + (16*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^10*d^11*f^2
 - 88*A^2*a^5*b^8*d^11*f^2 + 40*A^2*a^7*b^6*d^11*f^2 + 84*A^2*a^9*b^4*d^11*
f^2 + 4*A^2*a^11*b^2*d^11*f^2 - 20*A^2*b^13*c^3*d^8*f^2 + 68*A^2*a*b^12*d^1
1*f^2 - 8*A^2*b^13*c*d^10*f^2 + 116*A^2*a*b^12*c^2*d^9*f^2 + 104*A^2*a^2*b^
11*c*d^10*f^2 + 48*A^2*a^4*b^9*c*d^10*f^2 - 304*A^2*a^6*b^7*c*d^10*f^2 - 29
6*A^2*a^8*b^5*c*d^10*f^2 - 56*A^2*a^10*b^3*c*d^10*f^2 + 116*A^2*a^2*b^11*c^
3*d^8*f^2 + 204*A^2*a^3*b^10*c^2*d^9*f^2 + 216*A^2*a^4*b^9*c^3*d^8*f^2 + 16
8*A^2*a^5*b^8*c^2*d^9*f^2 + 8*A^2*a^6*b^7*c^3*d^8*f^2 + 184*A^2*a^7*b^6*c^2
*d^9*f^2 - 68*A^2*a^8*b^5*c^3*d^8*f^2 + 100*A^2*a^9*b^4*c^2*d^9*f^2 + 4*A^2
*a^10*b^3*c^3*d^8*f^2 - 4*A^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 + b^8*f^4 + 4
*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-4*(A^2*b^5*d^2 + 16*A^2*a^
2*b^3*c^2 - 6*A^2*a^2*b^3*d^2 + 9*A^2*a^4*b*d^2 - 24*A^2*a^3*b^2*c*d + 8*A^
2*a*b^4*c*d)*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4
*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*
b*c*f^2 + a*b^8*d*f^2))^(1/2))/(4*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2
 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4
*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2)))*(-4*(A^2*b^5*d^2 + 16*A^2*a^2
*b^3*c^2 - 6*A^2*a^2*b^3*d^2 + 9*A^2*a^4*b*d^2 - 24*A^2*a^3*b^2*c*d + 8*A^2
*a*b^4*c*d)*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*
a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b
*c*f^2 + a*b^8*d*f^2))^(1/2))/(4*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 -
 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*
a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2)) - (16*(c + d*tan(e + f*x))^(1/2
)*(3*A^4*b^9*d^12 - 3*A^4*a^2*b^7*d^12 + 17*A^4*a^4*b^5*d^12 - 9*A^4*a^6*b^
3*d^12 + 3*A^4*b^9*c^2*d^10 + 2*A^4*b^9*c^4*d^8 + 63*A^4*a^2*b^7*c^2*d^10 -
 12*A^4*a^2*b^7*c^4*d^8 + 96*A^4*a^3*b^6*c^3*d^9 - 123*A^4*a^4*b^5*c^2*d^10
 + 18*A^4*a^4*b^5*c^4*d^8 - 24*A^4*a^5*b^4*c^3*d^9 + 9*A^4*a^6*b^3*c^2*d^10
 + 12*A^4*a*b^8*c*d^11 - 8*A^4*a*b^8*c^3*d^9 - 56*A^4*a^3*b^6*c*d^11 + 60*A
^4*a^5*b^4*c*d^11))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*
a^6*b^2*f^4))*(-4*(A^2*b^5*d^2 + 16*A^2*a^2*b^3*c^2 - 6*A^2*a^2*b^3*d^2 + 9
*A^2*a^4*b*d^2 - 24*A^2*a^3*b^2*c*d + 8*A^2*a*b^4*c*d)*(a^9*d*f^2 - b^9*c*f
^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2

+ 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2))^(1/2)*1i)
/(4*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*
c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 +
a*b^8*d*f^2)) - (((((8*(128*A^3*a^3*b^8*d^12*f^2 + 24*A^3*a^5*b^6*d^12*f^2
- 160*A^3*a^7*b^4*d^12*f^2 - 4*A^3*a^9*b^2*d^12*f^2 + 20*A^3*b^11*c^3*d^9*
f^2 - 52*A^3*a*b^10*d^12*f^2 + 20*A^3*b^11*c*d^11*f^2 + 12*A^3*a*b^10*c^2*d
^10*f^2 + 64*A^3*a*b^10*c^4*d^8*f^2 - 256*A^3*a^2*b^9*c*d^11*f^2 + 72*A^3*a
^4*b^7*c*d^11*f^2 + 352*A^3*a^6*b^5*c*d^11*f^2 + 4*A^3*a^8*b^3*c*d^11*f^2 -
256*A^3*a^2*b^9*c^3*d^9*f^2 - 128*A^3*a^3*b^8*c^4*d^8*f^2 + 72*A^3*a^4*b^7
*c^3*d^9*f^2 - 168*A^3*a^5*b^6*c^2*d^10*f^2 - 192*A^3*a^5*b^6*c^4*d^8*f^2 +
352*A^3*a^6*b^5*c^3*d^9*f^2 - 160*A^3*a^7*b^4*c^2*d^10*f^2 + 4*A^3*a^8*b^3
*c^3*d^9*f^2 - 4*A^3*a^9*b^2*c^2*d^10*f^2))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*
f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((((8*(32*A*b^15*d^11*f^4 + 96*A*a^
2*b^13*d^11*f^4 - 320*A*a^6*b^9*d^11*f^4 - 480*A*a^8*b^7*d^11*f^4 - 288*A*a
^10*b^5*d^11*f^4 - 64*A*a^12*b^3*d^11*f^4 + 32*A*b^15*c^2*d^9*f^4 + 64*A*a*
b^14*c^3*d^8*f^4 + 320*A*a^3*b^12*c*d^10*f^4 + 640*A*a^5*b^10*c*d^10*f^4 +
640*A*a^7*b^8*c*d^10*f^4 + 320*A*a^9*b^6*c*d^10*f^4 + 64*A*a^11*b^4*c*d^10*
f^4 + 96*A*a^2*b^13*c^2*d^9*f^4 + 320*A*a^3*b^12*c^3*d^8*f^4 + 640*A*a^5*b^
10*c^3*d^8*f^4 - 320*A*a^6*b^9*c^2*d^9*f^4 + 640*A*a^7*b^8*c^3*d^8*f^4 - 48
0*A*a^8*b^7*c^2*d^9*f^4 + 320*A*a^9*b^6*c^3*d^8*f^4 - 288*A*a^10*b^5*c^2*d^
9*f^4 + 64*A*a^11*b^4*c^3*d^8*f^4 - 64*A*a^12*b^3*c^2*d^9*f^4 + 64*A*a*b^14
*c*d^10*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^
2*f^5) + (4*(c + d*tan(e + f*x))^(1/2)*(-4*(A^2*b^5*d^2 + 16*A^2*a^2*b^3*c^
2 - 6*A^2*a^2*b^3*d^2 + 9*A^2*a^4*b*d^2 - 24*A^2*a^3*b^2*c*d + 8*A^2*a*b^4*
c*d)*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3
*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2
+ a*b^8*d*f^2))^(1/2)*(32*b^17*d^10*f^4 + 160*a^2*b^15*d^10*f^4 + 288*a^4*b
^13*d^10*f^4 + 160*a^6*b^11*d^10*f^4 - 160*a^8*b^9*d^10*f^4 - 288*a^10*b^7*
d^10*f^4 - 160*a^12*b^5*d^10*f^4 - 32*a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^8*f
^4 + 272*a^2*b^15*c^2*d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^4 + 720*a^6*b^11*c^2
*d^8*f^4 + 400*a^8*b^9*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b^5*
c^2*d^8*f^4 - 16*a^14*b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9*f^4 + 112*a^3*b^14*
c*d^9*f^4 + 336*a^5*b^12*c*d^9*f^4 + 560*a^7*b^10*c*d^9*f^4 + 560*a^9*b^8*c
*d^9*f^4 + 336*a^11*b^6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2*c*
d^9*f^4))/((a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f
^4)*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*
c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 +
a*b^8*d*f^2)))*(-4*(A^2*b^5*d^2 + 16*A^2*a^2*b^3*c^2 - 6*A^2*a^2*b^3*d^2 +
9*A^2*a^4*b*d^2 - 24*A^2*a^3*b^2*c*d + 8*A^2*a*b^4*c*d)*(a^9*d*f^2 - b^9*c
*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^
2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2))^(1/2))/
(4*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c
*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 +
a*b^8*d*f^2)) - (16*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^10*d^11*f^2 -
88*A^2*a^5*b^8*d^11*f^2 + 40*A^2*a^7*b^6*d^11*f^2 + 84*A^2*a^9*b^4*d^11*f^2

+ 4*A^2*a^11*b^2*d^11*f^2 - 20*A^2*b^13*c^3*d^8*f^2 + 68*A^2*a*b^12*d^11*f
^2 - 8*A^2*b^13*c*d^10*f^2 + 116*A^2*a*b^12*c^2*d^9*f^2 + 104*A^2*a^2*b^11*
c*d^10*f^2 + 48*A^2*a^4*b^9*c*d^10*f^2 - 304*A^2*a^6*b^7*c*d^10*f^2 - 296*A
^2*a^8*b^5*c*d^10*f^2 - 56*A^2*a^10*b^3*c*d^10*f^2 + 116*A^2*a^2*b^11*c^3*d
^8*f^2 + 204*A^2*a^3*b^10*c^2*d^9*f^2 + 216*A^2*a^4*b^9*c^3*d^8*f^2 + 168*A
^2*a^5*b^8*c^2*d^9*f^2 + 8*A^2*a^6*b^7*c^3*d^8*f^2 + 184*A^2*a^7*b^6*c^2*d^
9*f^2 - 68*A^2*a^8*b^5*c^3*d^8*f^2 + 100*A^2*a^9*b^4*c^2*d^9*f^2 + 4*A^2*a^
10*b^3*c^3*d^8*f^2 - 4*A^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 + b^8*f^4 + 4*a^
2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-4*(A^2*b^5*d^2 + 16*A^2*a^2*b
^3*c^2 - 6*A^2*a^2*b^3*d^2 + 9*A^2*a^4*b*d^2 - 24*A^2*a^3*b^2*c*d + 8*A^2*a
*b^4*c*d)*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^
6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c
*f^2 + a*b^8*d*f^2))^(1/2))/(4*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6
*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^
7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2)))*(-4*(A^2*b^5*d^2 + 16*A^2*a^2*b^
3*c^2 - 6*A^2*a^2*b^3*d^2 + 9*A^2*a^4*b*d^2 - 24*A^2*a^3*b^2*c*d + 8*A^2*a*
b^4*c*d)*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6
*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*
f^2 + a*b^8*d*f^2))^(1/2))/(4*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*
a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7
*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2)) + (16*(c + d*tan(e + f*x))^(1/2)*(
3*A^4*b^9*d^12 - 3*A^4*a^2*b^7*d^12 + 17*A^4*a^4*b^5*d^12 - 9*A^4*a^6*b^3*d
^12 + 3*A^4*b^9*c^2*d^10 + 2*A^4*b^9*c^4*d^8 + 63*A^4*a^2*b^7*c^2*d^10 - 12
*A^4*a^2*b^7*c^4*d^8 + 96*A^4*a^3*b^6*c^3*d^9 - 123*A^4*a^4*b^5*c^2*d^10 +
18*A^4*a^4*b^5*c^4*d^8 - 24*A^4*a^5*b^4*c^3*d^9 + 9*A^4*a^6*b^3*c^2*d^10 +
12*A^4*a*b^8*c*d^11 - 8*A^4*a*b^8*c^3*d^9 - 56*A^4*a^3*b^6*c*d^11 + 60*A^4*
a^5*b^4*c*d^11))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6
*b^2*f^4))*(-4*(A^2*b^5*d^2 + 16*A^2*a^2*b^3*c^2 - 6*A^2*a^2*b^3*d^2 + 9*A^
2*a^4*b*d^2 - 24*A^2*a^3*b^2*c*d + 8*A^2*a*b^4*c*d)*(a^9*d*f^2 - b^9*c*f^2
- 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6
*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2))^(1/2)*1i)/(4
*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f
^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*
b^8*d*f^2)))/((16*(A^5*b^7*d^13 - 9*A^5*a^4*b^3*d^13 + 3*A^5*b^7*c^2*d^11 +
2*A^5*b^7*c^4*d^9 - 22*A^5*a^2*b^5*c^2*d^11 - 22*A^5*a^2*b^5*c^4*d^9 + 24*
A^5*a^3*b^4*c^3*d^10 - 9*A^5*a^4*b^3*c^2*d^11 + 8*A^5*a*b^6*c^3*d^10 + 8*A^
5*a*b^6*c^5*d^8 + 24*A^5*a^3*b^4*c*d^12))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^
5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((((8*(128*A^3*a^3*b^8*d^12*f^2 + 24*
A^3*a^5*b^6*d^12*f^2 - 160*A^3*a^7*b^4*d^12*f^2 - 4*A^3*a^9*b^2*d^12*f^2 +
20*A^3*b^11*c^3*d^9*f^2 - 52*A^3*a*b^10*d^12*f^2 + 20*A^3*b^11*c*d^11*f^2 +
12*A^3*a*b^10*c^2*d^10*f^2 + 64*A^3*a*b^10*c^4*d^8*f^2 - 256*A^3*a^2*b^9*c
*d^11*f^2 + 72*A^3*a^4*b^7*c*d^11*f^2 + 352*A^3*a^6*b^5*c*d^11*f^2 + 4*A^3*
a^8*b^3*c*d^11*f^2 - 256*A^3*a^2*b^9*c^3*d^9*f^2 - 128*A^3*a^3*b^8*c^4*d^8*
f^2 + 72*A^3*a^4*b^7*c^3*d^9*f^2 - 168*A^3*a^5*b^6*c^2*d^10*f^2 - 192*A^3*a
^5*b^6*c^4*d^8*f^2 + 352*A^3*a^6*b^5*c^3*d^9*f^2 - 160*A^3*a^7*b^4*c^2*d^10

*f^2 + 4*A^3*a^8*b^3*c^3*d^9*f^2 - 4*A^3*a^9*b^2*c^2*d^10*f^2))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((((8*(32*A*b^15*d^11*f^4 + 96*A*a^2*b^13*d^11*f^4 - 320*A*a^6*b^9*d^11*f^4 - 480*A*a^8*b^7*d^11*f^4 - 288*A*a^10*b^5*d^11*f^4 - 64*A*a^12*b^3*d^11*f^4 + 32*A*b^15*c^2*d^9*f^4 + 64*A*a*b^14*c^3*d^8*f^4 + 320*A*a^3*b^12*c*d^10*f^4 + 640*A*a^5*b^10*c*d^10*f^4 + 640*A*a^7*b^8*c*d^10*f^4 + 320*A*a^9*b^6*c*d^10*f^4 + 64*A*a^11*b^4*c*d^10*f^4 + 96*A*a^2*b^13*c^2*d^9*f^4 + 320*A*a^3*b^12*c^3*d^8*f^4 + 640*A*a^5*b^10*c^3*d^8*f^4 - 320*A*a^6*b^9*c^2*d^9*f^4 + 640*A*a^7*b^8*c^3*d^8*f^4 - 480*A*a^8*b^7*c^2*d^9*f^4 + 320*A*a^9*b^6*c^3*d^8*f^4 - 288*A*a^10*b^5*c^2*d^9*f^4 + 64*A*a^11*b^4*c^3*d^8*f^4 - 64*A*a^12*b^3*c^2*d^9*f^4 + 64*A*a*b^14*c*d^10*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) - (4*(c + d*tan(e + f*x))^(1/2)*(-4*(A^2*b^5*d^2 + 16*A^2*a^2*b^3*c^2 - 6*A^2*a^2*b^3*d^2 + 9*A^2*a^4*b*d^2 - 24*A^2*a^3*b^2*c*d + 8*A^2*a*b^4*c*d)*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2))^(1/2)*(32*b^17*d^10*f^4 + 160*a^2*b^15*d^10*f^4 + 288*a^4*b^13*d^10*f^4 + 160*a^6*b^11*d^10*f^4 - 160*a^8*b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b^5*d^10*f^4 - 32*a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^9*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14*b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336*a^5*b^12*c*d^9*f^4 + 560*a^7*b^10*c*d^9*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*a^11*b^6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2*c*d^9*f^4))/((a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2)))*(-4*(A^2*b^5*d^2 + 16*A^2*a^2*b^3*c^2 - 6*A^2*a^2*b^3*d^2 + 9*A^2*a^4*b*d^2 - 24*A^2*a^3*b^2*c*d + 8*A^2*a*b^4*c*d)*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2))^(1/2))/(4*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2)) + (16*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^10*d^11*f^2 - 88*A^2*a^5*b^8*d^11*f^2 + 40*A^2*a^7*b^6*d^11*f^2 + 84*A^2*a^9*b^4*d^11*f^2 + 4*A^2*a^11*b^2*d^11*f^2 - 20*A^2*b^13*c^3*d^8*f^2 + 68*A^2*a*b^12*d^11*f^2 - 8*A^2*b^13*c*d^10*f^2 + 116*A^2*a*b^12*c^2*d^9*f^2 + 104*A^2*a^2*b^11*c*d^10*f^2 + 48*A^2*a^4*b^9*c*d^10*f^2 - 304*A^2*a^6*b^7*c*d^10*f^2 - 296*A^2*a^8*b^5*c*d^10*f^2 - 56*A^2*a^10*b^3*c*d^10*f^2 + 116*A^2*a^2*b^11*c^3*d^8*f^2 + 204*A^2*a^3*b^10*c^2*d^9*f^2 + 216*A^2*a^4*b^9*c^3*d^8*f^2 + 168*A^2*a^5*b^8*c^2*d^9*f^2 + 8*A^2*a^6*b^7*c^3*d^8*f^2 + 184*A^2*a^7*b^6*c^2*d^9*f^2 - 68*A^2*a^8*b^5*c^3*d^8*f^2 + 100*A^2*a^9*b^4*c^2*d^9*f^2 + 4*A^2*a^10*b^3*c^3*d^8*f^2 - 4*A^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-4*(A^2*b^5*d^2 + 16*A^2*a^2*b^3*c^2 - 6*A^2*a^2*b^3*d^2 + 9*A^2*a^4*b*d^2 - 24*A^2*a^3*b^2*c*d + 8*A^2*a*b^4*c*d)*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*

a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7
*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2))^(1/2))/(4*(a^9*d*f^2 - b^9*c*f^2 -
4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*
a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2)))*(-4*(A^2*b^5
*d^2 + 16*A^2*a^2*b^3*c^2 - 6*A^2*a^2*b^3*d^2 + 9*A^2*a^4*b*d^2 - 24*A^2*a^
3*b^2*c*d + 8*A^2*a*b^4*c*d)*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a
^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*
b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2))^(1/2))/(4*(a^9*d*f^2 - b^9*c*f^2 -
4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a
^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2)) - (16*(c + d*t
an(e + f*x))^(1/2)*(3*A^4*b^9*d^12 - 3*A^4*a^2*b^7*d^12 + 17*A^4*a^4*b^5*d^
12 - 9*A^4*a^6*b^3*d^12 + 3*A^4*b^9*c^2*d^10 + 2*A^4*b^9*c^4*d^8 + 63*A^4*a
^2*b^7*c^2*d^10 - 12*A^4*a^2*b^7*c^4*d^8 + 96*A^4*a^3*b^6*c^3*d^9 - 123*A^4
*a^4*b^5*c^2*d^10 + 18*A^4*a^4*b^5*c^4*d^8 - 24*A^4*a^5*b^4*c^3*d^9 + 9*A^4
*a^6*b^3*c^2*d^10 + 12*A^4*a*b^8*c*d^11 - 8*A^4*a*b^8*c^3*d^9 - 56*A^4*a^3*
b^6*c*d^11 + 60*A^4*a^5*b^4*c*d^11))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6
*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-4*(A^2*b^5*d^2 + 16*A^2*a^2*b^3*c^2 - 6*A^
2*a^2*b^3*d^2 + 9*A^2*a^4*b*d^2 - 24*A^2*a^3*b^2*c*d + 8*A^2*a*b^4*c*d)*(a^
9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 +
4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*
d*f^2))^(1/2))/(4*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^
2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 -
a^8*b*c*f^2 + a*b^8*d*f^2)) + (((((8*(128*A^3*a^3*b^8*d^12*f^2 + 24*A^3*a^
5*b^6*d^12*f^2 - 160*A^3*a^7*b^4*d^12*f^2 - 4*A^3*a^9*b^2*d^12*f^2 + 20*A^3
*b^11*c^3*d^9*f^2 - 52*A^3*a*b^10*d^12*f^2 + 20*A^3*b^11*c*d^11*f^2 + 12*A^
3*a*b^10*c^2*d^10*f^2 + 64*A^3*a*b^10*c^4*d^8*f^2 - 256*A^3*a^2*b^9*c*d^11*
f^2 + 72*A^3*a^4*b^7*c*d^11*f^2 + 352*A^3*a^6*b^5*c*d^11*f^2 + 4*A^3*a^8*b^
3*c*d^11*f^2 - 256*A^3*a^2*b^9*c^3*d^9*f^2 - 128*A^3*a^3*b^8*c^4*d^8*f^2 +
72*A^3*a^4*b^7*c^3*d^9*f^2 - 168*A^3*a^5*b^6*c^2*d^10*f^2 - 192*A^3*a^5*b^6
*c^4*d^8*f^2 + 352*A^3*a^6*b^5*c^3*d^9*f^2 - 160*A^3*a^7*b^4*c^2*d^10*f^2 +
4*A^3*a^8*b^3*c^3*d^9*f^2 - 4*A^3*a^9*b^2*c^2*d^10*f^2))/(a^8*f^5 + b^8*f^
5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((((8*(32*A*b^15*d^11
*f^4 + 96*A*a^2*b^13*d^11*f^4 - 320*A*a^6*b^9*d^11*f^4 - 480*A*a^8*b^7*d^11
*f^4 - 288*A*a^10*b^5*d^11*f^4 - 64*A*a^12*b^3*d^11*f^4 + 32*A*b^15*c^2*d^9
*f^4 + 64*A*a*b^14*c^3*d^8*f^4 + 320*A*a^3*b^12*c*d^10*f^4 + 640*A*a^5*b^10
*c*d^10*f^4 + 640*A*a^7*b^8*c*d^10*f^4 + 320*A*a^9*b^6*c*d^10*f^4 + 64*A*a^
11*b^4*c*d^10*f^4 + 96*A*a^2*b^13*c^2*d^9*f^4 + 320*A*a^3*b^12*c^3*d^8*f^4
+ 640*A*a^5*b^10*c^3*d^8*f^4 - 320*A*a^6*b^9*c^2*d^9*f^4 + 640*A*a^7*b^8*c^
3*d^8*f^4 - 480*A*a^8*b^7*c^2*d^9*f^4 + 320*A*a^9*b^6*c^3*d^8*f^4 - 288*A*a
^10*b^5*c^2*d^9*f^4 + 64*A*a^11*b^4*c^3*d^8*f^4 - 64*A*a^12*b^3*c^2*d^9*f^4
+ 64*A*a*b^14*c*d^10*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*
f^5 + 4*a^6*b^2*f^5) + (4*(c + d*tan(e + f*x))^(1/2)*(-4*(A^2*b^5*d^2 + 16*
A^2*a^2*b^3*c^2 - 6*A^2*a^2*b^3*d^2 + 9*A^2*a^4*b*d^2 - 24*A^2*a^3*b^2*c*d
+ 8*A^2*a*b^4*c*d)*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f
^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2

- a^8*b*c*f^2 + a*b^8*d*f^2))^(1/2)*(32*b^17*d^10*f^4 + 160*a^2*b^15*d^10*f
^4 + 288*a^4*b^13*d^10*f^4 + 160*a^6*b^11*d^10*f^4 - 160*a^8*b^9*d^10*f^4 -
288*a^10*b^7*d^10*f^4 - 160*a^12*b^5*d^10*f^4 - 32*a^14*b^3*d^10*f^4 + 48*
b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^4 + 72
0*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^9*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^8*f^4
- 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14*b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9*f^4 +
112*a^3*b^14*c*d^9*f^4 + 336*a^5*b^12*c*d^9*f^4 + 560*a^7*b^10*c*d^9*f^4 +
560*a^9*b^8*c*d^9*f^4 + 336*a^11*b^6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*f^4 +
16*a^15*b^2*c*d^9*f^4))/((a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4
+ 4*a^6*b^2*f^4)*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^
2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 -
a^8*b*c*f^2 + a*b^8*d*f^2)))*(-4*(A^2*b^5*d^2 + 16*A^2*a^2*b^3*c^2 - 6*A^2
*a^2*b^3*d^2 + 9*A^2*a^4*b*d^2 - 24*A^2*a^3*b^2*c*d + 8*A^2*a*b^4*c*d)*(a^9
*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 +
4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d
f^2))^(1/2))/(4(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2
- 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 -
a^8*b*c*f^2 + a*b^8*d*f^2)) - (16*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^
10*d^11*f^2 - 88*A^2*a^5*b^8*d^11*f^2 + 40*A^2*a^7*b^6*d^11*f^2 + 84*A^2*a^
9*b^4*d^11*f^2 + 4*A^2*a^11*b^2*d^11*f^2 - 20*A^2*b^13*c^3*d^8*f^2 + 68*A^2
*a*b^12*d^11*f^2 - 8*A^2*b^13*c*d^10*f^2 + 116*A^2*a*b^12*c^2*d^9*f^2 + 104
*A^2*a^2*b^11*c*d^10*f^2 + 48*A^2*a^4*b^9*c*d^10*f^2 - 304*A^2*a^6*b^7*c*d^
10*f^2 - 296*A^2*a^8*b^5*c*d^10*f^2 - 56*A^2*a^10*b^3*c*d^10*f^2 + 116*A^2*
a^2*b^11*c^3*d^8*f^2 + 204*A^2*a^3*b^10*c^2*d^9*f^2 + 216*A^2*a^4*b^9*c^3*d
^8*f^2 + 168*A^2*a^5*b^8*c^2*d^9*f^2 + 8*A^2*a^6*b^7*c^3*d^8*f^2 + 184*A^2*
a^7*b^6*c^2*d^9*f^2 - 68*A^2*a^8*b^5*c^3*d^8*f^2 + 100*A^2*a^9*b^4*c^2*d^9*
f^2 + 4*A^2*a^10*b^3*c^3*d^8*f^2 - 4*A^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 +
b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-4*(A^2*b^5*d^2
+ 16*A^2*a^2*b^3*c^2 - 6*A^2*a^2*b^3*d^2 + 9*A^2*a^4*b*d^2 - 24*A^2*a^3*b^2
*c*d + 8*A^2*a*b^4*c*d)*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^
5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d
*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2))^(1/2))/(4*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2
*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^
4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2)))*(-4*(A^2*b^5*d^2 +
16*A^2*a^2*b^3*c^2 - 6*A^2*a^2*b^3*d^2 + 9*A^2*a^4*b*d^2 - 24*A^2*a^3*b^2*
c*d + 8*A^2*a*b^4*c*d)*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5
*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*
f^2 - a^8*b*c*f^2 + a*b^8*d*f^2))^(1/2))/(4*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*
b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4
*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2)) + (16*(c + d*tan(e +
f*x))^(1/2)*(3*A^4*b^9*d^12 - 3*A^4*a^2*b^7*d^12 + 17*A^4*a^4*b^5*d^12 - 9
*A^4*a^6*b^3*d^12 + 3*A^4*b^9*c^2*d^10 + 2*A^4*b^9*c^4*d^8 + 63*A^4*a^2*b^7
*c^2*d^10 - 12*A^4*a^2*b^7*c^4*d^8 + 96*A^4*a^3*b^6*c^3*d^9 - 123*A^4*a^4*b
^5*c^2*d^10 + 18*A^4*a^4*b^5*c^4*d^8 - 24*A^4*a^5*b^4*c^3*d^9 + 9*A^4*a^6*b
^3*c^2*d^10 + 12*A^4*a*b^8*c*d^11 - 8*A^4*a*b^8*c^3*d^9 - 56*A^4*a^3*b^6*c*

d^11 + 60*A^4*a^5*b^4*c*d^11))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b ^4*f^4 + 4*a^6*b^2*f^4))*(-4*(A^2*b^5*d^2 + 16*A^2*a^2*b^3*c^2 - 6*A^2*a^2* b^3*d^2 + 9*A^2*a^4*b*d^2 - 24*A^2*a^3*b^2*c*d + 8*A^2*a*b^4*c*d)*(a^9*d*f^ 2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3 *b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2))^(1/2))/(4*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4* a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b *c*f^2 + a*b^8*d*f^2))))*(-4*(A^2*b^5*d^2 + 16*A^2*a^2*b^3*c^2 - 6*A^2*a^2* b^3*d^2 + 9*A^2*a^4*b*d^2 - 24*A^2*a^3*b^2*c*d + 8*A^2*a*b^4*c*d)*(a^9*d*f^ 2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3 *b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2))^(1/2)*1i)/(2*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^ 8*b*c*f^2 + a*b^8*d*f^2)) - (A*b*d*(c + d*tan(e + f*x))^(1/2))/((a^2 + b^2) *(b*f*(c + d*tan(e + f*x)) + a*d*f - b*c*f)) + (B*a*d*(c + d*tan(e + f*x))^ (1/2))/((a^2 + b^2)*(b*f*(c + d*tan(e + f*x)) + a*d*f - b*c*f)) - (C*a^2*d* (c + d*tan(e + f*x))^(1/2))/(b*(a^2 + b^2)*(b*f*(c + d*tan(e + f*x)) + a*d* f - b*c*f))

3.96 $\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx$

Optimal result 1266
Rubi [A] (verified) 1267
Mathematica [B] (verified) 1271
Maple [B] (verified) 1273
Fricas [F(-1)] 1273
Sympy [F] 1273
Maxima [F(-2)] 1274
Giac [F(-1)] 1274
Mupad [F(-1)] 1274

Optimal result

Integrand size = 47, antiderivative size = 543

$$
\begin{aligned}
&\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx \\
&= -\frac{(A-iB-C)\sqrt{c-id}\,\mathrm{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(ia+b)^3f} \\
&\quad+\frac{(A+iB-C)\sqrt{c+id}\,\mathrm{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(ia-b)^3f} \\
&\quad+\frac{(3a^5bBd^2+a^6Cd^2-3a^4b^2d(4Bc+5Ad-6Cd)-3a^2b^4(8Ac^2-8c^2C-16Bcd-6Ad^2+5Cd^2)+2a^3b\cdots}{\cdots} \\
&\quad-\frac{(Ab^2-a(bB-aC))\sqrt{c+d\tan(e+fx)}}{2b(a^2+b^2)f(a+b\tan(e+fx))^2} \\
&\quad-\frac{(3a^3bBd+a^4Cd+b^4(4Bc+Ad)+ab^3(8Ac-8cC-5Bd)-a^2b^2(4Bc+7Ad-9Cd))\sqrt{c+d\tan(e+\cdots}}{4b(a^2+b^2)^2(bc-ad)f(a+b\tan(e+fx))}
\end{aligned}
$$

```
[Out] 1/4*(3*a^5*b*B*d^2+a^6*C*d^2-3*a^4*b^2*d*(5*A*d+4*B*c-6*C*d)-3*a^2*b^4*(8*A
*c^2-6*A*d^2-16*B*c*d-8*C*c^2+5*C*d^2)+2*a^3*b^3*(20*c*(A-C)*d+B*(4*c^2-13*
d^2))-3*a*b^5*(8*c*(A-C)*d+B*(8*c^2-d^2))-b^6*(4*c*(B*d+2*C*c)-A*(8*c^2+d^2
)))*arctanh(b^(1/2)*(c+d*tan(f*x+e))^(1/2)/(-a*d+b*c)^(1/2))/b^(3/2)/(a^2+b
^2)^3/(-a*d+b*c)^(3/2)/f-(A-I*B-C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(
1/2))*(c-I*d)^(1/2)/(I*a+b)^3/f+(A+I*B-C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c
+I*d)^(1/2))*(c+I*d)^(1/2)/(I*a-b)^3/f-1/2*(A*b^2-a*(B*b-C*a))*(c+d*tan(f*x
+e))^(1/2)/b/(a^2+b^2)/f/(a+b*tan(f*x+e))^2-1/4*(3*a^3*b*B*d+a^4*C*d+b^4*(A
*d+4*B*c)+a*b^3*(8*A*c-5*B*d-8*C*c)-a^2*b^2*(7*A*d+4*B*c-9*C*d))*(c+d*tan(f
*x+e))^(1/2)/b/(a^2+b^2)^2/(-a*d+b*c)/f/(a+b*tan(f*x+e))
```

Rubi [A] (verified)

Time = 4.68 (sec) , antiderivative size = 543, normalized size of antiderivative = 1.00, number of steps used = 13, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}} = 0.170$, Rules used = {3726, 3730, 3734, 3620, 3618, 65, 214, 3715}

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx$$
$$= -\frac{(Ab^2-a(bB-aC))\sqrt{c+d\tan(e+fx)}}{2bf(a^2+b^2)(a+b\tan(e+fx))^2}$$
$$-\frac{\sqrt{c+d\tan(e+fx)}(a^4Cd+3a^3bBd-a^2b^2(7Ad+4Bc-9Cd)+ab^3(8Ac-5Bd-8cC)+b^4(Ad+4}{4bf(a^2+b^2)^2(bc-ad)(a+b\tan(e+fx))}$$
$$+\frac{(a^6Cd^2+3a^5bBd^2-3a^4b^2d(5Ad+4Bc-6Cd)+2a^3b^3(20cd(A-C)+B(4c^2-13d^2))-3a^2b^4(8Ac^2}{}$$
$$-\frac{\sqrt{c-id}(A-iB-C)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f(b+ia)^3}$$
$$+\frac{\sqrt{c+id}(A+iB-C)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f(-b+ia)^3}$$

```
[In] Int[(Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a +
 b*Tan[e + f*x])^3,x]
```

```
[Out] -(((A - I*B - C)*Sqrt[c - I*d]*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*
d]])/((I*a + b)^3*f)) + ((A + I*B - C)*Sqrt[c + I*d]*ArcTanh[Sqrt[c + d*Tan
[e + f*x]]/Sqrt[c + I*d]])/((I*a - b)^3*f) + ((3*a^5*b*B*d^2 + a^6*C*d^2 -
3*a^4*b^2*d*(4*B*c + 5*A*d - 6*C*d) - 3*a^2*b^4*(8*A*c^2 - 8*c^2*C - 16*B*c
*d - 6*A*d^2 + 5*C*d^2) + 2*a^3*b^3*(20*c*(A - C)*d + B*(4*c^2 - 13*d^2)) -
 3*a*b^5*(8*c*(A - C)*d + B*(8*c^2 - d^2)) - b^6*(4*c*(2*c*C + B*d) - A*(8*
c^2 + d^2)))*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*d]])/(
4*b^(3/2)*(a^2 + b^2)^3*(b*c - a*d)^(3/2)*f) - ((A*b^2 - a*(b*B - a*C))*Sqr
t[c + d*Tan[e + f*x]])/(2*b*(a^2 + b^2)*f*(a + b*Tan[e + f*x])^2) - ((3*a^3
*b*B*d + a^4*C*d + b^4*(4*B*c + A*d) + a*b^3*(8*A*c - 8*c*C - 5*B*d) - a^2*
b^2*(4*B*c + 7*A*d - 9*C*d))*Sqrt[c + d*Tan[e + f*x]])/(4*b*(a^2 + b^2)^2*(
b*c - a*d)*f*(a + b*Tan[e + f*x]))
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3715

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_) + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :>
 Dist[A/f, Subst[Int[(a + b*x)^m*(c + d*x)^n, x], x, Tan[e + f*x]], x] /; F
reeQ[{a, b, c, d, e, f, A, C, m, n}, x] && EqQ[A, C]
```

Rule 3726

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*d^2 + c*(c*C - B*d))*(a + b*Tan[e +
 f*x])^m*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3730

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 - a*(b*B - a*C))*(a + b*Tan[e +
 f*x])^(m + 1)*((c + d*Tan[e + f*x])^(n + 1)/(f*(m + 1)*(b*c - a*d)*(a^2 +
```

```
b^2))), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3734

```
Int[(((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2))/((a_.) + (b_.)*tan[(e_.)
+ (f_.)*(x_)]), x_Symbol] :> Dist[1/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n
*Simp[b*B + a*(A - C) + (a*B - b*(A - C))*Tan[e + f*x], x], x], x] + Dist[(
A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n*((1 + Tan[e
+ f*x]^2)/(a + b*Tan[e + f*x])), x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C
, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&
 !GtQ[n, 0] &&  !LeQ[n, -1]
```

Rubi steps

$$\text{integral} = -\frac{(Ab^2 - a(bB - aC))\sqrt{c + d\tan(e + fx)}}{2b(a^2 + b^2) f(a + b\tan(e + fx))^2} + \frac{\int \frac{\frac{1}{2}\left(2(bB-aC)\left(2bc-\frac{ad}{2}\right)+Ab(4ac+bd)\right)-2b((A-C)(bc-ad)-B(ac+bd))\tan(e+fx)-\frac{1}{2}(3Ab^2-3abB-a^2C-4b^2C)d\tan^2(e+fx)}{(a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}}\,dx}{2b(a^2 + b^2)}$$

$$= -\frac{(Ab^2 - a(bB - aC))\sqrt{c + d\tan(e + fx)}}{2b(a^2 + b^2) f(a + b\tan(e + fx))^2} - \frac{(3a^3bBd + a^4Cd + b^4(4Bc + Ad) + ab^3(8Ac - 8cC - 5Bd) - a^2b^2(4Bc + 7Ad - 9Cd))\sqrt{c + d}}{4b(a^2 + b^2)^2 (bc - ad) f(a + b\tan(e + fx))} - \frac{\int \frac{1}{4}(-((2abc-2a^2d-b^2d)(a^2Cd+b^2(4Bc+Ad)+ab(4Ac-4cC-Bd)))+(2bc-ad)(3a^2bBd+a^3Cd+Ab^2(4bc-7ad)-4b^3(cC+Bd)-}{}$$

$$= -\frac{(Ab^2 - a(bB - aC))\sqrt{c + d\tan(e + fx)}}{2b(a^2 + b^2) f(a + b\tan(e + fx))^2} - \frac{(3a^3bBd + a^4Cd + b^4(4Bc + Ad) + ab^3(8Ac - 8cC - 5Bd) - a^2b^2(4Bc + 7Ad - 9Cd))\sqrt{c + }}{4b(a^2 + b^2)^2 (bc - ad) f(a + b\tan(e + fx))} - \frac{\int \frac{-2b(bc-ad)(a^3(Ac-cC-Bd)-3ab^2(Ac-cC-Bd)+3a^2b(Bc+(A-C)d)-b^3(Bc+(A-C)d))+2b(bc-ad)(3a^2b(Ac-cC-Bd)-b^3}{\sqrt{c+d\tan(e+fx)}}}{2b(a^2 + b^2)^3 (bc - ad)} - \frac{(3a^5bBd^2 + a^6Cd^2 - 3a^4b^2d(4Bc + 5Ad - 6Cd) - 3a^2b^4(8Ac^2 - 8c^2C - 16Bcd - 6Ad^2 + 5Cd}{}$$

$$= -\frac{(Ab^2 - a(bB - aC))\sqrt{c + d\tan(e + fx)}}{2b(a^2 + b^2) f(a + b\tan(e + fx))^2}$$
$$- \frac{(3a^3bBd + a^4Cd + b^4(4Bc + Ad) + ab^3(8Ac - 8cC - 5Bd) - a^2b^2(4Bc + 7Ad - 9Cd))\sqrt{c + d}}{4b(a^2 + b^2)^2(bc - ad)f(a + b\tan(e + fx))}$$
$$+ \frac{((A - iB - C)(c - id))\int \frac{1 + i\tan(e+fx)}{\sqrt{c + d\tan(e+fx)}}\,dx}{2(a - ib)^3}$$
$$+ \frac{((A + iB - C)(c + id))\int \frac{1 - i\tan(e+fx)}{\sqrt{c + d\tan(e+fx)}}\,dx}{2(a + ib)^3}$$
$$- \frac{(3a^5bBd^2 + a^6Cd^2 - 3a^4b^2d(4Bc + 5Ad - 6Cd) - 3a^2b^4(8Ac^2 - 8c^2C - 16Bcd - 6Ad^2 + 5Cd^2)}{}$$

$$= -\frac{(Ab^2 - a(bB - aC))\sqrt{c + d\tan(e + fx)}}{2b(a^2 + b^2) f(a + b\tan(e + fx))^2}$$
$$- \frac{(3a^3bBd + a^4Cd + b^4(4Bc + Ad) + ab^3(8Ac - 8cC - 5Bd) - a^2b^2(4Bc + 7Ad - 9Cd))\sqrt{c + d}}{4b(a^2 + b^2)^2(bc - ad)f(a + b\tan(e + fx))}$$
$$- \frac{((A + iB - C)(c + id))\text{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c+idx}}\,dx, x, -i\tan(e + fx)\right)}{2(ia - b)^3 f}$$
$$+ \frac{((A - iB - C)(ic + d))\text{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c-idx}}\,dx, x, i\tan(e + fx)\right)}{2(a - ib)^3 f}$$
$$- \frac{(3a^5bBd^2 + a^6Cd^2 - 3a^4b^2d(4Bc + 5Ad - 6Cd) - 3a^2b^4(8Ac^2 - 8c^2C - 16Bcd - 6Ad^2 + 5Cd^2)}{}$$

$$= \frac{(3a^5bBd^2 + a^6Cd^2 - 3a^4b^2d(4Bc + 5Ad - 6Cd) - 3a^2b^4(8Ac^2 - 8c^2C - 16Bcd - 6Ad^2 + 5Cd^2) +}{}$$
$$- \frac{(Ab^2 - a(bB - aC))\sqrt{c + d\tan(e + fx)}}{2b(a^2 + b^2) f(a + b\tan(e + fx))^2}$$
$$- \frac{(3a^3bBd + a^4Cd + b^4(4Bc + Ad) + ab^3(8Ac - 8cC - 5Bd) - a^2b^2(4Bc + 7Ad - 9Cd))\sqrt{c + d}}{4b(a^2 + b^2)^2(bc - ad)f(a + b\tan(e + fx))}$$
$$- \frac{((A + iB - C)(c + id))\text{Subst}\left(\int \frac{1}{-1 + \frac{ic}{d} - \frac{ix^2}{d}}\,dx, x, \sqrt{c + d\tan(e + fx)}\right)}{(a + ib)^3 df}$$
$$+ \frac{((A - iB - C)(ic + d))\text{Subst}\left(\int \frac{1}{-1 - \frac{ic}{d} + \frac{ix^2}{d}}\,dx, x, \sqrt{c + d\tan(e + fx)}\right)}{(ia + b)^3 df}$$

$$= -\frac{(A-iB-C)\sqrt{c-id}\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(ia+b)^3f} + \frac{(A+iB-C)\sqrt{c+id}\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(ia-b)^3f} + \frac{(3a^5bBd^2+a^6Cd^2-3a^4b^2d(4Bc+5Ad-6Cd)-3a^2b^4(8Ac^2-8c^2C-16Bcd-6Ad^2+5Cd}{} - \frac{(Ab^2-a(bB-aC))\sqrt{c+d\tan(e+fx)}}{2b(a^2+b^2)f(a+b\tan(e+fx))^2} - \frac{(3a^3bBd+a^4Cd+b^4(4Bc+Ad)+ab^3(8Ac-8cC-5Bd)-a^2b^2(4Bc+7Ad-9Cd))\sqrt{c+}}{4b(a^2+b^2)^2(bc-ad)f(a+b\tan(e+fx))}$$

Mathematica [B] (verified)

Both result and optimal contain complex but leaf count is larger than twice the leaf count of optimal. 2819 vs. $2(543) = 1086$.

Time = 6.72 (sec) , antiderivative size = 2819, normalized size of antiderivative = 5.19

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx = \text{Result too large to show}$$

```
[In] Integrate[(Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)
)/(a + b*Tan[e + f*x])^3,x]
```

```
[Out] (-2*C*Sqrt[c + d*Tan[e + f*x]])/(3*b*f*(a + b*Tan[e + f*x])^2) - (2*(-1/2*(
((b^2*(-3*A*b*c + 4*b*c*C - a*C*d))/2 - a*((-3*b^2*(B*c + (A - C)*d))/2 - (
a*(b*c*C - 3*b*B*d - a*C*d))/2))*Sqrt[c + d*Tan[e + f*x]])/((a^2 + b^2)*(b*
c - a*d)*f*(a + b*Tan[e + f*x])^2) - (-((((I*Sqrt[c - I*d]*(b*(b*c - a*d)*(
(3*b*(3*A*b^2 - 3*a*b*B - a^2*C - 4*b^2*C)*d*(b*c - a*d))/4 + 3*a*b*(b*c -
a*d)*(A*b*c - a*B*c - b*c*C - a*A*d - b*B*d + a*C*d) + (3*b*(b*c - a*d)*(a^
2*C*d + b^2*(4*B*c + A*d) + a*b*(4*A*c - 4*c*C - B*d)))/4) + a*((3*(b*c - a
*d)*((b^2*d)/2 - a*(b*c - a*d))*(a^2*C*d + b^2*(4*B*c + A*d) + a*b*(4*A*c -
 4*c*C - B*d)))/4 + (-(b*c) + (a*d)/2)*((3*a*(3*A*b^2 - 3*a*b*B - a^2*C - 4
*b^2*C)*d*(b*c - a*d))/4 - 3*b^2*(b*c - a*d)*(A*b*c - a*B*c - b*c*C - a*A*d
 - b*B*d + a*C*d)) - (d*((3*b^2*(b*c - a*d)*(a^2*C*d + b^2*(4*B*c + A*d) +
a*b*(4*A*c - 4*c*C - B*d)))/4 - a*((3*a*(3*A*b^2 - 3*a*b*B - a^2*C - 4*b^2*
C)*d*(b*c - a*d))/4 - 3*b^2*(b*c - a*d)*(A*b*c - a*B*c - b*c*C - a*A*d - b*
B*d + a*C*d))))/2) - I*(a*(b*c - a*d)*((3*b*(3*A*b^2 - 3*a*b*B - a^2*C - 4*
b^2*C)*d*(b*c - a*d))/4 + 3*a*b*(b*c - a*d)*(A*b*c - a*B*c - b*c*C - a*A*d
- b*B*d + a*C*d) + (3*b*(b*c - a*d)*(a^2*C*d + b^2*(4*B*c + A*d) + a*b*(4*A
*c - 4*c*C - B*d)))/4) - b*((3*(b*c - a*d)*((b^2*d)/2 - a*(b*c - a*d))*(a^2
*C*d + b^2*(4*B*c + A*d) + a*b*(4*A*c - 4*c*C - B*d)))/4 + (-(b*c) + (a*d)/
```

```
2)*((3*a*(3*A*b^2 - 3*a*b*B - a^2*C - 4*b^2*C)*d*(b*c - a*d))/4 - 3*b^2*(b*
c - a*d)*(A*b*c - a*B*c - b*c*C - a*A*d - b*B*d + a*C*d)) - (d*((3*b^2*(b*c
 - a*d)*(a^2*C*d + b^2*(4*B*c + A*d) + a*b*(4*A*c - 4*c*C - B*d)))/4 - a*((
3*a*(3*A*b^2 - 3*a*b*B - a^2*C - 4*b^2*C)*d*(b*c - a*d))/4 - 3*b^2*(b*c - a
*d)*(A*b*c - a*B*c - b*c*C - a*A*d - b*B*d + a*C*d))))/2)))*ArcTanh[Sqrt[c
+ d*Tan[e + f*x]]/Sqrt[c - I*d]])/((-c + I*d)*f) - (I*Sqrt[c + I*d]*(b*(b*c
 - a*d)*((3*b*(3*A*b^2 - 3*a*b*B - a^2*C - 4*b^2*C)*d*(b*c - a*d))/4 + 3*a*
b*(b*c - a*d)*(A*b*c - a*B*c - b*c*C - a*A*d - b*B*d + a*C*d) + (3*b*(b*c -
 a*d)*(a^2*C*d + b^2*(4*B*c + A*d) + a*b*(4*A*c - 4*c*C - B*d)))/4) + a*((3
*(b*c - a*d)*((b^2*d)/2 - a*(b*c - a*d))*(a^2*C*d + b^2*(4*B*c + A*d) + a*b
*(4*A*c - 4*c*C - B*d)))/4 + (-(b*c) + (a*d)/2)*((3*a*(3*A*b^2 - 3*a*b*B -
a^2*C - 4*b^2*C)*d*(b*c - a*d))/4 - 3*b^2*(b*c - a*d)*(A*b*c - a*B*c - b*c*
C - a*A*d - b*B*d + a*C*d)) - (d*((3*b^2*(b*c - a*d)*(a^2*C*d + b^2*(4*B*c
+ A*d) + a*b*(4*A*c - 4*c*C - B*d)))/4 - a*((3*a*(3*A*b^2 - 3*a*b*B - a^2*C
 - 4*b^2*C)*d*(b*c - a*d))/4 - 3*b^2*(b*c - a*d)*(A*b*c - a*B*c - b*c*C - a
*A*d - b*B*d + a*C*d))))/2) + I*(a*(b*c - a*d)*((3*b*(3*A*b^2 - 3*a*b*B - a
^2*C - 4*b^2*C)*d*(b*c - a*d))/4 + 3*a*b*(b*c - a*d)*(A*b*c - a*B*c - b*c*C
 - a*A*d - b*B*d + a*C*d) + (3*b*(b*c - a*d)*(a^2*C*d + b^2*(4*B*c + A*d) +
 a*b*(4*A*c - 4*c*C - B*d)))/4) - b*((3*(b*c - a*d)*((b^2*d)/2 - a*(b*c - a
*d))*(a^2*C*d + b^2*(4*B*c + A*d) + a*b*(4*A*c - 4*c*C - B*d)))/4 + (-(b*c)
 + (a*d)/2)*((3*a*(3*A*b^2 - 3*a*b*B - a^2*C - 4*b^2*C)*d*(b*c - a*d))/4 -
3*b^2*(b*c - a*d)*(A*b*c - a*B*c - b*c*C - a*A*d - b*B*d + a*C*d)) - (d*((3
*b^2*(b*c - a*d)*(a^2*C*d + b^2*(4*B*c + A*d) + a*b*(4*A*c - 4*c*C - B*d)))
/4 - a*((3*a*(3*A*b^2 - 3*a*b*B - a^2*C - 4*b^2*C)*d*(b*c - a*d))/4 - 3*b^2
*(b*c - a*d)*(A*b*c - a*B*c - b*c*C - a*A*d - b*B*d + a*C*d))))/2)))*ArcTan
h[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]])/((-c - I*d)*f))/(a^2 + b^2) + (2
*Sqrt[b*c - a*d]*(-(a*b*(b*c - a*d)*((3*b*(3*A*b^2 - 3*a*b*B - a^2*C - 4*b^
2*C)*d*(b*c - a*d))/4 + 3*a*b*(b*c - a*d)*(A*b*c - a*B*c - b*c*C - a*A*d -
b*B*d + a*C*d) + (3*b*(b*c - a*d)*(a^2*C*d + b^2*(4*B*c + A*d) + a*b*(4*A*c
 - 4*c*C - B*d)))/4)) + (a^2*d*((3*b^2*(b*c - a*d)*(a^2*C*d + b^2*(4*B*c +
A*d) + a*b*(4*A*c - 4*c*C - B*d)))/4 - a*((3*a*(3*A*b^2 - 3*a*b*B - a^2*C -
 4*b^2*C)*d*(b*c - a*d))/4 - 3*b^2*(b*c - a*d)*(A*b*c - a*B*c - b*c*C - a*A
*d - b*B*d + a*C*d))))/2 + b^2*((3*(b*c - a*d)*((b^2*d)/2 - a*(b*c - a*d))*
(a^2*C*d + b^2*(4*B*c + A*d) + a*b*(4*A*c - 4*c*C - B*d)))/4 + (-(b*c) + (a
*d)/2)*((3*a*(3*A*b^2 - 3*a*b*B - a^2*C - 4*b^2*C)*d*(b*c - a*d))/4 - 3*b^2
*(b*c - a*d)*(A*b*c - a*B*c - b*c*C - a*A*d - b*B*d + a*C*d))))*ArcTanh[(Sq
rt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*d]])/(Sqrt[b]*(a^2 + b^2)*(-(b
*c) + a*d)*f))/((a^2 + b^2)*(b*c - a*d))) - (((3*b^2*(b*c - a*d)*(a^2*C*d +
 b^2*(4*B*c + A*d) + a*b*(4*A*c - 4*c*C - B*d)))/4 - a*((3*a*(3*A*b^2 - 3*a
*b*B - a^2*C - 4*b^2*C)*d*(b*c - a*d))/4 - 3*b^2*(b*c - a*d)*(A*b*c - a*B*c
 - b*c*C - a*A*d - b*B*d + a*C*d)))*Sqrt[c + d*Tan[e + f*x]])/((a^2 + b^2)*
(b*c - a*d)*f*(a + b*Tan[e + f*x]))))/(2*(a^2 + b^2)*(b*c - a*d))))/(3*b)
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 9796 vs. $2(503) = 1006$.

Time = 0.15 (sec) , antiderivative size = 9797, normalized size of antiderivative = 18.04

method	result	size
derivativedivides	Expression too large to display	9797
default	Expression too large to display	9797

```
[In] int((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^3,x,method=_RETURNVERBOSE)
```

```
[Out] result too large to display
```

Fricas [F(-1)]

Timed out.

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^3,x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx = \int \frac{\sqrt{c+d\tan{(e+fx)}}(A+B\tan{(e+fx)}+C\tan^2{(e+fx)})}{(a+b\tan{(e+fx)})^3}\,dx$$

```
[In] integrate((c+d*tan(f*x+e))**(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e))**3,x)
```

```
[Out] Integral(sqrt(c + d*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/
(a + b*tan(e + f*x))**3, x)
```

Maxima [F(-2)]

Exception generated.

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx$$
$$= \text{Exception raised: ValueError}$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^3,x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 detail
```

Giac [F(-1)]

Timed out.

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^3,x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx = \text{Hanged}$$

```
[In] int(((c + d*tan(e + f*x))^(1/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x))^3,x)
```

```
[Out] \text{Hanged}
```

3.97 $\int (a+b\tan(e+fx))^3(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e\right.$

Optimal result 1275
Rubi [A] (verified) 1276
Mathematica [B] (verified) 1282
Maple [B] (verified) 1284
Fricas [B] (verification not implemented) 1284
Sympy [F] 1285
Maxima [F(-1)] 1285
Giac [F(-1)] 1285
Mupad [F(-1)] 1286

Optimal result

Integrand size = 47, antiderivative size = 550

$$\int (a+b\tan(e+fx))^3(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)\right.$$
$$\left.+C\tan^2(e+fx)\right)\,dx=\frac{(ia+b)^3(A-iB-C)(c-id)^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f}$$
$$+\frac{(a+ib)^3(iA-B-iC)(c+id)^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f}$$
$$+\frac{2(3a^2b(Ac-cC-Bd)-b^3(Ac-cC-Bd)+a^3(Bc+(A-C)d)-3ab^2(Bc+(A-C)d))\sqrt{c+d\tan(e}}{f}$$
$$+\frac{2(a^3B-3ab^2B+3a^2b(A-C)-b^3(A-C))\,(c+d\tan(e+fx))^{3/2}}{3f}$$
$$+\frac{2(168a^3Cd^3-2a^2bd^2(192cC-847Bd)+33ab^2d(8c^2C-18Bcd+63(A-C)d^2)-b^3(48c^3C-88Bc^2d+}{3465d^4f}$$
$$+\frac{2b(99b(Ab+aB-bC)d^2+4(bc-ad)(6bcC-11bBd-6aCd))\tan(e+fx)(c+d\tan(e+fx))^{5/2}}{693d^3f}$$
$$-\frac{2(6bcC-11bBd-6aCd)(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}}{99d^2f}$$
$$+\frac{2C(a+b\tan(e+fx))^3(c+d\tan(e+fx))^{5/2}}{11df}$$

```
[Out] (I*a+b)^3*(A-I*B-C)*(c-I*d)^(3/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1
/2))/f+(a+I*b)^3*(I*A-B-I*C)*(c+I*d)^(3/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(
c+I*d)^(1/2))/f+2*(3*a^2*b*(A*c-B*d-C*c)-b^3*(A*c-B*d-C*c)+a^3*(B*c+(A-C)*d
)-3*a*b^2*(B*c+(A-C)*d))*(c+d*tan(f*x+e))^(1/2)/f+2/3*(B*a^3-3*B*a*b^2+3*a^
2*b*(A-C)-b^3*(A-C))*(c+d*tan(f*x+e))^(3/2)/f+2/3465*(168*a^3*C*d^3-2*a^2*b
*d^2*(-847*B*d+192*C*c)+33*a*b^2*d*(8*c^2*C-18*B*c*d+63*(A-C)*d^2)-b^3*(48*
```

```
c^3*C-88*B*c^2*d+198*c*(A-C)*d^2+693*B*d^3))*(c+d*tan(f*x+e))^(5/2)/d^4/f+2
/693*b*(99*b*(A*b+B*a-C*b)*d^2+4*(-a*d+b*c)*(-11*B*b*d-6*C*a*d+6*C*b*c))*ta
n(f*x+e)*(c+d*tan(f*x+e))^(5/2)/d^3/f-2/99*(-11*B*b*d-6*C*a*d+6*C*b*c)*(a+b
*tan(f*x+e))^2*(c+d*tan(f*x+e))^(5/2)/d^2/f+2/11*C*(a+b*tan(f*x+e))^3*(c+d*
tan(f*x+e))^(5/2)/d/f
```

Rubi [A] (verified)

Time = 2.85 (sec) , antiderivative size = 550, normalized size of antiderivative = 1.00, number of steps used = 13, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}} = 0.170$, Rules used = {3728, 3718, 3711, 3609, 3620, 3618, 65, 214}

$$\int (a+b\tan(e+fx))^3(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \frac{2(c+d\tan(e+fx))^{5/2}\left(168a^3Cd^3-2a^2bd^2(192cC-847Bd)+33ab^2d(63d^2(A-C)\right.}{3465d^4f}$$
$$+\frac{2(a^3B+3a^2b(A-C)-3ab^2B-b^3(A-C))\,(c+d\tan(e+fx))^{3/2}}{3f}$$
$$+\frac{2\sqrt{c+d\tan(e+fx)}(a^3(d(A-C)+Bc)+3a^2b(Ac-Bd-cC)-3ab^2(d(A-C)+Bc)-b^3(Ac-Bd-}{f}$$
$$+\frac{(a+ib)^3(c+id)^{3/2}(iA-B-iC)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f}$$
$$+\frac{(b+ia)^3(c-id)^{3/2}(A-iB-C)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f}$$
$$+\frac{2b\tan(e+fx)(c+d\tan(e+fx))^{5/2}\left(99bd^2(aB+Ab-bC)+4(bc-ad)(-6aCd-11bBd+6bcC)\right)}{693d^3f}$$
$$-\frac{2(-6aCd-11bBd+6bcC)(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}}{99d^2f}$$
$$+\frac{2C(a+b\tan(e+fx))^3(c+d\tan(e+fx))^{5/2}}{11df}$$

```
[In] Int[(a + b*Tan[e + f*x])^3*(c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] +
 C*Tan[e + f*x]^2),x]
```

```
[Out] ((I*a + b)^3*(A - I*B - C)*(c - I*d)^(3/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]
/Sqrt[c - I*d]])/f + ((a + I*b)^3*(I*A - B - I*C)*(c + I*d)^(3/2)*ArcTanh[S
qrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]])/f + (2*(3*a^2*b*(A*c - c*C - B*d) -
 b^3*(A*c - c*C - B*d) + a^3*(B*c + (A - C)*d) - 3*a*b^2*(B*c + (A - C)*d))
*Sqrt[c + d*Tan[e + f*x]])/f + (2*(a^3*B - 3*a*b^2*B + 3*a^2*b*(A - C) - b^
3*(A - C))*(c + d*Tan[e + f*x])^(3/2))/(3*f) + (2*(168*a^3*C*d^3 - 2*a^2*b*
d^2*(192*c*C - 847*B*d) + 33*a*b^2*d*(8*c^2*C - 18*B*c*d + 63*(A - C)*d^2)
- b^3*(48*c^3*C - 88*B*c^2*d + 198*c*(A - C)*d^2 + 693*B*d^3))*(c + d*Tan[e
```

```
 + f*x])^(5/2))/(3465*d^4*f) + (2*b*(99*b*(A*b + a*B - b*C)*d^2 + 4*(b*c -
a*d)*(6*b*c*C - 11*b*B*d - 6*a*C*d))*Tan[e + f*x]*(c + d*Tan[e + f*x])^(5/2
))/(693*d^3*f) - (2*(6*b*c*C - 11*b*B*d - 6*a*C*d)*(a + b*Tan[e + f*x])^2*(
c + d*Tan[e + f*x])^(5/2))/(99*d^2*f) + (2*C*(a + b*Tan[e + f*x])^3*(c + d*
Tan[e + f*x])^(5/2))/(11*d*f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3609

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[d*((a + b*Tan[e + f*x])^m/(f*m)), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
```

```
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\text{integral} = \frac{2C(a+b\tan(e+fx))^3(c+d\tan(e+fx))^{5/2}}{11df} + \frac{2\int(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}\left(\frac{1}{2}(-6bcC+a(11A-5C)d)+\frac{11}{2}(Ab+aB-bC)d\tan(e+\right.}{11d}$$

$$= -\frac{2(6bcC-11bBd-6aCd)(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}}{99d^2f} + \frac{2C(a+b\tan(e+fx))^3(c+d\tan(e+fx))^{5/2}}{11df} + \frac{4\int(a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}\left(\frac{1}{4}(3a^2(33A-25C)d^2+4b^2c(6cC-11Bd)-abd(4\right.}{}$$

$$= \frac{2b(99b(Ab+aB-bC)d^2+4(bc-ad)(6bcC-11bBd-6aCd))\tan(e+fx)(c+d\tan(e+fx))^{5/}}{693d^3f}$$
$$-\frac{2(6bcC-11bBd-6aCd)(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}}{99d^2f}$$
$$+\frac{2C(a+b\tan(e+fx))^3(c+d\tan(e+fx))^{5/2}}{11df}$$
$$-\frac{8\int(c+d\tan(e+fx))^{3/2}\left(\frac{1}{8}(-21a^3(33A-25C)d^3-66ab^2cd(4cC-9Bd)+a^2bd^2(384cC+38\right.}{}$$

$$= \frac{2(168a^3Cd^3-2a^2bd^2(192cC-847Bd)+33ab^2d(8c^2C-18Bcd+63(A-C)d^2)-b^3(48c^3C-8}{3465d^4f}$$
$$+\frac{2b(99b(Ab+aB-bC)d^2+4(bc-ad)(6bcC-11bBd-6aCd))\tan(e+fx)(c+d\tan(e+fx)}{693d^3f}$$
$$-\frac{2(6bcC-11bBd-6aCd)(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}}{99d^2f}$$
$$+\frac{2C(a+b\tan(e+fx))^3(c+d\tan(e+fx))^{5/2}}{11df}$$
$$-\frac{8\int(c+d\tan(e+fx))^{3/2}\left(\frac{693}{8}(3a^2bB-b^3B-a^3(A-C)+3ab^2(A-C))\,d^3-\frac{693}{8}(a^3B-3ab^2\right.}{693d^3}$$

$$= \frac{2(a^3B-3ab^2B+3a^2b(A-C)-b^3(A-C))\,(c+d\tan(e+fx))^{3/2}}{3f}$$
$$+\frac{2(168a^3Cd^3-2a^2bd^2(192cC-847Bd)+33ab^2d(8c^2C-18Bcd+63(A-C)d^2)-b^3(48c^3C-}{3465d^4f}$$
$$+\frac{2b(99b(Ab+aB-bC)d^2+4(bc-ad)(6bcC-11bBd-6aCd))\tan(e+fx)(c+d\tan(e+fx)}{693d^3f}$$
$$-\frac{2(6bcC-11bBd-6aCd)(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}}{99d^2f}$$
$$+\frac{2C(a+b\tan(e+fx))^3(c+d\tan(e+fx))^{5/2}}{11df}$$
$$-\frac{8\int\sqrt{c+d\tan(e+fx)}\left(-\frac{693}{8}d^3(a^3(Ac-cC-Bd)-3ab^2(Ac-cC-Bd)-3a^2b(Bc+(A-\right.}{}$$

$$= \frac{2(3a^2b(Ac-cC-Bd)-b^3(Ac-cC-Bd)+a^3(Bc+(A-C)d)-3ab^2(Bc+(A-C)d))\sqrt{c+}}{f}$$

$$+\frac{2(a^3B-3ab^2B+3a^2b(A-C)-b^3(A-C))(c+d\tan(e+fx))^{3/2}}{3f}$$

$$+\frac{2(168a^3Cd^3-2a^2bd^2(192cC-847Bd)+33ab^2d(8c^2C-18Bcd+63(A-C)d^2)-b^3(48c^3C-}{3465d^4f}$$

$$+\frac{2b(99b(Ab+aB-bC)d^2+4(bc-ad)(6bcC-11bBd-6aCd))\tan(e+fx)(c+d\tan(e+fx))}{693d^3f}$$

$$-\frac{2(6bcC-11bBd-6aCd)(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}}{99d^2f}$$

$$+\frac{2C(a+b\tan(e+fx))^3(c+d\tan(e+fx))^{5/2}}{11df}$$

$$-\frac{8\int\frac{\frac{693}{8}d^3\left(a^3\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-3ab^2\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)+3a^2b\left(2c(A-C)d+B\left(c^2-d^2\right)\right)-b^3\left(2c(A-C)d+\right.\right.}{}}{}$$

$$= \frac{2(3a^2b(Ac-cC-Bd)-b^3(Ac-cC-Bd)+a^3(Bc+(A-C)d)-3ab^2(Bc+(A-C)d))\sqrt{c+}}{f}$$

$$+\frac{2(a^3B-3ab^2B+3a^2b(A-C)-b^3(A-C))(c+d\tan(e+fx))^{3/2}}{3f}$$

$$+\frac{2(168a^3Cd^3-2a^2bd^2(192cC-847Bd)+33ab^2d(8c^2C-18Bcd+63(A-C)d^2)-b^3(48c^3C-}{3465d^4f}$$

$$+\frac{2b(99b(Ab+aB-bC)d^2+4(bc-ad)(6bcC-11bBd-6aCd))\tan(e+fx)(c+d\tan(e+fx))}{693d^3f}$$

$$-\frac{2(6bcC-11bBd-6aCd)(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}}{99d^2f}$$

$$+\frac{2C(a+b\tan(e+fx))^3(c+d\tan(e+fx))^{5/2}}{11df}$$

$$+\frac{1}{2}\left((a-ib)^3(A-iB-C)(c-id)^2\right)\int\frac{1+i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx+\frac{1}{2}\left((a+ib)^3(A+iB-C)(c+id)^2\right)\int$$

$$=\frac{2(3a^2b(Ac-cC-Bd)-b^3(Ac-cC-Bd)+a^3(Bc+(A-C)d)-3ab^2(Bc+(A-C)d))\sqrt{c\cdot}}{f}$$
$$+\frac{2(a^3B-3ab^2B+3a^2b(A-C)-b^3(A-C))(c+d\tan(e+fx))^{3/2}}{3f}$$
$$+\frac{2(168a^3Cd^3-2a^2bd^2(192cC-847Bd)+33ab^2d(8c^2C-18Bcd+63(A-C)d^2)-b^3(48c^3C-}{3465d^4f}$$
$$+\frac{2b(99b(Ab+aB-bC)d^2+4(bc-ad)(6bcC-11bBd-6aCd))\tan(e+fx)(c+d\tan(e+fx)}{693d^3f}$$
$$-\frac{2(6bcC-11bBd-6aCd)(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}}{99d^2f}$$
$$+\frac{2C(a+b\tan(e+fx))^3(c+d\tan(e+fx))^{5/2}}{11df}$$
$$+\frac{((a-ib)^3(iA+B-iC)(c-id)^2)\operatorname{Subst}\left(\int\frac{1}{(-1+x)\sqrt{c-idx}}\,dx,x,i\tan(e+fx)\right)}{2f}$$
$$-\frac{(i(a+ib)^3(A+iB-C)(c+id)^2)\operatorname{Subst}\left(\int\frac{1}{(-1+x)\sqrt{c+idx}}\,dx,x,-i\tan(e+fx)\right)}{2f}$$

$$=\frac{2(3a^2b(Ac-cC-Bd)-b^3(Ac-cC-Bd)+a^3(Bc+(A-C)d)-3ab^2(Bc+(A-C)d))\sqrt{c\cdot}}{f}$$
$$+\frac{2(a^3B-3ab^2B+3a^2b(A-C)-b^3(A-C))(c+d\tan(e+fx))^{3/2}}{3f}$$
$$+\frac{2(168a^3Cd^3-2a^2bd^2(192cC-847Bd)+33ab^2d(8c^2C-18Bcd+63(A-C)d^2)-b^3(48c^3C-}{3465d^4f}$$
$$+\frac{2b(99b(Ab+aB-bC)d^2+4(bc-ad)(6bcC-11bBd-6aCd))\tan(e+fx)(c+d\tan(e+fx)}{693d^3f}$$
$$-\frac{2(6bcC-11bBd-6aCd)(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}}{99d^2f}$$
$$+\frac{2C(a+b\tan(e+fx))^3(c+d\tan(e+fx))^{5/2}}{11df}$$
$$-\frac{((a-ib)^3(A-iB-C)(c-id)^2)\operatorname{Subst}\left(\int\frac{1}{-1-\frac{ic}{d}+\frac{ix^2}{d}}\,dx,x,\sqrt{c+d\tan(e+fx)}\right)}{df}$$
$$-\frac{((a+ib)^3(A+iB-C)(c+id)^2)\operatorname{Subst}\left(\int\frac{1}{-1+\frac{ic}{d}-\frac{ix^2}{d}}\,dx,x,\sqrt{c+d\tan(e+fx)}\right)}{df}$$

$$= -\frac{(a-ib)^3(iA+B-iC)(c-id)^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f}$$
$$-\frac{(ia-b)^3(A+iB-C)(c+id)^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f}$$
$$+\frac{2(3a^2b(Ac-cC-Bd)-b^3(Ac-cC-Bd)+a^3(Bc+(A-C)d)-3ab^2(Bc+(A-C)d))\sqrt{c}}{f}$$
$$+\frac{2(a^3B-3ab^2B+3a^2b(A-C)-b^3(A-C))(c+d\tan(e+fx))^{3/2}}{3f}$$
$$+\frac{2(168a^3Cd^3-2a^2bd^2(192cC-847Bd)+33ab^2d(8c^2C-18Bcd+63(A-C)d^2)-b^3(48c^3C-}{3465d^4f}$$
$$+\frac{2b(99b(Ab+aB-bC)d^2+4(bc-ad)(6bcC-11bBd-6aCd))\tan(e+fx)(c+d\tan(e+fx))^{}}{693d^3f}$$
$$-\frac{2(6bcC-11bBd-6aCd)(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}}{99d^2f}$$
$$+\frac{2C(a+b\tan(e+fx))^3(c+d\tan(e+fx))^{5/2}}{11df}$$

Mathematica [B] (verified)

Both result and optimal contain complex but leaf count is larger than twice the leaf count of optimal. 1290 vs. 2(550) = 1100.

Time = 6.57 (sec) , antiderivative size = 1290, normalized size of antiderivative = 2.35

$$\int (a+b\tan(e+fx))^3(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \frac{2C(a+b\tan(e+fx))^3(c+d\tan(e+fx))^{5/2}}{11df} + \frac{2\left(\frac{(-6bcC+11bBd+6aCd)(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}}{9df} + \frac{2\left(\frac{b\left(99b(Ab+aB-bC)d^2+4(bc-ad)(6bcC-11bBd-6aCd)\right)\tan(e+fx)(c+d\tan(e\cdots}{14df}\cdots\right)}{\cdots}\right)}{\cdots}$$

```
[In] Integrate[(a + b*Tan[e + f*x])^3*(c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e +
f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] (2*C*(a + b*Tan[e + f*x])^3*(c + d*Tan[e + f*x])^(5/2))/(11*d*f) + (2*(((-6
*b*c*C + 11*b*B*d + 6*a*C*d)*(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])^(5
/2))/(9*d*f) + (2*((b*(99*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C
- 11*b*B*d - 6*a*C*d))*Tan[e + f*x]*(c + d*Tan[e + f*x])^(5/2))/(14*d*f) -
(2*((2*((-7*a*d*(99*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C - 11*b
*B*d - 6*a*C*d)))/8 + b*((-693*(a^2*B - b^2*B + 2*a*b*(A - C))*d^3)/8 + (c*
(99*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C - 11*b*B*d - 6*a*C*d))
)/4))*(c + d*Tan[e + f*x])^(5/2))/(5*d*f) + ((I/2)*((-7*a*d*(3*a^2*(33*A -
25*C)*d^2 + 4*b^2*c*(6*c*C - 11*B*d) - a*b*d*(48*c*C + 55*B*d)))/8 + (b*c*(
99*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C - 11*b*B*d - 6*a*C*d)))
/4 + (7*a*d*(99*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C - 11*b*B*d
 - 6*a*C*d)))/8 + ((7*I)/2)*d*((99*a*(a^2*B - b^2*B + 2*a*b*(A - C))*d^2)/4
 + (b*(3*a^2*(33*A - 25*C)*d^2 + 4*b^2*c*(6*c*C - 11*B*d) - a*b*d*(48*c*C +
 55*B*d)))/4 - (b*(99*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C - 11
*b*B*d - 6*a*C*d)))/4) - b*((-693*(a^2*B - b^2*B + 2*a*b*(A - C))*d^3)/8 +
(c*(99*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C - 11*b*B*d - 6*a*C*
d)))/4))*((2*(c + d*Tan[e + f*x])^(3/2))/3 + (c - I*d)*((2*(c - I*d)^(3/2)*
```

```
ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]])/(-c + I*d) + 2*Sqrt[c + d*
Tan[e + f*x]])))/f - ((I/2)*((-7*a*d*(3*a^2*(33*A - 25*C)*d^2 + 4*b^2*c*(6*
c*C - 11*B*d) - a*b*d*(48*c*C + 55*B*d)))/8 + (b*c*(99*b*(A*b + a*B - b*C)*
d^2 + 4*(b*c - a*d)*(6*b*c*C - 11*b*B*d - 6*a*C*d)))/4 + (7*a*d*(99*b*(A*b
+ a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C - 11*b*B*d - 6*a*C*d)))/8 - ((7*I
)/2)*d*((99*a*(a^2*B - b^2*B + 2*a*b*(A - C))*d^2)/4 + (b*(3*a^2*(33*A - 25
*C)*d^2 + 4*b^2*c*(6*c*C - 11*B*d) - a*b*d*(48*c*C + 55*B*d)))/4 - (b*(99*b
*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C - 11*b*B*d - 6*a*C*d)))/4)
- b*((-693*(a^2*B - b^2*B + 2*a*b*(A - C))*d^3)/8 + (c*(99*b*(A*b + a*B - b
*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C - 11*b*B*d - 6*a*C*d)))/4))*((2*(c + d*Tan
[e + f*x])^(3/2))/3 + (c + I*d)*((2*(c + I*d)^(3/2)*ArcTanh[Sqrt[c + d*Tan[
e + f*x]]/Sqrt[c + I*d]])/(-c - I*d) + 2*Sqrt[c + d*Tan[e + f*x]])))/f))/(7
*d)))/(9*d)))/(11*d)
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 10951 vs. $2(507) = 1014$.

Time = 0.34 (sec) , antiderivative size = 10952, normalized size of antiderivative = 19.91

method	result	size
parts	Expression too large to display	10952
derivativedivides	Expression too large to display	11056
default	Expression too large to display	11056

```
[In] int((a+b*tan(f*x+e))^3*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x,method=_RETURNVERBOSE)
```

```
[Out] result too large to display
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 84950 vs. $2(498) = 996$.

Time = 194.27 (sec) , antiderivative size = 84950, normalized size of antiderivative = 154.45

$$\int (a + b\tan(e + fx))^3 (c + d\tan(e + fx))^{3/2} \left(A + B\tan(e + fx) + C\tan^2(e + fx)\right) dx = \text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))^3*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f
*x+e)^2),x, algorithm="fricas")
```

```
[Out] Too large to include
```

Sympy [F]

$$\int (a+b\tan(e+fx))^3(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\int (a+b\tan{(e+fx)})^3\,(c+d\tan{(e+fx)})^{\frac{3}{2}}\left(A+B\tan{(e+fx)}+C\tan^2{(e+fx)}\right)\,dx$$

```
[In] integrate((a+b*tan(f*x+e))**3*(c+d*tan(f*x+e))**(3/2)*(A+B*tan(f*x+e)+C*tan
(f*x+e)**2),x)
[Out] Integral((a + b*tan(e + f*x))**3*(c + d*tan(e + f*x))**(3/2)*(A + B*tan(e +
 f*x) + C*tan(e + f*x)**2), x)
```

Maxima [F(-1)]

Timed out.

$$\int (a+b\tan(e+fx))^3(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^3*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f
*x+e)^2),x, algorithm="maxima")
[Out] Timed out
```

Giac [F(-1)]

Timed out.

$$\int (a+b\tan(e+fx))^3(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^3*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f
*x+e)^2),x, algorithm="giac")
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int (a+b\tan(e+fx))^3(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\text{Hanged}$$

```
[In] int((a + b*tan(e + f*x))^3*(c + d*tan(e + f*x))^(3/2)*(A + B*tan(e + f*x) +
 C*tan(e + f*x)^2),x)
```

```
[Out] \text{Hanged}
```

3.98 $\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e\right.$

Optimal result 1287
Rubi [A] (verified) 1288
Mathematica [A] (verified) 1293
Maple [B] (verified) 1294
Fricas [B] (verification not implemented) 1294
Sympy [F] 1295
Maxima [F(-1)] 1295
Giac [F(-1)] 1295
Mupad [F(-1)] 1296

Optimal result

Integrand size = 47, antiderivative size = 396

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = -\frac{(a-ib)^2(B+i(A-C))(c-id)^{3/2}\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f} + \frac{(a+ib)^2(iA-B-iC)(c+id)^{3/2}\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f} + \frac{2(2ab(Ac-cC-Bd)+a^2(Bc+(A-C)d)-b^2(Bc+(A-C)d))\sqrt{c+d\tan(e+fx)}}{f} + \frac{2(a^2B-b^2B+2ab(A-C))(c+d\tan(e+fx))^{3/2}}{3f} + \frac{2(28a^2Cd^2-18abd(2cC-7Bd)+b^2(8c^2C-18Bcd+63(A-C)d^2))(c+d\tan(e+fx))^{5/2}}{315d^3f} - \frac{2b(4bcC-9bBd-4aCd)\tan(e+fx)(c+d\tan(e+fx))^{5/2}}{63d^2f} + \frac{2C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}}{9df}$$

```
[Out] -(a-I*b)^2*(B+I*(A-C))*(c-I*d)^(3/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)
^(1/2))/f+(a+I*b)^2*(I*A-B-I*C)*(c+I*d)^(3/2)*arctanh((c+d*tan(f*x+e))^(1/2
)/(c+I*d)^(1/2))/f+2*(2*a*b*(A*c-B*d-C*c)+a^2*(B*c+(A-C)*d)-b^2*(B*c+(A-C)*
d))*(c+d*tan(f*x+e))^(1/2)/f+2/3*(B*a^2-B*b^2+2*a*b*(A-C))*(c+d*tan(f*x+e))
^(3/2)/f+2/315*(28*a^2*C*d^2-18*a*b*d*(-7*B*d+2*C*c)+b^2*(8*c^2*C-18*B*c*d+
63*(A-C)*d^2))*(c+d*tan(f*x+e))^(5/2)/d^3/f-2/63*b*(-9*B*b*d-4*C*a*d+4*C*b*
c)*tan(f*x+e)*(c+d*tan(f*x+e))^(5/2)/d^2/f+2/9*C*(a+b*tan(f*x+e))^2*(c+d*ta
n(f*x+e))^(5/2)/d/f
```

Rubi [A] (verified)

Time = 1.83 (sec) , antiderivative size = 396, normalized size of antiderivative = 1.00, number of steps used = 12, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.170, Rules used = {3728, 3718, 3711, 3609, 3620, 3618, 65, 214}

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \frac{2(c+d\tan(e+fx))^{5/2}\left(28a^2Cd^2-18abd(2cC-7Bd)+b^2(63d^2(A-C)-18Bcd+\right.}{315d^3f}$$

$$+\frac{2(a^2B+2ab(A-C)-b^2B)\,(c+d\tan(e+fx))^{3/2}}{3f}$$

$$+\frac{2\sqrt{c+d\tan(e+fx)}(a^2(d(A-C)+Bc)+2ab(Ac-Bd-cC)-b^2(d(A-C)+Bc))}{f}$$

$$-\frac{(a-ib)^2(c-id)^{3/2}(B+i(A-C))\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f}$$

$$+\frac{(a+ib)^2(c+id)^{3/2}(iA-B-iC)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f}$$

$$-\frac{2b\tan(e+fx)(-4aCd-9bBd+4bcC)(c+d\tan(e+fx))^{5/2}}{63d^2f}$$

$$+\frac{2C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}}{9df}$$

```
[In] Int[(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] +
 C*Tan[e + f*x]^2),x]
```

```
[Out] -(((a - I*b)^2*(B + I*(A - C))*(c - I*d)^(3/2)*ArcTanh[Sqrt[c + d*Tan[e + f
*x]]/Sqrt[c - I*d]])/f) + ((a + I*b)^2*(I*A - B - I*C)*(c + I*d)^(3/2)*ArcT
anh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]])/f + (2*(2*a*b*(A*c - c*C - B*d
) + a^2*(B*c + (A - C)*d) - b^2*(B*c + (A - C)*d))*Sqrt[c + d*Tan[e + f*x]]
)/f + (2*(a^2*B - b^2*B + 2*a*b*(A - C))*(c + d*Tan[e + f*x])^(3/2))/(3*f)
+ (2*(28*a^2*C*d^2 - 18*a*b*d*(2*c*C - 7*B*d) + b^2*(8*c^2*C - 18*B*c*d + 6
3*(A - C)*d^2))*(c + d*Tan[e + f*x])^(5/2))/(315*d^3*f) - (2*b*(4*b*c*C - 9
*b*B*d - 4*a*C*d)*Tan[e + f*x]*(c + d*Tan[e + f*x])^(5/2))/(63*d^2*f) + (2*
C*(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])^(5/2))/(9*d*f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3609

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[d*((a + b*Tan[e + f*x])^m/(f*m)), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\text{integral} = \frac{2C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}}{9df} + \frac{2\int(a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}\left(\frac{1}{2}(-4bcC+a(9A-5C)d)+\frac{9}{2}(Ab+aB-bC)d\tan(e+f}{9d}$$

$$= -\frac{2b(4bcC-9bBd-4aCd)\tan(e+fx)(c+d\tan(e+fx))^{5/2}}{63d^2f} + \frac{2C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}}{9df} - \frac{4\int(c+d\tan(e+fx))^{3/2}\left(\frac{1}{4}\left(36abcCd-7a^2(9A-5C)d^2-4b^2\left(2c^2C-\frac{9Bcd}{2}\right)\right)-\frac{63}{4}(a^2B-b^2B}{}$$

$$= \frac{2(28a^2Cd^2-18abd(2cC-7Bd)+b^2(8c^2C-18Bcd+63(A-C)d^2))(c+d\tan(e+fx))^{5/2}}{315d^3f} - \frac{2b(4bcC-9bBd-4aCd)\tan(e+fx)(c+d\tan(e+fx))^{5/2}}{63d^2f} + \frac{2C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}}{9df} - \frac{4\int(c+d\tan(e+fx))^{3/2}\left(\frac{63}{4}(2abB-a^2(A-C)+b^2(A-C))d^2-\frac{63}{4}(a^2B-b^2B+2ab(A-C}{63d^2}$$

$$= \frac{2(a^2B - b^2B + 2ab(A-C))\,(c+d\tan(e+fx))^{3/2}}{3f}$$
$$+ \frac{2(28a^2Cd^2 - 18abd(2cC-7Bd) + b^2(8c^2C - 18Bcd + 63(A-C)d^2))\,(c+d\tan(e+fx))^{5/2}}{315d^3f}$$
$$- \frac{2b(4bcC - 9bBd - 4aCd)\tan(e+fx)(c+d\tan(e+fx))^{5/2}}{63d^2f}$$
$$+ \frac{2C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}}{9df}$$
$$- \frac{4\int \sqrt{c+d\tan(e+fx)}\left(-\frac{63}{4}d^2(a^2(Ac-cC-Bd) - b^2(Ac-cC-Bd) - 2ab(Bc+(A-C)d\right.}{63d}$$

$$= \frac{2(2ab(Ac-cC-Bd) + a^2(Bc+(A-C)d) - b^2(Bc+(A-C)d))\sqrt{c+d\tan(e+fx)}}{f}$$
$$+ \frac{2(a^2B - b^2B + 2ab(A-C))\,(c+d\tan(e+fx))^{3/2}}{3f}$$
$$+ \frac{2(28a^2Cd^2 - 18abd(2cC-7Bd) + b^2(8c^2C - 18Bcd + 63(A-C)d^2))\,(c+d\tan(e+fx))^{5/2}}{315d^3f}$$
$$- \frac{2b(4bcC - 9bBd - 4aCd)\tan(e+fx)(c+d\tan(e+fx))^{5/2}}{63d^2f}$$
$$+ \frac{2C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}}{9df}$$
$$- \frac{4\int \frac{\frac{63}{4}d^2\left(a^2\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-b^2\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)+2ab\left(2c(A-C)d+B\left(c^2-d^2\right)\right)\right)+\frac{63}{4}d^2\left(2ab\left(c^2C+\right.\right.}{\sqrt{c+d\tan(e+fx)}}}{63d^2}$$

$$= \frac{2(2ab(Ac-cC-Bd) + a^2(Bc+(A-C)d) - b^2(Bc+(A-C)d))\sqrt{c+d\tan(e+fx)}}{f}$$
$$+ \frac{2(a^2B - b^2B + 2ab(A-C))\,(c+d\tan(e+fx))^{3/2}}{3f}$$
$$+ \frac{2(28a^2Cd^2 - 18abd(2cC-7Bd) + b^2(8c^2C - 18Bcd + 63(A-C)d^2))\,(c+d\tan(e+fx))^{5/2}}{315d^3f}$$
$$- \frac{2b(4bcC - 9bBd - 4aCd)\tan(e+fx)(c+d\tan(e+fx))^{5/2}}{63d^2f}$$
$$+ \frac{2C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}}{9df}$$
$$+ \frac{1}{2}\left((a-ib)^2(A-iB-C)(c-id)^2\right)\int \frac{1+i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx + \frac{1}{2}\left((a+ib)^2(A+iB-C)(c+id)^2\right)\int$$

$$= \frac{2(2ab(Ac-cC-Bd)+a^2(Bc+(A-C)d)-b^2(Bc+(A-C)d))\sqrt{c+d\tan(e+fx)}}{f} + \frac{2(a^2B-b^2B+2ab(A-C))\,(c+d\tan(e+fx))^{3/2}}{3f} + \frac{2(28a^2Cd^2-18abd(2cC-7Bd)+b^2(8c^2C-18Bcd+63(A-C)d^2))\,(c+d\tan(e+fx))^{5/2}}{315d^3f} - \frac{2b(4bcC-9bBd-4aCd)\tan(e+fx)(c+d\tan(e+fx))^{5/2}}{63d^2f} + \frac{2C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}}{9df} + \frac{((a-ib)^2(iA+B-iC)(c-id)^2)\,\text{Subst}\left(\int\frac{1}{(-1+x)\sqrt{c-idx}}\,dx,x,i\tan(e+fx)\right)}{2f} - \frac{(i(a+ib)^2(A+iB-C)(c+id)^2)\,\text{Subst}\left(\int\frac{1}{(-1+x)\sqrt{c+idx}}\,dx,x,-i\tan(e+fx)\right)}{2f}$$

$$= \frac{2(2ab(Ac-cC-Bd)+a^2(Bc+(A-C)d)-b^2(Bc+(A-C)d))\sqrt{c+d\tan(e+fx)}}{f} + \frac{2(a^2B-b^2B+2ab(A-C))\,(c+d\tan(e+fx))^{3/2}}{3f} + \frac{2(28a^2Cd^2-18abd(2cC-7Bd)+b^2(8c^2C-18Bcd+63(A-C)d^2))\,(c+d\tan(e+fx))^{5/2}}{315d^3f} - \frac{2b(4bcC-9bBd-4aCd)\tan(e+fx)(c+d\tan(e+fx))^{5/2}}{63d^2f} + \frac{2C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}}{9df} - \frac{((a-ib)^2(A-iB-C)(c-id)^2)\,\text{Subst}\left(\int\frac{1}{-1-\frac{ic}{d}+\frac{ix^2}{d}}\,dx,x,\sqrt{c+d\tan(e+fx)}\right)}{df} - \frac{((a+ib)^2(A+iB-C)(c+id)^2)\,\text{Subst}\left(\int\frac{1}{-1+\frac{ic}{d}-\frac{ix^2}{d}}\,dx,x,\sqrt{c+d\tan(e+fx)}\right)}{df}$$

$$
\begin{aligned}
&= -\frac{(a-ib)^2(iA+B-iC)(c-id)^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f} \\
&\quad - \frac{(a+ib)^2(B-i(A-C))(c+id)^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f} \\
&\quad + \frac{2(2ab(Ac-cC-Bd)+a^2(Bc+(A-C)d)-b^2(Bc+(A-C)d))\sqrt{c+d\tan(e+fx)}}{f} \\
&\quad + \frac{2(a^2B-b^2B+2ab(A-C))\,(c+d\tan(e+fx))^{3/2}}{3f} \\
&\quad + \frac{2(28a^2Cd^2-18abd(2cC-7Bd)+b^2(8c^2C-18Bcd+63(A-C)d^2))\,(c+d\tan(e+fx))^{5/2}}{315d^3f} \\
&\quad - \frac{2b(4bcC-9bBd-4aCd)\tan(e+fx)(c+d\tan(e+fx))^{5/2}}{63d^2f} \\
&\quad + \frac{2C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}}{9df}
\end{aligned}
$$

Mathematica [A] (verified)

Time = 6.43 (sec) , antiderivative size = 510, normalized size of antiderivative = 1.29

$$
\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \frac{2C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}}{9df}
$$

$$
+ \frac{2\left(\frac{b(-4bcC+9bBd+4aCd)\tan(e+fx)(c+d\tan(e+fx))^{5/2}}{7df} - \frac{2\left(\frac{\left(-28a^2Cd^2+18abd(2cC-7Bd)-b^2\left(8c^2C-18Bcd+63(A-C)d^2\right)\right)(c+d\tan(e+fx))^{5/2}}{10df}\right.}{}\right.}{}
$$

[In] Integrate[(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2),x]

[Out] (2*C*(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])^(5/2))/(9*d*f) + (2*((b*(-4*b*c*C + 9*b*B*d + 4*a*C*d)*Tan[e + f*x]*(c + d*Tan[e + f*x])^(5/2))/(7*d*f) - (2*(((-28*a^2*C*d^2 + 18*a*b*d*(2*c*C - 7*B*d) - b^2*(8*c^2*C - 18*B*c*d + 63*(A - C)*d^2))*(c + d*Tan[e + f*x])^(5/2))/(10*d*f) + ((I/2)*(((63*I)/4)*(a^2*B - b^2*B + 2*a*b*(A - C))*d^2 + (63*(2*a*b*B - a^2*(A - C) + b^2*(A - C))*d^2)/4)*((2*(c + d*Tan[e + f*x])^(3/2))/3 + (c - I*d)*((2*(c - I*

```
d)^(3/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]])/(-c + I*d) + 2*Sq
rt[c + d*Tan[e + f*x]])))/f - ((I/2)*(((-63*I)/4)*(a^2*B - b^2*B + 2*a*b*(A
 - C))*d^2 + (63*(2*a*b*B - a^2*(A - C) + b^2*(A - C))*d^2)/4)*((2*(c + d*T
an[e + f*x])^(3/2))/3 + (c + I*d)*((2*(c + I*d)^(3/2)*ArcTanh[Sqrt[c + d*Ta
n[e + f*x]]/Sqrt[c + I*d]])/(-c - I*d) + 2*Sqrt[c + d*Tan[e + f*x]])))/f))/
(7*d)))/(9*d)
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 7938 vs. $2(357) = 714$.

Time = 0.21 (sec) , antiderivative size = 7939, normalized size of antiderivative = 20.05

method	result	size
parts	Expression too large to display	7939
derivativedivides	Expression too large to display	8031
default	Expression too large to display	8031

```
[In] int((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x,method=_RETURNVERBOSE)
```

```
[Out] result too large to display
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 58971 vs. $2(347) = 694$.

Time = 69.93 (sec) , antiderivative size = 58971, normalized size of antiderivative = 148.92

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f
*x+e)^2),x, algorithm="fricas")
```

```
[Out] Too large to include
```

Sympy [F]

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\int (a+b\tan{(e+fx)})^2\,(c+d\tan{(e+fx)})^{\frac{3}{2}}\left(A+B\tan{(e+fx)}+C\tan^2{(e+fx)}\right)\,dx$$

```
[In] integrate((a+b*tan(f*x+e))**2*(c+d*tan(f*x+e))**(3/2)*(A+B*tan(f*x+e)+C*tan
(f*x+e)**2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**2*(c + d*tan(e + f*x))**(3/2)*(A + B*tan(e +
 f*x) + C*tan(e + f*x)**2), x)
```

Maxima [F(-1)]

Timed out.

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f
*x+e)^2),x, algorithm="maxima")
```

```
[Out] Timed out
```

Giac [F(-1)]

Timed out.

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f
*x+e)^2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\text{Hanged}$$

```
[In] int((a + b*tan(e + f*x))^2*(c + d*tan(e + f*x))^(3/2)*(A + B*tan(e + f*x) +
 C*tan(e + f*x)^2),x)
```

```
[Out] \text{Hanged}
```

3.99 $\int (a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+$

Optimal result 1297
Rubi [A] (verified) 1298
Mathematica [A] (verified) 1302
Maple [B] (verified) 1302
Fricas [B] (verification not implemented) 1303
Sympy [F] 1303
Maxima [F(-1)] 1303
Giac [F(-1)] 1304
Mupad [F(-1)] 1304

Optimal result

Integrand size = 45, antiderivative size = 273

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = -\frac{(ia+b)(A-iB-C)(c-id)^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f} + \frac{(ia-b)(A+iB-C)(c+id)^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f} + \frac{2(Abc+aBc-bcC+aAd-bBd-aCd)\sqrt{c+d\tan(e+fx)}}{f} + \frac{2(Ab+aB-bC)(c+d\tan(e+fx))^{3/2}}{3f} - \frac{2(2bcC-7bBd-7aCd)(c+d\tan(e+fx))^{5/2}}{35d^2f} + \frac{2bC\tan(e+fx)(c+d\tan(e+fx))^{5/2}}{7df}$$

```
[Out] -(I*a+b)*(A-I*B-C)*(c-I*d)^(3/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/
2))/f+(I*a-b)*(A+I*B-C)*(c+I*d)^(3/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d
)^(1/2))/f+2*(A*a*d+A*b*c+B*a*c-B*b*d-C*a*d-C*b*c)*(c+d*tan(f*x+e))^(1/2)/f
+2/3*(A*b+B*a-C*b)*(c+d*tan(f*x+e))^(3/2)/f-2/35*(-7*B*b*d-7*C*a*d+2*C*b*c)
*(c+d*tan(f*x+e))^(5/2)/d^2/f+2/7*b*C*tan(f*x+e)*(c+d*tan(f*x+e))^(5/2)/d/f
```

Rubi [A] (verified)

Time = 1.00 (sec) , antiderivative size = 273, normalized size of antiderivative = 1.00, number of steps used = 11, number of rules used = 7, $\frac{\text{number of rules}}{\text{integrand size}} = 0.156$, Rules used = {3718, 3711, 3609, 3620, 3618, 65, 214}

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = -\frac{(b+ia)(c-id)^{3/2}(A-iB-C)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f} + \frac{(-b+ia)(c+id)^{3/2}(A+iB-C)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f} + \frac{2(aB+Ab-bC)(c+d\tan(e+fx))^{3/2}}{3f} + \frac{2\sqrt{c+d\tan(e+fx)}(aAd+aBc-aCd+Abc-bBd-bcC)}{f} - \frac{2(-7aCd-7bBd+2bcC)(c+d\tan(e+fx))^{5/2}}{35d^2f} + \frac{2bC\tan(e+fx)(c+d\tan(e+fx))^{5/2}}{7df}$$

```
[In] Int[(a + b*Tan[e + f*x])*(c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C
*Tan[e + f*x]^2),x]
```

```
[Out] -(((I*a + b)*(A - I*B - C)*(c - I*d)^(3/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]
/Sqrt[c - I*d]])/f) + ((I*a - b)*(A + I*B - C)*(c + I*d)^(3/2)*ArcTanh[Sqrt
[c + d*Tan[e + f*x]]/Sqrt[c + I*d]])/f + (2*(A*b*c + a*B*c - b*c*C + a*A*d
- b*B*d - a*C*d)*Sqrt[c + d*Tan[e + f*x]])/f + (2*(A*b + a*B - b*C)*(c + d*
Tan[e + f*x])^(3/2))/(3*f) - (2*(2*b*c*C - 7*b*B*d - 7*a*C*d)*(c + d*Tan[e
+ f*x])^(5/2))/(35*d^2*f) + (2*b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(5/2))
/(7*d*f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3609

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[d*((a + b*Tan[e + f*x])^m/(f*m)), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rubi steps

$$\text{integral} = \frac{2bC\tan(e+fx)(c+d\tan(e+fx))^{5/2}}{7df} - \frac{2\int(c+d\tan(e+fx))^{3/2}\left(\frac{1}{2}(2bcC-7aAd)-\frac{7}{2}(Ab+aB-bC)d\tan(e+fx)+\frac{1}{2}(2bcC-7bBd-7aCd\ldots\right.}{7d}$$

$$= -\frac{2(2bcC-7bBd-7aCd)(c+d\tan(e+fx))^{5/2}}{35d^2f} + \frac{2bC\tan(e+fx)(c+d\tan(e+fx))^{5/2}}{7df} - \frac{2\int(c+d\tan(e+fx))^{3/2}\left(\frac{7}{2}(bB-a(A-C))d-\frac{7}{2}(Ab+aB-bC)d\tan(e+fx)\right)dx}{7d}$$

$$= \frac{2(Ab+aB-bC)(c+d\tan(e+fx))^{3/2}}{3f} - \frac{2(2bcC-7bBd-7aCd)(c+d\tan(e+fx))^{5/2}}{35d^2f} + \frac{2bC\tan(e+fx)(c+d\tan(e+fx))^{5/2}}{7df} - \frac{2\int\sqrt{c+d\tan(e+fx)}\left(\frac{7}{2}d(bBc+b(A-C)d-a(Ac-cC-Bd))-\frac{7}{2}d(Abc+aBc-bcC+aA\ldots\right.}{7d}$$

$$= \frac{2(Abc+aBc-bcC+aAd-bBd-aCd)\sqrt{c+d\tan(e+fx)}}{f} + \frac{2(Ab+aB-bC)(c+d\tan(e+fx))^{3/2}}{3f} - \frac{2(2bcC-7bBd-7aCd)(c+d\tan(e+fx))^{5/2}}{35d^2f} + \frac{2bC\tan(e+fx)(c+d\tan(e+fx))^{5/2}}{7df} - \frac{2\int\frac{\frac{7}{2}d\left(a\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)+b\left(2c(A-C)d+B\left(c^2-d^2\right)\right)\right)-\frac{7}{2}d\left(2aAcd-2acCd+Ab\left(c^2-d^2\right)+aB\left(c^2-d^2\right)-b\left(c^2C+2Bc\ldots\right.\right.}{\sqrt{c+d\tan(e+fx)}}}{7d}$$

$$= \frac{2(Abc+aBc-bcC+aAd-bBd-aCd)\sqrt{c+d\tan(e+fx)}}{f} + \frac{2(Ab+aB-bC)(c+d\tan(e+fx))^{3/2}}{3f} - \frac{2(2bcC-7bBd-7aCd)(c+d\tan(e+fx))^{5/2}}{35d^2f} + \frac{2bC\tan(e+fx)(c+d\tan(e+fx))^{5/2}}{7df} + \frac{1}{2}\left((a-ib)(A-iB-C)(c-id)^2\right)\int\frac{1+i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}dx + \frac{1}{2}\left((a+ib)(A+iB-C)(c+id)^2\right)\int\frac{1\ldots}{\sqrt{\ldots}}$$

$$= \frac{2(Abc + aBc - bcC + aAd - bBd - aCd)\sqrt{c + d\tan(e + fx)}}{f} + \frac{2(Ab + aB - bC)(c + d\tan(e + fx))^{3/2}}{3f} - \frac{2(2bcC - 7bBd - 7aCd)(c + d\tan(e + fx))^{5/2}}{35d^2f} + \frac{2bC\tan(e + fx)(c + d\tan(e + fx))^{5/2}}{7df} + \frac{((ia + b)(A - iB - C)(c - id)^2)\operatorname{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c-idx}}\,dx, x, i\tan(e + fx)\right)}{2f} - \frac{((ia - b)(A + iB - C)(c + id)^2)\operatorname{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c+idx}}\,dx, x, -i\tan(e + fx)\right)}{2f}$$

$$= \frac{2(Abc + aBc - bcC + aAd - bBd - aCd)\sqrt{c + d\tan(e + fx)}}{f} + \frac{2(Ab + aB - bC)(c + d\tan(e + fx))^{3/2}}{3f} - \frac{2(2bcC - 7bBd - 7aCd)(c + d\tan(e + fx))^{5/2}}{35d^2f} + \frac{2bC\tan(e + fx)(c + d\tan(e + fx))^{5/2}}{7df} - \frac{((a - ib)(A - iB - C)(c - id)^2)\operatorname{Subst}\left(\int \frac{1}{-1-\frac{ic}{d}+\frac{ix^2}{d}}\,dx, x, \sqrt{c + d\tan(e + fx)}\right)}{df} - \frac{((a + ib)(A + iB - C)(c + id)^2)\operatorname{Subst}\left(\int \frac{1}{-1+\frac{ic}{d}-\frac{ix^2}{d}}\,dx, x, \sqrt{c + d\tan(e + fx)}\right)}{df}$$

$$= -\frac{(a - ib)(iA + B - iC)(c - id)^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f} + \frac{(ia - b)(A + iB - C)(c + id)^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f} + \frac{2(Abc + aBc - bcC + aAd - bBd - aCd)\sqrt{c + d\tan(e + fx)}}{f} + \frac{2(Ab + aB - bC)(c + d\tan(e + fx))^{3/2}}{3f} - \frac{2(2bcC - 7bBd - 7aCd)(c + d\tan(e + fx))^{5/2}}{35d^2f} + \frac{2bC\tan(e + fx)(c + d\tan(e + fx))^{5/2}}{7df}$$

Mathematica [A] (verified)

Time = 4.84 (sec) , antiderivative size = 260, normalized size of antiderivative = 0.95

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \frac{\frac{2(-2bcC+7bBd+7aCd)(c+d\tan(e+fx))^{5/2}}{d}+10bC\tan(e+fx)(c+d\tan(e+fx))^{5/2}+\frac{35}{3}(ia+$$

[In] Integrate[(a + b*Tan[e + f*x])*(c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2),x]

[Out] ((2*(-2*b*c*C + 7*b*B*d + 7*a*C*d)*(c + d*Tan[e + f*x])^(5/2))/d + 10*b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(5/2) + (35*(I*a + b)*(A - I*B - C)*d*(-3*(c - I*d)^(3/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]] + Sqrt[c + d*Tan[e + f*x]]*(4*c - (3*I)*d + d*Tan[e + f*x])))/3 + (35*((-I)*a + b)*(A + I*B - C)*d*(-3*(c + I*d)^(3/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]] + Sqrt[c + d*Tan[e + f*x]]*(4*c + (3*I)*d + d*Tan[e + f*x])))/3)/(35*d*f)

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 5106 vs. 2(239) = 478.

Time = 0.19 (sec) , antiderivative size = 5107, normalized size of antiderivative = 18.71

method	result	size
parts	Expression too large to display	5107
derivativedivides	Expression too large to display	5149
default	Expression too large to display	5149

[In] int((a+b*tan(f*x+e))*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x,method=_RETURNVERBOSE)

[Out] result too large to display

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 31081 vs. $2(232) = 464$.

Time = 16.16 (sec) , antiderivative size = 31081, normalized size of antiderivative = 113.85

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x, algorithm="fricas")
```

```
[Out] Too large to include
```

Sympy [F]

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \int (a+b\tan{(e+fx)})(c+d\tan{(e+fx)})^{\frac{3}{2}}\left(A+B\tan{(e+fx)}+C\tan^2{(e+fx)}\right)\,dx$$

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))**(3/2)*(A+B*tan(f*x+e)+C*tan(f*
x+e)**2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))*(c + d*tan(e + f*x))**(3/2)*(A + B*tan(e + f*
x) + C*tan(e + f*x)**2), x)
```

Maxima [F(-1)]

Timed out.

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x, algorithm="maxima")
```

```
[Out] Timed out
```

Giac [F(-1)]

Timed out.

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\text{Hanged}$$

```
[In] int((a + b*tan(e + f*x))*(c + d*tan(e + f*x))^(3/2)*(A + B*tan(e + f*x) + C
*tan(e + f*x)^2),x)
```

```
[Out] \text{Hanged}
```

3.100 $\int (c+d\tan(e+fx))^{3/2} (A + B\tan(e + fx) + C\tan^2($

Optimal result 1305
Rubi [A] (verified) 1305
Mathematica [A] (verified) 1308
Maple [B] (verified) 1309
Fricas [B] (verification not implemented) 1310
Sympy [F] 1311
Maxima [F] 1311
Giac [F(-1)] 1311
Mupad [B] (verification not implemented) 1312

Optimal result

Integrand size = 35, antiderivative size = 187

$$\int (c+d\tan(e+fx))^{3/2} \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = -\frac{(iA+B-iC)(c-id)^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f} - \frac{(B-i(A-C))(c+id)^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f} + \frac{2(Bc+(A-C)d)\sqrt{c+d\tan(e+fx)}}{f} + \frac{2B(c+d\tan(e+fx))^{3/2}}{3f} + \frac{2C(c+d\tan(e+fx))^{5/2}}{5df}$$

```
[Out] -(I*A+B-I*C)*(c-I*d)^(3/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/f-
(B-I*(A-C))*(c+I*d)^(3/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))/f+2
*(B*c+(A-C)*d)*(c+d*tan(f*x+e))^(1/2)/f+2/3*B*(c+d*tan(f*x+e))^(3/2)/f+2/5*
C*(c+d*tan(f*x+e))^(5/2)/d/f
```

Rubi [A] (verified)

Time = 0.49 (sec) , antiderivative size = 187, normalized size of antiderivative = 1.00, number of steps used = 10, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}} = 0.171$, Rules used

$= \{3711, 3609, 3620, 3618, 65, 214\}$

$$\int (c+d\tan(e+fx))^{3/2} \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right) dx = -\frac{(c-id)^{3/2}(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f} - \frac{(c+id)^{3/2}(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f} + \frac{2(d(A-C)+Bc)\sqrt{c+d\tan(e+fx)}}{f} + \frac{2B(c+d\tan(e+fx))^{3/2}}{3f} + \frac{2C(c+d\tan(e+fx))^{5/2}}{5df}$$

```
[In] Int[(c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] -(((I*A + B - I*C)*(c - I*d)^(3/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c
- I*d]])/f) - ((B - I*(A - C))*(c + I*d)^(3/2)*ArcTanh[Sqrt[c + d*Tan[e + f
*x]]/Sqrt[c + I*d]])/f + (2*(B*c + (A - C)*d)*Sqrt[c + d*Tan[e + f*x]])/f +
 (2*B*(c + d*Tan[e + f*x])^(3/2))/(3*f) + (2*C*(c + d*Tan[e + f*x])^(5/2))/
(5*d*f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3609

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[d*((a + b*Tan[e + f*x])^m/(f*m)), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
```

```
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rubi steps

$$
\begin{aligned}
\text{integral} &= \frac{2C(c+d\tan(e+fx))^{5/2}}{5df} + \int (A-C+B\tan(e+fx))(c+d\tan(e+fx))^{3/2}\,dx \\
&= \frac{2B(c+d\tan(e+fx))^{3/2}}{3f} + \frac{2C(c+d\tan(e+fx))^{5/2}}{5df} \\
&\quad + \int \sqrt{c+d\tan(e+fx)}(Ac-cC-Bd+(Bc+(A-C)d)\tan(e+fx))\,dx \\
&= \frac{2(Bc+(A-C)d)\sqrt{c+d\tan(e+fx)}}{f} \\
&\quad + \frac{2B(c+d\tan(e+fx))^{3/2}}{3f} + \frac{2C(c+d\tan(e+fx))^{5/2}}{5df} \\
&\quad + \int \frac{-c^2C-2Bcd+Cd^2+A(c^2-d^2)+(2c(A-C)d+B(c^2-d^2))\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx \\
&= \frac{2(Bc+(A-C)d)\sqrt{c+d\tan(e+fx)}}{f} + \frac{2B(c+d\tan(e+fx))^{3/2}}{3f} \\
&\quad + \frac{2C(c+d\tan(e+fx))^{5/2}}{5df} + \frac{1}{2}\left((A-iB-C)(c-id)^2\right)\int \frac{1+i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx \\
&\quad + \frac{1}{2}\left((A+iB-C)(c+id)^2\right)\int \frac{1-i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{2(Bc+(A-C)d)\sqrt{c+d\tan(e+fx)}}{f} \\
&\quad + \frac{2B(c+d\tan(e+fx))^{3/2}}{3f} + \frac{2C(c+d\tan(e+fx))^{5/2}}{5df} \\
&\quad + \frac{((iA+B-iC)(c-id)^2)\,\mathrm{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c-idx}}\,dx, x, i\tan(e+fx)\right)}{2f} \\
&\quad - \frac{(i(A+iB-C)(c+id)^2)\,\mathrm{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c+idx}}\,dx, x, -i\tan(e+fx)\right)}{2f} \\
&= \frac{2(Bc+(A-C)d)\sqrt{c+d\tan(e+fx)}}{f} \\
&\quad + \frac{2B(c+d\tan(e+fx))^{3/2}}{3f} + \frac{2C(c+d\tan(e+fx))^{5/2}}{5df} \\
&\quad - \frac{((A-iB-C)(c-id)^2)\,\mathrm{Subst}\left(\int \frac{1}{-1-\frac{ic}{d}+\frac{ix^2}{d}}\,dx, x, \sqrt{c+d\tan(e+fx)}\right)}{df} \\
&\quad - \frac{((A+iB-C)(c+id)^2)\,\mathrm{Subst}\left(\int \frac{1}{-1+\frac{ic}{d}-\frac{ix^2}{d}}\,dx, x, \sqrt{c+d\tan(e+fx)}\right)}{df} \\
&= -\frac{(iA+B-iC)(c-id)^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f} \\
&\quad - \frac{(B-i(A-C))(c+id)^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f} \\
&\quad + \frac{2(Bc+(A-C)d)\sqrt{c+d\tan(e+fx)}}{f} \\
&\quad + \frac{2B(c+d\tan(e+fx))^{3/2}}{3f} + \frac{2C(c+d\tan(e+fx))^{5/2}}{5df}
\end{aligned}
$$

Mathematica [A] (verified)

Time = 1.32 (sec) , antiderivative size = 202, normalized size of antiderivative = 1.08

$$
\int (c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \frac{\frac{6C(c+d\tan(e+fx))^{5/2}}{d} + 5(iA+B-iC)\left(-3(c-id)^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right) + \sqrt{c+d
$$

```
[In] Integrate[(c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2
),x]
```

```
[Out] ((6*C*(c + d*Tan[e + f*x])^(5/2))/d + 5*(I*A + B - I*C)*(-3*(c - I*d)^(3/2)
*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]] + Sqrt[c + d*Tan[e + f*x]]
*(4*c - (3*I)*d + d*Tan[e + f*x])) + 5*((-I)*A + B + I*C)*(-3*(c + I*d)^(3/
2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]] + Sqrt[c + d*Tan[e + f*x
]]*(4*c + (3*I)*d + d*Tan[e + f*x])))/(15*f)
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 2499 vs. 2(158) = 316.

Time = 0.14 (sec) , antiderivative size = 2500, normalized size of antiderivative = 13.37

method	result	size
parts	Expression too large to display	2500
derivativedivides	Expression too large to display	2517
default	Expression too large to display	2517

```
[In] int((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x,method=_RETURN
VERBOSE)
```

```
[Out] A*(2/f*d*(c+d*tan(f*x+e))^(1/2)+1/4/f/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d
^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)+2*c
)^(1/2)*(c^2+d^2)^(1/2)*c-1/4/f/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1
/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)+2*c)^(1/2
)*c^2+1/4/f*d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan
(f*x+e)-c-(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2/f*d/(2*(c^2+d^2)
^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^
(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*c+1/f*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)
*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^
2)^(1/2)-2*c)^(1/2))*(c^2+d^2)^(1/2)-1/4/f/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x
+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2
)+2*c)^(1/2)*(c^2+d^2)^(1/2)*c+1/4/f/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(
1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)+2*c)
^(1/2)*c^2-1/4/f*d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1
/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+2/f*d/(2*(c^2
+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+
2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*c-1/f*d/(2*(c^2+d^2)^(1/2)-2*c)^
(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c
^2+d^2)^(1/2)-2*c)^(1/2))*(c^2+d^2)^(1/2))+B*(2/3/f*(c+d*tan(f*x+e))^(3/2)+
2/f*(c+d*tan(f*x+e))^(1/2)*c-1/4/f*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(
1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)+2*c)^(1/
2)*(c^2+d^2)^(1/2)+1/2/f*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^
(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c+1/f/(
2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*t
an(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*(c^2+d^2)^(1/2)*c+1/f/(2*(
```

```
c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(
f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*(c^2+d^2)-2/f/(2*(c^2+d^2)^(1
/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/
2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*c^2+1/4/f*ln(d*tan(f*x+e)+c+(c+d*tan(f*x
+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2
)+2*c)^(1/2)*(c^2+d^2)^(1/2)-1/2/f*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)
*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)+2*c)^(1/
2)*c-1/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*
(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*(c^2+d^2)^(1/2)*
c-1/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^
2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*(c^2+d^2)+2/f/(2*(c
^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2
)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*c^2)+C*(2/5/f/d*(c+d*tan(f*x+e
))^(5/2)-2/f*d*(c+d*tan(f*x+e))^(1/2)-1/4/f/d*ln((c+d*tan(f*x+e))^(1/2)*(2*
(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/
2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*c+1/4/f/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d
^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)+2*c
)^(1/2)*c^2-1/4/f*d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)
-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-1/f*d/(2*(c^
2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*
x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*(c^2+d^2)^(1/2)+2/f*d/(2*(c^2+d
^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e
))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*c+1/4/f/d*ln(d*tan(f*x+e)+c+(c+d*t
an(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(2*(c^2+d^2
)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*c-1/4/f/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x
+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2
)+2*c)^(1/2)*c^2+1/4/f*d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d
^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+1/f*d/(
2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^
(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*(c^2+d^2)^(1/2)-2/f*d/(2*(
c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/
2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*c)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 6846 vs. $2(151) = 302$.

Time = 0.95 (sec) , antiderivative size = 6846, normalized size of antiderivative = 36.61

$$\int (c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \text{Too large to display}$$

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorit
hm="fricas")
```

```
[Out] Too large to include
```

Sympy [F]

$$\int (c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\int (c+d\tan{(e+fx)})^{\frac{3}{2}}\left(A+B\tan{(e+fx)}+C\tan^2{(e+fx)}\right)\,dx$$

[In] integrate((c+d*tan(f*x+e))**(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2),x)

[Out] Integral((c + d*tan(e + f*x))**(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)**2), x)

Maxima [F]

$$\int (c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\int \left(C\tan{(fx+e)}^2+B\tan{(fx+e)}+A\right)(d\tan{(fx+e)}+c)^{\frac{3}{2}}\,dx$$

[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorithm="maxima")

[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(d*tan(f*x + e) + c)^(3/2), x)

Giac [F(-1)]

Timed out.

$$\int (c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\text{Timed out}$$

[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorithm="giac")

[Out] Timed out

Mupad [B] (verification not implemented)

Time = 42.33 (sec) , antiderivative size = 4260, normalized size of antiderivative = 22.78

$$\int (c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\text{Too large to display}$$

```
[In] int((c + d*tan(e + f*x))^(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2),x)
```

```
[Out] ((2*C*c^2)/(d*f) - (2*C*(d^3*f + c^2*d*f))/(d^2*f^2))*(c + d*tan(e + f*x))^
(1/2) - log((((16*c*d^2*(((-B^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) + B^2*c^3*f^
2 - 3*B^2*c*d^2*f^2)/f^4)^(1/2)*(B*c^2 + B*d^2 + f*(((-B^4*d^2*f^4*(3*c^2 -
 d^2)^2)^(1/2) + B^2*c^3*f^2 - 3*B^2*c*d^2*f^2)/f^4)^(1/2)*(c + d*tan(e + f
*x))^(1/2)))/f - (16*B^2*d^2*(c + d*tan(e + f*x))^(1/2)*(c^4 + d^4 - 6*c^2*
d^2))/f^2)*(((-B^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) + B^2*c^3*f^2 - 3*B^2*c*d
^2*f^2)/f^4)^(1/2))/2 - (8*B^3*d^2*(c^2 - d^2)*(c^2 + d^2)^2)/f^3)*(((6*B^4
*c^2*d^4*f^4 - B^4*d^6*f^4 - 9*B^4*c^4*d^2*f^4)^(1/2) + B^2*c^3*f^2 - 3*B^2
*c*d^2*f^2)/(4*f^4))^(1/2) - log((((16*c*d^2*(-((-B^4*d^2*f^4*(3*c^2 - d^2)
^2)^(1/2) - B^2*c^3*f^2 + 3*B^2*c*d^2*f^2)/f^4)^(1/2)*(B*c^2 + B*d^2 + f*(-
((-B^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) - B^2*c^3*f^2 + 3*B^2*c*d^2*f^2)/f^4)
^(1/2)*(c + d*tan(e + f*x))^(1/2)))/f - (16*B^2*d^2*(c + d*tan(e + f*x))^(1
/2)*(c^4 + d^4 - 6*c^2*d^2))/f^2)*(-((-B^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) -
 B^2*c^3*f^2 + 3*B^2*c*d^2*f^2)/f^4)^(1/2))/2 - (8*B^3*d^2*(c^2 - d^2)*(c^2
 + d^2)^2)/f^3)*(-((6*B^4*c^2*d^4*f^4 - B^4*d^6*f^4 - 9*B^4*c^4*d^2*f^4)^(1
/2) - B^2*c^3*f^2 + 3*B^2*c*d^2*f^2)/(4*f^4))^(1/2) + log((((16*c*d^2*(((-B
^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) + B^2*c^3*f^2 - 3*B^2*c*d^2*f^2)/f^4)^(1/
2)*(B*c^2 + B*d^2 - f*(((-B^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) + B^2*c^3*f^2
- 3*B^2*c*d^2*f^2)/f^4)^(1/2)*(c + d*tan(e + f*x))^(1/2)))/f + (16*B^2*d^2*
(c + d*tan(e + f*x))^(1/2)*(c^4 + d^4 - 6*c^2*d^2))/f^2)*(((-B^4*d^2*f^4*(3
*c^2 - d^2)^2)^(1/2) + B^2*c^3*f^2 - 3*B^2*c*d^2*f^2)/f^4)^(1/2))/2 - (8*B^
3*d^2*(c^2 - d^2)*(c^2 + d^2)^2)/f^3)*((6*B^4*c^2*d^4*f^4 - B^4*d^6*f^4 - 9
*B^4*c^4*d^2*f^4)^(1/2)/(4*f^4) + (B^2*c^3)/(4*f^2) - (3*B^2*c*d^2)/(4*f^2)
)^(1/2) + log((((16*c*d^2*(-((-B^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) - B^2*c^3
*f^2 + 3*B^2*c*d^2*f^2)/f^4)^(1/2)*(B*c^2 + B*d^2 - f*(-((-B^4*d^2*f^4*(3*c
^2 - d^2)^2)^(1/2) - B^2*c^3*f^2 + 3*B^2*c*d^2*f^2)/f^4)^(1/2)*(c + d*tan(e
 + f*x))^(1/2)))/f + (16*B^2*d^2*(c + d*tan(e + f*x))^(1/2)*(c^4 + d^4 - 6*
c^2*d^2))/f^2)*(-((-B^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) - B^2*c^3*f^2 + 3*B^
2*c*d^2*f^2)/f^4)^(1/2))/2 - (8*B^3*d^2*(c^2 - d^2)*(c^2 + d^2)^2)/f^3)*((B
^2*c^3)/(4*f^2) - (6*B^4*c^2*d^4*f^4 - B^4*d^6*f^4 - 9*B^4*c^4*d^2*f^4)^(1/
2)/(4*f^4) - (3*B^2*c*d^2)/(4*f^2))^(1/2) - log((((16*d^2*(-((-A^4*d^2*f^4*
(3*c^2 - d^2)^2)^(1/2) + A^2*c^3*f^2 - 3*A^2*c*d^2*f^2)/f^4)^(1/2)*(A*d^3 +
 A*c^2*d + c*f*(-((-A^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) + A^2*c^3*f^2 - 3*A^
2*c*d^2*f^2)/f^4)^(1/2)*(c + d*tan(e + f*x))^(1/2)))/f + (16*A^2*d^2*(c + d
*tan(e + f*x))^(1/2)*(c^4 + d^4 - 6*c^2*d^2))/f^2)*(-((-A^4*d^2*f^4*(3*c^2
- d^2)^2)^(1/2) + A^2*c^3*f^2 - 3*A^2*c*d^2*f^2)/f^4)^(1/2))/2 - (16*A^3*c*
```

d^3*(c^2 + d^2)^2)/f^3)*(-((6*A^4*c^2*d^4*f^4 - A^4*d^6*f^4 - 9*A^4*c^4*d^2
*f^4)^(1/2) + A^2*c^3*f^2 - 3*A^2*c*d^2*f^2)/(4*f^4))^(1/2) - log((((16*d^2
*(((-A^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) - A^2*c^3*f^2 + 3*A^2*c*d^2*f^2)/f^
4)^(1/2)*(A*d^3 + A*c^2*d + c*f*(((-A^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) - A^
2*c^3*f^2 + 3*A^2*c*d^2*f^2)/f^4)^(1/2)*(c + d*tan(e + f*x))^(1/2)))/f + (1
6*A^2*d^2*(c + d*tan(e + f*x))^(1/2)*(c^4 + d^4 - 6*c^2*d^2))/f^2)*(((-A^4*
d^2*f^4*(3*c^2 - d^2)^2)^(1/2) - A^2*c^3*f^2 + 3*A^2*c*d^2*f^2)/f^4)^(1/2))
/2 - (16*A^3*c*d^3*(c^2 + d^2)^2)/f^3)*(((6*A^4*c^2*d^4*f^4 - A^4*d^6*f^4 -
9*A^4*c^4*d^2*f^4)^(1/2) - A^2*c^3*f^2 + 3*A^2*c*d^2*f^2)/(4*f^4))^(1/2) +
log((((16*d^2*(((-A^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) - A^2*c^3*f^2 + 3*A^2
*c*d^2*f^2)/f^4)^(1/2)*(A*d^3 + A*c^2*d - c*f*(((-A^4*d^2*f^4*(3*c^2 - d^2)
^2)^(1/2) - A^2*c^3*f^2 + 3*A^2*c*d^2*f^2)/f^4)^(1/2)*(c + d*tan(e + f*x))^
(1/2)))/f - (16*A^2*d^2*(c + d*tan(e + f*x))^(1/2)*(c^4 + d^4 - 6*c^2*d^2))
/f^2)*(((-A^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) - A^2*c^3*f^2 + 3*A^2*c*d^2*f^
2)/f^4)^(1/2))/2 - (16*A^3*c*d^3*(c^2 + d^2)^2)/f^3)*((6*A^4*c^2*d^4*f^4 -
A^4*d^6*f^4 - 9*A^4*c^4*d^2*f^4)^(1/2)/(4*f^4) - (A^2*c^3)/(4*f^2) + (3*A^2
*c*d^2)/(4*f^2))^(1/2) + log((((16*d^2*(-((-A^4*d^2*f^4*(3*c^2 - d^2)^2)^(1
/2) + A^2*c^3*f^2 - 3*A^2*c*d^2*f^2)/f^4)^(1/2)*(A*d^3 + A*c^2*d - c*f*(-((
-A^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) + A^2*c^3*f^2 - 3*A^2*c*d^2*f^2)/f^4)^(
1/2)*(c + d*tan(e + f*x))^(1/2)))/f - (16*A^2*d^2*(c + d*tan(e + f*x))^(1/2
)*(c^4 + d^4 - 6*c^2*d^2))/f^2)*(-((-A^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) + A
^2*c^3*f^2 - 3*A^2*c*d^2*f^2)/f^4)^(1/2))/2 - (16*A^3*c*d^3*(c^2 + d^2)^2)/
f^3)*((3*A^2*c*d^2)/(4*f^2) - (A^2*c^3)/(4*f^2) - (6*A^4*c^2*d^4*f^4 - A^4*
d^6*f^4 - 9*A^4*c^4*d^2*f^4)^(1/2)/(4*f^4))^(1/2) - log((16*C^3*c*d^3*(c^2
+ d^2)^2)/f^3 - (((16*d^2*(-((-C^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) + C^2*c^3
*f^2 - 3*C^2*c*d^2*f^2)/f^4)^(1/2)*(C*d^3 + C*c^2*d - c*f*(-((-C^4*d^2*f^4*
(3*c^2 - d^2)^2)^(1/2) + C^2*c^3*f^2 - 3*C^2*c*d^2*f^2)/f^4)^(1/2)*(c + d*t
an(e + f*x))^(1/2)))/f - (16*C^2*d^2*(c + d*tan(e + f*x))^(1/2)*(c^4 + d^4
- 6*c^2*d^2))/f^2)*(-((-C^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) + C^2*c^3*f^2 -
3*C^2*c*d^2*f^2)/f^4)^(1/2))/2)*(-((6*C^4*c^2*d^4*f^4 - C^4*d^6*f^4 - 9*C^4
*c^4*d^2*f^4)^(1/2) + C^2*c^3*f^2 - 3*C^2*c*d^2*f^2)/(4*f^4))^(1/2) - log((
16*C^3*c*d^3*(c^2 + d^2)^2)/f^3 - (((16*d^2*(((-C^4*d^2*f^4*(3*c^2 - d^2)^2
)^(1/2) - C^2*c^3*f^2 + 3*C^2*c*d^2*f^2)/f^4)^(1/2)*(C*d^3 + C*c^2*d - c*f*
(((-C^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) - C^2*c^3*f^2 + 3*C^2*c*d^2*f^2)/f^4
)^(1/2)*(c + d*tan(e + f*x))^(1/2)))/f - (16*C^2*d^2*(c + d*tan(e + f*x))^(
1/2)*(c^4 + d^4 - 6*c^2*d^2))/f^2)*(((-C^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) -
C^2*c^3*f^2 + 3*C^2*c*d^2*f^2)/f^4)^(1/2))/2)*(((6*C^4*c^2*d^4*f^4 - C^4*d
^6*f^4 - 9*C^4*c^4*d^2*f^4)^(1/2) - C^2*c^3*f^2 + 3*C^2*c*d^2*f^2)/(4*f^4))
^(1/2) + log((16*C^3*c*d^3*(c^2 + d^2)^2)/f^3 - (((16*d^2*(((-C^4*d^2*f^4*(
3*c^2 - d^2)^2)^(1/2) - C^2*c^3*f^2 + 3*C^2*c*d^2*f^2)/f^4)^(1/2)*(C*d^3 +
C*c^2*d + c*f*(((-C^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) - C^2*c^3*f^2 + 3*C^2*
c*d^2*f^2)/f^4)^(1/2)*(c + d*tan(e + f*x))^(1/2)))/f + (16*C^2*d^2*(c + d*t
an(e + f*x))^(1/2)*(c^4 + d^4 - 6*c^2*d^2))/f^2)*(((-C^4*d^2*f^4*(3*c^2 - d
^2)^2)^(1/2) - C^2*c^3*f^2 + 3*C^2*c*d^2*f^2)/f^4)^(1/2))/2)*((6*C^4*c^2*d^
4*f^4 - C^4*d^6*f^4 - 9*C^4*c^4*d^2*f^4)^(1/2)/(4*f^4) - (C^2*c^3)/(4*f^2)

```
+ (3*C^2*c*d^2)/(4*f^2))^(1/2) + log((16*C^3*c*d^3*(c^2 + d^2)^2)/f^3 - (((
16*d^2*(-((-C^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) + C^2*c^3*f^2 - 3*C^2*c*d^2*
f^2)/f^4)^(1/2)*(C*d^3 + C*c^2*d + c*f*(-((-C^4*d^2*f^4*(3*c^2 - d^2)^2)^(1
/2) + C^2*c^3*f^2 - 3*C^2*c*d^2*f^2)/f^4)^(1/2)*(c + d*tan(e + f*x))^(1/2))
)/f + (16*C^2*d^2*(c + d*tan(e + f*x))^(1/2)*(c^4 + d^4 - 6*c^2*d^2))/f^2)*
(-((-C^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) + C^2*c^3*f^2 - 3*C^2*c*d^2*f^2)/f^
4)^(1/2))/2)*((3*C^2*c*d^2)/(4*f^2) - (C^2*c^3)/(4*f^2) - (6*C^4*c^2*d^4*f^
4 - C^4*d^6*f^4 - 9*C^4*c^4*d^2*f^4)^(1/2)/(4*f^4))^(1/2) + (2*B*(c + d*tan
(e + f*x))^(3/2))/(3*f) + (2*A*d*(c + d*tan(e + f*x))^(1/2))/f + (2*B*c*(c
+ d*tan(e + f*x))^(1/2))/f + (2*C*(c + d*tan(e + f*x))^(5/2))/(5*d*f)
```

3.101 $\int \frac{(c+d\tan(e+fx))^{3/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx$

Optimal result 1315
Rubi [A] (verified) 1316
Mathematica [A] (verified) 1319
Maple [B] (verified) 1320
Fricas [F(-1)] 1320
Sympy [F] 1320
Maxima [F(-2)] 1321
Giac [F(-1)] 1321
Mupad [B] (verification not implemented) 1321

Optimal result

Integrand size = 47, antiderivative size = 271

$$\begin{aligned}\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{a+b\tan(e+fx)}\,dx = & -\frac{(iA+B-iC)(c-id)^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(a-ib)f} \\ & -\frac{(A+iB-C)(c+id)^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(ia-b)f} \\ & -\frac{2\left(Ab^2-a(bB-aC)\right)(bc-ad)^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{b^{5/2}\left(a^2+b^2\right)f} \\ & +\frac{2(bcC+bBd-aCd)\sqrt{c+d\tan(e+fx)}}{b^2f}+\frac{2C(c+d\tan(e+fx))^{3/2}}{3bf}\end{aligned}$$

```
[Out] -(I*A+B-I*C)*(c-I*d)^(3/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(a
-I*b)/f-(A+I*B-C)*(c+I*d)^(3/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2
))/(I*a-b)/f-2*(A*b^2-a*(B*b-C*a))*(-a*d+b*c)^(3/2)*arctanh(b^(1/2)*(c+d*ta
n(f*x+e))^(1/2)/(-a*d+b*c)^(1/2))/b^(5/2)/(a^2+b^2)/f+2*(B*b*d-C*a*d+C*b*c)
*(c+d*tan(f*x+e))^(1/2)/b^2/f+2/3*C*(c+d*tan(f*x+e))^(3/2)/b/f
```

Rubi [A] (verified)

Time = 2.04 (sec) , antiderivative size = 271, normalized size of antiderivative = 1.00, number of steps used = 13, number of rules used = 7, $\frac{\text{number of rules}}{\text{integrand size}} = 0.149$, Rules used = {3728, 3734, 3620, 3618, 65, 214, 3715}

$$\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{a+b\tan(e+fx)}\,dx = -\frac{2(bc-ad)^{3/2}\left(Ab^2-a(bB-aC)\right)\operatorname{arctanh}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{b^{5/2}f\left(a^2+b^2\right)} - \frac{(c-id)^{3/2}(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f(a-ib)} - \frac{(c+id)^{3/2}(A+iB-C)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f(-b+ia)} + \frac{2(-aCd+bBd+bcC)\sqrt{c+d\tan(e+fx)}}{b^2f} + \frac{2C(c+d\tan(e+fx))^{3/2}}{3bf}$$

```
[In] Int[((c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a
 + b*Tan[e + f*x]),x]
```

```
[Out] -(((I*A + B - I*C)*(c - I*d)^(3/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c
- I*d]])/((a - I*b)*f)) - ((A + I*B - C)*(c + I*d)^(3/2)*ArcTanh[Sqrt[c + d
*Tan[e + f*x]]/Sqrt[c + I*d]])/((I*a - b)*f) - (2*(A*b^2 - a*(b*B - a*C))*(
b*c - a*d)^(3/2)*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*d]
])/(b^(5/2)*(a^2 + b^2)*f) + (2*(b*c*C + b*B*d - a*C*d)*Sqrt[c + d*Tan[e +
f*x]])/(b^2*f) + (2*C*(c + d*Tan[e + f*x])^(3/2))/(3*b*f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
```

```
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3715

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_) + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :>
 Dist[A/f, Subst[Int[(a + b*x)^m*(c + d*x)^n, x], x, Tan[e + f*x]], x] /; F
reeQ[{a, b, c, d, e, f, A, C, m, n}, x] && EqQ[A, C]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3734

```
Int[(((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2))/((a_.) + (b_.)*tan[(e_.)
+ (f_.)*(x_)]), x_Symbol] :> Dist[1/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n
*Simp[b*B + a*(A - C) + (a*B - b*(A - C))*Tan[e + f*x], x], x], x] + Dist[(
A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n*((1 + Tan[e
+ f*x]^2)/(a + b*Tan[e + f*x])), x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C
, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&
 !GtQ[n, 0] &&  !LeQ[n, -1]
```

Rubi steps

$$\text{integral} = \frac{2C(c+d\tan(e+fx))^{3/2}}{3bf} + \frac{2\int \frac{\sqrt{c+d\tan(e+fx)}\left(\frac{3}{2}(Abc-aCd)+\frac{3}{2}b(Bc+(A-C)d)\tan(e+fx)+\frac{3}{2}(bcC+bBd-aCd)\tan^2(e+fx)\right)}{a+b\tan(e+fx)}\,dx}{3b}$$

$$= \frac{2(bcC+bBd-aCd)\sqrt{c+d\tan(e+fx)}}{b^2f} + \frac{2C(c+d\tan(e+fx))^{3/2}}{3bf} + \frac{4\int \frac{\frac{3}{4}\left(Ab^2c^2+ad(aCd-b(2cC+Bd))\right)+\frac{3}{4}b^2\left(2c(A-C)d+B\left(c^2-d^2\right)\right)\tan(e+fx)+\frac{3}{4}\left(b^2d(Bc+(A-C)d)+(bc-ad)(bcC+bBd-aCd)\right)}{(a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}}}{3b^2}$$

$$= \frac{2(bcC+bBd-aCd)\sqrt{c+d\tan(e+fx)}}{b^2f} + \frac{2C(c+d\tan(e+fx))^{3/2}}{3bf} + \frac{4\int \frac{-\frac{3}{4}b^2\left(a\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-b\left(2c(A-C)d+B\left(c^2-d^2\right)\right)\right)+\frac{3}{4}b^2\left(2aAcd-2acCd-Ab\left(c^2-d^2\right)+aB\left(c^2-d^2\right)+b(c^2C+2\right.}{\sqrt{c+d\tan(e+fx)}}}{3b^2\left(a^2+b^2\right)} + \frac{\left(\left(Ab^2-a(bB-aC)\right)(bc-ad)^2\right)\int \frac{1+\tan^2(e+fx)}{(a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}}\,dx}{b^2\left(a^2+b^2\right)}$$

$$= \frac{2(bcC+bBd-aCd)\sqrt{c+d\tan(e+fx)}}{b^2f} + \frac{2C(c+d\tan(e+fx))^{3/2}}{3bf} + \frac{\left((A-iB-C)(c-id)^2\right)\int \frac{1+i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(a-ib)} + \frac{\left((A+iB-C)(c+id)^2\right)\int \frac{1-i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(a+ib)} + \frac{\left(\left(Ab^2-a(bB-aC)\right)(bc-ad)^2\right)\text{Subst}\left(\int \frac{1}{(a+bx)\sqrt{c+dx}}\,dx, x, \tan(e+fx)\right)}{b^2\left(a^2+b^2\right)f}$$

$$= \frac{2(bcC+bBd-aCd)\sqrt{c+d\tan(e+fx)}}{b^2f} + \frac{2C(c+d\tan(e+fx))^{3/2}}{3bf} + \frac{\left((iA+B-iC)(c-id)^2\right)\text{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c-idx}}\,dx, x, i\tan(e+fx)\right)}{2(a-ib)f} - \frac{\left(i(A+iB-C)(c+id)^2\right)\text{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c+idx}}\,dx, x, -i\tan(e+fx)\right)}{2(a+ib)f} + \frac{\left(2\left(Ab^2-a(bB-aC)\right)(bc-ad)^2\right)\text{Subst}\left(\int \frac{1}{a-\frac{bc}{d}+\frac{bx^2}{d}}\,dx, x, \sqrt{c+d\tan(e+fx)}\right)}{b^2\left(a^2+b^2\right)df}$$

$$= -\frac{2(Ab^2 - a(bB - aC))\,(bc - ad)^{3/2}\text{arctanh}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{b^{5/2}\,(a^2+b^2)\,f} + \frac{2(bcC + bBd - aCd)\sqrt{c+d\tan(e+fx)}}{b^2 f} + \frac{2C(c+d\tan(e+fx))^{3/2}}{3bf} - \frac{((A-iB-C)(c-id)^2)\,\text{Subst}\left(\int \frac{1}{-1-\frac{ic}{d}+\frac{ix^2}{d}}\,dx, x, \sqrt{c+d\tan(e+fx)}\right)}{(a-ib)df} - \frac{((A+iB-C)(c+id)^2)\,\text{Subst}\left(\int \frac{1}{-1+\frac{ic}{d}-\frac{ix^2}{d}}\,dx, x, \sqrt{c+d\tan(e+fx)}\right)}{(a+ib)df}$$

$$= -\frac{(iA+B-iC)(c-id)^{3/2}\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(a-ib)f} - \frac{(A+iB-C)(c+id)^{3/2}\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(ia-b)f} - \frac{2(Ab^2 - a(bB - aC))\,(bc - ad)^{3/2}\text{arctanh}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{b^{5/2}\,(a^2+b^2)\,f} + \frac{2(bcC + bBd - aCd)\sqrt{c+d\tan(e+fx)}}{b^2 f} + \frac{2C(c+d\tan(e+fx))^{3/2}}{3bf}$$

Mathematica [A] (verified)

Time = 2.66 (sec) , antiderivative size = 266, normalized size of antiderivative = 0.98

$$\int \frac{(c+d\tan(e+fx))^{3/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx = \frac{3ib\left(-\left((a+ib)(A-iB-C)(c-id)^{3/2}\text{arctanh}\left(\right.\right.\right.}{}$$

```
[In] Integrate[((c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/(a + b*Tan[e + f*x]),x]
```

```
[Out] (((3*I)*b*(-((a + I*b)*(A - I*B - C)*(c - I*d)^(3/2)*ArcTanh[Sqrt[c + d*Tan
[e + f*x]]/Sqrt[c - I*d]]) + (a - I*b)*(A + I*B - C)*(c + I*d)^(3/2)*ArcTan
h[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]]))/(a^2 + b^2) - (6*(A*b^2 + a*(-(
b*B) + a*C))*(b*c - a*d)^(3/2)*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/S
qrt[b*c - a*d]])/(b^(3/2)*(a^2 + b^2)) + (6*(b*c*C + b*B*d - a*C*d)*Sqrt[c
+ d*Tan[e + f*x]])/b + 2*C*(c + d*Tan[e + f*x])^(3/2))/(3*b*f)
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 6054 vs. $2(234) = 468$.

Time = 0.15 (sec) , antiderivative size = 6055, normalized size of antiderivative = 22.34

method	result	size
derivativedivides	Expression too large to display	6055
default	Expression too large to display	6055

```
[In] int((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
,x,method=_RETURNVERBOSE)
```

```
[Out] result too large to display
```

Fricas [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{3/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e)),x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int \frac{(c+d\tan(e+fx))^{3/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx = \int \frac{(c+d\tan(e+fx))^{\frac{3}{2}}(A+B\tan(e$$

```
[In] integrate((c+d*tan(f*x+e))**(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e)),x)
```

```
[Out] Integral((c + d*tan(e + f*x))**(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)**
2)/(a + b*tan(e + f*x)), x)
```

Maxima [F(-2)]

Exception generated.

$$\int \frac{(c+d\tan(e+fx))^{3/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx = \text{Exception raised: ValueError}$$

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e)),x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 detail
```

Giac [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{3/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e)),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [B] (verification not implemented)

Time = 55.55 (sec) , antiderivative size = 106783, normalized size of antiderivative = 394.03

$$\int \frac{(c+d\tan(e+fx))^{3/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx = \text{Too large to display}$$

```
[In] int(((c + d*tan(e + f*x))^(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x)),x)
```

```
[Out] atan(((((((32*(4*B*a*b^8*d^12*f^4 - 4*B*b^9*c*d^11*f^4 + 8*B*a^3*b^6*d^12*f
^4 + 4*B*a^5*b^4*d^12*f^4 - 4*B*b^9*c^3*d^9*f^4 + 8*B*a*b^8*c^2*d^10*f^4 +
4*B*a*b^8*c^4*d^8*f^4 - 12*B*a^2*b^7*c*d^11*f^4 - 12*B*a^4*b^5*c*d^11*f^4 -
 4*B*a^6*b^3*c*d^11*f^4 - 12*B*a^2*b^7*c^3*d^9*f^4 + 16*B*a^3*b^6*c^2*d^10*
f^4 + 8*B*a^3*b^6*c^4*d^8*f^4 - 12*B*a^4*b^5*c^3*d^9*f^4 + 8*B*a^5*b^4*c^2*
d^10*f^4 + 4*B*a^5*b^4*c^4*d^8*f^4 - 4*B*a^6*b^3*c^3*d^9*f^4))/(b*f^5) - (3
2*(c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 1
6*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*
```

b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) - 4*B^2*a^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 + 8*B^2*a*b*d^3*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 - 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^10*d^10*f^4 + 16*a^2*b^8*d^10*f^4 - 16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*d^10*f^4 + 24*b^10*c^2*d^8*f^4 + 40*a^2*b^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^9*c*d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 24*a^5*b^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9*f^4))/(b*f^4))*(-(((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) - 4*B^2*a^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 + 8*B^2*a*b*d^3*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 - 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(4*B^2*a^3*b^5*d^13*f^2 + 2*B^2*a^5*b^3*d^13*f^2 + 28*B^2*b^8*c^3*d^10*f^2 - 10*B^2*b^8*c^5*d^8*f^2 - 14*B^2*a*b^7*d^13*f^2 + 16*B^2*a^7*b*d^13*f^2 - 8*B^2*a^8*c*d^12*f^2 + 22*B^2*b^8*c*d^12*f^2 + 20*B^2*a*b^7*c^2*d^11*f^2 + 50*B^2*a*b^7*c^4*d^9*f^2 - 28*B^2*a^2*b^6*c*d^12*f^2 - 2*B^2*a^4*b^4*c*d^12*f^2 - 56*B^2*a^6*b^2*c*d^12*f^2 + 32*B^2*a^7*b*c^2*d^11*f^2 + 8*B^2*a^2*b^6*c^3*d^10*f^2 + 12*B^2*a^2*b^6*c^5*d^8*f^2 - 24*B^2*a^3*b^5*c^2*d^11*f^2 - 12*B^2*a^3*b^5*c^4*d^9*f^2 - 4*B^2*a^4*b^4*c^3*d^10*f^2 - 10*B^2*a^4*b^4*c^5*d^8*f^2 + 52*B^2*a^5*b^3*c^2*d^11*f^2 + 34*B^2*a^5*b^3*c^4*d^9*f^2 - 48*B^2*a^6*b^2*c^3*d^10*f^2))/(b*f^4))*(-(((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) - 4*B^2*a^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 + 8*B^2*a*b*d^3*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 - 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(15*B^3*a^4*b^3*d^15*f^2 - B^3*a^2*b^5*d^15*f^2 - 4*B^3*a^7*c^3*d^12*f^2 + 2*B^3*b^7*c^2*d^13*f^2 + 4*B^3*b^7*c^4*d^11*f^2 + 2*B^3*b^7*c^6*d^9*f^2 - 12*B^3*a^6*b*d^15*f^2 - 4*B^3*a^7*c*d^14*f^2 - B^3*a*b^6*c*d^14*f^2 - 27*B^3*a*b^6*c^3*d^12*f^2 - 19*B^3*a*b^6*c^5*d^10*f^2 + 7*B^3*a*b^6*c^7*d^8*f^2 - 57*B^3*a^3*b^4*c*d^14*f^2 + 64*B^3*a^5*b^2*c*d^14*f^2 + 4*B^3*a^6*b*c^2*d^13*f^2 + 16*B^3*a^6*b*c^4*d^11*f^2 + 65*B^3*a^2*b^5*c^2*d^13*f^2 + 9*B^3*a^2*b^5*c^4*d^11*f^2 - 57*B^3*a^2*b^5*c^6*d^9*f^2 + 77*B^3*a^3*b^4*c^3*d^12*f^2 + 129*B^3*a^3*b^4*c^5*d^10*f^2 - 5*B^3*a^3*b^4*c^7*d^8*f^2 - 121*B^3*a^4*b^3*c^2*d^13*f^2 - 119*B^3*a^4*b^3*c^4*d^11*f^2 + 17*B^3*a^4*b^3*c^6*d^9*f^2 + 40*B^3*a^5*b^2*c^3*d^12*f^2 - 24*B^3*a^5*b^2*c^5*d^10*f^2))/(b*f^5))*(-(((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) - 4*B^2*a^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 + 8*B^2*a*b*d^3*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 - 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(B^4*b^6*d^16 - 2*B^4*a^6*d^16 + 12*B^4*a^6*c^2*d^14 - 2*B^4*a^6*c^4*d^12 + 4*B^4

*b^6*c^2*d^14 + 6*B^4*b^6*c^4*d^12 + 4*B^4*b^6*c^6*d^10 + B^4*b^6*c^8*d^8 -
2*B^4*a^2*b^4*c^4*d^12 + 12*B^4*a^2*b^4*c^6*d^10 - 2*B^4*a^2*b^4*c^8*d^8 +
8*B^4*a^3*b^3*c^3*d^13 - 48*B^4*a^3*b^3*c^5*d^11 + 8*B^4*a^3*b^3*c^7*d^9 -
12*B^4*a^4*b^2*c^2*d^14 + 72*B^4*a^4*b^2*c^4*d^12 - 12*B^4*a^4*b^2*c^6*d^1
0 + 8*B^4*a^5*b*c*d^15 - 48*B^4*a^5*b*c^3*d^13 + 8*B^4*a^5*b*c^5*d^11))/(b*
f^4))*(-(((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*
B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*
a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 +
3*B^4*c^4*d^2))^(1/2) - 4*B^2*a^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 + 8*B^2*a*b*
d^3*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 - 24*B^2*a*b*c^2*d*f^
2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*1i - (((((32*(4*B*a*b^8*
d^12*f^4 - 4*B*b^9*c*d^11*f^4 + 8*B*a^3*b^6*d^12*f^4 + 4*B*a^5*b^4*d^12*f^4
- 4*B*b^9*c^3*d^9*f^4 + 8*B*a*b^8*c^2*d^10*f^4 + 4*B*a*b^8*c^4*d^8*f^4 - 1
2*B*a^2*b^7*c*d^11*f^4 - 12*B*a^4*b^5*c*d^11*f^4 - 4*B*a^6*b^3*c*d^11*f^4 -
12*B*a^2*b^7*c^3*d^9*f^4 + 16*B*a^3*b^6*c^2*d^10*f^4 + 8*B*a^3*b^6*c^4*d^8
*f^4 - 12*B*a^4*b^5*c^3*d^9*f^4 + 8*B*a^5*b^4*c^2*d^10*f^4 + 4*B*a^5*b^4*c^
4*d^8*f^4 - 4*B*a^6*b^3*c^3*d^9*f^4))/(b*f^5) + (32*(c + d*tan(e + f*x))^(1
/2)*(-(((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^
2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^
4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3
*B^4*c^4*d^2))^(1/2) - 4*B^2*a^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 + 8*B^2*a*b*d^
3*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 - 24*B^2*a*b*c^2*d*f^2)
/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^10*d^10*f^4 + 16*a^2
*b^8*d^10*f^4 - 16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*d^10*f^4 + 24*b^10*c^2*d^8
*f^4 + 40*a^2*b^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^2*d^8*f
^4 + 8*a*b^9*c*d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 24*a^5*b^5*c*d^9*f^4 + 8*a^
7*b^3*c*d^9*f^4))/(b*f^4))*(-(((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*
B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*
c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*
d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) - 4*B^2*a^2*c^3*f^2 + 4*B^2*b^2
*c^3*f^2 + 8*B^2*a*b*d^3*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2
- 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (
32*(c + d*tan(e + f*x))^(1/2)*(4*B^2*a^3*b^5*d^13*f^2 + 2*B^2*a^5*b^3*d^13*
f^2 + 28*B^2*b^8*c^3*d^10*f^2 - 10*B^2*b^8*c^5*d^8*f^2 - 14*B^2*a*b^7*d^13*
f^2 + 16*B^2*a^7*b*d^13*f^2 - 8*B^2*a^8*c*d^12*f^2 + 22*B^2*b^8*c*d^12*f^2
+ 20*B^2*a*b^7*c^2*d^11*f^2 + 50*B^2*a*b^7*c^4*d^9*f^2 - 28*B^2*a^2*b^6*c*d
^12*f^2 - 2*B^2*a^4*b^4*c*d^12*f^2 - 56*B^2*a^6*b^2*c*d^12*f^2 + 32*B^2*a^7
*b*c^2*d^11*f^2 + 8*B^2*a^2*b^6*c^3*d^10*f^2 + 12*B^2*a^2*b^6*c^5*d^8*f^2 -
24*B^2*a^3*b^5*c^2*d^11*f^2 - 12*B^2*a^3*b^5*c^4*d^9*f^2 - 4*B^2*a^4*b^4*c
^3*d^10*f^2 - 10*B^2*a^4*b^4*c^5*d^8*f^2 + 52*B^2*a^5*b^3*c^2*d^11*f^2 + 34
*B^2*a^5*b^3*c^4*d^9*f^2 - 48*B^2*a^6*b^2*c^3*d^10*f^2))/(b*f^4))*(-(((8*B^
2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f
^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^
4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2)
)^(1/2) - 4*B^2*a^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 + 8*B^2*a*b*d^3*f^2 + 12*B^

2*a^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 - 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(15*B^3*a^4*b^3*d^15*f^2 - B^3*a^2*b^5*d^15*f^2 - 4*B^3*a^7*c^3*d^12*f^2 + 2*B^3*b^7*c^2*d^13*f^2 + 4*B^3*b^7*c^4*d^11*f^2 + 2*B^3*b^7*c^6*d^9*f^2 - 12*B^3*a^6*b*d^15*f^2 - 4*B^3*a^7*c*d^14*f^2 - B^3*a*b^6*c*d^14*f^2 - 27*B^3*a*b^6*c^3*d^12*f^2 - 19*B^3*a*b^6*c^5*d^10*f^2 + 7*B^3*a*b^6*c^7*d^8*f^2 - 57*B^3*a^3*b^4*c*d^14*f^2 + 64*B^3*a^5*b^2*c*d^14*f^2 + 4*B^3*a^6*b*c^2*d^13*f^2 + 16*B^3*a^6*b*c^4*d^11*f^2 + 65*B^3*a^2*b^5*c^2*d^13*f^2 + 9*B^3*a^2*b^5*c^4*d^11*f^2 - 57*B^3*a^2*b^5*c^6*d^9*f^2 + 77*B^3*a^3*b^4*c^3*d^12*f^2 + 129*B^3*a^3*b^4*c^5*d^10*f^2 - 5*B^3*a^3*b^4*c^7*d^8*f^2 - 121*B^3*a^4*b^3*c^2*d^13*f^2 - 119*B^3*a^4*b^3*c^4*d^11*f^2 + 17*B^3*a^4*b^3*c^6*d^9*f^2 + 40*B^3*a^5*b^2*c^3*d^12*f^2 - 24*B^3*a^5*b^2*c^5*d^10*f^2))/(b*f^5))*(-(((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) - 4*B^2*a^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 + 8*B^2*a*b*d^3*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 - 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(B^4*b^6*d^16 - 2*B^4*a^6*d^16 + 12*B^4*a^6*c^2*d^14 - 2*B^4*a^6*c^4*d^12 + 4*B^4*b^6*c^2*d^14 + 6*B^4*b^6*c^4*d^12 + 4*B^4*b^6*c^6*d^10 + B^4*b^6*c^8*d^8 - 2*B^4*a^2*b^4*c^4*d^12 + 12*B^4*a^2*b^4*c^6*d^10 - 2*B^4*a^2*b^4*c^8*d^8 + 8*B^4*a^3*b^3*c^3*d^13 - 48*B^4*a^3*b^3*c^5*d^11 + 8*B^4*a^3*b^3*c^7*d^9 - 12*B^4*a^4*b^2*c^2*d^14 + 72*B^4*a^4*b^2*c^4*d^12 - 12*B^4*a^4*b^2*c^6*d^10 + 8*B^4*a^5*b*c*d^15 - 48*B^4*a^5*b*c^3*d^13 + 8*B^4*a^5*b*c^5*d^11))/(b*f^4))*(-(((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) - 4*B^2*a^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 + 8*B^2*a*b*d^3*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 - 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*1i)/((((((32*(4*B*a*b^8*d^12*f^4 - 4*B*b^9*c*d^11*f^4 + 8*B*a^3*b^6*d^12*f^4 + 4*B*a^5*b^4*d^12*f^4 - 4*B*b^9*c^3*d^9*f^4 + 8*B*a*b^8*c^2*d^10*f^4 + 4*B*a*b^8*c^4*d^8*f^4 - 12*B*a^2*b^7*c*d^11*f^4 - 12*B*a^4*b^5*c*d^11*f^4 - 4*B*a^6*b^3*c*d^11*f^4 - 12*B*a^2*b^7*c^3*d^9*f^4 + 16*B*a^3*b^6*c^2*d^10*f^4 + 8*B*a^3*b^6*c^4*d^8*f^4 - 12*B*a^4*b^5*c^3*d^9*f^4 + 8*B*a^5*b^4*c^2*d^10*f^4 + 4*B*a^5*b^4*c^4*d^8*f^4 - 4*B*a^6*b^3*c^3*d^9*f^4))/(b*f^5) - (32*(c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) - 4*B^2*a^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 + 8*B^2*a*b*d^3*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 - 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^10*d^10*f^4 + 16*a^2*b^8*d^10*f^4 - 16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*d^10*f^4 + 24*b^10*c^2*d^8*f^4 + 40*a^2*b^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^9*c*d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 24*a^5*b^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9*f^4))/(b*f^4)

)*(-(((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) - 4*B^2*a^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 + 8*B^2*a*b*d^3*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 - 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(4*B^2*a^3*b^5*d^13*f^2 + 2*B^2*a^5*b^3*d^13*f^2 + 28*B^2*b^8*c^3*d^10*f^2 - 10*B^2*b^8*c^5*d^8*f^2 - 14*B^2*a*b^7*d^13*f^2 + 16*B^2*a^7*b*d^13*f^2 - 8*B^2*a^8*c*d^12*f^2 + 22*B^2*b^8*c*d^12*f^2 + 20*B^2*a*b^7*c^2*d^11*f^2 + 50*B^2*a*b^7*c^4*d^9*f^2 - 28*B^2*a^2*b^6*c*d^12*f^2 - 2*B^2*a^4*b^4*c*d^12*f^2 - 56*B^2*a^6*b^2*c*d^12*f^2 + 32*B^2*a^7*b*c^2*d^11*f^2 + 8*B^2*a^2*b^6*c^3*d^10*f^2 + 12*B^2*a^2*b^6*c^5*d^8*f^2 - 24*B^2*a^3*b^5*c^2*d^11*f^2 - 12*B^2*a^3*b^5*c^4*d^9*f^2 - 4*B^2*a^4*b^4*c^3*d^10*f^2 - 10*B^2*a^4*b^4*c^5*d^8*f^2 + 52*B^2*a^5*b^3*c^2*d^11*f^2 + 34*B^2*a^5*b^3*c^4*d^9*f^2 - 48*B^2*a^6*b^2*c^3*d^10*f^2))/(b*f^4))*(-(((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) - 4*B^2*a^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 + 8*B^2*a*b*d^3*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 - 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(15*B^3*a^4*b^3*d^15*f^2 - B^3*a^2*b^5*d^15*f^2 - 4*B^3*a^7*c^3*d^12*f^2 + 2*B^3*b^7*c^2*d^13*f^2 + 4*B^3*b^7*c^4*d^11*f^2 + 2*B^3*b^7*c^6*d^9*f^2 - 12*B^3*a^6*b*d^15*f^2 - 4*B^3*a^7*c*d^14*f^2 - B^3*a*b^6*c*d^14*f^2 - 27*B^3*a*b^6*c^3*d^12*f^2 - 19*B^3*a*b^6*c^5*d^10*f^2 + 7*B^3*a*b^6*c^7*d^8*f^2 - 57*B^3*a^3*b^4*c*d^14*f^2 + 64*B^3*a^5*b^2*c*d^14*f^2 + 4*B^3*a^6*b*c^2*d^13*f^2 + 16*B^3*a^6*b*c^4*d^11*f^2 + 65*B^3*a^2*b^5*c^2*d^13*f^2 + 9*B^3*a^2*b^5*c^4*d^11*f^2 - 57*B^3*a^2*b^5*c^6*d^9*f^2 + 77*B^3*a^3*b^4*c^3*d^12*f^2 + 129*B^3*a^3*b^4*c^5*d^10*f^2 - 5*B^3*a^3*b^4*c^7*d^8*f^2 - 121*B^3*a^4*b^3*c^2*d^13*f^2 - 119*B^3*a^4*b^3*c^4*d^11*f^2 + 17*B^3*a^4*b^3*c^6*d^9*f^2 + 40*B^3*a^5*b^2*c^3*d^12*f^2 - 24*B^3*a^5*b^2*c^5*d^10*f^2))/(b*f^5))*(-(((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) - 4*B^2*a^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 + 8*B^2*a*b*d^3*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 - 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(B^4*b^6*d^16 - 2*B^4*a^6*d^16 + 12*B^4*a^6*c^2*d^14 - 2*B^4*a^6*c^4*d^12 + 4*B^4*b^6*c^2*d^14 + 6*B^4*b^6*c^4*d^12 + 4*B^4*b^6*c^6*d^10 + B^4*b^6*c^8*d^8 - 2*B^4*a^2*b^4*c^4*d^12 + 12*B^4*a^2*b^4*c^6*d^10 - 2*B^4*a^2*b^4*c^8*d^8 + 8*B^4*a^3*b^3*c^3*d^13 - 48*B^4*a^3*b^3*c^5*d^11 + 8*B^4*a^3*b^3*c^7*d^9 - 12*B^4*a^4*b^2*c^2*d^14 + 72*B^4*a^4*b^2*c^4*d^12 - 12*B^4*a^4*b^2*c^6*d^10 + 8*B^4*a^5*b*c*d^15 - 48*B^4*a^5*b*c^3*d^13 + 8*B^4*a^5*b*c^5*d^11))/(b*f^4))*(-(((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 +

B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) - 4*B^2*a^2*c^3*f^2 + 4*B^2
*b^2*c^3*f^2 + 8*B^2*a*b*d^3*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*
f^2 - 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)
+ (((((32*(4*B*a*b^8*d^12*f^4 - 4*B*b^9*c*d^11*f^4 + 8*B*a^3*b^6*d^12*f^4
+ 4*B*a^5*b^4*d^12*f^4 - 4*B*b^9*c^3*d^9*f^4 + 8*B*a*b^8*c^2*d^10*f^4 + 4*B
*a*b^8*c^4*d^8*f^4 - 12*B*a^2*b^7*c*d^11*f^4 - 12*B*a^4*b^5*c*d^11*f^4 - 4*
B*a^6*b^3*c*d^11*f^4 - 12*B*a^2*b^7*c^3*d^9*f^4 + 16*B*a^3*b^6*c^2*d^10*f^4
+ 8*B*a^3*b^6*c^4*d^8*f^4 - 12*B*a^4*b^5*c^3*d^9*f^4 + 8*B*a^5*b^4*c^2*d^1
0*f^4 + 4*B*a^5*b^4*c^4*d^8*f^4 - 4*B*a^6*b^3*c^3*d^9*f^4))/(b*f^5) + (32*(
c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B
^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c
^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d
^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) - 4*B^2*a^2*c^3*f^2 + 4*B^2*b^2*
c^3*f^2 + 8*B^2*a*b*d^3*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 -
24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*
b^10*d^10*f^4 + 16*a^2*b^8*d^10*f^4 - 16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*d^10
*f^4 + 24*b^10*c^2*d^8*f^4 + 40*a^2*b^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8*f^4
- 8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^9*c*d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 24*a^
5*b^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9*f^4))/(b*f^4))*(-(((8*B^2*a^2*c^3*f^2 - 8
*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c
*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^
2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) - 4*B^2*a
^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 + 8*B^2*a*b*d^3*f^2 + 12*B^2*a^2*c*d^2*f^2 -
12*B^2*b^2*c*d^2*f^2 - 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^
2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(4*B^2*a^3*b^5*d^13*f^2
+ 2*B^2*a^5*b^3*d^13*f^2 + 28*B^2*b^8*c^3*d^10*f^2 - 10*B^2*b^8*c^5*d^8*f^
2 - 14*B^2*a*b^7*d^13*f^2 + 16*B^2*a^7*b*d^13*f^2 - 8*B^2*a^8*c*d^12*f^2 +
22*B^2*b^8*c*d^12*f^2 + 20*B^2*a*b^7*c^2*d^11*f^2 + 50*B^2*a*b^7*c^4*d^9*f^
2 - 28*B^2*a^2*b^6*c*d^12*f^2 - 2*B^2*a^4*b^4*c*d^12*f^2 - 56*B^2*a^6*b^2*c
*d^12*f^2 + 32*B^2*a^7*b*c^2*d^11*f^2 + 8*B^2*a^2*b^6*c^3*d^10*f^2 + 12*B^2
*a^2*b^6*c^5*d^8*f^2 - 24*B^2*a^3*b^5*c^2*d^11*f^2 - 12*B^2*a^3*b^5*c^4*d^9
*f^2 - 4*B^2*a^4*b^4*c^3*d^10*f^2 - 10*B^2*a^4*b^4*c^5*d^8*f^2 + 52*B^2*a^5
*b^3*c^2*d^11*f^2 + 34*B^2*a^5*b^3*c^4*d^9*f^2 - 48*B^2*a^6*b^2*c^3*d^10*f^
2))/(b*f^4))*(-(((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^
2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4
- (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^
2*d^4 + 3*B^4*c^4*d^2))^(1/2) - 4*B^2*a^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 + 8*B
^2*a*b*d^3*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 - 24*B^2*a*b*c
^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(15*B^3*a^4
*b^3*d^15*f^2 - B^3*a^2*b^5*d^15*f^2 - 4*B^3*a^7*c^3*d^12*f^2 + 2*B^3*b^7*c
^2*d^13*f^2 + 4*B^3*b^7*c^4*d^11*f^2 + 2*B^3*b^7*c^6*d^9*f^2 - 12*B^3*a^6*b
*d^15*f^2 - 4*B^3*a^7*c*d^14*f^2 - B^3*a*b^6*c*d^14*f^2 - 27*B^3*a*b^6*c^3*
d^12*f^2 - 19*B^3*a*b^6*c^5*d^10*f^2 + 7*B^3*a*b^6*c^7*d^8*f^2 - 57*B^3*a^3
*b^4*c*d^14*f^2 + 64*B^3*a^5*b^2*c*d^14*f^2 + 4*B^3*a^6*b*c^2*d^13*f^2 + 16
*B^3*a^6*b*c^4*d^11*f^2 + 65*B^3*a^2*b^5*c^2*d^13*f^2 + 9*B^3*a^2*b^5*c^4*d

^11*f^2 - 57*B^3*a^2*b^5*c^6*d^9*f^2 + 77*B^3*a^3*b^4*c^3*d^12*f^2 + 129*B^
3*a^3*b^4*c^5*d^10*f^2 - 5*B^3*a^3*b^4*c^7*d^8*f^2 - 121*B^3*a^4*b^3*c^2*d^
13*f^2 - 119*B^3*a^4*b^3*c^4*d^11*f^2 + 17*B^3*a^4*b^3*c^6*d^9*f^2 + 40*B^3
*a^5*b^2*c^3*d^12*f^2 - 24*B^3*a^5*b^2*c^5*d^10*f^2))/(b*f^5))*(-(((8*B^2*a
^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2
+ 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f
^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(
1/2) - 4*B^2*a^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 + 8*B^2*a*b*d^3*f^2 + 12*B^2*a
^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 - 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 +
b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(B^4*b^6*
d^16 - 2*B^4*a^6*d^16 + 12*B^4*a^6*c^2*d^14 - 2*B^4*a^6*c^4*d^12 + 4*B^4*b^
6*c^2*d^14 + 6*B^4*b^6*c^4*d^12 + 4*B^4*b^6*c^6*d^10 + B^4*b^6*c^8*d^8 - 2*
B^4*a^2*b^4*c^4*d^12 + 12*B^4*a^2*b^4*c^6*d^10 - 2*B^4*a^2*b^4*c^8*d^8 + 8*
B^4*a^3*b^3*c^3*d^13 - 48*B^4*a^3*b^3*c^5*d^11 + 8*B^4*a^3*b^3*c^7*d^9 - 12
*B^4*a^4*b^2*c^2*d^14 + 72*B^4*a^4*b^2*c^4*d^12 - 12*B^4*a^4*b^2*c^6*d^10 +
8*B^4*a^5*b*c*d^15 - 48*B^4*a^5*b*c^3*d^13 + 8*B^4*a^5*b*c^5*d^11))/(b*f^4
))*(-(((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2
*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4
*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*
B^4*c^4*d^2))^(1/2) - 4*B^2*a^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 + 8*B^2*a*b*d^3
*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 - 24*B^2*a*b*c^2*d*f^2)/
(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (64*(B^5*a^3*b^2*d^18 - B
^5*a^5*d^18 - B^5*a^5*c^2*d^16 + B^5*a^5*c^4*d^14 + B^5*a^5*c^6*d^12 - 8*B^
5*a^2*b^3*c^3*d^15 - 14*B^5*a^2*b^3*c^5*d^13 - 12*B^5*a^2*b^3*c^7*d^11 - 4*
B^5*a^2*b^3*c^9*d^9 + 3*B^5*a^3*b^2*c^2*d^16 + 9*B^5*a^3*b^2*c^4*d^14 + 13*
B^5*a^3*b^2*c^6*d^12 + 6*B^5*a^3*b^2*c^8*d^10 + 2*B^5*a^4*b*c*d^17 + B^5*a*
b^4*c^2*d^16 + 4*B^5*a*b^4*c^4*d^14 + 6*B^5*a*b^4*c^6*d^12 + 4*B^5*a*b^4*c^
8*d^10 + B^5*a*b^4*c^10*d^8 - 2*B^5*a^2*b^3*c*d^17 - 6*B^5*a^4*b*c^5*d^13 -
4*B^5*a^4*b*c^7*d^11))/(b*f^5)))*(-(((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^
2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B
^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6
+ B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) - 4*B^2*a^2*c^3*f^2 + 4*
B^2*b^2*c^3*f^2 + 8*B^2*a*b*d^3*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*B^2*b^2*c*d
^2*f^2 - 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1
/2)*2i - atan((((((32*(4*A*a^2*b^6*d^12*f^4 + 8*A*a^4*b^4*d^12*f^4 + 4*A*a
^6*b^2*d^12*f^4 + 12*A*b^8*c^2*d^10*f^4 + 12*A*b^8*c^4*d^8*f^4 - 16*A*a*b^7
*c^3*d^9*f^4 - 32*A*a^3*b^5*c*d^11*f^4 - 16*A*a^5*b^3*c*d^11*f^4 + 28*A*a^2
*b^6*c^2*d^10*f^4 + 24*A*a^2*b^6*c^4*d^8*f^4 - 32*A*a^3*b^5*c^3*d^9*f^4 + 2
0*A*a^4*b^4*c^2*d^10*f^4 + 12*A*a^4*b^4*c^4*d^8*f^4 - 16*A*a^5*b^3*c^3*d^9*
f^4 + 4*A*a^6*b^2*c^2*d^10*f^4 - 16*A*a*b^7*c*d^11*f^4))/f^5 - (32*(c + d*t
an(e + f*x))^(1/2)*((((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d
^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2
)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A
^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*A^2*b^2*c^3*f^2
+ 8*A^2*a*b*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*A^2*b^2*c*d^2*f^2 - 24*A^2*

a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/f^4)*((((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*A^2*b^2*c^3*f^2 + 8*A^2*a*b*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*A^2*b^2*c*d^2*f^2 - 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(4*A^2*a^3*b^4*d^13*f^2 - 14*A^2*a^5*b^2*d^13*f^2 + 28*A^2*b^7*c^3*d^10*f^2 - 18*A^2*b^7*c^5*d^8*f^2 - 14*A^2*a*b^6*d^13*f^2 + 22*A^2*b^7*c*d^12*f^2 + 8*A^2*a^6*b*c*d^12*f^2 + 20*A^2*a*b^6*c^2*d^11*f^2 + 66*A^2*a*b^6*c^4*d^9*f^2 - 28*A^2*a^2*b^5*c*d^12*f^2 + 54*A^2*a^4*b^3*c*d^12*f^2 + 24*A^2*a^2*b^5*c^3*d^10*f^2 + 12*A^2*a^2*b^5*c^5*d^8*f^2 - 88*A^2*a^3*b^4*c^2*d^11*f^2 - 28*A^2*a^3*b^4*c^4*d^9*f^2 + 60*A^2*a^4*b^3*c^3*d^10*f^2 - 2*A^2*a^4*b^3*c^5*d^8*f^2 - 44*A^2*a^5*b^2*c^2*d^11*f^2 + 2*A^2*a^5*b^2*c^4*d^9*f^2))/f^4)*((((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*A^2*b^2*c^3*f^2 + 8*A^2*a*b*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*A^2*b^2*c*d^2*f^2 - 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(23*A^3*b^6*c^3*d^12*f^2 - 15*A^3*a^3*b^3*d^15*f^2 + 21*A^3*b^6*c^5*d^10*f^2 - 3*A^3*b^6*c^7*d^8*f^2 + A^3*a*b^5*d^15*f^2 + 4*A^3*a^5*b*d^15*f^2 - A^3*b^6*c*d^14*f^2 - 61*A^3*a*b^5*c^2*d^13*f^2 - 25*A^3*a*b^5*c^4*d^11*f^2 + 37*A^3*a*b^5*c^6*d^9*f^2 + 53*A^3*a^2*b^4*c*d^14*f^2 - 30*A^3*a^4*b^2*c*d^14*f^2 + 4*A^3*a^5*b*c^2*d^13*f^2 - 29*A^3*a^2*b^4*c^3*d^12*f^2 - 81*A^3*a^2*b^4*c^5*d^10*f^2 + A^3*a^2*b^4*c^7*d^8*f^2 + 59*A^3*a^3*b^3*c^2*d^13*f^2 + 75*A^3*a^3*b^3*c^4*d^11*f^2 + A^3*a^3*b^3*c^6*d^9*f^2 - 32*A^3*a^4*b^2*c^3*d^12*f^2 - 2*A^3*a^4*b^2*c^5*d^10*f^2))/f^5)*((((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*A^2*b^2*c^3*f^2 + 8*A^2*a*b*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*A^2*b^2*c*d^2*f^2 - 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*d^16 + 4*A^4*b^5*c^2*d^14 + 8*A^4*b^5*c^4*d^12 - 8*A^4*b^5*c^6*d^10 + 3*A^4*b^5*c^8*d^8 + 2*A^4*a^4*b*d^16 + 12*A^4*a^2*b^3*c^2*d^14 - 72*A^4*a^2*b^3*c^4*d^12 + 12*A^4*a^2*b^3*c^6*d^10 + 48*A^4*a^3*b^2*c^3*d^13 - 8*A^4*a^3*b^2*c^5*d^11 - 8*A^4*a*b^4*c^3*d^13 + 48*A^4*a*b^4*c^5*d^11 - 8*A^4*a*b^4*c^7*d^9 - 8*A^4*a^3*b^2*c*d^15 - 12*A^4*a^4*b*c^2*d^14 + 2*A^4*a^4*b*c^4*d^12))/f^4)*((((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4

*c^4*d^2))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*A^2*b^2*c^3*f^2 + 8*A^2*a*b*d^3*f^
2 + 12*A^2*a^2*c*d^2*f^2 - 12*A^2*b^2*c*d^2*f^2 - 24*A^2*a*b*c^2*d*f^2)/(16
*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*1i - (((((32*(4*A*a^2*b^6*d^12
*f^4 + 8*A*a^4*b^4*d^12*f^4 + 4*A*a^6*b^2*d^12*f^4 + 12*A*b^8*c^2*d^10*f^4
+ 12*A*b^8*c^4*d^8*f^4 - 16*A*a*b^7*c^3*d^9*f^4 - 32*A*a^3*b^5*c*d^11*f^4 -
16*A*a^5*b^3*c*d^11*f^4 + 28*A*a^2*b^6*c^2*d^10*f^4 + 24*A*a^2*b^6*c^4*d^8
*f^4 - 32*A*a^3*b^5*c^3*d^9*f^4 + 20*A*a^4*b^4*c^2*d^10*f^4 + 12*A*a^4*b^4*
c^4*d^8*f^4 - 16*A*a^5*b^3*c^3*d^9*f^4 + 4*A*a^6*b^2*c^2*d^10*f^4 - 16*A*a*
b^7*c*d^11*f^4))/f^5 + (32*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^2*c^3*f^2
- 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b
^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^
2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) - 4*A
^2*a^2*c^3*f^2 + 4*A^2*b^2*c^3*f^2 + 8*A^2*a*b*d^3*f^2 + 12*A^2*a^2*c*d^2*f
^2 - 12*A^2*b^2*c*d^2*f^2 - 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 +
2*a^2*b^2*f^4)))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*
d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^
4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*
a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/f^4)*((((8
*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^
2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16
*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d
^2))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*A^2*b^2*c^3*f^2 + 8*A^2*a*b*d^3*f^2 + 12
*A^2*a^2*c*d^2*f^2 - 12*A^2*b^2*c*d^2*f^2 - 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*
f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(4*
A^2*a^3*b^4*d^13*f^2 - 14*A^2*a^5*b^2*d^13*f^2 + 28*A^2*b^7*c^3*d^10*f^2 -
18*A^2*b^7*c^5*d^8*f^2 - 14*A^2*a*b^6*d^13*f^2 + 22*A^2*b^7*c*d^12*f^2 + 8*
A^2*a^6*b*c*d^12*f^2 + 20*A^2*a*b^6*c^2*d^11*f^2 + 66*A^2*a*b^6*c^4*d^9*f^2
- 28*A^2*a^2*b^5*c*d^12*f^2 + 54*A^2*a^4*b^3*c*d^12*f^2 + 24*A^2*a^2*b^5*c
^3*d^10*f^2 + 12*A^2*a^2*b^5*c^5*d^8*f^2 - 88*A^2*a^3*b^4*c^2*d^11*f^2 - 28
*A^2*a^3*b^4*c^4*d^9*f^2 + 60*A^2*a^4*b^3*c^3*d^10*f^2 - 2*A^2*a^4*b^3*c^5*
d^8*f^2 - 44*A^2*a^5*b^2*c^2*d^11*f^2 + 2*A^2*a^5*b^2*c^4*d^9*f^2))/f^4)*((
((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c
*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 +
16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^
4*d^2))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*A^2*b^2*c^3*f^2 + 8*A^2*a*b*d^3*f^2 +
12*A^2*a^2*c*d^2*f^2 - 12*A^2*b^2*c*d^2*f^2 - 24*A^2*a*b*c^2*d*f^2)/(16*(a
^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(23*A^3*b^6*c^3*d^12*f^2 -
15*A^3*a^3*b^3*d^15*f^2 + 21*A^3*b^6*c^5*d^10*f^2 - 3*A^3*b^6*c^7*d^8*f^2 +
A^3*a*b^5*d^15*f^2 + 4*A^3*a^5*b*d^15*f^2 - A^3*b^6*c*d^14*f^2 - 61*A^3*a*
b^5*c^2*d^13*f^2 - 25*A^3*a*b^5*c^4*d^11*f^2 + 37*A^3*a*b^5*c^6*d^9*f^2 + 5
3*A^3*a^2*b^4*c*d^14*f^2 - 30*A^3*a^4*b^2*c*d^14*f^2 + 4*A^3*a^5*b*c^2*d^13
*f^2 - 29*A^3*a^2*b^4*c^3*d^12*f^2 - 81*A^3*a^2*b^4*c^5*d^10*f^2 + A^3*a^2*
b^4*c^7*d^8*f^2 + 59*A^3*a^3*b^3*c^2*d^13*f^2 + 75*A^3*a^3*b^3*c^4*d^11*f^2
+ A^3*a^3*b^3*c^6*d^9*f^2 - 32*A^3*a^4*b^2*c^3*d^12*f^2 - 2*A^3*a^4*b^2*c^
5*d^10*f^2))/f^5)*((((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^

3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)
^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^
4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*A^2*b^2*c^3*f^2 +
8*A^2*a*b*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*A^2*b^2*c*d^2*f^2 - 24*A^2*a
*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*
tan(e + f*x))^(1/2)*(A^4*b^5*d^16 + 4*A^4*b^5*c^2*d^14 + 8*A^4*b^5*c^4*d^12
- 8*A^4*b^5*c^6*d^10 + 3*A^4*b^5*c^8*d^8 + 2*A^4*a^4*b*d^16 + 12*A^4*a^2*b
^3*c^2*d^14 - 72*A^4*a^2*b^3*c^4*d^12 + 12*A^4*a^2*b^3*c^6*d^10 + 48*A^4*a^
3*b^2*c^3*d^13 - 8*A^4*a^3*b^2*c^5*d^11 - 8*A^4*a*b^4*c^3*d^13 + 48*A^4*a*b
^4*c^5*d^11 - 8*A^4*a*b^4*c^7*d^9 - 8*A^4*a^3*b^2*c*d^15 - 12*A^4*a^4*b*c^2
*d^14 + 2*A^4*a^4*b*c^4*d^12))/f^4)*((((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f
^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*
A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^
6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4
*A^2*b^2*c^3*f^2 + 8*A^2*a*b*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*A^2*b^2*c*
d^2*f^2 - 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(
1/2)*1i)/((((((32*(4*A*a^2*b^6*d^12*f^4 + 8*A*a^4*b^4*d^12*f^4 + 4*A*a^6*b^
2*d^12*f^4 + 12*A*b^8*c^2*d^10*f^4 + 12*A*b^8*c^4*d^8*f^4 - 16*A*a*b^7*c^3*
d^9*f^4 - 32*A*a^3*b^5*c*d^11*f^4 - 16*A*a^5*b^3*c*d^11*f^4 + 28*A*a^2*b^6*
c^2*d^10*f^4 + 24*A*a^2*b^6*c^4*d^8*f^4 - 32*A*a^3*b^5*c^3*d^9*f^4 + 20*A*a
^4*b^4*c^2*d^10*f^4 + 12*A*a^4*b^4*c^4*d^8*f^4 - 16*A*a^5*b^3*c^3*d^9*f^4 +
4*A*a^6*b^2*c^2*d^10*f^4 - 16*A*a*b^7*c*d^11*f^4))/f^5 - (32*(c + d*tan(e
+ f*x))^(1/2)*((((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^
2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4
- (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^
2*d^4 + 3*A^4*c^4*d^2))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*A^2*b^2*c^3*f^2 + 8*A
^2*a*b*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*A^2*b^2*c*d^2*f^2 - 24*A^2*a*b*c
^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^9*d^10*f^4
+ 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*
c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^
2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4
+ 8*a^7*b^2*c*d^9*f^4))/f^4)*((((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 1
6*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*
b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^
4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*A^2*b
^2*c^3*f^2 + 8*A^2*a*b*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*A^2*b^2*c*d^2*f^
2 - 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) -
(32*(c + d*tan(e + f*x))^(1/2)*(4*A^2*a^3*b^4*d^13*f^2 - 14*A^2*a^5*b^2*d^
13*f^2 + 28*A^2*b^7*c^3*d^10*f^2 - 18*A^2*b^7*c^5*d^8*f^2 - 14*A^2*a*b^6*d^
13*f^2 + 22*A^2*b^7*c*d^12*f^2 + 8*A^2*a^6*b*c*d^12*f^2 + 20*A^2*a*b^6*c^2*
d^11*f^2 + 66*A^2*a*b^6*c^4*d^9*f^2 - 28*A^2*a^2*b^5*c*d^12*f^2 + 54*A^2*a^
4*b^3*c*d^12*f^2 + 24*A^2*a^2*b^5*c^3*d^10*f^2 + 12*A^2*a^2*b^5*c^5*d^8*f^2
- 88*A^2*a^3*b^4*c^2*d^11*f^2 - 28*A^2*a^3*b^4*c^4*d^9*f^2 + 60*A^2*a^4*b^
3*c^3*d^10*f^2 - 2*A^2*a^4*b^3*c^5*d^8*f^2 - 44*A^2*a^5*b^2*c^2*d^11*f^2 +
2*A^2*a^5*b^2*c^4*d^9*f^2))/f^4)*((((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2

- 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2
*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 +
A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*A^
2*b^2*c^3*f^2 + 8*A^2*a*b*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*A^2*b^2*c*d^2
*f^2 - 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2
) + (32*(23*A^3*b^6*c^3*d^12*f^2 - 15*A^3*a^3*b^3*d^15*f^2 + 21*A^3*b^6*c^5
*d^10*f^2 - 3*A^3*b^6*c^7*d^8*f^2 + A^3*a*b^5*d^15*f^2 + 4*A^3*a^5*b*d^15*f
^2 - A^3*b^6*c*d^14*f^2 - 61*A^3*a*b^5*c^2*d^13*f^2 - 25*A^3*a*b^5*c^4*d^11
*f^2 + 37*A^3*a*b^5*c^6*d^9*f^2 + 53*A^3*a^2*b^4*c*d^14*f^2 - 30*A^3*a^4*b^
2*c*d^14*f^2 + 4*A^3*a^5*b*c^2*d^13*f^2 - 29*A^3*a^2*b^4*c^3*d^12*f^2 - 81*
A^3*a^2*b^4*c^5*d^10*f^2 + A^3*a^2*b^4*c^7*d^8*f^2 + 59*A^3*a^3*b^3*c^2*d^1
3*f^2 + 75*A^3*a^3*b^3*c^4*d^11*f^2 + A^3*a^3*b^3*c^6*d^9*f^2 - 32*A^3*a^4*
b^2*c^3*d^12*f^2 - 2*A^3*a^4*b^2*c^5*d^10*f^2))/f^5)*((((8*A^2*a^2*c^3*f^2
- 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^
2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2
*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) - 4*A^
2*a^2*c^3*f^2 + 4*A^2*b^2*c^3*f^2 + 8*A^2*a*b*d^3*f^2 + 12*A^2*a^2*c*d^2*f^
2 - 12*A^2*b^2*c*d^2*f^2 - 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2
*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*d^16 + 4*A^
4*b^5*c^2*d^14 + 8*A^4*b^5*c^4*d^12 - 8*A^4*b^5*c^6*d^10 + 3*A^4*b^5*c^8*d^
8 + 2*A^4*a^4*b*d^16 + 12*A^4*a^2*b^3*c^2*d^14 - 72*A^4*a^2*b^3*c^4*d^12 +
12*A^4*a^2*b^3*c^6*d^10 + 48*A^4*a^3*b^2*c^3*d^13 - 8*A^4*a^3*b^2*c^5*d^11
- 8*A^4*a*b^4*c^3*d^13 + 48*A^4*a*b^4*c^5*d^11 - 8*A^4*a*b^4*c^7*d^9 - 8*A^
4*a^3*b^2*c*d^15 - 12*A^4*a^4*b*c^2*d^14 + 2*A^4*a^4*b*c^4*d^12))/f^4)*((((
8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d
^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 1
6*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*
d^2))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*A^2*b^2*c^3*f^2 + 8*A^2*a*b*d^3*f^2 + 1
2*A^2*a^2*c*d^2*f^2 - 12*A^2*b^2*c*d^2*f^2 - 24*A^2*a*b*c^2*d*f^2)/(16*(a^4
*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (((((32*(4*A*a^2*b^6*d^12*f^4 + 8
*A*a^4*b^4*d^12*f^4 + 4*A*a^6*b^2*d^12*f^4 + 12*A*b^8*c^2*d^10*f^4 + 12*A*b
^8*c^4*d^8*f^4 - 16*A*a*b^7*c^3*d^9*f^4 - 32*A*a^3*b^5*c*d^11*f^4 - 16*A*a^
5*b^3*c*d^11*f^4 + 28*A*a^2*b^6*c^2*d^10*f^4 + 24*A*a^2*b^6*c^4*d^8*f^4 - 3
2*A*a^3*b^5*c^3*d^9*f^4 + 20*A*a^4*b^4*c^2*d^10*f^4 + 12*A*a^4*b^4*c^4*d^8*
f^4 - 16*A*a^5*b^3*c^3*d^9*f^4 + 4*A*a^6*b^2*c^2*d^10*f^4 - 16*A*a*b^7*c*d^
11*f^4))/f^5 + (32*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^2*c^3*f^2 - 8*A^2
*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2
*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^
4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) - 4*A^2*a^2*c
^3*f^2 + 4*A^2*b^2*c^3*f^2 + 8*A^2*a*b*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*
A^2*b^2*c*d^2*f^2 - 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^
2*f^4)))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4
- 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^
4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*
c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/f^4)*((((8*A^2*a^2

*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 +
24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4
+ 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/
2) - 4*A^2*a^2*c^3*f^2 + 4*A^2*b^2*c^3*f^2 + 8*A^2*a*b*d^3*f^2 + 12*A^2*a^2
*c*d^2*f^2 - 12*A^2*b^2*c*d^2*f^2 - 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^
4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(4*A^2*a^3*
b^4*d^13*f^2 - 14*A^2*a^5*b^2*d^13*f^2 + 28*A^2*b^7*c^3*d^10*f^2 - 18*A^2*b
^7*c^5*d^8*f^2 - 14*A^2*a*b^6*d^13*f^2 + 22*A^2*b^7*c*d^12*f^2 + 8*A^2*a^6*
b*c*d^12*f^2 + 20*A^2*a*b^6*c^2*d^11*f^2 + 66*A^2*a*b^6*c^4*d^9*f^2 - 28*A^
2*a^2*b^5*c*d^12*f^2 + 54*A^2*a^4*b^3*c*d^12*f^2 + 24*A^2*a^2*b^5*c^3*d^10*
f^2 + 12*A^2*a^2*b^5*c^5*d^8*f^2 - 88*A^2*a^3*b^4*c^2*d^11*f^2 - 28*A^2*a^3
*b^4*c^4*d^9*f^2 + 60*A^2*a^4*b^3*c^3*d^10*f^2 - 2*A^2*a^4*b^3*c^5*d^8*f^2
- 44*A^2*a^5*b^2*c^2*d^11*f^2 + 2*A^2*a^5*b^2*c^4*d^9*f^2))/f^4)*((((8*A^2*
a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2
+ 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*
f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^
(1/2) - 4*A^2*a^2*c^3*f^2 + 4*A^2*b^2*c^3*f^2 + 8*A^2*a*b*d^3*f^2 + 12*A^2*
a^2*c*d^2*f^2 - 12*A^2*b^2*c*d^2*f^2 - 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 +
b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(23*A^3*b^6*c^3*d^12*f^2 - 15*A^3*a
^3*b^3*d^15*f^2 + 21*A^3*b^6*c^5*d^10*f^2 - 3*A^3*b^6*c^7*d^8*f^2 + A^3*a*b
^5*d^15*f^2 + 4*A^3*a^5*b*d^15*f^2 - A^3*b^6*c*d^14*f^2 - 61*A^3*a*b^5*c^2*
d^13*f^2 - 25*A^3*a*b^5*c^4*d^11*f^2 + 37*A^3*a*b^5*c^6*d^9*f^2 + 53*A^3*a^
2*b^4*c*d^14*f^2 - 30*A^3*a^4*b^2*c*d^14*f^2 + 4*A^3*a^5*b*c^2*d^13*f^2 - 2
9*A^3*a^2*b^4*c^3*d^12*f^2 - 81*A^3*a^2*b^4*c^5*d^10*f^2 + A^3*a^2*b^4*c^7*
d^8*f^2 + 59*A^3*a^3*b^3*c^2*d^13*f^2 + 75*A^3*a^3*b^3*c^4*d^11*f^2 + A^3*a
^3*b^3*c^6*d^9*f^2 - 32*A^3*a^4*b^2*c^3*d^12*f^2 - 2*A^3*a^4*b^2*c^5*d^10*f
^2))/f^5)*((((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 -
24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (
16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^
4 + 3*A^4*c^4*d^2))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*A^2*b^2*c^3*f^2 + 8*A^2*a
*b*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*A^2*b^2*c*d^2*f^2 - 24*A^2*a*b*c^2*d
f^2)/(16(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e +
f*x))^(1/2)*(A^4*b^5*d^16 + 4*A^4*b^5*c^2*d^14 + 8*A^4*b^5*c^4*d^12 - 8*A^4
*b^5*c^6*d^10 + 3*A^4*b^5*c^8*d^8 + 2*A^4*a^4*b*d^16 + 12*A^4*a^2*b^3*c^2*d
^14 - 72*A^4*a^2*b^3*c^4*d^12 + 12*A^4*a^2*b^3*c^6*d^10 + 48*A^4*a^3*b^2*c^
3*d^13 - 8*A^4*a^3*b^2*c^5*d^11 - 8*A^4*a*b^4*c^3*d^13 + 48*A^4*a*b^4*c^5*d
^11 - 8*A^4*a*b^4*c^7*d^9 - 8*A^4*a^3*b^2*c*d^15 - 12*A^4*a^4*b*c^2*d^14 +
2*A^4*a^4*b*c^4*d^12))/f^4)*((((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*
A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*
c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*
d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*A^2*b^2
*c^3*f^2 + 8*A^2*a*b*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*A^2*b^2*c*d^2*f^2
- 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (
64*(A^5*a^2*b^2*d^18 + A^5*b^4*c^2*d^16 + 5*A^5*b^4*c^4*d^14 + 7*A^5*b^4*c^
6*d^12 + 3*A^5*b^4*c^8*d^10 + 9*A^5*a^2*b^2*c^2*d^16 + 15*A^5*a^2*b^2*c^4*d

^14 + 7*A^5*a^2*b^2*c^6*d^12 - 2*A^5*a*b^3*c*d^17 - 2*A^5*a^3*b*c*d^17 - 12
*A^5*a*b^3*c^3*d^15 - 18*A^5*a*b^3*c^5*d^13 - 8*A^5*a*b^3*c^7*d^11 - 4*A^5*
a^3*b*c^3*d^15 - 2*A^5*a^3*b*c^5*d^13))/f^5))*((((8*A^2*a^2*c^3*f^2 - 8*A^2
*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2
*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^
4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) - 4*A^2*a^2*c
^3*f^2 + 4*A^2*b^2*c^3*f^2 + 8*A^2*a*b*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*
A^2*b^2*c*d^2*f^2 - 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^
2*f^4)))^(1/2)*2i - atan((((((32*(4*A*a^2*b^6*d^12*f^4 + 8*A*a^4*b^4*d^12*
f^4 + 4*A*a^6*b^2*d^12*f^4 + 12*A*b^8*c^2*d^10*f^4 + 12*A*b^8*c^4*d^8*f^4 -
16*A*a*b^7*c^3*d^9*f^4 - 32*A*a^3*b^5*c*d^11*f^4 - 16*A*a^5*b^3*c*d^11*f^4
+ 28*A*a^2*b^6*c^2*d^10*f^4 + 24*A*a^2*b^6*c^4*d^8*f^4 - 32*A*a^3*b^5*c^3*
d^9*f^4 + 20*A*a^4*b^4*c^2*d^10*f^4 + 12*A*a^4*b^4*c^4*d^8*f^4 - 16*A*a^5*b
^3*c^3*d^9*f^4 + 4*A*a^6*b^2*c^2*d^10*f^4 - 16*A*a*b^7*c*d^11*f^4))/f^5 - (
32*(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 -
16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a
*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A
^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*A^2*
b^2*c^3*f^2 - 8*A^2*a*b*d^3*f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*A^2*b^2*c*d^2*f
^2 + 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*
(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d
^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f
^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*
a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/f^4)*(-(((8*A^2*a^2*c^3*f^2 - 8*A
^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d
^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*
f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) + 4*A^2*a^2
*c^3*f^2 - 4*A^2*b^2*c^3*f^2 - 8*A^2*a*b*d^3*f^2 - 12*A^2*a^2*c*d^2*f^2 + 1
2*A^2*b^2*c*d^2*f^2 + 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*
b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(4*A^2*a^3*b^4*d^13*f^2 -
14*A^2*a^5*b^2*d^13*f^2 + 28*A^2*b^7*c^3*d^10*f^2 - 18*A^2*b^7*c^5*d^8*f^2
- 14*A^2*a*b^6*d^13*f^2 + 22*A^2*b^7*c*d^12*f^2 + 8*A^2*a^6*b*c*d^12*f^2 +
20*A^2*a*b^6*c^2*d^11*f^2 + 66*A^2*a*b^6*c^4*d^9*f^2 - 28*A^2*a^2*b^5*c*d^
12*f^2 + 54*A^2*a^4*b^3*c*d^12*f^2 + 24*A^2*a^2*b^5*c^3*d^10*f^2 + 12*A^2*a
^2*b^5*c^5*d^8*f^2 - 88*A^2*a^3*b^4*c^2*d^11*f^2 - 28*A^2*a^3*b^4*c^4*d^9*f
^2 + 60*A^2*a^4*b^3*c^3*d^10*f^2 - 2*A^2*a^4*b^3*c^5*d^8*f^2 - 44*A^2*a^5*b
^2*c^2*d^11*f^2 + 2*A^2*a^5*b^2*c^4*d^9*f^2))/f^4)*(-(((8*A^2*a^2*c^3*f^2 -
8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2
*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*
b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) + 4*A^2
*a^2*c^3*f^2 - 4*A^2*b^2*c^3*f^2 - 8*A^2*a*b*d^3*f^2 - 12*A^2*a^2*c*d^2*f^2
+ 12*A^2*b^2*c*d^2*f^2 + 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*
a^2*b^2*f^4)))^(1/2) + (32*(23*A^3*b^6*c^3*d^12*f^2 - 15*A^3*a^3*b^3*d^15*f
^2 + 21*A^3*b^6*c^5*d^10*f^2 - 3*A^3*b^6*c^7*d^8*f^2 + A^3*a*b^5*d^15*f^2 +
4*A^3*a^5*b*d^15*f^2 - A^3*b^6*c*d^14*f^2 - 61*A^3*a*b^5*c^2*d^13*f^2 - 25

*A^3*a*b^5*c^4*d^11*f^2 + 37*A^3*a*b^5*c^6*d^9*f^2 + 53*A^3*a^2*b^4*c*d^14*
f^2 - 30*A^3*a^4*b^2*c*d^14*f^2 + 4*A^3*a^5*b*c^2*d^13*f^2 - 29*A^3*a^2*b^4
*c^3*d^12*f^2 - 81*A^3*a^2*b^4*c^5*d^10*f^2 + A^3*a^2*b^4*c^7*d^8*f^2 + 59*
A^3*a^3*b^3*c^2*d^13*f^2 + 75*A^3*a^3*b^3*c^4*d^11*f^2 + A^3*a^3*b^3*c^6*d^
9*f^2 - 32*A^3*a^4*b^2*c^3*d^12*f^2 - 2*A^3*a^4*b^2*c^5*d^10*f^2))/f^5)*(-(
((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c
*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 +
16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^
4*d^2))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*A^2*b^2*c^3*f^2 - 8*A^2*a*b*d^3*f^2 -
12*A^2*a^2*c*d^2*f^2 + 12*A^2*b^2*c*d^2*f^2 + 24*A^2*a*b*c^2*d*f^2)/(16*(a
^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*
(A^4*b^5*d^16 + 4*A^4*b^5*c^2*d^14 + 8*A^4*b^5*c^4*d^12 - 8*A^4*b^5*c^6*d^1
0 + 3*A^4*b^5*c^8*d^8 + 2*A^4*a^4*b*d^16 + 12*A^4*a^2*b^3*c^2*d^14 - 72*A^4
*a^2*b^3*c^4*d^12 + 12*A^4*a^2*b^3*c^6*d^10 + 48*A^4*a^3*b^2*c^3*d^13 - 8*A
^4*a^3*b^2*c^5*d^11 - 8*A^4*a*b^4*c^3*d^13 + 48*A^4*a*b^4*c^5*d^11 - 8*A^4*
a*b^4*c^7*d^9 - 8*A^4*a^3*b^2*c*d^15 - 12*A^4*a^4*b*c^2*d^14 + 2*A^4*a^4*b*
c^4*d^12))/f^4)*(-(((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3
*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^
2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4
*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*A^2*b^2*c^3*f^2 -
8*A^2*a*b*d^3*f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*A^2*b^2*c*d^2*f^2 + 24*A^2*a*
b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*1i - (((((32*(
4*A*a^2*b^6*d^12*f^4 + 8*A*a^4*b^4*d^12*f^4 + 4*A*a^6*b^2*d^12*f^4 + 12*A*b
^8*c^2*d^10*f^4 + 12*A*b^8*c^4*d^8*f^4 - 16*A*a*b^7*c^3*d^9*f^4 - 32*A*a^3*
b^5*c*d^11*f^4 - 16*A*a^5*b^3*c*d^11*f^4 + 28*A*a^2*b^6*c^2*d^10*f^4 + 24*A
*a^2*b^6*c^4*d^8*f^4 - 32*A*a^3*b^5*c^3*d^9*f^4 + 20*A*a^4*b^4*c^2*d^10*f^4
+ 12*A*a^4*b^4*c^4*d^8*f^4 - 16*A*a^5*b^3*c^3*d^9*f^4 + 4*A*a^6*b^2*c^2*d^
10*f^4 - 16*A*a*b^7*c*d^11*f^4))/f^5 + (32*(c + d*tan(e + f*x))^(1/2)*(-(((
8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d
^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 1
6*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*
d^2))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*A^2*b^2*c^3*f^2 - 8*A^2*a*b*d^3*f^2 - 1
2*A^2*a^2*c*d^2*f^2 + 12*A^2*b^2*c*d^2*f^2 + 24*A^2*a*b*c^2*d*f^2)/(16*(a^4
*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*
f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a
^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^
8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9
f^4))/f^4)(-(((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2
- 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4
- (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2
*d^4 + 3*A^4*c^4*d^2))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*A^2*b^2*c^3*f^2 - 8*A^
2*a*b*d^3*f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*A^2*b^2*c*d^2*f^2 + 24*A^2*a*b*c^
2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e
+ f*x))^(1/2)*(4*A^2*a^3*b^4*d^13*f^2 - 14*A^2*a^5*b^2*d^13*f^2 + 28*A^2*b
^7*c^3*d^10*f^2 - 18*A^2*b^7*c^5*d^8*f^2 - 14*A^2*a*b^6*d^13*f^2 + 22*A^2*b

^7*c*d^12*f^2 + 8*A^2*a^6*b*c*d^12*f^2 + 20*A^2*a*b^6*c^2*d^11*f^2 + 66*A^2
*a*b^6*c^4*d^9*f^2 - 28*A^2*a^2*b^5*c*d^12*f^2 + 54*A^2*a^4*b^3*c*d^12*f^2
+ 24*A^2*a^2*b^5*c^3*d^10*f^2 + 12*A^2*a^2*b^5*c^5*d^8*f^2 - 88*A^2*a^3*b^4
*c^2*d^11*f^2 - 28*A^2*a^3*b^4*c^4*d^9*f^2 + 60*A^2*a^4*b^3*c^3*d^10*f^2 -
2*A^2*a^4*b^3*c^5*d^8*f^2 - 44*A^2*a^5*b^2*c^2*d^11*f^2 + 2*A^2*a^5*b^2*c^4
*d^9*f^2))/f^4)*(-(((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3
*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^
2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4
*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*A^2*b^2*c^3*f^2 -
8*A^2*a*b*d^3*f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*A^2*b^2*c*d^2*f^2 + 24*A^2*a*
b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(23*A^3*
b^6*c^3*d^12*f^2 - 15*A^3*a^3*b^3*d^15*f^2 + 21*A^3*b^6*c^5*d^10*f^2 - 3*A^
3*b^6*c^7*d^8*f^2 + A^3*a*b^5*d^15*f^2 + 4*A^3*a^5*b*d^15*f^2 - A^3*b^6*c*d
^14*f^2 - 61*A^3*a*b^5*c^2*d^13*f^2 - 25*A^3*a*b^5*c^4*d^11*f^2 + 37*A^3*a*
b^5*c^6*d^9*f^2 + 53*A^3*a^2*b^4*c*d^14*f^2 - 30*A^3*a^4*b^2*c*d^14*f^2 + 4
*A^3*a^5*b*c^2*d^13*f^2 - 29*A^3*a^2*b^4*c^3*d^12*f^2 - 81*A^3*a^2*b^4*c^5*
d^10*f^2 + A^3*a^2*b^4*c^7*d^8*f^2 + 59*A^3*a^3*b^3*c^2*d^13*f^2 + 75*A^3*a
^3*b^3*c^4*d^11*f^2 + A^3*a^3*b^3*c^6*d^9*f^2 - 32*A^3*a^4*b^2*c^3*d^12*f^2
 - 2*A^3*a^4*b^2*c^5*d^10*f^2))/f^5)*(-(((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3
*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 4
8*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*
c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) + 4*A^2*a^2*c^3*f^2 -
 4*A^2*b^2*c^3*f^2 - 8*A^2*a*b*d^3*f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*A^2*b^2*
c*d^2*f^2 + 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))
^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*d^16 + 4*A^4*b^5*c^2*d^14
+ 8*A^4*b^5*c^4*d^12 - 8*A^4*b^5*c^6*d^10 + 3*A^4*b^5*c^8*d^8 + 2*A^4*a^4*b
*d^16 + 12*A^4*a^2*b^3*c^2*d^14 - 72*A^4*a^2*b^3*c^4*d^12 + 12*A^4*a^2*b^3*
c^6*d^10 + 48*A^4*a^3*b^2*c^3*d^13 - 8*A^4*a^3*b^2*c^5*d^11 - 8*A^4*a*b^4*c
^3*d^13 + 48*A^4*a*b^4*c^5*d^11 - 8*A^4*a*b^4*c^7*d^9 - 8*A^4*a^3*b^2*c*d^1
5 - 12*A^4*a^4*b*c^2*d^14 + 2*A^4*a^4*b*c^4*d^12))/f^4)*(-(((8*A^2*a^2*c^3*
f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^
2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32
*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) +
4*A^2*a^2*c^3*f^2 - 4*A^2*b^2*c^3*f^2 - 8*A^2*a*b*d^3*f^2 - 12*A^2*a^2*c*d^
2*f^2 + 12*A^2*b^2*c*d^2*f^2 + 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4
 + 2*a^2*b^2*f^4)))^(1/2)*1i)/((((((32*(4*A*a^2*b^6*d^12*f^4 + 8*A*a^4*b^4*
d^12*f^4 + 4*A*a^6*b^2*d^12*f^4 + 12*A*b^8*c^2*d^10*f^4 + 12*A*b^8*c^4*d^8*
f^4 - 16*A*a*b^7*c^3*d^9*f^4 - 32*A*a^3*b^5*c*d^11*f^4 - 16*A*a^5*b^3*c*d^1
1*f^4 + 28*A*a^2*b^6*c^2*d^10*f^4 + 24*A*a^2*b^6*c^4*d^8*f^4 - 32*A*a^3*b^5
*c^3*d^9*f^4 + 20*A*a^4*b^4*c^2*d^10*f^4 + 12*A*a^4*b^4*c^4*d^8*f^4 - 16*A*
a^5*b^3*c^3*d^9*f^4 + 4*A*a^6*b^2*c^2*d^10*f^4 - 16*A*a*b^7*c*d^11*f^4))/f^
5 - (32*(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f
^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*
A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^
6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4

*A^2*b^2*c^3*f^2 - 8*A^2*a*b*d^3*f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*A^2*b^2*c*
d^2*f^2 + 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(
1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*
b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*
d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4
+ 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/f^4)*(-(((8*A^2*a^2*c^3*f^2
- 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^
2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2
*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) + 4*A^
2*a^2*c^3*f^2 - 4*A^2*b^2*c^3*f^2 - 8*A^2*a*b*d^3*f^2 - 12*A^2*a^2*c*d^2*f^
2 + 12*A^2*b^2*c*d^2*f^2 + 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2
*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(4*A^2*a^3*b^4*d^13*
f^2 - 14*A^2*a^5*b^2*d^13*f^2 + 28*A^2*b^7*c^3*d^10*f^2 - 18*A^2*b^7*c^5*d^
8*f^2 - 14*A^2*a*b^6*d^13*f^2 + 22*A^2*b^7*c*d^12*f^2 + 8*A^2*a^6*b*c*d^12*
f^2 + 20*A^2*a*b^6*c^2*d^11*f^2 + 66*A^2*a*b^6*c^4*d^9*f^2 - 28*A^2*a^2*b^5
*c*d^12*f^2 + 54*A^2*a^4*b^3*c*d^12*f^2 + 24*A^2*a^2*b^5*c^3*d^10*f^2 + 12*
A^2*a^2*b^5*c^5*d^8*f^2 - 88*A^2*a^3*b^4*c^2*d^11*f^2 - 28*A^2*a^3*b^4*c^4*
d^9*f^2 + 60*A^2*a^4*b^3*c^3*d^10*f^2 - 2*A^2*a^4*b^3*c^5*d^8*f^2 - 44*A^2*
a^5*b^2*c^2*d^11*f^2 + 2*A^2*a^5*b^2*c^4*d^9*f^2))/f^4)*(-(((8*A^2*a^2*c^3*
f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^
2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32
*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) +
4*A^2*a^2*c^3*f^2 - 4*A^2*b^2*c^3*f^2 - 8*A^2*a*b*d^3*f^2 - 12*A^2*a^2*c*d^
2*f^2 + 12*A^2*b^2*c*d^2*f^2 + 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4
+ 2*a^2*b^2*f^4)))^(1/2) + (32*(23*A^3*b^6*c^3*d^12*f^2 - 15*A^3*a^3*b^3*d
^15*f^2 + 21*A^3*b^6*c^5*d^10*f^2 - 3*A^3*b^6*c^7*d^8*f^2 + A^3*a*b^5*d^15*
f^2 + 4*A^3*a^5*b*d^15*f^2 - A^3*b^6*c*d^14*f^2 - 61*A^3*a*b^5*c^2*d^13*f^2
- 25*A^3*a*b^5*c^4*d^11*f^2 + 37*A^3*a*b^5*c^6*d^9*f^2 + 53*A^3*a^2*b^4*c*
d^14*f^2 - 30*A^3*a^4*b^2*c*d^14*f^2 + 4*A^3*a^5*b*c^2*d^13*f^2 - 29*A^3*a^
2*b^4*c^3*d^12*f^2 - 81*A^3*a^2*b^4*c^5*d^10*f^2 + A^3*a^2*b^4*c^7*d^8*f^2
+ 59*A^3*a^3*b^3*c^2*d^13*f^2 + 75*A^3*a^3*b^3*c^4*d^11*f^2 + A^3*a^3*b^3*c
^6*d^9*f^2 - 32*A^3*a^4*b^2*c^3*d^12*f^2 - 2*A^3*a^4*b^2*c^5*d^10*f^2))/f^5
)*(-(((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*
a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*
f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A
^4*c^4*d^2))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*A^2*b^2*c^3*f^2 - 8*A^2*a*b*d^3*
f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*A^2*b^2*c*d^2*f^2 + 24*A^2*a*b*c^2*d*f^2)/(
16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(
1/2)*(A^4*b^5*d^16 + 4*A^4*b^5*c^2*d^14 + 8*A^4*b^5*c^4*d^12 - 8*A^4*b^5*c^
6*d^10 + 3*A^4*b^5*c^8*d^8 + 2*A^4*a^4*b*d^16 + 12*A^4*a^2*b^3*c^2*d^14 - 7
2*A^4*a^2*b^3*c^4*d^12 + 12*A^4*a^2*b^3*c^6*d^10 + 48*A^4*a^3*b^2*c^3*d^13
- 8*A^4*a^3*b^2*c^5*d^11 - 8*A^4*a*b^4*c^3*d^13 + 48*A^4*a*b^4*c^5*d^11 - 8
*A^4*a*b^4*c^7*d^9 - 8*A^4*a^3*b^2*c*d^15 - 12*A^4*a^4*b*c^2*d^14 + 2*A^4*a
^4*b*c^4*d^12))/f^4)*(-(((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*
b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*

f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*A^2*b^2*c^3*f^2 - 8*A^2*a*b*d^3*f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*A^2*b^2*c*d^2*f^2 + 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (((((32*(4*A*a^2*b^6*d^12*f^4 + 8*A*a^4*b^4*d^12*f^4 + 4*A*a^6*b^2*d^12*f^4 + 12*A*b^8*c^2*d^10*f^4 + 12*A*b^8*c^4*d^8*f^4 - 16*A*a*b^7*c^3*d^9*f^4 - 32*A*a^3*b^5*c*d^11*f^4 - 16*A*a^5*b^3*c*d^11*f^4 + 28*A*a^2*b^6*c^2*d^10*f^4 + 24*A*a^2*b^6*c^4*d^8*f^4 - 32*A*a^3*b^5*c^3*d^9*f^4 + 20*A*a^4*b^4*c^2*d^10*f^4 + 12*A*a^4*b^4*c^4*d^8*f^4 - 16*A*a^5*b^3*c^3*d^9*f^4 + 4*A*a^6*b^2*c^2*d^10*f^4 - 16*A*a*b^7*c*d^11*f^4))/f^5 + (32*(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*A^2*b^2*c^3*f^2 - 8*A^2*a*b*d^3*f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*A^2*b^2*c*d^2*f^2 + 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/f^4)*(-(((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*A^2*b^2*c^3*f^2 - 8*A^2*a*b*d^3*f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*A^2*b^2*c*d^2*f^2 + 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(4*A^2*a^3*b^4*d^13*f^2 - 14*A^2*a^5*b^2*d^13*f^2 + 28*A^2*b^7*c^3*d^10*f^2 - 18*A^2*b^7*c^5*d^8*f^2 - 14*A^2*a*b^6*d^13*f^2 + 22*A^2*b^7*c*d^12*f^2 + 8*A^2*a^6*b*c*d^12*f^2 + 20*A^2*a*b^6*c^2*d^11*f^2 + 66*A^2*a*b^6*c^4*d^9*f^2 - 28*A^2*a^2*b^5*c*d^12*f^2 + 54*A^2*a^4*b^3*c*d^12*f^2 + 24*A^2*a^2*b^5*c^3*d^10*f^2 + 12*A^2*a^2*b^5*c^5*d^8*f^2 - 88*A^2*a^3*b^4*c^2*d^11*f^2 - 28*A^2*a^3*b^4*c^4*d^9*f^2 + 60*A^2*a^4*b^3*c^3*d^10*f^2 - 2*A^2*a^4*b^3*c^5*d^8*f^2 - 44*A^2*a^5*b^2*c^2*d^11*f^2 + 2*A^2*a^5*b^2*c^4*d^9*f^2))/f^4)*(-(((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*A^2*b^2*c^3*f^2 - 8*A^2*a*b*d^3*f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*A^2*b^2*c*d^2*f^2 + 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(23*A^3*b^6*c^3*d^12*f^2 - 15*A^3*a^3*b^3*d^15*f^2 + 21*A^3*b^6*c^5*d^10*f^2 - 3*A^3*b^6*c^7*d^8*f^2 + A^3*a*b^5*d^15*f^2 + 4*A^3*a^5*b*d^15*f^2 - A^3*b^6*c*d^14*f^2 - 61*A^3*a*b^5*c^2*d^13*f^2 - 25*A^3*a*b^5*c^4*d^11*f^2 + 37*A^3*a*b^5*c^6*d^9*f^2 + 53*A^3*a^2*b^4*c*d^14*f^2 - 30*A^3*a^4*b^2*c*d^14*f^2 + 4*A^3*a^5*b*c^2*d^13*f^2 - 29*A^3*a^2*b^4*c^3*d^12*f^2 - 81*A^3*a^2*b^4*c^5*d^10*f^2 + A^3*a^2*b^4*c^7*d^8*f^2 + 59*A^3*a^3*b^3*c^2*d^13*f^2 + 75*A^3*a^3*b^3*c^4*d^11*f^2 + A^3*a^3*b^3*c^6*d^9*f^2 - 32*A^3*a^4*b^2*c^3*d^12*f

^2 - 2*A^3*a^4*b^2*c^5*d^10*f^2))/f^5)*(-(((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c
^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 +
48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^
4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) + 4*A^2*a^2*c^3*f^2
- 4*A^2*b^2*c^3*f^2 - 8*A^2*a*b*d^3*f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*A^2*b^
2*c*d^2*f^2 + 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)
))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*d^16 + 4*A^4*b^5*c^2*d^1
4 + 8*A^4*b^5*c^4*d^12 - 8*A^4*b^5*c^6*d^10 + 3*A^4*b^5*c^8*d^8 + 2*A^4*a^4
*b*d^16 + 12*A^4*a^2*b^3*c^2*d^14 - 72*A^4*a^2*b^3*c^4*d^12 + 12*A^4*a^2*b^
3*c^6*d^10 + 48*A^4*a^3*b^2*c^3*d^13 - 8*A^4*a^3*b^2*c^5*d^11 - 8*A^4*a*b^4
*c^3*d^13 + 48*A^4*a*b^4*c^5*d^11 - 8*A^4*a*b^4*c^7*d^9 - 8*A^4*a^3*b^2*c*d
^15 - 12*A^4*a^4*b*c^2*d^14 + 2*A^4*a^4*b*c^4*d^12))/f^4)*(-(((8*A^2*a^2*c^
3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*
A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 +
32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2)
+ 4*A^2*a^2*c^3*f^2 - 4*A^2*b^2*c^3*f^2 - 8*A^2*a*b*d^3*f^2 - 12*A^2*a^2*c*
d^2*f^2 + 12*A^2*b^2*c*d^2*f^2 + 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f
^4 + 2*a^2*b^2*f^4)))^(1/2) + (64*(A^5*a^2*b^2*d^18 + A^5*b^4*c^2*d^16 + 5*
A^5*b^4*c^4*d^14 + 7*A^5*b^4*c^6*d^12 + 3*A^5*b^4*c^8*d^10 + 9*A^5*a^2*b^2*
c^2*d^16 + 15*A^5*a^2*b^2*c^4*d^14 + 7*A^5*a^2*b^2*c^6*d^12 - 2*A^5*a*b^3*c
*d^17 - 2*A^5*a^3*b*c*d^17 - 12*A^5*a*b^3*c^3*d^15 - 18*A^5*a*b^3*c^5*d^13
- 8*A^5*a*b^3*c^7*d^11 - 4*A^5*a^3*b*c^3*d^15 - 2*A^5*a^3*b*c^5*d^13))/f^5)
)*(-(((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*
a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*
f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A
^4*c^4*d^2))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*A^2*b^2*c^3*f^2 - 8*A^2*a*b*d^3*
f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*A^2*b^2*c*d^2*f^2 + 24*A^2*a*b*c^2*d*f^2)/(
16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*2i - ((4*C*c)/(b*f) + (2*C*(
a*d*f - 3*b*c*f))/(b^2*f^2))*(c + d*tan(e + f*x))^(1/2) + atan(((((((32*(4*
B*a*b^8*d^12*f^4 - 4*B*b^9*c*d^11*f^4 + 8*B*a^3*b^6*d^12*f^4 + 4*B*a^5*b^4*
d^12*f^4 - 4*B*b^9*c^3*d^9*f^4 + 8*B*a*b^8*c^2*d^10*f^4 + 4*B*a*b^8*c^4*d^8
*f^4 - 12*B*a^2*b^7*c*d^11*f^4 - 12*B*a^4*b^5*c*d^11*f^4 - 4*B*a^6*b^3*c*d^
11*f^4 - 12*B*a^2*b^7*c^3*d^9*f^4 + 16*B*a^3*b^6*c^2*d^10*f^4 + 8*B*a^3*b^6
*c^4*d^8*f^4 - 12*B*a^4*b^5*c^3*d^9*f^4 + 8*B*a^5*b^4*c^2*d^10*f^4 + 4*B*a^
5*b^4*c^4*d^8*f^4 - 4*B*a^6*b^3*c^3*d^9*f^4))/(b*f^5) - (32*(c + d*tan(e +
f*x))^(1/2)*((((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2
- 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 -
(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*
d^4 + 3*B^4*c^4*d^2))^(1/2) + 4*B^2*a^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 - 8*B^2
*a*b*d^3*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 + 24*B^2*a*b*c^2
*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^10*d^10*f^4 +
16*a^2*b^8*d^10*f^4 - 16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*d^10*f^4 + 24*b^10*
c^2*d^8*f^4 + 40*a^2*b^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^
2*d^8*f^4 + 8*a*b^9*c*d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 24*a^5*b^5*c*d^9*f^4
+ 8*a^7*b^3*c*d^9*f^4))/(b*f^4))*((((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2

- 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^
2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6
+ B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) + 4*B^2*a^2*c^3*f^2 - 4*B
^2*b^2*c^3*f^2 - 8*B^2*a*b*d^3*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*B^2*b^2*c*d^
2*f^2 + 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/
2) + (32*(c + d*tan(e + f*x))^(1/2)*(4*B^2*a^3*b^5*d^13*f^2 + 2*B^2*a^5*b^3
*d^13*f^2 + 28*B^2*b^8*c^3*d^10*f^2 - 10*B^2*b^8*c^5*d^8*f^2 - 14*B^2*a*b^7
*d^13*f^2 + 16*B^2*a^7*b*d^13*f^2 - 8*B^2*a^8*c*d^12*f^2 + 22*B^2*b^8*c*d^1
2*f^2 + 20*B^2*a*b^7*c^2*d^11*f^2 + 50*B^2*a*b^7*c^4*d^9*f^2 - 28*B^2*a^2*b
^6*c*d^12*f^2 - 2*B^2*a^4*b^4*c*d^12*f^2 - 56*B^2*a^6*b^2*c*d^12*f^2 + 32*B
^2*a^7*b*c^2*d^11*f^2 + 8*B^2*a^2*b^6*c^3*d^10*f^2 + 12*B^2*a^2*b^6*c^5*d^8
*f^2 - 24*B^2*a^3*b^5*c^2*d^11*f^2 - 12*B^2*a^3*b^5*c^4*d^9*f^2 - 4*B^2*a^4
*b^4*c^3*d^10*f^2 - 10*B^2*a^4*b^4*c^5*d^8*f^2 + 52*B^2*a^5*b^3*c^2*d^11*f^
2 + 34*B^2*a^5*b^3*c^4*d^9*f^2 - 48*B^2*a^6*b^2*c^3*d^10*f^2))/(b*f^4))*(((
(8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*
d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 +
16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4
*d^2))^(1/2) + 4*B^2*a^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 - 8*B^2*a*b*d^3*f^2 -
12*B^2*a^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 + 24*B^2*a*b*c^2*d*f^2)/(16*(a^
4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(15*B^3*a^4*b^3*d^15*f^2 - B
^3*a^2*b^5*d^15*f^2 - 4*B^3*a^7*c^3*d^12*f^2 + 2*B^3*b^7*c^2*d^13*f^2 + 4*B
^3*b^7*c^4*d^11*f^2 + 2*B^3*b^7*c^6*d^9*f^2 - 12*B^3*a^6*b*d^15*f^2 - 4*B^3
*a^7*c*d^14*f^2 - B^3*a*b^6*c*d^14*f^2 - 27*B^3*a*b^6*c^3*d^12*f^2 - 19*B^3
*a*b^6*c^5*d^10*f^2 + 7*B^3*a*b^6*c^7*d^8*f^2 - 57*B^3*a^3*b^4*c*d^14*f^2 +
64*B^3*a^5*b^2*c*d^14*f^2 + 4*B^3*a^6*b*c^2*d^13*f^2 + 16*B^3*a^6*b*c^4*d^
11*f^2 + 65*B^3*a^2*b^5*c^2*d^13*f^2 + 9*B^3*a^2*b^5*c^4*d^11*f^2 - 57*B^3*
a^2*b^5*c^6*d^9*f^2 + 77*B^3*a^3*b^4*c^3*d^12*f^2 + 129*B^3*a^3*b^4*c^5*d^1
0*f^2 - 5*B^3*a^3*b^4*c^7*d^8*f^2 - 121*B^3*a^4*b^3*c^2*d^13*f^2 - 119*B^3*
a^4*b^3*c^4*d^11*f^2 + 17*B^3*a^4*b^3*c^6*d^9*f^2 + 40*B^3*a^5*b^2*c^3*d^12
*f^2 - 24*B^3*a^5*b^2*c^5*d^10*f^2))/(b*f^5))*((((8*B^2*a^2*c^3*f^2 - 8*B^2
*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2
*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^
4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) + 4*B^2*a^2*c
^3*f^2 - 4*B^2*b^2*c^3*f^2 - 8*B^2*a*b*d^3*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*
B^2*b^2*c*d^2*f^2 + 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^
2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(B^4*b^6*d^16 - 2*B^4*a^6*d
^16 + 12*B^4*a^6*c^2*d^14 - 2*B^4*a^6*c^4*d^12 + 4*B^4*b^6*c^2*d^14 + 6*B^4
*b^6*c^4*d^12 + 4*B^4*b^6*c^6*d^10 + B^4*b^6*c^8*d^8 - 2*B^4*a^2*b^4*c^4*d^
12 + 12*B^4*a^2*b^4*c^6*d^10 - 2*B^4*a^2*b^4*c^8*d^8 + 8*B^4*a^3*b^3*c^3*d^
13 - 48*B^4*a^3*b^3*c^5*d^11 + 8*B^4*a^3*b^3*c^7*d^9 - 12*B^4*a^4*b^2*c^2*d
^14 + 72*B^4*a^4*b^2*c^4*d^12 - 12*B^4*a^4*b^2*c^6*d^10 + 8*B^4*a^5*b*c*d^1
5 - 48*B^4*a^5*b*c^3*d^13 + 8*B^4*a^5*b*c^5*d^11))/(b*f^4))*((((8*B^2*a^2*c
^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24
*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 +
32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2)

+ 4*B^2*a^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 - 8*B^2*a*b*d^3*f^2 - 12*B^2*a^2*c
*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 + 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*
f^4 + 2*a^2*b^2*f^4)))^(1/2)*1i - (((((32*(4*B*a*b^8*d^12*f^4 - 4*B*b^9*c*d
^11*f^4 + 8*B*a^3*b^6*d^12*f^4 + 4*B*a^5*b^4*d^12*f^4 - 4*B*b^9*c^3*d^9*f^4
+ 8*B*a*b^8*c^2*d^10*f^4 + 4*B*a*b^8*c^4*d^8*f^4 - 12*B*a^2*b^7*c*d^11*f^4
- 12*B*a^4*b^5*c*d^11*f^4 - 4*B*a^6*b^3*c*d^11*f^4 - 12*B*a^2*b^7*c^3*d^9*
f^4 + 16*B*a^3*b^6*c^2*d^10*f^4 + 8*B*a^3*b^6*c^4*d^8*f^4 - 12*B*a^4*b^5*c^
3*d^9*f^4 + 8*B*a^5*b^4*c^2*d^10*f^4 + 4*B*a^5*b^4*c^4*d^8*f^4 - 4*B*a^6*b^
3*c^3*d^9*f^4))/(b*f^5) + (32*(c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^2*c^3*
f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^
2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32
*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) +
4*B^2*a^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 - 8*B^2*a*b*d^3*f^2 - 12*B^2*a^2*c*d^
2*f^2 + 12*B^2*b^2*c*d^2*f^2 + 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4
+ 2*a^2*b^2*f^4)))^(1/2)*(16*b^10*d^10*f^4 + 16*a^2*b^8*d^10*f^4 - 16*a^4*
b^6*d^10*f^4 - 16*a^6*b^4*d^10*f^4 + 24*b^10*c^2*d^8*f^4 + 40*a^2*b^8*c^2*d
^8*f^4 + 8*a^4*b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^9*c*d^9*f^4
+ 24*a^3*b^7*c*d^9*f^4 + 24*a^5*b^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9*f^4))/(b*f^
4))*((((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2
*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4
*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*
B^4*c^4*d^2))^(1/2) + 4*B^2*a^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 - 8*B^2*a*b*d^3
*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 + 24*B^2*a*b*c^2*d*f^2)/
(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^
(1/2)*(4*B^2*a^3*b^5*d^13*f^2 + 2*B^2*a^5*b^3*d^13*f^2 + 28*B^2*b^8*c^3*d^1
0*f^2 - 10*B^2*b^8*c^5*d^8*f^2 - 14*B^2*a*b^7*d^13*f^2 + 16*B^2*a^7*b*d^13*
f^2 - 8*B^2*a^8*c*d^12*f^2 + 22*B^2*b^8*c*d^12*f^2 + 20*B^2*a*b^7*c^2*d^11*
f^2 + 50*B^2*a*b^7*c^4*d^9*f^2 - 28*B^2*a^2*b^6*c*d^12*f^2 - 2*B^2*a^4*b^4*
c*d^12*f^2 - 56*B^2*a^6*b^2*c*d^12*f^2 + 32*B^2*a^7*b*c^2*d^11*f^2 + 8*B^2*
a^2*b^6*c^3*d^10*f^2 + 12*B^2*a^2*b^6*c^5*d^8*f^2 - 24*B^2*a^3*b^5*c^2*d^11
*f^2 - 12*B^2*a^3*b^5*c^4*d^9*f^2 - 4*B^2*a^4*b^4*c^3*d^10*f^2 - 10*B^2*a^4
*b^4*c^5*d^8*f^2 + 52*B^2*a^5*b^3*c^2*d^11*f^2 + 34*B^2*a^5*b^3*c^4*d^9*f^2
- 48*B^2*a^6*b^2*c^3*d^10*f^2))/(b*f^4))*((((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2
*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2
+ 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(
B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) + 4*B^2*a^2*c^3*f
^2 - 4*B^2*b^2*c^3*f^2 - 8*B^2*a*b*d^3*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*B^2*
b^2*c*d^2*f^2 + 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^
4)))^(1/2) + (32*(15*B^3*a^4*b^3*d^15*f^2 - B^3*a^2*b^5*d^15*f^2 - 4*B^3*a^
7*c^3*d^12*f^2 + 2*B^3*b^7*c^2*d^13*f^2 + 4*B^3*b^7*c^4*d^11*f^2 + 2*B^3*b^
7*c^6*d^9*f^2 - 12*B^3*a^6*b*d^15*f^2 - 4*B^3*a^7*c*d^14*f^2 - B^3*a*b^6*c*
d^14*f^2 - 27*B^3*a*b^6*c^3*d^12*f^2 - 19*B^3*a*b^6*c^5*d^10*f^2 + 7*B^3*a*
b^6*c^7*d^8*f^2 - 57*B^3*a^3*b^4*c*d^14*f^2 + 64*B^3*a^5*b^2*c*d^14*f^2 + 4
*B^3*a^6*b*c^2*d^13*f^2 + 16*B^3*a^6*b*c^4*d^11*f^2 + 65*B^3*a^2*b^5*c^2*d^
13*f^2 + 9*B^3*a^2*b^5*c^4*d^11*f^2 - 57*B^3*a^2*b^5*c^6*d^9*f^2 + 77*B^3*a

^3*b^4*c^3*d^12*f^2 + 129*B^3*a^3*b^4*c^5*d^10*f^2 - 5*B^3*a^3*b^4*c^7*d^8*
f^2 - 121*B^3*a^4*b^3*c^2*d^13*f^2 - 119*B^3*a^4*b^3*c^4*d^11*f^2 + 17*B^3*
a^4*b^3*c^6*d^9*f^2 + 40*B^3*a^5*b^2*c^3*d^12*f^2 - 24*B^3*a^5*b^2*c^5*d^10
*f^2))/(b*f^5))*((((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*
f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2
/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*
c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) + 4*B^2*a^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 - 8
*B^2*a*b*d^3*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 + 24*B^2*a*b
*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*ta
n(e + f*x))^(1/2)*(B^4*b^6*d^16 - 2*B^4*a^6*d^16 + 12*B^4*a^6*c^2*d^14 - 2*
B^4*a^6*c^4*d^12 + 4*B^4*b^6*c^2*d^14 + 6*B^4*b^6*c^4*d^12 + 4*B^4*b^6*c^6*
d^10 + B^4*b^6*c^8*d^8 - 2*B^4*a^2*b^4*c^4*d^12 + 12*B^4*a^2*b^4*c^6*d^10 -
 2*B^4*a^2*b^4*c^8*d^8 + 8*B^4*a^3*b^3*c^3*d^13 - 48*B^4*a^3*b^3*c^5*d^11 +
 8*B^4*a^3*b^3*c^7*d^9 - 12*B^4*a^4*b^2*c^2*d^14 + 72*B^4*a^4*b^2*c^4*d^12
- 12*B^4*a^4*b^2*c^6*d^10 + 8*B^4*a^5*b*c*d^15 - 48*B^4*a^5*b*c^3*d^13 + 8*
B^4*a^5*b*c^5*d^11))/(b*f^4))*((((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 1
6*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*
b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^
4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) + 4*B^2*a^2*c^3*f^2 - 4*B^2*b
^2*c^3*f^2 - 8*B^2*a*b*d^3*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^
2 + 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*1
i)/((((((32*(4*B*a*b^8*d^12*f^4 - 4*B*b^9*c*d^11*f^4 + 8*B*a^3*b^6*d^12*f^4
 + 4*B*a^5*b^4*d^12*f^4 - 4*B*b^9*c^3*d^9*f^4 + 8*B*a*b^8*c^2*d^10*f^4 + 4*
B*a*b^8*c^4*d^8*f^4 - 12*B*a^2*b^7*c*d^11*f^4 - 12*B*a^4*b^5*c*d^11*f^4 - 4
*B*a^6*b^3*c*d^11*f^4 - 12*B*a^2*b^7*c^3*d^9*f^4 + 16*B*a^3*b^6*c^2*d^10*f^
4 + 8*B*a^3*b^6*c^4*d^8*f^4 - 12*B*a^4*b^5*c^3*d^9*f^4 + 8*B*a^5*b^4*c^2*d^
10*f^4 + 4*B*a^5*b^4*c^4*d^8*f^4 - 4*B*a^6*b^3*c^3*d^9*f^4))/(b*f^5) - (32*
(c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B
^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c
^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d
^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) + 4*B^2*a^2*c^3*f^2 - 4*B^2*b^2*
c^3*f^2 - 8*B^2*a*b*d^3*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 +
 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*
b^10*d^10*f^4 + 16*a^2*b^8*d^10*f^4 - 16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*d^10
*f^4 + 24*b^10*c^2*d^8*f^4 + 40*a^2*b^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8*f^4
 - 8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^9*c*d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 24*a^
5*b^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9*f^4))/(b*f^4))*((((8*B^2*a^2*c^3*f^2 - 8*
B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*
d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2
f^4)(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) + 4*B^2*a^
2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 - 8*B^2*a*b*d^3*f^2 - 12*B^2*a^2*c*d^2*f^2 +
12*B^2*b^2*c*d^2*f^2 + 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2
*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(4*B^2*a^3*b^5*d^13*f^2
+ 2*B^2*a^5*b^3*d^13*f^2 + 28*B^2*b^8*c^3*d^10*f^2 - 10*B^2*b^8*c^5*d^8*f^2
 - 14*B^2*a*b^7*d^13*f^2 + 16*B^2*a^7*b*d^13*f^2 - 8*B^2*a^8*c*d^12*f^2 + 2

2*B^2*b^8*c*d^12*f^2 + 20*B^2*a*b^7*c^2*d^11*f^2 + 50*B^2*a*b^7*c^4*d^9*f^2 - 28*B^2*a^2*b^6*c*d^12*f^2 - 2*B^2*a^4*b^4*c*d^12*f^2 - 56*B^2*a^6*b^2*c*d^12*f^2 + 32*B^2*a^7*b*c^2*d^11*f^2 + 8*B^2*a^2*b^6*c^3*d^10*f^2 + 12*B^2*a^2*b^6*c^5*d^8*f^2 - 24*B^2*a^3*b^5*c^2*d^11*f^2 - 12*B^2*a^3*b^5*c^4*d^9*f^2 - 4*B^2*a^4*b^4*c^3*d^10*f^2 - 10*B^2*a^4*b^4*c^5*d^8*f^2 + 52*B^2*a^5*b^3*c^2*d^11*f^2 + 34*B^2*a^5*b^3*c^4*d^9*f^2 - 48*B^2*a^6*b^2*c^3*d^10*f^2))/(b*f^4))*((((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) + 4*B^2*a^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 - 8*B^2*a*b*d^3*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 + 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(15*B^3*a^4*b^3*d^15*f^2 - B^3*a^2*b^5*d^15*f^2 - 4*B^3*a^7*c^3*d^12*f^2 + 2*B^3*b^7*c^2*d^13*f^2 + 4*B^3*b^7*c^4*d^11*f^2 + 2*B^3*b^7*c^6*d^9*f^2 - 12*B^3*a^6*b*d^15*f^2 - 4*B^3*a^7*c*d^14*f^2 - B^3*a*b^6*c*d^14*f^2 - 27*B^3*a*b^6*c^3*d^12*f^2 - 19*B^3*a*b^6*c^5*d^10*f^2 + 7*B^3*a*b^6*c^7*d^8*f^2 - 57*B^3*a^3*b^4*c*d^14*f^2 + 64*B^3*a^5*b^2*c*d^14*f^2 + 4*B^3*a^6*b*c^2*d^13*f^2 + 16*B^3*a^6*b*c^4*d^11*f^2 + 65*B^3*a^2*b^5*c^2*d^13*f^2 + 9*B^3*a^2*b^5*c^4*d^11*f^2 - 57*B^3*a^2*b^5*c^6*d^9*f^2 + 77*B^3*a^3*b^4*c^3*d^12*f^2 + 129*B^3*a^3*b^4*c^5*d^10*f^2 - 5*B^3*a^3*b^4*c^7*d^8*f^2 - 121*B^3*a^4*b^3*c^2*d^13*f^2 - 119*B^3*a^4*b^3*c^4*d^11*f^2 + 17*B^3*a^4*b^3*c^6*d^9*f^2 + 40*B^3*a^5*b^2*c^3*d^12*f^2 - 24*B^3*a^5*b^2*c^5*d^10*f^2))/(b*f^5))*((((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) + 4*B^2*a^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 - 8*B^2*a*b*d^3*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 + 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(B^4*b^6*d^16 - 2*B^4*a^6*d^16 + 12*B^4*a^6*c^2*d^14 - 2*B^4*a^6*c^4*d^12 + 4*B^4*b^6*c^2*d^14 + 6*B^4*b^6*c^4*d^12 + 4*B^4*b^6*c^6*d^10 + B^4*b^6*c^8*d^8 - 2*B^4*a^2*b^4*c^4*d^12 + 12*B^4*a^2*b^4*c^6*d^10 - 2*B^4*a^2*b^4*c^8*d^8 + 8*B^4*a^3*b^3*c^3*d^13 - 48*B^4*a^3*b^3*c^5*d^11 + 8*B^4*a^3*b^3*c^7*d^9 - 12*B^4*a^4*b^2*c^2*d^14 + 72*B^4*a^4*b^2*c^4*d^12 - 12*B^4*a^4*b^2*c^6*d^10 + 8*B^4*a^5*b*c*d^15 - 48*B^4*a^5*b*c^3*d^13 + 8*B^4*a^5*b*c^5*d^11))/(b*f^4))*((((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) + 4*B^2*a^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 - 8*B^2*a*b*d^3*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 + 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (((((32*(4*B*a*b^8*d^12*f^4 - 4*B*b^9*c*d^11*f^4 + 8*B*a^3*b^6*d^12*f^4 + 4*B*a^5*b^4*d^12*f^4 - 4*B*b^9*c^3*d^9*f^4 + 8*B*a*b^8*c^2*d^10*f^4 + 4*B*a*b^8*c^4*d^8*f^4 - 12*B*a^2*b^7*c*d^11*f^4 - 12*B*a^4*b^5*c*d^11*f^4 - 4*B*a^6*b^3*c*d^11*f^4 - 12*B*a^2*b^7*c^3*d^9*f^4 + 16*B*a^3*b^6*c^2*d^10*f^4 + 8*B*a^3*b^6*c^4*d^8*f^4 - 12*B*a^4*b^5*c^3*d^9*f^4 + 8*B*a^5*b^4*c^2*d^10*f^4 + 4*B*a^5*b^4*c^4*d^8*f^4

- 4*B*a^6*b^3*c^3*d^9*f^4))/(b*f^5) + (32*(c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) + 4*B^2*a^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 - 8*B^2*a*b*d^3*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 + 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^10*d^10*f^4 + 16*a^2*b^8*d^10*f^4 - 16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*d^10*f^4 + 24*b^10*c^2*d^8*f^4 + 40*a^2*b^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^9*c*d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 24*a^5*b^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9*f^4))/(b*f^4))*((((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) + 4*B^2*a^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 - 8*B^2*a*b*d^3*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 + 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(4*B^2*a^3*b^5*d^13*f^2 + 2*B^2*a^5*b^3*d^13*f^2 + 28*B^2*b^8*c^3*d^10*f^2 - 10*B^2*b^8*c^5*d^8*f^2 - 14*B^2*a*b^7*d^13*f^2 + 16*B^2*a^7*b*d^13*f^2 - 8*B^2*a^8*c*d^12*f^2 + 22*B^2*b^8*c*d^12*f^2 + 20*B^2*a*b^7*c^2*d^11*f^2 + 50*B^2*a*b^7*c^4*d^9*f^2 - 28*B^2*a^2*b^6*c*d^12*f^2 - 2*B^2*a^4*b^4*c*d^12*f^2 - 56*B^2*a^6*b^2*c*d^12*f^2 + 32*B^2*a^7*b*c^2*d^11*f^2 + 8*B^2*a^2*b^6*c^3*d^10*f^2 + 12*B^2*a^2*b^6*c^5*d^8*f^2 - 24*B^2*a^3*b^5*c^2*d^11*f^2 - 12*B^2*a^3*b^5*c^4*d^9*f^2 - 4*B^2*a^4*b^4*c^3*d^10*f^2 - 10*B^2*a^4*b^4*c^5*d^8*f^2 + 52*B^2*a^5*b^3*c^2*d^11*f^2 + 34*B^2*a^5*b^3*c^4*d^9*f^2 - 48*B^2*a^6*b^2*c^3*d^10*f^2))/(b*f^4))*((((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) + 4*B^2*a^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 - 8*B^2*a*b*d^3*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 + 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(15*B^3*a^4*b^3*d^15*f^2 - B^3*a^2*b^5*d^15*f^2 - 4*B^3*a^7*c^3*d^12*f^2 + 2*B^3*b^7*c^2*d^13*f^2 + 4*B^3*b^7*c^4*d^11*f^2 + 2*B^3*b^7*c^6*d^9*f^2 - 12*B^3*a^6*b*d^15*f^2 - 4*B^3*a^7*c*d^14*f^2 - B^3*a*b^6*c*d^14*f^2 - 27*B^3*a*b^6*c^3*d^12*f^2 - 19*B^3*a*b^6*c^5*d^10*f^2 + 7*B^3*a*b^6*c^7*d^8*f^2 - 57*B^3*a^3*b^4*c*d^14*f^2 + 64*B^3*a^5*b^2*c*d^14*f^2 + 4*B^3*a^6*b*c^2*d^13*f^2 + 16*B^3*a^6*b*c^4*d^11*f^2 + 65*B^3*a^2*b^5*c^2*d^13*f^2 + 9*B^3*a^2*b^5*c^4*d^11*f^2 - 57*B^3*a^2*b^5*c^6*d^9*f^2 + 77*B^3*a^3*b^4*c^3*d^12*f^2 + 129*B^3*a^3*b^4*c^5*d^10*f^2 - 5*B^3*a^3*b^4*c^7*d^8*f^2 - 121*B^3*a^4*b^3*c^2*d^13*f^2 - 119*B^3*a^4*b^3*c^4*d^11*f^2 + 17*B^3*a^4*b^3*c^6*d^9*f^2 + 40*B^3*a^5*b^2*c^3*d^12*f^2 - 24*B^3*a^5*b^2*c^5*d^10*f^2))/(b*f^5))*((((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) + 4*B^2*a^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 - 8*B^2*a*b*d^3*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2

+ 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (
32*(c + d*tan(e + f*x))^(1/2)*(B^4*b^6*d^16 - 2*B^4*a^6*d^16 + 12*B^4*a^6*c
^2*d^14 - 2*B^4*a^6*c^4*d^12 + 4*B^4*b^6*c^2*d^14 + 6*B^4*b^6*c^4*d^12 + 4*
B^4*b^6*c^6*d^10 + B^4*b^6*c^8*d^8 - 2*B^4*a^2*b^4*c^4*d^12 + 12*B^4*a^2*b^
4*c^6*d^10 - 2*B^4*a^2*b^4*c^8*d^8 + 8*B^4*a^3*b^3*c^3*d^13 - 48*B^4*a^3*b^
3*c^5*d^11 + 8*B^4*a^3*b^3*c^7*d^9 - 12*B^4*a^4*b^2*c^2*d^14 + 72*B^4*a^4*b
^2*c^4*d^12 - 12*B^4*a^4*b^2*c^6*d^10 + 8*B^4*a^5*b*c*d^15 - 48*B^4*a^5*b*c
^3*d^13 + 8*B^4*a^5*b*c^5*d^11))/(b*f^4))*((((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2
*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2
 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(
B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) + 4*B^2*a^2*c^3*f
^2 - 4*B^2*b^2*c^3*f^2 - 8*B^2*a*b*d^3*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*B^2*
b^2*c*d^2*f^2 + 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^
4)))^(1/2) + (64*(B^5*a^3*b^2*d^18 - B^5*a^5*d^18 - B^5*a^5*c^2*d^16 + B^5*
a^5*c^4*d^14 + B^5*a^5*c^6*d^12 - 8*B^5*a^2*b^3*c^3*d^15 - 14*B^5*a^2*b^3*c
^5*d^13 - 12*B^5*a^2*b^3*c^7*d^11 - 4*B^5*a^2*b^3*c^9*d^9 + 3*B^5*a^3*b^2*c
^2*d^16 + 9*B^5*a^3*b^2*c^4*d^14 + 13*B^5*a^3*b^2*c^6*d^12 + 6*B^5*a^3*b^2*
c^8*d^10 + 2*B^5*a^4*b*c*d^17 + B^5*a*b^4*c^2*d^16 + 4*B^5*a*b^4*c^4*d^14 +
 6*B^5*a*b^4*c^6*d^12 + 4*B^5*a*b^4*c^8*d^10 + B^5*a*b^4*c^10*d^8 - 2*B^5*a
^2*b^3*c*d^17 - 6*B^5*a^4*b*c^5*d^13 - 4*B^5*a^4*b*c^7*d^11))/(b*f^5)))*(((
(8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*
d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 +
16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4
*d^2))^(1/2) + 4*B^2*a^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 - 8*B^2*a*b*d^3*f^2 -
12*B^2*a^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 + 24*B^2*a*b*c^2*d*f^2)/(16*(a^
4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*2i - atan(((((((32*(12*C*a^2*b^9*d
^12*f^4 + 24*C*a^4*b^7*d^12*f^4 + 12*C*a^6*b^5*d^12*f^4 + 4*C*b^11*c^2*d^10
*f^4 + 4*C*b^11*c^4*d^8*f^4 - 16*C*a*b^10*c^3*d^9*f^4 - 32*C*a^3*b^8*c*d^11
*f^4 - 16*C*a^5*b^6*c*d^11*f^4 + 20*C*a^2*b^9*c^2*d^10*f^4 + 8*C*a^2*b^9*c^
4*d^8*f^4 - 32*C*a^3*b^8*c^3*d^9*f^4 + 28*C*a^4*b^7*c^2*d^10*f^4 + 4*C*a^4*
b^7*c^4*d^8*f^4 - 16*C*a^5*b^6*c^3*d^9*f^4 + 12*C*a^6*b^5*c^2*d^10*f^4 - 16
*C*a*b^10*c*d^11*f^4))/(b^3*f^5) - (32*(c + d*tan(e + f*x))^(1/2)*((((8*C^2
*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^
2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4
*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))
^(1/2) - 4*C^2*a^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 + 8*C^2*a*b*d^3*f^2 + 12*C^2
*a^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 - 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4
+ b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^12*d^10*f^4 + 16*a^2*b^10*d^10*f^4
 - 16*a^4*b^8*d^10*f^4 - 16*a^6*b^6*d^10*f^4 + 24*b^12*c^2*d^8*f^4 + 40*a^2
*b^10*c^2*d^8*f^4 + 8*a^4*b^8*c^2*d^8*f^4 - 8*a^6*b^6*c^2*d^8*f^4 + 8*a*b^1
1*c*d^9*f^4 + 24*a^3*b^9*c*d^9*f^4 + 24*a^5*b^7*c*d^9*f^4 + 8*a^7*b^5*c*d^9
*f^4))/(b^3*f^4))*((((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^
3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)
^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^
4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) - 4*C^2*a^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 +

8*C^2*a*b*d^3*f^2 + 12*C^2*a^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 - 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(4*C^2*a^3*b^7*d^13*f^2 + 2*C^2*a^5*b^5*d^13*f^2 + 28*C^2*b^10*c^3*d^10*f^2 - 10*C^2*b^10*c^5*d^8*f^2 - 14*C^2*a*b^9*d^13*f^2 - 16*C^2*a^9*b*d^13*f^2 + 8*C^2*a^10*c*d^12*f^2 + 22*C^2*b^10*c*d^12*f^2 + 20*C^2*a*b^9*c^2*d^11*f^2 + 50*C^2*a*b^9*c^4*d^9*f^2 - 28*C^2*a^2*b^8*c*d^12*f^2 - 2*C^2*a^4*b^6*c*d^12*f^2 + 56*C^2*a^8*b^2*c*d^12*f^2 - 32*C^2*a^9*b*c^2*d^11*f^2 + 8*C^2*a^2*b^8*c^3*d^10*f^2 + 4*C^2*a^2*b^8*c^5*d^8*f^2 - 24*C^2*a^3*b^7*c^2*d^11*f^2 + 4*C^2*a^3*b^7*c^4*d^9*f^2 + 12*C^2*a^4*b^6*c^3*d^10*f^2 - 10*C^2*a^4*b^6*c^5*d^8*f^2 - 12*C^2*a^5*b^5*c^2*d^11*f^2 + 18*C^2*a^5*b^5*c^4*d^9*f^2 + 16*C^2*a^6*b^4*c^3*d^10*f^2 + 8*C^2*a^6*b^4*c^5*d^8*f^2 - 64*C^2*a^7*b^3*c^2*d^11*f^2 - 32*C^2*a^7*b^3*c^4*d^9*f^2 + 48*C^2*a^8*b^2*c^3*d^10*f^2))/(b^3*f^4))*((((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) - 4*C^2*a^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 + 8*C^2*a*b*d^3*f^2 + 12*C^2*a^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 - 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(4*C^3*a^9*d^15*f^2 + C^3*a^3*b^6*d^15*f^2 + 16*C^3*a^5*b^4*d^15*f^2 - 16*C^3*a^7*b^2*d^15*f^2 + 4*C^3*a^9*c^2*d^13*f^2 - C^3*b^9*c^3*d^12*f^2 + C^3*b^9*c^5*d^10*f^2 + C^3*b^9*c^7*d^8*f^2 + C^3*a*b^8*d^15*f^2 - C^3*b^9*c*d^14*f^2 - 28*C^3*a^8*b*c*d^14*f^2 + 3*C^3*a*b^8*c^2*d^13*f^2 + 3*C^3*a*b^8*c^4*d^11*f^2 + C^3*a*b^8*c^6*d^9*f^2 - 3*C^3*a^2*b^7*c*d^14*f^2 - 58*C^3*a^4*b^5*c*d^14*f^2 + 80*C^3*a^6*b^3*c*d^14*f^2 - 28*C^3*a^8*b*c^3*d^12*f^2 - 29*C^3*a^2*b^7*c^3*d^12*f^2 - 17*C^3*a^2*b^7*c^5*d^10*f^2 + 9*C^3*a^2*b^7*c^7*d^8*f^2 + 67*C^3*a^3*b^6*c^2*d^13*f^2 + 3*C^3*a^3*b^6*c^4*d^11*f^2 - 63*C^3*a^3*b^6*c^6*d^9*f^2 + 92*C^3*a^4*b^5*c^3*d^12*f^2 + 138*C^3*a^4*b^5*c^5*d^10*f^2 - 12*C^3*a^4*b^5*c^7*d^8*f^2 - 144*C^3*a^5*b^4*c^2*d^13*f^2 - 108*C^3*a^5*b^4*c^4*d^11*f^2 + 52*C^3*a^5*b^4*c^6*d^9*f^2 - 8*C^3*a^6*b^3*c^3*d^12*f^2 - 88*C^3*a^6*b^3*c^5*d^10*f^2 + 56*C^3*a^7*b^2*c^2*d^13*f^2 + 72*C^3*a^7*b^2*c^4*d^11*f^2))/(b^3*f^5))*((((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) - 4*C^2*a^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 + 8*C^2*a*b*d^3*f^2 + 12*C^2*a^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 - 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(2*C^4*a^8*d^16 + C^4*b^8*d^16 - 12*C^4*a^8*c^2*d^14 + 2*C^4*a^8*c^4*d^12 + 4*C^4*b^8*c^2*d^14 + 6*C^4*b^8*c^4*d^12 + 4*C^4*b^8*c^6*d^10 + C^4*b^8*c^8*d^8 + 2*C^4*a^4*b^4*c^4*d^12 - 12*C^4*a^4*b^4*c^6*d^10 + 2*C^4*a^4*b^4*c^8*d^8 - 8*C^4*a^5*b^3*c^3*d^13 + 48*C^4*a^5*b^3*c^5*d^11 - 8*C^4*a^5*b^3*c^7*d^9 + 12*C^4*a^6*b^2*c^2*d^14 - 72*C^4*a^6*b^2*c^4*d^12 + 12*C^4*a^6*b^2*c^6*d^10 - 8*C^4*a^7*b*c*d^15 + 48*C^4*a^7*b*c^3*d^13 - 8*C^4*a^7*b*c^5*d^11))/(b^3*f^4))*((((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2

*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) - 4*C^
2*a^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 + 8*C^2*a*b*d^3*f^2 + 12*C^2*a^2*c*d^2*f^
2 - 12*C^2*b^2*c*d^2*f^2 - 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2
*a^2*b^2*f^4)))^(1/2)*1i - (((((32*(12*C*a^2*b^9*d^12*f^4 + 24*C*a^4*b^7*d^
12*f^4 + 12*C*a^6*b^5*d^12*f^4 + 4*C*b^11*c^2*d^10*f^4 + 4*C*b^11*c^4*d^8*f
^4 - 16*C*a*b^10*c^3*d^9*f^4 - 32*C*a^3*b^8*c*d^11*f^4 - 16*C*a^5*b^6*c*d^1
1*f^4 + 20*C*a^2*b^9*c^2*d^10*f^4 + 8*C*a^2*b^9*c^4*d^8*f^4 - 32*C*a^3*b^8*
c^3*d^9*f^4 + 28*C*a^4*b^7*c^2*d^10*f^4 + 4*C*a^4*b^7*c^4*d^8*f^4 - 16*C*a^
5*b^6*c^3*d^9*f^4 + 12*C*a^6*b^5*c^2*d^10*f^4 - 16*C*a*b^10*c*d^11*f^4))/(b
^3*f^5) + (32*(c + d*tan(e + f*x))^(1/2)*((((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*
c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2
+ 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C
^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) - 4*C^2*a^2*c^3*f^
2 + 4*C^2*b^2*c^3*f^2 + 8*C^2*a*b*d^3*f^2 + 12*C^2*a^2*c*d^2*f^2 - 12*C^2*b
^2*c*d^2*f^2 - 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4
)))^(1/2)*(16*b^12*d^10*f^4 + 16*a^2*b^10*d^10*f^4 - 16*a^4*b^8*d^10*f^4 -
16*a^6*b^6*d^10*f^4 + 24*b^12*c^2*d^8*f^4 + 40*a^2*b^10*c^2*d^8*f^4 + 8*a^4
*b^8*c^2*d^8*f^4 - 8*a^6*b^6*c^2*d^8*f^4 + 8*a*b^11*c*d^9*f^4 + 24*a^3*b^9*
c*d^9*f^4 + 24*a^5*b^7*c*d^9*f^4 + 8*a^7*b^5*c*d^9*f^4))/(b^3*f^4))*((((8*C
^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*
f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b
^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2
))^(1/2) - 4*C^2*a^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 + 8*C^2*a*b*d^3*f^2 + 12*C
^2*a^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 - 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^
4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(4*C^
2*a^3*b^7*d^13*f^2 + 2*C^2*a^5*b^5*d^13*f^2 + 28*C^2*b^10*c^3*d^10*f^2 - 10
*C^2*b^10*c^5*d^8*f^2 - 14*C^2*a*b^9*d^13*f^2 - 16*C^2*a^9*b*d^13*f^2 + 8*C
^2*a^10*c*d^12*f^2 + 22*C^2*b^10*c*d^12*f^2 + 20*C^2*a*b^9*c^2*d^11*f^2 + 5
0*C^2*a*b^9*c^4*d^9*f^2 - 28*C^2*a^2*b^8*c*d^12*f^2 - 2*C^2*a^4*b^6*c*d^12*
f^2 + 56*C^2*a^8*b^2*c*d^12*f^2 - 32*C^2*a^9*b*c^2*d^11*f^2 + 8*C^2*a^2*b^8
*c^3*d^10*f^2 + 4*C^2*a^2*b^8*c^5*d^8*f^2 - 24*C^2*a^3*b^7*c^2*d^11*f^2 + 4
*C^2*a^3*b^7*c^4*d^9*f^2 + 12*C^2*a^4*b^6*c^3*d^10*f^2 - 10*C^2*a^4*b^6*c^5
*d^8*f^2 - 12*C^2*a^5*b^5*c^2*d^11*f^2 + 18*C^2*a^5*b^5*c^4*d^9*f^2 + 16*C^
2*a^6*b^4*c^3*d^10*f^2 + 8*C^2*a^6*b^4*c^5*d^8*f^2 - 64*C^2*a^7*b^3*c^2*d^1
1*f^2 - 32*C^2*a^7*b^3*c^4*d^9*f^2 + 48*C^2*a^8*b^2*c^3*d^10*f^2))/(b^3*f^4
))*((((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*
a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*
f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C
^4*c^4*d^2))^(1/2) - 4*C^2*a^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 + 8*C^2*a*b*d^3*
f^2 + 12*C^2*a^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 - 24*C^2*a*b*c^2*d*f^2)/(
16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(4*C^3*a^9*d^15*f^2 +
C^3*a^3*b^6*d^15*f^2 + 16*C^3*a^5*b^4*d^15*f^2 - 16*C^3*a^7*b^2*d^15*f^2 +
4*C^3*a^9*c^2*d^13*f^2 - C^3*b^9*c^3*d^12*f^2 + C^3*b^9*c^5*d^10*f^2 + C^3*
b^9*c^7*d^8*f^2 + C^3*a*b^8*d^15*f^2 - C^3*b^9*c*d^14*f^2 - 28*C^3*a^8*b*c*
d^14*f^2 + 3*C^3*a*b^8*c^2*d^13*f^2 + 3*C^3*a*b^8*c^4*d^11*f^2 + C^3*a*b^8*

c^6*d^9*f^2 - 3*C^3*a^2*b^7*c*d^14*f^2 - 58*C^3*a^4*b^5*c*d^14*f^2 + 80*C^3
*a^6*b^3*c*d^14*f^2 - 28*C^3*a^8*b*c^3*d^12*f^2 - 29*C^3*a^2*b^7*c^3*d^12*f
^2 - 17*C^3*a^2*b^7*c^5*d^10*f^2 + 9*C^3*a^2*b^7*c^7*d^8*f^2 + 67*C^3*a^3*b
^6*c^2*d^13*f^2 + 3*C^3*a^3*b^6*c^4*d^11*f^2 - 63*C^3*a^3*b^6*c^6*d^9*f^2 +
92*C^3*a^4*b^5*c^3*d^12*f^2 + 138*C^3*a^4*b^5*c^5*d^10*f^2 - 12*C^3*a^4*b^
5*c^7*d^8*f^2 - 144*C^3*a^5*b^4*c^2*d^13*f^2 - 108*C^3*a^5*b^4*c^4*d^11*f^2
+ 52*C^3*a^5*b^4*c^6*d^9*f^2 - 8*C^3*a^6*b^3*c^3*d^12*f^2 - 88*C^3*a^6*b^3
*c^5*d^10*f^2 + 56*C^3*a^7*b^2*c^2*d^13*f^2 + 72*C^3*a^7*b^2*c^4*d^11*f^2))
/(b^3*f^5))*((((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2
- 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 -
(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*
d^4 + 3*C^4*c^4*d^2))^(1/2) - 4*C^2*a^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 + 8*C^2
*a*b*d^3*f^2 + 12*C^2*a^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 - 24*C^2*a*b*c^2
*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e
+ f*x))^(1/2)*(2*C^4*a^8*d^16 + C^4*b^8*d^16 - 12*C^4*a^8*c^2*d^14 + 2*C^4*
a^8*c^4*d^12 + 4*C^4*b^8*c^2*d^14 + 6*C^4*b^8*c^4*d^12 + 4*C^4*b^8*c^6*d^10
+ C^4*b^8*c^8*d^8 + 2*C^4*a^4*b^4*c^4*d^12 - 12*C^4*a^4*b^4*c^6*d^10 + 2*C
^4*a^4*b^4*c^8*d^8 - 8*C^4*a^5*b^3*c^3*d^13 + 48*C^4*a^5*b^3*c^5*d^11 - 8*C
^4*a^5*b^3*c^7*d^9 + 12*C^4*a^6*b^2*c^2*d^14 - 72*C^4*a^6*b^2*c^4*d^12 + 12
*C^4*a^6*b^2*c^6*d^10 - 8*C^4*a^7*b*c*d^15 + 48*C^4*a^7*b*c^3*d^13 - 8*C^4*
a^7*b*c^5*d^11))/(b^3*f^4))*((((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*
C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*
c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*
d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) - 4*C^2*a^2*c^3*f^2 + 4*C^2*b^2
*c^3*f^2 + 8*C^2*a*b*d^3*f^2 + 12*C^2*a^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2
- 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*1i)
/((((((32*(12*C*a^2*b^9*d^12*f^4 + 24*C*a^4*b^7*d^12*f^4 + 12*C*a^6*b^5*d^1
2*f^4 + 4*C*b^11*c^2*d^10*f^4 + 4*C*b^11*c^4*d^8*f^4 - 16*C*a*b^10*c^3*d^9*
f^4 - 32*C*a^3*b^8*c*d^11*f^4 - 16*C*a^5*b^6*c*d^11*f^4 + 20*C*a^2*b^9*c^2*
d^10*f^4 + 8*C*a^2*b^9*c^4*d^8*f^4 - 32*C*a^3*b^8*c^3*d^9*f^4 + 28*C*a^4*b^
7*c^2*d^10*f^4 + 4*C*a^4*b^7*c^4*d^8*f^4 - 16*C*a^5*b^6*c^3*d^9*f^4 + 12*C*
a^6*b^5*c^2*d^10*f^4 - 16*C*a*b^10*c*d^11*f^4))/(b^3*f^5) - (32*(c + d*tan(
e + f*x))^(1/2)*((((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*
f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2
/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*
c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) - 4*C^2*a^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 + 8
*C^2*a*b*d^3*f^2 + 12*C^2*a^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 - 24*C^2*a*b
*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^12*d^10*f
^4 + 16*a^2*b^10*d^10*f^4 - 16*a^4*b^8*d^10*f^4 - 16*a^6*b^6*d^10*f^4 + 24*
b^12*c^2*d^8*f^4 + 40*a^2*b^10*c^2*d^8*f^4 + 8*a^4*b^8*c^2*d^8*f^4 - 8*a^6*
b^6*c^2*d^8*f^4 + 8*a*b^11*c*d^9*f^4 + 24*a^3*b^9*c*d^9*f^4 + 24*a^5*b^7*c*
d^9*f^4 + 8*a^7*b^5*c*d^9*f^4))/(b^3*f^4))*((((8*C^2*a^2*c^3*f^2 - 8*C^2*b^
2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^
2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*
(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) - 4*C^2*a^2*c^3*

f^2 + 4*C^2*b^2*c^3*f^2 + 8*C^2*a*b*d^3*f^2 + 12*C^2*a^2*c*d^2*f^2 - 12*C^2
*b^2*c*d^2*f^2 - 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f
^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(4*C^2*a^3*b^7*d^13*f^2 + 2*C^
2*a^5*b^5*d^13*f^2 + 28*C^2*b^10*c^3*d^10*f^2 - 10*C^2*b^10*c^5*d^8*f^2 - 1
4*C^2*a*b^9*d^13*f^2 - 16*C^2*a^9*b*d^13*f^2 + 8*C^2*a^10*c*d^12*f^2 + 22*C
^2*b^10*c*d^12*f^2 + 20*C^2*a*b^9*c^2*d^11*f^2 + 50*C^2*a*b^9*c^4*d^9*f^2 -
28*C^2*a^2*b^8*c*d^12*f^2 - 2*C^2*a^4*b^6*c*d^12*f^2 + 56*C^2*a^8*b^2*c*d^
12*f^2 - 32*C^2*a^9*b*c^2*d^11*f^2 + 8*C^2*a^2*b^8*c^3*d^10*f^2 + 4*C^2*a^2
*b^8*c^5*d^8*f^2 - 24*C^2*a^3*b^7*c^2*d^11*f^2 + 4*C^2*a^3*b^7*c^4*d^9*f^2
+ 12*C^2*a^4*b^6*c^3*d^10*f^2 - 10*C^2*a^4*b^6*c^5*d^8*f^2 - 12*C^2*a^5*b^5
*c^2*d^11*f^2 + 18*C^2*a^5*b^5*c^4*d^9*f^2 + 16*C^2*a^6*b^4*c^3*d^10*f^2 +
8*C^2*a^6*b^4*c^5*d^8*f^2 - 64*C^2*a^7*b^3*c^2*d^11*f^2 - 32*C^2*a^7*b^3*c^
4*d^9*f^2 + 48*C^2*a^8*b^2*c^3*d^10*f^2))/(b^3*f^4))*((((8*C^2*a^2*c^3*f^2
- 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^
2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2
*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) - 4*C^
2*a^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 + 8*C^2*a*b*d^3*f^2 + 12*C^2*a^2*c*d^2*f^
2 - 12*C^2*b^2*c*d^2*f^2 - 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2
*a^2*b^2*f^4)))^(1/2) - (32*(4*C^3*a^9*d^15*f^2 + C^3*a^3*b^6*d^15*f^2 + 16
*C^3*a^5*b^4*d^15*f^2 - 16*C^3*a^7*b^2*d^15*f^2 + 4*C^3*a^9*c^2*d^13*f^2 -
C^3*b^9*c^3*d^12*f^2 + C^3*b^9*c^5*d^10*f^2 + C^3*b^9*c^7*d^8*f^2 + C^3*a*b
^8*d^15*f^2 - C^3*b^9*c*d^14*f^2 - 28*C^3*a^8*b*c*d^14*f^2 + 3*C^3*a*b^8*c^
2*d^13*f^2 + 3*C^3*a*b^8*c^4*d^11*f^2 + C^3*a*b^8*c^6*d^9*f^2 - 3*C^3*a^2*b
^7*c*d^14*f^2 - 58*C^3*a^4*b^5*c*d^14*f^2 + 80*C^3*a^6*b^3*c*d^14*f^2 - 28*
C^3*a^8*b*c^3*d^12*f^2 - 29*C^3*a^2*b^7*c^3*d^12*f^2 - 17*C^3*a^2*b^7*c^5*d
^10*f^2 + 9*C^3*a^2*b^7*c^7*d^8*f^2 + 67*C^3*a^3*b^6*c^2*d^13*f^2 + 3*C^3*a
^3*b^6*c^4*d^11*f^2 - 63*C^3*a^3*b^6*c^6*d^9*f^2 + 92*C^3*a^4*b^5*c^3*d^12*
f^2 + 138*C^3*a^4*b^5*c^5*d^10*f^2 - 12*C^3*a^4*b^5*c^7*d^8*f^2 - 144*C^3*a
^5*b^4*c^2*d^13*f^2 - 108*C^3*a^5*b^4*c^4*d^11*f^2 + 52*C^3*a^5*b^4*c^6*d^9
*f^2 - 8*C^3*a^6*b^3*c^3*d^12*f^2 - 88*C^3*a^6*b^3*c^5*d^10*f^2 + 56*C^3*a^
7*b^2*c^2*d^13*f^2 + 72*C^3*a^7*b^2*c^4*d^11*f^2))/(b^3*f^5))*((((8*C^2*a^2
*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 +
24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4
+ 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/
2) - 4*C^2*a^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 + 8*C^2*a*b*d^3*f^2 + 12*C^2*a^2
*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 - 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^
4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(2*C^4*a^8*
d^16 + C^4*b^8*d^16 - 12*C^4*a^8*c^2*d^14 + 2*C^4*a^8*c^4*d^12 + 4*C^4*b^8*
c^2*d^14 + 6*C^4*b^8*c^4*d^12 + 4*C^4*b^8*c^6*d^10 + C^4*b^8*c^8*d^8 + 2*C^
4*a^4*b^4*c^4*d^12 - 12*C^4*a^4*b^4*c^6*d^10 + 2*C^4*a^4*b^4*c^8*d^8 - 8*C^
4*a^5*b^3*c^3*d^13 + 48*C^4*a^5*b^3*c^5*d^11 - 8*C^4*a^5*b^3*c^7*d^9 + 12*C
^4*a^6*b^2*c^2*d^14 - 72*C^4*a^6*b^2*c^4*d^12 + 12*C^4*a^6*b^2*c^6*d^10 - 8
*C^4*a^7*b*c*d^15 + 48*C^4*a^7*b*c^3*d^13 - 8*C^4*a^7*b*c^5*d^11))/(b^3*f^4
))*((((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*
a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*

f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C
^4*c^4*d^2))^(1/2) - 4*C^2*a^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 + 8*C^2*a*b*d^3*
f^2 + 12*C^2*a^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 - 24*C^2*a*b*c^2*d*f^2)/(
16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (((((32*(12*C*a^2*b^9*d^12
*f^4 + 24*C*a^4*b^7*d^12*f^4 + 12*C*a^6*b^5*d^12*f^4 + 4*C*b^11*c^2*d^10*f^
4 + 4*C*b^11*c^4*d^8*f^4 - 16*C*a*b^10*c^3*d^9*f^4 - 32*C*a^3*b^8*c*d^11*f^
4 - 16*C*a^5*b^6*c*d^11*f^4 + 20*C*a^2*b^9*c^2*d^10*f^4 + 8*C*a^2*b^9*c^4*d
^8*f^4 - 32*C*a^3*b^8*c^3*d^9*f^4 + 28*C*a^4*b^7*c^2*d^10*f^4 + 4*C*a^4*b^7
*c^4*d^8*f^4 - 16*C*a^5*b^6*c^3*d^9*f^4 + 12*C*a^6*b^5*c^2*d^10*f^4 - 16*C*
a*b^10*c*d^11*f^4))/(b^3*f^5) + (32*(c + d*tan(e + f*x))^(1/2)*((((8*C^2*a^
2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 +
 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^
4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1
/2) - 4*C^2*a^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 + 8*C^2*a*b*d^3*f^2 + 12*C^2*a^
2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 - 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b
^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^12*d^10*f^4 + 16*a^2*b^10*d^10*f^4 -
16*a^4*b^8*d^10*f^4 - 16*a^6*b^6*d^10*f^4 + 24*b^12*c^2*d^8*f^4 + 40*a^2*b^
10*c^2*d^8*f^4 + 8*a^4*b^8*c^2*d^8*f^4 - 8*a^6*b^6*c^2*d^8*f^4 + 8*a*b^11*c
*d^9*f^4 + 24*a^3*b^9*c*d^9*f^4 + 24*a^5*b^7*c*d^9*f^4 + 8*a^7*b^5*c*d^9*f^
4))/(b^3*f^4))*((((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f
^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/
4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c
^2*d^4 + 3*C^4*c^4*d^2))^(1/2) - 4*C^2*a^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 + 8*
C^2*a*b*d^3*f^2 + 12*C^2*a^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 - 24*C^2*a*b*
c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan
(e + f*x))^(1/2)*(4*C^2*a^3*b^7*d^13*f^2 + 2*C^2*a^5*b^5*d^13*f^2 + 28*C^2*
b^10*c^3*d^10*f^2 - 10*C^2*b^10*c^5*d^8*f^2 - 14*C^2*a*b^9*d^13*f^2 - 16*C^
2*a^9*b*d^13*f^2 + 8*C^2*a^10*c*d^12*f^2 + 22*C^2*b^10*c*d^12*f^2 + 20*C^2*
a*b^9*c^2*d^11*f^2 + 50*C^2*a*b^9*c^4*d^9*f^2 - 28*C^2*a^2*b^8*c*d^12*f^2 -
 2*C^2*a^4*b^6*c*d^12*f^2 + 56*C^2*a^8*b^2*c*d^12*f^2 - 32*C^2*a^9*b*c^2*d^
11*f^2 + 8*C^2*a^2*b^8*c^3*d^10*f^2 + 4*C^2*a^2*b^8*c^5*d^8*f^2 - 24*C^2*a^
3*b^7*c^2*d^11*f^2 + 4*C^2*a^3*b^7*c^4*d^9*f^2 + 12*C^2*a^4*b^6*c^3*d^10*f^
2 - 10*C^2*a^4*b^6*c^5*d^8*f^2 - 12*C^2*a^5*b^5*c^2*d^11*f^2 + 18*C^2*a^5*b
^5*c^4*d^9*f^2 + 16*C^2*a^6*b^4*c^3*d^10*f^2 + 8*C^2*a^6*b^4*c^5*d^8*f^2 -
64*C^2*a^7*b^3*c^2*d^11*f^2 - 32*C^2*a^7*b^3*c^4*d^9*f^2 + 48*C^2*a^8*b^2*c
^3*d^10*f^2))/(b^3*f^4))*((((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2
*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2
*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6
 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) - 4*C^2*a^2*c^3*f^2 + 4*C^2*b^2*c^
3*f^2 + 8*C^2*a*b*d^3*f^2 + 12*C^2*a^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 - 2
4*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*
(4*C^3*a^9*d^15*f^2 + C^3*a^3*b^6*d^15*f^2 + 16*C^3*a^5*b^4*d^15*f^2 - 16*C
^3*a^7*b^2*d^15*f^2 + 4*C^3*a^9*c^2*d^13*f^2 - C^3*b^9*c^3*d^12*f^2 + C^3*b
^9*c^5*d^10*f^2 + C^3*b^9*c^7*d^8*f^2 + C^3*a*b^8*d^15*f^2 - C^3*b^9*c*d^14
*f^2 - 28*C^3*a^8*b*c*d^14*f^2 + 3*C^3*a*b^8*c^2*d^13*f^2 + 3*C^3*a*b^8*c^4

*d^11*f^2 + C^3*a*b^8*c^6*d^9*f^2 - 3*C^3*a^2*b^7*c*d^14*f^2 - 58*C^3*a^4*b ^5*c*d^14*f^2 + 80*C^3*a^6*b^3*c*d^14*f^2 - 28*C^3*a^8*b*c^3*d^12*f^2 - 29* C^3*a^2*b^7*c^3*d^12*f^2 - 17*C^3*a^2*b^7*c^5*d^10*f^2 + 9*C^3*a^2*b^7*c^7* d^8*f^2 + 67*C^3*a^3*b^6*c^2*d^13*f^2 + 3*C^3*a^3*b^6*c^4*d^11*f^2 - 63*C^3 *a^3*b^6*c^6*d^9*f^2 + 92*C^3*a^4*b^5*c^3*d^12*f^2 + 138*C^3*a^4*b^5*c^5*d^ 10*f^2 - 12*C^3*a^4*b^5*c^7*d^8*f^2 - 144*C^3*a^5*b^4*c^2*d^13*f^2 - 108*C^ 3*a^5*b^4*c^4*d^11*f^2 + 52*C^3*a^5*b^4*c^6*d^9*f^2 - 8*C^3*a^6*b^3*c^3*d^1 2*f^2 - 88*C^3*a^6*b^3*c^5*d^10*f^2 + 56*C^3*a^7*b^2*c^2*d^13*f^2 + 72*C^3* a^7*b^2*c^4*d^11*f^2))/(b^3*f^5))*((((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^ 2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) - 4*C^2*a^2*c^3*f^2 + 4*C ^2*b^2*c^3*f^2 + 8*C^2*a*b*d^3*f^2 + 12*C^2*a^2*c*d^2*f^2 - 12*C^2*b^2*c*d^ 2*f^2 - 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/ 2) + (32*(c + d*tan(e + f*x))^(1/2)*(2*C^4*a^8*d^16 + C^4*b^8*d^16 - 12*C^4 *a^8*c^2*d^14 + 2*C^4*a^8*c^4*d^12 + 4*C^4*b^8*c^2*d^14 + 6*C^4*b^8*c^4*d^1 2 + 4*C^4*b^8*c^6*d^10 + C^4*b^8*c^8*d^8 + 2*C^4*a^4*b^4*c^4*d^12 - 12*C^4* a^4*b^4*c^6*d^10 + 2*C^4*a^4*b^4*c^8*d^8 - 8*C^4*a^5*b^3*c^3*d^13 + 48*C^4* a^5*b^3*c^5*d^11 - 8*C^4*a^5*b^3*c^7*d^9 + 12*C^4*a^6*b^2*c^2*d^14 - 72*C^4 *a^6*b^2*c^4*d^12 + 12*C^4*a^6*b^2*c^6*d^10 - 8*C^4*a^7*b*c*d^15 + 48*C^4*a ^7*b*c^3*d^13 - 8*C^4*a^7*b*c^5*d^11))/(b^3*f^4))*((((8*C^2*a^2*c^3*f^2 - 8 *C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c *d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^ 2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) - 4*C^2*a ^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 + 8*C^2*a*b*d^3*f^2 + 12*C^2*a^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 - 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^ 2*b^2*f^4)))^(1/2) + (64*(C^5*a^4*b^3*d^18 + 4*C^5*a^7*c^3*d^15 + 2*C^5*a^7 *c^5*d^13 - C^5*a^6*b*d^18 + 2*C^5*a^7*c*d^17 + C^5*a^2*b^5*c^2*d^16 + 4*C^ 5*a^2*b^5*c^4*d^14 + 6*C^5*a^2*b^5*c^6*d^12 + 4*C^5*a^2*b^5*c^8*d^10 + C^5* a^2*b^5*c^10*d^8 - 8*C^5*a^3*b^4*c^3*d^15 - 12*C^5*a^3*b^4*c^5*d^13 - 8*C^5 *a^3*b^4*c^7*d^11 - 2*C^5*a^3*b^4*c^9*d^9 + 3*C^5*a^4*b^3*c^2*d^16 + C^5*a^ 4*b^3*c^4*d^14 - 3*C^5*a^4*b^3*c^6*d^12 - 2*C^5*a^4*b^3*c^8*d^10 + 12*C^5*a ^5*b^2*c^3*d^15 + 18*C^5*a^5*b^2*c^5*d^13 + 8*C^5*a^5*b^2*c^7*d^11 - 2*C^5* a^3*b^4*c*d^17 + 2*C^5*a^5*b^2*c*d^17 - 9*C^5*a^6*b*c^2*d^16 - 15*C^5*a^6*b *c^4*d^14 - 7*C^5*a^6*b*c^6*d^12))/(b^3*f^5)))*((((8*C^2*a^2*c^3*f^2 - 8*C^ 2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^ 2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f ^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) - 4*C^2*a^2* c^3*f^2 + 4*C^2*b^2*c^3*f^2 + 8*C^2*a*b*d^3*f^2 + 12*C^2*a^2*c*d^2*f^2 - 12 *C^2*b^2*c*d^2*f^2 - 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b ^2*f^4)))^(1/2)*2i - atan((((((32*(12*C*a^2*b^9*d^12*f^4 + 24*C*a^4*b^7*d^ 12*f^4 + 12*C*a^6*b^5*d^12*f^4 + 4*C*b^11*c^2*d^10*f^4 + 4*C*b^11*c^4*d^8*f ^4 - 16*C*a*b^10*c^3*d^9*f^4 - 32*C*a^3*b^8*c*d^11*f^4 - 16*C*a^5*b^6*c*d^1 1*f^4 + 20*C*a^2*b^9*c^2*d^10*f^4 + 8*C*a^2*b^9*c^4*d^8*f^4 - 32*C*a^3*b^8* c^3*d^9*f^4 + 28*C*a^4*b^7*c^2*d^10*f^4 + 4*C*a^4*b^7*c^4*d^8*f^4 - 16*C*a^

5*b^6*c^3*d^9*f^4 + 12*C*a^6*b^5*c^2*d^10*f^4 - 16*C*a*b^10*c*d^11*f^4))/(b
^3*f^5) - (32*(c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2
*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2
 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(
C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) + 4*C^2*a^2*c^3*f
^2 - 4*C^2*b^2*c^3*f^2 - 8*C^2*a*b*d^3*f^2 - 12*C^2*a^2*c*d^2*f^2 + 12*C^2*
b^2*c*d^2*f^2 + 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^
4)))^(1/2)*(16*b^12*d^10*f^4 + 16*a^2*b^10*d^10*f^4 - 16*a^4*b^8*d^10*f^4 -
 16*a^6*b^6*d^10*f^4 + 24*b^12*c^2*d^8*f^4 + 40*a^2*b^10*c^2*d^8*f^4 + 8*a^
4*b^8*c^2*d^8*f^4 - 8*a^6*b^6*c^2*d^8*f^4 + 8*a*b^11*c*d^9*f^4 + 24*a^3*b^9
*c*d^9*f^4 + 24*a^5*b^7*c*d^9*f^4 + 8*a^7*b^5*c*d^9*f^4))/(b^3*f^4))*(-(((8
*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^
2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16
*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d
^2))^(1/2) + 4*C^2*a^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 - 8*C^2*a*b*d^3*f^2 - 12
*C^2*a^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 + 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*
f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(4*
C^2*a^3*b^7*d^13*f^2 + 2*C^2*a^5*b^5*d^13*f^2 + 28*C^2*b^10*c^3*d^10*f^2 -
10*C^2*b^10*c^5*d^8*f^2 - 14*C^2*a*b^9*d^13*f^2 - 16*C^2*a^9*b*d^13*f^2 + 8
*C^2*a^10*c*d^12*f^2 + 22*C^2*b^10*c*d^12*f^2 + 20*C^2*a*b^9*c^2*d^11*f^2 +
 50*C^2*a*b^9*c^4*d^9*f^2 - 28*C^2*a^2*b^8*c*d^12*f^2 - 2*C^2*a^4*b^6*c*d^1
2*f^2 + 56*C^2*a^8*b^2*c*d^12*f^2 - 32*C^2*a^9*b*c^2*d^11*f^2 + 8*C^2*a^2*b
^8*c^3*d^10*f^2 + 4*C^2*a^2*b^8*c^5*d^8*f^2 - 24*C^2*a^3*b^7*c^2*d^11*f^2 +
 4*C^2*a^3*b^7*c^4*d^9*f^2 + 12*C^2*a^4*b^6*c^3*d^10*f^2 - 10*C^2*a^4*b^6*c
^5*d^8*f^2 - 12*C^2*a^5*b^5*c^2*d^11*f^2 + 18*C^2*a^5*b^5*c^4*d^9*f^2 + 16*
C^2*a^6*b^4*c^3*d^10*f^2 + 8*C^2*a^6*b^4*c^5*d^8*f^2 - 64*C^2*a^7*b^3*c^2*d
^11*f^2 - 32*C^2*a^7*b^3*c^4*d^9*f^2 + 48*C^2*a^8*b^2*c^3*d^10*f^2))/(b^3*f
^4))*(-(((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C
^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a
^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 +
3*C^4*c^4*d^2))^(1/2) + 4*C^2*a^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 - 8*C^2*a*b*d
^3*f^2 - 12*C^2*a^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 + 24*C^2*a*b*c^2*d*f^2
)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(4*C^3*a^9*d^15*f^2
 + C^3*a^3*b^6*d^15*f^2 + 16*C^3*a^5*b^4*d^15*f^2 - 16*C^3*a^7*b^2*d^15*f^2
 + 4*C^3*a^9*c^2*d^13*f^2 - C^3*b^9*c^3*d^12*f^2 + C^3*b^9*c^5*d^10*f^2 + C
^3*b^9*c^7*d^8*f^2 + C^3*a*b^8*d^15*f^2 - C^3*b^9*c*d^14*f^2 - 28*C^3*a^8*b
*c*d^14*f^2 + 3*C^3*a*b^8*c^2*d^13*f^2 + 3*C^3*a*b^8*c^4*d^11*f^2 + C^3*a*b
^8*c^6*d^9*f^2 - 3*C^3*a^2*b^7*c*d^14*f^2 - 58*C^3*a^4*b^5*c*d^14*f^2 + 80*
C^3*a^6*b^3*c*d^14*f^2 - 28*C^3*a^8*b*c^3*d^12*f^2 - 29*C^3*a^2*b^7*c^3*d^1
2*f^2 - 17*C^3*a^2*b^7*c^5*d^10*f^2 + 9*C^3*a^2*b^7*c^7*d^8*f^2 + 67*C^3*a^
3*b^6*c^2*d^13*f^2 + 3*C^3*a^3*b^6*c^4*d^11*f^2 - 63*C^3*a^3*b^6*c^6*d^9*f^
2 + 92*C^3*a^4*b^5*c^3*d^12*f^2 + 138*C^3*a^4*b^5*c^5*d^10*f^2 - 12*C^3*a^4
*b^5*c^7*d^8*f^2 - 144*C^3*a^5*b^4*c^2*d^13*f^2 - 108*C^3*a^5*b^4*c^4*d^11*
f^2 + 52*C^3*a^5*b^4*c^6*d^9*f^2 - 8*C^3*a^6*b^3*c^3*d^12*f^2 - 88*C^3*a^6*
b^3*c^5*d^10*f^2 + 56*C^3*a^7*b^2*c^2*d^13*f^2 + 72*C^3*a^7*b^2*c^4*d^11*f^

2))/(b^3*f^5))*(-(((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*
f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2
/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*
c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) + 4*C^2*a^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 - 8
*C^2*a*b*d^3*f^2 - 12*C^2*a^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 + 24*C^2*a*b
*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*ta
n(e + f*x))^(1/2)*(2*C^4*a^8*d^16 + C^4*b^8*d^16 - 12*C^4*a^8*c^2*d^14 + 2*
C^4*a^8*c^4*d^12 + 4*C^4*b^8*c^2*d^14 + 6*C^4*b^8*c^4*d^12 + 4*C^4*b^8*c^6*
d^10 + C^4*b^8*c^8*d^8 + 2*C^4*a^4*b^4*c^4*d^12 - 12*C^4*a^4*b^4*c^6*d^10 +
2*C^4*a^4*b^4*c^8*d^8 - 8*C^4*a^5*b^3*c^3*d^13 + 48*C^4*a^5*b^3*c^5*d^11 -
8*C^4*a^5*b^3*c^7*d^9 + 12*C^4*a^6*b^2*c^2*d^14 - 72*C^4*a^6*b^2*c^4*d^12
+ 12*C^4*a^6*b^2*c^6*d^10 - 8*C^4*a^7*b*c*d^15 + 48*C^4*a^7*b*c^3*d^13 - 8*
C^4*a^7*b*c^5*d^11))/(b^3*f^4))*(-(((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2
- 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2
*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 +
C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) + 4*C^2*a^2*c^3*f^2 - 4*C^
2*b^2*c^3*f^2 - 8*C^2*a*b*d^3*f^2 - 12*C^2*a^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2
*f^2 + 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2
)*1i - (((((32*(12*C*a^2*b^9*d^12*f^4 + 24*C*a^4*b^7*d^12*f^4 + 12*C*a^6*b^
5*d^12*f^4 + 4*C*b^11*c^2*d^10*f^4 + 4*C*b^11*c^4*d^8*f^4 - 16*C*a*b^10*c^3
*d^9*f^4 - 32*C*a^3*b^8*c*d^11*f^4 - 16*C*a^5*b^6*c*d^11*f^4 + 20*C*a^2*b^9
*c^2*d^10*f^4 + 8*C*a^2*b^9*c^4*d^8*f^4 - 32*C*a^3*b^8*c^3*d^9*f^4 + 28*C*a
^4*b^7*c^2*d^10*f^4 + 4*C*a^4*b^7*c^4*d^8*f^4 - 16*C*a^5*b^6*c^3*d^9*f^4 +
12*C*a^6*b^5*c^2*d^10*f^4 - 16*C*a*b^10*c*d^11*f^4))/(b^3*f^5) + (32*(c + d
*tan(e + f*x))^(1/2)*(-(((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*
b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*
f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 +
3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) + 4*C^2*a^2*c^3*f^2 - 4*C^2*b^2*c^3*f
^2 - 8*C^2*a*b*d^3*f^2 - 12*C^2*a^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 + 24*C
^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^12*
d^10*f^4 + 16*a^2*b^10*d^10*f^4 - 16*a^4*b^8*d^10*f^4 - 16*a^6*b^6*d^10*f^4
+ 24*b^12*c^2*d^8*f^4 + 40*a^2*b^10*c^2*d^8*f^4 + 8*a^4*b^8*c^2*d^8*f^4 -
8*a^6*b^6*c^2*d^8*f^4 + 8*a*b^11*c*d^9*f^4 + 24*a^3*b^9*c*d^9*f^4 + 24*a^5*
b^7*c*d^9*f^4 + 8*a^7*b^5*c*d^9*f^4))/(b^3*f^4))*(-(((8*C^2*a^2*c^3*f^2 - 8
*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c
*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^
2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) + 4*C^2*a
^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 - 8*C^2*a*b*d^3*f^2 - 12*C^2*a^2*c*d^2*f^2 +
12*C^2*b^2*c*d^2*f^2 + 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^
2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(4*C^2*a^3*b^7*d^13*f^2
+ 2*C^2*a^5*b^5*d^13*f^2 + 28*C^2*b^10*c^3*d^10*f^2 - 10*C^2*b^10*c^5*d^8*
f^2 - 14*C^2*a*b^9*d^13*f^2 - 16*C^2*a^9*b*d^13*f^2 + 8*C^2*a^10*c*d^12*f^2
+ 22*C^2*b^10*c*d^12*f^2 + 20*C^2*a*b^9*c^2*d^11*f^2 + 50*C^2*a*b^9*c^4*d^
9*f^2 - 28*C^2*a^2*b^8*c*d^12*f^2 - 2*C^2*a^4*b^6*c*d^12*f^2 + 56*C^2*a^8*b
^2*c*d^12*f^2 - 32*C^2*a^9*b*c^2*d^11*f^2 + 8*C^2*a^2*b^8*c^3*d^10*f^2 + 4*

C^2*a^2*b^8*c^5*d^8*f^2 - 24*C^2*a^3*b^7*c^2*d^11*f^2 + 4*C^2*a^3*b^7*c^4*d
^9*f^2 + 12*C^2*a^4*b^6*c^3*d^10*f^2 - 10*C^2*a^4*b^6*c^5*d^8*f^2 - 12*C^2*
a^5*b^5*c^2*d^11*f^2 + 18*C^2*a^5*b^5*c^4*d^9*f^2 + 16*C^2*a^6*b^4*c^3*d^10
*f^2 + 8*C^2*a^6*b^4*c^5*d^8*f^2 - 64*C^2*a^7*b^3*c^2*d^11*f^2 - 32*C^2*a^7
*b^3*c^4*d^9*f^2 + 48*C^2*a^8*b^2*c^3*d^10*f^2))/(b^3*f^4))*(-(((8*C^2*a^2*
c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 2
4*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4
+ 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2
) + 4*C^2*a^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 - 8*C^2*a*b*d^3*f^2 - 12*C^2*a^2*
c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 + 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4
*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(4*C^3*a^9*d^15*f^2 + C^3*a^3*b^6*d^15*
f^2 + 16*C^3*a^5*b^4*d^15*f^2 - 16*C^3*a^7*b^2*d^15*f^2 + 4*C^3*a^9*c^2*d^1
3*f^2 - C^3*b^9*c^3*d^12*f^2 + C^3*b^9*c^5*d^10*f^2 + C^3*b^9*c^7*d^8*f^2 +
 C^3*a*b^8*d^15*f^2 - C^3*b^9*c*d^14*f^2 - 28*C^3*a^8*b*c*d^14*f^2 + 3*C^3*
a*b^8*c^2*d^13*f^2 + 3*C^3*a*b^8*c^4*d^11*f^2 + C^3*a*b^8*c^6*d^9*f^2 - 3*C
^3*a^2*b^7*c*d^14*f^2 - 58*C^3*a^4*b^5*c*d^14*f^2 + 80*C^3*a^6*b^3*c*d^14*f
^2 - 28*C^3*a^8*b*c^3*d^12*f^2 - 29*C^3*a^2*b^7*c^3*d^12*f^2 - 17*C^3*a^2*b
^7*c^5*d^10*f^2 + 9*C^3*a^2*b^7*c^7*d^8*f^2 + 67*C^3*a^3*b^6*c^2*d^13*f^2 +
 3*C^3*a^3*b^6*c^4*d^11*f^2 - 63*C^3*a^3*b^6*c^6*d^9*f^2 + 92*C^3*a^4*b^5*c
^3*d^12*f^2 + 138*C^3*a^4*b^5*c^5*d^10*f^2 - 12*C^3*a^4*b^5*c^7*d^8*f^2 - 1
44*C^3*a^5*b^4*c^2*d^13*f^2 - 108*C^3*a^5*b^4*c^4*d^11*f^2 + 52*C^3*a^5*b^4
*c^6*d^9*f^2 - 8*C^3*a^6*b^3*c^3*d^12*f^2 - 88*C^3*a^6*b^3*c^5*d^10*f^2 + 5
6*C^3*a^7*b^2*c^2*d^13*f^2 + 72*C^3*a^7*b^2*c^4*d^11*f^2))/(b^3*f^5))*(-(((
8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d
^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 1
6*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*
d^2))^(1/2) + 4*C^2*a^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 - 8*C^2*a*b*d^3*f^2 - 1
2*C^2*a^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 + 24*C^2*a*b*c^2*d*f^2)/(16*(a^4
*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(2
*C^4*a^8*d^16 + C^4*b^8*d^16 - 12*C^4*a^8*c^2*d^14 + 2*C^4*a^8*c^4*d^12 + 4
*C^4*b^8*c^2*d^14 + 6*C^4*b^8*c^4*d^12 + 4*C^4*b^8*c^6*d^10 + C^4*b^8*c^8*d
^8 + 2*C^4*a^4*b^4*c^4*d^12 - 12*C^4*a^4*b^4*c^6*d^10 + 2*C^4*a^4*b^4*c^8*d
^8 - 8*C^4*a^5*b^3*c^3*d^13 + 48*C^4*a^5*b^3*c^5*d^11 - 8*C^4*a^5*b^3*c^7*d
^9 + 12*C^4*a^6*b^2*c^2*d^14 - 72*C^4*a^6*b^2*c^4*d^12 + 12*C^4*a^6*b^2*c^6
*d^10 - 8*C^4*a^7*b*c*d^15 + 48*C^4*a^7*b*c^3*d^13 - 8*C^4*a^7*b*c^5*d^11))
/(b^3*f^4))*(-(((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2
 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4
- (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2
*d^4 + 3*C^4*c^4*d^2))^(1/2) + 4*C^2*a^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 - 8*C^
2*a*b*d^3*f^2 - 12*C^2*a^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 + 24*C^2*a*b*c^
2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*1i)/((((((32*(12*C
*a^2*b^9*d^12*f^4 + 24*C*a^4*b^7*d^12*f^4 + 12*C*a^6*b^5*d^12*f^4 + 4*C*b^1
1*c^2*d^10*f^4 + 4*C*b^11*c^4*d^8*f^4 - 16*C*a*b^10*c^3*d^9*f^4 - 32*C*a^3*
b^8*c*d^11*f^4 - 16*C*a^5*b^6*c*d^11*f^4 + 20*C*a^2*b^9*c^2*d^10*f^4 + 8*C*
a^2*b^9*c^4*d^8*f^4 - 32*C*a^3*b^8*c^3*d^9*f^4 + 28*C*a^4*b^7*c^2*d^10*f^4

+ 4*C*a^4*b^7*c^4*d^8*f^4 - 16*C*a^5*b^6*c^3*d^9*f^4 + 12*C*a^6*b^5*c^2*d^1
0*f^4 - 16*C*a*b^10*c*d^11*f^4))/(b^3*f^5) - (32*(c + d*tan(e + f*x))^(1/2)
*(-(((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a
^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f
^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^
4*c^4*d^2))^(1/2) + 4*C^2*a^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 - 8*C^2*a*b*d^3*f
^2 - 12*C^2*a^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 + 24*C^2*a*b*c^2*d*f^2)/(1
6*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^12*d^10*f^4 + 16*a^2*b^
10*d^10*f^4 - 16*a^4*b^8*d^10*f^4 - 16*a^6*b^6*d^10*f^4 + 24*b^12*c^2*d^8*f
^4 + 40*a^2*b^10*c^2*d^8*f^4 + 8*a^4*b^8*c^2*d^8*f^4 - 8*a^6*b^6*c^2*d^8*f^
4 + 8*a*b^11*c*d^9*f^4 + 24*a^3*b^9*c*d^9*f^4 + 24*a^5*b^7*c*d^9*f^4 + 8*a^
7*b^5*c*d^9*f^4))/(b^3*f^4))*(-(((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 1
6*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*
b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^
4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) + 4*C^2*a^2*c^3*f^2 - 4*C^2*b
^2*c^3*f^2 - 8*C^2*a*b*d^3*f^2 - 12*C^2*a^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^
2 + 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) -
(32*(c + d*tan(e + f*x))^(1/2)*(4*C^2*a^3*b^7*d^13*f^2 + 2*C^2*a^5*b^5*d^1
3*f^2 + 28*C^2*b^10*c^3*d^10*f^2 - 10*C^2*b^10*c^5*d^8*f^2 - 14*C^2*a*b^9*d
^13*f^2 - 16*C^2*a^9*b*d^13*f^2 + 8*C^2*a^10*c*d^12*f^2 + 22*C^2*b^10*c*d^1
2*f^2 + 20*C^2*a*b^9*c^2*d^11*f^2 + 50*C^2*a*b^9*c^4*d^9*f^2 - 28*C^2*a^2*b
^8*c*d^12*f^2 - 2*C^2*a^4*b^6*c*d^12*f^2 + 56*C^2*a^8*b^2*c*d^12*f^2 - 32*C
^2*a^9*b*c^2*d^11*f^2 + 8*C^2*a^2*b^8*c^3*d^10*f^2 + 4*C^2*a^2*b^8*c^5*d^8*
f^2 - 24*C^2*a^3*b^7*c^2*d^11*f^2 + 4*C^2*a^3*b^7*c^4*d^9*f^2 + 12*C^2*a^4*
b^6*c^3*d^10*f^2 - 10*C^2*a^4*b^6*c^5*d^8*f^2 - 12*C^2*a^5*b^5*c^2*d^11*f^2
+ 18*C^2*a^5*b^5*c^4*d^9*f^2 + 16*C^2*a^6*b^4*c^3*d^10*f^2 + 8*C^2*a^6*b^4
*c^5*d^8*f^2 - 64*C^2*a^7*b^3*c^2*d^11*f^2 - 32*C^2*a^7*b^3*c^4*d^9*f^2 + 4
8*C^2*a^8*b^2*c^3*d^10*f^2))/(b^3*f^4))*(-(((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*
c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2
+ 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C
^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) + 4*C^2*a^2*c^3*f^
2 - 4*C^2*b^2*c^3*f^2 - 8*C^2*a*b*d^3*f^2 - 12*C^2*a^2*c*d^2*f^2 + 12*C^2*b
^2*c*d^2*f^2 + 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4
)))^(1/2) - (32*(4*C^3*a^9*d^15*f^2 + C^3*a^3*b^6*d^15*f^2 + 16*C^3*a^5*b^4
*d^15*f^2 - 16*C^3*a^7*b^2*d^15*f^2 + 4*C^3*a^9*c^2*d^13*f^2 - C^3*b^9*c^3*
d^12*f^2 + C^3*b^9*c^5*d^10*f^2 + C^3*b^9*c^7*d^8*f^2 + C^3*a*b^8*d^15*f^2
- C^3*b^9*c*d^14*f^2 - 28*C^3*a^8*b*c*d^14*f^2 + 3*C^3*a*b^8*c^2*d^13*f^2 +
3*C^3*a*b^8*c^4*d^11*f^2 + C^3*a*b^8*c^6*d^9*f^2 - 3*C^3*a^2*b^7*c*d^14*f^
2 - 58*C^3*a^4*b^5*c*d^14*f^2 + 80*C^3*a^6*b^3*c*d^14*f^2 - 28*C^3*a^8*b*c^
3*d^12*f^2 - 29*C^3*a^2*b^7*c^3*d^12*f^2 - 17*C^3*a^2*b^7*c^5*d^10*f^2 + 9*
C^3*a^2*b^7*c^7*d^8*f^2 + 67*C^3*a^3*b^6*c^2*d^13*f^2 + 3*C^3*a^3*b^6*c^4*d
^11*f^2 - 63*C^3*a^3*b^6*c^6*d^9*f^2 + 92*C^3*a^4*b^5*c^3*d^12*f^2 + 138*C^
3*a^4*b^5*c^5*d^10*f^2 - 12*C^3*a^4*b^5*c^7*d^8*f^2 - 144*C^3*a^5*b^4*c^2*d
^13*f^2 - 108*C^3*a^5*b^4*c^4*d^11*f^2 + 52*C^3*a^5*b^4*c^6*d^9*f^2 - 8*C^3
*a^6*b^3*c^3*d^12*f^2 - 88*C^3*a^6*b^3*c^5*d^10*f^2 + 56*C^3*a^7*b^2*c^2*d^

13*f^2 + 72*C^3*a^7*b^2*c^4*d^11*f^2))/(b^3*f^5))*(-(((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) + 4*C^2*a^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 - 8*C^2*a*b*d^3*f^2 - 12*C^2*a^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 + 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(2*C^4*a^8*d^16 + C^4*b^8*d^16 - 12*C^4*a^8*c^2*d^14 + 2*C^4*a^8*c^4*d^12 + 4*C^4*b^8*c^2*d^14 + 6*C^4*b^8*c^4*d^12 + 4*C^4*b^8*c^6*d^10 + C^4*b^8*c^8*d^8 + 2*C^4*a^4*b^4*c^4*d^12 - 12*C^4*a^4*b^4*c^6*d^10 + 2*C^4*a^4*b^4*c^8*d^8 - 8*C^4*a^5*b^3*c^3*d^13 + 48*C^4*a^5*b^3*c^5*d^11 - 8*C^4*a^5*b^3*c^7*d^9 + 12*C^4*a^6*b^2*c^2*d^14 - 72*C^4*a^6*b^2*c^4*d^12 + 12*C^4*a^6*b^2*c^6*d^10 - 8*C^4*a^7*b*c*d^15 + 48*C^4*a^7*b*c^3*d^13 - 8*C^4*a^7*b*c^5*d^11))/(b^3*f^4))*(-(((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) + 4*C^2*a^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 - 8*C^2*a*b*d^3*f^2 - 12*C^2*a^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 + 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (((((32*(12*C*a^2*b^9*d^12*f^4 + 24*C*a^4*b^7*d^12*f^4 + 12*C*a^6*b^5*d^12*f^4 + 4*C*b^11*c^2*d^10*f^4 + 4*C*b^11*c^4*d^8*f^4 - 16*C*a*b^10*c^3*d^9*f^4 - 32*C*a^3*b^8*c*d^11*f^4 - 16*C*a^5*b^6*c*d^11*f^4 + 20*C*a^2*b^9*c^2*d^10*f^4 + 8*C*a^2*b^9*c^4*d^8*f^4 - 32*C*a^3*b^8*c^3*d^9*f^4 + 28*C*a^4*b^7*c^2*d^10*f^4 + 4*C*a^4*b^7*c^4*d^8*f^4 - 16*C*a^5*b^6*c^3*d^9*f^4 + 12*C*a^6*b^5*c^2*d^10*f^4 - 16*C*a*b^10*c*d^11*f^4))/(b^3*f^5) + (32*(c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) + 4*C^2*a^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 - 8*C^2*a*b*d^3*f^2 - 12*C^2*a^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 + 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^12*d^10*f^4 + 16*a^2*b^10*d^10*f^4 - 16*a^4*b^8*d^10*f^4 - 16*a^6*b^6*d^10*f^4 + 24*b^12*c^2*d^8*f^4 + 40*a^2*b^10*c^2*d^8*f^4 + 8*a^4*b^8*c^2*d^8*f^4 - 8*a^6*b^6*c^2*d^8*f^4 + 8*a*b^11*c*d^9*f^4 + 24*a^3*b^9*c*d^9*f^4 + 24*a^5*b^7*c*d^9*f^4 + 8*a^7*b^5*c*d^9*f^4))/(b^3*f^4))*(-(((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) + 4*C^2*a^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 - 8*C^2*a*b*d^3*f^2 - 12*C^2*a^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 + 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(4*C^2*a^3*b^7*d^13*f^2 + 2*C^2*a^5*b^5*d^13*f^2 + 28*C^2*b^10*c^3*d^10*f^2 - 10*C^2*b^10*c^5*d^8*f^2 - 14*C^2*a*b^9*d^13*f^2 - 16*C^2*a^9*b*d^13*f^2 + 8*C^2*a^10*c*d^12*f^2 + 22*C^2*b^10*c*d^12*f^2 + 20*C^2*a*b^9*c^2*d^11*f^2 + 50*C^2*a*b^9*c^4*d^9*f^2 - 28*C^2*a^2*b^8*c*d^12*f^2 - 2*C^2*a^4*b^6*c*d^12*f^2 + 56*C^2*a^8*b^2*c*d^12*f^2 - 32*C^2*a^9*b*c^2*d^11*f^2 +

8*C^2*a^2*b^8*c^3*d^10*f^2 + 4*C^2*a^2*b^8*c^5*d^8*f^2 - 24*C^2*a^3*b^7*c^
2*d^11*f^2 + 4*C^2*a^3*b^7*c^4*d^9*f^2 + 12*C^2*a^4*b^6*c^3*d^10*f^2 - 10*C
^2*a^4*b^6*c^5*d^8*f^2 - 12*C^2*a^5*b^5*c^2*d^11*f^2 + 18*C^2*a^5*b^5*c^4*d
^9*f^2 + 16*C^2*a^6*b^4*c^3*d^10*f^2 + 8*C^2*a^6*b^4*c^5*d^8*f^2 - 64*C^2*a
^7*b^3*c^2*d^11*f^2 - 32*C^2*a^7*b^3*c^4*d^9*f^2 + 48*C^2*a^8*b^2*c^3*d^10*
f^2))/(b^3*f^4))*(-(((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^
3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)
^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^
4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) + 4*C^2*a^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 -
8*C^2*a*b*d^3*f^2 - 12*C^2*a^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 + 24*C^2*a
*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(4*C^3*
a^9*d^15*f^2 + C^3*a^3*b^6*d^15*f^2 + 16*C^3*a^5*b^4*d^15*f^2 - 16*C^3*a^7*
b^2*d^15*f^2 + 4*C^3*a^9*c^2*d^13*f^2 - C^3*b^9*c^3*d^12*f^2 + C^3*b^9*c^5*
d^10*f^2 + C^3*b^9*c^7*d^8*f^2 + C^3*a*b^8*d^15*f^2 - C^3*b^9*c*d^14*f^2 -
28*C^3*a^8*b*c*d^14*f^2 + 3*C^3*a*b^8*c^2*d^13*f^2 + 3*C^3*a*b^8*c^4*d^11*f
^2 + C^3*a*b^8*c^6*d^9*f^2 - 3*C^3*a^2*b^7*c*d^14*f^2 - 58*C^3*a^4*b^5*c*d^
14*f^2 + 80*C^3*a^6*b^3*c*d^14*f^2 - 28*C^3*a^8*b*c^3*d^12*f^2 - 29*C^3*a^2
*b^7*c^3*d^12*f^2 - 17*C^3*a^2*b^7*c^5*d^10*f^2 + 9*C^3*a^2*b^7*c^7*d^8*f^2
+ 67*C^3*a^3*b^6*c^2*d^13*f^2 + 3*C^3*a^3*b^6*c^4*d^11*f^2 - 63*C^3*a^3*b^
6*c^6*d^9*f^2 + 92*C^3*a^4*b^5*c^3*d^12*f^2 + 138*C^3*a^4*b^5*c^5*d^10*f^2
- 12*C^3*a^4*b^5*c^7*d^8*f^2 - 144*C^3*a^5*b^4*c^2*d^13*f^2 - 108*C^3*a^5*b
^4*c^4*d^11*f^2 + 52*C^3*a^5*b^4*c^6*d^9*f^2 - 8*C^3*a^6*b^3*c^3*d^12*f^2 -
88*C^3*a^6*b^3*c^5*d^10*f^2 + 56*C^3*a^7*b^2*c^2*d^13*f^2 + 72*C^3*a^7*b^2
*c^4*d^11*f^2))/(b^3*f^5))*(-(((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*
C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*
c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*
d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) + 4*C^2*a^2*c^3*f^2 - 4*C^2*b^2
*c^3*f^2 - 8*C^2*a*b*d^3*f^2 - 12*C^2*a^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2
+ 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (
32*(c + d*tan(e + f*x))^(1/2)*(2*C^4*a^8*d^16 + C^4*b^8*d^16 - 12*C^4*a^8*c
^2*d^14 + 2*C^4*a^8*c^4*d^12 + 4*C^4*b^8*c^2*d^14 + 6*C^4*b^8*c^4*d^12 + 4*
C^4*b^8*c^6*d^10 + C^4*b^8*c^8*d^8 + 2*C^4*a^4*b^4*c^4*d^12 - 12*C^4*a^4*b^
4*c^6*d^10 + 2*C^4*a^4*b^4*c^8*d^8 - 8*C^4*a^5*b^3*c^3*d^13 + 48*C^4*a^5*b^
3*c^5*d^11 - 8*C^4*a^5*b^3*c^7*d^9 + 12*C^4*a^6*b^2*c^2*d^14 - 72*C^4*a^6*b
^2*c^4*d^12 + 12*C^4*a^6*b^2*c^6*d^10 - 8*C^4*a^7*b*c*d^15 + 48*C^4*a^7*b*c
^3*d^13 - 8*C^4*a^7*b*c^5*d^11))/(b^3*f^4))*(-(((8*C^2*a^2*c^3*f^2 - 8*C^2*
b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*
f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4
)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) + 4*C^2*a^2*c^
3*f^2 - 4*C^2*b^2*c^3*f^2 - 8*C^2*a*b*d^3*f^2 - 12*C^2*a^2*c*d^2*f^2 + 12*C
^2*b^2*c*d^2*f^2 + 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2
f^4)))^(1/2) + (64(C^5*a^4*b^3*d^18 + 4*C^5*a^7*c^3*d^15 + 2*C^5*a^7*c^5*
d^13 - C^5*a^6*b*d^18 + 2*C^5*a^7*c*d^17 + C^5*a^2*b^5*c^2*d^16 + 4*C^5*a^2
*b^5*c^4*d^14 + 6*C^5*a^2*b^5*c^6*d^12 + 4*C^5*a^2*b^5*c^8*d^10 + C^5*a^2*b
^5*c^10*d^8 - 8*C^5*a^3*b^4*c^3*d^15 - 12*C^5*a^3*b^4*c^5*d^13 - 8*C^5*a^3*

b^4*c^7*d^11 - 2*C^5*a^3*b^4*c^9*d^9 + 3*C^5*a^4*b^3*c^2*d^16 + C^5*a^4*b^3
*c^4*d^14 - 3*C^5*a^4*b^3*c^6*d^12 - 2*C^5*a^4*b^3*c^8*d^10 + 12*C^5*a^5*b^
2*c^3*d^15 + 18*C^5*a^5*b^2*c^5*d^13 + 8*C^5*a^5*b^2*c^7*d^11 - 2*C^5*a^3*b
^4*c*d^17 + 2*C^5*a^5*b^2*c*d^17 - 9*C^5*a^6*b*c^2*d^16 - 15*C^5*a^6*b*c^4*
d^14 - 7*C^5*a^6*b*c^6*d^12))/(b^3*f^5)))*(-(((8*C^2*a^2*c^3*f^2 - 8*C^2*b^
2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^
2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*
(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) + 4*C^2*a^2*c^3*
f^2 - 4*C^2*b^2*c^3*f^2 - 8*C^2*a*b*d^3*f^2 - 12*C^2*a^2*c*d^2*f^2 + 12*C^2
*b^2*c*d^2*f^2 + 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f
^4)))^(1/2)*2i + (2*C*(c + d*tan(e + f*x))^(3/2))/(3*b*f) - (atan((((-(b^7*
f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2)*(B^2*a^5*d^3 - B^2*a^2*b^3*c^3 - 3*B^2*a
^4*b*c*d^2 + 3*B^2*a^3*b^2*c^2*d))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(B
^4*b^6*d^16 - 2*B^4*a^6*d^16 + 12*B^4*a^6*c^2*d^14 - 2*B^4*a^6*c^4*d^12 + 4
*B^4*b^6*c^2*d^14 + 6*B^4*b^6*c^4*d^12 + 4*B^4*b^6*c^6*d^10 + B^4*b^6*c^8*d
^8 - 2*B^4*a^2*b^4*c^4*d^12 + 12*B^4*a^2*b^4*c^6*d^10 - 2*B^4*a^2*b^4*c^8*d
^8 + 8*B^4*a^3*b^3*c^3*d^13 - 48*B^4*a^3*b^3*c^5*d^11 + 8*B^4*a^3*b^3*c^7*d
^9 - 12*B^4*a^4*b^2*c^2*d^14 + 72*B^4*a^4*b^2*c^4*d^12 - 12*B^4*a^4*b^2*c^6
*d^10 + 8*B^4*a^5*b*c*d^15 - 48*B^4*a^5*b*c^3*d^13 + 8*B^4*a^5*b*c^5*d^11))
/(b*f^4) - ((-(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2)*(B^2*a^5*d^3 - B^2*a^
2*b^3*c^3 - 3*B^2*a^4*b*c*d^2 + 3*B^2*a^3*b^2*c^2*d))^(1/2)*((32*(15*B^3*a^
4*b^3*d^15*f^2 - B^3*a^2*b^5*d^15*f^2 - 4*B^3*a^7*c^3*d^12*f^2 + 2*B^3*b^7*
c^2*d^13*f^2 + 4*B^3*b^7*c^4*d^11*f^2 + 2*B^3*b^7*c^6*d^9*f^2 - 12*B^3*a^6*
b*d^15*f^2 - 4*B^3*a^7*c*d^14*f^2 - B^3*a*b^6*c*d^14*f^2 - 27*B^3*a*b^6*c^3
*d^12*f^2 - 19*B^3*a*b^6*c^5*d^10*f^2 + 7*B^3*a*b^6*c^7*d^8*f^2 - 57*B^3*a^
3*b^4*c*d^14*f^2 + 64*B^3*a^5*b^2*c*d^14*f^2 + 4*B^3*a^6*b*c^2*d^13*f^2 + 1
6*B^3*a^6*b*c^4*d^11*f^2 + 65*B^3*a^2*b^5*c^2*d^13*f^2 + 9*B^3*a^2*b^5*c^4*
d^11*f^2 - 57*B^3*a^2*b^5*c^6*d^9*f^2 + 77*B^3*a^3*b^4*c^3*d^12*f^2 + 129*B
^3*a^3*b^4*c^5*d^10*f^2 - 5*B^3*a^3*b^4*c^7*d^8*f^2 - 121*B^3*a^4*b^3*c^2*d
^13*f^2 - 119*B^3*a^4*b^3*c^4*d^11*f^2 + 17*B^3*a^4*b^3*c^6*d^9*f^2 + 40*B^
3*a^5*b^2*c^3*d^12*f^2 - 24*B^3*a^5*b^2*c^5*d^10*f^2))/(b*f^5) + ((-(b^7*f^
2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2)*(B^2*a^5*d^3 - B^2*a^2*b^3*c^3 - 3*B^2*a^4
*b*c*d^2 + 3*B^2*a^3*b^2*c^2*d))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(4*B
^2*a^3*b^5*d^13*f^2 + 2*B^2*a^5*b^3*d^13*f^2 + 28*B^2*b^8*c^3*d^10*f^2 - 10
*B^2*b^8*c^5*d^8*f^2 - 14*B^2*a*b^7*d^13*f^2 + 16*B^2*a^7*b*d^13*f^2 - 8*B^
2*a^8*c*d^12*f^2 + 22*B^2*b^8*c*d^12*f^2 + 20*B^2*a*b^7*c^2*d^11*f^2 + 50*B
^2*a*b^7*c^4*d^9*f^2 - 28*B^2*a^2*b^6*c*d^12*f^2 - 2*B^2*a^4*b^4*c*d^12*f^2
 - 56*B^2*a^6*b^2*c*d^12*f^2 + 32*B^2*a^7*b*c^2*d^11*f^2 + 8*B^2*a^2*b^6*c^
3*d^10*f^2 + 12*B^2*a^2*b^6*c^5*d^8*f^2 - 24*B^2*a^3*b^5*c^2*d^11*f^2 - 12*
B^2*a^3*b^5*c^4*d^9*f^2 - 4*B^2*a^4*b^4*c^3*d^10*f^2 - 10*B^2*a^4*b^4*c^5*d
^8*f^2 + 52*B^2*a^5*b^3*c^2*d^11*f^2 + 34*B^2*a^5*b^3*c^4*d^9*f^2 - 48*B^2*
a^6*b^2*c^3*d^10*f^2))/(b*f^4) + ((-(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2)
*(B^2*a^5*d^3 - B^2*a^2*b^3*c^3 - 3*B^2*a^4*b*c*d^2 + 3*B^2*a^3*b^2*c^2*d))
^(1/2)*((32*(4*B*a*b^8*d^12*f^4 - 4*B*b^9*c*d^11*f^4 + 8*B*a^3*b^6*d^12*f^4
 + 4*B*a^5*b^4*d^12*f^4 - 4*B*b^9*c^3*d^9*f^4 + 8*B*a*b^8*c^2*d^10*f^4 + 4*

B*a*b^8*c^4*d^8*f^4 - 12*B*a^2*b^7*c*d^11*f^4 - 12*B*a^4*b^5*c*d^11*f^4 - 4
*B*a^6*b^3*c*d^11*f^4 - 12*B*a^2*b^7*c^3*d^9*f^4 + 16*B*a^3*b^6*c^2*d^10*f^
4 + 8*B*a^3*b^6*c^4*d^8*f^4 - 12*B*a^4*b^5*c^3*d^9*f^4 + 8*B*a^5*b^4*c^2*d^
10*f^4 + 4*B*a^5*b^4*c^4*d^8*f^4 - 4*B*a^6*b^3*c^3*d^9*f^4))/(b*f^5) - (32*
(-(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2)*(B^2*a^5*d^3 - B^2*a^2*b^3*c^3 -
3*B^2*a^4*b*c*d^2 + 3*B^2*a^3*b^2*c^2*d))^(1/2)*(c + d*tan(e + f*x))^(1/2)*
(16*b^10*d^10*f^4 + 16*a^2*b^8*d^10*f^4 - 16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*
d^10*f^4 + 24*b^10*c^2*d^8*f^4 + 40*a^2*b^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8
*f^4 - 8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^9*c*d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 2
4*a^5*b^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9*f^4))/(b^4*f^6*(a^2 + b^2)^2)))/(b^3*
f^2*(a^2 + b^2)^2)))/(b^3*f^2*(a^2 + b^2)^2)))/(b^3*f^2*(a^2 + b^2)^2))*1i)
/(b^3*f^2*(a^2 + b^2)^2) + ((-(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2)*(B^2*
a^5*d^3 - B^2*a^2*b^3*c^3 - 3*B^2*a^4*b*c*d^2 + 3*B^2*a^3*b^2*c^2*d))^(1/2)
((32(c + d*tan(e + f*x))^(1/2)*(B^4*b^6*d^16 - 2*B^4*a^6*d^16 + 12*B^4*a^
6*c^2*d^14 - 2*B^4*a^6*c^4*d^12 + 4*B^4*b^6*c^2*d^14 + 6*B^4*b^6*c^4*d^12 +
4*B^4*b^6*c^6*d^10 + B^4*b^6*c^8*d^8 - 2*B^4*a^2*b^4*c^4*d^12 + 12*B^4*a^2
*b^4*c^6*d^10 - 2*B^4*a^2*b^4*c^8*d^8 + 8*B^4*a^3*b^3*c^3*d^13 - 48*B^4*a^3
*b^3*c^5*d^11 + 8*B^4*a^3*b^3*c^7*d^9 - 12*B^4*a^4*b^2*c^2*d^14 + 72*B^4*a^
4*b^2*c^4*d^12 - 12*B^4*a^4*b^2*c^6*d^10 + 8*B^4*a^5*b*c*d^15 - 48*B^4*a^5*
b*c^3*d^13 + 8*B^4*a^5*b*c^5*d^11))/(b*f^4) + ((-(b^7*f^2 + 2*a^2*b^5*f^2 +
a^4*b^3*f^2)*(B^2*a^5*d^3 - B^2*a^2*b^3*c^3 - 3*B^2*a^4*b*c*d^2 + 3*B^2*a^
3*b^2*c^2*d))^(1/2)*((32*(15*B^3*a^4*b^3*d^15*f^2 - B^3*a^2*b^5*d^15*f^2 -
4*B^3*a^7*c^3*d^12*f^2 + 2*B^3*b^7*c^2*d^13*f^2 + 4*B^3*b^7*c^4*d^11*f^2 +
2*B^3*b^7*c^6*d^9*f^2 - 12*B^3*a^6*b*d^15*f^2 - 4*B^3*a^7*c*d^14*f^2 - B^3*
a*b^6*c*d^14*f^2 - 27*B^3*a*b^6*c^3*d^12*f^2 - 19*B^3*a*b^6*c^5*d^10*f^2 +
7*B^3*a*b^6*c^7*d^8*f^2 - 57*B^3*a^3*b^4*c*d^14*f^2 + 64*B^3*a^5*b^2*c*d^14
*f^2 + 4*B^3*a^6*b*c^2*d^13*f^2 + 16*B^3*a^6*b*c^4*d^11*f^2 + 65*B^3*a^2*b^
5*c^2*d^13*f^2 + 9*B^3*a^2*b^5*c^4*d^11*f^2 - 57*B^3*a^2*b^5*c^6*d^9*f^2 +
77*B^3*a^3*b^4*c^3*d^12*f^2 + 129*B^3*a^3*b^4*c^5*d^10*f^2 - 5*B^3*a^3*b^4*
c^7*d^8*f^2 - 121*B^3*a^4*b^3*c^2*d^13*f^2 - 119*B^3*a^4*b^3*c^4*d^11*f^2 +
17*B^3*a^4*b^3*c^6*d^9*f^2 + 40*B^3*a^5*b^2*c^3*d^12*f^2 - 24*B^3*a^5*b^2*
c^5*d^10*f^2))/(b*f^5) - ((-(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2)*(B^2*a^
5*d^3 - B^2*a^2*b^3*c^3 - 3*B^2*a^4*b*c*d^2 + 3*B^2*a^3*b^2*c^2*d))^(1/2)*(
(32*(c + d*tan(e + f*x))^(1/2)*(4*B^2*a^3*b^5*d^13*f^2 + 2*B^2*a^5*b^3*d^13
*f^2 + 28*B^2*b^8*c^3*d^10*f^2 - 10*B^2*b^8*c^5*d^8*f^2 - 14*B^2*a*b^7*d^13
*f^2 + 16*B^2*a^7*b*d^13*f^2 - 8*B^2*a^8*c*d^12*f^2 + 22*B^2*b^8*c*d^12*f^2
+ 20*B^2*a*b^7*c^2*d^11*f^2 + 50*B^2*a*b^7*c^4*d^9*f^2 - 28*B^2*a^2*b^6*c*
d^12*f^2 - 2*B^2*a^4*b^4*c*d^12*f^2 - 56*B^2*a^6*b^2*c*d^12*f^2 + 32*B^2*a^
7*b*c^2*d^11*f^2 + 8*B^2*a^2*b^6*c^3*d^10*f^2 + 12*B^2*a^2*b^6*c^5*d^8*f^2
- 24*B^2*a^3*b^5*c^2*d^11*f^2 - 12*B^2*a^3*b^5*c^4*d^9*f^2 - 4*B^2*a^4*b^4*
c^3*d^10*f^2 - 10*B^2*a^4*b^4*c^5*d^8*f^2 + 52*B^2*a^5*b^3*c^2*d^11*f^2 + 3
4*B^2*a^5*b^3*c^4*d^9*f^2 - 48*B^2*a^6*b^2*c^3*d^10*f^2))/(b*f^4) - ((-(b^7
*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2)*(B^2*a^5*d^3 - B^2*a^2*b^3*c^3 - 3*B^2*
a^4*b*c*d^2 + 3*B^2*a^3*b^2*c^2*d))^(1/2)*((32*(4*B*a*b^8*d^12*f^4 - 4*B*b^
9*c*d^11*f^4 + 8*B*a^3*b^6*d^12*f^4 + 4*B*a^5*b^4*d^12*f^4 - 4*B*b^9*c^3*d^

9*f^4 + 8*B*a*b^8*c^2*d^10*f^4 + 4*B*a*b^8*c^4*d^8*f^4 - 12*B*a^2*b^7*c*d^1
1*f^4 - 12*B*a^4*b^5*c*d^11*f^4 - 4*B*a^6*b^3*c*d^11*f^4 - 12*B*a^2*b^7*c^3
*d^9*f^4 + 16*B*a^3*b^6*c^2*d^10*f^4 + 8*B*a^3*b^6*c^4*d^8*f^4 - 12*B*a^4*b
^5*c^3*d^9*f^4 + 8*B*a^5*b^4*c^2*d^10*f^4 + 4*B*a^5*b^4*c^4*d^8*f^4 - 4*B*a
^6*b^3*c^3*d^9*f^4))/(b*f^5) + (32*(-(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2
)*(B^2*a^5*d^3 - B^2*a^2*b^3*c^3 - 3*B^2*a^4*b*c*d^2 + 3*B^2*a^3*b^2*c^2*d)
)^(1/2)*(c + d*tan(e + f*x))^(1/2)*(16*b^10*d^10*f^4 + 16*a^2*b^8*d^10*f^4
- 16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*d^10*f^4 + 24*b^10*c^2*d^8*f^4 + 40*a^2*
b^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^9*c
*d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 24*a^5*b^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9*f^
4))/(b^4*f^6*(a^2 + b^2)^2)))/(b^3*f^2*(a^2 + b^2)^2)))/(b^3*f^2*(a^2 + b^2
)^2)))/(b^3*f^2*(a^2 + b^2)^2))*1i)/(b^3*f^2*(a^2 + b^2)^2))/((64*(B^5*a^3*
b^2*d^18 - B^5*a^5*d^18 - B^5*a^5*c^2*d^16 + B^5*a^5*c^4*d^14 + B^5*a^5*c^6
*d^12 - 8*B^5*a^2*b^3*c^3*d^15 - 14*B^5*a^2*b^3*c^5*d^13 - 12*B^5*a^2*b^3*c
^7*d^11 - 4*B^5*a^2*b^3*c^9*d^9 + 3*B^5*a^3*b^2*c^2*d^16 + 9*B^5*a^3*b^2*c^
4*d^14 + 13*B^5*a^3*b^2*c^6*d^12 + 6*B^5*a^3*b^2*c^8*d^10 + 2*B^5*a^4*b*c*d
^17 + B^5*a*b^4*c^2*d^16 + 4*B^5*a*b^4*c^4*d^14 + 6*B^5*a*b^4*c^6*d^12 + 4*
B^5*a*b^4*c^8*d^10 + B^5*a*b^4*c^10*d^8 - 2*B^5*a^2*b^3*c*d^17 - 6*B^5*a^4*
b*c^5*d^13 - 4*B^5*a^4*b*c^7*d^11))/(b*f^5) - ((-(b^7*f^2 + 2*a^2*b^5*f^2 +
a^4*b^3*f^2)*(B^2*a^5*d^3 - B^2*a^2*b^3*c^3 - 3*B^2*a^4*b*c*d^2 + 3*B^2*a^
3*b^2*c^2*d))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(B^4*b^6*d^16 - 2*B^4*a
^6*d^16 + 12*B^4*a^6*c^2*d^14 - 2*B^4*a^6*c^4*d^12 + 4*B^4*b^6*c^2*d^14 + 6
*B^4*b^6*c^4*d^12 + 4*B^4*b^6*c^6*d^10 + B^4*b^6*c^8*d^8 - 2*B^4*a^2*b^4*c^
4*d^12 + 12*B^4*a^2*b^4*c^6*d^10 - 2*B^4*a^2*b^4*c^8*d^8 + 8*B^4*a^3*b^3*c^
3*d^13 - 48*B^4*a^3*b^3*c^5*d^11 + 8*B^4*a^3*b^3*c^7*d^9 - 12*B^4*a^4*b^2*c
^2*d^14 + 72*B^4*a^4*b^2*c^4*d^12 - 12*B^4*a^4*b^2*c^6*d^10 + 8*B^4*a^5*b*c
*d^15 - 48*B^4*a^5*b*c^3*d^13 + 8*B^4*a^5*b*c^5*d^11))/(b*f^4) - ((-(b^7*f^
2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2)*(B^2*a^5*d^3 - B^2*a^2*b^3*c^3 - 3*B^2*a^4
*b*c*d^2 + 3*B^2*a^3*b^2*c^2*d))^(1/2)*((32*(15*B^3*a^4*b^3*d^15*f^2 - B^3*
a^2*b^5*d^15*f^2 - 4*B^3*a^7*c^3*d^12*f^2 + 2*B^3*b^7*c^2*d^13*f^2 + 4*B^3*
b^7*c^4*d^11*f^2 + 2*B^3*b^7*c^6*d^9*f^2 - 12*B^3*a^6*b*d^15*f^2 - 4*B^3*a^
7*c*d^14*f^2 - B^3*a*b^6*c*d^14*f^2 - 27*B^3*a*b^6*c^3*d^12*f^2 - 19*B^3*a*
b^6*c^5*d^10*f^2 + 7*B^3*a*b^6*c^7*d^8*f^2 - 57*B^3*a^3*b^4*c*d^14*f^2 + 64
*B^3*a^5*b^2*c*d^14*f^2 + 4*B^3*a^6*b*c^2*d^13*f^2 + 16*B^3*a^6*b*c^4*d^11*
f^2 + 65*B^3*a^2*b^5*c^2*d^13*f^2 + 9*B^3*a^2*b^5*c^4*d^11*f^2 - 57*B^3*a^2
*b^5*c^6*d^9*f^2 + 77*B^3*a^3*b^4*c^3*d^12*f^2 + 129*B^3*a^3*b^4*c^5*d^10*f
^2 - 5*B^3*a^3*b^4*c^7*d^8*f^2 - 121*B^3*a^4*b^3*c^2*d^13*f^2 - 119*B^3*a^4
*b^3*c^4*d^11*f^2 + 17*B^3*a^4*b^3*c^6*d^9*f^2 + 40*B^3*a^5*b^2*c^3*d^12*f^
2 - 24*B^3*a^5*b^2*c^5*d^10*f^2))/(b*f^5) + ((-(b^7*f^2 + 2*a^2*b^5*f^2 + a
^4*b^3*f^2)*(B^2*a^5*d^3 - B^2*a^2*b^3*c^3 - 3*B^2*a^4*b*c*d^2 + 3*B^2*a^3*
b^2*c^2*d))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(4*B^2*a^3*b^5*d^13*f^2 +
2*B^2*a^5*b^3*d^13*f^2 + 28*B^2*b^8*c^3*d^10*f^2 - 10*B^2*b^8*c^5*d^8*f^2
- 14*B^2*a*b^7*d^13*f^2 + 16*B^2*a^7*b*d^13*f^2 - 8*B^2*a^8*c*d^12*f^2 + 22
*B^2*b^8*c*d^12*f^2 + 20*B^2*a*b^7*c^2*d^11*f^2 + 50*B^2*a*b^7*c^4*d^9*f^2
- 28*B^2*a^2*b^6*c*d^12*f^2 - 2*B^2*a^4*b^4*c*d^12*f^2 - 56*B^2*a^6*b^2*c*d

^12*f^2 + 32*B^2*a^7*b*c^2*d^11*f^2 + 8*B^2*a^2*b^6*c^3*d^10*f^2 + 12*B^2*a
^2*b^6*c^5*d^8*f^2 - 24*B^2*a^3*b^5*c^2*d^11*f^2 - 12*B^2*a^3*b^5*c^4*d^9*f
^2 - 4*B^2*a^4*b^4*c^3*d^10*f^2 - 10*B^2*a^4*b^4*c^5*d^8*f^2 + 52*B^2*a^5*b
^3*c^2*d^11*f^2 + 34*B^2*a^5*b^3*c^4*d^9*f^2 - 48*B^2*a^6*b^2*c^3*d^10*f^2)
)/(b*f^4) + ((-(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2)*(B^2*a^5*d^3 - B^2*a
^2*b^3*c^3 - 3*B^2*a^4*b*c*d^2 + 3*B^2*a^3*b^2*c^2*d))^(1/2)*((32*(4*B*a*b^
8*d^12*f^4 - 4*B*b^9*c*d^11*f^4 + 8*B*a^3*b^6*d^12*f^4 + 4*B*a^5*b^4*d^12*f
^4 - 4*B*b^9*c^3*d^9*f^4 + 8*B*a*b^8*c^2*d^10*f^4 + 4*B*a*b^8*c^4*d^8*f^4 -
12*B*a^2*b^7*c*d^11*f^4 - 12*B*a^4*b^5*c*d^11*f^4 - 4*B*a^6*b^3*c*d^11*f^4
- 12*B*a^2*b^7*c^3*d^9*f^4 + 16*B*a^3*b^6*c^2*d^10*f^4 + 8*B*a^3*b^6*c^4*d
^8*f^4 - 12*B*a^4*b^5*c^3*d^9*f^4 + 8*B*a^5*b^4*c^2*d^10*f^4 + 4*B*a^5*b^4*
c^4*d^8*f^4 - 4*B*a^6*b^3*c^3*d^9*f^4))/(b*f^5) - (32*(-(b^7*f^2 + 2*a^2*b^
5*f^2 + a^4*b^3*f^2)*(B^2*a^5*d^3 - B^2*a^2*b^3*c^3 - 3*B^2*a^4*b*c*d^2 + 3
*B^2*a^3*b^2*c^2*d))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(16*b^10*d^10*f^4 + 1
6*a^2*b^8*d^10*f^4 - 16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*d^10*f^4 + 24*b^10*c^
2*d^8*f^4 + 40*a^2*b^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^2*
d^8*f^4 + 8*a*b^9*c*d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 24*a^5*b^5*c*d^9*f^4 +
8*a^7*b^3*c*d^9*f^4))/(b^4*f^6*(a^2 + b^2)^2)))/(b^3*f^2*(a^2 + b^2)^2)))/
(b^3*f^2*(a^2 + b^2)^2)))/(b^3*f^2*(a^2 + b^2)^2)))/(b^3*f^2*(a^2 + b^2)^2)
+ ((-(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2)*(B^2*a^5*d^3 - B^2*a^2*b^3*c^
3 - 3*B^2*a^4*b*c*d^2 + 3*B^2*a^3*b^2*c^2*d))^(1/2)*((32*(c + d*tan(e + f*x
))^(1/2)*(B^4*b^6*d^16 - 2*B^4*a^6*d^16 + 12*B^4*a^6*c^2*d^14 - 2*B^4*a^6*c
^4*d^12 + 4*B^4*b^6*c^2*d^14 + 6*B^4*b^6*c^4*d^12 + 4*B^4*b^6*c^6*d^10 + B^
4*b^6*c^8*d^8 - 2*B^4*a^2*b^4*c^4*d^12 + 12*B^4*a^2*b^4*c^6*d^10 - 2*B^4*a^
2*b^4*c^8*d^8 + 8*B^4*a^3*b^3*c^3*d^13 - 48*B^4*a^3*b^3*c^5*d^11 + 8*B^4*a^
3*b^3*c^7*d^9 - 12*B^4*a^4*b^2*c^2*d^14 + 72*B^4*a^4*b^2*c^4*d^12 - 12*B^4*
a^4*b^2*c^6*d^10 + 8*B^4*a^5*b*c*d^15 - 48*B^4*a^5*b*c^3*d^13 + 8*B^4*a^5*b
*c^5*d^11))/(b*f^4) + ((-(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2)*(B^2*a^5*d
^3 - B^2*a^2*b^3*c^3 - 3*B^2*a^4*b*c*d^2 + 3*B^2*a^3*b^2*c^2*d))^(1/2)*((32
*(15*B^3*a^4*b^3*d^15*f^2 - B^3*a^2*b^5*d^15*f^2 - 4*B^3*a^7*c^3*d^12*f^2 +
2*B^3*b^7*c^2*d^13*f^2 + 4*B^3*b^7*c^4*d^11*f^2 + 2*B^3*b^7*c^6*d^9*f^2 -
12*B^3*a^6*b*d^15*f^2 - 4*B^3*a^7*c*d^14*f^2 - B^3*a*b^6*c*d^14*f^2 - 27*B^
3*a*b^6*c^3*d^12*f^2 - 19*B^3*a*b^6*c^5*d^10*f^2 + 7*B^3*a*b^6*c^7*d^8*f^2
- 57*B^3*a^3*b^4*c*d^14*f^2 + 64*B^3*a^5*b^2*c*d^14*f^2 + 4*B^3*a^6*b*c^2*d
^13*f^2 + 16*B^3*a^6*b*c^4*d^11*f^2 + 65*B^3*a^2*b^5*c^2*d^13*f^2 + 9*B^3*a
^2*b^5*c^4*d^11*f^2 - 57*B^3*a^2*b^5*c^6*d^9*f^2 + 77*B^3*a^3*b^4*c^3*d^12*
f^2 + 129*B^3*a^3*b^4*c^5*d^10*f^2 - 5*B^3*a^3*b^4*c^7*d^8*f^2 - 121*B^3*a^
4*b^3*c^2*d^13*f^2 - 119*B^3*a^4*b^3*c^4*d^11*f^2 + 17*B^3*a^4*b^3*c^6*d^9*
f^2 + 40*B^3*a^5*b^2*c^3*d^12*f^2 - 24*B^3*a^5*b^2*c^5*d^10*f^2))/(b*f^5) -
((-(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2)*(B^2*a^5*d^3 - B^2*a^2*b^3*c^3
- 3*B^2*a^4*b*c*d^2 + 3*B^2*a^3*b^2*c^2*d))^(1/2)*((32*(c + d*tan(e + f*x))
^(1/2)*(4*B^2*a^3*b^5*d^13*f^2 + 2*B^2*a^5*b^3*d^13*f^2 + 28*B^2*b^8*c^3*d^
10*f^2 - 10*B^2*b^8*c^5*d^8*f^2 - 14*B^2*a*b^7*d^13*f^2 + 16*B^2*a^7*b*d^13
*f^2 - 8*B^2*a^8*c*d^12*f^2 + 22*B^2*b^8*c*d^12*f^2 + 20*B^2*a*b^7*c^2*d^11
*f^2 + 50*B^2*a*b^7*c^4*d^9*f^2 - 28*B^2*a^2*b^6*c*d^12*f^2 - 2*B^2*a^4*b^4

*c*d^12*f^2 - 56*B^2*a^6*b^2*c*d^12*f^2 + 32*B^2*a^7*b*c^2*d^11*f^2 + 8*B^2
*a^2*b^6*c^3*d^10*f^2 + 12*B^2*a^2*b^6*c^5*d^8*f^2 - 24*B^2*a^3*b^5*c^2*d^1
1*f^2 - 12*B^2*a^3*b^5*c^4*d^9*f^2 - 4*B^2*a^4*b^4*c^3*d^10*f^2 - 10*B^2*a^
4*b^4*c^5*d^8*f^2 + 52*B^2*a^5*b^3*c^2*d^11*f^2 + 34*B^2*a^5*b^3*c^4*d^9*f^
2 - 48*B^2*a^6*b^2*c^3*d^10*f^2))/(b*f^4) - ((-(b^7*f^2 + 2*a^2*b^5*f^2 + a
^4*b^3*f^2)*(B^2*a^5*d^3 - B^2*a^2*b^3*c^3 - 3*B^2*a^4*b*c*d^2 + 3*B^2*a^3*
b^2*c^2*d))^(1/2)*((32*(4*B*a*b^8*d^12*f^4 - 4*B*b^9*c*d^11*f^4 + 8*B*a^3*b
^6*d^12*f^4 + 4*B*a^5*b^4*d^12*f^4 - 4*B*b^9*c^3*d^9*f^4 + 8*B*a*b^8*c^2*d^
10*f^4 + 4*B*a*b^8*c^4*d^8*f^4 - 12*B*a^2*b^7*c*d^11*f^4 - 12*B*a^4*b^5*c*d
^11*f^4 - 4*B*a^6*b^3*c*d^11*f^4 - 12*B*a^2*b^7*c^3*d^9*f^4 + 16*B*a^3*b^6*
c^2*d^10*f^4 + 8*B*a^3*b^6*c^4*d^8*f^4 - 12*B*a^4*b^5*c^3*d^9*f^4 + 8*B*a^5
*b^4*c^2*d^10*f^4 + 4*B*a^5*b^4*c^4*d^8*f^4 - 4*B*a^6*b^3*c^3*d^9*f^4))/(b*
f^5) + (32*(-(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2)*(B^2*a^5*d^3 - B^2*a^2
*b^3*c^3 - 3*B^2*a^4*b*c*d^2 + 3*B^2*a^3*b^2*c^2*d))^(1/2)*(c + d*tan(e + f
x))^(1/2)(16*b^10*d^10*f^4 + 16*a^2*b^8*d^10*f^4 - 16*a^4*b^6*d^10*f^4 -
16*a^6*b^4*d^10*f^4 + 24*b^10*c^2*d^8*f^4 + 40*a^2*b^8*c^2*d^8*f^4 + 8*a^4*
b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^9*c*d^9*f^4 + 24*a^3*b^7*c*
d^9*f^4 + 24*a^5*b^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9*f^4))/(b^4*f^6*(a^2 + b^2)
^2)))/(b^3*f^2*(a^2 + b^2)^2)))/(b^3*f^2*(a^2 + b^2)^2)))/(b^3*f^2*(a^2 + b
^2)^2)))/(b^3*f^2*(a^2 + b^2)^2)))*(-(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2
)*(B^2*a^5*d^3 - B^2*a^2*b^3*c^3 - 3*B^2*a^4*b*c*d^2 + 3*B^2*a^3*b^2*c^2*d)
)^(1/2)*2i)/(b^3*f^2*(a^2 + b^2)^2) + (atan((((-(b^5*f^2 + a^4*b*f^2 + 2*a^
2*b^3*f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A^2*a*b^2*c^2*d - 3*A^2*a^2*b*c*d
^2))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*d^16 + 4*A^4*b^5*c^2*d^
14 + 8*A^4*b^5*c^4*d^12 - 8*A^4*b^5*c^6*d^10 + 3*A^4*b^5*c^8*d^8 + 2*A^4*a^
4*b*d^16 + 12*A^4*a^2*b^3*c^2*d^14 - 72*A^4*a^2*b^3*c^4*d^12 + 12*A^4*a^2*b
^3*c^6*d^10 + 48*A^4*a^3*b^2*c^3*d^13 - 8*A^4*a^3*b^2*c^5*d^11 - 8*A^4*a*b^
4*c^3*d^13 + 48*A^4*a*b^4*c^5*d^11 - 8*A^4*a*b^4*c^7*d^9 - 8*A^4*a^3*b^2*c*
d^15 - 12*A^4*a^4*b*c^2*d^14 + 2*A^4*a^4*b*c^4*d^12))/f^4 + ((-(b^5*f^2 + a
^4*b*f^2 + 2*a^2*b^3*f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A^2*a*b^2*c^2*d -
3*A^2*a^2*b*c*d^2))^(1/2)*((32*(23*A^3*b^6*c^3*d^12*f^2 - 15*A^3*a^3*b^3*d^
15*f^2 + 21*A^3*b^6*c^5*d^10*f^2 - 3*A^3*b^6*c^7*d^8*f^2 + A^3*a*b^5*d^15*f
^2 + 4*A^3*a^5*b*d^15*f^2 - A^3*b^6*c*d^14*f^2 - 61*A^3*a*b^5*c^2*d^13*f^2
- 25*A^3*a*b^5*c^4*d^11*f^2 + 37*A^3*a*b^5*c^6*d^9*f^2 + 53*A^3*a^2*b^4*c*d
^14*f^2 - 30*A^3*a^4*b^2*c*d^14*f^2 + 4*A^3*a^5*b*c^2*d^13*f^2 - 29*A^3*a^2
*b^4*c^3*d^12*f^2 - 81*A^3*a^2*b^4*c^5*d^10*f^2 + A^3*a^2*b^4*c^7*d^8*f^2 +
59*A^3*a^3*b^3*c^2*d^13*f^2 + 75*A^3*a^3*b^3*c^4*d^11*f^2 + A^3*a^3*b^3*c^
6*d^9*f^2 - 32*A^3*a^4*b^2*c^3*d^12*f^2 - 2*A^3*a^4*b^2*c^5*d^10*f^2))/f^5
+ ((-(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A
^2*a*b^2*c^2*d - 3*A^2*a^2*b*c*d^2))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*
(4*A^2*a^3*b^4*d^13*f^2 - 14*A^2*a^5*b^2*d^13*f^2 + 28*A^2*b^7*c^3*d^10*f^2
- 18*A^2*b^7*c^5*d^8*f^2 - 14*A^2*a*b^6*d^13*f^2 + 22*A^2*b^7*c*d^12*f^2 +
8*A^2*a^6*b*c*d^12*f^2 + 20*A^2*a*b^6*c^2*d^11*f^2 + 66*A^2*a*b^6*c^4*d^9*
f^2 - 28*A^2*a^2*b^5*c*d^12*f^2 + 54*A^2*a^4*b^3*c*d^12*f^2 + 24*A^2*a^2*b^
5*c^3*d^10*f^2 + 12*A^2*a^2*b^5*c^5*d^8*f^2 - 88*A^2*a^3*b^4*c^2*d^11*f^2 -

28*A^2*a^3*b^4*c^4*d^9*f^2 + 60*A^2*a^4*b^3*c^3*d^10*f^2 - 2*A^2*a^4*b^3*c
^5*d^8*f^2 - 44*A^2*a^5*b^2*c^2*d^11*f^2 + 2*A^2*a^5*b^2*c^4*d^9*f^2))/f^4
+ ((-(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A
^2*a*b^2*c^2*d - 3*A^2*a^2*b*c*d^2))^(1/2)*((32*(4*A*a^2*b^6*d^12*f^4 + 8*A
*a^4*b^4*d^12*f^4 + 4*A*a^6*b^2*d^12*f^4 + 12*A*b^8*c^2*d^10*f^4 + 12*A*b^8
*c^4*d^8*f^4 - 16*A*a*b^7*c^3*d^9*f^4 - 32*A*a^3*b^5*c*d^11*f^4 - 16*A*a^5*
b^3*c*d^11*f^4 + 28*A*a^2*b^6*c^2*d^10*f^4 + 24*A*a^2*b^6*c^4*d^8*f^4 - 32*
A*a^3*b^5*c^3*d^9*f^4 + 20*A*a^4*b^4*c^2*d^10*f^4 + 12*A*a^4*b^4*c^4*d^8*f^
4 - 16*A*a^5*b^3*c^3*d^9*f^4 + 4*A*a^6*b^2*c^2*d^10*f^4 - 16*A*a*b^7*c*d^11
f^4))/f^5 + (32(-(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2)*(A^2*a^3*d^3 - A^2
*b^3*c^3 + 3*A^2*a*b^2*c^2*d - 3*A^2*a^2*b*c*d^2))^(1/2)*(c + d*tan(e + f*x
))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*
a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*
c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*
f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/(b*f^6*(a^2 + b^2)^2)))/
(b*f^2*(a^2 + b^2)^2)))/(b*f^2*(a^2 + b^2)^2)))/(b*f^2*(a^2 + b^2)^2))*1i)/
(b*f^2*(a^2 + b^2)^2) + ((-(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2)*(A^2*a^3*d
^3 - A^2*b^3*c^3 + 3*A^2*a*b^2*c^2*d - 3*A^2*a^2*b*c*d^2))^(1/2)*((32*(c +
d*tan(e + f*x))^(1/2)*(A^4*b^5*d^16 + 4*A^4*b^5*c^2*d^14 + 8*A^4*b^5*c^4*d^
12 - 8*A^4*b^5*c^6*d^10 + 3*A^4*b^5*c^8*d^8 + 2*A^4*a^4*b*d^16 + 12*A^4*a^2
*b^3*c^2*d^14 - 72*A^4*a^2*b^3*c^4*d^12 + 12*A^4*a^2*b^3*c^6*d^10 + 48*A^4*
a^3*b^2*c^3*d^13 - 8*A^4*a^3*b^2*c^5*d^11 - 8*A^4*a*b^4*c^3*d^13 + 48*A^4*a
*b^4*c^5*d^11 - 8*A^4*a*b^4*c^7*d^9 - 8*A^4*a^3*b^2*c*d^15 - 12*A^4*a^4*b*c
^2*d^14 + 2*A^4*a^4*b*c^4*d^12))/f^4 - ((-(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*
f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A^2*a*b^2*c^2*d - 3*A^2*a^2*b*c*d^2))^(
1/2)*((32*(23*A^3*b^6*c^3*d^12*f^2 - 15*A^3*a^3*b^3*d^15*f^2 + 21*A^3*b^6*c
^5*d^10*f^2 - 3*A^3*b^6*c^7*d^8*f^2 + A^3*a*b^5*d^15*f^2 + 4*A^3*a^5*b*d^15
*f^2 - A^3*b^6*c*d^14*f^2 - 61*A^3*a*b^5*c^2*d^13*f^2 - 25*A^3*a*b^5*c^4*d^
11*f^2 + 37*A^3*a*b^5*c^6*d^9*f^2 + 53*A^3*a^2*b^4*c*d^14*f^2 - 30*A^3*a^4*
b^2*c*d^14*f^2 + 4*A^3*a^5*b*c^2*d^13*f^2 - 29*A^3*a^2*b^4*c^3*d^12*f^2 - 8
1*A^3*a^2*b^4*c^5*d^10*f^2 + A^3*a^2*b^4*c^7*d^8*f^2 + 59*A^3*a^3*b^3*c^2*d
^13*f^2 + 75*A^3*a^3*b^3*c^4*d^11*f^2 + A^3*a^3*b^3*c^6*d^9*f^2 - 32*A^3*a^
4*b^2*c^3*d^12*f^2 - 2*A^3*a^4*b^2*c^5*d^10*f^2))/f^5 - ((-(b^5*f^2 + a^4*b
*f^2 + 2*a^2*b^3*f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A^2*a*b^2*c^2*d - 3*A^
2*a^2*b*c*d^2))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(4*A^2*a^3*b^4*d^13*f
^2 - 14*A^2*a^5*b^2*d^13*f^2 + 28*A^2*b^7*c^3*d^10*f^2 - 18*A^2*b^7*c^5*d^8
*f^2 - 14*A^2*a*b^6*d^13*f^2 + 22*A^2*b^7*c*d^12*f^2 + 8*A^2*a^6*b*c*d^12*f
^2 + 20*A^2*a*b^6*c^2*d^11*f^2 + 66*A^2*a*b^6*c^4*d^9*f^2 - 28*A^2*a^2*b^5*
c*d^12*f^2 + 54*A^2*a^4*b^3*c*d^12*f^2 + 24*A^2*a^2*b^5*c^3*d^10*f^2 + 12*A
^2*a^2*b^5*c^5*d^8*f^2 - 88*A^2*a^3*b^4*c^2*d^11*f^2 - 28*A^2*a^3*b^4*c^4*d
^9*f^2 + 60*A^2*a^4*b^3*c^3*d^10*f^2 - 2*A^2*a^4*b^3*c^5*d^8*f^2 - 44*A^2*a
^5*b^2*c^2*d^11*f^2 + 2*A^2*a^5*b^2*c^4*d^9*f^2))/f^4 - ((-(b^5*f^2 + a^4*b
*f^2 + 2*a^2*b^3*f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A^2*a*b^2*c^2*d - 3*A^
2*a^2*b*c*d^2))^(1/2)*((32*(4*A*a^2*b^6*d^12*f^4 + 8*A*a^4*b^4*d^12*f^4 + 4
*A*a^6*b^2*d^12*f^4 + 12*A*b^8*c^2*d^10*f^4 + 12*A*b^8*c^4*d^8*f^4 - 16*A*a

*b^7*c^3*d^9*f^4 - 32*A*a^3*b^5*c*d^11*f^4 - 16*A*a^5*b^3*c*d^11*f^4 + 28*A*a^2*b^6*c^2*d^10*f^4 + 24*A*a^2*b^6*c^4*d^8*f^4 - 32*A*a^3*b^5*c^3*d^9*f^4 + 20*A*a^4*b^4*c^2*d^10*f^4 + 12*A*a^4*b^4*c^4*d^8*f^4 - 16*A*a^5*b^3*c^3*d^9*f^4 + 4*A*a^6*b^2*c^2*d^10*f^4 - 16*A*a*b^7*c*d^11*f^4))/f^5 - (32*(-(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A^2*a*b^2*c^2*d - 3*A^2*a^2*b*c*d^2))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/(b*f^6*(a^2 + b^2)^2)))/(b*f^2*(a^2 + b^2)^2)))/(b*f^2*(a^2 + b^2)^2)))/(b*f^2*(a^2 + b^2)^2))*1i)/(b*f^2*(a^2 + b^2)^2))/((64*(A^5*a^2*b^2*d^18 + A^5*b^4*c^2*d^16 + 5*A^5*b^4*c^4*d^14 + 7*A^5*b^4*c^6*d^12 + 3*A^5*b^4*c^8*d^10 + 9*A^5*a^2*b^2*c^2*d^16 + 15*A^5*a^2*b^2*c^4*d^14 + 7*A^5*a^2*b^2*c^6*d^12 - 2*A^5*a*b^3*c*d^17 - 2*A^5*a^3*b*c*d^17 - 12*A^5*a*b^3*c^3*d^15 - 18*A^5*a*b^3*c^5*d^13 - 8*A^5*a*b^3*c^7*d^11 - 4*A^5*a^3*b*c^3*d^15 - 2*A^5*a^3*b*c^5*d^13))/f^5 + ((-(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A^2*a*b^2*c^2*d - 3*A^2*a^2*b*c*d^2))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*d^16 + 4*A^4*b^5*c^2*d^14 + 8*A^4*b^5*c^4*d^12 - 8*A^4*b^5*c^6*d^10 + 3*A^4*b^5*c^8*d^8 + 2*A^4*a^4*b*d^16 + 12*A^4*a^2*b^3*c^2*d^14 - 72*A^4*a^2*b^3*c^4*d^12 + 12*A^4*a^2*b^3*c^6*d^10 + 48*A^4*a^3*b^2*c^3*d^13 - 8*A^4*a^3*b^2*c^5*d^11 - 8*A^4*a*b^4*c^3*d^13 + 48*A^4*a*b^4*c^5*d^11 - 8*A^4*a*b^4*c^7*d^9 - 8*A^4*a^3*b^2*c*d^15 - 12*A^4*a^4*b*c^2*d^14 + 2*A^4*a^4*b*c^4*d^12))/f^4 + ((-(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A^2*a*b^2*c^2*d - 3*A^2*a^2*b*c*d^2))^(1/2)*((32*(23*A^3*b^6*c^3*d^12*f^2 - 15*A^3*a^3*b^3*d^15*f^2 + 21*A^3*b^6*c^5*d^10*f^2 - 3*A^3*b^6*c^7*d^8*f^2 + A^3*a*b^5*d^15*f^2 + 4*A^3*a^5*b*d^15*f^2 - A^3*b^6*c*d^14*f^2 - 61*A^3*a*b^5*c^2*d^13*f^2 - 25*A^3*a*b^5*c^4*d^11*f^2 + 37*A^3*a*b^5*c^6*d^9*f^2 + 53*A^3*a^2*b^4*c*d^14*f^2 - 30*A^3*a^4*b^2*c*d^14*f^2 + 4*A^3*a^5*b*c^2*d^13*f^2 - 29*A^3*a^2*b^4*c^3*d^12*f^2 - 81*A^3*a^2*b^4*c^5*d^10*f^2 + A^3*a^2*b^4*c^7*d^8*f^2 + 59*A^3*a^3*b^3*c^2*d^13*f^2 + 75*A^3*a^3*b^3*c^4*d^11*f^2 + A^3*a^3*b^3*c^6*d^9*f^2 - 32*A^3*a^4*b^2*c^3*d^12*f^2 - 2*A^3*a^4*b^2*c^5*d^10*f^2))/f^5 + ((-(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A^2*a*b^2*c^2*d - 3*A^2*a^2*b*c*d^2))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(4*A^2*a^3*b^4*d^13*f^2 - 14*A^2*a^5*b^2*d^13*f^2 + 28*A^2*b^7*c^3*d^10*f^2 - 18*A^2*b^7*c^5*d^8*f^2 - 14*A^2*a*b^6*d^13*f^2 + 22*A^2*b^7*c*d^12*f^2 + 8*A^2*a^6*b*c*d^12*f^2 + 20*A^2*a*b^6*c^2*d^11*f^2 + 66*A^2*a*b^6*c^4*d^9*f^2 - 28*A^2*a^2*b^5*c*d^12*f^2 + 54*A^2*a^4*b^3*c*d^12*f^2 + 24*A^2*a^2*b^5*c^3*d^10*f^2 + 12*A^2*a^2*b^5*c^5*d^8*f^2 - 88*A^2*a^3*b^4*c^2*d^11*f^2 - 28*A^2*a^3*b^4*c^4*d^9*f^2 + 60*A^2*a^4*b^3*c^3*d^10*f^2 - 2*A^2*a^4*b^3*c^5*d^8*f^2 - 44*A^2*a^5*b^2*c^2*d^11*f^2 + 2*A^2*a^5*b^2*c^4*d^9*f^2))/f^4 + ((-(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A^2*a*b^2*c^2*d - 3*A^2*a^2*b*c*d^2))^(1/2)*((32*(4*A*a^2*b^6*d^12*f^4 + 8*A*a^4*b^4*d^12*f^4 + 4*A*a^6*b^2*d^12*f^4 + 12*A*b^8*c^2*d^10*f^4 + 12*A*b^8*c^4*d^8*f^4 - 16*A*a*b^7*c^3*d^9*f^4 - 32*A*a^3*b^5*c*d^11*f^4 - 16*A *a^5*b^3*c*d^11*f^4 + 28*A*a^2*b^6*c^2*d^10*f^4 + 24*A*a^2*b^6*c^4*d^8*f^4 - 32*A*a^3*b^5*c^3*d^9*f^4 + 20*A*a^4*b^4*c^2*d^10*f^4 + 12*A*a^4*b^4*c^4*d ^8*f^4 - 16*A*a^5*b^3*c^3*d^9*f^4 + 4*A*a^6*b^2*c^2*d^10*f^4 - 16*A*a*b^7*c *d^11*f^4))/f^5 + (32*(-(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A^2*a*b^2*c^2*d - 3*A^2*a^2*b*c*d^2))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4 *b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c *d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/(b*f^6*(a^2 + b^2)^ 2)))/(b*f^2*(a^2 + b^2)^2)))/(b*f^2*(a^2 + b^2)^2)))/(b*f^2*(a^2 + b^2)^2)))/(b*f^2*(a^2 + b^2)^2) - ((-(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2)*(A^2*a^3 *d^3 - A^2*b^3*c^3 + 3*A^2*a*b^2*c^2*d - 3*A^2*a^2*b*c*d^2))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*d^16 + 4*A^4*b^5*c^2*d^14 + 8*A^4*b^5*c^4* d^12 - 8*A^4*b^5*c^6*d^10 + 3*A^4*b^5*c^8*d^8 + 2*A^4*a^4*b*d^16 + 12*A^4*a ^2*b^3*c^2*d^14 - 72*A^4*a^2*b^3*c^4*d^12 + 12*A^4*a^2*b^3*c^6*d^10 + 48*A^ 4*a^3*b^2*c^3*d^13 - 8*A^4*a^3*b^2*c^5*d^11 - 8*A^4*a*b^4*c^3*d^13 + 48*A^4 *a*b^4*c^5*d^11 - 8*A^4*a*b^4*c^7*d^9 - 8*A^4*a^3*b^2*c*d^15 - 12*A^4*a^4*b *c^2*d^14 + 2*A^4*a^4*b*c^4*d^12))/f^4 - ((-(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^ 3*f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A^2*a*b^2*c^2*d - 3*A^2*a^2*b*c*d^2)) ^(1/2)*((32*(23*A^3*b^6*c^3*d^12*f^2 - 15*A^3*a^3*b^3*d^15*f^2 + 21*A^3*b^6 *c^5*d^10*f^2 - 3*A^3*b^6*c^7*d^8*f^2 + A^3*a*b^5*d^15*f^2 + 4*A^3*a^5*b*d^ 15*f^2 - A^3*b^6*c*d^14*f^2 - 61*A^3*a*b^5*c^2*d^13*f^2 - 25*A^3*a*b^5*c^4* d^11*f^2 + 37*A^3*a*b^5*c^6*d^9*f^2 + 53*A^3*a^2*b^4*c*d^14*f^2 - 30*A^3*a^ 4*b^2*c*d^14*f^2 + 4*A^3*a^5*b*c^2*d^13*f^2 - 29*A^3*a^2*b^4*c^3*d^12*f^2 - 81*A^3*a^2*b^4*c^5*d^10*f^2 + A^3*a^2*b^4*c^7*d^8*f^2 + 59*A^3*a^3*b^3*c^2 *d^13*f^2 + 75*A^3*a^3*b^3*c^4*d^11*f^2 + A^3*a^3*b^3*c^6*d^9*f^2 - 32*A^3* a^4*b^2*c^3*d^12*f^2 - 2*A^3*a^4*b^2*c^5*d^10*f^2))/f^5 - ((-(b^5*f^2 + a^4 *b*f^2 + 2*a^2*b^3*f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A^2*a*b^2*c^2*d - 3* A^2*a^2*b*c*d^2))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(4*A^2*a^3*b^4*d^13 *f^2 - 14*A^2*a^5*b^2*d^13*f^2 + 28*A^2*b^7*c^3*d^10*f^2 - 18*A^2*b^7*c^5*d ^8*f^2 - 14*A^2*a*b^6*d^13*f^2 + 22*A^2*b^7*c*d^12*f^2 + 8*A^2*a^6*b*c*d^12 *f^2 + 20*A^2*a*b^6*c^2*d^11*f^2 + 66*A^2*a*b^6*c^4*d^9*f^2 - 28*A^2*a^2*b^ 5*c*d^12*f^2 + 54*A^2*a^4*b^3*c*d^12*f^2 + 24*A^2*a^2*b^5*c^3*d^10*f^2 + 12 *A^2*a^2*b^5*c^5*d^8*f^2 - 88*A^2*a^3*b^4*c^2*d^11*f^2 - 28*A^2*a^3*b^4*c^4 *d^9*f^2 + 60*A^2*a^4*b^3*c^3*d^10*f^2 - 2*A^2*a^4*b^3*c^5*d^8*f^2 - 44*A^2 *a^5*b^2*c^2*d^11*f^2 + 2*A^2*a^5*b^2*c^4*d^9*f^2))/f^4 - ((-(b^5*f^2 + a^4 *b*f^2 + 2*a^2*b^3*f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A^2*a*b^2*c^2*d - 3* A^2*a^2*b*c*d^2))^(1/2)*((32*(4*A*a^2*b^6*d^12*f^4 + 8*A*a^4*b^4*d^12*f^4 + 4*A*a^6*b^2*d^12*f^4 + 12*A*b^8*c^2*d^10*f^4 + 12*A*b^8*c^4*d^8*f^4 - 16*A *a*b^7*c^3*d^9*f^4 - 32*A*a^3*b^5*c*d^11*f^4 - 16*A*a^5*b^3*c*d^11*f^4 + 28 *A*a^2*b^6*c^2*d^10*f^4 + 24*A*a^2*b^6*c^4*d^8*f^4 - 32*A*a^3*b^5*c^3*d^9*f ^4 + 20*A*a^4*b^4*c^2*d^10*f^4 + 12*A*a^4*b^4*c^4*d^8*f^4 - 16*A*a^5*b^3*c^ 3*d^9*f^4 + 4*A*a^6*b^2*c^2*d^10*f^4 - 16*A*a*b^7*c*d^11*f^4))/f^5 - (32*(- (b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A^2*a* b^2*c^2*d - 3*A^2*a^2*b*c*d^2))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(16*b^9*d^

10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/(b*f^6*(a^2 + b^2)^2)))/(b*f^2*(a^2 + b^2)^2)))/(b*f^2*(a^2 + b^2)^2)))/(b*f^2*(a^2 + b^2)^2)))/(b*f^2*(a^2 + b^2)^2)))*(-(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A^2*a*b^2*c^2*d - 3*A^2*a^2*b*c*d^2))^(1/2)*2i)/(b*f^2*(a^2 + b^2)^2) + (2*B*d*(c + d*tan(e + f*x))^(1/2))/(b*f) + (atan((((-(b^9*f^2 + 2*a^2*b^7*f^2 + a^4*b^5*f^2)*(C^2*a^7*d^3 - C^2*a^4*b^3*c^3 - 3*C^2*a^6*b*c*d^2 + 3*C^2*a^5*b^2*c^2*d))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(2*C^4*a^8*d^16 + C^4*b^8*d^16 - 12*C^4*a^8*c^2*d^14 + 2*C^4*a^8*c^4*d^12 + 4*C^4*b^8*c^2*d^14 + 6*C^4*b^8*c^4*d^12 + 4*C^4*b^8*c^6*d^10 + C^4*b^8*c^8*d^8 + 2*C^4*a^4*b^4*c^4*d^12 - 12*C^4*a^4*b^4*c^6*d^10 + 2*C^4*a^4*b^4*c^8*d^8 - 8*C^4*a^5*b^3*c^3*d^13 + 48*C^4*a^5*b^3*c^5*d^11 - 8*C^4*a^5*b^3*c^7*d^9 + 12*C^4*a^6*b^2*c^2*d^14 - 72*C^4*a^6*b^2*c^4*d^12 + 12*C^4*a^6*b^2*c^6*d^10 - 8*C^4*a^7*b*c*d^15 + 48*C^4*a^7*b*c^3*d^13 - 8*C^4*a^7*b*c^5*d^11))/(b^3*f^4) + ((-(b^9*f^2 + 2*a^2*b^7*f^2 + a^4*b^5*f^2)*(C^2*a^7*d^3 - C^2*a^4*b^3*c^3 - 3*C^2*a^6*b*c*d^2 + 3*C^2*a^5*b^2*c^2*d))^(1/2)*((32*(4*C^3*a^9*d^15*f^2 + C^3*a^3*b^6*d^15*f^2 + 16*C^3*a^5*b^4*d^15*f^2 - 16*C^3*a^7*b^2*d^15*f^2 + 4*C^3*a^9*c^2*d^13*f^2 - C^3*b^9*c^3*d^12*f^2 + C^3*b^9*c^5*d^10*f^2 + C^3*b^9*c^7*d^8*f^2 + C^3*a*b^8*d^15*f^2 - C^3*b^9*c*d^14*f^2 - 28*C^3*a^8*b*c*d^14*f^2 + 3*C^3*a*b^8*c^2*d^13*f^2 + 3*C^3*a*b^8*c^4*d^11*f^2 + C^3*a*b^8*c^6*d^9*f^2 - 3*C^3*a^2*b^7*c*d^14*f^2 - 58*C^3*a^4*b^5*c*d^14*f^2 + 80*C^3*a^6*b^3*c*d^14*f^2 - 28*C^3*a^8*b*c^3*d^12*f^2 - 29*C^3*a^2*b^7*c^3*d^12*f^2 - 17*C^3*a^2*b^7*c^5*d^10*f^2 + 9*C^3*a^2*b^7*c^7*d^8*f^2 + 67*C^3*a^3*b^6*c^2*d^13*f^2 + 3*C^3*a^3*b^6*c^4*d^11*f^2 - 63*C^3*a^3*b^6*c^6*d^9*f^2 + 92*C^3*a^4*b^5*c^3*d^12*f^2 + 138*C^3*a^4*b^5*c^5*d^10*f^2 - 12*C^3*a^4*b^5*c^7*d^8*f^2 - 144*C^3*a^5*b^4*c^2*d^13*f^2 - 108*C^3*a^5*b^4*c^4*d^11*f^2 + 52*C^3*a^5*b^4*c^6*d^9*f^2 - 8*C^3*a^6*b^3*c^3*d^12*f^2 - 88*C^3*a^6*b^3*c^5*d^10*f^2 + 56*C^3*a^7*b^2*c^2*d^13*f^2 + 72*C^3*a^7*b^2*c^4*d^11*f^2))/(b^3*f^5) + ((-(b^9*f^2 + 2*a^2*b^7*f^2 + a^4*b^5*f^2)*(C^2*a^7*d^3 - C^2*a^4*b^3*c^3 - 3*C^2*a^6*b*c*d^2 + 3*C^2*a^5*b^2*c^2*d))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(4*C^2*a^3*b^7*d^13*f^2 + 2*C^2*a^5*b^5*d^13*f^2 + 28*C^2*b^10*c^3*d^10*f^2 - 10*C^2*b^10*c^5*d^8*f^2 - 14*C^2*a*b^9*d^13*f^2 - 16*C^2*a^9*b*d^13*f^2 + 8*C^2*a^10*c*d^12*f^2 + 22*C^2*b^10*c*d^12*f^2 + 20*C^2*a*b^9*c^2*d^11*f^2 + 50*C^2*a*b^9*c^4*d^9*f^2 - 28*C^2*a^2*b^8*c*d^12*f^2 - 2*C^2*a^4*b^6*c*d^12*f^2 + 56*C^2*a^8*b^2*c*d^12*f^2 - 32*C^2*a^9*b*c^2*d^11*f^2 + 8*C^2*a^2*b^8*c^3*d^10*f^2 + 4*C^2*a^2*b^8*c^5*d^8*f^2 - 24*C^2*a^3*b^7*c^2*d^11*f^2 + 4*C^2*a^3*b^7*c^4*d^9*f^2 + 12*C^2*a^4*b^6*c^3*d^10*f^2 - 10*C^2*a^4*b^6*c^5*d^8*f^2 - 12*C^2*a^5*b^5*c^2*d^11*f^2 + 18*C^2*a^5*b^5*c^4*d^9*f^2 + 16*C^2*a^6*b^4*c^3*d^10*f^2 + 8*C^2*a^6*b^4*c^5*d^8*f^2 - 64*C^2*a^7*b^3*c^2*d^11*f^2 - 32*C^2*a^7*b^3*c^4*d^9*f^2 + 48*C^2*a^8*b^2*c^3*d^10*f^2))/(b^3*f^4) - ((-(b^9*f^2 + 2*a^2*b^7*f^2 + a^4*b^5*f^2)*(C^2*a^7*d^3 - C^2*a^4*b^3*c^3 - 3*C^2*a^6*b*c*d^2 + 3*C^2*a^5*b^2*c^2*d))^(1/2)*((32*(12*C*a^2*b^9*d^12*f^4 + 24*C*a^4*b^7*d^12*f^4 + 12*C*a^6*b^5*d^12*f^4 + 4*C

*b^11*c^2*d^10*f^4 + 4*C*b^11*c^4*d^8*f^4 - 16*C*a*b^10*c^3*d^9*f^4 - 32*C*
a^3*b^8*c*d^11*f^4 - 16*C*a^5*b^6*c*d^11*f^4 + 20*C*a^2*b^9*c^2*d^10*f^4 +
8*C*a^2*b^9*c^4*d^8*f^4 - 32*C*a^3*b^8*c^3*d^9*f^4 + 28*C*a^4*b^7*c^2*d^10*
f^4 + 4*C*a^4*b^7*c^4*d^8*f^4 - 16*C*a^5*b^6*c^3*d^9*f^4 + 12*C*a^6*b^5*c^2
*d^10*f^4 - 16*C*a*b^10*c*d^11*f^4))/(b^3*f^5) - (32*(-(b^9*f^2 + 2*a^2*b^7
*f^2 + a^4*b^5*f^2)*(C^2*a^7*d^3 - C^2*a^4*b^3*c^3 - 3*C^2*a^6*b*c*d^2 + 3*
C^2*a^5*b^2*c^2*d))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(16*b^12*d^10*f^4 + 16
*a^2*b^10*d^10*f^4 - 16*a^4*b^8*d^10*f^4 - 16*a^6*b^6*d^10*f^4 + 24*b^12*c^
2*d^8*f^4 + 40*a^2*b^10*c^2*d^8*f^4 + 8*a^4*b^8*c^2*d^8*f^4 - 8*a^6*b^6*c^2
*d^8*f^4 + 8*a*b^11*c*d^9*f^4 + 24*a^3*b^9*c*d^9*f^4 + 24*a^5*b^7*c*d^9*f^4
 + 8*a^7*b^5*c*d^9*f^4))/(b^8*f^6*(a^2 + b^2)^2)))/(b^5*f^2*(a^2 + b^2)^2))
)/(b^5*f^2*(a^2 + b^2)^2)))/(b^5*f^2*(a^2 + b^2)^2))*1i)/(b^5*f^2*(a^2 + b^
2)^2) + ((-(b^9*f^2 + 2*a^2*b^7*f^2 + a^4*b^5*f^2)*(C^2*a^7*d^3 - C^2*a^4*b
^3*c^3 - 3*C^2*a^6*b*c*d^2 + 3*C^2*a^5*b^2*c^2*d))^(1/2)*((32*(c + d*tan(e
+ f*x))^(1/2)*(2*C^4*a^8*d^16 + C^4*b^8*d^16 - 12*C^4*a^8*c^2*d^14 + 2*C^4*
a^8*c^4*d^12 + 4*C^4*b^8*c^2*d^14 + 6*C^4*b^8*c^4*d^12 + 4*C^4*b^8*c^6*d^10
 + C^4*b^8*c^8*d^8 + 2*C^4*a^4*b^4*c^4*d^12 - 12*C^4*a^4*b^4*c^6*d^10 + 2*C
^4*a^4*b^4*c^8*d^8 - 8*C^4*a^5*b^3*c^3*d^13 + 48*C^4*a^5*b^3*c^5*d^11 - 8*C
^4*a^5*b^3*c^7*d^9 + 12*C^4*a^6*b^2*c^2*d^14 - 72*C^4*a^6*b^2*c^4*d^12 + 12
*C^4*a^6*b^2*c^6*d^10 - 8*C^4*a^7*b*c*d^15 + 48*C^4*a^7*b*c^3*d^13 - 8*C^4*
a^7*b*c^5*d^11))/(b^3*f^4) - ((-(b^9*f^2 + 2*a^2*b^7*f^2 + a^4*b^5*f^2)*(C^
2*a^7*d^3 - C^2*a^4*b^3*c^3 - 3*C^2*a^6*b*c*d^2 + 3*C^2*a^5*b^2*c^2*d))^(1/
2)*((32*(4*C^3*a^9*d^15*f^2 + C^3*a^3*b^6*d^15*f^2 + 16*C^3*a^5*b^4*d^15*f^
2 - 16*C^3*a^7*b^2*d^15*f^2 + 4*C^3*a^9*c^2*d^13*f^2 - C^3*b^9*c^3*d^12*f^2
 + C^3*b^9*c^5*d^10*f^2 + C^3*b^9*c^7*d^8*f^2 + C^3*a*b^8*d^15*f^2 - C^3*b^
9*c*d^14*f^2 - 28*C^3*a^8*b*c*d^14*f^2 + 3*C^3*a*b^8*c^2*d^13*f^2 + 3*C^3*a
*b^8*c^4*d^11*f^2 + C^3*a*b^8*c^6*d^9*f^2 - 3*C^3*a^2*b^7*c*d^14*f^2 - 58*C
^3*a^4*b^5*c*d^14*f^2 + 80*C^3*a^6*b^3*c*d^14*f^2 - 28*C^3*a^8*b*c^3*d^12*f
^2 - 29*C^3*a^2*b^7*c^3*d^12*f^2 - 17*C^3*a^2*b^7*c^5*d^10*f^2 + 9*C^3*a^2*
b^7*c^7*d^8*f^2 + 67*C^3*a^3*b^6*c^2*d^13*f^2 + 3*C^3*a^3*b^6*c^4*d^11*f^2
- 63*C^3*a^3*b^6*c^6*d^9*f^2 + 92*C^3*a^4*b^5*c^3*d^12*f^2 + 138*C^3*a^4*b^
5*c^5*d^10*f^2 - 12*C^3*a^4*b^5*c^7*d^8*f^2 - 144*C^3*a^5*b^4*c^2*d^13*f^2
- 108*C^3*a^5*b^4*c^4*d^11*f^2 + 52*C^3*a^5*b^4*c^6*d^9*f^2 - 8*C^3*a^6*b^3
*c^3*d^12*f^2 - 88*C^3*a^6*b^3*c^5*d^10*f^2 + 56*C^3*a^7*b^2*c^2*d^13*f^2 +
 72*C^3*a^7*b^2*c^4*d^11*f^2))/(b^3*f^5) - ((-(b^9*f^2 + 2*a^2*b^7*f^2 + a^
4*b^5*f^2)*(C^2*a^7*d^3 - C^2*a^4*b^3*c^3 - 3*C^2*a^6*b*c*d^2 + 3*C^2*a^5*b
^2*c^2*d))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(4*C^2*a^3*b^7*d^13*f^2 +
2*C^2*a^5*b^5*d^13*f^2 + 28*C^2*b^10*c^3*d^10*f^2 - 10*C^2*b^10*c^5*d^8*f^2
 - 14*C^2*a*b^9*d^13*f^2 - 16*C^2*a^9*b*d^13*f^2 + 8*C^2*a^10*c*d^12*f^2 +
22*C^2*b^10*c*d^12*f^2 + 20*C^2*a*b^9*c^2*d^11*f^2 + 50*C^2*a*b^9*c^4*d^9*f
^2 - 28*C^2*a^2*b^8*c*d^12*f^2 - 2*C^2*a^4*b^6*c*d^12*f^2 + 56*C^2*a^8*b^2*
c*d^12*f^2 - 32*C^2*a^9*b*c^2*d^11*f^2 + 8*C^2*a^2*b^8*c^3*d^10*f^2 + 4*C^2
*a^2*b^8*c^5*d^8*f^2 - 24*C^2*a^3*b^7*c^2*d^11*f^2 + 4*C^2*a^3*b^7*c^4*d^9*
f^2 + 12*C^2*a^4*b^6*c^3*d^10*f^2 - 10*C^2*a^4*b^6*c^5*d^8*f^2 - 12*C^2*a^5
*b^5*c^2*d^11*f^2 + 18*C^2*a^5*b^5*c^4*d^9*f^2 + 16*C^2*a^6*b^4*c^3*d^10*f^

2 + 8*C^2*a^6*b^4*c^5*d^8*f^2 - 64*C^2*a^7*b^3*c^2*d^11*f^2 - 32*C^2*a^7*b^
3*c^4*d^9*f^2 + 48*C^2*a^8*b^2*c^3*d^10*f^2))/(b^3*f^4) + ((-(b^9*f^2 + 2*a
^2*b^7*f^2 + a^4*b^5*f^2)*(C^2*a^7*d^3 - C^2*a^4*b^3*c^3 - 3*C^2*a^6*b*c*d^
2 + 3*C^2*a^5*b^2*c^2*d))^(1/2)*((32*(12*C*a^2*b^9*d^12*f^4 + 24*C*a^4*b^7*
d^12*f^4 + 12*C*a^6*b^5*d^12*f^4 + 4*C*b^11*c^2*d^10*f^4 + 4*C*b^11*c^4*d^8
*f^4 - 16*C*a*b^10*c^3*d^9*f^4 - 32*C*a^3*b^8*c*d^11*f^4 - 16*C*a^5*b^6*c*d
^11*f^4 + 20*C*a^2*b^9*c^2*d^10*f^4 + 8*C*a^2*b^9*c^4*d^8*f^4 - 32*C*a^3*b^
8*c^3*d^9*f^4 + 28*C*a^4*b^7*c^2*d^10*f^4 + 4*C*a^4*b^7*c^4*d^8*f^4 - 16*C*
a^5*b^6*c^3*d^9*f^4 + 12*C*a^6*b^5*c^2*d^10*f^4 - 16*C*a*b^10*c*d^11*f^4))/
(b^3*f^5) + (32*(-(b^9*f^2 + 2*a^2*b^7*f^2 + a^4*b^5*f^2)*(C^2*a^7*d^3 - C^
2*a^4*b^3*c^3 - 3*C^2*a^6*b*c*d^2 + 3*C^2*a^5*b^2*c^2*d))^(1/2)*(c + d*tan(
e + f*x))^(1/2)*(16*b^12*d^10*f^4 + 16*a^2*b^10*d^10*f^4 - 16*a^4*b^8*d^10*
f^4 - 16*a^6*b^6*d^10*f^4 + 24*b^12*c^2*d^8*f^4 + 40*a^2*b^10*c^2*d^8*f^4 +
 8*a^4*b^8*c^2*d^8*f^4 - 8*a^6*b^6*c^2*d^8*f^4 + 8*a*b^11*c*d^9*f^4 + 24*a^
3*b^9*c*d^9*f^4 + 24*a^5*b^7*c*d^9*f^4 + 8*a^7*b^5*c*d^9*f^4))/(b^8*f^6*(a^
2 + b^2)^2)))/(b^5*f^2*(a^2 + b^2)^2)))/(b^5*f^2*(a^2 + b^2)^2)))/(b^5*f^2*
(a^2 + b^2)^2))*1i)/(b^5*f^2*(a^2 + b^2)^2))/((64*(C^5*a^4*b^3*d^18 + 4*C^5
*a^7*c^3*d^15 + 2*C^5*a^7*c^5*d^13 - C^5*a^6*b*d^18 + 2*C^5*a^7*c*d^17 + C^
5*a^2*b^5*c^2*d^16 + 4*C^5*a^2*b^5*c^4*d^14 + 6*C^5*a^2*b^5*c^6*d^12 + 4*C^
5*a^2*b^5*c^8*d^10 + C^5*a^2*b^5*c^10*d^8 - 8*C^5*a^3*b^4*c^3*d^15 - 12*C^5
*a^3*b^4*c^5*d^13 - 8*C^5*a^3*b^4*c^7*d^11 - 2*C^5*a^3*b^4*c^9*d^9 + 3*C^5*
a^4*b^3*c^2*d^16 + C^5*a^4*b^3*c^4*d^14 - 3*C^5*a^4*b^3*c^6*d^12 - 2*C^5*a^
4*b^3*c^8*d^10 + 12*C^5*a^5*b^2*c^3*d^15 + 18*C^5*a^5*b^2*c^5*d^13 + 8*C^5*
a^5*b^2*c^7*d^11 - 2*C^5*a^3*b^4*c*d^17 + 2*C^5*a^5*b^2*c*d^17 - 9*C^5*a^6*
b*c^2*d^16 - 15*C^5*a^6*b*c^4*d^14 - 7*C^5*a^6*b*c^6*d^12))/(b^3*f^5) - ((-
(b^9*f^2 + 2*a^2*b^7*f^2 + a^4*b^5*f^2)*(C^2*a^7*d^3 - C^2*a^4*b^3*c^3 - 3*
C^2*a^6*b*c*d^2 + 3*C^2*a^5*b^2*c^2*d))^(1/2)*((32*(c + d*tan(e + f*x))^(1/
2)*(2*C^4*a^8*d^16 + C^4*b^8*d^16 - 12*C^4*a^8*c^2*d^14 + 2*C^4*a^8*c^4*d^1
2 + 4*C^4*b^8*c^2*d^14 + 6*C^4*b^8*c^4*d^12 + 4*C^4*b^8*c^6*d^10 + C^4*b^8*
c^8*d^8 + 2*C^4*a^4*b^4*c^4*d^12 - 12*C^4*a^4*b^4*c^6*d^10 + 2*C^4*a^4*b^4*
c^8*d^8 - 8*C^4*a^5*b^3*c^3*d^13 + 48*C^4*a^5*b^3*c^5*d^11 - 8*C^4*a^5*b^3*
c^7*d^9 + 12*C^4*a^6*b^2*c^2*d^14 - 72*C^4*a^6*b^2*c^4*d^12 + 12*C^4*a^6*b^
2*c^6*d^10 - 8*C^4*a^7*b*c*d^15 + 48*C^4*a^7*b*c^3*d^13 - 8*C^4*a^7*b*c^5*d
^11))/(b^3*f^4) + ((-(b^9*f^2 + 2*a^2*b^7*f^2 + a^4*b^5*f^2)*(C^2*a^7*d^3 -
 C^2*a^4*b^3*c^3 - 3*C^2*a^6*b*c*d^2 + 3*C^2*a^5*b^2*c^2*d))^(1/2)*((32*(4*
C^3*a^9*d^15*f^2 + C^3*a^3*b^6*d^15*f^2 + 16*C^3*a^5*b^4*d^15*f^2 - 16*C^3*
a^7*b^2*d^15*f^2 + 4*C^3*a^9*c^2*d^13*f^2 - C^3*b^9*c^3*d^12*f^2 + C^3*b^9*
c^5*d^10*f^2 + C^3*b^9*c^7*d^8*f^2 + C^3*a*b^8*d^15*f^2 - C^3*b^9*c*d^14*f^
2 - 28*C^3*a^8*b*c*d^14*f^2 + 3*C^3*a*b^8*c^2*d^13*f^2 + 3*C^3*a*b^8*c^4*d^
11*f^2 + C^3*a*b^8*c^6*d^9*f^2 - 3*C^3*a^2*b^7*c*d^14*f^2 - 58*C^3*a^4*b^5*
c*d^14*f^2 + 80*C^3*a^6*b^3*c*d^14*f^2 - 28*C^3*a^8*b*c^3*d^12*f^2 - 29*C^3
*a^2*b^7*c^3*d^12*f^2 - 17*C^3*a^2*b^7*c^5*d^10*f^2 + 9*C^3*a^2*b^7*c^7*d^8
*f^2 + 67*C^3*a^3*b^6*c^2*d^13*f^2 + 3*C^3*a^3*b^6*c^4*d^11*f^2 - 63*C^3*a^
3*b^6*c^6*d^9*f^2 + 92*C^3*a^4*b^5*c^3*d^12*f^2 + 138*C^3*a^4*b^5*c^5*d^10*
f^2 - 12*C^3*a^4*b^5*c^7*d^8*f^2 - 144*C^3*a^5*b^4*c^2*d^13*f^2 - 108*C^3*a

^5*b^4*c^4*d^11*f^2 + 52*C^3*a^5*b^4*c^6*d^9*f^2 - 8*C^3*a^6*b^3*c^3*d^12*f
^2 - 88*C^3*a^6*b^3*c^5*d^10*f^2 + 56*C^3*a^7*b^2*c^2*d^13*f^2 + 72*C^3*a^7
*b^2*c^4*d^11*f^2))/(b^3*f^5) + ((-(b^9*f^2 + 2*a^2*b^7*f^2 + a^4*b^5*f^2)*
(C^2*a^7*d^3 - C^2*a^4*b^3*c^3 - 3*C^2*a^6*b*c*d^2 + 3*C^2*a^5*b^2*c^2*d))^
(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(4*C^2*a^3*b^7*d^13*f^2 + 2*C^2*a^5*b
^5*d^13*f^2 + 28*C^2*b^10*c^3*d^10*f^2 - 10*C^2*b^10*c^5*d^8*f^2 - 14*C^2*a
*b^9*d^13*f^2 - 16*C^2*a^9*b*d^13*f^2 + 8*C^2*a^10*c*d^12*f^2 + 22*C^2*b^10
*c*d^12*f^2 + 20*C^2*a*b^9*c^2*d^11*f^2 + 50*C^2*a*b^9*c^4*d^9*f^2 - 28*C^2
*a^2*b^8*c*d^12*f^2 - 2*C^2*a^4*b^6*c*d^12*f^2 + 56*C^2*a^8*b^2*c*d^12*f^2
- 32*C^2*a^9*b*c^2*d^11*f^2 + 8*C^2*a^2*b^8*c^3*d^10*f^2 + 4*C^2*a^2*b^8*c^
5*d^8*f^2 - 24*C^2*a^3*b^7*c^2*d^11*f^2 + 4*C^2*a^3*b^7*c^4*d^9*f^2 + 12*C^
2*a^4*b^6*c^3*d^10*f^2 - 10*C^2*a^4*b^6*c^5*d^8*f^2 - 12*C^2*a^5*b^5*c^2*d^
11*f^2 + 18*C^2*a^5*b^5*c^4*d^9*f^2 + 16*C^2*a^6*b^4*c^3*d^10*f^2 + 8*C^2*a
^6*b^4*c^5*d^8*f^2 - 64*C^2*a^7*b^3*c^2*d^11*f^2 - 32*C^2*a^7*b^3*c^4*d^9*f
^2 + 48*C^2*a^8*b^2*c^3*d^10*f^2))/(b^3*f^4) - ((-(b^9*f^2 + 2*a^2*b^7*f^2
+ a^4*b^5*f^2)*(C^2*a^7*d^3 - C^2*a^4*b^3*c^3 - 3*C^2*a^6*b*c*d^2 + 3*C^2*a
^5*b^2*c^2*d))^(1/2)*((32*(12*C*a^2*b^9*d^12*f^4 + 24*C*a^4*b^7*d^12*f^4 +
12*C*a^6*b^5*d^12*f^4 + 4*C*b^11*c^2*d^10*f^4 + 4*C*b^11*c^4*d^8*f^4 - 16*C
*a*b^10*c^3*d^9*f^4 - 32*C*a^3*b^8*c*d^11*f^4 - 16*C*a^5*b^6*c*d^11*f^4 + 2
0*C*a^2*b^9*c^2*d^10*f^4 + 8*C*a^2*b^9*c^4*d^8*f^4 - 32*C*a^3*b^8*c^3*d^9*f
^4 + 28*C*a^4*b^7*c^2*d^10*f^4 + 4*C*a^4*b^7*c^4*d^8*f^4 - 16*C*a^5*b^6*c^3
*d^9*f^4 + 12*C*a^6*b^5*c^2*d^10*f^4 - 16*C*a*b^10*c*d^11*f^4))/(b^3*f^5) -
(32*(-(b^9*f^2 + 2*a^2*b^7*f^2 + a^4*b^5*f^2)*(C^2*a^7*d^3 - C^2*a^4*b^3*c
^3 - 3*C^2*a^6*b*c*d^2 + 3*C^2*a^5*b^2*c^2*d))^(1/2)*(c + d*tan(e + f*x))^(
1/2)*(16*b^12*d^10*f^4 + 16*a^2*b^10*d^10*f^4 - 16*a^4*b^8*d^10*f^4 - 16*a^
6*b^6*d^10*f^4 + 24*b^12*c^2*d^8*f^4 + 40*a^2*b^10*c^2*d^8*f^4 + 8*a^4*b^8*
c^2*d^8*f^4 - 8*a^6*b^6*c^2*d^8*f^4 + 8*a*b^11*c*d^9*f^4 + 24*a^3*b^9*c*d^9
*f^4 + 24*a^5*b^7*c*d^9*f^4 + 8*a^7*b^5*c*d^9*f^4))/(b^8*f^6*(a^2 + b^2)^2)
))/(b^5*f^2*(a^2 + b^2)^2)))/(b^5*f^2*(a^2 + b^2)^2)))/(b^5*f^2*(a^2 + b^2)
^2)))/(b^5*f^2*(a^2 + b^2)^2) + ((-(b^9*f^2 + 2*a^2*b^7*f^2 + a^4*b^5*f^2)*
(C^2*a^7*d^3 - C^2*a^4*b^3*c^3 - 3*C^2*a^6*b*c*d^2 + 3*C^2*a^5*b^2*c^2*d))^
(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(2*C^4*a^8*d^16 + C^4*b^8*d^16 - 12*C
^4*a^8*c^2*d^14 + 2*C^4*a^8*c^4*d^12 + 4*C^4*b^8*c^2*d^14 + 6*C^4*b^8*c^4*d
^12 + 4*C^4*b^8*c^6*d^10 + C^4*b^8*c^8*d^8 + 2*C^4*a^4*b^4*c^4*d^12 - 12*C^
4*a^4*b^4*c^6*d^10 + 2*C^4*a^4*b^4*c^8*d^8 - 8*C^4*a^5*b^3*c^3*d^13 + 48*C^
4*a^5*b^3*c^5*d^11 - 8*C^4*a^5*b^3*c^7*d^9 + 12*C^4*a^6*b^2*c^2*d^14 - 72*C
^4*a^6*b^2*c^4*d^12 + 12*C^4*a^6*b^2*c^6*d^10 - 8*C^4*a^7*b*c*d^15 + 48*C^4
*a^7*b*c^3*d^13 - 8*C^4*a^7*b*c^5*d^11))/(b^3*f^4) - ((-(b^9*f^2 + 2*a^2*b^
7*f^2 + a^4*b^5*f^2)*(C^2*a^7*d^3 - C^2*a^4*b^3*c^3 - 3*C^2*a^6*b*c*d^2 + 3
*C^2*a^5*b^2*c^2*d))^(1/2)*((32*(4*C^3*a^9*d^15*f^2 + C^3*a^3*b^6*d^15*f^2
+ 16*C^3*a^5*b^4*d^15*f^2 - 16*C^3*a^7*b^2*d^15*f^2 + 4*C^3*a^9*c^2*d^13*f^
2 - C^3*b^9*c^3*d^12*f^2 + C^3*b^9*c^5*d^10*f^2 + C^3*b^9*c^7*d^8*f^2 + C^3
*a*b^8*d^15*f^2 - C^3*b^9*c*d^14*f^2 - 28*C^3*a^8*b*c*d^14*f^2 + 3*C^3*a*b^
8*c^2*d^13*f^2 + 3*C^3*a*b^8*c^4*d^11*f^2 + C^3*a*b^8*c^6*d^9*f^2 - 3*C^3*a
^2*b^7*c*d^14*f^2 - 58*C^3*a^4*b^5*c*d^14*f^2 + 80*C^3*a^6*b^3*c*d^14*f^2 -

```
 28*C^3*a^8*b*c^3*d^12*f^2 - 29*C^3*a^2*b^7*c^3*d^12*f^2 - 17*C^3*a^2*b^7*c
^5*d^10*f^2 + 9*C^3*a^2*b^7*c^7*d^8*f^2 + 67*C^3*a^3*b^6*c^2*d^13*f^2 + 3*C
^3*a^3*b^6*c^4*d^11*f^2 - 63*C^3*a^3*b^6*c^6*d^9*f^2 + 92*C^3*a^4*b^5*c^3*d
^12*f^2 + 138*C^3*a^4*b^5*c^5*d^10*f^2 - 12*C^3*a^4*b^5*c^7*d^8*f^2 - 144*C
^3*a^5*b^4*c^2*d^13*f^2 - 108*C^3*a^5*b^4*c^4*d^11*f^2 + 52*C^3*a^5*b^4*c^6
*d^9*f^2 - 8*C^3*a^6*b^3*c^3*d^12*f^2 - 88*C^3*a^6*b^3*c^5*d^10*f^2 + 56*C^
3*a^7*b^2*c^2*d^13*f^2 + 72*C^3*a^7*b^2*c^4*d^11*f^2))/(b^3*f^5) - ((-(b^9*
f^2 + 2*a^2*b^7*f^2 + a^4*b^5*f^2)*(C^2*a^7*d^3 - C^2*a^4*b^3*c^3 - 3*C^2*a
^6*b*c*d^2 + 3*C^2*a^5*b^2*c^2*d))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(4
*C^2*a^3*b^7*d^13*f^2 + 2*C^2*a^5*b^5*d^13*f^2 + 28*C^2*b^10*c^3*d^10*f^2 -
 10*C^2*b^10*c^5*d^8*f^2 - 14*C^2*a*b^9*d^13*f^2 - 16*C^2*a^9*b*d^13*f^2 +
8*C^2*a^10*c*d^12*f^2 + 22*C^2*b^10*c*d^12*f^2 + 20*C^2*a*b^9*c^2*d^11*f^2
+ 50*C^2*a*b^9*c^4*d^9*f^2 - 28*C^2*a^2*b^8*c*d^12*f^2 - 2*C^2*a^4*b^6*c*d^
12*f^2 + 56*C^2*a^8*b^2*c*d^12*f^2 - 32*C^2*a^9*b*c^2*d^11*f^2 + 8*C^2*a^2*
b^8*c^3*d^10*f^2 + 4*C^2*a^2*b^8*c^5*d^8*f^2 - 24*C^2*a^3*b^7*c^2*d^11*f^2
+ 4*C^2*a^3*b^7*c^4*d^9*f^2 + 12*C^2*a^4*b^6*c^3*d^10*f^2 - 10*C^2*a^4*b^6*
c^5*d^8*f^2 - 12*C^2*a^5*b^5*c^2*d^11*f^2 + 18*C^2*a^5*b^5*c^4*d^9*f^2 + 16
*C^2*a^6*b^4*c^3*d^10*f^2 + 8*C^2*a^6*b^4*c^5*d^8*f^2 - 64*C^2*a^7*b^3*c^2*
d^11*f^2 - 32*C^2*a^7*b^3*c^4*d^9*f^2 + 48*C^2*a^8*b^2*c^3*d^10*f^2))/(b^3*
f^4) + ((-(b^9*f^2 + 2*a^2*b^7*f^2 + a^4*b^5*f^2)*(C^2*a^7*d^3 - C^2*a^4*b^
3*c^3 - 3*C^2*a^6*b*c*d^2 + 3*C^2*a^5*b^2*c^2*d))^(1/2)*((32*(12*C*a^2*b^9*
d^12*f^4 + 24*C*a^4*b^7*d^12*f^4 + 12*C*a^6*b^5*d^12*f^4 + 4*C*b^11*c^2*d^1
0*f^4 + 4*C*b^11*c^4*d^8*f^4 - 16*C*a*b^10*c^3*d^9*f^4 - 32*C*a^3*b^8*c*d^1
1*f^4 - 16*C*a^5*b^6*c*d^11*f^4 + 20*C*a^2*b^9*c^2*d^10*f^4 + 8*C*a^2*b^9*c
^4*d^8*f^4 - 32*C*a^3*b^8*c^3*d^9*f^4 + 28*C*a^4*b^7*c^2*d^10*f^4 + 4*C*a^4
*b^7*c^4*d^8*f^4 - 16*C*a^5*b^6*c^3*d^9*f^4 + 12*C*a^6*b^5*c^2*d^10*f^4 - 1
6*C*a*b^10*c*d^11*f^4))/(b^3*f^5) + (32*(-(b^9*f^2 + 2*a^2*b^7*f^2 + a^4*b^
5*f^2)*(C^2*a^7*d^3 - C^2*a^4*b^3*c^3 - 3*C^2*a^6*b*c*d^2 + 3*C^2*a^5*b^2*c
^2*d))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(16*b^12*d^10*f^4 + 16*a^2*b^10*d^1
0*f^4 - 16*a^4*b^8*d^10*f^4 - 16*a^6*b^6*d^10*f^4 + 24*b^12*c^2*d^8*f^4 + 4
0*a^2*b^10*c^2*d^8*f^4 + 8*a^4*b^8*c^2*d^8*f^4 - 8*a^6*b^6*c^2*d^8*f^4 + 8*
a*b^11*c*d^9*f^4 + 24*a^3*b^9*c*d^9*f^4 + 24*a^5*b^7*c*d^9*f^4 + 8*a^7*b^5*
c*d^9*f^4))/(b^8*f^6*(a^2 + b^2)^2)))/(b^5*f^2*(a^2 + b^2)^2)))/(b^5*f^2*(a
^2 + b^2)^2)))/(b^5*f^2*(a^2 + b^2)^2)))/(b^5*f^2*(a^2 + b^2)^2))*(-(b^9*f
^2 + 2*a^2*b^7*f^2 + a^4*b^5*f^2)*(C^2*a^7*d^3 - C^2*a^4*b^3*c^3 - 3*C^2*a^
6*b*c*d^2 + 3*C^2*a^5*b^2*c^2*d))^(1/2)*2i)/(b^5*f^2*(a^2 + b^2)^2)
```

3.102 $\int \frac{(c+d\tan(e+fx))^{3/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx$

Optimal result 1370
Rubi [A] (verified) 1371
Mathematica [B] (verified) 1375
Maple [B] (verified) 1377
Fricas [F(-1)] 1377
Sympy [F] 1377
Maxima [F(-2)] 1378
Giac [F(-1)] 1378
Mupad [F(-1)] 1378

Optimal result

Integrand size = 47, antiderivative size = 372

$$\int \frac{(c+d\tan(e+fx))^{3/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx =$$
$$-\frac{(iA+B-iC)(c-id)^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(a-ib)^2f}$$
$$-\frac{(B-i(A-C))(c+id)^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(a+ib)^2f}$$
$$+\frac{\sqrt{bc-ad}(a^3bBd-3a^4Cd-b^4(2Bc+3Ad)-ab^3(4Ac-4cC-5Bd)+a^2b^2(2Bc+(A-7C)d))\operatorname{arctanh}\big(}{b^{5/2}\left(a^2+b^2\right)^2f}$$
$$+\frac{(Ab^2-abB+3a^2C+2b^2C)\,d\sqrt{c+d\tan(e+fx)}}{b^2\,(a^2+b^2)\,f}$$
$$-\frac{(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{3/2}}{b\,(a^2+b^2)\,f(a+b\tan(e+fx))}$$

```
[Out] -(I*A+B-I*C)*(c-I*d)^(3/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(a
-I*b)^2/f-(B-I*(A-C))*(c+I*d)^(3/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^
(1/2))/(a+I*b)^2/f+(a^3*b*B*d-3*a^4*C*d-b^4*(3*A*d+2*B*c)-a*b^3*(4*A*c-5*B*
d-4*C*c)+a^2*b^2*(2*B*c+(A-7*C)*d))*arctanh(b^(1/2)*(c+d*tan(f*x+e))^(1/2)/
(-a*d+b*c)^(1/2))*(-a*d+b*c)^(1/2)/b^(5/2)/(a^2+b^2)^2/f+(A*b^2-B*a*b+3*C*a
^2+2*C*b^2)*d*(c+d*tan(f*x+e))^(1/2)/b^2/(a^2+b^2)/f-(A*b^2-a*(B*b-C*a))*(c
+d*tan(f*x+e))^(3/2)/b/(a^2+b^2)/f/(a+b*tan(f*x+e))
```

Rubi [A] (verified)

Time = 2.67 (sec) , antiderivative size = 372, normalized size of antiderivative = 1.00, number of steps used = 13, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}} = 0.170$, Rules used = {3726, 3728, 3734, 3620, 3618, 65, 214, 3715}

$$\int \frac{(c+d\tan(e+fx))^{3/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx =$$
$$-\frac{(Ab^2-a(bB-aC))(c+d\tan(e+fx))^{3/2}}{bf(a^2+b^2)(a+b\tan(e+fx))}$$
$$+\frac{d(3a^2C-abB+Ab^2+2b^2C)\sqrt{c+d\tan(e+fx)}}{b^2f(a^2+b^2)}$$
$$+\frac{\sqrt{bc-ad}(-3a^4Cd+a^3bBd+a^2b^2(d(A-7C)+2Bc)-ab^3(4Ac-5Bd-4cC)-b^4(3Ad+2Bc))\operatorname{arctan}}{b^{5/2}f(a^2+b^2)^2}$$
$$-\frac{(c-id)^{3/2}(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f(a-ib)^2}$$
$$-\frac{(c+id)^{3/2}(B-i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f(a+ib)^2}$$

```
[In] Int[((c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a
 + b*Tan[e + f*x])^2,x]
```

```
[Out] -(((I*A + B - I*C)*(c - I*d)^(3/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c
- I*d]])/((a - I*b)^2*f)) - ((B - I*(A - C))*(c + I*d)^(3/2)*ArcTanh[Sqrt[c
 + d*Tan[e + f*x]]/Sqrt[c + I*d]])/((a + I*b)^2*f) + (Sqrt[b*c - a*d]*(a^3*
b*B*d - 3*a^4*C*d - b^4*(2*B*c + 3*A*d) - a*b^3*(4*A*c - 4*c*C - 5*B*d) + a
^2*b^2*(2*B*c + (A - 7*C)*d))*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sq
rt[b*c - a*d]])/(b^(5/2)*(a^2 + b^2)^2*f) + ((A*b^2 - a*b*B + 3*a^2*C + 2*b
^2*C)*d*Sqrt[c + d*Tan[e + f*x]])/(b^2*(a^2 + b^2)*f) - ((A*b^2 - a*(b*B -
a*C))*(c + d*Tan[e + f*x])^(3/2))/(b*(a^2 + b^2)*f*(a + b*Tan[e + f*x]))
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3715

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_) + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :>
 Dist[A/f, Subst[Int[(a + b*x)^m*(c + d*x)^n, x], x, Tan[e + f*x]], x] /; F
reeQ[{a, b, c, d, e, f, A, C, m, n}, x] && EqQ[A, C]
```

Rule 3726

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*d^2 + c*(c*C - B*d))*(a + b*Tan[e +
 f*x])^m*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
```

```
, 0] && NeQ[a, 0])))
```

Rule 3734

```
Int[(((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2))/((a_.) + (b_.)*tan[(e_.)
+ (f_.)*(x_)]), x_Symbol] :> Dist[1/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n
*Simp[b*B + a*(A - C) + (a*B - b*(A - C))*Tan[e + f*x], x], x], x] + Dist[(
A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n*((1 + Tan[e
+ f*x]^2)/(a + b*Tan[e + f*x])), x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C
, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&
 !GtQ[n, 0] &&  !LeQ[n, -1]
```

Rubi steps

$$\text{integral} = -\frac{\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^{3/2}}{b\left(a^2+b^2\right)f(a+b\tan(e+fx))} + \frac{\int \frac{\sqrt{c+d\tan(e+fx)}\left(\frac{1}{2}\left(2(bB-aC)\left(bc-\frac{3ad}{2}\right)+2Ab\left(ac+\frac{3bd}{2}\right)\right)-b((A-C)(bc-ad)-B(ac+bd))\tan(e+fx)+\frac{1}{2}\left(Ab^2-abB+3a^2C+2b^2C\right)d\tan \cdots}{a+b\tan(e+fx)}}{b\left(a^2+b^2\right)}$$

$$= \frac{\left(Ab^2-abB+3a^2C+2b^2C\right)d\sqrt{c+d\tan(e+fx)}}{b^2\left(a^2+b^2\right)f} - \frac{\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^{3/2}}{b\left(a^2+b^2\right)f(a+b\tan(e+fx))} + \frac{2\int \frac{\frac{1}{4}\left(-a\left(Ab^2-abB+3a^2C+2b^2C\right)d^2+bc((bB-aC)(2bc-3ad)+Ab(2ac+3bd))\right)+\frac{1}{2}b^2\left(2aAcd-2acCd-Ab\left(c^2-d^2\right)+aB\left(c^2-d^2\right)+\cdots\right.}{(a+b\tan(e+fx))\sqrt{\cdots}}}{b^2\left(a^2+\cdots\right)}$$

$$= \frac{\left(Ab^2-abB+3a^2C+2b^2C\right)d\sqrt{c+d\tan(e+fx)}}{b^2\left(a^2+b^2\right)f} - \frac{\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^{3/2}}{b\left(a^2+b^2\right)f(a+b\tan(e+fx))} + \frac{2\int \frac{-\frac{1}{2}b^2\left(a^2\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-b^2\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-2ab\left(2c(A-C)d+B\left(c^2-d^2\right)\right)\right)+\frac{1}{2}b^2\left(2ab\left(c^2C+2\cdots\right.\right.}{\sqrt{c+d\tan(e+fx)}}}{b^2\left(a^2+b^2\right)^2} - \frac{\left((bc-ad)\left(a^3bBd-3a^4Cd-b^4(2Bc+3Ad)-ab^3(4Ac-4cC-5Bd)+a^2b^2(2Bc+(A-7C\cdots\right.\right.}{2b^2\left(a^2+b^2\right)^2}$$

$$= \frac{(Ab^2 - abB + 3a^2C + 2b^2C)\, d\sqrt{c + d\tan(e + fx)}}{b^2\,(a^2 + b^2)\, f} - \frac{(Ab^2 - a(bB - aC))\,(c + d\tan(e + fx))^{3/2}}{b\,(a^2 + b^2)\, f(a + b\tan(e + fx))} + \frac{((A - iB - C)(c - id)^2)\int \frac{1 + i\tan(e+fx)}{\sqrt{c + d\tan(e+fx)}}\,dx}{2(a - ib)^2} + \frac{((A + iB - C)(c + id)^2)\int \frac{1 - i\tan(e+fx)}{\sqrt{c + d\tan(e+fx)}}\,dx}{2(a + ib)^2} - \frac{((bc - ad)\,(a^3bBd - 3a^4Cd - b^4(2Bc + 3Ad) - ab^3(4Ac - 4cC - 5Bd) + a^2b^2(2Bc + (A - 7C)}{2b^2\,(a^2 + b^2)^2\, f}$$

$$= \frac{(Ab^2 - abB + 3a^2C + 2b^2C)\, d\sqrt{c + d\tan(e + fx)}}{b^2\,(a^2 + b^2)\, f} - \frac{(Ab^2 - a(bB - aC))\,(c + d\tan(e + fx))^{3/2}}{b\,(a^2 + b^2)\, f(a + b\tan(e + fx))} + \frac{((iA + B - iC)(c - id)^2)\,\mathrm{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c - idx}}\,dx, x, i\tan(e + fx)\right)}{2(a - ib)^2 f} - \frac{(i(A + iB - C)(c + id)^2)\,\mathrm{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c + idx}}\,dx, x, -i\tan(e + fx)\right)}{2(a + ib)^2 f} - \frac{((bc - ad)\,(a^3bBd - 3a^4Cd - b^4(2Bc + 3Ad) - ab^3(4Ac - 4cC - 5Bd) + a^2b^2(2Bc + (A - 7C)}{b^2\,(a^2 + b^2)^2\, df}$$

$$= \frac{\sqrt{bc - ad}(a^3bBd - 3a^4Cd - b^4(2Bc + 3Ad) - ab^3(4Ac - 4cC - 5Bd) + a^2b^2(2Bc + (A - 7C)d))}{b^{5/2}\,(a^2 + b^2)^2\, f} + \frac{(Ab^2 - abB + 3a^2C + 2b^2C)\, d\sqrt{c + d\tan(e + fx)}}{b^2\,(a^2 + b^2)\, f} - \frac{(Ab^2 - a(bB - aC))\,(c + d\tan(e + fx))^{3/2}}{b\,(a^2 + b^2)\, f(a + b\tan(e + fx))} - \frac{((A - iB - C)(c - id)^2)\,\mathrm{Subst}\left(\int \frac{1}{-1 - \frac{ic}{d} + \frac{ix^2}{d}}\,dx, x, \sqrt{c + d\tan(e + fx)}\right)}{(a - ib)^2 df} - \frac{((A + iB - C)(c + id)^2)\,\mathrm{Subst}\left(\int \frac{1}{-1 + \frac{ic}{d} - \frac{ix^2}{d}}\,dx, x, \sqrt{c + d\tan(e + fx)}\right)}{(a + ib)^2 df}$$

$$= -\frac{(iA+B-iC)(c-id)^{3/2}\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(a-ib)^2 f} - \frac{(B-i(A-C))(c+id)^{3/2}\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(a+ib)^2 f} + \frac{\sqrt{bc-ad}(a^3bBd - 3a^4Cd - b^4(2Bc+3Ad) - ab^3(4Ac-4cC-5Bd) + a^2b^2(2Bc+(A-7C)}{b^{5/2}(a^2+b^2)^2 f} + \frac{(Ab^2-abB+3a^2C+2b^2C)\,d\sqrt{c+d\tan(e+fx)}}{b^2(a^2+b^2)f} - \frac{(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{3/2}}{b(a^2+b^2)f(a+b\tan(e+fx))}$$

Mathematica [B] (verified)

Both result and optimal contain complex but leaf count is larger than twice the leaf count of optimal. 2738 vs. 2(372) = 744.

Time = 6.61 (sec) , antiderivative size = 2738, normalized size of antiderivative = 7.36

$$\int \frac{(c+d\tan(e+fx))^{3/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx = \text{Result too large to show}$$

```
[In] Integrate[((c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/(a + b*Tan[e + f*x])^2,x]
```

```
[Out] (2*C*(c + d*Tan[e + f*x])^(3/2))/(b*f*(a + b*Tan[e + f*x])) + (2*(-(((3*b*c
*C + b*B*d - 3*a*C*d)*Sqrt[c + d*Tan[e + f*x]])/(b*f*(a + b*Tan[e + f*x])))
 - (2*(-((((I*Sqrt[c - I*d]*(b*(b*c - a*d)*((b*(-(A*b^2*c^2) + 3*a^2*C*d^2
+ 2*b^2*c*(2*c*C + B*d) - a*b*d*(6*c*C + B*d)))/4 + (a*b^2*(2*c*(A - C)*d +
 B*(c^2 - d^2)))/4 - (b*(3*a^2*C*d^2 - a*b*d*(6*c*C + B*d) + b^2*(3*c^2*C -
 (A - C)*d^2)))/4) + a*((((b^2*d)/2 - a*(b*c - a*d))*(-(A*b^2*c^2) + 3*a^2*
C*d^2 + 2*b^2*c*(2*c*C + B*d) - a*b*d*(6*c*C + B*d)))/4 + (-(b*c) + (a*d)/2
)*(-1/4*(b^3*(2*c*(A - C)*d + B*(c^2 - d^2))) - (a*(3*a^2*C*d^2 - a*b*d*(6*
c*C + B*d) + b^2*(3*c^2*C - (A - C)*d^2)))/4) - (d*((b^2*(-(A*b^2*c^2) + 3*
a^2*C*d^2 + 2*b^2*c*(2*c*C + B*d) - a*b*d*(6*c*C + B*d)))/4 - a*(-1/4*(b^3*
(2*c*(A - C)*d + B*(c^2 - d^2))) - (a*(3*a^2*C*d^2 - a*b*d*(6*c*C + B*d) +
b^2*(3*c^2*C - (A - C)*d^2)))/4)))/2) - I*(a*(b*c - a*d)*((b*(-(A*b^2*c^2)
+ 3*a^2*C*d^2 + 2*b^2*c*(2*c*C + B*d) - a*b*d*(6*c*C + B*d)))/4 + (a*b^2*(2
*c*(A - C)*d + B*(c^2 - d^2)))/4 - (b*(3*a^2*C*d^2 - a*b*d*(6*c*C + B*d) +
b^2*(3*c^2*C - (A - C)*d^2)))/4) - b*((((b^2*d)/2 - a*(b*c - a*d))*(-(A*b^2
*c^2) + 3*a^2*C*d^2 + 2*b^2*c*(2*c*C + B*d) - a*b*d*(6*c*C + B*d)))/4 + (-(
b*c) + (a*d)/2)*(-1/4*(b^3*(2*c*(A - C)*d + B*(c^2 - d^2))) - (a*(3*a^2*C*d
^2 - a*b*d*(6*c*C + B*d) + b^2*(3*c^2*C - (A - C)*d^2)))/4) - (d*((b^2*(-(A
*b^2*c^2) + 3*a^2*C*d^2 + 2*b^2*c*(2*c*C + B*d) - a*b*d*(6*c*C + B*d)))/4 -
```

```
 a*(-1/4*(b^3*(2*c*(A - C)*d + B*(c^2 - d^2))) - (a*(3*a^2*C*d^2 - a*b*d*(6
*c*C + B*d) + b^2*(3*c^2*C - (A - C)*d^2)))/4)))/2)))*ArcTanh[Sqrt[c + d*Ta
n[e + f*x]]/Sqrt[c - I*d]])/((-c + I*d)*f) - (I*Sqrt[c + I*d]*(b*(b*c - a*d
)*((b*(-(A*b^2*c^2) + 3*a^2*C*d^2 + 2*b^2*c*(2*c*C + B*d) - a*b*d*(6*c*C +
B*d)))/4 + (a*b^2*(2*c*(A - C)*d + B*(c^2 - d^2)))/4 - (b*(3*a^2*C*d^2 - a*
b*d*(6*c*C + B*d) + b^2*(3*c^2*C - (A - C)*d^2)))/4) + a*((((b^2*d)/2 - a*(
b*c - a*d))*(-(A*b^2*c^2) + 3*a^2*C*d^2 + 2*b^2*c*(2*c*C + B*d) - a*b*d*(6*
c*C + B*d)))/4 + (-(b*c) + (a*d)/2)*(-1/4*(b^3*(2*c*(A - C)*d + B*(c^2 - d^
2))) - (a*(3*a^2*C*d^2 - a*b*d*(6*c*C + B*d) + b^2*(3*c^2*C - (A - C)*d^2))
)/4) - (d*((b^2*(-(A*b^2*c^2) + 3*a^2*C*d^2 + 2*b^2*c*(2*c*C + B*d) - a*b*d
*(6*c*C + B*d)))/4 - a*(-1/4*(b^3*(2*c*(A - C)*d + B*(c^2 - d^2))) - (a*(3*
a^2*C*d^2 - a*b*d*(6*c*C + B*d) + b^2*(3*c^2*C - (A - C)*d^2)))/4)))/2) + I
*(a*(b*c - a*d)*((b*(-(A*b^2*c^2) + 3*a^2*C*d^2 + 2*b^2*c*(2*c*C + B*d) - a
*b*d*(6*c*C + B*d)))/4 + (a*b^2*(2*c*(A - C)*d + B*(c^2 - d^2)))/4 - (b*(3*
a^2*C*d^2 - a*b*d*(6*c*C + B*d) + b^2*(3*c^2*C - (A - C)*d^2)))/4) - b*((((
b^2*d)/2 - a*(b*c - a*d))*(-(A*b^2*c^2) + 3*a^2*C*d^2 + 2*b^2*c*(2*c*C + B*
d) - a*b*d*(6*c*C + B*d)))/4 + (-(b*c) + (a*d)/2)*(-1/4*(b^3*(2*c*(A - C)*d
 + B*(c^2 - d^2))) - (a*(3*a^2*C*d^2 - a*b*d*(6*c*C + B*d) + b^2*(3*c^2*C -
 (A - C)*d^2)))/4) - (d*((b^2*(-(A*b^2*c^2) + 3*a^2*C*d^2 + 2*b^2*c*(2*c*C
+ B*d) - a*b*d*(6*c*C + B*d)))/4 - a*(-1/4*(b^3*(2*c*(A - C)*d + B*(c^2 - d
^2))) - (a*(3*a^2*C*d^2 - a*b*d*(6*c*C + B*d) + b^2*(3*c^2*C - (A - C)*d^2)
))/4)))/2)))*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]])/((-c - I*d)*f
))/(a^2 + b^2) + (2*Sqrt[b*c - a*d]*(-(a*b*(b*c - a*d)*((b*(-(A*b^2*c^2) +
3*a^2*C*d^2 + 2*b^2*c*(2*c*C + B*d) - a*b*d*(6*c*C + B*d)))/4 + (a*b^2*(2*c
*(A - C)*d + B*(c^2 - d^2)))/4 - (b*(3*a^2*C*d^2 - a*b*d*(6*c*C + B*d) + b^
2*(3*c^2*C - (A - C)*d^2)))/4)) + (a^2*d*((b^2*(-(A*b^2*c^2) + 3*a^2*C*d^2
+ 2*b^2*c*(2*c*C + B*d) - a*b*d*(6*c*C + B*d)))/4 - a*(-1/4*(b^3*(2*c*(A -
C)*d + B*(c^2 - d^2))) - (a*(3*a^2*C*d^2 - a*b*d*(6*c*C + B*d) + b^2*(3*c^2
*C - (A - C)*d^2)))/4)))/2 + b^2*((((b^2*d)/2 - a*(b*c - a*d))*(-(A*b^2*c^2
) + 3*a^2*C*d^2 + 2*b^2*c*(2*c*C + B*d) - a*b*d*(6*c*C + B*d)))/4 + (-(b*c)
 + (a*d)/2)*(-1/4*(b^3*(2*c*(A - C)*d + B*(c^2 - d^2))) - (a*(3*a^2*C*d^2 -
 a*b*d*(6*c*C + B*d) + b^2*(3*c^2*C - (A - C)*d^2)))/4)))*ArcTanh[(Sqrt[b]*
Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*d]])/(Sqrt[b]*(a^2 + b^2)*(-(b*c) +
a*d)*f))/((a^2 + b^2)*(b*c - a*d))) - (((b^2*(-(A*b^2*c^2) + 3*a^2*C*d^2 +
2*b^2*c*(2*c*C + B*d) - a*b*d*(6*c*C + B*d)))/4 - a*(-1/4*(b^3*(2*c*(A - C)
*d + B*(c^2 - d^2))) - (a*(3*a^2*C*d^2 - a*b*d*(6*c*C + B*d) + b^2*(3*c^2*C
 - (A - C)*d^2)))/4))*Sqrt[c + d*Tan[e + f*x]])/((a^2 + b^2)*(b*c - a*d)*f*
(a + b*Tan[e + f*x]))))/b))/b
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 9864 vs. 2(337) = 674.

Time = 0.17 (sec) , antiderivative size = 9865, normalized size of antiderivative = 26.52

method	result	size
derivativedivides	Expression too large to display	9865
default	Expression too large to display	9865

```
[In] int((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^2,x,method=_RETURNVERBOSE)
```

```
[Out] result too large to display
```

Fricas [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^2}\,dx=\text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^2,x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^2}\,dx=\int \frac{(c+d\tan{(e+fx)})^{\frac{3}{2}}\left(A+B\tan{(\right.}}{(a+b\tan{(e+}}$$

```
[In] integrate((c+d*tan(f*x+e))**(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e))**2,x)
```

```
[Out] Integral((c + d*tan(e + f*x))**(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)**
2)/(a + b*tan(e + f*x))**2, x)
```

Maxima [F(-2)]

Exception generated.

$$\int \frac{(c+d\tan(e+fx))^{3/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx = \text{Exception raised: ValueError}$$

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^2,x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 detail
```

Giac [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{3/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^2,x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{3/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx = \text{Hanged}$$

```
[In] int(((c + d*tan(e + f*x))^(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x))^2,x)
```

```
[Out] \text{Hanged}
```

3.103 $\int \frac{(c+d\tan(e+fx))^{3/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx$

Optimal result 1379
Rubi [A] (verified) 1380
Mathematica [B] (verified) 1384
Maple [B] (verified) 1384
Fricas [F(-1)] 1384
Sympy [F] 1385
Maxima [F(-2)] 1385
Giac [F(-1)] 1385
Mupad [F(-1)] 1386

Optimal result

Integrand size = 47, antiderivative size = 532

$$\int \frac{(c+d\tan(e+fx))^{3/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx =$$
$$-\frac{(A-iB-C)(c-id)^{3/2}\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(ia+b)^3f}$$
$$+\frac{(A+iB-C)(c+id)^{3/2}\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(ia-b)^3f}$$
$$-\frac{(a^5bBd^2+3a^6Cd^2+a^4b^2d(4Bc+3(A+2C)d)-b^6(8Ac^2-8c^2C-12Bcd-3Ad^2)+a^2b^4(24Ac^2-24c}{}$$
$$-\frac{(a^3bBd+3a^4Cd+b^4(4Bc+3Ad)+ab^3(8Ac-8cC-7Bd)-a^2b^2(4Bc+5Ad-11Cd))\sqrt{c+d\tan(e}}{4b^2(a^2+b^2)^2f(a+b\tan(e+fx))}$$
$$-\frac{(Ab^2-a(bB-aC))(c+d\tan(e+fx))^{3/2}}{2b(a^2+b^2)f(a+b\tan(e+fx))^2}$$

```
[Out] -(A-I*B-C)*(c-I*d)^(3/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(I*a
+b)^3/f+(A+I*B-C)*(c+I*d)^(3/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2
))/(I*a-b)^3/f-1/4*(a^5*b*B*d^2+3*a^6*C*d^2+a^4*b^2*d*(4*B*c+3*(A+2*C)*d)-b
^6*(8*A*c^2-3*A*d^2-12*B*c*d-8*C*c^2)+a^2*b^4*(24*A*c^2-26*A*d^2-48*B*c*d-2
4*C*c^2+35*C*d^2)-2*a^3*b^3*(12*c*(A-C)*d+B*(4*c^2-9*d^2))+a*b^5*(40*c*(A-C
)*d+3*B*(8*c^2-5*d^2)))*arctanh(b^(1/2)*(c+d*tan(f*x+e))^(1/2)/(-a*d+b*c)^(
1/2))/b^(5/2)/(a^2+b^2)^3/f/(-a*d+b*c)^(1/2)-1/4*(a^3*b*B*d+3*a^4*C*d+b^4*(
3*A*d+4*B*c)+a*b^3*(8*A*c-7*B*d-8*C*c)-a^2*b^2*(5*A*d+4*B*c-11*C*d))*(c+d*t
an(f*x+e))^(1/2)/b^2/(a^2+b^2)^2/f/(a+b*tan(f*x+e))-1/2*(A*b^2-a*(B*b-C*a))
*(c+d*tan(f*x+e))^(3/2)/b/(a^2+b^2)/f/(a+b*tan(f*x+e))^2
```

Rubi [A] (verified)

Time = 5.13 (sec) , antiderivative size = 532, normalized size of antiderivative = 1.00, number of steps used = 13, number of rules used = 7, $\frac{\text{number of rules}}{\text{integrand size}} = 0.149$, Rules used = {3726, 3734, 3620, 3618, 65, 214, 3715}

$$\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^3}\,dx =$$

$$-\frac{(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{3/2}}{2bf\,(a^2+b^2)\,(a+b\tan(e+fx))^2}$$

$$-\frac{\sqrt{c+d\tan(e+fx)}(3a^4Cd+a^3bBd-a^2b^2(5Ad+4Bc-11Cd)+ab^3(8Ac-7Bd-8cC)+b^4(3Ad+4B}{4b^2f\,(a^2+b^2)^2\,(a+b\tan(e+fx))}$$

$$-\frac{(3a^6Cd^2+a^5bBd^2+a^4b^2d(3d(A+2C)+4Bc)-2a^3b^3(12cd(A-C)+B(4c^2-9d^2))+a^2b^4(24Ac^2-26}{}$$

$$-\frac{(c-id)^{3/2}(A-iB-C)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f(b+ia)^3}$$

$$+\frac{(c+id)^{3/2}(A+iB-C)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f(-b+ia)^3}$$

```
[In] Int[((c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a
 + b*Tan[e + f*x])^3,x]
```

```
[Out] -(((A - I*B - C)*(c - I*d)^(3/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c -
I*d]])/((I*a + b)^3*f)) + ((A + I*B - C)*(c + I*d)^(3/2)*ArcTanh[Sqrt[c + d
*Tan[e + f*x]]/Sqrt[c + I*d]])/((I*a - b)^3*f) - ((a^5*b*B*d^2 + 3*a^6*C*d^
2 + a^4*b^2*d*(4*B*c + 3*(A + 2*C)*d) - b^6*(8*A*c^2 - 8*c^2*C - 12*B*c*d -
 3*A*d^2) + a^2*b^4*(24*A*c^2 - 24*c^2*C - 48*B*c*d - 26*A*d^2 + 35*C*d^2)
- 2*a^3*b^3*(12*c*(A - C)*d + B*(4*c^2 - 9*d^2)) + a*b^5*(40*c*(A - C)*d +
3*B*(8*c^2 - 5*d^2)))*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c -
 a*d]])/(4*b^(5/2)*(a^2 + b^2)^3*Sqrt[b*c - a*d]*f) - ((a^3*b*B*d + 3*a^4*C
*d + b^4*(4*B*c + 3*A*d) + a*b^3*(8*A*c - 8*c*C - 7*B*d) - a^2*b^2*(4*B*c +
 5*A*d - 11*C*d))*Sqrt[c + d*Tan[e + f*x]])/(4*b^2*(a^2 + b^2)^2*f*(a + b*T
an[e + f*x])) - ((A*b^2 - a*(b*B - a*C))*(c + d*Tan[e + f*x])^(3/2))/(2*b*(
a^2 + b^2)*f*(a + b*Tan[e + f*x])^2)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3715

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_) + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :>
 Dist[A/f, Subst[Int[(a + b*x)^m*(c + d*x)^n, x], x, Tan[e + f*x]], x] /; F
reeQ[{a, b, c, d, e, f, A, C, m, n}, x] && EqQ[A, C]
```

Rule 3726

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*d^2 + c*(c*C - B*d))*(a + b*Tan[e +
 f*x])^m*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3734

```
Int[(((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2))/((a_.) + (b_.)*tan[(e_.)
+ (f_.)*(x_)]), x_Symbol] :> Dist[1/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n
*Simp[b*B + a*(A - C) + (a*B - b*(A - C))*Tan[e + f*x], x], x], x] + Dist[(
```

```
A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n*((1 + Tan[e
+ f*x]^2)/(a + b*Tan[e + f*x])), x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C
, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&
 !GtQ[n, 0] &&  !LeQ[n, -1]
```

Rubi steps

$$\text{integral} = -\frac{(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{3/2}}{2b\,(a^2+b^2)\,f(a+b\tan(e+fx))^2} + \frac{\int \frac{\sqrt{c+d\tan(e+fx)}\left(\frac{1}{2}\left(2(bB-aC)\left(2bc-\frac{3ad}{2}\right)+2Ab\left(2ac+\frac{3bd}{2}\right)\right)-2b((A-C)(bc-ad)-B(ac+bd))\tan(e+fx)-\frac{1}{2}(Ab^2-abB-3a^2C-4b^2C)\,d\,\tan}{(a+b\tan(e+fx))^2}}{2b\,(a^2+b^2)}$$

$$= -\frac{(a^3bBd+3a^4Cd+b^4(4Bc+3Ad)+ab^3(8Ac-8cC-7Bd)-a^2b^2(4Bc+5Ad-11Cd))\sqrt{c+}}{4b^2\,(a^2+b^2)^2\,f(a+b\tan(e+fx))} - \frac{(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{3/2}}{2b\,(a^2+b^2)\,f(a+b\tan(e+fx))^2} + \frac{\int \frac{1}{4}(b(2ac+bd)(3a^2Cd+b^2(4Bc+3Ad)+ab(4Ac-4cC-3Bd))-(2bc-ad)(a^2bBd+3a^3Cd+Ab^2(4bc-5ad)-4b^3(cC+Bd)-4ab^2(Bc-}{}$$

$$= -\frac{(a^3bBd+3a^4Cd+b^4(4Bc+3Ad)+ab^3(8Ac-8cC-7Bd)-a^2b^2(4Bc+5Ad-11Cd))\sqrt{c+}}{4b^2\,(a^2+b^2)^2\,f(a+b\tan(e+fx))} - \frac{(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{3/2}}{2b\,(a^2+b^2)\,f(a+b\tan(e+fx))^2} + \frac{\int \frac{-2b^2(a^3(c^2C+2Bcd-Cd^2-A(c^2-d^2))-3ab^2(c^2C+2Bcd-Cd^2-A(c^2-d^2))-3a^2b(2c(A-C)d+B(c^2-d^2))+b^3(2c(A-C)d+B(}{}}{} + \frac{(a^5bBd^2+3a^6Cd^2+a^4b^2d(4Bc+3(A+2C)d)-b^6(8Ac^2-8c^2C-12Bcd-3Ad^2)+a^2b^4(24A}{}$$

$$= -\frac{(a^3bBd+3a^4Cd+b^4(4Bc+3Ad)+ab^3(8Ac-8cC-7Bd)-a^2b^2(4Bc+5Ad-11Cd))\sqrt{c+}}{4b^2\,(a^2+b^2)^2\,f(a+b\tan(e+fx))} - \frac{(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{3/2}}{2b\,(a^2+b^2)\,f(a+b\tan(e+fx))^2} + \frac{((A-iB-C)(c-id)^2)\int \frac{1+i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(a-ib)^3} + \frac{((A+iB-C)(c+id)^2)\int \frac{1-i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(a+ib)^3} + \frac{(a^5bBd^2+3a^6Cd^2+a^4b^2d(4Bc+3(A+2C)d)-b^6(8Ac^2-8c^2C-12Bcd-3Ad^2)+a^2b^4(24A}{}$$

$$= -\frac{\left(a^3bBd+3a^4Cd+b^4(4Bc+3Ad)+ab^3(8Ac-8cC-7Bd)-a^2b^2(4Bc+5Ad-11Cd)\right)\sqrt{c}}{4b^2\left(a^2+b^2\right)^2 f(a+b\tan(e+fx))}$$

$$-\frac{\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^{3/2}}{2b\left(a^2+b^2\right)f(a+b\tan(e+fx))^2}$$

$$+\frac{((A-iB-C)(c-id)^2)\operatorname{Subst}\left(\int\frac{1}{(-1+x)\sqrt{c-idx}}\,dx,x,i\tan(e+fx)\right)}{2(ia+b)^3f}$$

$$-\frac{((A+iB-C)(c+id)^2)\operatorname{Subst}\left(\int\frac{1}{(-1+x)\sqrt{c+idx}}\,dx,x,-i\tan(e+fx)\right)}{2(ia-b)^3f}$$

$$+\frac{\left(a^5bBd^2+3a^6Cd^2+a^4b^2d(4Bc+3(A+2C)d)-b^6(8Ac^2-8c^2C-12Bcd-3Ad^2)+a^2b^4(24\right.}{}$$

$$= -\frac{\left(a^5bBd^2+3a^6Cd^2+a^4b^2d(4Bc+3(A+2C)d)-b^6(8Ac^2-8c^2C-12Bcd-3Ad^2)+a^2b^4(24\right.}{}$$

$$-\frac{\left(a^3bBd+3a^4Cd+b^4(4Bc+3Ad)+ab^3(8Ac-8cC-7Bd)-a^2b^2(4Bc+5Ad-11Cd)\right)\sqrt{c}}{4b^2\left(a^2+b^2\right)^2 f(a+b\tan(e+fx))}$$

$$-\frac{\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^{3/2}}{2b\left(a^2+b^2\right)f(a+b\tan(e+fx))^2}$$

$$-\frac{((A-iB-C)(c-id)^2)\operatorname{Subst}\left(\int\frac{1}{-1-\frac{ic}{d}+\frac{ix^2}{d}}\,dx,x,\sqrt{c+d\tan(e+fx)}\right)}{(a-ib)^3df}$$

$$-\frac{((A+iB-C)(c+id)^2)\operatorname{Subst}\left(\int\frac{1}{-1+\frac{ic}{d}-\frac{ix^2}{d}}\,dx,x,\sqrt{c+d\tan(e+fx)}\right)}{(a+ib)^3df}$$

$$= -\frac{(A-iB-C)(c-id)^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(ia+b)^3f}$$

$$+\frac{(A+iB-C)(c+id)^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(ia-b)^3f}$$

$$-\frac{\left(a^5bBd^2+3a^6Cd^2+a^4b^2d(4Bc+3(A+2C)d)-b^6(8Ac^2-8c^2C-12Bcd-3Ad^2)+a^2b^4(24\right.}{}$$

$$-\frac{\left(a^3bBd+3a^4Cd+b^4(4Bc+3Ad)+ab^3(8Ac-8cC-7Bd)-a^2b^2(4Bc+5Ad-11Cd)\right)\sqrt{c}}{4b^2\left(a^2+b^2\right)^2 f(a+b\tan(e+fx))}$$

$$-\frac{\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^{3/2}}{2b\left(a^2+b^2\right)f(a+b\tan(e+fx))^2}$$

Mathematica [B] (verified)

Both result and optimal contain complex but leaf count is larger than twice the leaf count of optimal. 7678 vs. $2(532) = 1064$.

Time = 7.18 (sec) , antiderivative size = 7678, normalized size of antiderivative = 14.43

$$\int \frac{(c+d\tan(e+fx))^{3/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx = \text{Result too large to show}$$

```
[In] Integrate[((c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/(a + b*Tan[e + f*x])^3,x]
```

```
[Out] Result too large to show
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 14440 vs. $2(492) = 984$.

Time = 0.16 (sec) , antiderivative size = 14441, normalized size of antiderivative = 27.14

method	result	size
derivativedivides	Expression too large to display	14441
default	Expression too large to display	14441

```
[In] int((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^3,x,method=_RETURNVERBOSE)
```

```
[Out] result too large to display
```

Fricas [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{3/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^3,x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int \frac{(c+d\tan(e+fx))^{3/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx = \int \frac{(c+d\tan{(e+fx)})^{\frac{3}{2}}\,(A+B\tan{(}}{(a+b\tan{(e+}}$$

[In] integrate((c+d*tan(f*x+e))**(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan (f*x+e))**3,x)

[Out] Integral((c + d*tan(e + f*x))**(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)** 2)/(a + b*tan(e + f*x))**3, x)

Maxima [F(-2)]

Exception generated.

$$\int \frac{(c+d\tan(e+fx))^{3/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx = \text{Exception raised: ValueError}$$

[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f *x+e))^3,x, algorithm="maxima")

[Out] Exception raised: ValueError >> Computation failed since Maxima requested a dditional constraints; using the 'assume' command before evaluation *may* h elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more detail

Giac [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{3/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx = \text{Timed out}$$

[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f *x+e))^3,x, algorithm="giac")

[Out] Timed out

Mupad [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{3/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx = \text{Hanged}$$

```
[In] int(((c + d*tan(e + f*x))^(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x))^3,x)
```

```
[Out] \text{Hanged}
```

3.104 $\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e\right.$

Optimal result 1387
Rubi [A] (verified) 1388
Mathematica [A] (verified) 1394
Maple [B] (verified) 1394
Fricas [B] (verification not implemented) 1395
Sympy [F] 1395
Maxima [F(-1)] 1396
Giac [F(-1)] 1396
Mupad [F(-1)] 1396

Optimal result

Integrand size = 47, antiderivative size = 503

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx =$$

$$-\frac{(a-ib)^2(iA+B-iC)(c-id)^{5/2}\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f}$$

$$+\frac{(a+ib)^2(iA-B-iC)(c+id)^{5/2}\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f}$$

$$-\frac{2(2ab(c^2C+2Bcd-Cd^2-A(c^2-d^2))-a^2(2c(A-C)d+B(c^2-d^2))+b^2(2c(A-C)d+B(c^2-d^2))}{f}$$

$$+\frac{2(2ab(Ac-cC-Bd)+a^2(Bc+(A-C)d)-b^2(Bc+(A-C)d))\,(c+d\tan(e+fx))^{3/2}}{3f}$$

$$+\frac{2(a^2B-b^2B+2ab(A-C))\,(c+d\tan(e+fx))^{5/2}}{5f}$$

$$+\frac{2(36a^2Cd^2-22abd(2cC-9Bd)+b^2(8c^2C-22Bcd+99(A-C)d^2))\,(c+d\tan(e+fx))^{7/2}}{693d^3f}$$

$$-\frac{2b(4bcC-11bBd-4aCd)\tan(e+fx)(c+d\tan(e+fx))^{7/2}}{99d^2f}$$

$$+\frac{2C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{7/2}}{11df}$$

```
[Out] -(a-I*b)^2*(I*A+B-I*C)*(c-I*d)^(5/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)
^(1/2))/f+(a+I*b)^2*(I*A-B-I*C)*(c+I*d)^(5/2)*arctanh((c+d*tan(f*x+e))^(1/2
)/(c+I*d)^(1/2))/f-2*(2*a*b*(c^2*C+2*B*c*d-C*d^2-A*(c^2-d^2))-a^2*(2*c*(A-C
)*d+B*(c^2-d^2))+b^2*(2*c*(A-C)*d+B*(c^2-d^2)))*(c+d*tan(f*x+e))^(1/2)/f+2/
3*(2*a*b*(A*c-B*d-C*c)+a^2*(B*c+(A-C)*d)-b^2*(B*c+(A-C)*d))*(c+d*tan(f*x+e)
)^(3/2)/f+2/5*(B*a^2-B*b^2+2*a*b*(A-C))*(c+d*tan(f*x+e))^(5/2)/f+2/693*(36*
```

```
a^2*C*d^2-22*a*b*d*(-9*B*d+2*C*c)+b^2*(8*c^2*C-22*B*c*d+99*(A-C)*d^2))*(c+d
*tan(f*x+e))^(7/2)/d^3/f-2/99*b*(-11*B*b*d-4*C*a*d+4*C*b*c)*tan(f*x+e)*(c+d
*tan(f*x+e))^(7/2)/d^2/f+2/11*C*(a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^(7/2)/d
/f
```

Rubi [A] (verified)

Time = 2.59 (sec) , antiderivative size = 503, normalized size of antiderivative = 1.00, number of steps used = 13, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.170, Rules used = {3728, 3718, 3711, 3609, 3620, 3618, 65, 214}

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx =$$

$$-\frac{2\sqrt{c+d\tan(e+fx)}(-(a^2(2cd(A-C)+B(c^2-d^2)))+2ab(-A(c^2-d^2)+2Bcd+c^2C-Cd^2)+b^2(2c}{f}$$

$$+\frac{2(c+d\tan(e+fx))^{7/2}\left(36a^2Cd^2-22abd(2cC-9Bd)+b^2(99d^2(A-C)-22Bcd+8c^2C)\right)}{693d^3f}$$

$$+\frac{2(a^2B+2ab(A-C)-b^2B)\,(c+d\tan(e+fx))^{5/2}}{5f}$$

$$+\frac{2(c+d\tan(e+fx))^{3/2}\left(a^2(d(A-C)+Bc)+2ab(Ac-Bd-cC)-b^2(d(A-C)+Bc)\right)}{3f}$$

$$-\frac{(a-ib)^2(c-id)^{5/2}(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f}$$

$$+\frac{(a+ib)^2(c+id)^{5/2}(iA-B-iC)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f}$$

$$-\frac{2b\tan(e+fx)(-4aCd-11bBd+4bcC)(c+d\tan(e+fx))^{7/2}}{99d^2f}$$

$$+\frac{2C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{7/2}}{11df}$$

```
[In] Int[(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] +
 C*Tan[e + f*x]^2),x]
```

```
[Out] -(((a - I*b)^2*(I*A + B - I*C)*(c - I*d)^(5/2)*ArcTanh[Sqrt[c + d*Tan[e + f
*x]]/Sqrt[c - I*d]])/f) + ((a + I*b)^2*(I*A - B - I*C)*(c + I*d)^(5/2)*ArcT
anh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]])/f - (2*(2*a*b*(c^2*C + 2*B*c*d
 - C*d^2 - A*(c^2 - d^2)) - a^2*(2*c*(A - C)*d + B*(c^2 - d^2)) + b^2*(2*c*
(A - C)*d + B*(c^2 - d^2)))*Sqrt[c + d*Tan[e + f*x]])/f + (2*(2*a*b*(A*c -
c*C - B*d) + a^2*(B*c + (A - C)*d) - b^2*(B*c + (A - C)*d))*(c + d*Tan[e +
f*x])^(3/2))/(3*f) + (2*(a^2*B - b^2*B + 2*a*b*(A - C))*(c + d*Tan[e + f*x]
)^(5/2))/(5*f) + (2*(36*a^2*C*d^2 - 22*a*b*d*(2*c*C - 9*B*d) + b^2*(8*c^2*C
 - 22*B*c*d + 99*(A - C)*d^2))*(c + d*Tan[e + f*x])^(7/2))/(693*d^3*f) - (2
```

```
*b*(4*b*c*C - 11*b*B*d - 4*a*C*d)*Tan[e + f*x]*(c + d*Tan[e + f*x])^(7/2))/
(99*d^2*f) + (2*C*(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])^(7/2))/(11*d*
f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3609

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[d*((a + b*Tan[e + f*x])^m/(f*m)), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\text{integral} = \frac{2C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{7/2}}{11df} + \frac{2\int(a+b\tan(e+fx))(c+d\tan(e+fx))^{5/2}\left(\frac{1}{2}(-4bcC+a(11A-7C)d)+\frac{11}{2}(Ab+aB-bC)d\tan(e+\right.}{11d}$$

$$= -\frac{2b(4bcC-11bBd-4aCd)\tan(e+fx)(c+d\tan(e+fx))^{7/2}}{99d^2f} + \frac{2C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{7/2}}{11df} - \frac{4\int(c+d\tan(e+fx))^{5/2}\left(\frac{1}{4}\left(44abcCd-9a^2(11A-7C)d^2-4b^2\left(2c^2C-\frac{11Bcd}{2}\right)\right)-\frac{99}{4}(a^2B-b^2\right.}{}$$

$$= \frac{2(36a^2Cd^2-22abd(2cC-9Bd)+b^2(8c^2C-22Bcd+99(A-C)d^2))(c+d\tan(e+fx))^{7/2}}{693d^3f} - \frac{2b(4bcC-11bBd-4aCd)\tan(e+fx)(c+d\tan(e+fx))^{7/2}}{99d^2f} + \frac{2C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{7/2}}{11df} - \frac{4\int(c+d\tan(e+fx))^{5/2}\left(\frac{99}{4}(2abB-a^2(A-C)+b^2(A-C))d^2-\frac{99}{4}(a^2B-b^2B+2ab(A-C\right.}{99d^2}$$

$$= \frac{2(a^2B - b^2B + 2ab(A-C))\,(c+d\tan(e+fx))^{5/2}}{5f}$$
$$+ \frac{2(36a^2Cd^2 - 22abd(2cC - 9Bd) + b^2(8c^2C - 22Bcd + 99(A-C)d^2))\,(c+d\tan(e+fx))^{7/2}}{693d^3f}$$
$$- \frac{2b(4bcC - 11bBd - 4aCd)\tan(e+fx)(c+d\tan(e+fx))^{7/2}}{99d^2f}$$
$$+ \frac{2C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{7/2}}{11df}$$
$$- \frac{4\int (c+d\tan(e+fx))^{3/2}\left(-\frac{99}{4}d^2(a^2(Ac - cC - Bd) - b^2(Ac - cC - Bd) - 2ab(Bc + (A-C\right.}{99}$$

$$= \frac{2(2ab(Ac - cC - Bd) + a^2(Bc + (A-C)d) - b^2(Bc + (A-C)d))\,(c+d\tan(e+fx))^{3/2}}{3f}$$
$$+ \frac{2(a^2B - b^2B + 2ab(A-C))\,(c+d\tan(e+fx))^{5/2}}{5f}$$
$$+ \frac{2(36a^2Cd^2 - 22abd(2cC - 9Bd) + b^2(8c^2C - 22Bcd + 99(A-C)d^2))\,(c+d\tan(e+fx))^{7/2}}{693d^3f}$$
$$- \frac{2b(4bcC - 11bBd - 4aCd)\tan(e+fx)(c+d\tan(e+fx))^{7/2}}{99d^2f}$$
$$+ \frac{2C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{7/2}}{11df}$$
$$- \frac{4\int \sqrt{c+d\tan(e+fx)}\left(\frac{99}{4}d^2(a^2(c^2C + 2Bcd - Cd^2 - A(c^2-d^2)) - b^2(c^2C + 2Bcd - Cd^2 - \right.}{}$$

$$=$$
$$- \frac{2(2ab(c^2C + 2Bcd - Cd^2 - A(c^2-d^2)) - a^2(2c(A-C)d + B(c^2-d^2)) + b^2(2c(A-C)d + B}{f}$$
$$+ \frac{2(2ab(Ac - cC - Bd) + a^2(Bc + (A-C)d) - b^2(Bc + (A-C)d))\,(c+d\tan(e+fx))^{3/2}}{3f}$$
$$+ \frac{2(a^2B - b^2B + 2ab(A-C))\,(c+d\tan(e+fx))^{5/2}}{5f}$$
$$+ \frac{2(36a^2Cd^2 - 22abd(2cC - 9Bd) + b^2(8c^2C - 22Bcd + 99(A-C)d^2))\,(c+d\tan(e+fx))^{7/2}}{693d^3f}$$
$$- \frac{2b(4bcC - 11bBd - 4aCd)\tan(e+fx)(c+d\tan(e+fx))^{7/2}}{99d^2f}$$
$$+ \frac{2C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{7/2}}{11df}$$
$$- \frac{4\int \frac{-\frac{99}{4}d^2(a^2(Ac^3 - c^3C - 3Bc^2d - 3Acd^2 + 3cCd^2 + Bd^3) + b^2(c^3C + 3Bc^2d - 3cCd^2 - Bd^3 - A(c^3 - 3cd^2)) - 2ab((A-C)d(3c^2-d^2) -}{}}{}$$

$$= -\frac{2(2ab(c^2C+2Bcd-Cd^2-A(c^2-d^2))-a^2(2c(A-C)d+B(c^2-d^2))+b^2(2c(A-C)d+B(}{f}$$
$$+\frac{2(2ab(Ac-cC-Bd)+a^2(Bc+(A-C)d)-b^2(Bc+(A-C)d))\,(c+d\tan(e+fx))^{3/2}}{3f}$$
$$+\frac{2(a^2B-b^2B+2ab(A-C))\,(c+d\tan(e+fx))^{5/2}}{5f}$$
$$+\frac{2(36a^2Cd^2-22abd(2cC-9Bd)+b^2(8c^2C-22Bcd+99(A-C)d^2))\,(c+d\tan(e+fx))^{7/2}}{693d^3f}$$
$$-\frac{2b(4bcC-11bBd-4aCd)\tan(e+fx)(c+d\tan(e+fx))^{7/2}}{99d^2f}$$
$$+\frac{2C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{7/2}}{11df}$$
$$+\frac{1}{2}\left((a-ib)^2(A-iB-C)(c-id)^3\right)\int\frac{1+i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx+\frac{1}{2}\left((a+ib)^2(A+iB-C)(c+id)^3\right)\int$$

$$= -\frac{2(2ab(c^2C+2Bcd-Cd^2-A(c^2-d^2))-a^2(2c(A-C)d+B(c^2-d^2))+b^2(2c(A-C)d+B(}{f}$$
$$+\frac{2(2ab(Ac-cC-Bd)+a^2(Bc+(A-C)d)-b^2(Bc+(A-C)d))\,(c+d\tan(e+fx))^{3/2}}{3f}$$
$$+\frac{2(a^2B-b^2B+2ab(A-C))\,(c+d\tan(e+fx))^{5/2}}{5f}$$
$$+\frac{2(36a^2Cd^2-22abd(2cC-9Bd)+b^2(8c^2C-22Bcd+99(A-C)d^2))\,(c+d\tan(e+fx))^{7/2}}{693d^3f}$$
$$-\frac{2b(4bcC-11bBd-4aCd)\tan(e+fx)(c+d\tan(e+fx))^{7/2}}{99d^2f}$$
$$+\frac{2C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{7/2}}{11df}$$
$$+\frac{\left((a-ib)^2(iA+B-iC)(c-id)^3\right)\operatorname{Subst}\left(\int\frac{1}{(-1+x)\sqrt{c-idx}}\,dx,x,i\tan(e+fx)\right)}{2f}$$
$$-\frac{\left(i(a+ib)^2(A+iB-C)(c+id)^3\right)\operatorname{Subst}\left(\int\frac{1}{(-1+x)\sqrt{c+idx}}\,dx,x,-i\tan(e+fx)\right)}{2f}$$

$$= -\frac{2(2ab(c^2C+2Bcd-Cd^2-A(c^2-d^2))-a^2(2c(A-C)d+B(c^2-d^2))+b^2(2c(A-C)d+B}{f}$$
$$+\frac{2(2ab(Ac-cC-Bd)+a^2(Bc+(A-C)d)-b^2(Bc+(A-C)d))\,(c+d\tan(e+fx))^{3/2}}{3f}$$
$$+\frac{2(a^2B-b^2B+2ab(A-C))\,(c+d\tan(e+fx))^{5/2}}{5f}$$
$$+\frac{2(36a^2Cd^2-22abd(2cC-9Bd)+b^2(8c^2C-22Bcd+99(A-C)d^2))\,(c+d\tan(e+fx))^{7/2}}{693d^3f}$$
$$-\frac{2b(4bcC-11bBd-4aCd)\tan(e+fx)(c+d\tan(e+fx))^{7/2}}{99d^2f}$$
$$+\frac{2C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{7/2}}{11df}$$
$$-\frac{((a-ib)^2(A-iB-C)(c-id)^3)\operatorname{Subst}\left(\int\frac{1}{-1-\frac{ic}{d}+\frac{ix^2}{d}}\,dx,x,\sqrt{c+d\tan(e+fx)}\right)}{df}$$
$$-\frac{((a+ib)^2(A+iB-C)(c+id)^3)\operatorname{Subst}\left(\int\frac{1}{-1+\frac{ic}{d}-\frac{ix^2}{d}}\,dx,x,\sqrt{c+d\tan(e+fx)}\right)}{df}$$

$$= -\frac{(a-ib)^2(iA+B-iC)(c-id)^{5/2}\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f}$$
$$-\frac{(a+ib)^2(B-i(A-C))(c+id)^{5/2}\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f}$$
$$-\frac{2(2ab(c^2C+2Bcd-Cd^2-A(c^2-d^2))-a^2(2c(A-C)d+B(c^2-d^2))+b^2(2c(A-C)d+B}{f}$$
$$+\frac{2(2ab(Ac-cC-Bd)+a^2(Bc+(A-C)d)-b^2(Bc+(A-C)d))\,(c+d\tan(e+fx))^{3/2}}{3f}$$
$$+\frac{2(a^2B-b^2B+2ab(A-C))\,(c+d\tan(e+fx))^{5/2}}{5f}$$
$$+\frac{2(36a^2Cd^2-22abd(2cC-9Bd)+b^2(8c^2C-22Bcd+99(A-C)d^2))\,(c+d\tan(e+fx))^{7/2}}{693d^3f}$$
$$-\frac{2b(4bcC-11bBd-4aCd)\tan(e+fx)(c+d\tan(e+fx))^{7/2}}{99d^2f}$$
$$+\frac{2C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{7/2}}{11df}$$

Mathematica [A] (verified)

Time = 6.58 (sec) , antiderivative size = 564, normalized size of antiderivative = 1.12

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\frac{2C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{7/2}}{11df}+\frac{2\left(\frac{b(-4bcC+11bBd+4aCd)\tan(e+fx)(c+d\tan(e+fx))^{7/2}}{9df}-\frac{2\left(\frac{\left(-36a^2Cd^2+22abd(2cC-9Bd)-b^2\left(8c^2C-22Bcd+99(A-C)d^2\right)\right)(c+d\tan(e+fx))^{7/2}}{14df}\right.}{}\right.}{}$$

```
[In] Integrate[(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e +
f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] (2*C*(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])^(7/2))/(11*d*f) + (2*((b*(
-4*b*c*C + 11*b*B*d + 4*a*C*d)*Tan[e + f*x]*(c + d*Tan[e + f*x])^(7/2))/(9*
d*f) - (2*(((-36*a^2*C*d^2 + 22*a*b*d*(2*c*C - 9*B*d) - b^2*(8*c^2*C - 22*B
*c*d + 99*(A - C)*d^2))*(c + d*Tan[e + f*x])^(7/2))/(14*d*f) + ((I/2)*(((99
*I)/4)*(a^2*B - b^2*B + 2*a*b*(A - C))*d^2 + (99*(2*a*b*B - a^2*(A - C) + b
^2*(A - C))*d^2)/4)*((2*(c + d*Tan[e + f*x])^(5/2))/5 + (c - I*d)*((2*(c +
d*Tan[e + f*x])^(3/2))/3 + (c - I*d)*((2*(c - I*d)^(3/2)*ArcTanh[Sqrt[c + d
*Tan[e + f*x]]/Sqrt[c - I*d]])/(-c + I*d) + 2*Sqrt[c + d*Tan[e + f*x]]))))/
f - ((I/2)*(((-99*I)/4)*(a^2*B - b^2*B + 2*a*b*(A - C))*d^2 + (99*(2*a*b*B
- a^2*(A - C) + b^2*(A - C))*d^2)/4)*((2*(c + d*Tan[e + f*x])^(5/2))/5 + (c
 + I*d)*((2*(c + d*Tan[e + f*x])^(3/2))/3 + (c + I*d)*((2*(c + I*d)^(3/2)*A
rcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]])/(-c - I*d) + 2*Sqrt[c + d*T
an[e + f*x]]))))/f))/(9*d)))/(11*d)
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 11279 vs. $2(459) = 918$.

Time = 0.44 (sec) , antiderivative size = 11280, normalized size of antiderivative = 22.43

method	result	size
parts	Expression too large to display	11280
derivativedivides	Expression too large to display	11478
default	Expression too large to display	11478

```
[In] int((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x,method=_RETURNVERBOSE)
```

```
[Out] result too large to display
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 91140 vs. $2(449) = 898$.

Time = 169.46 (sec) , antiderivative size = 91140, normalized size of antiderivative = 181.19

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f
*x+e)^2),x, algorithm="fricas")
```

```
[Out] Too large to include
```

Sympy [F]

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\int (a+b\tan{(e+fx)})^2\,(c+d\tan{(e+fx)})^{\frac{5}{2}}\left(A+B\tan{(e+fx)}+C\tan^2{(e+fx)}\right)\,dx$$

```
[In] integrate((a+b*tan(f*x+e))**2*(c+d*tan(f*x+e))**(5/2)*(A+B*tan(f*x+e)+C*tan
(f*x+e)**2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**2*(c + d*tan(e + f*x))**(5/2)*(A + B*tan(e +
 f*x) + C*tan(e + f*x)**2), x)
```

Maxima [F(-1)]

Timed out.

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f
*x+e)^2),x, algorithm="maxima")
```

```
[Out] Timed out
```

Giac [F(-1)]

Timed out.

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f
*x+e)^2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \text{Hanged}$$

```
[In] int((a + b*tan(e + f*x))^2*(c + d*tan(e + f*x))^(5/2)*(A + B*tan(e + f*x) +
 C*tan(e + f*x)^2),x)
```

```
[Out] \text{Hanged}
```

3.105 $\int (a+b\tan(e+fx))(c+d\tan(e+fx))^{5/2}\,(A+B\tan(e$

Optimal result 1397
Rubi [A] (verified) 1398
Mathematica [A] (verified) 1403
Maple [B] (verified) 1403
Fricas [B] (verification not implemented) 1404
Sympy [F] 1404
Maxima [F(-1)] 1404
Giac [F(-1)] 1405
Mupad [F(-1)] 1405

Optimal result

Integrand size = 45, antiderivative size = 353

$$\begin{aligned}
&\int (a+b\tan(e+fx))(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\\
&-\frac{(ia+b)(A-iB-C)(c-id)^{5/2}\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f}\\
&+\frac{(ia-b)(A+iB-C)(c+id)^{5/2}\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f}\\
&+\frac{2(a(Bc^2-2cCd-Bd^2)-b(c^2C+2Bcd-Cd^2)+A(2acd+b(c^2-d^2)))\sqrt{c+d\tan(e+fx)}}{f}\\
&+\frac{2(Abc+aBc-bcC+aAd-bBd-aCd)(c+d\tan(e+fx))^{3/2}}{3f}\\
&+\frac{2(Ab+aB-bC)(c+d\tan(e+fx))^{5/2}}{5f}\\
&-\frac{2(2bcC-9bBd-9aCd)(c+d\tan(e+fx))^{7/2}}{63d^2f}+\frac{2bC\tan(e+fx)(c+d\tan(e+fx))^{7/2}}{9df}
\end{aligned}$$

```
[Out] -(I*a+b)*(A-I*B-C)*(c-I*d)^(5/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/
2))/f+(I*a-b)*(A+I*B-C)*(c+I*d)^(5/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d
)^(1/2))/f+2*(a*(B*c^2-B*d^2-2*C*c*d)-b*(2*B*c*d+C*c^2-C*d^2)+A*(2*a*c*d+b*
(c^2-d^2)))*(c+d*tan(f*x+e))^(1/2)/f+2/3*(A*a*d+A*b*c+B*a*c-B*b*d-C*a*d-C*b
*c)*(c+d*tan(f*x+e))^(3/2)/f+2/5*(A*b+B*a-C*b)*(c+d*tan(f*x+e))^(5/2)/f-2/6
3*(-9*B*b*d-9*C*a*d+2*C*b*c)*(c+d*tan(f*x+e))^(7/2)/d^2/f+2/9*b*C*tan(f*x+e
)*(c+d*tan(f*x+e))^(7/2)/d/f
```

Rubi [A] (verified)

Time = 1.29 (sec) , antiderivative size = 351, normalized size of antiderivative = 0.99, number of steps used = 12, number of rules used = 7, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.156, Rules used = {3718, 3711, 3609, 3620, 3618, 65, 214}

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx =$$
$$-\frac{(b+ia)(c-id)^{5/2}(A-iB-C)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f}$$
$$+\frac{(-b+ia)(c+id)^{5/2}(A+iB-C)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f}$$
$$+\frac{2\sqrt{c+d\tan(e+fx)}(2aAcd+aB(c^2-d^2)-2acCd+Ab(c^2-d^2)-b(2Bcd+c^2C-Cd^2))}{f}$$
$$+\frac{2(aB+Ab-bC)(c+d\tan(e+fx))^{5/2}}{5f}$$
$$+\frac{2(c+d\tan(e+fx))^{3/2}(aAd+aBc-aCd+Abc-bBd-bcC)}{3f}$$
$$-\frac{2(-9aCd-9bBd+2bcC)(c+d\tan(e+fx))^{7/2}}{63d^2f}+\frac{2bC\tan(e+fx)(c+d\tan(e+fx))^{7/2}}{9df}$$

```
[In] Int[(a + b*Tan[e + f*x])*(c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C
*Tan[e + f*x]^2),x]
```

```
[Out] -((((I*a + b)*(A - I*B - C)*(c - I*d)^(5/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]
/Sqrt[c - I*d]])/f) + ((I*a - b)*(A + I*B - C)*(c + I*d)^(5/2)*ArcTanh[Sqrt
[c + d*Tan[e + f*x]]/Sqrt[c + I*d]])/f + (2*(2*a*A*c*d - 2*a*c*C*d + A*b*(c
^2 - d^2) + a*B*(c^2 - d^2) - b*(c^2*C + 2*B*c*d - C*d^2))*Sqrt[c + d*Tan[e
 + f*x]])/f + (2*(A*b*c + a*B*c - b*c*C + a*A*d - b*B*d - a*C*d)*(c + d*Tan
[e + f*x])^(3/2))/(3*f) + (2*(A*b + a*B - b*C)*(c + d*Tan[e + f*x])^(5/2))/
(5*f) - (2*(2*b*c*C - 9*b*B*d - 9*a*C*d)*(c + d*Tan[e + f*x])^(7/2))/(63*d^
2*f) + (2*b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(7/2))/(9*d*f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3609

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[d*((a + b*Tan[e + f*x])^m/(f*m)), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rubi steps

$$\text{integral} = \frac{2bC\tan(e+fx)(c+d\tan(e+fx))^{7/2}}{9df} - \frac{2\int(c+d\tan(e+fx))^{5/2}\left(\frac{1}{2}(2bcC-9aAd)-\frac{9}{2}(Ab+aB-bC)d\tan(e+fx)+\frac{1}{2}(2bcC-9bBd-9aCd\right.}{9d}$$

$$= -\frac{2(2bcC-9bBd-9aCd)(c+d\tan(e+fx))^{7/2}}{63d^2f} + \frac{2bC\tan(e+fx)(c+d\tan(e+fx))^{7/2}}{9df} - \frac{2\int(c+d\tan(e+fx))^{5/2}\left(\frac{9}{2}(bB-a(A-C))d-\frac{9}{2}(Ab+aB-bC)d\tan(e+fx)\right)dx}{9d}$$

$$= \frac{2(Ab+aB-bC)(c+d\tan(e+fx))^{5/2}}{5f} - \frac{2(2bcC-9bBd-9aCd)(c+d\tan(e+fx))^{7/2}}{63d^2f} + \frac{2bC\tan(e+fx)(c+d\tan(e+fx))^{7/2}}{9df} - \frac{2\int(c+d\tan(e+fx))^{3/2}\left(\frac{9}{2}d(bBc+b(A-C)d-a(Ac-cC-Bd))-\frac{9}{2}d(Abc+aBc-bcC+\right.}{9d}$$

$$= \frac{2(Abc+aBc-bcC+aAd-bBd-aCd)(c+d\tan(e+fx))^{3/2}}{3f} + \frac{2(Ab+aB-bC)(c+d\tan(e+fx))^{5/2}}{5f} - \frac{2(2bcC-9bBd-9aCd)(c+d\tan(e+fx))^{7/2}}{63d^2f} + \frac{2bC\tan(e+fx)(c+d\tan(e+fx))^{7/2}}{9df} - \frac{2\int\sqrt{c+d\tan(e+fx)}\left(\frac{9}{2}d(a(c^2C+2Bcd-Cd^2-A(c^2-d^2))+b(2c(A-C)d+B(c^2-d^2)))\right.}{9d}$$

$$= \frac{2(2aAcd-2acCd+Ab(c^2-d^2)+aB(c^2-d^2)-b(c^2C+2Bcd-Cd^2))\sqrt{c+d\tan(e+fx)}}{f} + \frac{2(Abc+aBc-bcC+aAd-bBd-aCd)(c+d\tan(e+fx))^{3/2}}{3f} + \frac{2(Ab+aB-bC)(c+d\tan(e+fx))^{5/2}}{5f} - \frac{2(2bcC-9bBd-9aCd)(c+d\tan(e+fx))^{7/2}}{63d^2f} + \frac{2bC\tan(e+fx)(c+d\tan(e+fx))^{7/2}}{9df} - \frac{2\int\frac{\frac{9}{2}d(a(c^3C+3Bc^2d-3cCd^2-Bd^3-A(c^3-3cd^2))+b((A-C)d(3c^2-d^2)+B(c^3-3cd^2)))-\frac{9}{2}d(A(bc^3+3ac^2d-3bcd^2-ad^3)-b(c^3C}{\sqrt{c+d\tan(e+fx)}}}{9d}$$

$$= \frac{2(2aAcd-2acCd+Ab(c^2-d^2)+aB(c^2-d^2)-b(c^2C+2Bcd-Cd^2))\sqrt{c+d\tan(e+fx)}}{f} + \frac{2(Abc+aBc-bcC+aAd-bBd-aCd)(c+d\tan(e+fx))^{3/2}}{3f} + \frac{2(Ab+aB-bC)(c+d\tan(e+fx))^{5/2}}{5f} - \frac{2(2bcC-9bBd-9aCd)(c+d\tan(e+fx))^{7/2}}{63d^2f} + \frac{2bC\tan(e+fx)(c+d\tan(e+fx))^{7/2}}{9df} + \frac{1}{2}\left((a-ib)(A-iB-C)(c-id)^3\right)\int\frac{1+i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx + \frac{1}{2}\left((a+ib)(A+iB-C)(c+id)^3\right)\int$$

$$= \frac{2(2aAcd-2acCd+Ab(c^2-d^2)+aB(c^2-d^2)-b(c^2C+2Bcd-Cd^2))\sqrt{c+d\tan(e+fx)}}{f} + \frac{2(Abc+aBc-bcC+aAd-bBd-aCd)(c+d\tan(e+fx))^{3/2}}{3f} + \frac{2(Ab+aB-bC)(c+d\tan(e+fx))^{5/2}}{5f} - \frac{2(2bcC-9bBd-9aCd)(c+d\tan(e+fx))^{7/2}}{63d^2f} + \frac{2bC\tan(e+fx)(c+d\tan(e+fx))^{7/2}}{9df} + \frac{((a-ib)(iA+B-iC)(c-id)^3)\operatorname{Subst}\left(\int\frac{1}{(-1+x)\sqrt{c-idx}}\,dx,x,i\tan(e+fx)\right)}{2f} - \frac{((ia-b)(A+iB-C)(c+id)^3)\operatorname{Subst}\left(\int\frac{1}{(-1+x)\sqrt{c+idx}}\,dx,x,-i\tan(e+fx)\right)}{2f}$$

$$= \frac{2(2aAcd - 2acCd + Ab(c^2 - d^2) + aB(c^2 - d^2) - b(c^2C + 2Bcd - Cd^2))\sqrt{c + d\tan(e + fx)}}{f} + \frac{2(Abc + aBc - bcC + aAd - bBd - aCd)(c + d\tan(e + fx))^{3/2}}{3f} + \frac{2(Ab + aB - bC)(c + d\tan(e + fx))^{5/2}}{5f} - \frac{2(2bcC - 9bBd - 9aCd)(c + d\tan(e + fx))^{7/2}}{63d^2f} + \frac{2bC\tan(e + fx)(c + d\tan(e + fx))^{7/2}}{9df} - \frac{((a - ib)(A - iB - C)(c - id)^3)\,\text{Subst}\left(\int \frac{1}{-1 - \frac{ic}{d} + \frac{ix^2}{d}}\,dx, x, \sqrt{c + d\tan(e + fx)}\right)}{df} - \frac{((a + ib)(A + iB - C)(c + id)^3)\,\text{Subst}\left(\int \frac{1}{-1 + \frac{ic}{d} - \frac{ix^2}{d}}\,dx, x, \sqrt{c + d\tan(e + fx)}\right)}{df}$$

$$= -\frac{(ia + b)(A - iB - C)(c - id)^{5/2}\text{arctanh}\left(\frac{\sqrt{c + d\tan(e + fx)}}{\sqrt{c - id}}\right)}{f} + \frac{(ia - b)(A + iB - C)(c + id)^{5/2}\text{arctanh}\left(\frac{\sqrt{c + d\tan(e + fx)}}{\sqrt{c + id}}\right)}{f} + \frac{2(2aAcd - 2acCd + Ab(c^2 - d^2) + aB(c^2 - d^2) - b(c^2C + 2Bcd - Cd^2))\sqrt{c + d\tan(e + fx)}}{f} + \frac{2(Abc + aBc - bcC + aAd - bBd - aCd)(c + d\tan(e + fx))^{3/2}}{3f} + \frac{2(Ab + aB - bC)(c + d\tan(e + fx))^{5/2}}{5f} - \frac{2(2bcC - 9bBd - 9aCd)(c + d\tan(e + fx))^{7/2}}{63d^2f} + \frac{2bC\tan(e + fx)(c + d\tan(e + fx))^{7/2}}{9df}$$

Mathematica [A] (verified)

Time = 5.51 (sec) , antiderivative size = 324, normalized size of antiderivative = 0.92

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \frac{\frac{2(-2bcC+9bBd+9aCd)(c+d\tan(e+fx))^{7/2}}{d}+14bC\tan(e+fx)(c+d\tan(e+fx))^{7/2}+\frac{63}{2}i(a}{}$$

```
[In] Integrate[(a + b*Tan[e + f*x])*(c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*
x] + C*Tan[e + f*x]^2),x]
```

```
[Out] ((2*(-2*b*c*C + 9*b*B*d + 9*a*C*d)*(c + d*Tan[e + f*x])^(7/2))/d + 14*b*C*T
an[e + f*x]*(c + d*Tan[e + f*x])^(7/2) + ((63*I)/2)*(a - I*b)*(A - I*B - C)
*d*((2*(c + d*Tan[e + f*x])^(5/2))/5 + (2*(c - I*d)*(-3*(c - I*d)^(3/2)*Arc
Tanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]] + Sqrt[c + d*Tan[e + f*x]]*(4*
c - (3*I)*d + d*Tan[e + f*x])))/3) - ((63*I)/2)*(a + I*b)*(A + I*B - C)*d*(
(2*(c + d*Tan[e + f*x])^(5/2))/5 + (2*(c + I*d)*(-3*(c + I*d)^(3/2)*ArcTanh
[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]] + Sqrt[c + d*Tan[e + f*x]]*(4*c +
(3*I)*d + d*Tan[e + f*x])))/3))/(63*d*f)
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 7293 vs. 2(315) = 630.

Time = 0.21 (sec) , antiderivative size = 7294, normalized size of antiderivative = 20.66

method	result	size
parts	Expression too large to display	7294
derivativedivides	Expression too large to display	7402
default	Expression too large to display	7402

```
[In] int((a+b*tan(f*x+e))*(c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)
,x,method=_RETURNVERBOSE)
```

```
[Out] result too large to display
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 48734 vs. 2(308) = 616.

Time = 37.57 (sec) , antiderivative size = 48734, normalized size of antiderivative = 138.06

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x, algorithm="fricas")
```

```
[Out] Too large to include
```

Sympy [F]

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \int (a+b\tan(e+fx))\,(c+d\tan(e+fx))^{\frac{5}{2}}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))**(5/2)*(A+B*tan(f*x+e)+C*tan(f*
x+e)**2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))*(c + d*tan(e + f*x))**(5/2)*(A + B*tan(e + f*
x) + C*tan(e + f*x)**2), x)
```

Maxima [F(-1)]

Timed out.

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x, algorithm="maxima")
```

```
[Out] Timed out
```

Giac [F(-1)]

Timed out.

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \text{Hanged}$$

```
[In] int((a + b*tan(e + f*x))*(c + d*tan(e + f*x))^(5/2)*(A + B*tan(e + f*x) + C
*tan(e + f*x)^2),x)
```

```
[Out] \text{Hanged}
```

3.106 $\int (c+d\tan(e+fx))^{5/2} (A + B\tan(e+fx) + C\tan^2(e$

Optimal result 1406
Rubi [A] (verified) 1407
Mathematica [A] (verified) 1410
Maple [B] (verified) 1410
Fricas [B] (verification not implemented) 1413
Sympy [F] 1413
Maxima [F] 1413
Giac [F(-1)] 1414
Mupad [B] (verification not implemented) 1414

Optimal result

Integrand size = 35, antiderivative size = 229

$$\begin{aligned}\int (c+d\tan(e+fx))^{5/2} \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \\ -\frac{(iA+B-iC)(c-id)^{5/2}\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f} \\ -\frac{(B-i(A-C))(c+id)^{5/2}\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f} \\ +\frac{2(2c(A-C)d+B(c^2-d^2))\sqrt{c+d\tan(e+fx)}}{f} \\ +\frac{2(Bc+(A-C)d)(c+d\tan(e+fx))^{3/2}}{3f} \\ +\frac{2B(c+d\tan(e+fx))^{5/2}}{5f}+\frac{2C(c+d\tan(e+fx))^{7/2}}{7df}\end{aligned}$$

```
[Out] -(I*A+B-I*C)*(c-I*d)^(5/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/f-
(B-I*(A-C))*(c+I*d)^(5/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))/f+2
*(2*c*(A-C)*d+B*(c^2-d^2))*(c+d*tan(f*x+e))^(1/2)/f+2/3*(B*c+(A-C)*d)*(c+d*
tan(f*x+e))^(3/2)/f+2/5*B*(c+d*tan(f*x+e))^(5/2)/f+2/7*C*(c+d*tan(f*x+e))^(
7/2)/d/f
```

Rubi [A] (verified)

Time = 0.66 (sec) , antiderivative size = 229, normalized size of antiderivative = 1.00, number of steps used = 11, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.171, Rules used = {3711, 3609, 3620, 3618, 65, 214}

$$\int (c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = -\frac{(c-id)^{5/2}(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f} - \frac{(c+id)^{5/2}(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f} + \frac{2(2cd(A-C)+B(c^2-d^2))\sqrt{c+d\tan(e+fx)}}{f} + \frac{2(d(A-C)+Bc)(c+d\tan(e+fx))^{3/2}}{3f} + \frac{2B(c+d\tan(e+fx))^{5/2}}{5f} + \frac{2C(c+d\tan(e+fx))^{7/2}}{7df}$$

```
[In] Int[(c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] -(((I*A + B - I*C)*(c - I*d)^(5/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c
- I*d]])/f) - ((B - I*(A - C))*(c + I*d)^(5/2)*ArcTanh[Sqrt[c + d*Tan[e + f
*x]]/Sqrt[c + I*d]])/f + (2*(2*c*(A - C)*d + B*(c^2 - d^2))*Sqrt[c + d*Tan[
e + f*x]])/f + (2*(B*c + (A - C)*d)*(c + d*Tan[e + f*x])^(3/2))/(3*f) + (2*
B*(c + d*Tan[e + f*x])^(5/2))/(5*f) + (2*C*(c + d*Tan[e + f*x])^(7/2))/(7*d
*f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3609

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[d*((a + b*Tan[e + f*x])^m/(f*m)), x] + Int
```

```
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \frac{2C(c+d\tan(e+fx))^{7/2}}{7df} + \int (A-C+B\tan(e+fx))(c+d\tan(e+fx))^{5/2}\,dx \\
&= \frac{2B(c+d\tan(e+fx))^{5/2}}{5f} + \frac{2C(c+d\tan(e+fx))^{7/2}}{7df} \\
&\quad + \int (c+d\tan(e+fx))^{3/2}(Ac-cC-Bd+(Bc+(A-C)d)\tan(e+fx))\,dx \\
&= \frac{2(Bc+(A-C)d)(c+d\tan(e+fx))^{3/2}}{3f} \\
&\quad + \frac{2B(c+d\tan(e+fx))^{5/2}}{5f} + \frac{2C(c+d\tan(e+fx))^{7/2}}{7df} \\
&\quad + \int \sqrt{c+d\tan(e+fx)}\left(-c^2C-2Bcd+Cd^2+A\left(c^2-d^2\right)+\left(2c(A-C)d+B\left(c^2-d^2\right)\right)\tan(e+fx)\right)
\end{aligned}$$

$$= \frac{2(2c(A-C)d+B(c^2-d^2))\sqrt{c+d\tan(e+fx)}}{f} + \frac{2(Bc+(A-C)d)(c+d\tan(e+fx))^{3/2}}{3f} + \frac{2B(c+d\tan(e+fx))^{5/2}}{5f} + \frac{2C(c+d\tan(e+fx))^{7/2}}{7df} + \int \frac{-c^3C-3Bc^2d+3cCd^2+Bd^3+A(c^3-3cd^2)+((A-C)d(3c^2-d^2)+B(c^3-3cd^2))\tan(}{\sqrt{c+d\tan(e+fx)}}$$

$$= \frac{2(2c(A-C)d+B(c^2-d^2))\sqrt{c+d\tan(e+fx)}}{f} + \frac{2(Bc+(A-C)d)(c+d\tan(e+fx))^{3/2}}{3f} + \frac{2B(c+d\tan(e+fx))^{5/2}}{5f} + \frac{2C(c+d\tan(e+fx))^{7/2}}{7df} + \frac{1}{2}\left((A-iB-C)(c-id)^3\right)\int \frac{1+i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}dx + \frac{1}{2}\left((A+iB-C)(c+id)^3\right)\int \frac{1-i\tan(e+}{\sqrt{c+d\tan(e-}}$$

$$= \frac{2(2c(A-C)d+B(c^2-d^2))\sqrt{c+d\tan(e+fx)}}{f} + \frac{2(Bc+(A-C)d)(c+d\tan(e+fx))^{3/2}}{3f} + \frac{2B(c+d\tan(e+fx))^{5/2}}{5f} + \frac{2C(c+d\tan(e+fx))^{7/2}}{7df} + \frac{((iA+B-iC)(c-id)^3)\operatorname{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c-idx}}dx, x, i\tan(e+fx)\right)}{2f} - \frac{(i(A+iB-C)(c+id)^3)\operatorname{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c+idx}}dx, x, -i\tan(e+fx)\right)}{2f}$$

$$= \frac{2(2c(A-C)d+B(c^2-d^2))\sqrt{c+d\tan(e+fx)}}{f} + \frac{2(Bc+(A-C)d)(c+d\tan(e+fx))^{3/2}}{3f} + \frac{2B(c+d\tan(e+fx))^{5/2}}{5f} + \frac{2C(c+d\tan(e+fx))^{7/2}}{7df} - \frac{((A-iB-C)(c-id)^3)\operatorname{Subst}\left(\int \frac{1}{-1-\frac{ic}{d}+\frac{ix^2}{d}}dx, x, \sqrt{c+d\tan(e+fx)}\right)}{df} - \frac{((A+iB-C)(c+id)^3)\operatorname{Subst}\left(\int \frac{1}{-1+\frac{ic}{d}-\frac{ix^2}{d}}dx, x, \sqrt{c+d\tan(e+fx)}\right)}{df}$$

$$= -\frac{(iA+B-iC)(c-id)^{5/2}\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f} - \frac{(B-i(A-C))(c+id)^{5/2}\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f} + \frac{2(2c(A-C)d+B(c^2-d^2))\sqrt{c+d\tan(e+fx)}}{f} + \frac{2(Bc+(A-C)d)(c+d\tan(e+fx))^{3/2}}{3f} + \frac{2B(c+d\tan(e+fx))^{5/2}}{5f} + \frac{2C(c+d\tan(e+fx))^{7/2}}{7df}$$

Mathematica [A] (verified)

Time = 2.20 (sec) , antiderivative size = 262, normalized size of antiderivative = 1.14

$$\int (c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \frac{\frac{4C(c+d\tan(e+fx))^{7/2}}{d}+7i(A-iB-C)\left(\frac{2}{5}(c+d\tan(e+fx))^{5/2}+\frac{2}{3}(c-id)\left(-3(c-id\right.\right.}{}$$

```
[In] Integrate[(c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2
),x]
```

```
[Out] ((4*C*(c + d*Tan[e + f*x])^(7/2))/d + (7*I)*(A - I*B - C)*((2*(c + d*Tan[e
+ f*x])^(5/2))/5 + (2*(c - I*d)*(-3*(c - I*d)^(3/2)*ArcTanh[Sqrt[c + d*Tan[
e + f*x]]/Sqrt[c - I*d]] + Sqrt[c + d*Tan[e + f*x]]*(4*c - (3*I)*d + d*Tan[
e + f*x])))/3) - (7*I)*(A + I*B - C)*((2*(c + d*Tan[e + f*x])^(5/2))/5 + (2
*(c + I*d)*(-3*(c + I*d)^(3/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*
d]] + Sqrt[c + d*Tan[e + f*x]]*(4*c + (3*I)*d + d*Tan[e + f*x])))/3))/(14*f
)
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 3561 vs. 2(196) = 392.

Time = 0.15 (sec) , antiderivative size = 3562, normalized size of antiderivative = 15.55

method	result	size
parts	Expression too large to display	3562
derivativedivides	Expression too large to display	3614
default	Expression too large to display	3614

```
[In] int((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x,method=_RETURN
VERBOSE)
```

```
[Out] A*(2/3/f*d*(c+d*tan(f*x+e))^(3/2)+4/f*d*(c+d*tan(f*x+e))^(1/2)*c+1/4/f/d*ln
(d*tan(f*x+e)+c-(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d
^2)^(1/2))*(c^2+d^2)^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c^2-1/4/f*d*ln(d*t
an(f*x+e)+c-(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^
(1/2))*(c^2+d^2)^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-1/4/f/d*ln(d*tan(f*x+e
)+c-(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(
2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c^3+3/4/f*d*ln(d*tan(f*x+e)+c-(c+d*tan(f*x+e))
^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)+2*
c)^(1/2)*c+3/f*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(
1/2)-(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*c^2-1/f*
d^3/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)-(2*(c^2+
d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))-2/f*d/(2*(c^2+d^2)^(1
/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)-(2*(c^2+d^2)^(1/2)+2*c)^(1/
2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*(c^2+d^2)^(1/2)*c-1/4/f/d*ln(d*tan(f*x+e
)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(
c^2+d^2)^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c^2+1/4/f*d*ln(d*tan(f*x+e)+c+
(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(c^2+
d^2)^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+1/4/f/d*ln(d*tan(f*x+e)+c+(c+d*tan
(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(2*(c^2+d^2)^
(1/2)+2*c)^(1/2)*c^3-3/4/f*d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c
^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c+3
/f*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2
+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*c^2-1/f*d^3/(2*(c^2+
d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2
*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))-2/f*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/
2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+
d^2)^(1/2)-2*c)^(1/2))*(c^2+d^2)^(1/2)*c)-3/f*d^2/(2*(c^2+d^2)^(1/2)-2*c)^(
1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^
2+d^2)^(1/2)-2*c)^(1/2))*B*c-1/2*B/f*ln(d*tan(f*x+e)+c-(c+d*tan(f*x+e))^(1/
2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)+2*c)^(
1/2)*(c^2+d^2)^(1/2)*c-B/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan
(f*x+e))^(1/2)-(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)
)*(c^2+d^2)^(1/2)*c^2+B/f*d^2/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*
tan(f*x+e))^(1/2)-(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1
/2))*(c^2+d^2)^(1/2)+1/2/f*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2
+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^
2+d^2)^(1/2)*c-1/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))
^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*(c^2
+d^2)^(1/2)*c^2+1/f*d^2/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*
x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B
*(c^2+d^2)^(1/2)+B/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e
))^(1/2)-(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*c^3-
```

3*B/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)-(2*(c^
2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*c*d^2+1/f/(2*(c^2+d
^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*
c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*c^3+2/5*B*(c+d*tan(f*x+e))^(5/2)
/f+2/3/f*B*c*(c+d*tan(f*x+e))^(3/2)+2/f*(c+d*tan(f*x+e))^(1/2)*B*c^2-2/f*d^
2*(c+d*tan(f*x+e))^(1/2)*B+3/4*B/f*ln(d*tan(f*x+e)+c-(c+d*tan(f*x+e))^(1/2)
(2(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)+2*c)^(1/
2)*c^2-1/4*B/f*d^2*ln(d*tan(f*x+e)+c-(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1
/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-3/4/f*ln(d*ta
n(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(
1/2))*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c^2+1/4/f*d^2*ln(d*tan(f*x+e)+c+(c+d*
tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*B*(2*(c^2+
d^2)^(1/2)+2*c)^(1/2)+C*(2/7/f/d*(c+d*tan(f*x+e))^(7/2)-2/3/f*d*(c+d*tan(f*
x+e))^(3/2)-4/f*d*(c+d*tan(f*x+e))^(1/2)*c-1/4/f/d*ln(d*tan(f*x+e)+c-(c+d*t
an(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(c^2+d^2)^(
1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c^2+1/4/f*d*ln(d*tan(f*x+e)+c-(c+d*tan(f
x+e))^(1/2)(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(c^2+d^2)^(1/2)
(2(c^2+d^2)^(1/2)+2*c)^(1/2)+1/4/f/d*ln(d*tan(f*x+e)+c-(c+d*tan(f*x+e))^(
1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)+2*c)
^(1/2)*c^3-3/4/f*d*ln(d*tan(f*x+e)+c-(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1
/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c+2/f*d/(2*(c
^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)-(2*(c^2+d^2)^(1/2
)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*(c^2+d^2)^(1/2)*c-3/f*d/(2*(c^
2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)-(2*(c^2+d^2)^(1/2)
+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*c^2+1/f*d^3/(2*(c^2+d^2)^(1/2)-
2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)-(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/
(2*(c^2+d^2)^(1/2)-2*c)^(1/2))-1/4/f/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(
1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)+2*c)
^(1/2)*c^3+1/4/f/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1
/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(c^2+d^2)^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/
2)*c^2-1/4/f*d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+
2*c)^(1/2)+(c^2+d^2)^(1/2))*(c^2+d^2)^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+3
/4/f*d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/
2)+(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c+2/f*d/(2*(c^2+d^2)^(1/2
)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2)
)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*(c^2+d^2)^(1/2)*c-3/f*d/(2*(c^2+d^2)^(1/2)
-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))
/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*c^2+1/f*d^3/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*a
rctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)
^(1/2)-2*c)^(1/2)))

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 10840 vs. 2(189) = 378.

Time = 1.98 (sec) , antiderivative size = 10840, normalized size of antiderivative = 47.34

$$\int (c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\text{Too large to display}$$

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorit
hm="fricas")
```

```
[Out] Too large to include
```

Sympy [F]

$$\int (c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\int (c+d\tan(e+fx))^{\frac{5}{2}}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$

```
[In] integrate((c+d*tan(f*x+e))**(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2),x)
```

```
[Out] Integral((c + d*tan(e + f*x))**(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)**
2), x)
```

Maxima [F]

$$\int (c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\int \left(C\tan(fx+e)^2+B\tan(fx+e)+A\right)(d\tan(fx+e)+c)^{\frac{5}{2}}\,dx$$

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorit
hm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(d*tan(f*x + e) + c)^(5/2
), x)
```

Giac [F(-1)]

Timed out.

$$\int (c + d\tan(e + fx))^{5/2} \left(A + B\tan(e + fx) + C\tan^2(e + fx)\right) dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorit
hm="giac")
```

```
[Out] Timed out
```

Mupad [B] (verification not implemented)

Time = 114.33 (sec) , antiderivative size = 5863, normalized size of antiderivative = 25.60

$$\int (c + d\tan(e + fx))^{5/2} \left(A + B\tan(e + fx) + C\tan^2(e + fx)\right) dx = \text{Too large to display}$$

```
[In] int((c + d*tan(e + f*x))^(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2),x)
```

```
[Out] ((2*C*c^2)/(3*d*f) - (2*C*(d^3*f + c^2*d*f))/(3*d^2*f^2))*(c + d*tan(e + f*
x))^(3/2) - log(((((-B^4*d^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) + B^2*
c^5*f^2 - 10*B^2*c^3*d^2*f^2 + 5*B^2*c*d^4*f^2)/f^4)^(1/2)*(((((-B^4*d^2*f^
4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) + B^2*c^5*f^2 - 10*B^2*c^3*d^2*f^2 +
5*B^2*c*d^4*f^2)/f^4)^(1/2)*(32*B*c^4*d^2 - 32*B*d^6 + 32*c*d^2*f*(((-B^4*d
^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) + B^2*c^5*f^2 - 10*B^2*c^3*d^2*f
^2 + 5*B^2*c*d^4*f^2)/f^4)^(1/2)*(c + d*tan(e + f*x))^(1/2)))/(2*f) - (16*B
^2*d^2*(c + d*tan(e + f*x))^(1/2)*(c^6 - d^6 + 15*c^2*d^4 - 15*c^4*d^2))/f^
2))/2 - (8*B^3*c*d^2*(c^2 - 3*d^2)*(c^2 + d^2)^3)/f^3)*(((20*B^4*c^2*d^8*f^
4 - B^4*d^10*f^4 - 110*B^4*c^4*d^6*f^4 + 100*B^4*c^6*d^4*f^4 - 25*B^4*c^8*d
^2*f^4)^(1/2) + B^2*c^5*f^2 - 10*B^2*c^3*d^2*f^2 + 5*B^2*c*d^4*f^2)/(4*f^4)
)^(1/2) + log(- ((((-B^4*d^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) + B^2*
c^5*f^2 - 10*B^2*c^3*d^2*f^2 + 5*B^2*c*d^4*f^2)/f^4)^(1/2)*(((((-B^4*d^2*f^
4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) + B^2*c^5*f^2 - 10*B^2*c^3*d^2*f^2 +
5*B^2*c*d^4*f^2)/f^4)^(1/2)*(32*B*d^6 - 32*B*c^4*d^2 + 32*c*d^2*f*(((-B^4*d
^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) + B^2*c^5*f^2 - 10*B^2*c^3*d^2*f
^2 + 5*B^2*c*d^4*f^2)/f^4)^(1/2)*(c + d*tan(e + f*x))^(1/2)))/(2*f) - (16*B
^2*d^2*(c + d*tan(e + f*x))^(1/2)*(c^6 - d^6 + 15*c^2*d^4 - 15*c^4*d^2))/f^
2))/2 - (8*B^3*c*d^2*(c^2 - 3*d^2)*(c^2 + d^2)^3)/f^3)*((20*B^4*c^2*d^8*f^4
 - B^4*d^10*f^4 - 110*B^4*c^4*d^6*f^4 + 100*B^4*c^6*d^4*f^4 - 25*B^4*c^8*d^
2*f^4)^(1/2)/(4*f^4) + (B^2*c^5)/(4*f^2) - (5*B^2*c^3*d^2)/(2*f^2) + (5*B^2
*c*d^4)/(4*f^2))^(1/2) - log(((-((-B^4*d^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2
)^(1/2) - B^2*c^5*f^2 + 10*B^2*c^3*d^2*f^2 - 5*B^2*c*d^4*f^2)/f^4)^(1/2)*((
(-((-B^4*d^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) - B^2*c^5*f^2 + 10*B^2
*c^3*d^2*f^2 - 5*B^2*c*d^4*f^2)/f^4)^(1/2)*(32*B*c^4*d^2 - 32*B*d^6 + 32*c*
```

d^2*f*(-((-B^4*d^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) - B^2*c^5*f^2 +
10*B^2*c^3*d^2*f^2 - 5*B^2*c*d^4*f^2)/f^4)^(1/2)*(c + d*tan(e + f*x))^(1/2)
))/(2*f) - (16*B^2*d^2*(c + d*tan(e + f*x))^(1/2)*(c^6 - d^6 + 15*c^2*d^4 -
15*c^4*d^2))/f^2))/2 - (8*B^3*c*d^2*(c^2 - 3*d^2)*(c^2 + d^2)^3)/f^3)*(-((
20*B^4*c^2*d^8*f^4 - B^4*d^10*f^4 - 110*B^4*c^4*d^6*f^4 + 100*B^4*c^6*d^4*f
^4 - 25*B^4*c^8*d^2*f^4)^(1/2) - B^2*c^5*f^2 + 10*B^2*c^3*d^2*f^2 - 5*B^2*c
*d^4*f^2)/(4*f^4))^(1/2) + log(- ((-((-B^4*d^2*f^4*(5*c^4 + d^4 - 10*c^2*d^
2)^2)^(1/2) - B^2*c^5*f^2 + 10*B^2*c^3*d^2*f^2 - 5*B^2*c*d^4*f^2)/f^4)^(1/2
)*(((-((-B^4*d^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) - B^2*c^5*f^2 + 10
*B^2*c^3*d^2*f^2 - 5*B^2*c*d^4*f^2)/f^4)^(1/2)*(32*B*d^6 - 32*B*c^4*d^2 + 3
2*c*d^2*f*(-((-B^4*d^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) - B^2*c^5*f^
2 + 10*B^2*c^3*d^2*f^2 - 5*B^2*c*d^4*f^2)/f^4)^(1/2)*(c + d*tan(e + f*x))^(
1/2)))/(2*f) - (16*B^2*d^2*(c + d*tan(e + f*x))^(1/2)*(c^6 - d^6 + 15*c^2*d
^4 - 15*c^4*d^2))/f^2))/2 - (8*B^3*c*d^2*(c^2 - 3*d^2)*(c^2 + d^2)^3)/f^3)*
((B^2*c^5)/(4*f^2) - (20*B^4*c^2*d^8*f^4 - B^4*d^10*f^4 - 110*B^4*c^4*d^6*f
^4 + 100*B^4*c^6*d^4*f^4 - 25*B^4*c^8*d^2*f^4)^(1/2)/(4*f^4) - (5*B^2*c^3*d
^2)/(2*f^2) + (5*B^2*c*d^4)/(4*f^2))^(1/2) + ((4*B*c^2)/f - (2*B*(c^2*f + d
^2*f))/f^2)*(c + d*tan(e + f*x))^(1/2) - log(((((-((-A^4*d^2*f^4*(5*c^4 + d
^4 - 10*c^2*d^2)^2)^(1/2) + A^2*c^5*f^2 - 10*A^2*c^3*d^2*f^2 + 5*A^2*c*d^4*
f^2)/f^4)^(1/2)*(64*A*c^3*d^3 + 64*A*c*d^5 + 32*c*d^2*f*(-((-A^4*d^2*f^4*(5
*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) + A^2*c^5*f^2 - 10*A^2*c^3*d^2*f^2 + 5*A^
2*c*d^4*f^2)/f^4)^(1/2)*(c + d*tan(e + f*x))^(1/2)))/(2*f) + (16*A^2*d^2*(c
+ d*tan(e + f*x))^(1/2)*(c^6 - d^6 + 15*c^2*d^4 - 15*c^4*d^2))/f^2)*(-((-A
^4*d^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) + A^2*c^5*f^2 - 10*A^2*c^3*d
^2*f^2 + 5*A^2*c*d^4*f^2)/f^4)^(1/2))/2 - (8*A^3*d^3*(3*c^2 - d^2)*(c^2 + d
^2)^3)/f^3)*(-((20*A^4*c^2*d^8*f^4 - A^4*d^10*f^4 - 110*A^4*c^4*d^6*f^4 + 1
00*A^4*c^6*d^4*f^4 - 25*A^4*c^8*d^2*f^4)^(1/2) + A^2*c^5*f^2 - 10*A^2*c^3*d
^2*f^2 + 5*A^2*c*d^4*f^2)/(4*f^4))^(1/2) - log((((((((-A^4*d^2*f^4*(5*c^4 +
d^4 - 10*c^2*d^2)^2)^(1/2) - A^2*c^5*f^2 + 10*A^2*c^3*d^2*f^2 - 5*A^2*c*d^4
f^2)/f^4)^(1/2)(64*A*c^3*d^3 + 64*A*c*d^5 + 32*c*d^2*f*(((-A^4*d^2*f^4*(5
*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) - A^2*c^5*f^2 + 10*A^2*c^3*d^2*f^2 - 5*A^
2*c*d^4*f^2)/f^4)^(1/2)*(c + d*tan(e + f*x))^(1/2)))/(2*f) + (16*A^2*d^2*(c
+ d*tan(e + f*x))^(1/2)*(c^6 - d^6 + 15*c^2*d^4 - 15*c^4*d^2))/f^2)*(((-A^
4*d^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) - A^2*c^5*f^2 + 10*A^2*c^3*d^
2*f^2 - 5*A^2*c*d^4*f^2)/f^4)^(1/2))/2 - (8*A^3*d^3*(3*c^2 - d^2)*(c^2 + d^
2)^3)/f^3)*(((20*A^4*c^2*d^8*f^4 - A^4*d^10*f^4 - 110*A^4*c^4*d^6*f^4 + 100
*A^4*c^6*d^4*f^4 - 25*A^4*c^8*d^2*f^4)^(1/2) - A^2*c^5*f^2 + 10*A^2*c^3*d^2
*f^2 - 5*A^2*c*d^4*f^2)/(4*f^4))^(1/2) + log((((((((-A^4*d^2*f^4*(5*c^4 + d^
4 - 10*c^2*d^2)^2)^(1/2) - A^2*c^5*f^2 + 10*A^2*c^3*d^2*f^2 - 5*A^2*c*d^4*f
^2)/f^4)^(1/2)*(64*A*c^3*d^3 + 64*A*c*d^5 - 32*c*d^2*f*(((-A^4*d^2*f^4*(5*c
^4 + d^4 - 10*c^2*d^2)^2)^(1/2) - A^2*c^5*f^2 + 10*A^2*c^3*d^2*f^2 - 5*A^2*
c*d^4*f^2)/f^4)^(1/2)*(c + d*tan(e + f*x))^(1/2)))/(2*f) - (16*A^2*d^2*(c +
d*tan(e + f*x))^(1/2)*(c^6 - d^6 + 15*c^2*d^4 - 15*c^4*d^2))/f^2)*(((-A^4*
d^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) - A^2*c^5*f^2 + 10*A^2*c^3*d^2*
f^2 - 5*A^2*c*d^4*f^2)/f^4)^(1/2))/2 - (8*A^3*d^3*(3*c^2 - d^2)*(c^2 + d^2)

^3)/f^3)*((20*A^4*c^2*d^8*f^4 - A^4*d^10*f^4 - 110*A^4*c^4*d^6*f^4 + 100*A^
4*c^6*d^4*f^4 - 25*A^4*c^8*d^2*f^4)^(1/2)/(4*f^4) - (A^2*c^5)/(4*f^2) + (5*
A^2*c^3*d^2)/(2*f^2) - (5*A^2*c*d^4)/(4*f^2))^(1/2) + log(((((-((-A^4*d^2*f
^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) + A^2*c^5*f^2 - 10*A^2*c^3*d^2*f^2 +
5*A^2*c*d^4*f^2)/f^4)^(1/2)*(64*A*c^3*d^3 + 64*A*c*d^5 - 32*c*d^2*f*(-((-A
^4*d^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) + A^2*c^5*f^2 - 10*A^2*c^3*d
^2*f^2 + 5*A^2*c*d^4*f^2)/f^4)^(1/2)*(c + d*tan(e + f*x))^(1/2)))/(2*f) - (
16*A^2*d^2*(c + d*tan(e + f*x))^(1/2)*(c^6 - d^6 + 15*c^2*d^4 - 15*c^4*d^2)
)/f^2)*(-((-A^4*d^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) + A^2*c^5*f^2 -
10*A^2*c^3*d^2*f^2 + 5*A^2*c*d^4*f^2)/f^4)^(1/2))/2 - (8*A^3*d^3*(3*c^2 -
d^2)*(c^2 + d^2)^3)/f^3)*((5*A^2*c^3*d^2)/(2*f^2) - (A^2*c^5)/(4*f^2) - (20
*A^4*c^2*d^8*f^4 - A^4*d^10*f^4 - 110*A^4*c^4*d^6*f^4 + 100*A^4*c^6*d^4*f^4
- 25*A^4*c^8*d^2*f^4)^(1/2)/(4*f^4) - (5*A^2*c*d^4)/(4*f^2))^(1/2) - log((
8*C^3*d^3*(3*c^2 - d^2)*(c^2 + d^2)^3)/f^3 - ((((-((-C^4*d^2*f^4*(5*c^4 + d
^4 - 10*c^2*d^2)^2)^(1/2) + C^2*c^5*f^2 - 10*C^2*c^3*d^2*f^2 + 5*C^2*c*d^4*
f^2)/f^4)^(1/2)*(64*C*c^3*d^3 + 64*C*c*d^5 - 32*c*d^2*f*(-((-C^4*d^2*f^4*(5
*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) + C^2*c^5*f^2 - 10*C^2*c^3*d^2*f^2 + 5*C^
2*c*d^4*f^2)/f^4)^(1/2)*(c + d*tan(e + f*x))^(1/2)))/(2*f) - (16*C^2*d^2*(c
+ d*tan(e + f*x))^(1/2)*(c^6 - d^6 + 15*c^2*d^4 - 15*c^4*d^2))/f^2)*(-((-C
^4*d^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) + C^2*c^5*f^2 - 10*C^2*c^3*d
^2*f^2 + 5*C^2*c*d^4*f^2)/f^4)^(1/2))/2)*(-((20*C^4*c^2*d^8*f^4 - C^4*d^10*
f^4 - 110*C^4*c^4*d^6*f^4 + 100*C^4*c^6*d^4*f^4 - 25*C^4*c^8*d^2*f^4)^(1/2)
+ C^2*c^5*f^2 - 10*C^2*c^3*d^2*f^2 + 5*C^2*c*d^4*f^2)/(4*f^4))^(1/2) - log
((8*C^3*d^3*(3*c^2 - d^2)*(c^2 + d^2)^3)/f^3 - ((((((-C^4*d^2*f^4*(5*c^4 +
d^4 - 10*c^2*d^2)^2)^(1/2) - C^2*c^5*f^2 + 10*C^2*c^3*d^2*f^2 - 5*C^2*c*d^4
f^2)/f^4)^(1/2)(64*C*c^3*d^3 + 64*C*c*d^5 - 32*c*d^2*f*(((-C^4*d^2*f^4*(5
*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) - C^2*c^5*f^2 + 10*C^2*c^3*d^2*f^2 - 5*C^
2*c*d^4*f^2)/f^4)^(1/2)*(c + d*tan(e + f*x))^(1/2)))/(2*f) - (16*C^2*d^2*(c
+ d*tan(e + f*x))^(1/2)*(c^6 - d^6 + 15*c^2*d^4 - 15*c^4*d^2))/f^2)*(((-C^
4*d^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) - C^2*c^5*f^2 + 10*C^2*c^3*d^
2*f^2 - 5*C^2*c*d^4*f^2)/f^4)^(1/2))/2)*(((20*C^4*c^2*d^8*f^4 - C^4*d^10*f^
4 - 110*C^4*c^4*d^6*f^4 + 100*C^4*c^6*d^4*f^4 - 25*C^4*c^8*d^2*f^4)^(1/2) -
C^2*c^5*f^2 + 10*C^2*c^3*d^2*f^2 - 5*C^2*c*d^4*f^2)/(4*f^4))^(1/2) + log((
8*C^3*d^3*(3*c^2 - d^2)*(c^2 + d^2)^3)/f^3 - ((((((-C^4*d^2*f^4*(5*c^4 + d^
4 - 10*c^2*d^2)^2)^(1/2) - C^2*c^5*f^2 + 10*C^2*c^3*d^2*f^2 - 5*C^2*c*d^4*f
^2)/f^4)^(1/2)*(64*C*c^3*d^3 + 64*C*c*d^5 + 32*c*d^2*f*(((-C^4*d^2*f^4*(5*c
^4 + d^4 - 10*c^2*d^2)^2)^(1/2) - C^2*c^5*f^2 + 10*C^2*c^3*d^2*f^2 - 5*C^2*
c*d^4*f^2)/f^4)^(1/2)*(c + d*tan(e + f*x))^(1/2)))/(2*f) + (16*C^2*d^2*(c +
d*tan(e + f*x))^(1/2)*(c^6 - d^6 + 15*c^2*d^4 - 15*c^4*d^2))/f^2)*(((-C^4*
d^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) - C^2*c^5*f^2 + 10*C^2*c^3*d^2*
f^2 - 5*C^2*c*d^4*f^2)/f^4)^(1/2))/2)*((20*C^4*c^2*d^8*f^4 - C^4*d^10*f^4 -
110*C^4*c^4*d^6*f^4 + 100*C^4*c^6*d^4*f^4 - 25*C^4*c^8*d^2*f^4)^(1/2)/(4*f
^4) - (C^2*c^5)/(4*f^2) + (5*C^2*c^3*d^2)/(2*f^2) - (5*C^2*c*d^4)/(4*f^2))^
(1/2) + log((8*C^3*d^3*(3*c^2 - d^2)*(c^2 + d^2)^3)/f^3 - ((((-((-C^4*d^2*f
^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) + C^2*c^5*f^2 - 10*C^2*c^3*d^2*f^2 +

```
 5*C^2*c*d^4*f^2)/f^4)^(1/2)*(64*C*c^3*d^3 + 64*C*c*d^5 + 32*c*d^2*f*(-((-C
^4*d^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) + C^2*c^5*f^2 - 10*C^2*c^3*d
^2*f^2 + 5*C^2*c*d^4*f^2)/f^4)^(1/2)*(c + d*tan(e + f*x))^(1/2)))/(2*f) + (
16*C^2*d^2*(c + d*tan(e + f*x))^(1/2)*(c^6 - d^6 + 15*c^2*d^4 - 15*c^4*d^2)
)/f^2)*(-((-C^4*d^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) + C^2*c^5*f^2 -
 10*C^2*c^3*d^2*f^2 + 5*C^2*c*d^4*f^2)/f^4)^(1/2))/2)*((5*C^2*c^3*d^2)/(2*f
^2) - (C^2*c^5)/(4*f^2) - (20*C^4*c^2*d^8*f^4 - C^4*d^10*f^4 - 110*C^4*c^4*
d^6*f^4 + 100*C^4*c^6*d^4*f^4 - 25*C^4*c^8*d^2*f^4)^(1/2)/(4*f^4) - (5*C^2*
c*d^4)/(4*f^2))^(1/2) + 2*c*((2*C*c^2)/(d*f) - (2*C*(d^3*f + c^2*d*f))/(d^2
*f^2))*(c + d*tan(e + f*x))^(1/2) + (2*B*(c + d*tan(e + f*x))^(5/2))/(5*f)
+ (2*A*d*(c + d*tan(e + f*x))^(3/2))/(3*f) + (2*B*c*(c + d*tan(e + f*x))^(3
/2))/(3*f) + (2*C*(c + d*tan(e + f*x))^(7/2))/(7*d*f) + (4*A*c*d*(c + d*tan
(e + f*x))^(1/2))/f
```

3.107 $\int \frac{(c+d\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx$

Optimal result 1418
Rubi [A] (verified) 1419
Mathematica [A] (verified) 1423
Maple [B] (verified) 1423
Fricas [F(-1)] 1423
Sympy [F(-1)] 1424
Maxima [F(-2)] 1424
Giac [F(-1)] 1424
Mupad [F(-1)] 1425

Optimal result

Integrand size = 47, antiderivative size = 336

$$\begin{aligned}
&\int \frac{(c+d\tan(e+fx))^{5/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx = \\
&-\frac{(iA+B-iC)(c-id)^{5/2}\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(a-ib)f} \\
&+\frac{(iA-B-iC)(c+id)^{5/2}\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(a+ib)f} \\
&-\frac{2(Ab^2-a(bB-aC))\,(bc-ad)^{5/2}\operatorname{arctanh}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{b^{7/2}\,(a^2+b^2)\,f} \\
&+\frac{2(b^2d(Bc+(A-C)d)+(bc-ad)(bcC+bBd-aCd))\,\sqrt{c+d\tan(e+fx)}}{b^3f} \\
&+\frac{2(bcC+bBd-aCd)(c+d\tan(e+fx))^{3/2}}{3b^2f}+\frac{2C(c+d\tan(e+fx))^{5/2}}{5bf}
\end{aligned}$$

```
[Out] -(I*A+B-I*C)*(c-I*d)^(5/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(a
-I*b)/f+(I*A-B-I*C)*(c+I*d)^(5/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1
/2))/(a+I*b)/f-2*(A*b^2-a*(B*b-C*a))*(-a*d+b*c)^(5/2)*arctanh(b^(1/2)*(c+d*
tan(f*x+e))^(1/2)/(-a*d+b*c)^(1/2))/b^(7/2)/(a^2+b^2)/f+2*(b^2*d*(B*c+(A-C)
*d)+(-a*d+b*c)*(B*b*d-C*a*d+C*b*c))*(c+d*tan(f*x+e))^(1/2)/b^3/f+2/3*(B*b*d
-C*a*d+C*b*c)*(c+d*tan(f*x+e))^(3/2)/b^2/f+2/5*C*(c+d*tan(f*x+e))^(5/2)/b/f
```

Rubi [A] (verified)

Time = 3.13 (sec) , antiderivative size = 336, normalized size of antiderivative = 1.00, number of steps used = 14, number of rules used = 7, $\frac{\text{number of rules}}{\text{integrand size}} = 0.149$, Rules used = {3728, 3734, 3620, 3618, 65, 214, 3715}

$$\int \frac{(c+d\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx = -\frac{2(bc-ad)^{5/2}(Ab^2-a(bB-aC))\operatorname{arctanh}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{b^{7/2}f(a^2+b^2)} - \frac{(c-id)^{5/2}(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f(a-ib)} + \frac{(c+id)^{5/2}(iA-B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f(a+ib)} + \frac{2\sqrt{c+d\tan(e+fx)}((bc-ad)(-aCd+bBd+bcC)+b^2d(d(A-C)+Bc))}{b^3f} + \frac{2(-aCd+bBd+bcC)(c+d\tan(e+fx))^{3/2}}{3b^2f} + \frac{2C(c+d\tan(e+fx))^{5/2}}{5bf}$$

```
[In] Int[((c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a
 + b*Tan[e + f*x]),x]
```

```
[Out] -(((I*A + B - I*C)*(c - I*d)^(5/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c
- I*d]])/((a - I*b)*f)) + ((I*A - B - I*C)*(c + I*d)^(5/2)*ArcTanh[Sqrt[c +
 d*Tan[e + f*x]]/Sqrt[c + I*d]])/((a + I*b)*f) - (2*(A*b^2 - a*(b*B - a*C))
*(b*c - a*d)^(5/2)*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*
d]])/(b^(7/2)*(a^2 + b^2)*f) + (2*(b^2*d*(B*c + (A - C)*d) + (b*c - a*d)*(b
*c*C + b*B*d - a*C*d))*Sqrt[c + d*Tan[e + f*x]])/(b^3*f) + (2*(b*c*C + b*B*
d - a*C*d)*(c + d*Tan[e + f*x])^(3/2))/(3*b^2*f) + (2*C*(c + d*Tan[e + f*x]
)^(5/2))/(5*b*f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3715

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_) + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :>
 Dist[A/f, Subst[Int[(a + b*x)^m*(c + d*x)^n, x], x, Tan[e + f*x]], x] /; F
reeQ[{a, b, c, d, e, f, A, C, m, n}, x] && EqQ[A, C]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3734

```
Int[(((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2))/((a_.) + (b_.)*tan[(e_.)
+ (f_.)*(x_)]), x_Symbol] :> Dist[1/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n
*Simp[b*B + a*(A - C) + (a*B - b*(A - C))*Tan[e + f*x], x], x], x] + Dist[(
A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n*((1 + Tan[e
+ f*x]^2)/(a + b*Tan[e + f*x])), x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C
, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&
 !GtQ[n, 0] &&  !LeQ[n, -1]
```

Rubi steps

$$\text{integral} = \frac{2C(c+d\tan(e+fx))^{5/2}}{5bf} + \frac{2\int \frac{(c+d\tan(e+fx))^{3/2}\left(\frac{5}{2}(Abc-aCd)+\frac{5}{2}b(Bc+(A-C)d)\tan(e+fx)+\frac{5}{2}(bcC+bBd-aCd)\tan^2(e+fx)\right)}{a+b\tan(e+fx)}\,dx}{5b}$$

$$= \frac{2(bcC+bBd-aCd)(c+d\tan(e+fx))^{3/2}}{3b^2f} + \frac{2C(c+d\tan(e+fx))^{5/2}}{5bf} + \frac{4\int \frac{\sqrt{c+d\tan(e+fx)}\left(\frac{15}{4}\left(Ab^2c^2+ad(aCd-b(2cC+Bd))\right)+\frac{15}{4}b^2\left(2c(A-C)d+B\left(c^2-d^2\right)\right)\tan(e+fx)+\frac{15}{4}\left(b^2d(Bc+(A-C)d)+(bc\right.}{a+b\tan(e+fx)}}{15b^2}$$

$$= \frac{2(b^2d(Bc+(A-C)d)+(bc-ad)(bcC+bBd-aCd))\sqrt{c+d\tan(e+fx)}}{b^3f} + \frac{2(bcC+bBd-aCd)(c+d\tan(e+fx))^{3/2}}{3b^2f} + \frac{2C(c+d\tan(e+fx))^{5/2}}{5bf} + \frac{8\int \frac{\frac{15}{8}\left(Ab^2\left(bc^3-ad^3\right)-ad\left(a^2Cd^2-abd(3cC+Bd)+b^2\left(3c^2C+3Bcd-Cd^2\right)\right)\right)+\frac{15}{8}b^3\left((A-C)d\left(3c^2-d^2\right)+B\left(c^3-3cd^2\right)\right)\tan(e+fx)}{(a+b\tan(e+fx))\sqrt{c+d\tan}}}{15b^3}$$

$$= \frac{2(b^2d(Bc+(A-C)d)+(bc-ad)(bcC+bBd-aCd))\sqrt{c+d\tan(e+fx)}}{b^3f} + \frac{2(bcC+bBd-aCd)(c+d\tan(e+fx))^{3/2}}{3b^2f} + \frac{2C(c+d\tan(e+fx))^{5/2}}{5bf} + \frac{8\int \frac{-\frac{15}{8}b^3\left(a\left(c^3C+3Bc^2d-3cCd^2-Bd^3-A\left(c^3-3cd^2\right)\right)-b\left((A-C)d\left(3c^2-d^2\right)+B\left(c^3-3cd^2\right)\right)\right)+\frac{15}{8}b^3\left(aAd\left(3c^2-d^2\right)-Ab\left(c^3-3cd\right.\right.}{\sqrt{c+d\tan(e+fx)}}}{15b^3(a^2+b^2)} + \frac{\left((Ab^2-a(bB-aC))(bc-ad)^3\right)\int \frac{1+\tan^2(e+fx)}{(a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}}\,dx}{b^3(a^2+b^2)}$$

$$= \frac{2(b^2d(Bc+(A-C)d)+(bc-ad)(bcC+bBd-aCd))\sqrt{c+d\tan(e+fx)}}{b^3f} + \frac{2(bcC+bBd-aCd)(c+d\tan(e+fx))^{3/2}}{3b^2f} + \frac{2C(c+d\tan(e+fx))^{5/2}}{5bf} + \frac{\left((A-iB-C)(c-id)^3\right)\int \frac{1+i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(a-ib)} + \frac{\left((A+iB-C)(c+id)^3\right)\int \frac{1-i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(a+ib)} + \frac{\left((Ab^2-a(bB-aC))(bc-ad)^3\right)\text{Subst}\left(\int \frac{1}{(a+bx)\sqrt{c+dx}}\,dx,x,\tan(e+fx)\right)}{b^3(a^2+b^2)f}$$

$$= \frac{2(b^2d(Bc+(A-C)d)+(bc-ad)(bcC+bBd-aCd))\sqrt{c+d\tan(e+fx)}}{b^3f}$$
$$+ \frac{2(bcC+bBd-aCd)(c+d\tan(e+fx))^{3/2}}{3b^2f} + \frac{2C(c+d\tan(e+fx))^{5/2}}{5bf}$$
$$+ \frac{((iA+B-iC)(c-id)^3)\,\text{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c-idx}}\,dx, x, i\tan(e+fx)\right)}{2(a-ib)f}$$
$$- \frac{(i(A+iB-C)(c+id)^3)\,\text{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c+idx}}\,dx, x, -i\tan(e+fx)\right)}{2(a+ib)f}$$
$$+ \frac{(2(Ab^2-a(bB-aC))\,(bc-ad)^3)\,\text{Subst}\left(\int \frac{1}{a-\frac{bc}{d}+\frac{bx^2}{d}}\,dx, x, \sqrt{c+d\tan(e+fx)}\right)}{b^3\,(a^2+b^2)\,df}$$

$$= -\frac{2(Ab^2-a(bB-aC))\,(bc-ad)^{5/2}\text{arctanh}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{b^{7/2}\,(a^2+b^2)\,f}$$
$$+ \frac{2(b^2d(Bc+(A-C)d)+(bc-ad)(bcC+bBd-aCd))\sqrt{c+d\tan(e+fx)}}{b^3f}$$
$$+ \frac{2(bcC+bBd-aCd)(c+d\tan(e+fx))^{3/2}}{3b^2f} + \frac{2C(c+d\tan(e+fx))^{5/2}}{5bf}$$
$$- \frac{((A-iB-C)(c-id)^3)\,\text{Subst}\left(\int \frac{1}{-1-\frac{ic}{d}+\frac{ix^2}{d}}\,dx, x, \sqrt{c+d\tan(e+fx)}\right)}{(a-ib)df}$$
$$- \frac{((A+iB-C)(c+id)^3)\,\text{Subst}\left(\int \frac{1}{-1+\frac{ic}{d}-\frac{ix^2}{d}}\,dx, x, \sqrt{c+d\tan(e+fx)}\right)}{(a+ib)df}$$

$$= -\frac{(iA+B-iC)(c-id)^{5/2}\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(a-ib)f}$$
$$- \frac{(A+iB-C)(c+id)^{5/2}\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(ia-b)f}$$
$$- \frac{2(Ab^2-a(bB-aC))\,(bc-ad)^{5/2}\text{arctanh}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{b^{7/2}\,(a^2+b^2)\,f}$$
$$+ \frac{2(b^2d(Bc+(A-C)d)+(bc-ad)(bcC+bBd-aCd))\sqrt{c+d\tan(e+fx)}}{b^3f}$$
$$+ \frac{2(bcC+bBd-aCd)(c+d\tan(e+fx))^{3/2}}{3b^2f} + \frac{2C(c+d\tan(e+fx))^{5/2}}{5bf}$$

Mathematica [A] (verified)

Time = 5.76 (sec) , antiderivative size = 322, normalized size of antiderivative = 0.96

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{a+b\tan(e+fx)}\,dx = \frac{15\Big(b^{7/2}(-ia+b)(A-iB-C)(c-id)^{5/2}\text{arctanh}\Big(}{}$$

```
[In] Integrate[((c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/(a + b*Tan[e + f*x]),x]
```

```
[Out] ((15*(b^(7/2)*((-I)*a + b)*(A - I*B - C)*(c - I*d)^(5/2)*ArcTanh[Sqrt[c + d
*Tan[e + f*x]]/Sqrt[c - I*d]] + b^(7/2)*(I*a + b)*(A + I*B - C)*(c + I*d)^(
5/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]] - 2*(A*b^2 + a*(-(b*B)
 + a*C))*(b*c - a*d)^(5/2)*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[
b*c - a*d]]))/(b^(5/2)*(a^2 + b^2)) + (30*(b^2*d*(B*c + (A - C)*d) + (b*c -
 a*d)*(b*c*C + b*B*d - a*C*d))*Sqrt[c + d*Tan[e + f*x]])/b^2 + (10*(b*c*C +
 b*B*d - a*C*d)*(c + d*Tan[e + f*x])^(3/2))/b + 6*C*(c + d*Tan[e + f*x])^(5
/2))/(15*b*f)
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 8697 vs. 2(294) = 588.

Time = 0.18 (sec) , antiderivative size = 8698, normalized size of antiderivative = 25.89

method	result	size
derivativedivides	Expression too large to display	8698
default	Expression too large to display	8698

```
[In] int((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
,x,method=_RETURNVERBOSE)
```

```
[Out] result too large to display
```

Fricas [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{a+b\tan(e+fx)}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e)),x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))**(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e)),x)
```

```
[Out] Timed out
```

Maxima [F(-2)]

Exception generated.

$$\int \frac{(c+d\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx = \text{Exception raised: ValueError}$$

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e)),x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 detail
```

Giac [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e)),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{5/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{a+b\tan(e+fx)}\,dx = \text{Hanged}$$

```
[In] int(((c + d*tan(e + f*x))^(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x)),x)
[Out] \text{Hanged}
```

3.108 $\int \frac{(c+d\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx$

Optimal result 1426
Rubi [A] (verified) 1427
Mathematica [B] (verified) 1432
Maple [B] (verified) 1432
Fricas [F(-1)] 1433
Sympy [F(-1)] 1433
Maxima [F(-2)] 1433
Giac [F(-1)] 1434
Mupad [F(-1)] 1434

Optimal result

Integrand size = 47, antiderivative size = 473

$$\int \frac{(c+d\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx =$$
$$-\frac{(iA+B-iC)(c-id)^{5/2}\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(a-ib)^2 f}$$
$$-\frac{(B-i(A-C))(c+id)^{5/2}\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(a+ib)^2 f}$$
$$+\frac{(bc-ad)^{3/2}\left(3a^3bBd-5a^4Cd-b^4(2Bc+5Ad)-ab^3(4Ac-4cC-7Bd)+a^2b^2(2Bc-(A+9C)d)\right)\text{arcta}}{b^{7/2}\left(a^2+b^2\right)^2 f}$$
$$-\frac{d\left(5a^3Cd-Ab^2(bc-ad)-2b^3(2cC+Bd)-a^2b(5cC+3Bd)+ab^2(Bc+4Cd)\right)\sqrt{c+d\tan(e+fx)}}{b^3\left(a^2+b^2\right)f}$$
$$+\frac{\left(3Ab^2-3abB+5a^2C+2b^2C\right)d(c+d\tan(e+fx))^{3/2}}{3b^2\left(a^2+b^2\right)f}$$
$$-\frac{\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^{5/2}}{b\left(a^2+b^2\right)f(a+b\tan(e+fx))}$$

```
[Out] -(I*A+B-I*C)*(c-I*d)^(5/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(a
-I*b)^2/f-(B-I*(A-C))*(c+I*d)^(5/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^
(1/2))/(a+I*b)^2/f+(-a*d+b*c)^(3/2)*(3*a^3*b*B*d-5*a^4*C*d-b^4*(5*A*d+2*B*c
)-a*b^3*(4*A*c-7*B*d-4*C*c)+a^2*b^2*(2*B*c-(A+9*C)*d))*arctanh(b^(1/2)*(c+d
*tan(f*x+e))^(1/2)/(-a*d+b*c)^(1/2))/b^(7/2)/(a^2+b^2)^2/f-d*(5*a^3*C*d-A*b
^2*(-a*d+b*c)-2*b^3*(B*d+2*C*c)-a^2*b*(3*B*d+5*C*c)+a*b^2*(B*c+4*C*d))*(c+d
*tan(f*x+e))^(1/2)/b^3/(a^2+b^2)/f+1/3*(3*A*b^2-3*B*a*b+5*C*a^2+2*C*b^2)*d*
(c+d*tan(f*x+e))^(3/2)/b^2/(a^2+b^2)/f-(A*b^2-a*(B*b-C*a))*(c+d*tan(f*x+e))
^(5/2)/b/(a^2+b^2)/f/(a+b*tan(f*x+e))
```

Rubi [A] (verified)

Time = 4.36 (sec) , antiderivative size = 473, normalized size of antiderivative = 1.00, number of steps used = 14, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.170, Rules used = {3726, 3728, 3734, 3620, 3618, 65, 214, 3715}

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^2}\,dx =$$
$$-\frac{(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{5/2}}{bf\,(a^2+b^2)\,(a+b\tan(e+fx))}$$
$$+\frac{d(5a^2C-3abB+3Ab^2+2b^2C)\,(c+d\tan(e+fx))^{3/2}}{3b^2f\,(a^2+b^2)}$$
$$-\frac{d\sqrt{c+d\tan(e+fx)}(5a^3Cd-a^2b(3Bd+5cC)-Ab^2(bc-ad)+ab^2(Bc+4Cd)-2b^3(Bd+2cC))}{b^3f\,(a^2+b^2)}$$
$$+\frac{(bc-ad)^{3/2}\,(-5a^4Cd+3a^3bBd+a^2b^2(2Bc-d(A+9C))-ab^3(4Ac-7Bd-4cC)-b^4(5Ad+2Bc))\,\text{ar}}{b^{7/2}f\,(a^2+b^2)^2}$$
$$-\frac{(c-id)^{5/2}(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f(a-ib)^2}$$
$$-\frac{(c+id)^{5/2}(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f(a+ib)^2}$$

```
[In] Int[((c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a
 + b*Tan[e + f*x])^2,x]
```

```
[Out] -(((I*A + B - I*C)*(c - I*d)^(5/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c
- I*d]])/((a - I*b)^2*f)) - ((B - I*(A - C))*(c + I*d)^(5/2)*ArcTanh[Sqrt[c
 + d*Tan[e + f*x]]/Sqrt[c + I*d]])/((a + I*b)^2*f) + ((b*c - a*d)^(3/2)*(3*
a^3*b*B*d - 5*a^4*C*d - b^4*(2*B*c + 5*A*d) - a*b^3*(4*A*c - 4*c*C - 7*B*d)
 + a^2*b^2*(2*B*c - (A + 9*C)*d))*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]]
)/Sqrt[b*c - a*d]])/(b^(7/2)*(a^2 + b^2)^2*f) - (d*(5*a^3*C*d - A*b^2*(b*c
- a*d) - 2*b^3*(2*c*C + B*d) - a^2*b*(5*c*C + 3*B*d) + a*b^2*(B*c + 4*C*d))
*Sqrt[c + d*Tan[e + f*x]])/(b^3*(a^2 + b^2)*f) + ((3*A*b^2 - 3*a*b*B + 5*a^
2*C + 2*b^2*C)*d*(c + d*Tan[e + f*x])^(3/2))/(3*b^2*(a^2 + b^2)*f) - ((A*b^
2 - a*(b*B - a*C))*(c + d*Tan[e + f*x])^(5/2))/(b*(a^2 + b^2)*f*(a + b*Tan[
e + f*x]))
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
```

```
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3715

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_) + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :>
 Dist[A/f, Subst[Int[(a + b*x)^m*(c + d*x)^n, x], x, Tan[e + f*x]], x] /; F
reeQ[{a, b, c, d, e, f, A, C, m, n}, x] && EqQ[A, C]
```

Rule 3726

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*d^2 + c*(c*C - B*d))*(a + b*Tan[e +
 f*x])^m*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
```

```
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3734

```
Int[(((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2))/((a_.) + (b_.)*tan[(e_.)
+ (f_.)*(x_)]), x_Symbol] :> Dist[1/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n
*Simp[b*B + a*(A - C) + (a*B - b*(A - C))*Tan[e + f*x], x], x], x] + Dist[(
A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n*((1 + Tan[e
+ f*x]^2)/(a + b*Tan[e + f*x])), x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C
, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&
 !GtQ[n, 0] &&  !LeQ[n, -1]
```

Rubi steps

$$\text{integral} = -\frac{(Ab^2 - a(bB - aC))\,(c + d\tan(e+fx))^{5/2}}{b\,(a^2+b^2)\,f(a + b\tan(e+fx))} + \frac{\int \frac{(c+d\tan(e+fx))^{3/2}\left(\frac{1}{2}\left(2(bB-aC)\left(bc-\frac{5ad}{2}\right)+2Ab\left(ac+\frac{5bd}{2}\right)\right)-b((A-C)(bc-ad)-B(ac+bd))\tan(e+fx)+\frac{1}{2}\left(3Ab^2-3abB+5a^2C+2b^2C\right.}{a+b\tan(e+fx)}}{b\,(a^2+b^2)}$$

$$= \frac{(3Ab^2 - 3abB + 5a^2C + 2b^2C)\,d(c + d\tan(e+fx))^{3/2}}{3b^2\,(a^2+b^2)\,f} - \frac{(Ab^2 - a(bB - aC))\,(c + d\tan(e+fx))^{5/2}}{b\,(a^2+b^2)\,f(a + b\tan(e+fx))} + \frac{2\int \frac{\sqrt{c+d\tan(e+fx)}\left(-\frac{3}{4}\left(a\left(3Ab^2-3abB+5a^2C+2b^2C\right)d^2-bc((bB-aC)(2bc-5ad)+Ab(2ac+5bd))\right)+\frac{3}{2}b^2\left(2aAcd-2acCd-Ab(c\right.\right.}{}}{}$$

$$= -\frac{d(5a^3Cd - Ab^2(bc - ad) - 2b^3(2cC + Bd) - a^2b(5cC + 3Bd) + ab^2(Bc + 4Cd))\sqrt{c + d\tan(e}}{b^3\,(a^2+b^2)\,f} + \frac{(3Ab^2 - 3abB + 5a^2C + 2b^2C)\,d(c + d\tan(e+fx))^{3/2}}{3b^2\,(a^2+b^2)\,f} - \frac{(Ab^2 - a(bB - aC))\,(c + d\tan(e+fx))^{5/2}}{b\,(a^2+b^2)\,f(a + b\tan(e+fx))} + \frac{4\int \frac{\frac{3}{8}\left(5a^4Cd^3+b^4c^2(2Bc+5Ad)-a^3bd^2(10cC+3Bd)+a^2b^2d\left(5c^2C+4Bcd+(A+4C)d^2\right)+ab^3\left(2Ac^3-2c^3C-5Bc^2d-4Acd^2-6cCd\right.\right.}{}}{}$$

$$= -\frac{d(5a^3Cd - Ab^2(bc-ad) - 2b^3(2cC+Bd) - a^2b(5cC+3Bd) + ab^2(Bc+4Cd))\sqrt{c+d\tan(e+}}{b^3(a^2+b^2)f}$$

$$+ \frac{(3Ab^2 - 3abB + 5a^2C + 2b^2C)\,d(c+d\tan(e+fx))^{3/2}}{3b^2(a^2+b^2)f}$$

$$- \frac{(Ab^2 - a(bB-aC))\,(c+d\tan(e+fx))^{5/2}}{b(a^2+b^2)f(a+b\tan(e+fx))}$$

$$+ \frac{4\int \frac{-\frac{3}{4}b^3\left(b^2\left(Ac^3-c^3C-3Bc^2d-3Acd^2+3cCd^2+Bd^3\right)+a^2\left(c^3C+3Bc^2d-3cCd^2-Bd^3-A\left(c^3-3cd^2\right)\right)-2ab\left((A-C)d\left(3c^2-d^2\right)+B\right.}{}}{}$$

$$- \frac{((bc-ad)^2\,(3a^3bBd - 5a^4Cd - b^4(2Bc+5Ad) - ab^3(4Ac-4cC-7Bd) + a^2b^2(2Bc-(A+9C}{2b^3(a^2+b^2)^2}$$

$$= -\frac{d(5a^3Cd - Ab^2(bc-ad) - 2b^3(2cC+Bd) - a^2b(5cC+3Bd) + ab^2(Bc+4Cd))\sqrt{c+d\tan(e+}}{b^3(a^2+b^2)f}$$

$$+ \frac{(3Ab^2 - 3abB + 5a^2C + 2b^2C)\,d(c+d\tan(e+fx))^{3/2}}{3b^2(a^2+b^2)f}$$

$$- \frac{(Ab^2 - a(bB-aC))\,(c+d\tan(e+fx))^{5/2}}{b(a^2+b^2)f(a+b\tan(e+fx))}$$

$$+ \frac{((A-iB-C)(c-id)^3)\int \frac{1+i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(a-ib)^2}$$

$$+ \frac{((A+iB-C)(c+id)^3)\int \frac{1-i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(a+ib)^2}$$

$$- \frac{((bc-ad)^2\,(3a^3bBd - 5a^4Cd - b^4(2Bc+5Ad) - ab^3(4Ac-4cC-7Bd) + a^2b^2(2Bc-(A+9C}{2b^3(a^2+b^2)^2 f}$$

$$= -\frac{d(5a^3Cd - Ab^2(bc-ad) - 2b^3(2cC+Bd) - a^2b(5cC+3Bd) + ab^2(Bc+4Cd))\sqrt{c+d\tan(e}{b^3(a^2+b^2)f} + \frac{(3Ab^2 - 3abB + 5a^2C + 2b^2C)\,d(c+d\tan(e+fx))^{3/2}}{3b^2(a^2+b^2)f} - \frac{(Ab^2 - a(bB-aC))(c+d\tan(e+fx))^{5/2}}{b(a^2+b^2)f(a+b\tan(e+fx))} + \frac{((iA+B-iC)(c-id)^3)\,\text{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c-idx}}\,dx, x, i\tan(e+fx)\right)}{2(a-ib)^2f} - \frac{(i(A+iB-C)(c+id)^3)\,\text{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c+idx}}\,dx, x, -i\tan(e+fx)\right)}{2(a+ib)^2f} - \frac{((bc-ad)^2(3a^3bBd - 5a^4Cd - b^4(2Bc+5Ad) - ab^3(4Ac-4cC-7Bd) + a^2b^2(2Bc-(A+9}{b^3(a^2+b^2)^2df}$$

$$= \frac{(bc-ad)^{3/2}(3a^3bBd - 5a^4Cd - b^4(2Bc+5Ad) - ab^3(4Ac-4cC-7Bd) + a^2b^2(2Bc-(A+9C}{b^{7/2}(a^2+b^2)^2f} - \frac{d(5a^3Cd - Ab^2(bc-ad) - 2b^3(2cC+Bd) - a^2b(5cC+3Bd) + ab^2(Bc+4Cd))\sqrt{c+d\tan(e}{b^3(a^2+b^2)f} + \frac{(3Ab^2 - 3abB + 5a^2C + 2b^2C)\,d(c+d\tan(e+fx))^{3/2}}{3b^2(a^2+b^2)f} - \frac{(Ab^2 - a(bB-aC))(c+d\tan(e+fx))^{5/2}}{b(a^2+b^2)f(a+b\tan(e+fx))} - \frac{((A-iB-C)(c-id)^3)\,\text{Subst}\left(\int \frac{1}{-1-\frac{ic}{d}+\frac{ix^2}{d}}\,dx, x, \sqrt{c+d\tan(e+fx)}\right)}{(a-ib)^2df} - \frac{((A+iB-C)(c+id)^3)\,\text{Subst}\left(\int \frac{1}{-1+\frac{ic}{d}-\frac{ix^2}{d}}\,dx, x, \sqrt{c+d\tan(e+fx)}\right)}{(a+ib)^2df}$$

$$= -\frac{(iA+B-iC)(c-id)^{5/2}\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(a-ib)^2f}$$
$$-\frac{(B-i(A-C))(c+id)^{5/2}\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(a+ib)^2f}$$
$$+\frac{(bc-ad)^{3/2}\left(3a^3bBd-5a^4Cd-b^4(2Bc+5Ad)-ab^3(4Ac-4cC-7Bd)+a^2b^2(2Bc-(A+9\right.}{b^{7/2}\left(a^2+b^2\right)^2f}$$
$$-\frac{d(5a^3Cd-Ab^2(bc-ad)-2b^3(2cC+Bd)-a^2b(5cC+3Bd)+ab^2(Bc+4Cd))\sqrt{c+d\tan(e+}}{b^3\left(a^2+b^2\right)f}$$
$$+\frac{(3Ab^2-3abB+5a^2C+2b^2C)\,d(c+d\tan(e+fx))^{3/2}}{3b^2\left(a^2+b^2\right)f}$$
$$-\frac{\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^{5/2}}{b\left(a^2+b^2\right)f(a+b\tan(e+fx))}$$

Mathematica [B] (verified)

Both result and optimal contain complex but leaf count is larger than twice the leaf count of optimal. 6112 vs. 2(473) = 946.

Time = 7.02 (sec) , antiderivative size = 6112, normalized size of antiderivative = 12.92

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^2}\,dx = \text{Result too large to show}$$

```
[In] Integrate[((c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/(a + b*Tan[e + f*x])^2,x]
```

```
[Out] Result too large to show
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 14118 vs. 2(434) = 868.

Time = 0.20 (sec) , antiderivative size = 14119, normalized size of antiderivative = 29.85

method	result	size
derivativedivides	Expression too large to display	14119
default	Expression too large to display	14119

```
[In] int((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^2,x,method=_RETURNVERBOSE)
```

```
[Out] result too large to display
```

Fricas [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^2,x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))**(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e))**2,x)
```

```
[Out] Timed out
```

Maxima [F(-2)]

Exception generated.

$$\int \frac{(c+d\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^2}\,dx = \text{Exception raised: ValueError}$$

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^2,x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 detail
```

Giac [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^2}\,dx=\text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^2,x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^2}\,dx=\text{Hanged}$$

```
[In] int(((c + d*tan(e + f*x))^(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x))^2,x)
```

```
[Out] \text{Hanged}
```

3.109 $\int \frac{(c+d\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx$

Optimal result 1435
Rubi [A] (verified) 1436
Mathematica [B] (verified) 1441
Maple [B] (verified) 1441
Fricas [F(-1)] 1442
Sympy [F(-1)] 1442
Maxima [F(-2)] 1442
Giac [F(-1)] 1443
Mupad [F(-1)] 1443

Optimal result

Integrand size = 47, antiderivative size = 643

$$\int \frac{(c+d\tan(e+fx))^{5/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx =$$

$$-\frac{(A-iB-C)(c-id)^{5/2}\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(ia+b)^3f}$$

$$+\frac{(A+iB-C)(c+id)^{5/2}\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(ia-b)^3f}$$

$$+\frac{\sqrt{bc-ad}(3a^5bBd^2-15a^6Cd^2+a^4b^2d(4Bc+(A-46C)d)-3a^2b^4(8Ac^2-8c^2C-16Bcd-6Ad^2+21C}{}$$

$$-\frac{d(3a^3bBd-15a^4Cd-ab^3(8Ac-8cC-11Bd)+a^2b^2(4Bc+(A-31C)d)-b^4(4Bc+7Ad+8Cd))\sqrt{c}}{4b^3\left(a^2+b^2\right)^2f}$$

$$+\frac{(a^3bBd-5a^4Cd-b^4(4Bc+5Ad)-ab^3(8Ac-8cC-9Bd)+a^2b^2(4Bc+3Ad-13Cd))\,(c+d\tan(e+}{4b^2\left(a^2+b^2\right)^2f(a+b\tan(e+fx))}$$

$$-\frac{(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{5/2}}{2b\,(a^2+b^2)\,f(a+b\tan(e+fx))^2}$$

```
[Out] -(A-I*B-C)*(c-I*d)^(5/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(I*a
+b)^3/f+(A+I*B-C)*(c+I*d)^(5/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2
))/(I*a-b)^3/f+1/4*(3*a^5*b*B*d^2-15*a^6*C*d^2+a^4*b^2*d*(4*B*c+(A-46*C)*d)
-3*a^2*b^4*(8*A*c^2-6*A*d^2-16*B*c*d-8*C*c^2+21*C*d^2)-a*b^5*(56*c*(A-C)*d+
B*(24*c^2-35*d^2))-b^6*(4*c*(5*B*d+2*C*c)-A*(8*c^2-15*d^2))+2*a^3*b^3*(4*c*
(A-C)*d+B*(4*c^2+3*d^2)))*arctanh(b^(1/2)*(c+d*tan(f*x+e))^(1/2)/(-a*d+b*c)
^(1/2))*(-a*d+b*c)^(1/2)/b^(7/2)/(a^2+b^2)^3/f-1/4*d*(3*a^3*b*B*d-15*a^4*C*
d-a*b^3*(8*A*c-11*B*d-8*C*c)+a^2*b^2*(4*B*c+(A-31*C)*d)-b^4*(7*A*d+4*B*c+8*
```

```
C*d))*(c+d*tan(f*x+e))^(1/2)/b^3/(a^2+b^2)^2/f+1/4*(a^3*b*B*d-5*a^4*C*d-b^4
*(5*A*d+4*B*c)-a*b^3*(8*A*c-9*B*d-8*C*c)+a^2*b^2*(3*A*d+4*B*c-13*C*d))*(c+d
*tan(f*x+e))^(3/2)/b^2/(a^2+b^2)^2/f/(a+b*tan(f*x+e))-1/2*(A*b^2-a*(B*b-C*a
))*(c+d*tan(f*x+e))^(5/2)/b/(a^2+b^2)/f/(a+b*tan(f*x+e))^2
```

Rubi [A] (verified)

Time = 7.36 (sec) , antiderivative size = 643, normalized size of antiderivative = 1.00, number of steps used = 14, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.170, Rules used = {3726, 3728, 3734, 3620, 3618, 65, 214, 3715}

$$\int \frac{(c+d\tan(e+fx))^{5/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^3}\,dx =$$
$$-\frac{(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{5/2}}{2bf\,(a^2+b^2)\,(a+b\tan(e+fx))^2}$$
$$+\frac{(c+d\tan(e+fx))^{3/2}\,(-5a^4Cd+a^3bBd+a^2b^2(3Ad+4Bc-13Cd)-ab^3(8Ac-9Bd-8cC)-b^4(5Ad+\dots}{4b^2f\,(a^2+b^2)^2\,(a+b\tan(e+fx))}$$
$$-\frac{d\sqrt{c+d\tan(e+fx)}(-15a^4Cd+3a^3bBd+a^2b^2(d(A-31C)+4Bc)-ab^3(8Ac-11Bd-8cC)-b^4(7Ac\dots}{4b^3f\,(a^2+b^2)^2}$$
$$+\frac{\sqrt{bc-ad}(-15a^6Cd^2+3a^5bBd^2+a^4b^2d(d(A-46C)+4Bc)+2a^3b^3(4cd(A-C)+B(4c^2+3d^2))-3a^2b^4\dots}{\dots}$$
$$-\frac{(c-id)^{5/2}(A-iB-C)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f(b+ia)^3}$$
$$+\frac{(c+id)^{5/2}(A+iB-C)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f(-b+ia)^3}$$

```
[In] Int[((c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a
 + b*Tan[e + f*x])^3,x]
```

```
[Out] -(((A - I*B - C)*(c - I*d)^(5/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c -
I*d]])/((I*a + b)^3*f)) + ((A + I*B - C)*(c + I*d)^(5/2)*ArcTanh[Sqrt[c + d
*Tan[e + f*x]]/Sqrt[c + I*d]])/((I*a - b)^3*f) + (Sqrt[b*c - a*d]*(3*a^5*b*
B*d^2 - 15*a^6*C*d^2 + a^4*b^2*d*(4*B*c + (A - 46*C)*d) - 3*a^2*b^4*(8*A*c^
2 - 8*c^2*C - 16*B*c*d - 6*A*d^2 + 21*C*d^2) - a*b^5*(56*c*(A - C)*d + B*(2
4*c^2 - 35*d^2)) - b^6*(4*c*(2*c*C + 5*B*d) - A*(8*c^2 - 15*d^2)) + 2*a^3*b
^3*(4*c*(A - C)*d + B*(4*c^2 + 3*d^2)))*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e +
 f*x]])/Sqrt[b*c - a*d]])/(4*b^(7/2)*(a^2 + b^2)^3*f) - (d*(3*a^3*b*B*d - 1
5*a^4*C*d - a*b^3*(8*A*c - 8*c*C - 11*B*d) + a^2*b^2*(4*B*c + (A - 31*C)*d)
 - b^4*(4*B*c + 7*A*d + 8*C*d))*Sqrt[c + d*Tan[e + f*x]])/(4*b^3*(a^2 + b^2
)^2*f) + ((a^3*b*B*d - 5*a^4*C*d - b^4*(4*B*c + 5*A*d) - a*b^3*(8*A*c - 8*c
*C - 9*B*d) + a^2*b^2*(4*B*c + 3*A*d - 13*C*d))*(c + d*Tan[e + f*x])^(3/2))
```

```
/(4*b^2*(a^2 + b^2)^2*f*(a + b*Tan[e + f*x])) - ((A*b^2 - a*(b*B - a*C))*(c
 + d*Tan[e + f*x])^(5/2))/(2*b*(a^2 + b^2)*f*(a + b*Tan[e + f*x])^2)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3715

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_) + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :>
 Dist[A/f, Subst[Int[(a + b*x)^m*(c + d*x)^n, x], x, Tan[e + f*x]], x] /; F
reeQ[{a, b, c, d, e, f, A, C, m, n}, x] && EqQ[A, C]
```

Rule 3726

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*d^2 + c*(c*C - B*d))*(a + b*Tan[e +
 f*x])^m*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
```

```
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3734

```
Int[(((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2))/((a_.) + (b_.)*tan[(e_.)
+ (f_.)*(x_)]), x_Symbol] :> Dist[1/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n
*Simp[b*B + a*(A - C) + (a*B - b*(A - C))*Tan[e + f*x], x], x], x] + Dist[(
A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n*((1 + Tan[e
+ f*x]^2)/(a + b*Tan[e + f*x])), x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C
, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&
 !GtQ[n, 0] &&  !LeQ[n, -1]
```

Rubi steps

$$\text{integral} = -\frac{\left(Ab^2-a(bB-aC)\right)\left(c+d\tan(e+fx)\right)^{5/2}}{2b\left(a^2+b^2\right)f(a+b\tan(e+fx))^2} + \frac{\int \frac{(c+d\tan(e+fx))^{3/2}\left(\frac{1}{2}\left(2(bB-aC)\left(2bc-\frac{5ad}{2}\right)+2Ab\left(2ac+\frac{5bd}{2}\right)\right)-2b((A-C)(bc-ad)-B(ac+bd))\tan(e+fx)+\frac{1}{2}\left(Ab^2-abB+5a^2C+4b^2C\right)}{(a+b\tan(e+fx))^2}}{2b\left(a^2+b^2\right)}$$

$$= \frac{\left(a^3bBd-5a^4Cd-b^4(4Bc+5Ad)-ab^3(8Ac-8cC-9Bd)+a^2b^2(4Bc+3Ad-13Cd)\right)(c+d\tan}{4b^2\left(a^2+b^2\right)^2f(a+b\tan(e+fx))} - \frac{\left(Ab^2-a(bB-aC)\right)\left(c+d\tan(e+fx)\right)^{5/2}}{2b\left(a^2+b^2\right)f(a+b\tan(e+fx))^2} + \frac{\int \frac{\sqrt{c+d\tan(e+fx)}\left(\frac{1}{4}\left(b(2ac+3bd)\left(5a^2Cd+b^2(4Bc+5Ad)+ab(4Ac-4cC-5Bd)\right)+(2bc-3ad)\left(a^2bBd-5a^3Cd-Ab^2(4bc-3ad)+4b^3\right.\right.\right.}{}}{}$$

$$= -\frac{d(3a^3bBd - 15a^4Cd - ab^3(8Ac - 8cC - 11Bd) + a^2b^2(4Bc + (A - 31C)d) - b^4(4Bc + 7Ad + }{4b^3\left(a^2+b^2\right)^2 f}$$
$$+ \frac{(a^3bBd - 5a^4Cd - b^4(4Bc + 5Ad) - ab^3(8Ac - 8cC - 9Bd) + a^2b^2(4Bc + 3Ad - 13Cd))\,(c + }{4b^2\left(a^2+b^2\right)^2 f(a + b\tan(e + fx))}$$
$$- \frac{(Ab^2 - a(bB - aC))\,(c + d\tan(e + fx))^{5/2}}{2b\,(a^2 + b^2)\,f(a + b\tan(e + fx))^2}$$
$$+ \frac{\int \frac{\frac{1}{8}\left(-15a^5Cd^3+3a^4bd^2(5cC+Bd)+a^3b^2d^2(Bc+(A-31C)d)-b^5c\left(8Ac^2-8c^2C-20Bcd-15Ad^2\right)+a^2b^3\left(8Ac^3-8c^3C-20Bc^2d-17\right.}{}}{}$$

$$= -\frac{d(3a^3bBd - 15a^4Cd - ab^3(8Ac - 8cC - 11Bd) + a^2b^2(4Bc + (A - 31C)d) - b^4(4Bc + 7Ad + }{4b^3\left(a^2+b^2\right)^2 f}$$
$$+ \frac{(a^3bBd - 5a^4Cd - b^4(4Bc + 5Ad) - ab^3(8Ac - 8cC - 9Bd) + a^2b^2(4Bc + 3Ad - 13Cd))\,(c + }{4b^2\left(a^2+b^2\right)^2 f(a + b\tan(e + fx))}$$
$$- \frac{(Ab^2 - a(bB - aC))\,(c + d\tan(e + fx))^{5/2}}{2b\,(a^2 + b^2)\,f(a + b\tan(e + fx))^2}$$
$$+ \frac{\int \frac{-b^3\left(3ab^2\left(Ac^3-c^3C-3Bc^2d-3Acd^2+3cCd^2+Bd^3\right)+a^3\left(c^3C+3Bc^2d-3cCd^2-Bd^3-A\left(c^3-3cd^2\right)\right)-3a^2b\left((A-C)d\left(3c^2-d^2\right)+\right.\right.}{}}{}$$
$$- \frac{\left((bc - ad)\left(3a^5bBd^2 - 15a^6Cd^2 + a^4b^2d(4Bc + (A - 46C)d) - 3a^2b^4(8Ac^2 - 8c^2C - 16Bcd - \right.\right.}{}$$

$$= -\frac{d(3a^3bBd - 15a^4Cd - ab^3(8Ac - 8cC - 11Bd) + a^2b^2(4Bc + (A - 31C)d) - b^4(4Bc + 7Ad + }{4b^3\left(a^2+b^2\right)^2 f}$$
$$+ \frac{(a^3bBd - 5a^4Cd - b^4(4Bc + 5Ad) - ab^3(8Ac - 8cC - 9Bd) + a^2b^2(4Bc + 3Ad - 13Cd))\,(c + }{4b^2\left(a^2+b^2\right)^2 f(a + b\tan(e + fx))}$$
$$- \frac{(Ab^2 - a(bB - aC))\,(c + d\tan(e + fx))^{5/2}}{2b\,(a^2 + b^2)\,f(a + b\tan(e + fx))^2}$$
$$+ \frac{((A - iB - C)(c - id)^3)\int \frac{1 + i\tan(e + fx)}{\sqrt{c + d\tan(e + fx)}}\,dx}{2(a - ib)^3}$$
$$+ \frac{((A + iB - C)(c + id)^3)\int \frac{1 - i\tan(e + fx)}{\sqrt{c + d\tan(e + fx)}}\,dx}{2(a + ib)^3}$$
$$- \frac{\left((bc - ad)\left(3a^5bBd^2 - 15a^6Cd^2 + a^4b^2d(4Bc + (A - 46C)d) - 3a^2b^4(8Ac^2 - 8c^2C - 16Bcd - \right.\right.}{}$$

$$= -\frac{d(3a^3bBd - 15a^4Cd - ab^3(8Ac - 8cC - 11Bd) + a^2b^2(4Bc + (A - 31C)d) - b^4(4Bc + 7Ad + 8}{4b^3(a^2+b^2)^2 f}$$

$$+ \frac{(a^3bBd - 5a^4Cd - b^4(4Bc + 5Ad) - ab^3(8Ac - 8cC - 9Bd) + a^2b^2(4Bc + 3Ad - 13Cd))(c + d}{4b^2(a^2+b^2)^2 f(a + b\tan(e+fx))}$$

$$- \frac{(Ab^2 - a(bB - aC))(c + d\tan(e+fx))^{5/2}}{2b(a^2+b^2) f(a + b\tan(e+fx))^2}$$

$$+ \frac{((A - iB - C)(c - id)^3)\,\mathrm{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c-idx}}\,dx, x, i\tan(e+fx)\right)}{2(ia+b)^3 f}$$

$$- \frac{((A + iB - C)(c + id)^3)\,\mathrm{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c+idx}}\,dx, x, -i\tan(e+fx)\right)}{2(ia-b)^3 f}$$

$$- \frac{((bc - ad)(3a^5bBd^2 - 15a^6Cd^2 + a^4b^2d(4Bc + (A - 46C)d) - 3a^2b^4(8Ac^2 - 8c^2C - 16Bcd - 6}{}$$

$$= \frac{\sqrt{bc-ad}(3a^5bBd^2 - 15a^6Cd^2 + a^4b^2d(4Bc + (A - 46C)d) - 3a^2b^4(8Ac^2 - 8c^2C - 16Bcd - 6Ad^2}{}$$

$$- \frac{d(3a^3bBd - 15a^4Cd - ab^3(8Ac - 8cC - 11Bd) + a^2b^2(4Bc + (A - 31C)d) - b^4(4Bc + 7Ad + 8}{4b^3(a^2+b^2)^2 f}$$

$$+ \frac{(a^3bBd - 5a^4Cd - b^4(4Bc + 5Ad) - ab^3(8Ac - 8cC - 9Bd) + a^2b^2(4Bc + 3Ad - 13Cd))(c + d}{4b^2(a^2+b^2)^2 f(a + b\tan(e+fx))}$$

$$- \frac{(Ab^2 - a(bB - aC))(c + d\tan(e+fx))^{5/2}}{2b(a^2+b^2) f(a + b\tan(e+fx))^2}$$

$$- \frac{((A - iB - C)(c - id)^3)\,\mathrm{Subst}\left(\int \frac{1}{-1-\frac{ic}{d}+\frac{ix^2}{d}}\,dx, x, \sqrt{c + d\tan(e+fx)}\right)}{(a-ib)^3 df}$$

$$- \frac{((A + iB - C)(c + id)^3)\,\mathrm{Subst}\left(\int \frac{1}{-1+\frac{ic}{d}-\frac{ix^2}{d}}\,dx, x, \sqrt{c + d\tan(e+fx)}\right)}{(a+ib)^3 df}$$

$$= -\frac{(A-iB-C)(c-id)^{5/2}\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(ia+b)^3f}$$
$$+\frac{(A+iB-C)(c+id)^{5/2}\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(ia-b)^3f}$$
$$+\frac{\sqrt{bc-ad}(3a^5bBd^2-15a^6Cd^2+a^4b^2d(4Bc+(A-46C)d)-3a^2b^4(8Ac^2-8c^2C-16Bcd-6}{}$$
$$-\frac{d(3a^3bBd-15a^4Cd-ab^3(8Ac-8cC-11Bd)+a^2b^2(4Bc+(A-31C)d)-b^4(4Bc+7Ad+}{4b^3\left(a^2+b^2\right)^2f}$$
$$+\frac{(a^3bBd-5a^4Cd-b^4(4Bc+5Ad)-ab^3(8Ac-8cC-9Bd)+a^2b^2(4Bc+3Ad-13Cd))\,(c+}{4b^2\left(a^2+b^2\right)^2f(a+b\tan(e+fx))}$$
$$-\frac{\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^{5/2}}{2b\left(a^2+b^2\right)f(a+b\tan(e+fx))^2}$$

Mathematica [B] (verified)

Both result and optimal contain complex but leaf count is larger than twice the leaf count of optimal. 17248 vs. 2(643) = 1286.

Time = 7.95 (sec) , antiderivative size = 17248, normalized size of antiderivative = 26.82

$$\int\frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^3}\,dx=\text{Result too large to show}$$

```
[In] Integrate[((c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/(a + b*Tan[e + f*x])^3,x]
```

```
[Out] Result too large to show
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 20662 vs. 2(599) = 1198.

Time = 0.30 (sec) , antiderivative size = 20663, normalized size of antiderivative = 32.14

method	result	size
derivativedivides	Expression too large to display	20663
default	Expression too large to display	20663

```
[In] int((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^3,x,method=_RETURNVERBOSE)
```

```
[Out] result too large to display
```

Fricas [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^3}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^3,x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^3}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))**(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e))**3,x)
```

```
[Out] Timed out
```

Maxima [F(-2)]

Exception generated.

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^3}\,dx = \text{Exception raised: ValueError}$$

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^3,x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 detail
```

Giac [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^3}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^3,x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^3}\,dx = \text{Hanged}$$

```
[In] int(((c + d*tan(e + f*x))^(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x))^3,x)
```

```
[Out] \text{Hanged}
```

3.110 $\int \frac{(a+b\tan(e+fx))^3(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{c+d\tan(e+fx)}}\,dx$

Optimal result 1444
Rubi [A] (verified) 1445
Mathematica [B] (verified) 1449
Maple [B] (verified) 1451
Fricas [B] (verification not implemented) 1451
Sympy [F] 1451
Maxima [F(-1)] 1452
Giac [F(-1)] 1452
Mupad [B] (verification not implemented) 1452

Optimal result

Integrand size = 47, antiderivative size = 407

$$\int \frac{(a+b\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx$$
$$= \frac{(ia+b)^3(A-iB-C)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{\sqrt{c-id}f}$$
$$- \frac{(ia-b)^3(A+iB-C)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{\sqrt{c+id}f}$$
$$+ \frac{2(72a^3Cd^3-6a^2bd^2(32cC-49Bd)+21ab^2d(8c^2C-10Bcd+15(A-C)d^2)-b^3(48c^3C-56Bc^2d+7}{105d^4f}$$
$$+ \frac{2b(35b(Ab+aB-bC)d^2+4(bc-ad)(6bcC-7bBd-6aCd))\tan(e+fx)\sqrt{c+d\tan(e+fx)}}{105d^3f}$$
$$- \frac{2(6bcC-7bBd-6aCd)(a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}}{35d^2f}$$
$$+ \frac{2C(a+b\tan(e+fx))^3\sqrt{c+d\tan(e+fx)}}{7df}$$

```
[Out] (I*a+b)^3*(A-I*B-C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/f/(c-I*d)
^(1/2)-(I*a-b)^3*(A+I*B-C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))/f/
(c+I*d)^(1/2)+2/105*(72*a^3*C*d^3-6*a^2*b*d^2*(-49*B*d+32*C*c)+21*a*b^2*d*(
8*c^2*C-10*B*c*d+15*(A-C)*d^2)-b^3*(48*c^3*C-56*B*c^2*d+70*c*(A-C)*d^2+105*
B*d^3))*(c+d*tan(f*x+e))^(1/2)/d^4/f+2/105*b*(35*b*(A*b+B*a-C*b)*d^2+4*(-a*
d+b*c)*(-7*B*b*d-6*C*a*d+6*C*b*c))*(c+d*tan(f*x+e))^(1/2)*tan(f*x+e)/d^3/f-
2/35*(-7*B*b*d-6*C*a*d+6*C*b*c)*(c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^2/d
^2/f+2/7*C*(c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^3/d/f
```

Rubi [A] (verified)

Time = 1.91 (sec) , antiderivative size = 407, normalized size of antiderivative = 1.00, number of steps used = 11, number of rules used = 7, $\frac{\text{number of rules}}{\text{integrand size}} = 0.149$, Rules used = {3728, 3718, 3711, 3620, 3618, 65, 214}

$$\int \frac{(a+b\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{c+d\tan(e+fx)}}\,dx$$
$$= \frac{2\sqrt{c+d\tan(e+fx)}(72a^3Cd^3-6a^2bd^2(32cC-49Bd)+21ab^2d(15d^2(A-C)-10Bcd+8c^2C)-(b^3(7}{105d^4f}$$
$$-\frac{(-b+ia)^3(A+iB-C)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f\sqrt{c+id}}$$
$$+\frac{(b+ia)^3(A-iB-C)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f\sqrt{c-id}}$$
$$+\frac{2b\tan(e+fx)\sqrt{c+d\tan(e+fx)}(35bd^2(aB+Ab-bC)+4(bc-ad)(-6aCd-7bBd+6bcC))}{105d^3f}$$
$$-\frac{2(-6aCd-7bBd+6bcC)(a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}}{35d^2f}$$
$$+\frac{2C(a+b\tan(e+fx))^3\sqrt{c+d\tan(e+fx)}}{7df}$$

```
[In] Int[((a + b*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/Sqrt[c
 + d*Tan[e + f*x]],x]
```

```
[Out] ((I*a + b)^3*(A - I*B - C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]])
/(Sqrt[c - I*d]*f) - ((I*a - b)^3*(A + I*B - C)*ArcTanh[Sqrt[c + d*Tan[e +
f*x]]/Sqrt[c + I*d]])/(Sqrt[c + I*d]*f) + (2*(72*a^3*C*d^3 - 6*a^2*b*d^2*(3
2*c*C - 49*B*d) + 21*a*b^2*d*(8*c^2*C - 10*B*c*d + 15*(A - C)*d^2) - b^3*(4
8*c^3*C - 56*B*c^2*d + 70*c*(A - C)*d^2 + 105*B*d^3))*Sqrt[c + d*Tan[e + f*
x]])/(105*d^4*f) + (2*b*(35*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*
C - 7*b*B*d - 6*a*C*d))*Tan[e + f*x]*Sqrt[c + d*Tan[e + f*x]])/(105*d^3*f)
- (2*(6*b*c*C - 7*b*B*d - 6*a*C*d)*(a + b*Tan[e + f*x])^2*Sqrt[c + d*Tan[e
+ f*x]])/(35*d^2*f) + (2*C*(a + b*Tan[e + f*x])^3*Sqrt[c + d*Tan[e + f*x]])
/(7*d*f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
```

```
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\text{integral} = \frac{2C(a+b\tan(e+fx))^3\sqrt{c+d\tan(e+fx)}}{7df} + \frac{2\int \frac{(a+b\tan(e+fx))^2\left(\frac{1}{2}(-6bcC+a(7A-C)d)+\frac{7}{2}(Ab+aB-bC)d\tan(e+fx)-\frac{1}{2}(6bcC-7bBd-6aCd)\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx}{7d}$$

$$= -\frac{2(6bcC-7bBd-6aCd)(a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}}{35d^2f} + \frac{2C(a+b\tan(e+fx))^3\sqrt{c+d\tan(e+fx)}}{7df} + \frac{4\int \frac{(a+b\tan(e+fx))\left(\frac{1}{4}(-5ad(6bcC-a(7A-C)d)+(4bc+ad)(6bcC-7bBd-6aCd))+\frac{35}{4}(a^2B-b^2B+2ab(A-C))d^2\tan(e+fx)+\frac{1}{4}\right.}{\sqrt{c+d\tan(e+fx)}}}{35d^2}$$

$$= \frac{2b(35b(Ab+aB-bC)d^2+4(bc-ad)(6bcC-7bBd-6aCd))\tan(e+fx)\sqrt{c+d\tan(e+fx)}}{105d^3f} - \frac{2(6bcC-7bBd-6aCd)(a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}}{35d^2f} + \frac{2C(a+b\tan(e+fx))^3\sqrt{c+d\tan(e+fx)}}{7df} - \frac{8\int \frac{1}{8}(-3a^3(35A-11C)d^3-42ab^2cd(4cC-5Bd)+3a^2bd^2(64cC+7Bd)+2b^3c(24c^2C-28Bcd+35(A-C)d^2))-\frac{105}{8}(a^3B-3ab^2B+}{}$$

$$= \frac{2(72a^3Cd^3-6a^2bd^2(32cC-49Bd)+21ab^2d(8c^2C-10Bcd+15(A-C)d^2)-b^3(48c^3C-56Bc}{105d^4f} + \frac{2b(35b(Ab+aB-bC)d^2+4(bc-ad)(6bcC-7bBd-6aCd))\tan(e+fx)\sqrt{c+d\tan(e+fx)}}{105d^3f} - \frac{2(6bcC-7bBd-6aCd)(a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}}{35d^2f} + \frac{2C(a+b\tan(e+fx))^3\sqrt{c+d\tan(e+fx)}}{7df} - \frac{8\int \frac{\frac{105}{8}(3a^2bB-b^3B-a^3(A-C)+3ab^2(A-C))d^3-\frac{105}{8}(a^3B-3ab^2B+3a^2b(A-C)-b^3(A-C))d^3\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{105d^3}$$

$$= \frac{2(72a^3Cd^3 - 6a^2bd^2(32cC - 49Bd) + 21ab^2d(8c^2C - 10Bcd + 15(A-C)d^2) - b^3(48c^3C - 56Bc^2}{105d^4f}$$
$$+ \frac{2b(35b(Ab + aB - bC)d^2 + 4(bc - ad)(6bcC - 7bBd - 6aCd))\tan(e+fx)\sqrt{c + d\tan(e+fx)}}{105d^3f}$$
$$- \frac{2(6bcC - 7bBd - 6aCd)(a + b\tan(e+fx))^2\sqrt{c + d\tan(e+fx)}}{35d^2f}$$
$$+ \frac{2C(a + b\tan(e+fx))^3\sqrt{c + d\tan(e+fx)}}{7df}$$
$$+ \frac{1}{2}\left((a-ib)^3(A - iB - C)\right)\int \frac{1 + i\tan(e+fx)}{\sqrt{c + d\tan(e+fx)}}\,dx$$
$$+ \frac{1}{2}\left((a+ib)^3(A + iB - C)\right)\int \frac{1 - i\tan(e+fx)}{\sqrt{c + d\tan(e+fx)}}\,dx$$

$$= \frac{2(72a^3Cd^3 - 6a^2bd^2(32cC - 49Bd) + 21ab^2d(8c^2C - 10Bcd + 15(A-C)d^2) - b^3(48c^3C - 56Bc^2}{105d^4f}$$
$$+ \frac{2b(35b(Ab + aB - bC)d^2 + 4(bc - ad)(6bcC - 7bBd - 6aCd))\tan(e+fx)\sqrt{c + d\tan(e+fx)}}{105d^3f}$$
$$- \frac{2(6bcC - 7bBd - 6aCd)(a + b\tan(e+fx))^2\sqrt{c + d\tan(e+fx)}}{35d^2f}$$
$$+ \frac{2C(a + b\tan(e+fx))^3\sqrt{c + d\tan(e+fx)}}{7df}$$
$$- \frac{(i(a+ib)^3(A + iB - C))\,\text{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c+idx}}\,dx, x, -i\tan(e+fx)\right)}{2f}$$
$$+ \frac{((a-ib)^3(iA + B - iC))\,\text{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c-idx}}\,dx, x, i\tan(e+fx)\right)}{2f}$$

$$= \frac{2(72a^3Cd^3 - 6a^2bd^2(32cC - 49Bd) + 21ab^2d(8c^2C - 10Bcd + 15(A-C)d^2) - b^3(48c^3C - 56Bc}{105d^4f}$$
$$+ \frac{2b(35b(Ab + aB - bC)d^2 + 4(bc - ad)(6bcC - 7bBd - 6aCd))\tan(e + fx)\sqrt{c + d\tan(e + fx)}}{105d^3f}$$
$$- \frac{2(6bcC - 7bBd - 6aCd)(a + b\tan(e + fx))^2\sqrt{c + d\tan(e + fx)}}{35d^2f}$$
$$+ \frac{2C(a + b\tan(e + fx))^3\sqrt{c + d\tan(e + fx)}}{7df}$$
$$- \frac{((a - ib)^3(A - iB - C))\,\mathrm{Subst}\left(\int \frac{1}{-1 - \frac{ic}{d} + \frac{ix^2}{d}}\,dx, x, \sqrt{c + d\tan(e + fx)}\right)}{df}$$
$$- \frac{((a + ib)^3(A + iB - C))\,\mathrm{Subst}\left(\int \frac{1}{-1 + \frac{ic}{d} - \frac{ix^2}{d}}\,dx, x, \sqrt{c + d\tan(e + fx)}\right)}{df}$$

$$= -\frac{(a - ib)^3(iA + B - iC)\mathrm{arctanh}\left(\frac{\sqrt{c + d\tan(e + fx)}}{\sqrt{c - id}}\right)}{\sqrt{c - id}f}$$
$$- \frac{(ia - b)^3(A + iB - C)\mathrm{arctanh}\left(\frac{\sqrt{c + d\tan(e + fx)}}{\sqrt{c + id}}\right)}{\sqrt{c + id}f}$$
$$+ \frac{2(72a^3Cd^3 - 6a^2bd^2(32cC - 49Bd) + 21ab^2d(8c^2C - 10Bcd + 15(A-C)d^2) - b^3(48c^3C - 56}{105d^4f}$$
$$+ \frac{2b(35b(Ab + aB - bC)d^2 + 4(bc - ad)(6bcC - 7bBd - 6aCd))\tan(e + fx)\sqrt{c + d\tan(e + fx)}}{105d^3f}$$
$$- \frac{2(6bcC - 7bBd - 6aCd)(a + b\tan(e + fx))^2\sqrt{c + d\tan(e + fx)}}{35d^2f}$$
$$+ \frac{2C(a + b\tan(e + fx))^3\sqrt{c + d\tan(e + fx)}}{7df}$$

Mathematica [B] (verified)

Both result and optimal contain complex but leaf count is larger than twice the leaf count of optimal. 1200 vs. $2(407) = 814$.

Time = 6.51 (sec) , antiderivative size = 1200, normalized size of antiderivative = 2.95

$$\int \frac{(a+b\tan(e+fx))^3 \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx$$
$$=\frac{2C(a+b\tan(e+fx))^3\sqrt{c+d\tan(e+fx)}}{7df}$$
$$+\frac{2\left(\frac{(-6bcC+7bBd+6aCd)(a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}}{5df}+\frac{2\left(\frac{b\left(35b(Ab+aB-bC)d^2+4(bc-ad)(6bcC-7bBd-6aCd)\right)\tan(e+fx)\sqrt{c+d\tan(e+}}{6df}\right.}{}\right.}{}$$

```
[In] Integrate[((a + b*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
Sqrt[c + d*Tan[e + f*x]],x]
```

```
[Out] (2*C*(a + b*Tan[e + f*x])^3*Sqrt[c + d*Tan[e + f*x]])/(7*d*f) + (2*(((-6*b*
c*C + 7*b*B*d + 6*a*C*d)*(a + b*Tan[e + f*x])^2*Sqrt[c + d*Tan[e + f*x]])/(
5*d*f) + (2*((b*(35*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C - 7*b*
B*d - 6*a*C*d))*Tan[e + f*x]*Sqrt[c + d*Tan[e + f*x]])/(6*d*f) - (2*((I*Sqr
t[c - I*d]*((b*c*(35*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C - 7*b
*B*d - 6*a*C*d)))/4 + (3*a*d*(35*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(6
*b*c*C - 7*b*B*d - 6*a*C*d)))/8 - (3*a*d*(-5*a*d*(6*b*c*C - a*(7*A - C)*d)
+ (4*b*c + a*d)*(6*b*c*C - 7*b*B*d - 6*a*C*d)))/8 - b*((-105*(a^2*B - b^2*B
 + 2*a*b*(A - C))*d^3)/8 + (c*(35*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(
6*b*c*C - 7*b*B*d - 6*a*C*d)))/4) + ((3*I)/2)*d*((35*a*(a^2*B - b^2*B + 2*a
*b*(A - C))*d^2)/4 - (b*(35*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*
C - 7*b*B*d - 6*a*C*d)))/4 + (b*(-5*a*d*(6*b*c*C - a*(7*A - C)*d) + (4*b*c
+ a*d)*(6*b*c*C - 7*b*B*d - 6*a*C*d)))/4))*ArcTanh[Sqrt[c + d*Tan[e + f*x]]
/Sqrt[c - I*d]])/((-c + I*d)*f) - (I*Sqrt[c + I*d]*((b*c*(35*b*(A*b + a*B -
 b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C - 7*b*B*d - 6*a*C*d)))/4 + (3*a*d*(35*b*
(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C - 7*b*B*d - 6*a*C*d)))/8 - (
3*a*d*(-5*a*d*(6*b*c*C - a*(7*A - C)*d) + (4*b*c + a*d)*(6*b*c*C - 7*b*B*d
- 6*a*C*d)))/8 - b*((-105*(a^2*B - b^2*B + 2*a*b*(A - C))*d^3)/8 + (c*(35*b
*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C - 7*b*B*d - 6*a*C*d)))/4) -
 ((3*I)/2)*d*((35*a*(a^2*B - b^2*B + 2*a*b*(A - C))*d^2)/4 - (b*(35*b*(A*b
+ a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C - 7*b*B*d - 6*a*C*d)))/4 + (b*(-5
*a*d*(6*b*c*C - a*(7*A - C)*d) + (4*b*c + a*d)*(6*b*c*C - 7*b*B*d - 6*a*C*d
)))/4))*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]])/((-c - I*d)*f) + (
2*((-3*a*d*(35*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C - 7*b*B*d -
 6*a*C*d)))/8 + b*((-105*(a^2*B - b^2*B + 2*a*b*(A - C))*d^3)/8 + (c*(35*b*
```

```
(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C - 7*b*B*d - 6*a*C*d)))/4))*S
qrt[c + d*Tan[e + f*x]])/(d*f)))/(3*d)))/(5*d)))/(7*d)
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 5977 vs. $2(371) = 742$.

Time = 0.31 (sec) , antiderivative size = 5978, normalized size of antiderivative = 14.69

method	result	size
parts	Expression too large to display	5978
derivativedivides	Expression too large to display	25426
default	Expression too large to display	25426

```
[In] int((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/
2),x,method=_RETURNVERBOSE)
```

```
[Out] result too large to display
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 37247 vs. $2(363) = 726$.

Time = 11.35 (sec) , antiderivative size = 37247, normalized size of antiderivative = 91.52

$$\int \frac{(a+b\tan(e+fx))^3 (A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{c+d\tan(e+fx)}}\,dx = \text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^(1/2),x, algorithm="fricas")
```

```
[Out] Too large to include
```

Sympy [F]

$$\int \frac{(a+b\tan(e+fx))^3 (A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{c+d\tan(e+fx)}}\,dx = \int \frac{(a+b\tan{(e+fx)})^3 (A+B\tan{(e+fx)}+C\tan^2{(e+fx)})}{\sqrt{c+d\tan{(e+fx)}}}\,dx$$

```
[In] integrate((a+b*tan(f*x+e))**3*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x
+e))**(1/2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**3*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/s
qrt(c + d*tan(e + f*x)), x)
```

Maxima [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{c+d\tan(e+fx)}}\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^(1/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

Giac [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{c+d\tan(e+fx)}}\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^(1/2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [B] (verification not implemented)

Time = 112.14 (sec) , antiderivative size = 28858, normalized size of antiderivative = 70.90

$$\int \frac{(a+b\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{c+d\tan(e+fx)}}\,dx = \text{Too large to display}$$

```
[In] int(((a + b*tan(e + f*x))^3*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d
*tan(e + f*x))^(1/2),x)
```

```
[Out] atan(((((8*(4*A*a^3*d^3*f^2 - 12*A*a*b^2*d^3*f^2 + 4*A*b^3*c*d^2*f^2 - 12*A
*a^2*b*c*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^6*
c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^2
*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)^2/4 - (16*c
^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 + 15*A^4*a^4*b^8
 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) - 4*A^2*a^6*c*f
^2 + 4*A^2*b^6*c*f^2 - 24*A^2*a*b^5*d*f^2 - 24*A^2*a^5*b*d*f^2 - 60*A^2*a^2
*b^4*c*f^2 + 60*A^2*a^4*b^2*c*f^2 + 80*A^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^
2*f^4)))^(1/2))*((((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f^2
+ 48*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 160*
A^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 + 6
```

*A^4*a^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) - 4*A^2*a^6*c*f^2 + 4*A^2*b^6*c*f^2 - 24*A^2*a*b^5*d*f^2 - 24*A^2*a^5*b*d*f^2 - 60*A^2*a^2*b^4*c*f^2 + 60*A^2*a^4*b^2*c*f^2 + 80*A^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^6*d^2 - A^2*b^6*d^2 + 15*A^2*a^2*b^4*d^2 - 15*A^2*a^4*b^2*d^2))/f^2)*((((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) - 4*A^2*a^6*c*f^2 + 4*A^2*b^6*c*f^2 - 24*A^2*a*b^5*d*f^2 - 24*A^2*a^5*b*d*f^2 - 60*A^2*a^2*b^4*c*f^2 + 60*A^2*a^4*b^2*c*f^2 + 80*A^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i - (((8*(4*A*a^3*d^3*f^2 - 12*A*a*b^2*d^3*f^2 + 4*A*b^3*c*d^2*f^2 - 12*A*a^2*b*c*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) - 4*A^2*a^6*c*f^2 + 4*A^2*b^6*c*f^2 - 24*A^2*a*b^5*d*f^2 - 24*A^2*a^5*b*d*f^2 - 60*A^2*a^2*b^4*c*f^2 + 60*A^2*a^4*b^2*c*f^2 + 80*A^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) - 4*A^2*a^6*c*f^2 + 4*A^2*b^6*c*f^2 - 24*A^2*a*b^5*d*f^2 - 24*A^2*a^5*b*d*f^2 - 60*A^2*a^2*b^4*c*f^2 + 60*A^2*a^4*b^2*c*f^2 + 80*A^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^6*d^2 - A^2*b^6*d^2 + 15*A^2*a^2*b^4*d^2 - 15*A^2*a^4*b^2*d^2))/f^2)*((((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) - 4*A^2*a^6*c*f^2 + 4*A^2*b^6*c*f^2 - 24*A^2*a*b^5*d*f^2 - 24*A^2*a^5*b*d*f^2 - 60*A^2*a^2*b^4*c*f^2 + 60*A^2*a^4*b^2*c*f^2 + 80*A^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i)/((16*(6*A^3*a^4*b^5*d^2 - A^3*b^9*d^2 + 8*A^3*a^6*b^3*d^2 + 3*A^3*a^8*b*d^2))/f^3 + (((8*(4*A*a^3*d^3*f^2 - 12*A*a*b^2*d^3*f^2 + 4*A*b^3*c*d^2*f^2 - 12*A*a^2*b*c*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) - 4*A^2*a^6*c*f^2 + 4*A^2*b^6*c*f^2 - 24*A^2*a*b^5*d*f^2 - 24*A^2*a^5*b*d*f^2 - 60*A^2*a^2*b^4*c*f^2 + 60*A^2*a^4*b^2*c*f^2 + 80*A^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 160*A^2 *a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) - 4*A^2*a^6*c*f^2 + 4*A^2*b^6*c*f^2 - 24*A^2*a*b^5*d*f^2 - 24*A^2*a^5*b*d*f^2 - 60*A^2*a^2*b^4*c*f^2 + 60*A^2*a^4*b^2*c*f^2 + 80*A^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^6*d^2 - A^2*b^6*d^2 + 15*A^2*a^2*b^4*d^2 - 15*A^2*a^4*b^2*d^2))/f^2)*((((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) - 4*A^2*a^6*c*f^2 + 4*A^2*b^6*c*f^2 - 24*A^2*a*b^5*d*f^2 - 24*A^2*a^5*b*d*f^2 - 60*A^2*a^2*b^4*c*f^2 + 60*A^2*a^4*b^2*c*f^2 + 80*A^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (((8*(4*A*a^3*d^3*f^2 - 12*A*a*b^2*d^3*f^2 + 4*A*b^3*c*d^2*f^2 - 12*A*a^2*b*c*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) - 4*A^2*a^6*c*f^2 + 4*A^2*b^6*c*f^2 - 24*A^2*a*b^5*d*f^2 - 24*A^2*a^5*b*d*f^2 - 60*A^2*a^2*b^4*c*f^2 + 60*A^2*a^4*b^2*c*f^2 + 80*A^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) - 4*A^2*a^6*c*f^2 + 4*A^2*b^6*c*f^2 - 24*A^2*a*b^5*d*f^2 - 24*A^2*a^5*b*d*f^2 - 60*A^2*a^2*b^4*c*f^2 + 60*A^2*a^4*b^2*c*f^2 + 80*A^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^6*d^2 - A^2*b^6*d^2 + 15*A^2*a^2*b^4*d^2 - 15*A^2*a^4*b^2*d^2))/f^2)*((((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) - 4*A^2*a^6*c*f^2 + 4*A^2*b^6*c*f^2 - 24*A^2*a*b^5*d*f^2 - 24*A^2*a^5*b*d*f^2 - 60*A^2*a^2*b^4*c*f^2 + 60*A^2*a^4*b^2*c*f^2 + 80*A^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)))*((((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) - 4*A^2*a^6*c*f^2 + 4*A^2*b^6*c*f^2 - 24*A^2*a*b^5*d*f^2 - 24*A^2*a^5*b*d*f^2 - 60*A^2*a^2*b^4*c*f^2 + 60*A^2*a^4*b^2*c*f^2 + 80*A^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*2i + atan(((((8*(4*A*a^3*d^3*f^2 - 12*A*a*b^2*d^3*f^2 + 4*A*b^3*c*d^2*f^2 - 12*A*a^2*b*c*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 160*A^2*a^3*b

^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*
b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^
(1/2) + 4*A^2*a^6*c*f^2 - 4*A^2*b^6*c*f^2 + 24*A^2*a*b^5*d*f^2 + 24*A^2*a^5
*b*d*f^2 + 60*A^2*a^2*b^4*c*f^2 - 60*A^2*a^4*b^2*c*f^2 - 80*A^2*a^3*b^3*d*f
^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2
 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^
2*a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A
^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*
A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) + 4*A^2*a^6*c*f^2 - 4*A^2*b^6*c*f^2 +
24*A^2*a*b^5*d*f^2 + 24*A^2*a^5*b*d*f^2 + 60*A^2*a^2*b^4*c*f^2 - 60*A^2*a^4
*b^2*c*f^2 - 80*A^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c
 + d*tan(e + f*x))^(1/2)*(A^2*a^6*d^2 - A^2*b^6*d^2 + 15*A^2*a^2*b^4*d^2 -
15*A^2*a^4*b^2*d^2))/f^2)*(-(((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a
*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c
*f^2 - 160*A^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A
^4*b^12 + 6*A^4*a^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4
 + 6*A^4*a^10*b^2))^(1/2) + 4*A^2*a^6*c*f^2 - 4*A^2*b^6*c*f^2 + 24*A^2*a*b^
5*d*f^2 + 24*A^2*a^5*b*d*f^2 + 60*A^2*a^2*b^4*c*f^2 - 60*A^2*a^4*b^2*c*f^2
- 80*A^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i - (((8*(4*A*a^3*
d^3*f^2 - 12*A*a*b^2*d^3*f^2 + 4*A*b^3*c*d^2*f^2 - 12*A*a^2*b*c*d^2*f^2))/f
^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c
*f^2 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 12
0*A^2*a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4
)*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 +
 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) + 4*A^2*a^6*c*f^2 - 4*A^2*b^6*c*f^
2 + 24*A^2*a*b^5*d*f^2 + 24*A^2*a^5*b*d*f^2 + 60*A^2*a^2*b^4*c*f^2 - 60*A^2
*a^4*b^2*c*f^2 - 80*A^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(
((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f
^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)
^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 + 15
*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) + 4
*A^2*a^6*c*f^2 - 4*A^2*b^6*c*f^2 + 24*A^2*a*b^5*d*f^2 + 24*A^2*a^5*b*d*f^2
+ 60*A^2*a^2*b^4*c*f^2 - 60*A^2*a^4*b^2*c*f^2 - 80*A^2*a^3*b^3*d*f^2)/(16*(
c^2*f^4 + d^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^6*d^2 -
A^2*b^6*d^2 + 15*A^2*a^2*b^4*d^2 - 15*A^2*a^4*b^2*d^2))/f^2)*(-(((8*A^2*a^6
*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^
2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)^2/4 - (16*
c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 + 15*A^4*a^4*b^
8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) + 4*A^2*a^6*c*
f^2 - 4*A^2*b^6*c*f^2 + 24*A^2*a*b^5*d*f^2 + 24*A^2*a^5*b*d*f^2 + 60*A^2*a^
2*b^4*c*f^2 - 60*A^2*a^4*b^2*c*f^2 - 80*A^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d
^2*f^4)))^(1/2)*1i)/((16*(6*A^3*a^4*b^5*d^2 - A^3*b^9*d^2 + 8*A^3*a^6*b^3*d
^2 + 3*A^3*a^8*b*d^2))/f^3 + (((8*(4*A*a^3*d^3*f^2 - 12*A*a*b^2*d^3*f^2 + 4
*A*b^3*c*d^2*f^2 - 12*A*a^2*b*c*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x
))^(1/2)*(-(((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f^2 + 48*A

^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 160*A^2*a^
3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 + 6*A^4*a
^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2
))^(1/2) + 4*A^2*a^6*c*f^2 - 4*A^2*b^6*c*f^2 + 24*A^2*a*b^5*d*f^2 + 24*A^2*
a^5*b*d*f^2 + 60*A^2*a^2*b^4*c*f^2 - 60*A^2*a^4*b^2*c*f^2 - 80*A^2*a^3*b^3*
d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*
f^2 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 120
*A^2*a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)
*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 +
15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) + 4*A^2*a^6*c*f^2 - 4*A^2*b^6*c*f^2
 + 24*A^2*a*b^5*d*f^2 + 24*A^2*a^5*b*d*f^2 + 60*A^2*a^2*b^4*c*f^2 - 60*A^2*
a^4*b^2*c*f^2 - 80*A^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16
*(c + d*tan(e + f*x))^(1/2)*(A^2*a^6*d^2 - A^2*b^6*d^2 + 15*A^2*a^2*b^4*d^2
 - 15*A^2*a^4*b^2*d^2))/f^2)*(-(((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^
2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^
2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^12
+ A^4*b^12 + 6*A^4*a^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*
b^4 + 6*A^4*a^10*b^2))^(1/2) + 4*A^2*a^6*c*f^2 - 4*A^2*b^6*c*f^2 + 24*A^2*a
*b^5*d*f^2 + 24*A^2*a^5*b*d*f^2 + 60*A^2*a^2*b^4*c*f^2 - 60*A^2*a^4*b^2*c*f
^2 - 80*A^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (((8*(4*A*a^3*
d^3*f^2 - 12*A*a*b^2*d^3*f^2 + 4*A*b^3*c*d^2*f^2 - 12*A*a^2*b*c*d^2*f^2))/f
^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c
*f^2 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 12
0*A^2*a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4
)*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 +
 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) + 4*A^2*a^6*c*f^2 - 4*A^2*b^6*c*f^
2 + 24*A^2*a*b^5*d*f^2 + 24*A^2*a^5*b*d*f^2 + 60*A^2*a^2*b^4*c*f^2 - 60*A^2
*a^4*b^2*c*f^2 - 80*A^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(
((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f
^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)
^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 + 15
*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) + 4
*A^2*a^6*c*f^2 - 4*A^2*b^6*c*f^2 + 24*A^2*a*b^5*d*f^2 + 24*A^2*a^5*b*d*f^2
+ 60*A^2*a^2*b^4*c*f^2 - 60*A^2*a^4*b^2*c*f^2 - 80*A^2*a^3*b^3*d*f^2)/(16*(
c^2*f^4 + d^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^6*d^2 -
A^2*b^6*d^2 + 15*A^2*a^2*b^4*d^2 - 15*A^2*a^4*b^2*d^2))/f^2)*(-(((8*A^2*a^6
*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^
2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)^2/4 - (16*
c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 + 15*A^4*a^4*b^
8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) + 4*A^2*a^6*c*
f^2 - 4*A^2*b^6*c*f^2 + 24*A^2*a*b^5*d*f^2 + 24*A^2*a^5*b*d*f^2 + 60*A^2*a^
2*b^4*c*f^2 - 60*A^2*a^4*b^2*c*f^2 - 80*A^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d
^2*f^4)))^(1/2)))*(-(((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f
^2 + 48*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 1
60*A^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12

+ 6*A^4*a^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4
*a^10*b^2))^(1/2) + 4*A^2*a^6*c*f^2 - 4*A^2*b^6*c*f^2 + 24*A^2*a*b^5*d*f^2
+ 24*A^2*a^5*b*d*f^2 + 60*A^2*a^2*b^4*c*f^2 - 60*A^2*a^4*b^2*c*f^2 - 80*A^2
*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*2i - (c + d*tan(e + f*x))^(
1/2)*(2*c*((8*B*b^3*c - 6*B*a*b^2*d)/(d^3*f) - (4*B*b^3*c)/(d^3*f)) - (12*B
*b^3*c^2 + 6*B*a^2*b*d^2 - 18*B*a*b^2*c*d)/(d^3*f) + (2*B*b^3*(d^5*f + c^2*
d^3*f))/(d^6*f^2)) + atan(((((8*(4*B*b^3*d^3*f^2 - 12*B*a^2*b*d^3*f^2 - 4*B
*a^3*c*d^2*f^2 + 12*B*a*b^2*c*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))
^(1/2)*(-(((8*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f^2 + 48*B^2*a*b^5*d*f^2 + 48*B^2
*a^5*b*d*f^2 + 120*B^2*a^2*b^4*c*f^2 - 120*B^2*a^4*b^2*c*f^2 - 160*B^2*a^3*
b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^12 + B^4*b^12 + 6*B^4*a^2
*b^10 + 15*B^4*a^4*b^8 + 20*B^4*a^6*b^6 + 15*B^4*a^8*b^4 + 6*B^4*a^10*b^2))
^(1/2) - 4*B^2*a^6*c*f^2 + 4*B^2*b^6*c*f^2 - 24*B^2*a*b^5*d*f^2 - 24*B^2*a^
5*b*d*f^2 - 60*B^2*a^2*b^4*c*f^2 + 60*B^2*a^4*b^2*c*f^2 + 80*B^2*a^3*b^3*d*
f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f^
2 + 48*B^2*a*b^5*d*f^2 + 48*B^2*a^5*b*d*f^2 + 120*B^2*a^2*b^4*c*f^2 - 120*B
^2*a^4*b^2*c*f^2 - 160*B^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(
B^4*a^12 + B^4*b^12 + 6*B^4*a^2*b^10 + 15*B^4*a^4*b^8 + 20*B^4*a^6*b^6 + 15
*B^4*a^8*b^4 + 6*B^4*a^10*b^2))^(1/2) - 4*B^2*a^6*c*f^2 + 4*B^2*b^6*c*f^2 -
24*B^2*a*b^5*d*f^2 - 24*B^2*a^5*b*d*f^2 - 60*B^2*a^2*b^4*c*f^2 + 60*B^2*a^
4*b^2*c*f^2 + 80*B^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (16*(
c + d*tan(e + f*x))^(1/2)*(B^2*a^6*d^2 - B^2*b^6*d^2 + 15*B^2*a^2*b^4*d^2 -
15*B^2*a^4*b^2*d^2))/f^2)*(-(((8*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f^2 + 48*B^2*
a*b^5*d*f^2 + 48*B^2*a^5*b*d*f^2 + 120*B^2*a^2*b^4*c*f^2 - 120*B^2*a^4*b^2*
c*f^2 - 160*B^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^12 +
B^4*b^12 + 6*B^4*a^2*b^10 + 15*B^4*a^4*b^8 + 20*B^4*a^6*b^6 + 15*B^4*a^8*b^
4 + 6*B^4*a^10*b^2))^(1/2) - 4*B^2*a^6*c*f^2 + 4*B^2*b^6*c*f^2 - 24*B^2*a*b
^5*d*f^2 - 24*B^2*a^5*b*d*f^2 - 60*B^2*a^2*b^4*c*f^2 + 60*B^2*a^4*b^2*c*f^2
+ 80*B^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i - (((8*(4*B*b^3
*d^3*f^2 - 12*B*a^2*b*d^3*f^2 - 4*B*a^3*c*d^2*f^2 + 12*B*a*b^2*c*d^2*f^2))/
f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^6*c*f^2 - 8*B^2*b^6*
c*f^2 + 48*B^2*a*b^5*d*f^2 + 48*B^2*a^5*b*d*f^2 + 120*B^2*a^2*b^4*c*f^2 - 1
20*B^2*a^4*b^2*c*f^2 - 160*B^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^
4)*(B^4*a^12 + B^4*b^12 + 6*B^4*a^2*b^10 + 15*B^4*a^4*b^8 + 20*B^4*a^6*b^6
+ 15*B^4*a^8*b^4 + 6*B^4*a^10*b^2))^(1/2) - 4*B^2*a^6*c*f^2 + 4*B^2*b^6*c*f
^2 - 24*B^2*a*b^5*d*f^2 - 24*B^2*a^5*b*d*f^2 - 60*B^2*a^2*b^4*c*f^2 + 60*B^
2*a^4*b^2*c*f^2 + 80*B^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-
(((8*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f^2 + 48*B^2*a*b^5*d*f^2 + 48*B^2*a^5*b*d*
f^2 + 120*B^2*a^2*b^4*c*f^2 - 120*B^2*a^4*b^2*c*f^2 - 160*B^2*a^3*b^3*d*f^2
)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^12 + B^4*b^12 + 6*B^4*a^2*b^10 + 1
5*B^4*a^4*b^8 + 20*B^4*a^6*b^6 + 15*B^4*a^8*b^4 + 6*B^4*a^10*b^2))^(1/2) -
4*B^2*a^6*c*f^2 + 4*B^2*b^6*c*f^2 - 24*B^2*a*b^5*d*f^2 - 24*B^2*a^5*b*d*f^2
- 60*B^2*a^2*b^4*c*f^2 + 60*B^2*a^4*b^2*c*f^2 + 80*B^2*a^3*b^3*d*f^2)/(16*
(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(B^2*a^6*d^2 -
B^2*b^6*d^2 + 15*B^2*a^2*b^4*d^2 - 15*B^2*a^4*b^2*d^2))/f^2)*(-(((8*B^2*a^

6*c*f^2 - 8*B^2*b^6*c*f^2 + 48*B^2*a*b^5*d*f^2 + 48*B^2*a^5*b*d*f^2 + 120*B
^2*a^2*b^4*c*f^2 - 120*B^2*a^4*b^2*c*f^2 - 160*B^2*a^3*b^3*d*f^2)^2/4 - (16
*c^2*f^4 + 16*d^2*f^4)*(B^4*a^12 + B^4*b^12 + 6*B^4*a^2*b^10 + 15*B^4*a^4*b
^8 + 20*B^4*a^6*b^6 + 15*B^4*a^8*b^4 + 6*B^4*a^10*b^2))^(1/2) - 4*B^2*a^6*c
*f^2 + 4*B^2*b^6*c*f^2 - 24*B^2*a*b^5*d*f^2 - 24*B^2*a^5*b*d*f^2 - 60*B^2*a
^2*b^4*c*f^2 + 60*B^2*a^4*b^2*c*f^2 + 80*B^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 +
d^2*f^4)))^(1/2)*1i)/((((8*(4*B*b^3*d^3*f^2 - 12*B*a^2*b*d^3*f^2 - 4*B*a^3*
c*d^2*f^2 + 12*B*a*b^2*c*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2
)*(-(((8*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f^2 + 48*B^2*a*b^5*d*f^2 + 48*B^2*a^5*
b*d*f^2 + 120*B^2*a^2*b^4*c*f^2 - 120*B^2*a^4*b^2*c*f^2 - 160*B^2*a^3*b^3*d
*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^12 + B^4*b^12 + 6*B^4*a^2*b^10
 + 15*B^4*a^4*b^8 + 20*B^4*a^6*b^6 + 15*B^4*a^8*b^4 + 6*B^4*a^10*b^2))^(1/2
) - 4*B^2*a^6*c*f^2 + 4*B^2*b^6*c*f^2 - 24*B^2*a*b^5*d*f^2 - 24*B^2*a^5*b*d
*f^2 - 60*B^2*a^2*b^4*c*f^2 + 60*B^2*a^4*b^2*c*f^2 + 80*B^2*a^3*b^3*d*f^2)/
(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f^2 + 4
8*B^2*a*b^5*d*f^2 + 48*B^2*a^5*b*d*f^2 + 120*B^2*a^2*b^4*c*f^2 - 120*B^2*a^
4*b^2*c*f^2 - 160*B^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a
^12 + B^4*b^12 + 6*B^4*a^2*b^10 + 15*B^4*a^4*b^8 + 20*B^4*a^6*b^6 + 15*B^4*
a^8*b^4 + 6*B^4*a^10*b^2))^(1/2) - 4*B^2*a^6*c*f^2 + 4*B^2*b^6*c*f^2 - 24*B
^2*a*b^5*d*f^2 - 24*B^2*a^5*b*d*f^2 - 60*B^2*a^2*b^4*c*f^2 + 60*B^2*a^4*b^2
*c*f^2 + 80*B^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (16*(c + d
*tan(e + f*x))^(1/2)*(B^2*a^6*d^2 - B^2*b^6*d^2 + 15*B^2*a^2*b^4*d^2 - 15*B
^2*a^4*b^2*d^2))/f^2)*(-(((8*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f^2 + 48*B^2*a*b^5
*d*f^2 + 48*B^2*a^5*b*d*f^2 + 120*B^2*a^2*b^4*c*f^2 - 120*B^2*a^4*b^2*c*f^2
 - 160*B^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^12 + B^4*b
^12 + 6*B^4*a^2*b^10 + 15*B^4*a^4*b^8 + 20*B^4*a^6*b^6 + 15*B^4*a^8*b^4 + 6
*B^4*a^10*b^2))^(1/2) - 4*B^2*a^6*c*f^2 + 4*B^2*b^6*c*f^2 - 24*B^2*a*b^5*d*
f^2 - 24*B^2*a^5*b*d*f^2 - 60*B^2*a^2*b^4*c*f^2 + 60*B^2*a^4*b^2*c*f^2 + 80
*B^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(8*B^3*a^3*b^6*d^
2 - B^3*a^9*d^2 + 6*B^3*a^5*b^4*d^2 + 3*B^3*a*b^8*d^2))/f^3 + (((8*(4*B*b^3
*d^3*f^2 - 12*B*a^2*b*d^3*f^2 - 4*B*a^3*c*d^2*f^2 + 12*B*a*b^2*c*d^2*f^2))/
f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^6*c*f^2 - 8*B^2*b^6*
c*f^2 + 48*B^2*a*b^5*d*f^2 + 48*B^2*a^5*b*d*f^2 + 120*B^2*a^2*b^4*c*f^2 - 1
20*B^2*a^4*b^2*c*f^2 - 160*B^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^
4)*(B^4*a^12 + B^4*b^12 + 6*B^4*a^2*b^10 + 15*B^4*a^4*b^8 + 20*B^4*a^6*b^6
 + 15*B^4*a^8*b^4 + 6*B^4*a^10*b^2))^(1/2) - 4*B^2*a^6*c*f^2 + 4*B^2*b^6*c*f
^2 - 24*B^2*a*b^5*d*f^2 - 24*B^2*a^5*b*d*f^2 - 60*B^2*a^2*b^4*c*f^2 + 60*B^
2*a^4*b^2*c*f^2 + 80*B^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-
(((8*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f^2 + 48*B^2*a*b^5*d*f^2 + 48*B^2*a^5*b*d*
f^2 + 120*B^2*a^2*b^4*c*f^2 - 120*B^2*a^4*b^2*c*f^2 - 160*B^2*a^3*b^3*d*f^2
)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^12 + B^4*b^12 + 6*B^4*a^2*b^10 + 1
5*B^4*a^4*b^8 + 20*B^4*a^6*b^6 + 15*B^4*a^8*b^4 + 6*B^4*a^10*b^2))^(1/2) -
4*B^2*a^6*c*f^2 + 4*B^2*b^6*c*f^2 - 24*B^2*a*b^5*d*f^2 - 24*B^2*a^5*b*d*f^2
 - 60*B^2*a^2*b^4*c*f^2 + 60*B^2*a^4*b^2*c*f^2 + 80*B^2*a^3*b^3*d*f^2)/(16*
(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(B^2*a^6*d^2 -

B^2*b^6*d^2 + 15*B^2*a^2*b^4*d^2 - 15*B^2*a^4*b^2*d^2))/f^2)*(-(((8*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f^2 + 48*B^2*a*b^5*d*f^2 + 48*B^2*a^5*b*d*f^2 + 120*B^2*a^2*b^4*c*f^2 - 120*B^2*a^4*b^2*c*f^2 - 160*B^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^12 + B^4*b^12 + 6*B^4*a^2*b^10 + 15*B^4*a^4*b^8 + 20*B^4*a^6*b^6 + 15*B^4*a^8*b^4 + 6*B^4*a^10*b^2))^(1/2) - 4*B^2*a^6*c*f^2 + 4*B^2*b^6*c*f^2 - 24*B^2*a*b^5*d*f^2 - 24*B^2*a^5*b*d*f^2 - 60*B^2*a^2*b^4*c*f^2 + 60*B^2*a^4*b^2*c*f^2 + 80*B^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)))*(-(((8*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f^2 + 48*B^2*a*b^5*d*f^2 + 48*B^2*a^5*b*d*f^2 + 120*B^2*a^2*b^4*c*f^2 - 120*B^2*a^4*b^2*c*f^2 - 160*B^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^12 + B^4*b^12 + 6*B^4*a^2*b^10 + 15*B^4*a^4*b^8 + 20*B^4*a^6*b^6 + 15*B^4*a^8*b^4 + 6*B^4*a^10*b^2))^(1/2) - 4*B^2*a^6*c*f^2 + 4*B^2*b^6*c*f^2 - 24*B^2*a*b^5*d*f^2 - 24*B^2*a^5*b*d*f^2 - 60*B^2*a^2*b^4*c*f^2 + 60*B^2*a^4*b^2*c*f^2 + 80*B^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*2i + atan(((((8*(4*B*b^3*d^3*f^2 - 12*B*a^2*b*d^3*f^2 - 4*B*a^3*c*d^2*f^2 + 12*B*a*b^2*c*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f^2 + 48*B^2*a*b^5*d*f^2 + 48*B^2*a^5*b*d*f^2 + 120*B^2*a^2*b^4*c*f^2 - 120*B^2*a^4*b^2*c*f^2 - 160*B^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^12 + B^4*b^12 + 6*B^4*a^2*b^10 + 15*B^4*a^4*b^8 + 20*B^4*a^6*b^6 + 15*B^4*a^8*b^4 + 6*B^4*a^10*b^2))^(1/2) + 4*B^2*a^6*c*f^2 - 4*B^2*b^6*c*f^2 + 24*B^2*a*b^5*d*f^2 + 24*B^2*a^5*b*d*f^2 + 60*B^2*a^2*b^4*c*f^2 - 60*B^2*a^4*b^2*c*f^2 - 80*B^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((((8*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f^2 + 48*B^2*a*b^5*d*f^2 + 48*B^2*a^5*b*d*f^2 + 120*B^2*a^2*b^4*c*f^2 - 120*B^2*a^4*b^2*c*f^2 - 160*B^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^12 + B^4*b^12 + 6*B^4*a^2*b^10 + 15*B^4*a^4*b^8 + 20*B^4*a^6*b^6 + 15*B^4*a^8*b^4 + 6*B^4*a^10*b^2))^(1/2) + 4*B^2*a^6*c*f^2 - 4*B^2*b^6*c*f^2 + 24*B^2*a*b^5*d*f^2 + 24*B^2*a^5*b*d*f^2 + 60*B^2*a^2*b^4*c*f^2 - 60*B^2*a^4*b^2*c*f^2 - 80*B^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(B^2*a^6*d^2 - B^2*b^6*d^2 + 15*B^2*a^2*b^4*d^2 - 15*B^2*a^4*b^2*d^2))/f^2)*((((8*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f^2 + 48*B^2*a*b^5*d*f^2 + 48*B^2*a^5*b*d*f^2 + 120*B^2*a^2*b^4*c*f^2 - 120*B^2*a^4*b^2*c*f^2 - 160*B^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^12 + B^4*b^12 + 6*B^4*a^2*b^10 + 15*B^4*a^4*b^8 + 20*B^4*a^6*b^6 + 15*B^4*a^8*b^4 + 6*B^4*a^10*b^2))^(1/2) + 4*B^2*a^6*c*f^2 - 4*B^2*b^6*c*f^2 + 24*B^2*a*b^5*d*f^2 + 24*B^2*a^5*b*d*f^2 + 60*B^2*a^2*b^4*c*f^2 - 60*B^2*a^4*b^2*c*f^2 - 80*B^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i - (((8*(4*B*b^3*d^3*f^2 - 12*B*a^2*b*d^3*f^2 - 4*B*a^3*c*d^2*f^2 + 12*B*a*b^2*c*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f^2 + 48*B^2*a*b^5*d*f^2 + 48*B^2*a^5*b*d*f^2 + 120*B^2*a^2*b^4*c*f^2 - 120*B^2*a^4*b^2*c*f^2 - 160*B^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^12 + B^4*b^12 + 6*B^4*a^2*b^10 + 15*B^4*a^4*b^8 + 20*B^4*a^6*b^6 + 15*B^4*a^8*b^4 + 6*B^4*a^10*b^2))^(1/2) + 4*B^2*a^6*c*f^2 - 4*B^2*b^6*c*f^2 + 24*B^2*a*b^5*d*f^2 + 24*B^2*a^5*b*d*f^2 + 60*B^2*a^2*b^4*c*f^2 - 60*B^2*a^4*b^2*c*f^2 - 80*B^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((((8*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f^2 + 48*B^2*a

*b^5*d*f^2 + 48*B^2*a^5*b*d*f^2 + 120*B^2*a^2*b^4*c*f^2 - 120*B^2*a^4*b^2*c
*f^2 - 160*B^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^12 + B
^4*b^12 + 6*B^4*a^2*b^10 + 15*B^4*a^4*b^8 + 20*B^4*a^6*b^6 + 15*B^4*a^8*b^4
 + 6*B^4*a^10*b^2))^(1/2) + 4*B^2*a^6*c*f^2 - 4*B^2*b^6*c*f^2 + 24*B^2*a*b^
5*d*f^2 + 24*B^2*a^5*b*d*f^2 + 60*B^2*a^2*b^4*c*f^2 - 60*B^2*a^4*b^2*c*f^2
- 80*B^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e
+ f*x))^(1/2)*(B^2*a^6*d^2 - B^2*b^6*d^2 + 15*B^2*a^2*b^4*d^2 - 15*B^2*a^4*
b^2*d^2))/f^2)*((((8*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f^2 + 48*B^2*a*b^5*d*f^2 +
 48*B^2*a^5*b*d*f^2 + 120*B^2*a^2*b^4*c*f^2 - 120*B^2*a^4*b^2*c*f^2 - 160*B
^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^12 + B^4*b^12 + 6*
B^4*a^2*b^10 + 15*B^4*a^4*b^8 + 20*B^4*a^6*b^6 + 15*B^4*a^8*b^4 + 6*B^4*a^1
0*b^2))^(1/2) + 4*B^2*a^6*c*f^2 - 4*B^2*b^6*c*f^2 + 24*B^2*a*b^5*d*f^2 + 24
*B^2*a^5*b*d*f^2 + 60*B^2*a^2*b^4*c*f^2 - 60*B^2*a^4*b^2*c*f^2 - 80*B^2*a^3
*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i)/((((8*(4*B*b^3*d^3*f^2 - 12
*B*a^2*b*d^3*f^2 - 4*B*a^3*c*d^2*f^2 + 12*B*a*b^2*c*d^2*f^2))/f^3 - 64*c*d^
2*(c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f^2 + 48*B^2
*a*b^5*d*f^2 + 48*B^2*a^5*b*d*f^2 + 120*B^2*a^2*b^4*c*f^2 - 120*B^2*a^4*b^2
*c*f^2 - 160*B^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^12 +
 B^4*b^12 + 6*B^4*a^2*b^10 + 15*B^4*a^4*b^8 + 20*B^4*a^6*b^6 + 15*B^4*a^8*b
^4 + 6*B^4*a^10*b^2))^(1/2) + 4*B^2*a^6*c*f^2 - 4*B^2*b^6*c*f^2 + 24*B^2*a*
b^5*d*f^2 + 24*B^2*a^5*b*d*f^2 + 60*B^2*a^2*b^4*c*f^2 - 60*B^2*a^4*b^2*c*f^
2 - 80*B^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((((8*B^2*a^6*c*
f^2 - 8*B^2*b^6*c*f^2 + 48*B^2*a*b^5*d*f^2 + 48*B^2*a^5*b*d*f^2 + 120*B^2*a
^2*b^4*c*f^2 - 120*B^2*a^4*b^2*c*f^2 - 160*B^2*a^3*b^3*d*f^2)^2/4 - (16*c^2
*f^4 + 16*d^2*f^4)*(B^4*a^12 + B^4*b^12 + 6*B^4*a^2*b^10 + 15*B^4*a^4*b^8 +
 20*B^4*a^6*b^6 + 15*B^4*a^8*b^4 + 6*B^4*a^10*b^2))^(1/2) + 4*B^2*a^6*c*f^2
 - 4*B^2*b^6*c*f^2 + 24*B^2*a*b^5*d*f^2 + 24*B^2*a^5*b*d*f^2 + 60*B^2*a^2*b
^4*c*f^2 - 60*B^2*a^4*b^2*c*f^2 - 80*B^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*
f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(B^2*a^6*d^2 - B^2*b^6*d^2 +
15*B^2*a^2*b^4*d^2 - 15*B^2*a^4*b^2*d^2))/f^2)*((((8*B^2*a^6*c*f^2 - 8*B^2*
b^6*c*f^2 + 48*B^2*a*b^5*d*f^2 + 48*B^2*a^5*b*d*f^2 + 120*B^2*a^2*b^4*c*f^2
 - 120*B^2*a^4*b^2*c*f^2 - 160*B^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^
2*f^4)*(B^4*a^12 + B^4*b^12 + 6*B^4*a^2*b^10 + 15*B^4*a^4*b^8 + 20*B^4*a^6*
b^6 + 15*B^4*a^8*b^4 + 6*B^4*a^10*b^2))^(1/2) + 4*B^2*a^6*c*f^2 - 4*B^2*b^6
*c*f^2 + 24*B^2*a*b^5*d*f^2 + 24*B^2*a^5*b*d*f^2 + 60*B^2*a^2*b^4*c*f^2 - 6
0*B^2*a^4*b^2*c*f^2 - 80*B^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)
 - (16*(8*B^3*a^3*b^6*d^2 - B^3*a^9*d^2 + 6*B^3*a^5*b^4*d^2 + 3*B^3*a*b^8*d
^2))/f^3 + (((8*(4*B*b^3*d^3*f^2 - 12*B*a^2*b*d^3*f^2 - 4*B*a^3*c*d^2*f^2 +
 12*B*a*b^2*c*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*B^2
*a^6*c*f^2 - 8*B^2*b^6*c*f^2 + 48*B^2*a*b^5*d*f^2 + 48*B^2*a^5*b*d*f^2 + 12
0*B^2*a^2*b^4*c*f^2 - 120*B^2*a^4*b^2*c*f^2 - 160*B^2*a^3*b^3*d*f^2)^2/4 -
(16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^12 + B^4*b^12 + 6*B^4*a^2*b^10 + 15*B^4*a^
4*b^8 + 20*B^4*a^6*b^6 + 15*B^4*a^8*b^4 + 6*B^4*a^10*b^2))^(1/2) + 4*B^2*a^
6*c*f^2 - 4*B^2*b^6*c*f^2 + 24*B^2*a*b^5*d*f^2 + 24*B^2*a^5*b*d*f^2 + 60*B^
2*a^2*b^4*c*f^2 - 60*B^2*a^4*b^2*c*f^2 - 80*B^2*a^3*b^3*d*f^2)/(16*(c^2*f^4

+ d^2*f^4)))^(1/2))*((((8*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f^2 + 48*B^2*a*b^5*d*f^2 + 48*B^2*a^5*b*d*f^2 + 120*B^2*a^2*b^4*c*f^2 - 120*B^2*a^4*b^2*c*f^2 - 160*B^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^12 + B^4*b^12 + 6*B^4*a^2*b^10 + 15*B^4*a^4*b^8 + 20*B^4*a^6*b^6 + 15*B^4*a^8*b^4 + 6*B^4*a^10*b^2))^(1/2) + 4*B^2*a^6*c*f^2 - 4*B^2*b^6*c*f^2 + 24*B^2*a*b^5*d*f^2 + 24*B^2*a^5*b*d*f^2 + 60*B^2*a^2*b^4*c*f^2 - 60*B^2*a^4*b^2*c*f^2 - 80*B^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(B^2*a^6*d^2 - B^2*b^6*d^2 + 15*B^2*a^2*b^4*d^2 - 15*B^2*a^4*b^2*d^2))/f^2)*((((8*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f^2 + 48*B^2*a*b^5*d*f^2 + 48*B^2*a^5*b*d*f^2 + 120*B^2*a^2*b^4*c*f^2 - 120*B^2*a^4*b^2*c*f^2 - 160*B^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^12 + B^4*b^12 + 6*B^4*a^2*b^10 + 15*B^4*a^4*b^8 + 20*B^4*a^6*b^6 + 15*B^4*a^8*b^4 + 6*B^4*a^10*b^2))^(1/2) + 4*B^2*a^6*c*f^2 - 4*B^2*b^6*c*f^2 + 24*B^2*a*b^5*d*f^2 + 24*B^2*a^5*b*d*f^2 + 60*B^2*a^2*b^4*c*f^2 - 60*B^2*a^4*b^2*c*f^2 - 80*B^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)))*((((8*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f^2 + 48*B^2*a*b^5*d*f^2 + 48*B^2*a^5*b*d*f^2 + 120*B^2*a^2*b^4*c*f^2 - 120*B^2*a^4*b^2*c*f^2 - 160*B^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^12 + B^4*b^12 + 6*B^4*a^2*b^10 + 15*B^4*a^4*b^8 + 20*B^4*a^6*b^6 + 15*B^4*a^8*b^4 + 6*B^4*a^10*b^2))^(1/2) + 4*B^2*a^6*c*f^2 - 4*B^2*b^6*c*f^2 + 24*B^2*a*b^5*d*f^2 + 24*B^2*a^5*b*d*f^2 + 60*B^2*a^2*b^4*c*f^2 - 60*B^2*a^4*b^2*c*f^2 - 80*B^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*2i - (c + d*tan(e + f*x))^(3/2)*((2*c*((10*C*b^3*c - 6*C*a*b^2*d)/(d^4*f) - (4*C*b^3*c)/(d^4*f)))/3 - (20*C*b^3*c^2 + 6*C*a^2*b*d^2 - 24*C*a*b^2*c*d)/(3*d^4*f) + (2*C*b^3*(d^6*f + c^2*d^4*f))/(3*d^8*f^2)) - atan(((((8*(4*C*a^3*d^3*f^2 - 12*C*a*b^2*d^3*f^2 + 4*C*b^3*c*d^2*f^2 - 12*C*a^2*b*c*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^2 + 48*C^2*a^5*b*d*f^2 + 120*C^2*a^2*b^4*c*f^2 - 120*C^2*a^4*b^2*c*f^2 - 160*C^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^12 + C^4*b^12 + 6*C^4*a^2*b^10 + 15*C^4*a^4*b^8 + 20*C^4*a^6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*a^10*b^2))^(1/2) - 4*C^2*a^6*c*f^2 + 4*C^2*b^6*c*f^2 - 24*C^2*a*b^5*d*f^2 - 24*C^2*a^5*b*d*f^2 - 60*C^2*a^2*b^4*c*f^2 + 60*C^2*a^4*b^2*c*f^2 + 80*C^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((((8*C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^2 + 48*C^2*a^5*b*d*f^2 + 120*C^2*a^2*b^4*c*f^2 - 120*C^2*a^4*b^2*c*f^2 - 160*C^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^12 + C^4*b^12 + 6*C^4*a^2*b^10 + 15*C^4*a^4*b^8 + 20*C^4*a^6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*a^10*b^2))^(1/2) - 4*C^2*a^6*c*f^2 + 4*C^2*b^6*c*f^2 - 24*C^2*a*b^5*d*f^2 - 24*C^2*a^5*b*d*f^2 - 60*C^2*a^2*b^4*c*f^2 + 60*C^2*a^4*b^2*c*f^2 + 80*C^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(C^2*a^6*d^2 - C^2*b^6*d^2 + 15*C^2*a^2*b^4*d^2 - 15*C^2*a^4*b^2*d^2))/f^2)*((((8*C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^2 + 48*C^2*a^5*b*d*f^2 + 120*C^2*a^2*b^4*c*f^2 - 120*C^2*a^4*b^2*c*f^2 - 160*C^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^12 + C^4*b^12 + 6*C^4*a^2*b^10 + 15*C^4*a^4*b^8 + 20*C^4*a^6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*a^10*b^2))^(1/2) - 4*C^2*a^6*c*f^2 + 4*C^2*b^6*c*f^2 - 24*C^2*a*b^5*d*f^2 - 24*C^2*a^5*b*d*f^2 - 60*C^2*a^2*b^4*c

*f^2 + 60*C^2*a^4*b^2*c*f^2 + 80*C^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)
))^(1/2)*1i - (((8*(4*C*a^3*d^3*f^2 - 12*C*a*b^2*d^3*f^2 + 4*C*b^3*c*d^2*f^
2 - 12*C*a^2*b*c*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*
C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^2 + 48*C^2*a^5*b*d*f^2 +
120*C^2*a^2*b^4*c*f^2 - 120*C^2*a^4*b^2*c*f^2 - 160*C^2*a^3*b^3*d*f^2)^2/4
- (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^12 + C^4*b^12 + 6*C^4*a^2*b^10 + 15*C^4
*a^4*b^8 + 20*C^4*a^6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*a^10*b^2))^(1/2) - 4*C^2
*a^6*c*f^2 + 4*C^2*b^6*c*f^2 - 24*C^2*a*b^5*d*f^2 - 24*C^2*a^5*b*d*f^2 - 60
*C^2*a^2*b^4*c*f^2 + 60*C^2*a^4*b^2*c*f^2 + 80*C^2*a^3*b^3*d*f^2)/(16*(c^2*
f^4 + d^2*f^4)))^(1/2))*((((8*C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2 + 48*C^2*a*b^
5*d*f^2 + 48*C^2*a^5*b*d*f^2 + 120*C^2*a^2*b^4*c*f^2 - 120*C^2*a^4*b^2*c*f^
2 - 160*C^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^12 + C^4*
b^12 + 6*C^4*a^2*b^10 + 15*C^4*a^4*b^8 + 20*C^4*a^6*b^6 + 15*C^4*a^8*b^4 +
6*C^4*a^10*b^2))^(1/2) - 4*C^2*a^6*c*f^2 + 4*C^2*b^6*c*f^2 - 24*C^2*a*b^5*d
*f^2 - 24*C^2*a^5*b*d*f^2 - 60*C^2*a^2*b^4*c*f^2 + 60*C^2*a^4*b^2*c*f^2 + 8
0*C^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (16*(c + d*tan(e + f
x))^(1/2)(C^2*a^6*d^2 - C^2*b^6*d^2 + 15*C^2*a^2*b^4*d^2 - 15*C^2*a^4*b^2
d^2))/f^2)((((8*C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^2 + 48
*C^2*a^5*b*d*f^2 + 120*C^2*a^2*b^4*c*f^2 - 120*C^2*a^4*b^2*c*f^2 - 160*C^2*
a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^12 + C^4*b^12 + 6*C^4
*a^2*b^10 + 15*C^4*a^4*b^8 + 20*C^4*a^6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*a^10*b
^2))^(1/2) - 4*C^2*a^6*c*f^2 + 4*C^2*b^6*c*f^2 - 24*C^2*a*b^5*d*f^2 - 24*C^
2*a^5*b*d*f^2 - 60*C^2*a^2*b^4*c*f^2 + 60*C^2*a^4*b^2*c*f^2 + 80*C^2*a^3*b^
3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i)/((((8*(4*C*a^3*d^3*f^2 - 12*C*
a*b^2*d^3*f^2 + 4*C*b^3*c*d^2*f^2 - 12*C*a^2*b*c*d^2*f^2))/f^3 - 64*c*d^2*(
c + d*tan(e + f*x))^(1/2)*((((8*C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2 + 48*C^2*a*
b^5*d*f^2 + 48*C^2*a^5*b*d*f^2 + 120*C^2*a^2*b^4*c*f^2 - 120*C^2*a^4*b^2*c*
f^2 - 160*C^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^12 + C^
4*b^12 + 6*C^4*a^2*b^10 + 15*C^4*a^4*b^8 + 20*C^4*a^6*b^6 + 15*C^4*a^8*b^4
+ 6*C^4*a^10*b^2))^(1/2) - 4*C^2*a^6*c*f^2 + 4*C^2*b^6*c*f^2 - 24*C^2*a*b^5
*d*f^2 - 24*C^2*a^5*b*d*f^2 - 60*C^2*a^2*b^4*c*f^2 + 60*C^2*a^4*b^2*c*f^2 +
80*C^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((((8*C^2*a^6*c*f^2
- 8*C^2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^2 + 48*C^2*a^5*b*d*f^2 + 120*C^2*a^2*
b^4*c*f^2 - 120*C^2*a^4*b^2*c*f^2 - 160*C^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^
4 + 16*d^2*f^4)*(C^4*a^12 + C^4*b^12 + 6*C^4*a^2*b^10 + 15*C^4*a^4*b^8 + 20
*C^4*a^6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*a^10*b^2))^(1/2) - 4*C^2*a^6*c*f^2 +
4*C^2*b^6*c*f^2 - 24*C^2*a*b^5*d*f^2 - 24*C^2*a^5*b*d*f^2 - 60*C^2*a^2*b^4*
c*f^2 + 60*C^2*a^4*b^2*c*f^2 + 80*C^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4
)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(C^2*a^6*d^2 - C^2*b^6*d^2 + 15*
C^2*a^2*b^4*d^2 - 15*C^2*a^4*b^2*d^2))/f^2)*((((8*C^2*a^6*c*f^2 - 8*C^2*b^6
*c*f^2 + 48*C^2*a*b^5*d*f^2 + 48*C^2*a^5*b*d*f^2 + 120*C^2*a^2*b^4*c*f^2 -
120*C^2*a^4*b^2*c*f^2 - 160*C^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f
^4)*(C^4*a^12 + C^4*b^12 + 6*C^4*a^2*b^10 + 15*C^4*a^4*b^8 + 20*C^4*a^6*b^6
+ 15*C^4*a^8*b^4 + 6*C^4*a^10*b^2))^(1/2) - 4*C^2*a^6*c*f^2 + 4*C^2*b^6*c*
f^2 - 24*C^2*a*b^5*d*f^2 - 24*C^2*a^5*b*d*f^2 - 60*C^2*a^2*b^4*c*f^2 + 60*C

^2*a^4*b^2*c*f^2 + 80*C^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) +
(((8*(4*C*a^3*d^3*f^2 - 12*C*a*b^2*d^3*f^2 + 4*C*b^3*c*d^2*f^2 - 12*C*a^2*b
*c*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*C^2*a^6*c*f^2
- 8*C^2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^2 + 48*C^2*a^5*b*d*f^2 + 120*C^2*a^2*b
^4*c*f^2 - 120*C^2*a^4*b^2*c*f^2 - 160*C^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4
 + 16*d^2*f^4)*(C^4*a^12 + C^4*b^12 + 6*C^4*a^2*b^10 + 15*C^4*a^4*b^8 + 20*
C^4*a^6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*a^10*b^2))^(1/2) - 4*C^2*a^6*c*f^2 + 4
*C^2*b^6*c*f^2 - 24*C^2*a*b^5*d*f^2 - 24*C^2*a^5*b*d*f^2 - 60*C^2*a^2*b^4*c
*f^2 + 60*C^2*a^4*b^2*c*f^2 + 80*C^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)
))^(1/2))*((((8*C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^2 + 48*C
^2*a^5*b*d*f^2 + 120*C^2*a^2*b^4*c*f^2 - 120*C^2*a^4*b^2*c*f^2 - 160*C^2*a^
3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^12 + C^4*b^12 + 6*C^4*a
^2*b^10 + 15*C^4*a^4*b^8 + 20*C^4*a^6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*a^10*b^2
))^(1/2) - 4*C^2*a^6*c*f^2 + 4*C^2*b^6*c*f^2 - 24*C^2*a*b^5*d*f^2 - 24*C^2*
a^5*b*d*f^2 - 60*C^2*a^2*b^4*c*f^2 + 60*C^2*a^4*b^2*c*f^2 + 80*C^2*a^3*b^3*
d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(C^
2*a^6*d^2 - C^2*b^6*d^2 + 15*C^2*a^2*b^4*d^2 - 15*C^2*a^4*b^2*d^2))/f^2)*((
((8*C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^2 + 48*C^2*a^5*b*d*f
^2 + 120*C^2*a^2*b^4*c*f^2 - 120*C^2*a^4*b^2*c*f^2 - 160*C^2*a^3*b^3*d*f^2)
^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^12 + C^4*b^12 + 6*C^4*a^2*b^10 + 15
*C^4*a^4*b^8 + 20*C^4*a^6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*a^10*b^2))^(1/2) - 4
*C^2*a^6*c*f^2 + 4*C^2*b^6*c*f^2 - 24*C^2*a*b^5*d*f^2 - 24*C^2*a^5*b*d*f^2
- 60*C^2*a^2*b^4*c*f^2 + 60*C^2*a^4*b^2*c*f^2 + 80*C^2*a^3*b^3*d*f^2)/(16*(
c^2*f^4 + d^2*f^4)))^(1/2) + (16*(6*C^3*a^4*b^5*d^2 - C^3*b^9*d^2 + 8*C^3*a
^6*b^3*d^2 + 3*C^3*a^8*b*d^2))/f^3))*((((8*C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2
+ 48*C^2*a*b^5*d*f^2 + 48*C^2*a^5*b*d*f^2 + 120*C^2*a^2*b^4*c*f^2 - 120*C^2
*a^4*b^2*c*f^2 - 160*C^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^
4*a^12 + C^4*b^12 + 6*C^4*a^2*b^10 + 15*C^4*a^4*b^8 + 20*C^4*a^6*b^6 + 15*C
^4*a^8*b^4 + 6*C^4*a^10*b^2))^(1/2) - 4*C^2*a^6*c*f^2 + 4*C^2*b^6*c*f^2 - 2
4*C^2*a*b^5*d*f^2 - 24*C^2*a^5*b*d*f^2 - 60*C^2*a^2*b^4*c*f^2 + 60*C^2*a^4*
b^2*c*f^2 + 80*C^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*2i - atan
(((((8*(4*C*a^3*d^3*f^2 - 12*C*a*b^2*d^3*f^2 + 4*C*b^3*c*d^2*f^2 - 12*C*a^2
*b*c*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^6*c*f
^2 - 8*C^2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^2 + 48*C^2*a^5*b*d*f^2 + 120*C^2*a^
2*b^4*c*f^2 - 120*C^2*a^4*b^2*c*f^2 - 160*C^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*
f^4 + 16*d^2*f^4)*(C^4*a^12 + C^4*b^12 + 6*C^4*a^2*b^10 + 15*C^4*a^4*b^8 +
20*C^4*a^6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*a^10*b^2))^(1/2) + 4*C^2*a^6*c*f^2
- 4*C^2*b^6*c*f^2 + 24*C^2*a*b^5*d*f^2 + 24*C^2*a^5*b*d*f^2 + 60*C^2*a^2*b^
4*c*f^2 - 60*C^2*a^4*b^2*c*f^2 - 80*C^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f
^4)))^(1/2))*(-(((8*C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^2 +
48*C^2*a^5*b*d*f^2 + 120*C^2*a^2*b^4*c*f^2 - 120*C^2*a^4*b^2*c*f^2 - 160*C^
2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^12 + C^4*b^12 + 6*C
^4*a^2*b^10 + 15*C^4*a^4*b^8 + 20*C^4*a^6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*a^10
*b^2))^(1/2) + 4*C^2*a^6*c*f^2 - 4*C^2*b^6*c*f^2 + 24*C^2*a*b^5*d*f^2 + 24*
C^2*a^5*b*d*f^2 + 60*C^2*a^2*b^4*c*f^2 - 60*C^2*a^4*b^2*c*f^2 - 80*C^2*a^3*

b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)
*(C^2*a^6*d^2 - C^2*b^6*d^2 + 15*C^2*a^2*b^4*d^2 - 15*C^2*a^4*b^2*d^2))/f^2
)*(-(((8*C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^2 + 48*C^2*a^5*
b*d*f^2 + 120*C^2*a^2*b^4*c*f^2 - 120*C^2*a^4*b^2*c*f^2 - 160*C^2*a^3*b^3*d
*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^12 + C^4*b^12 + 6*C^4*a^2*b^10
 + 15*C^4*a^4*b^8 + 20*C^4*a^6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*a^10*b^2))^(1/2
) + 4*C^2*a^6*c*f^2 - 4*C^2*b^6*c*f^2 + 24*C^2*a*b^5*d*f^2 + 24*C^2*a^5*b*d
*f^2 + 60*C^2*a^2*b^4*c*f^2 - 60*C^2*a^4*b^2*c*f^2 - 80*C^2*a^3*b^3*d*f^2)/
(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i - (((8*(4*C*a^3*d^3*f^2 - 12*C*a*b^2*d^3
*f^2 + 4*C*b^3*c*d^2*f^2 - 12*C*a^2*b*c*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan
(e + f*x))^(1/2)*(-(((8*C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^
2 + 48*C^2*a^5*b*d*f^2 + 120*C^2*a^2*b^4*c*f^2 - 120*C^2*a^4*b^2*c*f^2 - 16
0*C^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^12 + C^4*b^12 +
 6*C^4*a^2*b^10 + 15*C^4*a^4*b^8 + 20*C^4*a^6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*
a^10*b^2))^(1/2) + 4*C^2*a^6*c*f^2 - 4*C^2*b^6*c*f^2 + 24*C^2*a*b^5*d*f^2 +
 24*C^2*a^5*b*d*f^2 + 60*C^2*a^2*b^4*c*f^2 - 60*C^2*a^4*b^2*c*f^2 - 80*C^2*
a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*C^2*a^6*c*f^2 - 8*C^
2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^2 + 48*C^2*a^5*b*d*f^2 + 120*C^2*a^2*b^4*c*f
^2 - 120*C^2*a^4*b^2*c*f^2 - 160*C^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*
d^2*f^4)*(C^4*a^12 + C^4*b^12 + 6*C^4*a^2*b^10 + 15*C^4*a^4*b^8 + 20*C^4*a^
6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*a^10*b^2))^(1/2) + 4*C^2*a^6*c*f^2 - 4*C^2*b
^6*c*f^2 + 24*C^2*a*b^5*d*f^2 + 24*C^2*a^5*b*d*f^2 + 60*C^2*a^2*b^4*c*f^2 -
 60*C^2*a^4*b^2*c*f^2 - 80*C^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/
2) + (16*(c + d*tan(e + f*x))^(1/2)*(C^2*a^6*d^2 - C^2*b^6*d^2 + 15*C^2*a^2
*b^4*d^2 - 15*C^2*a^4*b^2*d^2))/f^2)*(-(((8*C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2
 + 48*C^2*a*b^5*d*f^2 + 48*C^2*a^5*b*d*f^2 + 120*C^2*a^2*b^4*c*f^2 - 120*C^
2*a^4*b^2*c*f^2 - 160*C^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C
^4*a^12 + C^4*b^12 + 6*C^4*a^2*b^10 + 15*C^4*a^4*b^8 + 20*C^4*a^6*b^6 + 15*
C^4*a^8*b^4 + 6*C^4*a^10*b^2))^(1/2) + 4*C^2*a^6*c*f^2 - 4*C^2*b^6*c*f^2 +
24*C^2*a*b^5*d*f^2 + 24*C^2*a^5*b*d*f^2 + 60*C^2*a^2*b^4*c*f^2 - 60*C^2*a^4
*b^2*c*f^2 - 80*C^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i)/((((
8*(4*C*a^3*d^3*f^2 - 12*C*a*b^2*d^3*f^2 + 4*C*b^3*c*d^2*f^2 - 12*C*a^2*b*c*
d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^6*c*f^2 -
8*C^2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^2 + 48*C^2*a^5*b*d*f^2 + 120*C^2*a^2*b^4
*c*f^2 - 120*C^2*a^4*b^2*c*f^2 - 160*C^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 +
 16*d^2*f^4)*(C^4*a^12 + C^4*b^12 + 6*C^4*a^2*b^10 + 15*C^4*a^4*b^8 + 20*C^
4*a^6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*a^10*b^2))^(1/2) + 4*C^2*a^6*c*f^2 - 4*C
^2*b^6*c*f^2 + 24*C^2*a*b^5*d*f^2 + 24*C^2*a^5*b*d*f^2 + 60*C^2*a^2*b^4*c*f
^2 - 60*C^2*a^4*b^2*c*f^2 - 80*C^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))
^(1/2))*(-(((8*C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^2 + 48*C^
2*a^5*b*d*f^2 + 120*C^2*a^2*b^4*c*f^2 - 120*C^2*a^4*b^2*c*f^2 - 160*C^2*a^3
*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^12 + C^4*b^12 + 6*C^4*a^
2*b^10 + 15*C^4*a^4*b^8 + 20*C^4*a^6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*a^10*b^2)
)^(1/2) + 4*C^2*a^6*c*f^2 - 4*C^2*b^6*c*f^2 + 24*C^2*a*b^5*d*f^2 + 24*C^2*a
^5*b*d*f^2 + 60*C^2*a^2*b^4*c*f^2 - 60*C^2*a^4*b^2*c*f^2 - 80*C^2*a^3*b^3*d

f^2)/(16(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(C^2
*a^6*d^2 - C^2*b^6*d^2 + 15*C^2*a^2*b^4*d^2 - 15*C^2*a^4*b^2*d^2))/f^2)*(-(
((8*C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^2 + 48*C^2*a^5*b*d*f
^2 + 120*C^2*a^2*b^4*c*f^2 - 120*C^2*a^4*b^2*c*f^2 - 160*C^2*a^3*b^3*d*f^2)
^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^12 + C^4*b^12 + 6*C^4*a^2*b^10 + 15
*C^4*a^4*b^8 + 20*C^4*a^6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*a^10*b^2))^(1/2) + 4
*C^2*a^6*c*f^2 - 4*C^2*b^6*c*f^2 + 24*C^2*a*b^5*d*f^2 + 24*C^2*a^5*b*d*f^2
+ 60*C^2*a^2*b^4*c*f^2 - 60*C^2*a^4*b^2*c*f^2 - 80*C^2*a^3*b^3*d*f^2)/(16*(
c^2*f^4 + d^2*f^4)))^(1/2) + (((8*(4*C*a^3*d^3*f^2 - 12*C*a*b^2*d^3*f^2 + 4
*C*b^3*c*d^2*f^2 - 12*C*a^2*b*c*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x
))^(1/2)*(-(((8*C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^2 + 48*C
^2*a^5*b*d*f^2 + 120*C^2*a^2*b^4*c*f^2 - 120*C^2*a^4*b^2*c*f^2 - 160*C^2*a^
3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^12 + C^4*b^12 + 6*C^4*a
^2*b^10 + 15*C^4*a^4*b^8 + 20*C^4*a^6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*a^10*b^2
))^(1/2) + 4*C^2*a^6*c*f^2 - 4*C^2*b^6*c*f^2 + 24*C^2*a*b^5*d*f^2 + 24*C^2*
a^5*b*d*f^2 + 60*C^2*a^2*b^4*c*f^2 - 60*C^2*a^4*b^2*c*f^2 - 80*C^2*a^3*b^3*
d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*C^2*a^6*c*f^2 - 8*C^2*b^6*c*
f^2 + 48*C^2*a*b^5*d*f^2 + 48*C^2*a^5*b*d*f^2 + 120*C^2*a^2*b^4*c*f^2 - 120
*C^2*a^4*b^2*c*f^2 - 160*C^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)
*(C^4*a^12 + C^4*b^12 + 6*C^4*a^2*b^10 + 15*C^4*a^4*b^8 + 20*C^4*a^6*b^6 +
15*C^4*a^8*b^4 + 6*C^4*a^10*b^2))^(1/2) + 4*C^2*a^6*c*f^2 - 4*C^2*b^6*c*f^2
 + 24*C^2*a*b^5*d*f^2 + 24*C^2*a^5*b*d*f^2 + 60*C^2*a^2*b^4*c*f^2 - 60*C^2*
a^4*b^2*c*f^2 - 80*C^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (16
*(c + d*tan(e + f*x))^(1/2)*(C^2*a^6*d^2 - C^2*b^6*d^2 + 15*C^2*a^2*b^4*d^2
 - 15*C^2*a^4*b^2*d^2))/f^2)*(-(((8*C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2 + 48*C^
2*a*b^5*d*f^2 + 48*C^2*a^5*b*d*f^2 + 120*C^2*a^2*b^4*c*f^2 - 120*C^2*a^4*b^
2*c*f^2 - 160*C^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^12
+ C^4*b^12 + 6*C^4*a^2*b^10 + 15*C^4*a^4*b^8 + 20*C^4*a^6*b^6 + 15*C^4*a^8*
b^4 + 6*C^4*a^10*b^2))^(1/2) + 4*C^2*a^6*c*f^2 - 4*C^2*b^6*c*f^2 + 24*C^2*a
*b^5*d*f^2 + 24*C^2*a^5*b*d*f^2 + 60*C^2*a^2*b^4*c*f^2 - 60*C^2*a^4*b^2*c*f
^2 - 80*C^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (16*(6*C^3*a^4
*b^5*d^2 - C^3*b^9*d^2 + 8*C^3*a^6*b^3*d^2 + 3*C^3*a^8*b*d^2))/f^3))*(-(((8
*C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^2 + 48*C^2*a^5*b*d*f^2
+ 120*C^2*a^2*b^4*c*f^2 - 120*C^2*a^4*b^2*c*f^2 - 160*C^2*a^3*b^3*d*f^2)^2/
4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^12 + C^4*b^12 + 6*C^4*a^2*b^10 + 15*C^
4*a^4*b^8 + 20*C^4*a^6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*a^10*b^2))^(1/2) + 4*C^
2*a^6*c*f^2 - 4*C^2*b^6*c*f^2 + 24*C^2*a*b^5*d*f^2 + 24*C^2*a^5*b*d*f^2 + 6
0*C^2*a^2*b^4*c*f^2 - 60*C^2*a^4*b^2*c*f^2 - 80*C^2*a^3*b^3*d*f^2)/(16*(c^2
*f^4 + d^2*f^4)))^(1/2)*2i - ((6*A*b^3*c - 6*A*a*b^2*d)/(d^2*f) - (4*A*b^3*
c)/(d^2*f))*(c + d*tan(e + f*x))^(1/2) - ((8*B*b^3*c - 6*B*a*b^2*d)/(3*d^3*
f) - (4*B*b^3*c)/(3*d^3*f))*(c + d*tan(e + f*x))^(3/2) - ((10*C*b^3*c - 6*C
*a*b^2*d)/(5*d^4*f) - (4*C*b^3*c)/(5*d^4*f))*(c + d*tan(e + f*x))^(5/2) + (
c + d*tan(e + f*x))^(1/2)*((2*C*a^3*d^3 - 20*C*b^3*c^3 + 36*C*a*b^2*c^2*d -
 18*C*a^2*b*c*d^2)/(d^4*f) - 2*c*(2*c*((10*C*b^3*c - 6*C*a*b^2*d)/(d^4*f) -
 (4*C*b^3*c)/(d^4*f)) - (20*C*b^3*c^2 + 6*C*a^2*b*d^2 - 24*C*a*b^2*c*d)/(d^

```
4*f) + (2*C*b^3*(d^6*f + c^2*d^4*f))/(d^8*f^2)) + ((d^6*f + c^2*d^4*f)*((10
*C*b^3*c - 6*C*a*b^2*d)/(d^4*f) - (4*C*b^3*c)/(d^4*f)))/(d^4*f)) + (2*A*b^3
*(c + d*tan(e + f*x))^(3/2))/(3*d^2*f) + (2*B*b^3*(c + d*tan(e + f*x))^(5/2
))/(5*d^3*f) + (2*C*b^3*(c + d*tan(e + f*x))^(7/2))/(7*d^4*f)
```

3.111 $\int \frac{(a+b\tan(e+fx))^2(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{c+d\tan(e+fx)}}\,dx$

Optimal result 1467
Rubi [A] (verified) 1468
Mathematica [A] (verified) 1472
Maple [B] (verified) 1472
Fricas [B] (verification not implemented) 1473
Sympy [F] 1473
Maxima [F(-1)] 1473
Giac [F(-1)] 1474
Mupad [B] (verification not implemented) 1474

Optimal result

Integrand size = 47, antiderivative size = 287

$$\int \frac{(a+b\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx$$
$$= -\frac{(a-ib)^2(B+i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{\sqrt{c-id}f}$$
$$+\frac{(a+ib)^2(iA-B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{\sqrt{c+id}f}$$
$$+\frac{2(12a^2Cd^2-10abd(2cC-3Bd)+b^2(8c^2C-10Bcd+15(A-C)d^2))\sqrt{c+d\tan(e+fx)}}{15d^3f}$$
$$-\frac{2b(4bcC-5bBd-4aCd)\tan(e+fx)\sqrt{c+d\tan(e+fx)}}{15d^2f}$$
$$+\frac{2C(a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}}{5df}$$

```
[Out] -(a-I*b)^2*(B+I*(A-C))*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/f/(c-I
*d)^(1/2)+(a+I*b)^2*(I*A-B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2
))/f/(c+I*d)^(1/2)+2/15*(12*a^2*C*d^2-10*a*b*d*(-3*B*d+2*C*c)+b^2*(8*c^2*C-
10*B*c*d+15*(A-C)*d^2))*(c+d*tan(f*x+e))^(1/2)/d^3/f-2/15*b*(-5*B*b*d-4*C*a
*d+4*C*b*c)*(c+d*tan(f*x+e))^(1/2)*tan(f*x+e)/d^2/f+2/5*C*(c+d*tan(f*x+e))^
(1/2)*(a+b*tan(f*x+e))^2/d/f
```

Rubi [A] (verified)

Time = 1.05 (sec) , antiderivative size = 287, normalized size of antiderivative = 1.00, number of steps used = 10, number of rules used = 7, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.149, Rules used = {3728, 3718, 3711, 3620, 3618, 65, 214}

$$\int \frac{(a+b\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx$$
$$=\frac{2\sqrt{c+d\tan(e+fx)}(12a^2Cd^2-10abd(2cC-3Bd)+b^2(15d^2(A-C)-10Bcd+8c^2C))}{15d^3f}$$
$$-\frac{(a-ib)^2(B+i(A-C))\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f\sqrt{c-id}}$$
$$+\frac{(a+ib)^2(iA-B-iC)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f\sqrt{c+id}}$$
$$-\frac{2b\tan(e+fx)(-4aCd-5bBd+4bcC)\sqrt{c+d\tan(e+fx)}}{15d^2f}$$
$$+\frac{2C(a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}}{5df}$$

```
[In] Int[((a + b*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/Sqrt[c
 + d*Tan[e + f*x]],x]
```

```
[Out] -(((a - I*b)^2*(B + I*(A - C))*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*
d]])/(Sqrt[c - I*d]*f)) + ((a + I*b)^2*(I*A - B - I*C)*ArcTanh[Sqrt[c + d*T
an[e + f*x]]/Sqrt[c + I*d]])/(Sqrt[c + I*d]*f) + (2*(12*a^2*C*d^2 - 10*a*b*
d*(2*c*C - 3*B*d) + b^2*(8*c^2*C - 10*B*c*d + 15*(A - C)*d^2))*Sqrt[c + d*T
an[e + f*x]])/(15*d^3*f) - (2*b*(4*b*c*C - 5*b*B*d - 4*a*C*d)*Tan[e + f*x]*
Sqrt[c + d*Tan[e + f*x]])/(15*d^2*f) + (2*C*(a + b*Tan[e + f*x])^2*Sqrt[c +
 d*Tan[e + f*x]])/(5*d*f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

integral

$$= \frac{2C(a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}}{5df} + \frac{2\int \frac{(a+b\tan(e+fx))\left(\frac{1}{2}(-4bcC+a(5A-C)d)+\frac{5}{2}(Ab+aB-bC)d\tan(e+fx)-\frac{1}{2}(4bcC-5bBd-4aCd)\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx}{5d}$$

$$= -\frac{2b(4bcC-5bBd-4aCd)\tan(e+fx)\sqrt{c+d\tan(e+fx)}}{15d^2f} + \frac{2C(a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}}{5df} - \frac{4\int \frac{\frac{1}{4}\left(20abcCd-3a^2(5A-C)d^2-4b^2\left(2c^2C-\frac{5Bcd}{2}\right)\right)-\frac{15}{4}\left(a^2B-b^2B+2ab(A-C)\right)d^2\tan(e+fx)-\frac{1}{4}\left(12a^2Cd^2-10abd(2cC-3Bd)+b^2(8c^2C-}{\sqrt{c+d\tan(e+fx)}}}{15d^2}$$

$$= \frac{2\left(12a^2Cd^2-10abd(2cC-3Bd)+b^2(8c^2C-10Bcd+15(A-C)d^2)\right)\sqrt{c+d\tan(e+fx)}}{15d^3f} - \frac{2b(4bcC-5bBd-4aCd)\tan(e+fx)\sqrt{c+d\tan(e+fx)}}{15d^2f} + \frac{2C(a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}}{5df} - \frac{4\int \frac{\frac{15}{4}\left(2abB-a^2(A-C)+b^2(A-C)\right)d^2-\frac{15}{4}\left(a^2B-b^2B+2ab(A-C)\right)d^2\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{15d^2}$$

$$= \frac{2\left(12a^2Cd^2-10abd(2cC-3Bd)+b^2(8c^2C-10Bcd+15(A-C)d^2)\right)\sqrt{c+d\tan(e+fx)}}{15d^3f} - \frac{2b(4bcC-5bBd-4aCd)\tan(e+fx)\sqrt{c+d\tan(e+fx)}}{15d^2f} + \frac{2C(a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}}{5df} + \frac{1}{2}\left((a-ib)^2(A-iB-C)\right)\int \frac{1+i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx + \frac{1}{2}\left((a+ib)^2(A+iB-C)\right)\int \frac{1-i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx$$

$$
\begin{aligned}
&= \frac{2(12a^2Cd^2 - 10abd(2cC - 3Bd) + b^2(8c^2C - 10Bcd + 15(A-C)d^2))\sqrt{c+d\tan(e+fx)}}{15d^3f} \\
&\quad - \frac{2b(4bcC - 5bBd - 4aCd)\tan(e+fx)\sqrt{c+d\tan(e+fx)}}{15d^2f} \\
&\quad + \frac{2C(a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}}{5df} \\
&\quad - \frac{(i(a+ib)^2(A+iB-C))\,\mathrm{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c+idx}}\,dx, x, -i\tan(e+fx)\right)}{2f} \\
&\quad + \frac{((a-ib)^2(iA+B-iC))\,\mathrm{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c-idx}}\,dx, x, i\tan(e+fx)\right)}{2f} \\
&= \frac{2(12a^2Cd^2 - 10abd(2cC - 3Bd) + b^2(8c^2C - 10Bcd + 15(A-C)d^2))\sqrt{c+d\tan(e+fx)}}{15d^3f} \\
&\quad - \frac{2b(4bcC - 5bBd - 4aCd)\tan(e+fx)\sqrt{c+d\tan(e+fx)}}{15d^2f} \\
&\quad + \frac{2C(a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}}{5df} \\
&\quad - \frac{((a-ib)^2(A-iB-C))\,\mathrm{Subst}\left(\int \frac{1}{-1-\frac{ic}{d}+\frac{ix^2}{d}}\,dx, x, \sqrt{c+d\tan(e+fx)}\right)}{df} \\
&\quad - \frac{((a+ib)^2(A+iB-C))\,\mathrm{Subst}\left(\int \frac{1}{-1+\frac{ic}{d}-\frac{ix^2}{d}}\,dx, x, \sqrt{c+d\tan(e+fx)}\right)}{df} \\
&= -\frac{(a-ib)^2(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{\sqrt{c-id}\,f} \\
&\quad - \frac{(a+ib)^2(B-i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{\sqrt{c+id}\,f} \\
&\quad + \frac{2(12a^2Cd^2 - 10abd(2cC - 3Bd) + b^2(8c^2C - 10Bcd + 15(A-C)d^2))\sqrt{c+d\tan(e+fx)}}{15d^3f} \\
&\quad - \frac{2b(4bcC - 5bBd - 4aCd)\tan(e+fx)\sqrt{c+d\tan(e+fx)}}{15d^2f} \\
&\quad + \frac{2C(a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}}{5df}
\end{aligned}
$$

Mathematica [A] (verified)

Time = 6.36 (sec) , antiderivative size = 414, normalized size of antiderivative = 1.44

$$\int \frac{(a+b\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{c+d\tan(e+fx)}}\,dx$$
$$= \frac{2C(a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}}{5df}$$
$$+ \frac{2\left(\frac{b(-4bcC+5bBd+4aCd)\tan(e+fx)\sqrt{c+d\tan(e+fx)}}{3df} - \frac{2\left(\frac{i\sqrt{c-id}\left(\frac{15}{4}i\left(a^2B-b^2B+2ab(A-C)\right)d^2+\frac{15}{4}\left(2abB-a^2(A-C)+b^2(A-C)\right)d^2\right)\text{arcta}}{(-c+id)f}\right.}{}\right.}{}$$

```
[In] Integrate[((a + b*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
Sqrt[c + d*Tan[e + f*x]],x]
```

```
[Out] (2*C*(a + b*Tan[e + f*x])^2*Sqrt[c + d*Tan[e + f*x]])/(5*d*f) + (2*((b*(-4*
b*c*C + 5*b*B*d + 4*a*C*d)*Tan[e + f*x]*Sqrt[c + d*Tan[e + f*x]])/(3*d*f) -
 (2*((I*Sqrt[c - I*d]*(((15*I)/4)*(a^2*B - b^2*B + 2*a*b*(A - C))*d^2 + (15
*(2*a*b*B - a^2*(A - C) + b^2*(A - C))*d^2)/4)*ArcTanh[Sqrt[c + d*Tan[e + f
*x]]/Sqrt[c - I*d]])/((-c + I*d)*f) - (I*Sqrt[c + I*d]*(((-15*I)/4)*(a^2*B
- b^2*B + 2*a*b*(A - C))*d^2 + (15*(2*a*b*B - a^2*(A - C) + b^2*(A - C))*d^
2)/4)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]])/((-c - I*d)*f) + ((-
12*a^2*C*d^2 + 10*a*b*d*(2*c*C - 3*B*d) - b^2*(8*c^2*C - 10*B*c*d + 15*(A -
 C)*d^2))*Sqrt[c + d*Tan[e + f*x]])/(2*d*f)))/(3*d)))/(5*d)
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 5512 vs. 2(254) = 508.

Time = 0.16 (sec) , antiderivative size = 5513, normalized size of antiderivative = 19.21

method	result	size
parts	Expression too large to display	5513
derivativedivides	Expression too large to display	18289
default	Expression too large to display	18289

```
[In] int((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/
2),x,method=_RETURNVERBOSE)
```

```
[Out] result too large to display
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 25627 vs. $2(244) = 488$.

Time = 4.81 (sec) , antiderivative size = 25627, normalized size of antiderivative = 89.29

$$\int \frac{(a+b\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{c+d\tan(e+fx)}}\,dx = \text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^(1/2),x, algorithm="fricas")
```

```
[Out] Too large to include
```

Sympy [F]

$$\int \frac{(a+b\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{c+d\tan(e+fx)}}\,dx = \int \frac{(a+b\tan{(e+fx)})^2\,(A+B\tan{(e+fx)}+C\tan^2{(e+fx)})}{\sqrt{c+d\tan{(e+fx)}}}\,dx$$

```
[In] integrate((a+b*tan(f*x+e))**2*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x
+e))**(1/2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**2*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/s
qrt(c + d*tan(e + f*x)), x)
```

Maxima [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{c+d\tan(e+fx)}}\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^(1/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

Giac [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx=\text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^(1/2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [B] (verification not implemented)

Time = 43.42 (sec) , antiderivative size = 21254, normalized size of antiderivative = 74.06

$$\int \frac{(a+b\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx=\text{Too large to display}$$

```
[In] int(((a + b*tan(e + f*x))^2*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d
*tan(e + f*x))^(1/2),x)
```

```
[Out] atan(((((16*(2*C*b^2*d^3*f^2 - 2*C*a^2*d^3*f^2 + 4*C*a*b*c*d^2*f^2))/f^3 -
64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 -
 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (16*
c^2*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 +
4*C^4*a^6*b^2))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*
f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^
(1/2))*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*
a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^
8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) - 4*C^2
*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24
*C^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*
x))^(1/2)*(C^2*a^4*d^2 + C^2*b^4*d^2 - 6*C^2*a^2*b^2*d^2))/f^2)*((((8*C^2*a
^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C
^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^
4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2
*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2
)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i - (((16*(2*C*b^2*d^3*f^2 - 2*C*a^2*d^3
*f^2 + 4*C*a*b*c*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*
C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 -
 48*C^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 +
 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) - 4*C^2*a^4*c*f^2 -
4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*
c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f
```

^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(C^2*a^4*d^2 + C^2*b^4*d^2 - 6*C^2*a^2*b^2*d^2))/f^2)*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i)/((((16*(2*C*b^2*d^3*f^2 - 2*C*a^2*d^3*f^2 + 4*C*a*b*c*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(C^2*a^4*d^2 + C^2*b^4*d^2 - 6*C^2*a^2*b^2*d^2))/f^2)*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (((16*(2*C*b^2*d^3*f^2 - 2*C*a^2*d^3*f^2 + 4*C*a*b*c*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(C^2*a^4*d^2 + C^2*b^4*d^2 - 6*C^2*a^2*b^2*d^2))/f^2)*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (32*(2*C^3*a^3*b^3*d^2 + C^3*a*b^5*d^2 + C^3*a^5*b*d^2))/f^3))*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2

/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a
^4*b^4 + 4*C^4*a^6*b^2))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2
*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^
2*f^4)))^(1/2)*2i - atan(((((8*(4*B*a^2*c*d^2*f^2 - 4*B*b^2*c*d^2*f^2 + 8*B
*a*b*d^3*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^4*c*f^
2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*
b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b
^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*
f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(
c^2*f^4 + d^2*f^4)))^(1/2))*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*
a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4
+ 16*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^
6*b^2))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16
*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) +
(16*(c + d*tan(e + f*x))^(1/2)*(B^2*a^4*d^2 + B^2*b^4*d^2 - 6*B^2*a^2*b^2*d
^2))/f^2)*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B
^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4
*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) + 4*
B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 -
24*B^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i - (((8*(4*B*a^2*c
*d^2*f^2 - 4*B*b^2*c*d^2*f^2 + 8*B*a*b*d^3*f^2))/f^3 + 64*c*d^2*(c + d*tan(
e + f*x))^(1/2)*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2
+ 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4
)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2
) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d
*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((((8*B^2*a^4
*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2
*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*
a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b
^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/
(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(B^2*a^4*d
^2 + B^2*b^4*d^2 - 6*B^2*a^2*b^2*d^2))/f^2)*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4
*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/
4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^
4*b^4 + 4*B^4*a^6*b^2))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*
a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2
*f^4)))^(1/2)*1i)/((((8*(4*B*a^2*c*d^2*f^2 - 4*B*b^2*c*d^2*f^2 + 8*B*a*b*d^
3*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^4*c*f^2 + 8*B
^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f
^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*
B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 1
6*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(c^2*f^4
+ d^2*f^4)))^(1/2))*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d
*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^
2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))

^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^
3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (16*(c
+ d*tan(e + f*x))^(1/2)*(B^2*a^4*d^2 + B^2*b^4*d^2 - 6*B^2*a^2*b^2*d^2))/f^
2)*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*
b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^8 +
B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) + 4*B^2*a^4
*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2
*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(B^3*a^6*d^2 - B^3*b^
6*d^2 - B^3*a^2*b^4*d^2 + B^3*a^4*b^2*d^2))/f^3 + (((8*(4*B*a^2*c*d^2*f^2 -
4*B*b^2*c*d^2*f^2 + 8*B*a*b*d^3*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^
(1/2)*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a
^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^8
+ B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) + 4*B^2*
a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*
B^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((((8*B^2*a^4*c*f^2 + 8
*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c
*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 +
6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 -
16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(c^2*f
^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(B^2*a^4*d^2 + B^2*b
^4*d^2 - 6*B^2*a^2*b^2*d^2))/f^2)*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 3
2*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (16*c^
2*f^4 + 16*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*
B^4*a^6*b^2))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^
2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1
/2)))*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a
^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^8
+ B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) + 4*B^2*
a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*
B^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*2i - atan(((((16*(2*A*b^
2*d^3*f^2 - 2*A*a^2*d^3*f^2 + 4*A*a*b*c*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan
(e + f*x))^(1/2)*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2
+ 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^
4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/
2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*
d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((((8*A^2*a^
4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^
2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4
*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*
b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)
/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^4*
d^2 + A^2*b^4*d^2 - 6*A^2*a^2*b^2*d^2))/f^2)*((((8*A^2*a^4*c*f^2 + 8*A^2*b^
4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2
/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a
^4*b^4 + 4*A^4*a^6*b^2))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2

```
*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^
2*f^4)))^(1/2)*1i - (((16*(2*A*b^2*d^3*f^2 - 2*A*a^2*d^3*f^2 + 4*A*a*b*c*d^
2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^4*c*f^2 + 8*A
^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f
^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*
A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 1
6*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(c^2*f^4
 + d^2*f^4)))^(1/2))*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d
*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^
2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))
^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^
3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (16*(c
+ d*tan(e + f*x))^(1/2)*(A^2*a^4*d^2 + A^2*b^4*d^2 - 6*A^2*a^2*b^2*d^2))/f^
2)*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*
b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^8 +
A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) - 4*A^2*a^4
*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2
*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i)/((((16*(2*A*b^2*d^3*f^2
 - 2*A*a^2*d^3*f^2 + 4*A*a*b*c*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x)
)^(1/2)*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2
*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a
^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) - 4*A^
2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 2
4*A^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((((8*A^2*a^4*c*f^2 +
 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2
*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6
+ 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2
 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(c^2
*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^4*d^2 + A^2
*b^4*d^2 - 6*A^2*a^2*b^2*d^2))/f^2)*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 -
 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (16*
c^2*f^4 + 16*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 +
4*A^4*a^6*b^2))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*
f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^
(1/2) + (((16*(2*A*b^2*d^3*f^2 - 2*A*a^2*d^3*f^2 + 4*A*a*b*c*d^2*f^2))/f^3
+ 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2
 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (1
6*c^2*f^4 + 16*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4
+ 4*A^4*a^6*b^2))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*
d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4))
)^(1/2))*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^
2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*
a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) - 4*A
^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 +
24*A^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (16*(c + d*tan(e +
```

f*x))^(1/2)*(A^2*a^4*d^2 + A^2*b^4*d^2 - 6*A^2*a^2*b^2*d^2))/f^2)*((((8*A^2
*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48
*A^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*
A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A
^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f
^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (32*(2*A^3*a^3*b^3*d^2 + A^3*a*b^5*d^
2 + A^3*a^5*b*d^2))/f^3))*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*
b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 +
16*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*
b^2))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A
^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*2i -
atan(((((16*(2*A*b^2*d^3*f^2 - 2*A*a^2*d^3*f^2 + 4*A*a*b*c*d^2*f^2))/f^3 -
64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2
- 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (1
6*c^2*f^4 + 16*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4
+ 4*A^4*a^6*b^2))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*
d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4))
)^(1/2))*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A
^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4
*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) + 4*
A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 -
24*A^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e +
f*x))^(1/2)*(A^2*a^4*d^2 + A^2*b^4*d^2 - 6*A^2*a^2*b^2*d^2))/f^2)*(-(((8*A
^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 -
48*A^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^8 + A^4*b^8 +
4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) + 4*A^2*a^4*c*f^2 + 4
*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c
f^2)/(16(c^2*f^4 + d^2*f^4)))^(1/2)*1i - (((16*(2*A*b^2*d^3*f^2 - 2*A*a^2
*d^3*f^2 + 4*A*a*b*c*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-
(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*
f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^8 + A^4*
b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) + 4*A^2*a^4*c*f
^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2
*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b
^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^
2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*
a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^
2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d
^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^4*d^2 + A^2*b^4*d^2
- 6*A^2*a^2*b^2*d^2))/f^2)*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2
*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4
+ 16*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a
^6*b^2))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 1
6*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1
i)/((((16*(2*A*b^2*d^3*f^2 - 2*A*a^2*d^3*f^2 + 4*A*a*b*c*d^2*f^2))/f^3 - 64

*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 -
32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (16*c
^2*f^4 + 16*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4
*A^4*a^6*b^2))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f
^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(
1/2))*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*
a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^
8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) + 4*A^2
*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24
*A^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*
x))^(1/2)*(A^2*a^4*d^2 + A^2*b^4*d^2 - 6*A^2*a^2*b^2*d^2))/f^2)*(-(((8*A^2*
a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*
A^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A
^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^
2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^
2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (((16*(2*A*b^2*d^3*f^2 - 2*A*a^2*d^3*f
^2 + 4*A*a*b*c*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*A
^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 -
48*A^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^8 + A^4*b^8 +
4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) + 4*A^2*a^4*c*f^2 + 4
*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c
f^2)/(16(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f
^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 -
(16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^
4 + 4*A^4*a^6*b^2))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^
3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4
)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^4*d^2 + A^2*b^4*d^2 - 6*A
^2*a^2*b^2*d^2))/f^2)*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3
*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*
d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2
))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*
a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (32*(
2*A^3*a^3*b^3*d^2 + A^3*a*b^5*d^2 + A^3*a^5*b*d^2))/f^3))*(-(((8*A^2*a^4*c*
f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^
2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2
*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*
c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16
*(c^2*f^4 + d^2*f^4)))^(1/2)*2i - ((6*B*b^2*c - 4*B*a*b*d)/(d^2*f) - (4*B*b
^2*c)/(d^2*f))*(c + d*tan(e + f*x))^(1/2) - ((8*C*b^2*c - 4*C*a*b*d)/(3*d^3
*f) - (4*C*b^2*c)/(3*d^3*f))*(c + d*tan(e + f*x))^(3/2) - (c + d*tan(e + f*
x))^(1/2)*(2*c*((8*C*b^2*c - 4*C*a*b*d)/(d^3*f) - (4*C*b^2*c)/(d^3*f)) - (2
*C*a^2*d^2 + 12*C*b^2*c^2 - 12*C*a*b*c*d)/(d^3*f) + (2*C*b^2*(d^5*f + c^2*d
^3*f))/(d^6*f^2)) - atan(((((8*(4*B*a^2*c*d^2*f^2 - 4*B*b^2*c*d^2*f^2 + 8*B
*a*b*d^3*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^4*c*f
^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2

*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*
b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c
*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*
(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^
2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^
4 + 16*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*
a^6*b^2))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 -
16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)
+ (16*(c + d*tan(e + f*x))^(1/2)*(B^2*a^4*d^2 + B^2*b^4*d^2 - 6*B^2*a^2*b^2
d^2))/f^2)(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 3
2*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(
B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) -
 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^
2 + 24*B^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i - (((8*(4*B*a^
2*c*d^2*f^2 - 4*B*b^2*c*d^2*f^2 + 8*B*a*b*d^3*f^2))/f^3 + 64*c*d^2*(c + d*t
an(e + f*x))^(1/2)*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*
f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2
f^4)(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^
(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3
*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*B^
2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 4
8*B^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4
*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) - 4*B^2*a^4*c*f^2 - 4*
B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*
f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(B^2*
a^4*d^2 + B^2*b^4*d^2 - 6*B^2*a^2*b^2*d^2))/f^2)*(-(((8*B^2*a^4*c*f^2 + 8*B
^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f
^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*
B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 1
6*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(c^2*f^4
 + d^2*f^4)))^(1/2)*1i)/((((8*(4*B*a^2*c*d^2*f^2 - 4*B*b^2*c*d^2*f^2 + 8*B*
a*b*d^3*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^4*c*f^
2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*
b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b
^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*
f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(
c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2
*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4
 + 16*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a
^6*b^2))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 1
6*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) +
 (16*(c + d*tan(e + f*x))^(1/2)*(B^2*a^4*d^2 + B^2*b^4*d^2 - 6*B^2*a^2*b^2*
d^2))/f^2)*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32
*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B
^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) -

4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(B^3*a^6*d^2 - B^3*b^6*d^2 - B^3*a^2*b^4*d^2 + B^3*a^4*b^2*d^2))/f^3 + (((8*(4*B*a^2*c*d^2*f^2 - 4*B*b^2*c*d^2*f^2 + 8*B*a*b*d^3*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(B^2*a^4*d^2 + B^2*b^4*d^2 - 6*B^2*a^2*b^2*d^2))/f^2)*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)))*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*2i + atan((((16*(2*C*b^2*d^3*f^2 - 2*C*a^2*d^3*f^2 + 4*C*a*b*c*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(C^2*a^4*d^2 + C^2*b^4*d^2 - 6*C^2*a^2*b^2*d^2))/f^2)*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i - (((16*(2*C*b^2*d^3*f^2 - 2*C*a^2*d^3*f^2 + 4*C*a*b*c*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) + 4*C^2*a^4*c*f^2 + 4*

C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*
f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^
2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (
16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4
 + 4*C^4*a^6*b^2))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3
*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)
))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(C^2*a^4*d^2 + C^2*b^4*d^2 - 6*C^
2*a^2*b^2*d^2))/f^2)*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*
d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d
^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)
)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a
^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i)/((((
16*(2*C*b^2*d^3*f^2 - 2*C*a^2*d^3*f^2 + 4*C*a*b*c*d^2*f^2))/f^3 - 64*c*d^2*
(c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*
a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4
+ 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^
6*b^2))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16
*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(
-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d
*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4
*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) + 4*C^2*a^4*c*
f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^
2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/
2)*(C^2*a^4*d^2 + C^2*b^4*d^2 - 6*C^2*a^2*b^2*d^2))/f^2)*(-(((8*C^2*a^4*c*f
^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2
*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*
b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c
*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*
(c^2*f^4 + d^2*f^4)))^(1/2) + (((16*(2*C*b^2*d^3*f^2 - 2*C*a^2*d^3*f^2 + 4*
C*a*b*c*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^4*
c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*
a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a
^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^
4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(
16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32
*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (16*c^2
*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C
^4*a^6*b^2))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2
 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/
2) + (16*(c + d*tan(e + f*x))^(1/2)*(C^2*a^4*d^2 + C^2*b^4*d^2 - 6*C^2*a^2*
b^2*d^2))/f^2)*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2
+ 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4
)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2
) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d
*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (32*(2*C^3*a

```
^3*b^3*d^2 + C^3*a*b^5*d^2 + C^3*a^5*b*d^2))/f^3))*(-(((8*C^2*a^4*c*f^2 + 8
*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c
*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 +
6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 -
 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(c^2*f
^4 + d^2*f^4)))^(1/2)*2i + (2*A*b^2*(c + d*tan(e + f*x))^(1/2))/(d*f) + (2*
B*b^2*(c + d*tan(e + f*x))^(3/2))/(3*d^2*f) + (2*C*b^2*(c + d*tan(e + f*x))
^(5/2))/(5*d^3*f)
```

3.112 $\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx$

Optimal result 1485
Rubi [A] (verified) 1485
Mathematica [A] (verified) 1488
Maple [B] (verified) 1489
Fricas [B] (verification not implemented) .1491
Sympy [F] 1491
Maxima [F] 1492
Giac [F(-1)] 1492
Mupad [B] (verification not implemented) 1492

Optimal result

Integrand size = 45, antiderivative size = 194

$$\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx$$
$$= -\frac{(ia+b)(A-iB-C)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{\sqrt{c-id}f}$$
$$+\frac{(ia-b)(A+iB-C)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{\sqrt{c+id}f}$$
$$-\frac{2(2bcC-3bBd-3aCd)\sqrt{c+d\tan(e+fx)}}{3d^2f}+\frac{2bC\tan(e+fx)\sqrt{c+d\tan(e+fx)}}{3df}$$

```
[Out] -(I*a+b)*(A-I*B-C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/f/(c-I*d)^
(1/2)+(I*a-b)*(A+I*B-C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))/f/(c+
I*d)^(1/2)-2/3*(-3*B*b*d-3*C*a*d+2*C*b*c)*(c+d*tan(f*x+e))^(1/2)/d^2/f+2/3*
b*C*(c+d*tan(f*x+e))^(1/2)*tan(f*x+e)/d/f
```

Rubi [A] (verified)

Time = 0.54 (sec) , antiderivative size = 194, normalized size of antiderivative = 1.00, number of steps used = 9, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.133, Rules used

$= \{3718, 3711, 3620, 3618, 65, 214\}$

$$\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx$$
$$= -\frac{(b+ia)(A-iB-C)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f\sqrt{c-id}} + \frac{(-b+ia)(A+iB-C)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f\sqrt{c+id}} - \frac{2(-3aCd-3bBd+2bcC)\sqrt{c+d\tan(e+fx)}}{3d^2f} + \frac{2bC\tan(e+fx)\sqrt{c+d\tan(e+fx)}}{3df}$$

```
[In] Int[((a + b*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/Sqrt[c +
 d*Tan[e + f*x]],x]
```

```
[Out] -(((I*a + b)*(A - I*B - C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]])
/(Sqrt[c - I*d]*f)) + ((I*a - b)*(A + I*B - C)*ArcTanh[Sqrt[c + d*Tan[e + f
*x]]/Sqrt[c + I*d]])/(Sqrt[c + I*d]*f) - (2*(2*b*c*C - 3*b*B*d - 3*a*C*d)*S
qrt[c + d*Tan[e + f*x]])/(3*d^2*f) + (2*b*C*Tan[e + f*x]*Sqrt[c + d*Tan[e +
 f*x]])/(3*d*f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
```

```
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \frac{2bC\tan(e+fx)\sqrt{c+d\tan(e+fx)}}{3df} \\
&\quad - \frac{2\int \frac{\frac{1}{2}(2bcC-3aAd)-\frac{3}{2}(Ab+aB-bC)d\tan(e+fx)+\frac{1}{2}(2bcC-3bBd-3aCd)\tan^2(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{3d} \\
&= -\frac{2(2bcC-3bBd-3aCd)\sqrt{c+d\tan(e+fx)}}{3d^2f} \\
&\quad + \frac{2bC\tan(e+fx)\sqrt{c+d\tan(e+fx)}}{3df} \\
&\quad - \frac{2\int \frac{\frac{3}{2}(bB-a(A-C))d-\frac{3}{2}(Ab+aB-bC)d\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{3d} \\
&= -\frac{2(2bcC-3bBd-3aCd)\sqrt{c+d\tan(e+fx)}}{3d^2f} \\
&\quad + \frac{2bC\tan(e+fx)\sqrt{c+d\tan(e+fx)}}{3df} \\
&\quad + \frac{1}{2}((a-ib)(A-iB-C))\int \frac{1+i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx \\
&\quad + \frac{1}{2}((a+ib)(A+iB-C))\int \frac{1-i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx
\end{aligned}$$

$$= -\frac{2(2bcC-3bBd-3aCd)\sqrt{c+d\tan(e+fx)}}{3d^2f} + \frac{2bC\tan(e+fx)\sqrt{c+d\tan(e+fx)}}{3df} + \frac{((ia+b)(A-iB-C))\text{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c-idx}}\,dx, x, i\tan(e+fx)\right)}{2f} - \frac{((ia-b)(A+iB-C))\text{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c+idx}}\,dx, x, -i\tan(e+fx)\right)}{2f}$$

$$= -\frac{2(2bcC-3bBd-3aCd)\sqrt{c+d\tan(e+fx)}}{3d^2f} + \frac{2bC\tan(e+fx)\sqrt{c+d\tan(e+fx)}}{3df} - \frac{((a-ib)(A-iB-C))\text{Subst}\left(\int \frac{1}{-1-\frac{ic}{d}+\frac{ix^2}{d}}\,dx, x, \sqrt{c+d\tan(e+fx)}\right)}{df} - \frac{((a+ib)(A+iB-C))\text{Subst}\left(\int \frac{1}{-1+\frac{ic}{d}-\frac{ix^2}{d}}\,dx, x, \sqrt{c+d\tan(e+fx)}\right)}{df}$$

$$= -\frac{(ia+b)(A-iB-C)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{\sqrt{c-id}f} + \frac{(ia-b)(A+iB-C)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{\sqrt{c+id}f} - \frac{2(2bcC-3bBd-3aCd)\sqrt{c+d\tan(e+fx)}}{3d^2f} + \frac{2bC\tan(e+fx)\sqrt{c+d\tan(e+fx)}}{3df}$$

Mathematica [A] (verified)

Time = 1.63 (sec) , antiderivative size = 192, normalized size of antiderivative = 0.99

$$\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx = \frac{2\left(-\frac{3i(a-ib)(A-iB-C)d\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{2\sqrt{c-id}} + \frac{3i(a+ib)(A+iB-C)d\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{2\sqrt{c+id}} + \frac{(-2bcC+3bBd+3aCd)\sqrt{c}}{d}\right.}{3df}$$

```
[In] Integrate[((a + b*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/Sq
rt[c + d*Tan[e + f*x]],x]
```

```
[Out] (2*((((-3*I)/2)*(a - I*b)*(A - I*B - C)*d*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/
Sqrt[c - I*d]])/Sqrt[c - I*d] + (((3*I)/2)*(a + I*b)*(A + I*B - C)*d*ArcTan
h[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]])/Sqrt[c + I*d] + ((-2*b*c*C + 3*b
*B*d + 3*a*C*d)*Sqrt[c + d*Tan[e + f*x]])/d + b*C*Tan[e + f*x]*Sqrt[c + d*T
an[e + f*x]]))/(3*d*f)
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 3852 vs. 2(166) = 332.

Time = 0.14 (sec) , antiderivative size = 3853, normalized size of antiderivative = 19.86

method	result	size
parts	Expression too large to display	3853
derivativedivides	Expression too large to display	4138
default	Expression too large to display	4138

```
[In] int((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)
,x,method=_RETURNVERBOSE)
```

```
[Out] A*a*(-1/4/f/d/(c^2+d^2)*ln(d*tan(f*x+e)+c-(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^
2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c^2-1/4/
f*d/(c^2+d^2)*ln(d*tan(f*x+e)+c-(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2
*c)^(1/2)+(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+1/4/f/d/(c^2+d^2)^
(3/2)*ln(d*tan(f*x+e)+c-(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2
)+(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c^3+1/4/f*d/(c^2+d^2)^(3/2
)*ln(d*tan(f*x+e)+c-(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c
^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c-1/f/d/(c^2+d^2)^(1/2)/(2*(c^
2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)-(2*(c^2+d^2)^(1/2)
+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*c^2-1/f*d/(c^2+d^2)^(1/2)/(2*(c
^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)-(2*(c^2+d^2)^(1/2
)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))+1/f/d/(c^2+d^2)^(3/2)/(2*(c^2+
d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)-(2*(c^2+d^2)^(1/2)+2
*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*c^4+3/f*d/(c^2+d^2)^(3/2)/(2*(c^2
+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)-(2*(c^2+d^2)^(1/2)+
2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*c^2+2/f*d^3/(c^2+d^2)^(3/2)/(2*(
c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)-(2*(c^2+d^2)^(1/
2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))+1/4/f/d/(c^2+d^2)*ln(d*tan(f*
x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2)
)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c^2+1/4/f*d/(c^2+d^2)*ln(d*tan(f*x+e)+c+(c+
d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(2*(c^2+
d^2)^(1/2)+2*c)^(1/2)-1/4/f/d/(c^2+d^2)^(3/2)*ln(d*tan(f*x+e)+c+(c+d*tan(f*
x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/
2)+2*c)^(1/2)*c^3-1/4/f*d/(c^2+d^2)^(3/2)*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e)
)^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)+2
```

*c)^(1/2)*c-1/f/d/(c^2+d^2)^(1/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(
c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c
)^(1/2))*c^2-1/f*d/(c^2+d^2)^(1/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*
(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*
c)^(1/2))+1/f/d/(c^2+d^2)^(3/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+
d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^
(1/2))*c^4+3/f*d/(c^2+d^2)^(3/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c
+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)
^(1/2))*c^2+2/f*d^3/(c^2+d^2)^(3/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2
*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2
*c)^(1/2)))+(A*b+B*a)/f*(1/2/(c^2+d^2)^(1/2)*(1/2*(2*(c^2+d^2)^(1/2)+2*c)^(
1/2)*ln(d*tan(f*x+e)+c-(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)
+(c^2+d^2)^(1/2))+2*((c^2+d^2)^(1/2)-c)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arcta
n((2*(c+d*tan(f*x+e))^(1/2)-(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/
2)-2*c)^(1/2)))+1/2/(c^2+d^2)^(1/2)*(-1/2*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*ln(
d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^
2)^(1/2))+2*((c^2+d^2)^(1/2)-c)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+
d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^
(1/2))))+(B*b+C*a)*(2/d/f*(c+d*tan(f*x+e))^(1/2)+1/4/f/d/(c^2+d^2)*ln(d*tan
(f*x+e)+c-(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1
/2))*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c^2+1/4/f*d/(c^2+d^2)*ln(d*tan(f*x+e)+c-
(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(2*(c
^2+d^2)^(1/2)+2*c)^(1/2)-1/4/f/d/(c^2+d^2)^(3/2)*ln(d*tan(f*x+e)+c-(c+d*tan
(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(2*(c^2+d^2)^
(1/2)+2*c)^(1/2)*c^3-1/4/f*d/(c^2+d^2)^(3/2)*ln(d*tan(f*x+e)+c-(c+d*tan(f*x
+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2
)+2*c)^(1/2)*c+1/f/d/(c^2+d^2)^(1/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((
2*(c+d*tan(f*x+e))^(1/2)-(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-
2*c)^(1/2))*c^2+1/f*d/(c^2+d^2)^(1/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(
(2*(c+d*tan(f*x+e))^(1/2)-(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)
-2*c)^(1/2))-1/f/d/(c^2+d^2)^(3/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*
(c+d*tan(f*x+e))^(1/2)-(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*
c)^(1/2))*c^4-3/f*d/(c^2+d^2)^(3/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2
*(c+d*tan(f*x+e))^(1/2)-(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2
*c)^(1/2))*c^2-2/f*d^3/(c^2+d^2)^(3/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan
((2*(c+d*tan(f*x+e))^(1/2)-(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2
)-2*c)^(1/2))-1/4/f/d/(c^2+d^2)*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2
*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*
c^2-1/4/f*d/(c^2+d^2)*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)
^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+1/4/f/d/(c
^2+d^2)^(3/2)*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2
c)^(1/2)+(c^2+d^2)^(1/2))(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c^3+1/4/f*d/(c^2+d
^2)^(3/2)*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^
(1/2)+(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c+1/f/d/(c^2+d^2)^(1/2
)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^

```
2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*c^2+1/f*d/(c^2+d^2)^(1/
2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d
^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))-1/f/d/(c^2+d^2)^(3/2)/
(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)
^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*c^4-3/f*d/(c^2+d^2)^(3/2)
/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2
)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*c^2-2/f*d^3/(c^2+d^2)^(3
/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+
d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)))+2*C*b/f/d^2*(1/3*(c+
d*tan(f*x+e))^(3/2)-(c+d*tan(f*x+e))^(1/2)*c-d^2*(1/4/(c^2+d^2)^(1/2)*(1/2*
(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*ln(d*tan(f*x+e)+c-(c+d*tan(f*x+e))^(1/2)*(2*(
c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))+2*((c^2+d^2)^(1/2)-c)/(2*(c^2+d^
2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)-(2*(c^2+d^2)^(1/2)+2*c
)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)))+1/4/(c^2+d^2)^(1/2)*(-1/2*(2*(c^2+
d^2)^(1/2)+2*c)^(1/2)*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)
^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))+2*((c^2+d^2)^(1/2)-c)/(2*(c^2+d^2)^(1/2)
-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))
/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)))))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 13473 vs. $2(159) = 318$.

Time = 1.75 (sec) , antiderivative size = 13473, normalized size of antiderivative = 69.45

$$\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx = \text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(1/2),x, algorithm="fricas")
```

```
[Out] Too large to include
```

Sympy [F]

$$\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx = \int \frac{(a+b\tan{(e+fx)})\left(A+B\tan{(e+fx)}+C\tan^2{(e+fx)}\right)}{\sqrt{c+d\tan{(e+fx)}}}\,dx$$

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x+e)
)**(1/2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/sqrt
(c + d*tan(e + f*x)), x)
```

Maxima [F]

$$\int \frac{(a + b\tan(e + fx))\,(A + B\tan(e + fx) + C\tan^2(e + fx))}{\sqrt{c + d\tan(e + fx)}}\,dx = \int \frac{\left(C\tan\left(fx + e\right)^2 + B\tan\left(fx + e\right) + A\right)\left(b\tan\left(fx + e\right) + a\right)}{\sqrt{d\tan\left(fx + e\right) + c}}\,dx$$

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(1/2),x, algorithm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)/sqrt
(d*tan(f*x + e) + c), x)
```

Giac [F(-1)]

Timed out.

$$\int \frac{(a + b\tan(e + fx))\,(A + B\tan(e + fx) + C\tan^2(e + fx))}{\sqrt{c + d\tan(e + fx)}}\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(1/2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [B] (verification not implemented)

Time = 21.65 (sec) , antiderivative size = 16400, normalized size of antiderivative = 84.54

$$\int \frac{(a + b\tan(e + fx))\,(A + B\tan(e + fx) + C\tan^2(e + fx))}{\sqrt{c + d\tan(e + fx)}}\,dx = \text{Too large to display}$$

```
[In] int(((a + b*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d*t
an(e + f*x))^(1/2),x)
```

```
[Out] ((2*B*b*d - 6*C*b*c)/(d^2*f) + (4*C*b*c)/(d^2*f))*(c + d*tan(e + f*x))^(1/2
) - atan(((((8*(4*C*a*d^3*f^2 - 4*A*a*d^3*f^2 + 4*B*a*c*d^2*f^2))/f^3 - 64*
c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*
C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4
 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4
*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4)
)^(1/2) - 4*A^2*a^2*c*f^2 + 4*B^2*a^2*c*f^2 - 4*C^2*a^2*c*f^2 - 8*A*B*a^2*d
*f^2 + 8*A*C*a^2*c*f^2 + 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*
```

((((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*B^2*a^2*c*f^2 - 4*C^2*a^2*c*f^2 - 8*A*B*a^2*d*f^2 + 8*A*C*a^2*c*f^2 + 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^2*d^2 - B^2*a^2*d^2 + C^2*a^2*d^2 - 2*A*C*a^2*d^2))/f^2)*((((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*B^2*a^2*c*f^2 - 4*C^2*a^2*c*f^2 - 8*A*B*a^2*d*f^2 + 8*A*C*a^2*c*f^2 + 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i - (((8*(4*C*a*d^3*f^2 - 4*A*a*d^3*f^2 + 4*B*a*c*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*B^2*a^2*c*f^2 - 4*C^2*a^2*c*f^2 - 8*A*B*a^2*d*f^2 + 8*A*C*a^2*c*f^2 + 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*B^2*a^2*c*f^2 - 4*C^2*a^2*c*f^2 - 8*A*B*a^2*d*f^2 + 8*A*C*a^2*c*f^2 + 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^2*d^2 - B^2*a^2*d^2 + C^2*a^2*d^2 - 2*A*C*a^2*d^2))/f^2)*((((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*B^2*a^2*c*f^2 - 4*C^2*a^2*c*f^2 - 8*A*B*a^2*d*f^2 + 8*A*C*a^2*c*f^2 + 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i)/((((8*(4*C*a*d^3*f^2 - 4*A*a*d^3*f^2 + 4*B*a*c*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*B^2*a^2*c*f^2 - 4*C^2*a^2*c*f^2 - 8*A*B*a^2*d*f^2 + 8*A*C*a^2*c*f^2 + 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*B^2*a^2*c*f^2 - 4*C^2*a^2*c*f^2 - 8*A*B*a^2*d*f^2 + 8*A*C*a^2*c*f^2 + 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^2*d^2 - B^2*a^2*d^2 +

C^2*a^2*d^2 - 2*A*C*a^2*d^2))/f^2)*((((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 +
8*C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^
2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4
- 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a
^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*B^2*a^2*c*f^2 - 4*C^2*a^2*c*f^2 - 8*A*B*a^
2*d*f^2 + 8*A*C*a^2*c*f^2 + 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2
) + (((8*(4*C*a*d^3*f^2 - 4*A*a*d^3*f^2 + 4*B*a*c*d^2*f^2))/f^3 + 64*c*d^2*
(c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^
2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (16
*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C
*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2
) - 4*A^2*a^2*c*f^2 + 4*B^2*a^2*c*f^2 - 4*C^2*a^2*c*f^2 - 8*A*B*a^2*d*f^2 +
8*A*C*a^2*c*f^2 + 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((((8*
A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A
*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 +
B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*
a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*B^2*a^2*c
*f^2 - 4*C^2*a^2*c*f^2 - 8*A*B*a^2*d*f^2 + 8*A*C*a^2*c*f^2 + 8*B*C*a^2*d*f^
2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^
2*d^2 - B^2*a^2*d^2 + C^2*a^2*d^2 - 2*A*C*a^2*d^2))/f^2)*((((8*A^2*a^2*c*f^
2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2
- 16*B*C*a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C
^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*
C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*B^2*a^2*c*f^2 - 4*C^2
*a^2*c*f^2 - 8*A*B*a^2*d*f^2 + 8*A*C*a^2*c*f^2 + 8*B*C*a^2*d*f^2)/(16*(c^2*
f^4 + d^2*f^4)))^(1/2) - (16*(B^3*a^3*d^2 + A^2*B*a^3*d^2 + B*C^2*a^3*d^2 -
2*A*B*C*a^3*d^2))/f^3))*((((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*
c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (16*c
^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a
^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2)
- 4*A^2*a^2*c*f^2 + 4*B^2*a^2*c*f^2 - 4*C^2*a^2*c*f^2 - 8*A*B*a^2*d*f^2 + 8
*A*C*a^2*c*f^2 + 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*2i - atan
(((((8*(4*C*a*d^3*f^2 - 4*A*a*d^3*f^2 + 4*B*a*c*d^2*f^2))/f^3 - 64*c*d^2*(c
+ d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2
*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (16*
c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*
a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2)
+ 4*A^2*a^2*c*f^2 - 4*B^2*a^2*c*f^2 + 4*C^2*a^2*c*f^2 + 8*A*B*a^2*d*f^2 -
8*A*C*a^2*c*f^2 - 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*
A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A
*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 +
B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*
a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*B^2*a^2*c
*f^2 + 4*C^2*a^2*c*f^2 + 8*A*B*a^2*d*f^2 - 8*A*C*a^2*c*f^2 - 8*B*C*a^2*d*f^
2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^

2*d^2 - B^2*a^2*d^2 + C^2*a^2*d^2 - 2*A*C*a^2*d^2))/f^2)*(-(((8*A^2*a^2*c*f
^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^
2 - 16*B*C*a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 +
C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2
*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*B^2*a^2*c*f^2 + 4*C^
2*a^2*c*f^2 + 8*A*B*a^2*d*f^2 - 8*A*C*a^2*c*f^2 - 8*B*C*a^2*d*f^2)/(16*(c^2
*f^4 + d^2*f^4)))^(1/2)*1i - (((8*(4*C*a*d^3*f^2 - 4*A*a*d^3*f^2 + 4*B*a*c*
d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^2*c*f^2 -
8*B^2*a^2*c*f^2 + 8*C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 1
6*B*C*a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a
^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*
a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*B^2*a^2*c*f^2 + 4*C^2*a^2
*c*f^2 + 8*A*B*a^2*d*f^2 - 8*A*C*a^2*c*f^2 - 8*B*C*a^2*d*f^2)/(16*(c^2*f^4
+ d^2*f^4)))^(1/2))*(-(((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*c*f^
2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (16*c^2*f
^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 +
2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*
A^2*a^2*c*f^2 - 4*B^2*a^2*c*f^2 + 4*C^2*a^2*c*f^2 + 8*A*B*a^2*d*f^2 - 8*A*C
*a^2*c*f^2 - 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (16*(c + d*
tan(e + f*x))^(1/2)*(A^2*a^2*d^2 - B^2*a^2*d^2 + C^2*a^2*d^2 - 2*A*C*a^2*d^
2))/f^2)*(-(((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*c*f^2 + 16*A*B*
a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2
f^4)(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*
a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c*f
^2 - 4*B^2*a^2*c*f^2 + 4*C^2*a^2*c*f^2 + 8*A*B*a^2*d*f^2 - 8*A*C*a^2*c*f^2
- 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i)/((((8*(4*C*a*d^3*f^2
- 4*A*a*d^3*f^2 + 4*B*a*c*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1
/2)*(-(((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*c*f^2 + 16*A*B*a^2*d
*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)
*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 +
6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c*f^2 -
4*B^2*a^2*c*f^2 + 4*C^2*a^2*c*f^2 + 8*A*B*a^2*d*f^2 - 8*A*C*a^2*c*f^2 - 8*B
*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*A^2*a^2*c*f^2 - 8*B^2
*a^2*c*f^2 + 8*C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C
*a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 -
4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 -
4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*B^2*a^2*c*f^2 + 4*C^2*a^2*c*f^
2 + 8*A*B*a^2*d*f^2 - 8*A*C*a^2*c*f^2 - 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2
f^4)))^(1/2) - (16(c + d*tan(e + f*x))^(1/2)*(A^2*a^2*d^2 - B^2*a^2*d^2 +
C^2*a^2*d^2 - 2*A*C*a^2*d^2))/f^2)*(-(((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2
+ 8*C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)
^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4
- 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*
a^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*B^2*a^2*c*f^2 + 4*C^2*a^2*c*f^2 + 8*A*B*a
^2*d*f^2 - 8*A*C*a^2*c*f^2 - 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/

2) + (((8*(4*C*a*d^3*f^2 - 4*A*a*d^3*f^2 + 4*B*a*c*d^2*f^2))/f^3 + 64*c*d^2
*(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*
a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (
16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3
*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1
/2) + 4*A^2*a^2*c*f^2 - 4*B^2*a^2*c*f^2 + 4*C^2*a^2*c*f^2 + 8*A*B*a^2*d*f^2
 - 8*A*C*a^2*c*f^2 - 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-((
(8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 1
6*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^
4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C
^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*B^2*a^
2*c*f^2 + 4*C^2*a^2*c*f^2 + 8*A*B*a^2*d*f^2 - 8*A*C*a^2*c*f^2 - 8*B*C*a^2*d
f^2)/(16(c^2*f^4 + d^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(A^2
*a^2*d^2 - B^2*a^2*d^2 + C^2*a^2*d^2 - 2*A*C*a^2*d^2))/f^2)*(-(((8*A^2*a^2*
c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c
*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4
 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*
B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*B^2*a^2*c*f^2 + 4
*C^2*a^2*c*f^2 + 8*A*B*a^2*d*f^2 - 8*A*C*a^2*c*f^2 - 8*B*C*a^2*d*f^2)/(16*(
c^2*f^4 + d^2*f^4)))^(1/2) - (16*(B^3*a^3*d^2 + A^2*B*a^3*d^2 + B*C^2*a^3*d
^2 - 2*A*B*C*a^3*d^2))/f^3))*(-(((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2
*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 -
(16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^
3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(
1/2) + 4*A^2*a^2*c*f^2 - 4*B^2*a^2*c*f^2 + 4*C^2*a^2*c*f^2 + 8*A*B*a^2*d*f^
2 - 8*A*C*a^2*c*f^2 - 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*2i -
 atan(((((8*(4*B*b*d^3*f^2 + 4*A*b*c*d^2*f^2 - 4*C*b*c*d^2*f^2))/f^3 - 64*c
d^2(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*
C^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2 - 16*A*C*b^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4
 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4
*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4)
)^(1/2) - 4*A^2*b^2*c*f^2 + 4*B^2*b^2*c*f^2 - 4*C^2*b^2*c*f^2 - 8*A*B*b^2*d
*f^2 + 8*A*C*b^2*c*f^2 + 8*B*C*b^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*
(-(((8*A^2*b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2
 - 16*A*C*b^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^
4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A
^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c*f^2 + 4*B^
2*b^2*c*f^2 - 4*C^2*b^2*c*f^2 - 8*A*B*b^2*d*f^2 + 8*A*C*b^2*c*f^2 + 8*B*C*b
^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*
(A^2*b^2*d^2 - B^2*b^2*d^2 + C^2*b^2*d^2 - 2*A*C*b^2*d^2))/f^2)*(-(((8*A^2*
b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2 - 16*A*C*b
^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*b^4 + B^4
*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4
+ 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c*f^2 + 4*B^2*b^2*c*f^2
 - 4*C^2*b^2*c*f^2 - 8*A*B*b^2*d*f^2 + 8*A*C*b^2*c*f^2 + 8*B*C*b^2*d*f^2)/(

16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i - (((8*(4*B*b*d^3*f^2 + 4*A*b*c*d^2*f^2 - 4*C*b*c*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2 - 16*A*C*b^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c*f^2 + 4*B^2*b^2*c*f^2 - 4*C^2*b^2*c*f^2 - 8*A*B*b^2*d*f^2 + 8*A*C*b^2*c*f^2 + 8*B*C*b^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*A^2*b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2 - 16*A*C*b^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c*f^2 + 4*B^2*b^2*c*f^2 - 4*C^2*b^2*c*f^2 - 8*A*B*b^2*d*f^2 + 8*A*C*b^2*c*f^2 + 8*B*C*b^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(A^2*b^2*d^2 - B^2*b^2*d^2 + C^2*b^2*d^2 - 2*A*C*b^2*d^2))/f^2)*(-(((8*A^2*b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2 - 16*A*C*b^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c*f^2 + 4*B^2*b^2*c*f^2 - 4*C^2*b^2*c*f^2 - 8*A*B*b^2*d*f^2 + 8*A*C*b^2*c*f^2 + 8*B*C*b^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i)/((((8*(4*B*b*d^3*f^2 + 4*A*b*c*d^2*f^2 - 4*C*b*c*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2 - 16*A*C*b^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c*f^2 + 4*B^2*b^2*c*f^2 - 4*C^2*b^2*c*f^2 - 8*A*B*b^2*d*f^2 + 8*A*C*b^2*c*f^2 + 8*B*C*b^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*A^2*b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2 - 16*A*C*b^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c*f^2 + 4*B^2*b^2*c*f^2 - 4*C^2*b^2*c*f^2 - 8*A*B*b^2*d*f^2 + 8*A*C*b^2*c*f^2 + 8*B*C*b^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(A^2*b^2*d^2 - B^2*b^2*d^2 + C^2*b^2*d^2 - 2*A*C*b^2*d^2))/f^2)*(-(((8*A^2*b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2 - 16*A*C*b^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c*f^2 + 4*B^2*b^2*c*f^2 - 4*C^2*b^2*c*f^2 - 8*A*B*b^2*d*f^2 + 8*A*C*b^2*c*f^2 + 8*B*C*b^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(A^3*b^3*d^2 - C^3*b^3*d^2 + A*B^2*b^3*d^2 + 3*A*C^2*b^3*d^2 - 3*A^2*C*b^3*d^2 - B^2*C*b^3*d^2))/f^3 + (((8*(4*B*b*d^3*f^2 + 4*A*b*c*d^2*f^2 - 4*C*b*c*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2 - 16*A*C*b^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2

```
*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c*f^2 + 4*B^2*
b^2*c*f^2 - 4*C^2*b^2*c*f^2 - 8*A*B*b^2*d*f^2 + 8*A*C*b^2*c*f^2 + 8*B*C*b^2
*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*A^2*b^2*c*f^2 - 8*B^2*b^2*c
*f^2 + 8*C^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2 - 16*A*C*b^2*c*f^2 - 16*B*C*b^2*d
*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^
3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B
^2*C*b^4))^(1/2) - 4*A^2*b^2*c*f^2 + 4*B^2*b^2*c*f^2 - 4*C^2*b^2*c*f^2 - 8*
A*B*b^2*d*f^2 + 8*A*C*b^2*c*f^2 + 8*B*C*b^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4))
)^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(A^2*b^2*d^2 - B^2*b^2*d^2 + C^2*b
^2*d^2 - 2*A*C*b^2*d^2))/f^2)*(-(((8*A^2*b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C^
2*b^2*c*f^2 + 16*A*B*b^2*d*f^2 - 16*A*C*b^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4 -
 (16*c^2*f^4 + 16*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A
^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^
(1/2) - 4*A^2*b^2*c*f^2 + 4*B^2*b^2*c*f^2 - 4*C^2*b^2*c*f^2 - 8*A*B*b^2*d*f
^2 + 8*A*C*b^2*c*f^2 + 8*B*C*b^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)))*(
-(((8*A^2*b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2
 - 16*A*C*b^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4
*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^
2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c*f^2 + 4*B^2
*b^2*c*f^2 - 4*C^2*b^2*c*f^2 - 8*A*B*b^2*d*f^2 + 8*A*C*b^2*c*f^2 + 8*B*C*b^
2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*2i - atan(((((8*(4*B*b*d^3*f^2 + 4
*A*b*c*d^2*f^2 - 4*C*b*c*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2
)*((((8*A^2*b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C^2*b^2*c*f^2 + 16*A*B*b^2*d*f^
2 - 16*A*C*b^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A
^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*
A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c*f^2 - 4*B
^2*b^2*c*f^2 + 4*C^2*b^2*c*f^2 + 8*A*B*b^2*d*f^2 - 8*A*C*b^2*c*f^2 - 8*B*C*
b^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((((8*A^2*b^2*c*f^2 - 8*B^2*b^2
*c*f^2 + 8*C^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2 - 16*A*C*b^2*c*f^2 - 16*B*C*b^2
*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*
C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A
*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c*f^2 - 4*B^2*b^2*c*f^2 + 4*C^2*b^2*c*f^2 +
8*A*B*b^2*d*f^2 - 8*A*C*b^2*c*f^2 - 8*B*C*b^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4
)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(A^2*b^2*d^2 - B^2*b^2*d^2 + C^2
*b^2*d^2 - 2*A*C*b^2*d^2))/f^2)*((((8*A^2*b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C
^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2 - 16*A*C*b^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4
- (16*c^2*f^4 + 16*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*
A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))
^(1/2) + 4*A^2*b^2*c*f^2 - 4*B^2*b^2*c*f^2 + 4*C^2*b^2*c*f^2 + 8*A*B*b^2*d*
f^2 - 8*A*C*b^2*c*f^2 - 8*B*C*b^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i
 - (((8*(4*B*b*d^3*f^2 + 4*A*b*c*d^2*f^2 - 4*C*b*c*d^2*f^2))/f^3 + 64*c*d^2
*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C^2*b
^2*c*f^2 + 16*A*B*b^2*d*f^2 - 16*A*C*b^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4 - (1
6*c^2*f^4 + 16*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*
C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/
```

2) + 4*A^2*b^2*c*f^2 - 4*B^2*b^2*c*f^2 + 4*C^2*b^2*c*f^2 + 8*A*B*b^2*d*f^2
- 8*A*C*b^2*c*f^2 - 8*B*C*b^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((((8
*A^2*b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2 - 16*
A*C*b^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*b^4
+ B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2
*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c*f^2 - 4*B^2*b^2*
c*f^2 + 4*C^2*b^2*c*f^2 + 8*A*B*b^2*d*f^2 - 8*A*C*b^2*c*f^2 - 8*B*C*b^2*d*f
^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(A^2*b
^2*d^2 - B^2*b^2*d^2 + C^2*b^2*d^2 - 2*A*C*b^2*d^2))/f^2)*((((8*A^2*b^2*c*f
^2 - 8*B^2*b^2*c*f^2 + 8*C^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2 - 16*A*C*b^2*c*f^
2 - 16*B*C*b^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*b^4 + B^4*b^4 +
C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2
*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c*f^2 - 4*B^2*b^2*c*f^2 + 4*C^
2*b^2*c*f^2 + 8*A*B*b^2*d*f^2 - 8*A*C*b^2*c*f^2 - 8*B*C*b^2*d*f^2)/(16*(c^2
*f^4 + d^2*f^4)))^(1/2)*1i)/((((8*(4*B*b*d^3*f^2 + 4*A*b*c*d^2*f^2 - 4*C*b*
c*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*b^2*c*f^2 -
8*B^2*b^2*c*f^2 + 8*C^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2 - 16*A*C*b^2*c*f^2 -
16*B*C*b^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*
b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2
*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c*f^2 - 4*B^2*b^2*c*f^2 + 4*C^2*b^
2*c*f^2 + 8*A*B*b^2*d*f^2 - 8*A*C*b^2*c*f^2 - 8*B*C*b^2*d*f^2)/(16*(c^2*f^4
+ d^2*f^4)))^(1/2))*((((8*A^2*b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C^2*b^2*c*f^
2 + 16*A*B*b^2*d*f^2 - 16*A*C*b^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4 - (16*c^2*f
^4 + 16*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 +
2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*
A^2*b^2*c*f^2 - 4*B^2*b^2*c*f^2 + 4*C^2*b^2*c*f^2 + 8*A*B*b^2*d*f^2 - 8*A*C
*b^2*c*f^2 - 8*B*C*b^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (16*(c + d*
tan(e + f*x))^(1/2)*(A^2*b^2*d^2 - B^2*b^2*d^2 + C^2*b^2*d^2 - 2*A*C*b^2*d^
2))/f^2)*((((8*A^2*b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C^2*b^2*c*f^2 + 16*A*B*b
^2*d*f^2 - 16*A*C*b^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*
f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b
^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c*f^
2 - 4*B^2*b^2*c*f^2 + 4*C^2*b^2*c*f^2 + 8*A*B*b^2*d*f^2 - 8*A*C*b^2*c*f^2 -
8*B*C*b^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(A^3*b^3*d^2 - C^3*
b^3*d^2 + A*B^2*b^3*d^2 + 3*A*C^2*b^3*d^2 - 3*A^2*C*b^3*d^2 - B^2*C*b^3*d^2
))/f^3 + (((8*(4*B*b*d^3*f^2 + 4*A*b*c*d^2*f^2 - 4*C*b*c*d^2*f^2))/f^3 + 64
*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8
*C^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2 - 16*A*C*b^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/
4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 -
4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4
))^(1/2) + 4*A^2*b^2*c*f^2 - 4*B^2*b^2*c*f^2 + 4*C^2*b^2*c*f^2 + 8*A*B*b^2*
d*f^2 - 8*A*C*b^2*c*f^2 - 8*B*C*b^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))
*((((8*A^2*b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2
- 16*A*C*b^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^
4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A

```
^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c*f^2 - 4*B^
2*b^2*c*f^2 + 4*C^2*b^2*c*f^2 + 8*A*B*b^2*d*f^2 - 8*A*C*b^2*c*f^2 - 8*B*C*b
^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*
(A^2*b^2*d^2 - B^2*b^2*d^2 + C^2*b^2*d^2 - 2*A*C*b^2*d^2))/f^2)*((((8*A^2*b
^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2 - 16*A*C*b^
2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*b^4 + B^4*
b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 +
 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c*f^2 - 4*B^2*b^2*c*f^2
+ 4*C^2*b^2*c*f^2 + 8*A*B*b^2*d*f^2 - 8*A*C*b^2*c*f^2 - 8*B*C*b^2*d*f^2)/(1
6*(c^2*f^4 + d^2*f^4)))^(1/2)))*((((8*A^2*b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C
^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2 - 16*A*C*b^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4
- (16*c^2*f^4 + 16*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*
A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))
^(1/2) + 4*A^2*b^2*c*f^2 - 4*B^2*b^2*c*f^2 + 4*C^2*b^2*c*f^2 + 8*A*B*b^2*d*
f^2 - 8*A*C*b^2*c*f^2 - 8*B*C*b^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*2i
 + (2*C*a*(c + d*tan(e + f*x))^(1/2))/(d*f) + (2*C*b*(c + d*tan(e + f*x))^(
3/2))/(3*d^2*f)
```

3.113 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx$

Optimal result 1501
Rubi [A] (verified) 1501
Mathematica [A] (verified) 1503
Maple [B] (verified) 1504
Fricas [B] (verification not implemented) 1506
Sympy [F] 1507
Maxima [F] 1508
Giac [F(-1)] 1508
Mupad [B] (verification not implemented) 1508

Optimal result

Integrand size = 35, antiderivative size = 133

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx = -\frac{(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{\sqrt{c-id}f} - \frac{(B-i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{\sqrt{c+id}f} + \frac{2C\sqrt{c+d\tan(e+fx)}}{df}$$

```
[Out] -(I*A+B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/f/(c-I*d)^(1/2)-
(B-I*(A-C))*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))/f/(c+I*d)^(1/2)+2
*C*(c+d*tan(f*x+e))^(1/2)/d/f
```

Rubi [A] (verified)

Time = 0.24 (sec) , antiderivative size = 133, normalized size of antiderivative = 1.00, number of steps used = 8, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.143, Rules used = {3711, 3620, 3618, 65, 214}

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx = -\frac{(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f\sqrt{c-id}} - \frac{(B-i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f\sqrt{c+id}} + \frac{2C\sqrt{c+d\tan(e+fx)}}{df}$$

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/Sqrt[c + d*Tan[e + f*x]],x]
```

```
[Out] -(((I*A + B - I*C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]])/(Sqrt[c
 - I*d]*f)) - ((B - I*(A - C))*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*
d]])/(Sqrt[c + I*d]*f) + (2*C*Sqrt[c + d*Tan[e + f*x]])/(d*f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rubi steps

$$\text{integral} = \frac{2C\sqrt{c+d\tan(e+fx)}}{df} + \int \frac{A-C+B\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx$$

$$= \frac{2C\sqrt{c+d\tan(e+fx)}}{df} + \frac{1}{2}(A-iB-C)\int \frac{1+i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx + \frac{1}{2}(A+iB-C)\int \frac{1-i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx$$

$$= \frac{2C\sqrt{c+d\tan(e+fx)}}{df} + \frac{(iA+B-iC)\mathrm{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c-idx}}\,dx, x, i\tan(e+fx)\right)}{2f} + \frac{(i(-A-iB+C))\mathrm{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c+idx}}\,dx, x, -i\tan(e+fx)\right)}{2f}$$

$$= \frac{2C\sqrt{c+d\tan(e+fx)}}{df} - \frac{(A-iB-C)\mathrm{Subst}\left(\int \frac{1}{-1-\frac{ic}{d}+\frac{ix^2}{d}}\,dx, x, \sqrt{c+d\tan(e+fx)}\right)}{df} - \frac{(A+iB-C)\mathrm{Subst}\left(\int \frac{1}{-1+\frac{ic}{d}-\frac{ix^2}{d}}\,dx, x, \sqrt{c+d\tan(e+fx)}\right)}{df}$$

$$= -\frac{(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{\sqrt{c-id}f} - \frac{(B-i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{\sqrt{c+id}f} + \frac{2C\sqrt{c+d\tan(e+fx)}}{df}$$

Mathematica [A] (verified)

Time = 0.24 (sec) , antiderivative size = 129, normalized size of antiderivative = 0.97

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx = \frac{-\frac{i(A-iB-C)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{\sqrt{c-id}} + \frac{i(A+iB-C)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{\sqrt{c+id}} + \frac{2C\sqrt{c+d\tan(e+fx)}}{d}}{f}$$

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/Sqrt[c + d*Tan[e + f*x]], x]
```

```
[Out] (((-I)*(A - I*B - C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]])/Sqrt[
c - I*d] + (I*(A + I*B - C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]]
)/Sqrt[c + I*d] + (2*C*Sqrt[c + d*Tan[e + f*x]])/d)/f
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 3462 vs. $2(112) = 224$.

Time = 0.12 (sec) , antiderivative size = 3463, normalized size of antiderivative = 26.04

method	result	size
parts	Expression too large to display	3463
derivativedivides	Expression too large to display	5570
default	Expression too large to display	5570

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2),x,method=_RETURN
VERBOSE)
```

```
[Out] A*(-1/4/f/d/(c^2+d^2)*ln(d*tan(f*x+e)+c-(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)
^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c^2-1/4/f*
d/(c^2+d^2)*ln(d*tan(f*x+e)+c-(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c
)^(1/2)+(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+1/4/f/d/(c^2+d^2)^(3
/2)*ln(d*tan(f*x+e)+c-(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+
(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c^3+1/4/f*d/(c^2+d^2)^(3/2)*
ln(d*tan(f*x+e)+c-(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2
+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c-1/f/d/(c^2+d^2)^(1/2)/(2*(c^2+
d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)-(2*(c^2+d^2)^(1/2)+2
*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*c^2-1/f*d/(c^2+d^2)^(1/2)/(2*(c^2
+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)-(2*(c^2+d^2)^(1/2)+
2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))+1/f/d/(c^2+d^2)^(3/2)/(2*(c^2+d^
2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)-(2*(c^2+d^2)^(1/2)+2*c
)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*c^4+3/f*d/(c^2+d^2)^(3/2)/(2*(c^2+d
^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)-(2*(c^2+d^2)^(1/2)+2*
c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*c^2+2/f*d^3/(c^2+d^2)^(3/2)/(2*(c^
2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)-(2*(c^2+d^2)^(1/2)
+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))+1/4/f/d/(c^2+d^2)*ln(d*tan(f*x+
e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*
(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c^2+1/4/f*d/(c^2+d^2)*ln(d*tan(f*x+e)+c+(c+d*
tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(2*(c^2+d^
2)^(1/2)+2*c)^(1/2)-1/4/f/d/(c^2+d^2)^(3/2)*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+
e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)
+2*c)^(1/2)*c^3-1/4/f*d/(c^2+d^2)^(3/2)*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^
(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)+2*c
)^(1/2)*c-1/f/d/(c^2+d^2)^(1/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+
d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^
(1/2))*c^2-1/f*d/(c^2+d^2)^(1/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c
+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)
^(1/2))+1/f/d/(c^2+d^2)^(3/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*
tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1
/2))*c^4+3/f*d/(c^2+d^2)^(3/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d
```

*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(
1/2))*c^2+2/f*d^3/(c^2+d^2)^(3/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(
c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c
)^(1/2)))+B/f*(1/2/(c^2+d^2)^(1/2)*(1/2*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*ln(d*
tan(f*x+e)+c-(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)
^(1/2))+2*((c^2+d^2)^(1/2)-c)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*
tan(f*x+e))^(1/2)-(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1
/2)))+1/2/(c^2+d^2)^(1/2)*(-1/2*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*ln(d*tan(f*x+
e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))+
2*((c^2+d^2)^(1/2)-c)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+
e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))))+C
(2/d/f(c+d*tan(f*x+e))^(1/2)+1/4/f/d/(c^2+d^2)*ln(d*tan(f*x+e)+c-(c+d*tan
(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(2*(c^2+d^2)^
(1/2)+2*c)^(1/2)*c^2+1/4/f*d/(c^2+d^2)*ln(d*tan(f*x+e)+c-(c+d*tan(f*x+e))^(
1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)+2*c)
^(1/2)-1/4/f/d/(c^2+d^2)^(3/2)*ln(d*tan(f*x+e)+c-(c+d*tan(f*x+e))^(1/2)*(2*
(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c
^3-1/4/f*d/(c^2+d^2)^(3/2)*ln(d*tan(f*x+e)+c-(c+d*tan(f*x+e))^(1/2)*(2*(c^2
+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c+1/f
/d/(c^2+d^2)^(1/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))
^(1/2)-(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*c^2+1/
f*d/(c^2+d^2)^(1/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e)
)^(1/2)-(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))-1/f/d
/(c^2+d^2)^(3/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(
1/2)-(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*c^4-3/f*
d/(c^2+d^2)^(3/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^
(1/2)-(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*c^2-2/f
d^3/(c^2+d^2)^(3/2)/(2(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e
))^(1/2)-(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))-1/4/
f/d/(c^2+d^2)*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2
c)^(1/2)+(c^2+d^2)^(1/2))(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c^2-1/4/f*d/(c^2+d
^2)*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+
(c^2+d^2)^(1/2))*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+1/4/f/d/(c^2+d^2)^(3/2)*ln(d
*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2
)^(1/2))*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c^3+1/4/f*d/(c^2+d^2)^(3/2)*ln(d*tan
(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1
/2))*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c+1/f/d/(c^2+d^2)^(1/2)/(2*(c^2+d^2)^(1/
2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2
))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*c^2+1/f*d/(c^2+d^2)^(1/2)/(2*(c^2+d^2)^(1
/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/
2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))-1/f/d/(c^2+d^2)^(3/2)/(2*(c^2+d^2)^(1/2)
-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))
/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*c^4-3/f*d/(c^2+d^2)^(3/2)/(2*(c^2+d^2)^(1/2
)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2)
)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*c^2-2/f*d^3/(c^2+d^2)^(3/2)/(2*(c^2+d^2)^(

```
1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1
/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)))
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 3194 vs. $2(105) = 210$.

Time = 0.39 (sec) , antiderivative size = 3194, normalized size of antiderivative = 24.02

$$\int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{\sqrt{c + d\tan(e + fx)}}\,dx = \text{Too large to display}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2),x, algorit
hm="fricas")
```

```
[Out] 1/2*(d*f*sqrt(-((c^2 + d^2)*f^2*sqrt(-(4*(A^2*B^2 - 2*A*B^2*C + B^2*C^2)*c^
2 - 4*(A^3*B - A*B^3 + 3*A*B*C^2 - B*C^3 - (3*A^2*B - B^3)*C)*c*d + (A^4 -
2*A^2*B^2 + B^4 - 4*A*C^3 + C^4 + 2*(3*A^2 - B^2)*C^2 - 4*(A^3 - A*B^2)*C)*
d^2)/((c^4 + 2*c^2*d^2 + d^4)*f^4)) + (A^2 - B^2 - 2*A*C + C^2)*c + 2*(A*B
- B*C)*d)/((c^2 + d^2)*f^2))*log(-(2*(A^3*B + A*B^3 + 3*A*B*C^2 - B*C^3 - (
3*A^2*B + B^3)*C)*c - (A^4 - B^4 - 4*A^3*C + 6*A^2*C^2 - 4*A*C^3 + C^4)*d)*
sqrt(d*tan(f*x + e) + c) + (((A - C)*c^3 + B*c^2*d + (A - C)*c*d^2 + B*d^3)
*f^3*sqrt(-(4*(A^2*B^2 - 2*A*B^2*C + B^2*C^2)*c^2 - 4*(A^3*B - A*B^3 + 3*A*
B*C^2 - B*C^3 - (3*A^2*B - B^3)*C)*c*d + (A^4 - 2*A^2*B^2 + B^4 - 4*A*C^3 +
 C^4 + 2*(3*A^2 - B^2)*C^2 - 4*(A^3 - A*B^2)*C)*d^2)/((c^4 + 2*c^2*d^2 + d^
4)*f^4)) + (2*(A*B^2 - B^2*C)*c^2 - (3*A^2*B - B^3 - 6*A*B*C + 3*B*C^2)*c*d
 + (A^3 - A*B^2 + 3*A*C^2 - C^3 - (3*A^2 - B^2)*C)*d^2)*f)*sqrt(-((c^2 + d^
2)*f^2*sqrt(-(4*(A^2*B^2 - 2*A*B^2*C + B^2*C^2)*c^2 - 4*(A^3*B - A*B^3 + 3*
A*B*C^2 - B*C^3 - (3*A^2*B - B^3)*C)*c*d + (A^4 - 2*A^2*B^2 + B^4 - 4*A*C^3
 + C^4 + 2*(3*A^2 - B^2)*C^2 - 4*(A^3 - A*B^2)*C)*d^2)/((c^4 + 2*c^2*d^2 +
d^4)*f^4)) + (A^2 - B^2 - 2*A*C + C^2)*c + 2*(A*B - B*C)*d)/((c^2 + d^2)*f^
2))) - d*f*sqrt(-((c^2 + d^2)*f^2*sqrt(-(4*(A^2*B^2 - 2*A*B^2*C + B^2*C^2)*
c^2 - 4*(A^3*B - A*B^3 + 3*A*B*C^2 - B*C^3 - (3*A^2*B - B^3)*C)*c*d + (A^4
- 2*A^2*B^2 + B^4 - 4*A*C^3 + C^4 + 2*(3*A^2 - B^2)*C^2 - 4*(A^3 - A*B^2)*C
)*d^2)/((c^4 + 2*c^2*d^2 + d^4)*f^4)) + (A^2 - B^2 - 2*A*C + C^2)*c + 2*(A*
B - B*C)*d)/((c^2 + d^2)*f^2))*log(-(2*(A^3*B + A*B^3 + 3*A*B*C^2 - B*C^3 -
 (3*A^2*B + B^3)*C)*c - (A^4 - B^4 - 4*A^3*C + 6*A^2*C^2 - 4*A*C^3 + C^4)*d
)*sqrt(d*tan(f*x + e) + c) - (((A - C)*c^3 + B*c^2*d + (A - C)*c*d^2 + B*d^
3)*f^3*sqrt(-(4*(A^2*B^2 - 2*A*B^2*C + B^2*C^2)*c^2 - 4*(A^3*B - A*B^3 + 3*
A*B*C^2 - B*C^3 - (3*A^2*B - B^3)*C)*c*d + (A^4 - 2*A^2*B^2 + B^4 - 4*A*C^3
 + C^4 + 2*(3*A^2 - B^2)*C^2 - 4*(A^3 - A*B^2)*C)*d^2)/((c^4 + 2*c^2*d^2 +
d^4)*f^4)) + (2*(A*B^2 - B^2*C)*c^2 - (3*A^2*B - B^3 - 6*A*B*C + 3*B*C^2)*c
*d + (A^3 - A*B^2 + 3*A*C^2 - C^3 - (3*A^2 - B^2)*C)*d^2)*f)*sqrt(-((c^2 +
d^2)*f^2*sqrt(-(4*(A^2*B^2 - 2*A*B^2*C + B^2*C^2)*c^2 - 4*(A^3*B - A*B^3 +
3*A*B*C^2 - B*C^3 - (3*A^2*B - B^3)*C)*c*d + (A^4 - 2*A^2*B^2 + B^4 - 4*A*C
```

```
^3 + C^4 + 2*(3*A^2 - B^2)*C^2 - 4*(A^3 - A*B^2)*C)*d^2)/((c^4 + 2*c^2*d^2
+ d^4)*f^4)) + (A^2 - B^2 - 2*A*C + C^2)*c + 2*(A*B - B*C)*d)/((c^2 + d^2)*
f^2))) - d*f*sqrt(((c^2 + d^2)*f^2*sqrt(-(4*(A^2*B^2 - 2*A*B^2*C + B^2*C^2)
*c^2 - 4*(A^3*B - A*B^3 + 3*A*B*C^2 - B*C^3 - (3*A^2*B - B^3)*C)*c*d + (A^4
 - 2*A^2*B^2 + B^4 - 4*A*C^3 + C^4 + 2*(3*A^2 - B^2)*C^2 - 4*(A^3 - A*B^2)*
C)*d^2)/((c^4 + 2*c^2*d^2 + d^4)*f^4)) - (A^2 - B^2 - 2*A*C + C^2)*c - 2*(A
*B - B*C)*d)/((c^2 + d^2)*f^2))*log(-(2*(A^3*B + A*B^3 + 3*A*B*C^2 - B*C^3
- (3*A^2*B + B^3)*C)*c - (A^4 - B^4 - 4*A^3*C + 6*A^2*C^2 - 4*A*C^3 + C^4)*
d)*sqrt(d*tan(f*x + e) + c) + (((A - C)*c^3 + B*c^2*d + (A - C)*c*d^2 + B*d
^3)*f^3*sqrt(-(4*(A^2*B^2 - 2*A*B^2*C + B^2*C^2)*c^2 - 4*(A^3*B - A*B^3 + 3
*A*B*C^2 - B*C^3 - (3*A^2*B - B^3)*C)*c*d + (A^4 - 2*A^2*B^2 + B^4 - 4*A*C^
3 + C^4 + 2*(3*A^2 - B^2)*C^2 - 4*(A^3 - A*B^2)*C)*d^2)/((c^4 + 2*c^2*d^2 +
 d^4)*f^4)) - (2*(A*B^2 - B^2*C)*c^2 - (3*A^2*B - B^3 - 6*A*B*C + 3*B*C^2)*
c*d + (A^3 - A*B^2 + 3*A*C^2 - C^3 - (3*A^2 - B^2)*C)*d^2)*f)*sqrt(((c^2 +
d^2)*f^2*sqrt(-(4*(A^2*B^2 - 2*A*B^2*C + B^2*C^2)*c^2 - 4*(A^3*B - A*B^3 +
3*A*B*C^2 - B*C^3 - (3*A^2*B - B^3)*C)*c*d + (A^4 - 2*A^2*B^2 + B^4 - 4*A*C
^3 + C^4 + 2*(3*A^2 - B^2)*C^2 - 4*(A^3 - A*B^2)*C)*d^2)/((c^4 + 2*c^2*d^2
+ d^4)*f^4)) - (A^2 - B^2 - 2*A*C + C^2)*c - 2*(A*B - B*C)*d)/((c^2 + d^2)*
f^2))) + d*f*sqrt(((c^2 + d^2)*f^2*sqrt(-(4*(A^2*B^2 - 2*A*B^2*C + B^2*C^2)
*c^2 - 4*(A^3*B - A*B^3 + 3*A*B*C^2 - B*C^3 - (3*A^2*B - B^3)*C)*c*d + (A^4
 - 2*A^2*B^2 + B^4 - 4*A*C^3 + C^4 + 2*(3*A^2 - B^2)*C^2 - 4*(A^3 - A*B^2)*
C)*d^2)/((c^4 + 2*c^2*d^2 + d^4)*f^4)) - (A^2 - B^2 - 2*A*C + C^2)*c - 2*(A
*B - B*C)*d)/((c^2 + d^2)*f^2))*log(-(2*(A^3*B + A*B^3 + 3*A*B*C^2 - B*C^3
- (3*A^2*B + B^3)*C)*c - (A^4 - B^4 - 4*A^3*C + 6*A^2*C^2 - 4*A*C^3 + C^4)*
d)*sqrt(d*tan(f*x + e) + c) - (((A - C)*c^3 + B*c^2*d + (A - C)*c*d^2 + B*d
^3)*f^3*sqrt(-(4*(A^2*B^2 - 2*A*B^2*C + B^2*C^2)*c^2 - 4*(A^3*B - A*B^3 + 3
*A*B*C^2 - B*C^3 - (3*A^2*B - B^3)*C)*c*d + (A^4 - 2*A^2*B^2 + B^4 - 4*A*C^
3 + C^4 + 2*(3*A^2 - B^2)*C^2 - 4*(A^3 - A*B^2)*C)*d^2)/((c^4 + 2*c^2*d^2 +
 d^4)*f^4)) - (2*(A*B^2 - B^2*C)*c^2 - (3*A^2*B - B^3 - 6*A*B*C + 3*B*C^2)*
c*d + (A^3 - A*B^2 + 3*A*C^2 - C^3 - (3*A^2 - B^2)*C)*d^2)*f)*sqrt(((c^2 +
d^2)*f^2*sqrt(-(4*(A^2*B^2 - 2*A*B^2*C + B^2*C^2)*c^2 - 4*(A^3*B - A*B^3 +
3*A*B*C^2 - B*C^3 - (3*A^2*B - B^3)*C)*c*d + (A^4 - 2*A^2*B^2 + B^4 - 4*A*C
^3 + C^4 + 2*(3*A^2 - B^2)*C^2 - 4*(A^3 - A*B^2)*C)*d^2)/((c^4 + 2*c^2*d^2
+ d^4)*f^4)) - (A^2 - B^2 - 2*A*C + C^2)*c - 2*(A*B - B*C)*d)/((c^2 + d^2)*
f^2))) + 4*sqrt(d*tan(f*x + e) + c)*C)/(d*f)
```

Sympy [F]

$$\int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{\sqrt{c + d\tan(e + fx)}}\,dx = \int \frac{A + B\tan\left(e + fx\right) + C\tan^2\left(e + fx\right)}{\sqrt{c + d\tan\left(e + fx\right)}}\,dx$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x+e))**(1/2),x)
[Out] Integral((A + B*tan(e + f*x) + C*tan(e + f*x)**2)/sqrt(c + d*tan(e + f*x)),
 x)
```

Maxima [F]

$$\int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{\sqrt{c + d\tan(e + fx)}}\,dx = \int \frac{C\tan\left(fx + e\right)^2 + B\tan\left(fx + e\right) + A}{\sqrt{d\tan\left(fx + e\right) + c}}\,dx$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2),x, algorit
hm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)/sqrt(d*tan(f*x + e) + c),
 x)
```

Giac [F(-1)]

Timed out.

$$\int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{\sqrt{c + d\tan(e + fx)}}\,dx = \text{Timed out}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2),x, algorit
hm="giac")
```

```
[Out] Timed out
```

Mupad [B] (verification not implemented)

Time = 12.95 (sec) , antiderivative size = 4326, normalized size of antiderivative = 32.53

$$\int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{\sqrt{c + d\tan(e + fx)}}\,dx = \text{Too large to display}$$

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/(c + d*tan(e + f*x))^(1/2),x)
```

```
[Out] 2*atanh((32*C^2*d^2*((-16*C^4*d^2*f^4)^(1/2)/(16*(c^2*f^4 + d^2*f^4)) - (C^
2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)))^(1/2)*(c + d*tan(e + f*x))^(1/2))/((16*C^
3*c*d^3*f^3)/(c^2*f^4 + d^2*f^4) - (4*C*d^3*f^2*(-16*C^4*d^2*f^4)^(1/2))/(c
^2*f^5 + d^2*f^5)) + (8*c*d^2*((-16*C^4*d^2*f^4)^(1/2)/(16*(c^2*f^4 + d^2*f
^4)) - (C^2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)))^(1/2)*(c + d*tan(e + f*x))^(1/2
)*(-16*C^4*d^2*f^4)^(1/2))/((16*C^3*c*d^5*f^5)/(c^2*f^4 + d^2*f^4) - (4*C*d
^5*f^4*(-16*C^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^2*f^5) + (16*C^3*c^3*d^3*f^5)/
(c^2*f^4 + d^2*f^4) - (4*C*c^2*d^3*f^4*(-16*C^4*d^2*f^4)^(1/2))/(c^2*f^5 +
d^2*f^5)) - (32*C^2*c^2*d^2*f^2*((-16*C^4*d^2*f^4)^(1/2)/(16*(c^2*f^4 + d^2
*f^4)) - (C^2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)))^(1/2)*(c + d*tan(e + f*x))^(1
/2))/((16*C^3*c*d^5*f^5)/(c^2*f^4 + d^2*f^4) - (4*C*d^5*f^4*(-16*C^4*d^2*f^
4)^(1/2))/(c^2*f^5 + d^2*f^5) + (16*C^3*c^3*d^3*f^5)/(c^2*f^4 + d^2*f^4) -
```

(4*C*c^2*d^3*f^4*(-16*C^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^2*f^5)))*((-16*C^4*d ^2*f^4)^(1/2)/(16*(c^2*f^4 + d^2*f^4)) - (C^2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)))^(1/2) - 2*atanh((8*c*d^2*(- (-16*C^4*d^2*f^4)^(1/2)/(16*(c^2*f^4 + d^2*f ^4)) - (C^2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(-16*C^4*d^2*f^4)^(1/2))/((16*C^3*c*d^5*f^5)/(c^2*f^4 + d^2*f^4) + (4*C*d ^5*f^4*(-16*C^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^2*f^5) + (16*C^3*c^3*d^3*f^5)/ (c^2*f^4 + d^2*f^4) + (4*C*c^2*d^3*f^4*(-16*C^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^2*f^5)) - (32*C^2*d^2*(- (-16*C^4*d^2*f^4)^(1/2)/(16*(c^2*f^4 + d^2*f^4)) - (C^2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)))^(1/2)*(c + d*tan(e + f*x))^(1/2))/((16*C^3*c*d^3*f^3)/(c^2*f^4 + d^2*f^4) + (4*C*d^3*f^2*(-16*C^4*d^2*f^4)^(1/ 2))/(c^2*f^5 + d^2*f^5)) + (32*C^2*c^2*d^2*f^2*(- (-16*C^4*d^2*f^4)^(1/2)/(16*(c^2*f^4 + d^2*f^4)) - (C^2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)))^(1/2)*(c + d *tan(e + f*x))^(1/2))/((16*C^3*c*d^5*f^5)/(c^2*f^4 + d^2*f^4) + (4*C*d^5*f^ 4*(-16*C^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^2*f^5) + (16*C^3*c^3*d^3*f^5)/(c^2* f^4 + d^2*f^4) + (4*C*c^2*d^3*f^4*(-16*C^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^2*f ^5)))*(- (-16*C^4*d^2*f^4)^(1/2)/(16*(c^2*f^4 + d^2*f^4)) - (C^2*c*f^2)/(4* (c^2*f^4 + d^2*f^4)))^(1/2) - 2*atanh((32*A^2*d^2*((-16*A^4*d^2*f^4)^(1/2)/ (16*(c^2*f^4 + d^2*f^4)) - (A^2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)))^(1/2)*(c + d*tan(e + f*x))^(1/2))/((16*A^3*c*d^3*f^3)/(c^2*f^4 + d^2*f^4) - (4*A*d^3*f ^2*(-16*A^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^2*f^5)) + (8*c*d^2*((-16*A^4*d^2*f ^4)^(1/2)/(16*(c^2*f^4 + d^2*f^4)) - (A^2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(-16*A^4*d^2*f^4)^(1/2))/((16*A^3*c*d^5*f^5)/(c^2*f^4 + d^2*f^4) - (4*A*d^5*f^4*(-16*A^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^ 2*f^5) + (16*A^3*c^3*d^3*f^5)/(c^2*f^4 + d^2*f^4) - (4*A*c^2*d^3*f^4*(-16*A ^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^2*f^5)) - (32*A^2*c^2*d^2*f^2*((-16*A^4*d^2 *f^4)^(1/2)/(16*(c^2*f^4 + d^2*f^4)) - (A^2*c*f^2)/(4*(c^2*f^4 + d^2*f^4))) ^(1/2)*(c + d*tan(e + f*x))^(1/2))/((16*A^3*c*d^5*f^5)/(c^2*f^4 + d^2*f^4) - (4*A*d^5*f^4*(-16*A^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^2*f^5) + (16*A^3*c^3*d ^3*f^5)/(c^2*f^4 + d^2*f^4) - (4*A*c^2*d^3*f^4*(-16*A^4*d^2*f^4)^(1/2))/(c^ 2*f^5 + d^2*f^5)))*((-16*A^4*d^2*f^4)^(1/2)/(16*(c^2*f^4 + d^2*f^4)) - (A^2 *c*f^2)/(4*(c^2*f^4 + d^2*f^4)))^(1/2) + 2*atanh((8*c*d^2*(- (-16*A^4*d^2*f ^4)^(1/2)/(16*(c^2*f^4 + d^2*f^4)) - (A^2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(-16*A^4*d^2*f^4)^(1/2))/((16*A^3*c*d^5*f^5)/(c^2*f^4 + d^2*f^4) + (4*A*d^5*f^4*(-16*A^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^ 2*f^5) + (16*A^3*c^3*d^3*f^5)/(c^2*f^4 + d^2*f^4) + (4*A*c^2*d^3*f^4*(-16*A ^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^2*f^5)) - (32*A^2*d^2*(- (-16*A^4*d^2*f^4)^ (1/2)/(16*(c^2*f^4 + d^2*f^4)) - (A^2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)))^(1/2) *(c + d*tan(e + f*x))^(1/2))/((16*A^3*c*d^3*f^3)/(c^2*f^4 + d^2*f^4) + (4*A *d^3*f^2*(-16*A^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^2*f^5)) + (32*A^2*c^2*d^2*f^ 2*(- (-16*A^4*d^2*f^4)^(1/2)/(16*(c^2*f^4 + d^2*f^4)) - (A^2*c*f^2)/(4*(c^2 *f^4 + d^2*f^4)))^(1/2)*(c + d*tan(e + f*x))^(1/2))/((16*A^3*c*d^5*f^5)/(c^ 2*f^4 + d^2*f^4) + (4*A*d^5*f^4*(-16*A^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^2*f^5) + (16*A^3*c^3*d^3*f^5)/(c^2*f^4 + d^2*f^4) + (4*A*c^2*d^3*f^4*(-16*A^4*d^ 2*f^4)^(1/2))/(c^2*f^5 + d^2*f^5)))*(- (-16*A^4*d^2*f^4)^(1/2)/(16*(c^2*f^4 + d^2*f^4)) - (A^2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)))^(1/2) - 2*atanh((32*B^2

```
*d^2*((B^2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)) - (-16*B^4*d^2*f^4)^(1/2)/(16*(c^
2*f^4 + d^2*f^4)))^(1/2)*(c + d*tan(e + f*x))^(1/2))/((16*B^3*d^2)/f - (16*
B^3*c^2*d^2*f^3)/(c^2*f^4 + d^2*f^4) + (4*B*c*d^2*f^2*(-16*B^4*d^2*f^4)^(1/
2))/(c^2*f^5 + d^2*f^5)) + (8*c*d^2*((B^2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)) -
(-16*B^4*d^2*f^4)^(1/2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*(c + d*tan(e + f*x)
)^(1/2)*(-16*B^4*d^2*f^4)^(1/2))/(16*B^3*d^4*f + 16*B^3*c^2*d^2*f - (16*B^3
*c^2*d^4*f^5)/(c^2*f^4 + d^2*f^4) - (16*B^3*c^4*d^2*f^5)/(c^2*f^4 + d^2*f^4
) + (4*B*c*d^4*f^4*(-16*B^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^2*f^5) + (4*B*c^3*
d^2*f^4*(-16*B^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^2*f^5)) - (32*B^2*c^2*d^2*f^2
*((B^2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)) - (-16*B^4*d^2*f^4)^(1/2)/(16*(c^2*f^
4 + d^2*f^4)))^(1/2)*(c + d*tan(e + f*x))^(1/2))/(16*B^3*d^4*f + 16*B^3*c^2
*d^2*f - (16*B^3*c^2*d^4*f^5)/(c^2*f^4 + d^2*f^4) - (16*B^3*c^4*d^2*f^5)/(c
^2*f^4 + d^2*f^4) + (4*B*c*d^4*f^4*(-16*B^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^2*
f^5) + (4*B*c^3*d^2*f^4*(-16*B^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^2*f^5)))*((B^
2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)) - (-16*B^4*d^2*f^4)^(1/2)/(16*(c^2*f^4 + d
^2*f^4)))^(1/2) - 2*atanh((8*c*d^2*((-16*B^4*d^2*f^4)^(1/2)/(16*(c^2*f^4 +
d^2*f^4)) + (B^2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)))^(1/2)*(c + d*tan(e + f*x))
^(1/2)*(-16*B^4*d^2*f^4)^(1/2))/((16*B^3*c^2*d^4*f^5)/(c^2*f^4 + d^2*f^4) -
 16*B^3*c^2*d^2*f - 16*B^3*d^4*f + (16*B^3*c^4*d^2*f^5)/(c^2*f^4 + d^2*f^4)
 + (4*B*c*d^4*f^4*(-16*B^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^2*f^5) + (4*B*c^3*d
^2*f^4*(-16*B^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^2*f^5)) - (32*B^2*d^2*((-16*B^
4*d^2*f^4)^(1/2)/(16*(c^2*f^4 + d^2*f^4)) + (B^2*c*f^2)/(4*(c^2*f^4 + d^2*f
^4)))^(1/2)*(c + d*tan(e + f*x))^(1/2))/((16*B^3*c^2*d^2*f^3)/(c^2*f^4 + d^
2*f^4) - (16*B^3*d^2)/f + (4*B*c*d^2*f^2*(-16*B^4*d^2*f^4)^(1/2))/(c^2*f^5
+ d^2*f^5)) + (32*B^2*c^2*d^2*f^2*((-16*B^4*d^2*f^4)^(1/2)/(16*(c^2*f^4 + d
^2*f^4)) + (B^2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)))^(1/2)*(c + d*tan(e + f*x))^
(1/2))/((16*B^3*c^2*d^4*f^5)/(c^2*f^4 + d^2*f^4) - 16*B^3*c^2*d^2*f - 16*B^
3*d^4*f + (16*B^3*c^4*d^2*f^5)/(c^2*f^4 + d^2*f^4) + (4*B*c*d^4*f^4*(-16*B^
4*d^2*f^4)^(1/2))/(c^2*f^5 + d^2*f^5) + (4*B*c^3*d^2*f^4*(-16*B^4*d^2*f^4)^
(1/2))/(c^2*f^5 + d^2*f^5)))*((-16*B^4*d^2*f^4)^(1/2)/(16*(c^2*f^4 + d^2*f^
4)) + (B^2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)))^(1/2) + (2*C*(c + d*tan(e + f*x)
)^(1/2))/(d*f)
```

3.114 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}}\,dx$

Optimal result 1511
Rubi [A] (verified) 1511
Mathematica [A] (verified) 1514
Maple [B] (verified) 1514
Fricas [F(-1)] 1514
Sympy [F] 1515
Maxima [F(-2)] 1515
Giac [F(-1)] 1515
Mupad [B] (verification not implemented) 1516

Optimal result

Integrand size = 47, antiderivative size = 210

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}}\,dx = -\frac{(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(a-ib)\sqrt{c-id}f} - \frac{(A+iB-C)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(ia-b)\sqrt{c+id}f} - \frac{2(Ab^2-a(bB-aC))\,\text{arctanh}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{\sqrt{b}\,(a^2+b^2)\,\sqrt{bc-ad}f}$$

```
[Out] -(I*A+B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(a-I*b)/f/(c-I*d
)^(1/2)-(A+I*B-C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))/(I*a-b)/f/(
c+I*d)^(1/2)-2*(A*b^2-a*(B*b-C*a))*arctanh(b^(1/2)*(c+d*tan(f*x+e))^(1/2)/(
-a*d+b*c)^(1/2))/(a^2+b^2)/f/b^(1/2)/(-a*d+b*c)^(1/2)
```

Rubi [A] (verified)

Time = 0.67 (sec) , antiderivative size = 210, normalized size of antiderivative = 1.00, number of steps used = 11, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.128, Rules used = {3734, 3620, 3618, 65, 214, 3715}

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}}\,dx = -\frac{2(Ab^2-a(bB-aC))\,\text{arctanh}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{\sqrt{b}f\,(a^2+b^2)\,\sqrt{bc-ad}} - \frac{(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f(a-ib)\sqrt{c-id}} - \frac{(A+iB-C)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f(-b+ia)\sqrt{c+id}}$$

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])*Sqrt[c +
d*Tan[e + f*x]]),x]
```

```
[Out] -(((I*A + B - I*C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]])/((a - I
*b)*Sqrt[c - I*d]*f)) - ((A + I*B - C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqr
t[c + I*d]])/((I*a - b)*Sqrt[c + I*d]*f) - (2*(A*b^2 - a*(b*B - a*C))*ArcTa
nh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*d]])/(Sqrt[b]*(a^2 + b^2
)*Sqrt[b*c - a*d]*f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3715

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_) + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :>
 Dist[A/f, Subst[Int[(a + b*x)^m*(c + d*x)^n, x], x, Tan[e + f*x]], x] /; F
reeQ[{a, b, c, d, e, f, A, C, m, n}, x] && EqQ[A, C]
```

Rule 3734

```
Int[(((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2))/((a_.) + (b_.)*tan[(e_.)
```

```
+ (f_.)*(x_)]), x_Symbol] :> Dist[1/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n
*Simp[b*B + a*(A - C) + (a*B - b*(A - C))*Tan[e + f*x], x], x], x] + Dist[(
A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n*((1 + Tan[e
+ f*x]^2)/(a + b*Tan[e + f*x])), x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C
, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&
 !GtQ[n, 0] &&  !LeQ[n, -1]
```

Rubi steps

$$\begin{aligned}
\text{integral} &= \frac{\int \frac{bB+a(A-C)-(Ab-aB-bC)\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{a^2+b^2} \\
&\quad+\frac{(Ab^2-abB+a^2C)\int \frac{1+\tan^2(e+fx)}{(a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}}\,dx}{a^2+b^2} \\
&= \frac{(A-iB-C)\int \frac{1+i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(a-ib)}+\frac{(A+iB-C)\int \frac{1-i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(a+ib)} \\
&\quad+\frac{(Ab^2-abB+a^2C)\,\text{Subst}\left(\int \frac{1}{(a+bx)\sqrt{c+dx}}\,dx,x,\tan(e+fx)\right)}{(a^2+b^2)\,f} \\
&= -\frac{(i(A+iB-C))\text{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c+idx}}\,dx,x,-i\tan(e+fx)\right)}{2(a+ib)f} \\
&\quad+\frac{(iA+B-iC)\text{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c-idx}}\,dx,x,i\tan(e+fx)\right)}{2(a-ib)f} \\
&\quad+\frac{(2(Ab^2-abB+a^2C))\,\text{Subst}\left(\int \frac{1}{a-\frac{bc}{d}+\frac{bx^2}{d}}\,dx,x,\sqrt{c+d\tan(e+fx)}\right)}{(a^2+b^2)\,df} \\
&= -\frac{2(Ab^2-a(bB-aC))\,\text{arctanh}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{\sqrt{b}\,(a^2+b^2)\,\sqrt{bc-ad}f} \\
&\quad-\frac{(A-iB-C)\text{Subst}\left(\int \frac{1}{-1-\frac{ic}{d}+\frac{ix^2}{d}}\,dx,x,\sqrt{c+d\tan(e+fx)}\right)}{(a-ib)df} \\
&\quad-\frac{(A+iB-C)\text{Subst}\left(\int \frac{1}{-1+\frac{ic}{d}-\frac{ix^2}{d}}\,dx,x,\sqrt{c+d\tan(e+fx)}\right)}{(a+ib)df} \\
&= -\frac{(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(a-ib)\sqrt{c-id}f}-\frac{(A+iB-C)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(ia-b)\sqrt{c+id}f} \\
&\quad-\frac{2(Ab^2-a(bB-aC))\,\text{arctanh}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{\sqrt{b}\,(a^2+b^2)\,\sqrt{bc-ad}f}
\end{aligned}$$

Mathematica [A] (verified)

Time = 0.45 (sec) , antiderivative size = 194, normalized size of antiderivative = 0.92

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}}\,dx$$
$$=\frac{\frac{(-ia+b)(A-iB-C)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{\sqrt{c-id}}+\frac{(ia+b)(A+iB-C)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{\sqrt{c+id}}-\frac{2(Ab^2+a(-bB+aC))\operatorname{arctanh}\left(\frac{\sqrt{b}\cdots}{\cdots}\right)}{\sqrt{b}\sqrt{bc-ad}}}{(a^2+b^2)\,f}$$

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])*Sqr
t[c + d*Tan[e + f*x]]),x]
```

```
[Out] ((((-I)*a + b)*(A - I*B - C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]
])/Sqrt[c - I*d] + ((I*a + b)*(A + I*B - C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]
]/Sqrt[c + I*d]])/Sqrt[c + I*d] - (2*(A*b^2 + a*(-(b*B) + a*C))*ArcTanh[(Sq
rt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*d]])/(Sqrt[b]*Sqrt[b*c - a*d])
)/((a^2 + b^2)*f)
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 13473 vs. 2(179) = 358.

Time = 0.14 (sec) , antiderivative size = 13474, normalized size of antiderivative = 64.16

method	result	size
derivativedivides	Expression too large to display	13474
default	Expression too large to display	13474

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f*x+e))
,x,method=_RETURNVERBOSE)
```

```
[Out] result too large to display
```

Fricas [F(-1)]

Timed out.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}}\,dx=\text{Timed out}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f
*x+e)),x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}}\,dx$$
$$=\int \frac{A+B\tan\left(e+fx\right)+C\tan^2\left(e+fx\right)}{\left(a+b\tan\left(e+fx\right)\right)\sqrt{c+d\tan\left(e+fx\right)}}\,dx$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x+e))**(1/2)/(a+b*tan
(f*x+e)),x)
[Out] Integral((A + B*tan(e + f*x) + C*tan(e + f*x)**2)/((a + b*tan(e + f*x))*sqr
t(c + d*tan(e + f*x))), x)
```

Maxima [F(-2)]

Exception generated.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}}\,dx=\text{Exception raised: ValueError}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f
*x+e)),x, algorithm="maxima")
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 detail
```

Giac [F(-1)]

Timed out.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}}\,dx=\text{Timed out}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f
*x+e)),x, algorithm="giac")
[Out] Timed out
```

Mupad [B] (verification not implemented)

Time = 65.48 (sec) , antiderivative size = 25341, normalized size of antiderivative = 120.67

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}}\,dx = \text{Too large to display}$$

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))*(c + d*ta
n(e + f*x))^(1/2)),x)
```

```
[Out] (log(((((((((128*C*b^2*d^8*(a*d + b*c)^2*(a^2 + b^2)^2)/f - 64*b^2*d^8*(a^
2 + b^2)^2*(c + d*tan(e + f*x))^(1/2)*((4*(-C^4*f^4*(a^2*d - b^2*d + 2*a*b*
c)^2)^(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(f^4*(a^
2 + b^2)^2*(c^2 + d^2)))^(1/2)*(3*b^3*c^2 + 2*b^3*d^2 - a^2*b*c^2 - 2*a^2*b
*d^2 + a^3*c*d + a*b^2*c*d))*((4*(-C^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/
2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^
2*(c^2 + d^2)))^(1/2))/4 + (64*C^2*b*d^8*(c + d*tan(e + f*x))^(1/2)*(5*b^6*
c - 4*a^6*c - 2*a^2*b^4*c + 5*a^4*b^2*c - 2*a^3*b^3*d + 7*a*b^5*d + 7*a^5*b
*d))/f^2)*((4*(-C^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) - 4*C^2*a^2*c*f^
2 + 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/
2))/4 + (32*C^3*b*d^8*(4*a^5*d - b^5*c - 9*a^2*b^3*c - 15*a^3*b^2*d + 12*a^
4*b*c + a*b^4*d))/f^3)*((4*(-C^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) - 4
*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2
 + d^2)))^(1/2))/4 - (32*C^4*b*d^8*(2*a^4 + b^4)*(c + d*tan(e + f*x))^(1/2)
)/f^4)*((4*(-C^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) - 4*C^2*a^2*c*f^2 +
 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))
/4 + (32*C^5*a^2*b^2*d^8)/f^5)*(((32*C^4*a^2*b^2*d^2*f^4 - 16*C^4*b^4*d^2*f
^4 - 64*C^4*a^2*b^2*c^2*f^4 - 16*C^4*a^4*d^2*f^4 + 64*C^4*a*b^3*c*d*f^4 - 6
4*C^4*a^3*b*c*d*f^4)^(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 + 8*C^2*a*b*
d*f^2)/(a^4*c^2*f^4 + a^4*d^2*f^4 + b^4*c^2*f^4 + b^4*d^2*f^4 + 2*a^2*b^2*c
^2*f^4 + 2*a^2*b^2*d^2*f^4))^(1/2))/4 + (log(((((((((128*C*b^2*d^8*(a*d +
b*c)^2*(a^2 + b^2)^2)/f - 64*b^2*d^8*(a^2 + b^2)^2*(c + d*tan(e + f*x))^(1/
2)*(-(4*(-C^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) + 4*C^2*a^2*c*f^2 - 4*
C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2)*(3*
b^3*c^2 + 2*b^3*d^2 - a^2*b*c^2 - 2*a^2*b*d^2 + a^3*c*d + a*b^2*c*d))*(-(4*
(-C^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*
c*f^2 - 8*C^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 + (64*C^
2*b*d^8*(c + d*tan(e + f*x))^(1/2)*(5*b^6*c - 4*a^6*c - 2*a^2*b^4*c + 5*a^4
*b^2*c - 2*a^3*b^3*d + 7*a*b^5*d + 7*a^5*b*d))/f^2)*(-(4*(-C^4*f^4*(a^2*d -
 b^2*d + 2*a*b*c)^2)^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 - 8*C^2*a*b*
d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 + (32*C^3*b*d^8*(4*a^5*d -
 b^5*c - 9*a^2*b^3*c - 15*a^3*b^2*d + 12*a^4*b*c + a*b^4*d))/f^3)*(-(4*(-C^
4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^
2 - 8*C^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 - (32*C^4*b*
d^8*(2*a^4 + b^4)*(c + d*tan(e + f*x))^(1/2))/f^4)*(-(4*(-C^4*f^4*(a^2*d -
```

b^2*d + 2*a*b*c)^2)^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d
f^2)/(f^4(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 + (32*C^5*a^2*b^2*d^8)/f^5)
*(-((32*C^4*a^2*b^2*d^2*f^4 - 16*C^4*b^4*d^2*f^4 - 64*C^4*a^2*b^2*c^2*f^4 -
16*C^4*a^4*d^2*f^4 + 64*C^4*a*b^3*c*d*f^4 - 64*C^4*a^3*b*c*d*f^4)^(1/2) +
4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(a^4*c^2*f^4 + a^4*d^2
*f^4 + b^4*c^2*f^4 + b^4*d^2*f^4 + 2*a^2*b^2*c^2*f^4 + 2*a^2*b^2*d^2*f^4))^
(1/2))/4 - log(((((((((128*C*b^2*d^8*(a*d + b*c)^2*(a^2 + b^2)^2)/f + 64*b
^2*d^8*(a^2 + b^2)^2*(c + d*tan(e + f*x))^(1/2)*((4*(-C^4*f^4*(a^2*d - b^2*
d + 2*a*b*c)^2)^(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2
)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2)*(3*b^3*c^2 + 2*b^3*d^2 - a^2*b*c^2
- 2*a^2*b*d^2 + a^3*c*d + a*b^2*c*d))*((4*(-C^4*f^4*(a^2*d - b^2*d + 2*a*b
*c)^2)^(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(f^4*(a
^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 - (64*C^2*b*d^8*(c + d*tan(e + f*x))^(1/
2)*(5*b^6*c - 4*a^6*c - 2*a^2*b^4*c + 5*a^4*b^2*c - 2*a^3*b^3*d + 7*a*b^5*d
+ 7*a^5*b*d))/f^2)*((4*(-C^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) - 4*C^
2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 +
d^2)))^(1/2))/4 + (32*C^3*b*d^8*(4*a^5*d - b^5*c - 9*a^2*b^3*c - 15*a^3*b^2
*d + 12*a^4*b*c + a*b^4*d))/f^3)*((4*(-C^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)
^(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(f^4*(a^2 + b
^2)^2*(c^2 + d^2)))^(1/2))/4 + (32*C^4*b*d^8*(2*a^4 + b^4)*(c + d*tan(e + f
x))^(1/2))/f^4)((4*(-C^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) - 4*C^2*a
^2*c*f^2 + 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2
)))^(1/2))/4 + (32*C^5*a^2*b^2*d^8)/f^5)*(((32*C^4*a^2*b^2*d^2*f^4 - 16*C^4
*b^4*d^2*f^4 - 64*C^4*a^2*b^2*c^2*f^4 - 16*C^4*a^4*d^2*f^4 + 64*C^4*a*b^3*c
*d*f^4 - 64*C^4*a^3*b*c*d*f^4)^(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 +
8*C^2*a*b*d*f^2)/(16*a^4*c^2*f^4 + 16*a^4*d^2*f^4 + 16*b^4*c^2*f^4 + 16*b^4
*d^2*f^4 + 32*a^2*b^2*c^2*f^4 + 32*a^2*b^2*d^2*f^4))^(1/2) - log(((((((((1
28*C*b^2*d^8*(a*d + b*c)^2*(a^2 + b^2)^2)/f + 64*b^2*d^8*(a^2 + b^2)^2*(c +
d*tan(e + f*x))^(1/2)*(-(4*(-C^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) +
4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^
2 + d^2)))^(1/2)*(3*b^3*c^2 + 2*b^3*d^2 - a^2*b*c^2 - 2*a^2*b*d^2 + a^3*c*d
+ a*b^2*c*d))*(-(4*(-C^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) + 4*C^2*a^
2*c*f^2 - 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)
))^(1/2))/4 - (64*C^2*b*d^8*(c + d*tan(e + f*x))^(1/2)*(5*b^6*c - 4*a^6*c -
2*a^2*b^4*c + 5*a^4*b^2*c - 2*a^3*b^3*d + 7*a*b^5*d + 7*a^5*b*d))/f^2)*(-(
4*(-C^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^
2*c*f^2 - 8*C^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 + (32*
C^3*b*d^8*(4*a^5*d - b^5*c - 9*a^2*b^3*c - 15*a^3*b^2*d + 12*a^4*b*c + a*b^
4*d))/f^3)*(-(4*(-C^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) + 4*C^2*a^2*c*
f^2 - 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(
1/2))/4 + (32*C^4*b*d^8*(2*a^4 + b^4)*(c + d*tan(e + f*x))^(1/2))/f^4)*(-(4
*(-C^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2
*c*f^2 - 8*C^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 + (32*C
^5*a^2*b^2*d^8)/f^5)*(-((32*C^4*a^2*b^2*d^2*f^4 - 16*C^4*b^4*d^2*f^4 - 64*C
^4*a^2*b^2*c^2*f^4 - 16*C^4*a^4*d^2*f^4 + 64*C^4*a*b^3*c*d*f^4 - 64*C^4*a^3

*b*c*d*f^4)^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(1
6*a^4*c^2*f^4 + 16*a^4*d^2*f^4 + 16*b^4*c^2*f^4 + 16*b^4*d^2*f^4 + 32*a^2*b
^2*c^2*f^4 + 32*a^2*b^2*d^2*f^4))^(1/2) + (log(- (((((((((128*B*b^2*d^8*(a^
2 + b^2)^2*(a*b*c^2 + 3*a*b*d^2 - a^2*c*d + b^2*c*d))/f + 64*b^2*d^8*(a^2 +
 b^2)^2*(c + d*tan(e + f*x))^(1/2)*((4*(-B^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^
2)^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(f^4*(a^2 +
 b^2)^2*(c^2 + d^2)))^(1/2)*(3*b^3*c^2 + 2*b^3*d^2 - a^2*b*c^2 - 2*a^2*b*d^
2 + a^3*c*d + a*b^2*c*d))*((4*(-B^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2)
+ 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(
c^2 + d^2)))^(1/2))/4 - (64*B^2*b^2*d^8*(c + d*tan(e + f*x))^(1/2)*(a^5*d -
 5*b^5*c + 6*a^2*b^3*c + 10*a^3*b^2*d - 5*a^4*b*c - 7*a*b^4*d))/f^2)*((4*(-
B^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*
f^2 - 8*B^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 + (32*B^3*
a*b^2*d^8*(a^3*d + 7*b^3*c - 5*a^2*b*c + 13*a*b^2*d))/f^3)*((4*(-B^4*f^4*(a
^2*d - b^2*d + 2*a*b*c)^2)^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 - 8*B^
2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 - (32*B^4*b^3*d^8*(2
a^2 - b^2)(c + d*tan(e + f*x))^(1/2))/f^4)*((4*(-B^4*f^4*(a^2*d - b^2*d +
 2*a*b*c)^2)^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(
f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 - (32*B^5*a*b^3*d^8)/f^5)*(((32*B^
4*a^2*b^2*d^2*f^4 - 16*B^4*b^4*d^2*f^4 - 64*B^4*a^2*b^2*c^2*f^4 - 16*B^4*a^
4*d^2*f^4 + 64*B^4*a*b^3*c*d*f^4 - 64*B^4*a^3*b*c*d*f^4)^(1/2) + 4*B^2*a^2*
c*f^2 - 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(a^4*c^2*f^4 + a^4*d^2*f^4 + b^4
*c^2*f^4 + b^4*d^2*f^4 + 2*a^2*b^2*c^2*f^4 + 2*a^2*b^2*d^2*f^4))^(1/2))/4 +
 (log(- (((((((((128*B*b^2*d^8*(a^2 + b^2)^2*(a*b*c^2 + 3*a*b*d^2 - a^2*c*d
 + b^2*c*d))/f + 64*b^2*d^8*(a^2 + b^2)^2*(c + d*tan(e + f*x))^(1/2)*(-(4*(
-B^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c
*f^2 + 8*B^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2)*(3*b^3*c^2 +
 2*b^3*d^2 - a^2*b*c^2 - 2*a^2*b*d^2 + a^3*c*d + a*b^2*c*d))*(-(4*(-B^4*f^4
*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 + 8
*B^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 - (64*B^2*b^2*d^8
*(c + d*tan(e + f*x))^(1/2)*(a^5*d - 5*b^5*c + 6*a^2*b^3*c + 10*a^3*b^2*d -
 5*a^4*b*c - 7*a*b^4*d))/f^2)*(-(4*(-B^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(
1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(f^4*(a^2 + b^2
)^2*(c^2 + d^2)))^(1/2))/4 + (32*B^3*a*b^2*d^8*(a^3*d + 7*b^3*c - 5*a^2*b*c
 + 13*a*b^2*d))/f^3)*(-(4*(-B^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) - 4*
B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2
+ d^2)))^(1/2))/4 - (32*B^4*b^3*d^8*(2*a^2 - b^2)*(c + d*tan(e + f*x))^(1/2
))/f^4)*(-(4*(-B^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) - 4*B^2*a^2*c*f^2
 + 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2
))/4 - (32*B^5*a*b^3*d^8)/f^5)*(-((32*B^4*a^2*b^2*d^2*f^4 - 16*B^4*b^4*d^2*
f^4 - 64*B^4*a^2*b^2*c^2*f^4 - 16*B^4*a^4*d^2*f^4 + 64*B^4*a*b^3*c*d*f^4 -
64*B^4*a^3*b*c*d*f^4)^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 + 8*B^2*a*b
*d*f^2)/(a^4*c^2*f^4 + a^4*d^2*f^4 + b^4*c^2*f^4 + b^4*d^2*f^4 + 2*a^2*b^2*
c^2*f^4 + 2*a^2*b^2*d^2*f^4))^(1/2))/4 - log(- (((((((((128*B*b^2*d^8*(a^2
+ b^2)^2*(a*b*c^2 + 3*a*b*d^2 - a^2*c*d + b^2*c*d))/f - 64*b^2*d^8*(a^2 + b

^2)^2*(c + d*tan(e + f*x))^(1/2)*((4*(-B^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)
^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(f^4*(a^2 + b
^2)^2*(c^2 + d^2)))^(1/2)*(3*b^3*c^2 + 2*b^3*d^2 - a^2*b*c^2 - 2*a^2*b*d^2
+ a^3*c*d + a*b^2*c*d))*((4*(-B^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) +
4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^
2 + d^2)))^(1/2))/4 + (64*B^2*b^2*d^8*(c + d*tan(e + f*x))^(1/2)*(a^5*d - 5
*b^5*c + 6*a^2*b^3*c + 10*a^3*b^2*d - 5*a^4*b*c - 7*a*b^4*d))/f^2)*((4*(-B^
4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^
2 - 8*B^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 + (32*B^3*a*
b^2*d^8*(a^3*d + 7*b^3*c - 5*a^2*b*c + 13*a*b^2*d))/f^3)*((4*(-B^4*f^4*(a^2
*d - b^2*d + 2*a*b*c)^2)^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 - 8*B^2*
a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 + (32*B^4*b^3*d^8*(2*a
^2 - b^2)*(c + d*tan(e + f*x))^(1/2))/f^4)*((4*(-B^4*f^4*(a^2*d - b^2*d + 2
*a*b*c)^2)^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(f^
4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 - (32*B^5*a*b^3*d^8)/f^5)*(((32*B^4*
a^2*b^2*d^2*f^4 - 16*B^4*b^4*d^2*f^4 - 64*B^4*a^2*b^2*c^2*f^4 - 16*B^4*a^4*
d^2*f^4 + 64*B^4*a*b^3*c*d*f^4 - 64*B^4*a^3*b*c*d*f^4)^(1/2) + 4*B^2*a^2*c*
f^2 - 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(16*a^4*c^2*f^4 + 16*a^4*d^2*f^4 +
16*b^4*c^2*f^4 + 16*b^4*d^2*f^4 + 32*a^2*b^2*c^2*f^4 + 32*a^2*b^2*d^2*f^4)
)^(1/2) - log(- ((((((((128*B*b^2*d^8*(a^2 + b^2)^2*(a*b*c^2 + 3*a*b*d^2 -
a^2*c*d + b^2*c*d))/f - 64*b^2*d^8*(a^2 + b^2)^2*(c + d*tan(e + f*x))^(1/2
)*(-(4*(-B^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) - 4*B^2*a^2*c*f^2 + 4*B
^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2)*(3*b
^3*c^2 + 2*b^3*d^2 - a^2*b*c^2 - 2*a^2*b*d^2 + a^3*c*d + a*b^2*c*d))*(-(4*(
-B^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c
*f^2 + 8*B^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 + (64*B^2
*b^2*d^8*(c + d*tan(e + f*x))^(1/2)*(a^5*d - 5*b^5*c + 6*a^2*b^3*c + 10*a^3
*b^2*d - 5*a^4*b*c - 7*a*b^4*d))/f^2)*(-(4*(-B^4*f^4*(a^2*d - b^2*d + 2*a*b
*c)^2)^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(f^4*(a
^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 + (32*B^3*a*b^2*d^8*(a^3*d + 7*b^3*c - 5
*a^2*b*c + 13*a*b^2*d))/f^3)*(-(4*(-B^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1
/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(f^4*(a^2 + b^2)
^2*(c^2 + d^2)))^(1/2))/4 + (32*B^4*b^3*d^8*(2*a^2 - b^2)*(c + d*tan(e + f*
x))^(1/2))/f^4)*(-(4*(-B^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) - 4*B^2*a
^2*c*f^2 + 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2
)))^(1/2))/4 - (32*B^5*a*b^3*d^8)/f^5)*(-((32*B^4*a^2*b^2*d^2*f^4 - 16*B^4*
b^4*d^2*f^4 - 64*B^4*a^2*b^2*c^2*f^4 - 16*B^4*a^4*d^2*f^4 + 64*B^4*a*b^3*c*
d*f^4 - 64*B^4*a^3*b*c*d*f^4)^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 + 8
*B^2*a*b*d*f^2)/(16*a^4*c^2*f^4 + 16*a^4*d^2*f^4 + 16*b^4*c^2*f^4 + 16*b^4*
d^2*f^4 + 32*a^2*b^2*c^2*f^4 + 32*a^2*b^2*d^2*f^4))^(1/2) + (log(((((((((128
*A*b^2*d^8*(a^2 + b^2)^2*(a^2*d^2 - 3*b^2*c^2 - 4*b^2*d^2 + 2*a*b*c*d))/f +
64*b^2*d^8*(a^2 + b^2)^2*(c + d*tan(e + f*x))^(1/2)*((4*(-A^4*f^4*(a^2*d -
b^2*d + 2*a*b*c)^2)^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 + 8*A^2*a*b*
d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2)*(3*b^3*c^2 + 2*b^3*d^2 - a^2*
b*c^2 - 2*a^2*b*d^2 + a^3*c*d + a*b^2*c*d))*((4*(-A^4*f^4*(a^2*d - b^2*d +

```
2*a*b*c)^2)^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(f
^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 + (64*A^2*b^2*d^8*(a^2 - 3*b^2)*(c
 + d*tan(e + f*x))^(1/2)*(a^3*d + 3*b^3*c - a^2*b*c + 5*a*b^2*d))/f^2)*((4*(
-A^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c
*f^2 + 8*A^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 + (32*A^3
*b^3*d^8*(a^3*d + 3*b^3*c - a^2*b*c + 5*a*b^2*d))/f^3)*((4*(-A^4*f^4*(a^2*d
 - b^2*d + 2*a*b*c)^2)^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 + 8*A^2*a*
b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 + (96*A^4*b^5*d^8*(c + d
*tan(e + f*x))^(1/2))/f^4)*(((32*A^4*a^2*b^2*d^2*f^4 - 16*A^4*b^4*d^2*f^4 -
 64*A^4*a^2*b^2*c^2*f^4 - 16*A^4*a^4*d^2*f^4 + 64*A^4*a*b^3*c*d*f^4 - 64*A^
4*a^3*b*c*d*f^4)^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^
2)/(a^4*c^2*f^4 + a^4*d^2*f^4 + b^4*c^2*f^4 + b^4*d^2*f^4 + 2*a^2*b^2*c^2*f
^4 + 2*a^2*b^2*d^2*f^4))^(1/2))/4 + (log(((((((((128*A*b^2*d^8*(a^2 + b^2)^2
*(a^2*d^2 - 3*b^2*c^2 - 4*b^2*d^2 + 2*a*b*c*d))/f + 64*b^2*d^8*(a^2 + b^2)^
2*(c + d*tan(e + f*x))^(1/2)*(-(4*(-A^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1
/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(f^4*(a^2 + b^2)
^2*(c^2 + d^2)))^(1/2)*(3*b^3*c^2 + 2*b^3*d^2 - a^2*b*c^2 - 2*a^2*b*d^2 + a
^3*c*d + a*b^2*c*d))*(-(4*(-A^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) + 4*
A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2
 + d^2)))^(1/2))/4 + (64*A^2*b^2*d^8*(a^2 - 3*b^2)*(c + d*tan(e + f*x))^(1/2
)*(a^3*d + 3*b^3*c - a^2*b*c + 5*a*b^2*d))/f^2)*(-(4*(-A^4*f^4*(a^2*d - b^2
*d + 2*a*b*c)^2)^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^
2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 + (32*A^3*b^3*d^8*(a^3*d + 3*b
^3*c - a^2*b*c + 5*a*b^2*d))/f^3)*(-(4*(-A^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^
2)^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(f^4*(a^2 +
 b^2)^2*(c^2 + d^2)))^(1/2))/4 + (96*A^4*b^5*d^8*(c + d*tan(e + f*x))^(1/2)
)/f^4)*(-((32*A^4*a^2*b^2*d^2*f^4 - 16*A^4*b^4*d^2*f^4 - 64*A^4*a^2*b^2*c^2
*f^4 - 16*A^4*a^4*d^2*f^4 + 64*A^4*a*b^3*c*d*f^4 - 64*A^4*a^3*b*c*d*f^4)^(1
/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(a^4*c^2*f^4 + a
^4*d^2*f^4 + b^4*c^2*f^4 + b^4*d^2*f^4 + 2*a^2*b^2*c^2*f^4 + 2*a^2*b^2*d^2*
f^4))^(1/2))/4 - log(((((((((128*A*b^2*d^8*(a^2 + b^2)^2*(a^2*d^2 - 3*b^2*c^
2 - 4*b^2*d^2 + 2*a*b*c*d))/f - 64*b^2*d^8*(a^2 + b^2)^2*(c + d*tan(e + f*x
))^(1/2)*((4*(-A^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) - 4*A^2*a^2*c*f^2
 + 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2
)*(3*b^3*c^2 + 2*b^3*d^2 - a^2*b*c^2 - 2*a^2*b*d^2 + a^3*c*d + a*b^2*c*d))*
((4*(-A^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*
b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 - (6
4*A^2*b^2*d^8*(a^2 - 3*b^2)*(c + d*tan(e + f*x))^(1/2)*(a^3*d + 3*b^3*c - a
^2*b*c + 5*a*b^2*d))/f^2)*((4*(-A^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2)
 - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(
c^2 + d^2)))^(1/2))/4 + (32*A^3*b^3*d^8*(a^3*d + 3*b^3*c - a^2*b*c + 5*a*b^
2*d))/f^3)*((4*(-A^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) - 4*A^2*a^2*c*f
^2 + 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1
/2))/4 - (96*A^4*b^5*d^8*(c + d*tan(e + f*x))^(1/2))/f^4)*(((32*A^4*a^2*b^2
*d^2*f^4 - 16*A^4*b^4*d^2*f^4 - 64*A^4*a^2*b^2*c^2*f^4 - 16*A^4*a^4*d^2*f^4
```

+ 64*A^4*a*b^3*c*d*f^4 - 64*A^4*a^3*b*c*d*f^4)^(1/2) - 4*A^2*a^2*c*f^2 + 4
*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(16*a^4*c^2*f^4 + 16*a^4*d^2*f^4 + 16*b^4
*c^2*f^4 + 16*b^4*d^2*f^4 + 32*a^2*b^2*c^2*f^4 + 32*a^2*b^2*d^2*f^4))^(1/2)
- log(((((((128*A*b^2*d^8*(a^2 + b^2)^2*(a^2*d^2 - 3*b^2*c^2 - 4*b^2*d^2
+ 2*a*b*c*d))/f - 64*b^2*d^8*(a^2 + b^2)^2*(c + d*tan(e + f*x))^(1/2)*(-(4*
(-A^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*
c*f^2 - 8*A^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2)*(3*b^3*c^2
+ 2*b^3*d^2 - a^2*b*c^2 - 2*a^2*b*d^2 + a^3*c*d + a*b^2*c*d))*(-(4*(-A^4*f^
4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 -
8*A^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 - (64*A^2*b^2*d^
8*(a^2 - 3*b^2)*(c + d*tan(e + f*x))^(1/2)*(a^3*d + 3*b^3*c - a^2*b*c + 5*a
*b^2*d))/f^2)*(-(4*(-A^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) + 4*A^2*a^2
*c*f^2 - 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2))
)^(1/2))/4 + (32*A^3*b^3*d^8*(a^3*d + 3*b^3*c - a^2*b*c + 5*a*b^2*d))/f^3)*
(-(4*(-A^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2
*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 - (
96*A^4*b^5*d^8*(c + d*tan(e + f*x))^(1/2))/f^4)*(-((32*A^4*a^2*b^2*d^2*f^4
- 16*A^4*b^4*d^2*f^4 - 64*A^4*a^2*b^2*c^2*f^4 - 16*A^4*a^4*d^2*f^4 + 64*A^4
*a*b^3*c*d*f^4 - 64*A^4*a^3*b*c*d*f^4)^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*
c*f^2 - 8*A^2*a*b*d*f^2)/(16*a^4*c^2*f^4 + 16*a^4*d^2*f^4 + 16*b^4*c^2*f^4
+ 16*b^4*d^2*f^4 + 32*a^2*b^2*c^2*f^4 + 32*a^2*b^2*d^2*f^4))^(1/2) - (A*ata
n(((A*((A*((32*(A^3*a^3*b^3*d^9 + 5*A^3*a*b^5*d^9 + 3*A^3*b^6*c*d^8 - A^3*a
^2*b^4*c*d^8))/f^3 + (A*((32*(c + d*tan(e + f*x))^(1/2)*(4*A^2*a^3*b^4*d^9*
f^2 + 2*A^2*a^5*b^2*d^9*f^2 - 30*A^2*a*b^6*d^9*f^2 - 18*A^2*b^7*c*d^8*f^2 +
12*A^2*a^2*b^5*c*d^8*f^2 - 2*A^2*a^4*b^3*c*d^8*f^2))/f^4 - (A*((32*(16*A*b
^8*d^10*f^2 + 28*A*a^2*b^6*d^10*f^2 + 8*A*a^4*b^4*d^10*f^2 - 4*A*a^6*b^2*d^
10*f^2 + 12*A*b^8*c^2*d^8*f^2 - 16*A*a^3*b^5*c*d^9*f^2 - 8*A*a^5*b^3*c*d^9*
f^2 + 24*A*a^2*b^6*c^2*d^8*f^2 + 12*A*a^4*b^4*c^2*d^8*f^2 - 8*A*a*b^7*c*d^9
*f^2))/f^3 - (32*A*(c + d*tan(e + f*x))^(1/2)*(b^8*c*f^2 + 2*a^2*b^6*c*f^2
+ a^4*b^4*c*f^2 - 2*a^3*b^5*d*f^2 - a^5*b^3*d*f^2 - a*b^7*d*f^2)^(1/2)*(16*
b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*
f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 -
8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*
b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/(f^4*(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b
^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2)))*(b^8*c*f^2 + 2*a^
2*b^6*c*f^2 + a^4*b^4*c*f^2 - 2*a^3*b^5*d*f^2 - a^5*b^3*d*f^2 - a*b^7*d*f^2
)^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b
*c*f^2 + a*b^4*d*f^2))*(b^8*c*f^2 + 2*a^2*b^6*c*f^2 + a^4*b^4*c*f^2 - 2*a^3
*b^5*d*f^2 - a^5*b^3*d*f^2 - a*b^7*d*f^2)^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2
*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2))*(b^8*c*f^2 +
2*a^2*b^6*c*f^2 + a^4*b^4*c*f^2 - 2*a^3*b^5*d*f^2 - a^5*b^3*d*f^2 - a*b^7*
d*f^2)^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 -
a^4*b*c*f^2 + a*b^4*d*f^2) + (96*A^4*b^5*d^8*(c + d*tan(e + f*x))^(1/2))/f^
4)*(b^8*c*f^2 + 2*a^2*b^6*c*f^2 + a^4*b^4*c*f^2 - 2*a^3*b^5*d*f^2 - a^5*b^3
*d*f^2 - a*b^7*d*f^2)^(1/2)*1i)/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 +

2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2) - (A*((A*((32*(A^3*a^3*b^3*d^9
 + 5*A^3*a*b^5*d^9 + 3*A^3*b^6*c*d^8 - A^3*a^2*b^4*c*d^8))/f^3 - (A*((32*(c
 + d*tan(e + f*x))^(1/2)*(4*A^2*a^3*b^4*d^9*f^2 + 2*A^2*a^5*b^2*d^9*f^2 - 3
0*A^2*a*b^6*d^9*f^2 - 18*A^2*b^7*c*d^8*f^2 + 12*A^2*a^2*b^5*c*d^8*f^2 - 2*A
^2*a^4*b^3*c*d^8*f^2))/f^4 + (A*((32*(16*A*b^8*d^10*f^2 + 28*A*a^2*b^6*d^10
*f^2 + 8*A*a^4*b^4*d^10*f^2 - 4*A*a^6*b^2*d^10*f^2 + 12*A*b^8*c^2*d^8*f^2 -
 16*A*a^3*b^5*c*d^9*f^2 - 8*A*a^5*b^3*c*d^9*f^2 + 24*A*a^2*b^6*c^2*d^8*f^2
+ 12*A*a^4*b^4*c^2*d^8*f^2 - 8*A*a*b^7*c*d^9*f^2))/f^3 + (32*A*(c + d*tan(e
 + f*x))^(1/2)*(b^8*c*f^2 + 2*a^2*b^6*c*f^2 + a^4*b^4*c*f^2 - 2*a^3*b^5*d*f
^2 - a^5*b^3*d*f^2 - a*b^7*d*f^2)^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*
f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a
^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^
8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9
f^4))/(f^4(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^
4*b*c*f^2 + a*b^4*d*f^2)))*(b^8*c*f^2 + 2*a^2*b^6*c*f^2 + a^4*b^4*c*f^2 - 2
*a^3*b^5*d*f^2 - a^5*b^3*d*f^2 - a*b^7*d*f^2)^(1/2))/(a^5*d*f^2 - b^5*c*f^2
 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2))*(b^8*c*f
^2 + 2*a^2*b^6*c*f^2 + a^4*b^4*c*f^2 - 2*a^3*b^5*d*f^2 - a^5*b^3*d*f^2 - a*
b^7*d*f^2)^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^
2 - a^4*b*c*f^2 + a*b^4*d*f^2))*(b^8*c*f^2 + 2*a^2*b^6*c*f^2 + a^4*b^4*c*f^
2 - 2*a^3*b^5*d*f^2 - a^5*b^3*d*f^2 - a*b^7*d*f^2)^(1/2))/(a^5*d*f^2 - b^5*
c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2) - (9
6*A^4*b^5*d^8*(c + d*tan(e + f*x))^(1/2))/f^4)*(b^8*c*f^2 + 2*a^2*b^6*c*f^2
 + a^4*b^4*c*f^2 - 2*a^3*b^5*d*f^2 - a^5*b^3*d*f^2 - a*b^7*d*f^2)^(1/2)*1i)
/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 +
 a*b^4*d*f^2))/((A*((A*((32*(A^3*a^3*b^3*d^9 + 5*A^3*a*b^5*d^9 + 3*A^3*b^6*
c*d^8 - A^3*a^2*b^4*c*d^8))/f^3 + (A*((32*(c + d*tan(e + f*x))^(1/2)*(4*A^2
*a^3*b^4*d^9*f^2 + 2*A^2*a^5*b^2*d^9*f^2 - 30*A^2*a*b^6*d^9*f^2 - 18*A^2*b^
7*c*d^8*f^2 + 12*A^2*a^2*b^5*c*d^8*f^2 - 2*A^2*a^4*b^3*c*d^8*f^2))/f^4 - (A
((32(16*A*b^8*d^10*f^2 + 28*A*a^2*b^6*d^10*f^2 + 8*A*a^4*b^4*d^10*f^2 - 4
*A*a^6*b^2*d^10*f^2 + 12*A*b^8*c^2*d^8*f^2 - 16*A*a^3*b^5*c*d^9*f^2 - 8*A*a
^5*b^3*c*d^9*f^2 + 24*A*a^2*b^6*c^2*d^8*f^2 + 12*A*a^4*b^4*c^2*d^8*f^2 - 8*
A*a*b^7*c*d^9*f^2))/f^3 - (32*A*(c + d*tan(e + f*x))^(1/2)*(b^8*c*f^2 + 2*a
^2*b^6*c*f^2 + a^4*b^4*c*f^2 - 2*a^3*b^5*d*f^2 - a^5*b^3*d*f^2 - a*b^7*d*f^
2)^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*
a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*
c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*
f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/(f^4*(a^5*d*f^2 - b^5*c*
f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2)))*(b^8
*c*f^2 + 2*a^2*b^6*c*f^2 + a^4*b^4*c*f^2 - 2*a^3*b^5*d*f^2 - a^5*b^3*d*f^2
 - a*b^7*d*f^2)^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*
d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2))*(b^8*c*f^2 + 2*a^2*b^6*c*f^2 + a^4*b^4*
c*f^2 - 2*a^3*b^5*d*f^2 - a^5*b^3*d*f^2 - a*b^7*d*f^2)^(1/2))/(a^5*d*f^2 -
b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2))
*(b^8*c*f^2 + 2*a^2*b^6*c*f^2 + a^4*b^4*c*f^2 - 2*a^3*b^5*d*f^2 - a^5*b^3*d

*f^2 - a*b^7*d*f^2)^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3
*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2) + (96*A^4*b^5*d^8*(c + d*tan(e + f*
x))^(1/2))/f^4)*(b^8*c*f^2 + 2*a^2*b^6*c*f^2 + a^4*b^4*c*f^2 - 2*a^3*b^5*d*
f^2 - a^5*b^3*d*f^2 - a*b^7*d*f^2)^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^
3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2) + (A*((A*((32*(A^3*a
^3*b^3*d^9 + 5*A^3*a*b^5*d^9 + 3*A^3*b^6*c*d^8 - A^3*a^2*b^4*c*d^8))/f^3 -
(A*((32*(c + d*tan(e + f*x))^(1/2)*(4*A^2*a^3*b^4*d^9*f^2 + 2*A^2*a^5*b^2*d
^9*f^2 - 30*A^2*a*b^6*d^9*f^2 - 18*A^2*b^7*c*d^8*f^2 + 12*A^2*a^2*b^5*c*d^8
*f^2 - 2*A^2*a^4*b^3*c*d^8*f^2))/f^4 + (A*((32*(16*A*b^8*d^10*f^2 + 28*A*a^
2*b^6*d^10*f^2 + 8*A*a^4*b^4*d^10*f^2 - 4*A*a^6*b^2*d^10*f^2 + 12*A*b^8*c^2
*d^8*f^2 - 16*A*a^3*b^5*c*d^9*f^2 - 8*A*a^5*b^3*c*d^9*f^2 + 24*A*a^2*b^6*c^
2*d^8*f^2 + 12*A*a^4*b^4*c^2*d^8*f^2 - 8*A*a*b^7*c*d^9*f^2))/f^3 + (32*A*(c
 + d*tan(e + f*x))^(1/2)*(b^8*c*f^2 + 2*a^2*b^6*c*f^2 + a^4*b^4*c*f^2 - 2*a
^3*b^5*d*f^2 - a^5*b^3*d*f^2 - a*b^7*d*f^2)^(1/2)*(16*b^9*d^10*f^4 + 16*a^2
*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*
f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^
4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7
*b^2*c*d^9*f^4))/(f^4*(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*
d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2)))*(b^8*c*f^2 + 2*a^2*b^6*c*f^2 + a^4*b^4
*c*f^2 - 2*a^3*b^5*d*f^2 - a^5*b^3*d*f^2 - a*b^7*d*f^2)^(1/2))/(a^5*d*f^2 -
 b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2)
)*(b^8*c*f^2 + 2*a^2*b^6*c*f^2 + a^4*b^4*c*f^2 - 2*a^3*b^5*d*f^2 - a^5*b^3*
d*f^2 - a*b^7*d*f^2)^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^
3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2))*(b^8*c*f^2 + 2*a^2*b^6*c*f^2 + a^
4*b^4*c*f^2 - 2*a^3*b^5*d*f^2 - a^5*b^3*d*f^2 - a*b^7*d*f^2)^(1/2))/(a^5*d*
f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d
*f^2) - (96*A^4*b^5*d^8*(c + d*tan(e + f*x))^(1/2))/f^4)*(b^8*c*f^2 + 2*a^2
*b^6*c*f^2 + a^4*b^4*c*f^2 - 2*a^3*b^5*d*f^2 - a^5*b^3*d*f^2 - a*b^7*d*f^2)
^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*
c*f^2 + a*b^4*d*f^2)))*(b^8*c*f^2 + 2*a^2*b^6*c*f^2 + a^4*b^4*c*f^2 - 2*a^3
*b^5*d*f^2 - a^5*b^3*d*f^2 - a*b^7*d*f^2)^(1/2)*2i)/(a^5*d*f^2 - b^5*c*f^2
- 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2) + (C*a^2*a
tan(((C*a^2*((32*(C^4*b^5*d^8 + 2*C^4*a^4*b*d^8)*(c + d*tan(e + f*x))^(1/2)
)/f^4 + (C*a^2*((32*(15*C^3*a^3*b^3*d^9*f^2 - C^3*a*b^5*d^9*f^2 - 4*C^3*a^5
*b*d^9*f^2 + C^3*b^6*c*d^8*f^2 + 9*C^3*a^2*b^4*c*d^8*f^2 - 12*C^3*a^4*b^2*c
*d^8*f^2))/f^5 - (C*a^2*((32*(c + d*tan(e + f*x))^(1/2)*(14*C^2*a^5*b^2*d^9
*f^2 - 4*C^2*a^3*b^4*d^9*f^2 + 14*C^2*a*b^6*d^9*f^2 + 10*C^2*b^7*c*d^8*f^2
- 8*C^2*a^6*b*c*d^8*f^2 - 4*C^2*a^2*b^5*c*d^8*f^2 + 10*C^2*a^4*b^3*c*d^8*f^
2))/f^4 + (C*a^2*((32*(4*C*a^2*b^6*d^10*f^4 + 8*C*a^4*b^4*d^10*f^4 + 4*C*a^
6*b^2*d^10*f^4 + 4*C*b^8*c^2*d^8*f^4 + 16*C*a^3*b^5*c*d^9*f^4 + 8*C*a^5*b^3
*c*d^9*f^4 + 8*C*a^2*b^6*c^2*d^8*f^4 + 4*C*a^4*b^4*c^2*d^8*f^4 + 8*C*a*b^7*
c*d^9*f^4))/f^5 - (32*C*a^2*(c + d*tan(e + f*x))^(1/2)*(16*b^9*d^10*f^4 + 1
6*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2
*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d
^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 +

8*a^7*b^2*c*d^9*f^4))/(f^4*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2))))/(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2)))/(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2)))/(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2))*1i)/(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2) + (C*a^2*((32*(C^4*b^5*d^8 + 2*C^4*a^4*b*d^8)*(c + d*tan(e + f*x))^(1/2))/f^4 - (C*a^2*((32*(15*C^3*a^3*b^3*d^9*f^2 - C^3*a*b^5*d^9*f^2 - 4*C^3*a^5*b*d^9*f^2 + C^3*b^6*c*d^8*f^2 + 9*C^3*a^2*b^4*c*d^8*f^2 - 12*C^3*a^4*b^2*c*d^8*f^2))/f^5 + (C*a^2*((32*(c + d*tan(e + f*x))^(1/2)*(14*C^2*a^5*b^2*d^9*f^2 - 4*C^2*a^3*b^4*d^9*f^2 + 14*C^2*a*b^6*d^9*f^2 + 10*C^2*b^7*c*d^8*f^2 - 8*C^2*a^6*b*c*d^8*f^2 - 4*C^2*a^2*b^5*c*d^8*f^2 + 10*C^2*a^4*b^3*c*d^8*f^2))/f^4 - (C*a^2*((32*(4*C*a^2*b^6*d^10*f^4 + 8*C*a^4*b^4*d^10*f^4 + 4*C*a^6*b^2*d^10*f^4 + 4*C*b^8*c^2*d^8*f^4 + 16*C*a^3*b^5*c*d^9*f^4 + 8*C*a^5*b^3*c*d^9*f^4 + 8*C*a^2*b^6*c^2*d^8*f^4 + 4*C*a^4*b^4*c^2*d^8*f^4 + 8*C*a*b^7*c*d^9*f^4))/f^5 + (32*C*a^2*(c + d*tan(e + f*x))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/(f^4*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2))))/(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2)))/(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2)))/(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2))*1i)/(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2))/(((C*a^2*((32*(C^4*b^5*d^8 + 2*C^4*a^4*b*d^8)*(c + d*tan(e + f*x))^(1/2))/f^4 - (C*a^2*((32*(15*C^3*a^3*b^3*d^9*f^2 - C^3*a*b^5*d^9*f^2 - 4*C^3*a^5*b*d^9*f^2 + C^3*b^6*c*d^8*f^2 + 9*C^3*a^2*b^4*c*d^8*f^2 - 12*C^3*a^4*b^2*c*d^8*f^2))/f^5 + (C*a^2*((32*(c + d*tan(e + f*x))^(1/2)*(14*C^2*a^5*b^2*d^9*f^2 - 4*C^2*a^3*b^4*d^9*f^2 + 14*C^2*a*b^6*d^9*f^2 + 10*C^2*b^7*c*d^8*f^2 - 8*C^2*a^6*b*c*d^8*f^2 - 4*C^2*a^2*b^5*c*d^8*f^2 + 10*C^2*a^4*b^3*c*d^8*f^2))/f^4 - (C*a^2*((32*(4*C*a^2*b^6*d^10*f^4 + 8*C*a^4*b^4*d^10*f^4 + 4*C*a^6*b^2*d^10*f^4 + 4*C*b^8*c^2*d^8*f^4 + 16*C*a^3*b^5*c*d^9*f^4 + 8*C*a^5*b^3*c*d^9*f^4 + 8*C*a^2*b^6*c^2*d^8*f^4 + 4*C*a^4*b^4*c^2*d^8*f^4 + 8*C*a*b^7*c*d^9*f^4))/f^5 + (32*C*a^2*(c + d*tan(e + f*x))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/(f^4*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2))))/(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2)))/(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2)))/(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2

- 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2)))/(b^6*c*f^2 + 2*a^2*b
^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/
2) - (C*a^2*((32*(C^4*b^5*d^8 + 2*C^4*a^4*b*d^8)*(c + d*tan(e + f*x))^(1/2)
)/f^4 + (C*a^2*((32*(15*C^3*a^3*b^3*d^9*f^2 - C^3*a*b^5*d^9*f^2 - 4*C^3*a^5
*b*d^9*f^2 + C^3*b^6*c*d^8*f^2 + 9*C^3*a^2*b^4*c*d^8*f^2 - 12*C^3*a^4*b^2*c
*d^8*f^2))/f^5 - (C*a^2*((32*(c + d*tan(e + f*x))^(1/2)*(14*C^2*a^5*b^2*d^9
*f^2 - 4*C^2*a^3*b^4*d^9*f^2 + 14*C^2*a*b^6*d^9*f^2 + 10*C^2*b^7*c*d^8*f^2
- 8*C^2*a^6*b*c*d^8*f^2 - 4*C^2*a^2*b^5*c*d^8*f^2 + 10*C^2*a^4*b^3*c*d^8*f^
2))/f^4 + (C*a^2*((32*(4*C*a^2*b^6*d^10*f^4 + 8*C*a^4*b^4*d^10*f^4 + 4*C*a^
6*b^2*d^10*f^4 + 4*C*b^8*c^2*d^8*f^4 + 16*C*a^3*b^5*c*d^9*f^4 + 8*C*a^5*b^3
*c*d^9*f^4 + 8*C*a^2*b^6*c^2*d^8*f^4 + 4*C*a^4*b^4*c^2*d^8*f^4 + 8*C*a*b^7*
c*d^9*f^4))/f^5 - (32*C*a^2*(c + d*tan(e + f*x))^(1/2)*(16*b^9*d^10*f^4 + 1
6*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2
*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d
^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 +
8*a^7*b^2*c*d^9*f^4))/(f^4*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2
*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2))))/(b^6*c*f^2 + 2*a^2*b^4
*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2)
))/(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d
*f^2 - a^5*b*d*f^2)^(1/2)))/(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 -
2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2)))/(b^6*c*f^2 + 2*a^2*b^4
*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2)
+ (64*C^5*a^2*b^2*d^8)/f^5))*2i)/(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*
f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2) - (B*a*atan(((B*a*
((32*(B^4*b^5*d^8 - 2*B^4*a^2*b^3*d^8)*(c + d*tan(e + f*x))^(1/2))/f^4 - (B
a((32*(13*B^3*a^2*b^4*d^9*f^2 + B^3*a^4*b^2*d^9*f^2 + 7*B^3*a*b^5*c*d^8*f
^2 - 5*B^3*a^3*b^3*c*d^8*f^2))/f^5 + (B*a*((32*(c + d*tan(e + f*x))^(1/2)*(
20*B^2*a^3*b^4*d^9*f^2 + 2*B^2*a^5*b^2*d^9*f^2 - 14*B^2*a*b^6*d^9*f^2 - 10*
B^2*b^7*c*d^8*f^2 + 12*B^2*a^2*b^5*c*d^8*f^2 - 10*B^2*a^4*b^3*c*d^8*f^2))/f
^4 + (B*a*((32*(12*B*a*b^7*d^10*f^4 + 4*B*b^8*c*d^9*f^4 + 24*B*a^3*b^5*d^10
*f^4 + 12*B*a^5*b^3*d^10*f^4 + 4*B*a*b^7*c^2*d^8*f^4 + 4*B*a^2*b^6*c*d^9*f^
4 - 4*B*a^4*b^4*c*d^9*f^4 - 4*B*a^6*b^2*c*d^9*f^4 + 8*B*a^3*b^5*c^2*d^8*f^4
+ 4*B*a^5*b^3*c^2*d^8*f^4))/f^5 - (32*B*a*(c + d*tan(e + f*x))^(1/2)*(b^6*
c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a
^5*b*d*f^2)^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*
f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8
*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b
^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/(f^4*(a^5*d*f^2
- b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^
2)))*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5
*d*f^2 - a^5*b*d*f^2)^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a
^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2))*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a
^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2))/(a^5*d*f
^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*
f^2))*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^

5*d*f^2 - a^5*b*d*f^2)^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2))*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2)*1i)/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2) + (B*a*((32*(B^4*b^5*d^8 - 2*B^4*a^2*b^3*d^8)*(c + d*tan(e + f*x))^(1/2))/f^4 + (B*a*((32*(13*B^3*a^2*b^4*d^9*f^2 + B^3*a^4*b^2*d^9*f^2 + 7*B^3*a*b^5*c*d^8*f^2 - 5*B^3*a^3*b^3*c*d^8*f^2))/f^5 - (B*a*((32*(c + d*tan(e + f*x))^(1/2)*(20*B^2*a^3*b^4*d^9*f^2 + 2*B^2*a^5*b^2*d^9*f^2 - 14*B^2*a*b^6*d^9*f^2 - 10*B^2*b^7*c*d^8*f^2 + 12*B^2*a^2*b^5*c*d^8*f^2 - 10*B^2*a^4*b^3*c*d^8*f^2))/f^4 - (B*a*((32*(12*B*a*b^7*d^10*f^4 + 4*B*b^8*c*d^9*f^4 + 24*B*a^3*b^5*d^10*f^4 + 12*B*a^5*b^3*d^10*f^4 + 4*B*a*b^7*c^2*d^8*f^4 + 4*B*a^2*b^6*c*d^9*f^4 - 4*B*a^4*b^4*c*d^9*f^4 - 4*B*a^6*b^2*c*d^9*f^4 + 8*B*a^3*b^5*c^2*d^8*f^4 + 4*B*a^5*b^3*c^2*d^8*f^4))/f^5 + (32*B*a*(c + d*tan(e + f*x))^(1/2)*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/(f^4*(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2)))*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2))*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2))*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2))*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2)*1i)/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2))/((B*a*((32*(B^4*b^5*d^8 - 2*B^4*a^2*b^3*d^8)*(c + d*tan(e + f*x))^(1/2))/f^4 + (B*a*((32*(13*B^3*a^2*b^4*d^9*f^2 + B^3*a^4*b^2*d^9*f^2 + 7*B^3*a*b^5*c*d^8*f^2 - 5*B^3*a^3*b^3*c*d^8*f^2))/f^5 - (B*a*((32*(c + d*tan(e + f*x))^(1/2)*(20*B^2*a^3*b^4*d^9*f^2 + 2*B^2*a^5*b^2*d^9*f^2 - 14*B^2*a*b^6*d^9*f^2 - 10*B^2*b^7*c*d^8*f^2 + 12*B^2*a^2*b^5*c*d^8*f^2 - 10*B^2*a^4*b^3*c*d^8*f^2))/f^4 - (B*a*((32*(12*B*a*b^7*d^10*f^4 + 4*B*b^8*c*d^9*f^4 + 24*B*a^3*b^5*d^10*f^4 + 12*B*a^5*b^3*d^10*f^4 + 4*B*a*b^7*c^2*d^8*f^4 + 4*B*a^2*b^6*c*d^9*f^4 - 4*B*a^4*b^4*c*d^9*f^4 - 4*B*a^6*b^2*c*d^9*f^4 + 8*B*a^3*b^5*c^2*d^8*f^4 + 4*B*a^5*b^3*c^2*d^8*f^4))/f^5 + (32*B*a*(c + d*tan(e + f*x))^(1/2)*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/(f^4*(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2)))*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2

*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2))/(a^5*d*f^2 - b
^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2))*
(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^
2 - a^5*b*d*f^2)^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^
2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2))*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^
2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2))/(a^5*d*f^2 -
b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2))
*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f
^2 - a^5*b*d*f^2)^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b
^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2) - (B*a*((32*(B^4*b^5*d^8 - 2*B^4*a^2*
b^3*d^8)*(c + d*tan(e + f*x))^(1/2))/f^4 - (B*a*((32*(13*B^3*a^2*b^4*d^9*f^
2 + B^3*a^4*b^2*d^9*f^2 + 7*B^3*a*b^5*c*d^8*f^2 - 5*B^3*a^3*b^3*c*d^8*f^2))
/f^5 + (B*a*((32*(c + d*tan(e + f*x))^(1/2)*(20*B^2*a^3*b^4*d^9*f^2 + 2*B^2
*a^5*b^2*d^9*f^2 - 14*B^2*a*b^6*d^9*f^2 - 10*B^2*b^7*c*d^8*f^2 + 12*B^2*a^2
*b^5*c*d^8*f^2 - 10*B^2*a^4*b^3*c*d^8*f^2))/f^4 + (B*a*((32*(12*B*a*b^7*d^1
0*f^4 + 4*B*b^8*c*d^9*f^4 + 24*B*a^3*b^5*d^10*f^4 + 12*B*a^5*b^3*d^10*f^4 +
 4*B*a*b^7*c^2*d^8*f^4 + 4*B*a^2*b^6*c*d^9*f^4 - 4*B*a^4*b^4*c*d^9*f^4 - 4*
B*a^6*b^2*c*d^9*f^4 + 8*B*a^3*b^5*c^2*d^8*f^4 + 4*B*a^5*b^3*c^2*d^8*f^4))/f
^5 - (32*B*a*(c + d*tan(e + f*x))^(1/2)*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*
b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2)*(16*b^9*d^10
*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24
*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b
^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^
9*f^4 + 8*a^7*b^2*c*d^9*f^4))/(f^4*(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2
 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2)))*(b^6*c*f^2 + 2*a^2*b^4*c*
f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2))/(
a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a
*b^4*d*f^2))*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2
 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f
^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2))*(b^6*c*f^2 + 2*a^2*b^4*c
*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2))/
(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 +
a*b^4*d*f^2))*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^
2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*
f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2) + (64*B^5*a*b^3*d^8)/f^5
))*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d
*f^2 - a^5*b*d*f^2)^(1/2)*2i)/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*
a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2)

3.115 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}}\,dx$

Optimal result 1528
Rubi [A] (verified) 1529
Mathematica [A] (verified) 1532
Maple [B] (verified) 1533
Fricas [F(-1)] 1533
Sympy [F] 1533
Maxima [F(-2)] 1534
Giac [F(-1)] 1534
Mupad [B] (verification not implemented) 1534

Optimal result

Integrand size = 47, antiderivative size = 327

$$\begin{aligned}
&\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}}\,dx \\
&= -\frac{(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(a-ib)^2\sqrt{c-id}f} - \frac{(B-i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(a+ib)^2\sqrt{c+id}f} \\
&\quad - \frac{(3a^3bBd-a^4Cd+b^4(2Bc-Ad)+ab^3(4Ac-4cC-Bd)-a^2b^2(2Bc+5Ad-3Cd))\operatorname{arctanh}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{\sqrt{b}\,(a^2+b^2)^2\,(bc-ad)^{3/2}f} \\
&\quad - \frac{(Ab^2-a(bB-aC))\sqrt{c+d\tan(e+fx)}}{(a^2+b^2)\,(bc-ad)f(a+b\tan(e+fx))}
\end{aligned}$$

```
[Out] -(3*a^3*b*B*d-a^4*C*d+b^4*(-A*d+2*B*c)+a*b^3*(4*A*c-B*d-4*C*c)-a^2*b^2*(5*A
*d+2*B*c-3*C*d))*arctanh(b^(1/2)*(c+d*tan(f*x+e))^(1/2)/(-a*d+b*c)^(1/2))/(
a^2+b^2)^2/(-a*d+b*c)^(3/2)/f/b^(1/2)-(I*A+B-I*C)*arctanh((c+d*tan(f*x+e))^
(1/2)/(c-I*d)^(1/2))/(a-I*b)^2/f/(c-I*d)^(1/2)-(B-I*(A-C))*arctanh((c+d*tan
(f*x+e))^(1/2)/(c+I*d)^(1/2))/(a+I*b)^2/f/(c+I*d)^(1/2)-(A*b^2-a*(B*b-C*a))
*(c+d*tan(f*x+e))^(1/2)/(a^2+b^2)/(-a*d+b*c)/f/(a+b*tan(f*x+e))
```

Rubi [A] (verified)

Time = 1.49 (sec) , antiderivative size = 327, normalized size of antiderivative = 1.00, number of steps used = 12, number of rules used = 7, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.149, Rules used = {3730, 3734, 3620, 3618, 65, 214, 3715}

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}}\,dx=-\frac{(Ab^2-a(bB-aC))\sqrt{c+d\tan(e+fx)}}{f\,(a^2+b^2)\,(bc-ad)(a+b\tan(e+fx))}$$
$$-\frac{(a^4(-C)d+3a^3bBd-a^2b^2(5Ad+2Bc-3Cd)+ab^3(4Ac-Bd-4cC)+b^4(2Bc-Ad))\operatorname{arctanh}\left(\frac{\sqrt{b}\cdots}{\cdots}\right)}{\sqrt{b}f\,(a^2+b^2)^2\,(bc-ad)^{3/2}}$$
$$-\frac{(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f(a-ib)^2\sqrt{c-id}}-\frac{(B-i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f(a+ib)^2\sqrt{c+id}}$$

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^2*Sqrt[c
+ d*Tan[e + f*x]]),x]
```

```
[Out] -(((I*A + B - I*C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]])/((a - I
*b)^2*Sqrt[c - I*d]*f)) - ((B - I*(A - C))*ArcTanh[Sqrt[c + d*Tan[e + f*x]]
/Sqrt[c + I*d]])/((a + I*b)^2*Sqrt[c + I*d]*f) - ((3*a^3*b*B*d - a^4*C*d +
b^4*(2*B*c - A*d) + a*b^3*(4*A*c - 4*c*C - B*d) - a^2*b^2*(2*B*c + 5*A*d -
3*C*d))*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*d]])/(Sqrt[
b]*(a^2 + b^2)^2*(b*c - a*d)^(3/2)*f) - ((A*b^2 - a*(b*B - a*C))*Sqrt[c + d
*Tan[e + f*x]])/((a^2 + b^2)*(b*c - a*d)*f*(a + b*Tan[e + f*x]))
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3715

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_) + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :>
 Dist[A/f, Subst[Int[(a + b*x)^m*(c + d*x)^n, x], x, Tan[e + f*x]], x] /; F
reeQ[{a, b, c, d, e, f, A, C, m, n}, x] && EqQ[A, C]
```

Rule 3730

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 - a*(b*B - a*C))*(a + b*Tan[e +
 f*x])^(m + 1)*((c + d*Tan[e + f*x])^(n + 1)/(f*(m + 1)*(b*c - a*d)*(a^2 +
b^2))), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3734

```
Int[(((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2))/((a_.) + (b_.)*tan[(e_.)
+ (f_.)*(x_)]), x_Symbol] :> Dist[1/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n
*Simp[b*B + a*(A - C) + (a*B - b*(A - C))*Tan[e + f*x], x], x], x] + Dist[(
A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n*((1 + Tan[e
+ f*x]^2)/(a + b*Tan[e + f*x])), x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C
, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&
 !GtQ[n, 0] &&  !LeQ[n, -1]
```

Rubi steps

$$\text{integral} = -\frac{(Ab^2-a(bB-aC))\sqrt{c+d\tan(e+fx)}}{(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))} - \frac{\int \frac{\frac{1}{2}\left(Ab^2d-2aA(bc-ad)-2(bB-aC)\left(bc-\frac{ad}{2}\right)\right)+(Ab-aB-bC)(bc-ad)\tan(e+fx)+\frac{1}{2}(Ab^2-a(bB-aC))d\tan^2(e+fx)}{(a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}}\,dx}{(a^2+b^2)(bc-ad)}$$

$$= -\frac{(Ab^2 - a(bB - aC))\sqrt{c + d\tan(e + fx)}}{(a^2 + b^2)(bc - ad)f(a + b\tan(e + fx))} - \frac{\int \frac{-((2abB + a^2(A-C) - b^2(A-C))(bc-ad)) - (a^2B - b^2B - 2ab(A-C))(bc-ad)\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{(a^2+b^2)^2(bc-ad)} + \frac{(3a^3bBd - a^4Cd + b^4(2Bc - Ad) + ab^3(4Ac - 4cC - Bd) - a^2b^2(2Bc + 5Ad - 3Cd))\int \frac{}{(a+b\tan}}{2(a^2+b^2)^2(bc-ad)}$$

$$= -\frac{(Ab^2 - a(bB - aC))\sqrt{c + d\tan(e + fx)}}{(a^2 + b^2)(bc - ad)f(a + b\tan(e + fx))} + \frac{(A - iB - C)\int \frac{1 + i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(a - ib)^2} + \frac{(A + iB - C)\int \frac{1 - i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(a + ib)^2} + \frac{(3a^3bBd - a^4Cd + b^4(2Bc - Ad) + ab^3(4Ac - 4cC - Bd) - a^2b^2(2Bc + 5Ad - 3Cd))\,\mathrm{Subst}\Big(\int}{2(a^2+b^2)^2(bc-ad)f}$$

$$= -\frac{(Ab^2 - a(bB - aC))\sqrt{c + d\tan(e + fx)}}{(a^2 + b^2)(bc - ad)f(a + b\tan(e + fx))} - \frac{(i(A + iB - C))\mathrm{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c+idx}}\,dx, x, -i\tan(e + fx)\right)}{2(a + ib)^2 f} + \frac{(iA + B - iC)\mathrm{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c-idx}}\,dx, x, i\tan(e + fx)\right)}{2(a - ib)^2 f} + \frac{(3a^3bBd - a^4Cd + b^4(2Bc - Ad) + ab^3(4Ac - 4cC - Bd) - a^2b^2(2Bc + 5Ad - 3Cd))\,\mathrm{Subst}\Big(\int}{(a^2+b^2)^2 d(bc-ad)f}$$

$$= -\frac{(3a^3bBd - a^4Cd + b^4(2Bc - Ad) + ab^3(4Ac - 4cC - Bd) - a^2b^2(2Bc + 5Ad - 3Cd))\,\mathrm{arctanh}}{\sqrt{b}(a^2+b^2)^2(bc-ad)^{3/2}f} - \frac{(Ab^2 - a(bB - aC))\sqrt{c + d\tan(e + fx)}}{(a^2 + b^2)(bc - ad)f(a + b\tan(e + fx))} - \frac{(A - iB - C)\mathrm{Subst}\left(\int \frac{1}{-1 - \frac{ic}{d} + \frac{ix^2}{d}}\,dx, x, \sqrt{c + d\tan(e + fx)}\right)}{(a - ib)^2 df} - \frac{(A + iB - C)\mathrm{Subst}\left(\int \frac{1}{-1 + \frac{ic}{d} - \frac{ix^2}{d}}\,dx, x, \sqrt{c + d\tan(e + fx)}\right)}{(a + ib)^2 df}$$

$$= -\frac{(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(a-ib)^2\sqrt{c-id}f} - \frac{(B-i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(a+ib)^2\sqrt{c+id}f}$$

$$-\frac{(3a^3bBd - a^4Cd + b^4(2Bc-Ad) + ab^3(4Ac-4cC-Bd) - a^2b^2(2Bc+5Ad-3Cd))\operatorname{arctanh}\Big(}{\sqrt{b}\,(a^2+b^2)^2\,(bc-ad)^{3/2}f}$$

$$-\frac{(Ab^2-a(bB-aC))\sqrt{c+d\tan(e+fx)}}{(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))}$$

Mathematica [A] (verified)

Time = 6.26 (sec) , antiderivative size = 521, normalized size of antiderivative = 1.59

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}}\,dx =$$

$$-\frac{\frac{i\sqrt{c-id}\left(i\left(a^2B-b^2B-2ab(A-C)\right)(bc-ad)-\left(2abB+a^2(A-C)-b^2(A-C)\right)(bc-ad)\right)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(-c+id)f} - \frac{i\sqrt{c+id}\left(-i\left(a^2B-b^2B-2ab(A-C)\right)(bc}{}}{a^2+b^2}$$

$$-\frac{(Ab^2-a(bB-aC))\sqrt{c+d\tan(e+fx)}}{(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))}$$

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^2*S
qrt[c + d*Tan[e + f*x]]),x]
```

```
[Out] -(((((I*Sqrt[c - I*d]*(I*(a^2*B - b^2*B - 2*a*b*(A - C))*(b*c - a*d) - (2*a*
b*B + a^2*(A - C) - b^2*(A - C))*(b*c - a*d))*ArcTanh[Sqrt[c + d*Tan[e + f*
x]]/Sqrt[c - I*d]])/((-c + I*d)*f) - (I*Sqrt[c + I*d]*((-I)*(a^2*B - b^2*B
- 2*a*b*(A - C))*(b*c - a*d) - (2*a*b*B + a^2*(A - C) - b^2*(A - C))*(b*c -
 a*d))*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]])/((-c - I*d)*f))/(a^
2 + b^2) + (2*Sqrt[b*c - a*d]*((a^2*(A*b^2 - a*(b*B - a*C))*d)/2 - a*b*(A*b
 - a*B - b*C)*(b*c - a*d) + (b^2*(A*b^2*d - 2*a*A*(b*c - a*d) - 2*(b*B - a*
C)*(b*c - (a*d)/2)))/2)*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c
 - a*d]])/(Sqrt[b]*(a^2 + b^2)*(-(b*c) + a*d)*f))/((a^2 + b^2)*(b*c - a*d))
) - ((A*b^2 - a*(b*B - a*C))*Sqrt[c + d*Tan[e + f*x]])/((a^2 + b^2)*(b*c -
a*d)*f*(a + b*Tan[e + f*x]))
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 20869 vs. 2(294) = 588.

Time = 0.16 (sec) , antiderivative size = 20870, normalized size of antiderivative = 63.82

method	result	size
derivativedivides	Expression too large to display	20870
default	Expression too large to display	20870

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f*x+e))
^2,x,method=_RETURNVERBOSE)
```

```
[Out] result too large to display
```

Fricas [F(-1)]

Timed out.

$$\int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{(a + b\tan(e + fx))^2\sqrt{c + d\tan(e + fx)}}\,dx = \text{Timed out}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f
*x+e))^2,x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{(a + b\tan(e + fx))^2\sqrt{c + d\tan(e + fx)}}\,dx = \int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{(a + b\tan(e + fx))^2\sqrt{c + d\tan(e + fx)}}\,dx$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x+e))**(1/2)/(a+b*tan
(f*x+e))**2,x)
```

```
[Out] Integral((A + B*tan(e + f*x) + C*tan(e + f*x)**2)/((a + b*tan(e + f*x))**2*
sqrt(c + d*tan(e + f*x))), x)
```

Maxima [F(-2)]

Exception generated.

$$\int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{(a + b\tan(e + fx))^2\sqrt{c + d\tan(e + fx)}}\,dx = \text{Exception raised: ValueError}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f
*x+e))^2,x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 detail
```

Giac [F(-1)]

Timed out.

$$\int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{(a + b\tan(e + fx))^2\sqrt{c + d\tan(e + fx)}}\,dx = \text{Timed out}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f
*x+e))^2,x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [B] (verification not implemented)

Time = 57.47 (sec) , antiderivative size = 225004, normalized size of antiderivative = 688.09

$$\int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{(a + b\tan(e + fx))^2\sqrt{c + d\tan(e + fx)}}\,dx = \text{Too large to display}$$

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))^2*(c + d*
tan(e + f*x))^(1/2)),x)
```

```
[Out] (atan(((((((16*(8*C^3*a^6*b^7*d^11*f^2 - 78*C^3*a^4*b^9*d^11*f^2 + 60*C^3*a
^8*b^5*d^11*f^2 - 24*C^3*a^10*b^3*d^11*f^2 + 2*C^3*a^12*b*d^11*f^2 - 32*C^3
*a*b^12*c^3*d^8*f^2 + 152*C^3*a^3*b^10*c*d^10*f^2 + 128*C^3*a^5*b^8*c*d^10*
f^2 - 64*C^3*a^7*b^6*c*d^10*f^2 - 32*C^3*a^9*b^4*c*d^10*f^2 + 8*C^3*a^11*b^
2*c*d^10*f^2 - 40*C^3*a^2*b^11*c^2*d^9*f^2 + 64*C^3*a^3*b^10*c^3*d^8*f^2 -
216*C^3*a^4*b^9*c^2*d^9*f^2 + 96*C^3*a^5*b^8*c^3*d^8*f^2 - 120*C^3*a^6*b^7*
c^2*d^9*f^2 + 56*C^3*a^8*b^5*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4
*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5
```

+ a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f
^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d
*f^5 - 8*a^7*b^3*c*d*f^5) - (((((16*(40*C*a^3*b^14*d^12*f^4 + 192*C*a^5*b^1
2*d^12*f^4 + 360*C*a^7*b^10*d^12*f^4 + 320*C*a^9*b^8*d^12*f^4 + 120*C*a^11*
b^6*d^12*f^4 - 8*C*a^15*b^2*d^12*f^4 + 8*C*b^17*c^3*d^9*f^4 + 40*C*a*b^16*c
^2*d^10*f^4 + 32*C*a*b^16*c^4*d^8*f^4 - 88*C*a^2*b^15*c*d^11*f^4 - 448*C*a^
4*b^13*c*d^11*f^4 - 920*C*a^6*b^11*c*d^11*f^4 - 960*C*a^8*b^9*c*d^11*f^4 -
520*C*a^10*b^7*c*d^11*f^4 - 128*C*a^12*b^5*c*d^11*f^4 - 8*C*a^14*b^3*c*d^11
*f^4 - 32*C*a^2*b^15*c^3*d^9*f^4 + 256*C*a^3*b^14*c^2*d^10*f^4 + 160*C*a^3*
b^14*c^4*d^8*f^4 - 280*C*a^4*b^13*c^3*d^9*f^4 + 680*C*a^5*b^12*c^2*d^10*f^4
+ 320*C*a^5*b^12*c^4*d^8*f^4 - 640*C*a^6*b^11*c^3*d^9*f^4 + 960*C*a^7*b^10
*c^2*d^10*f^4 + 320*C*a^7*b^10*c^4*d^8*f^4 - 680*C*a^8*b^9*c^3*d^9*f^4 + 76
0*C*a^9*b^8*c^2*d^10*f^4 + 160*C*a^9*b^8*c^4*d^8*f^4 - 352*C*a^10*b^7*c^3*d
^9*f^4 + 320*C*a^11*b^6*c^2*d^10*f^4 + 32*C*a^11*b^6*c^4*d^8*f^4 - 72*C*a^1
2*b^5*c^3*d^9*f^4 + 56*C*a^13*b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f
^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^
2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2
*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b
^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - (16*((C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 +
9*C^2*a^4*b^4*d^2 - 6*C^2*a^6*b^2*d^2 - 24*C^2*a^3*b^5*c*d + 8*C^2*a^5*b^3*
c*d)*(b^12*c^3*f^2 - a^11*b*d^3*f^2 + 4*a^2*b^10*c^3*f^2 + 6*a^4*b^8*c^3*f^
2 + 4*a^6*b^6*c^3*f^2 + a^8*b^4*c^3*f^2 - a^3*b^9*d^3*f^2 - 4*a^5*b^7*d^3*f
^2 - 6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*d^3*f^2 - 3*a*b^11*c^2*d*f^2 + 3*a^2*b^1
0*c*d^2*f^2 - 12*a^3*b^9*c^2*d*f^2 + 12*a^4*b^8*c*d^2*f^2 - 18*a^5*b^7*c^2*
d*f^2 + 18*a^6*b^6*c*d^2*f^2 - 12*a^7*b^5*c^2*d*f^2 + 12*a^8*b^4*c*d^2*f^2
- 3*a^9*b^3*c^2*d*f^2 + 3*a^10*b^2*c*d^2*f^2))^(1/2)*(c + d*tan(e + f*x))^(
1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4
+ 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 1
60*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^1
9*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*
a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4
- 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*
d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^
11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48
*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^
4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d
^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c
^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c
*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10
*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15
*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((b^10*(8*a^2*c^3*f^2 + 6*a^2*c*
d^2*f^2) + b^4*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^8*(12*a^4*c^3*f^2 + 2
4*a^4*c*d^2*f^2) + b^6*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^3*(8*a^9*d^3*
f^2 + 6*a^9*c^2*d*f^2) - b^9*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^5*(12*a
^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^7*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) + 2

*b^12*c^3*f^2 - 2*a^11*b*d^3*f^2 - 6*a*b^11*c^2*d*f^2 + 6*a^10*b^2*c*d^2*f^2)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4)))*((C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 9*C^2*a^4*b^4*d^2 - 6*C^2*a^6*b^2*d^2 - 24*C^2*a^3*b^5*c*d + 8*C^2*a^5*b^3*c*d)*(b^12*c^3*f^2 - a^11*b*d^3*f^2 + 4*a^2*b^10*c^3*f^2 + 6*a^4*b^8*c^3*f^2 + 4*a^6*b^6*c^3*f^2 + a^8*b^4*c^3*f^2 - a^3*b^9*d^3*f^2 - 4*a^5*b^7*d^3*f^2 - 6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*d^3*f^2 - 3*a*b^11*c^2*d*f^2 + 3*a^2*b^10*c*d^2*f^2 - 12*a^3*b^9*c^2*d*f^2 + 12*a^4*b^8*c*d^2*f^2 - 18*a^5*b^7*c^2*d*f^2 + 18*a^6*b^6*c*d^2*f^2 - 12*a^7*b^5*c^2*d*f^2 + 12*a^8*b^4*c*d^2*f^2 - 3*a^9*b^3*c^2*d*f^2 + 3*a^10*b^2*c*d^2*f^2))^(1/2))/(b^10*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^4*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^8*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^6*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^3*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^9*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^5*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^7*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) + 2*b^12*c^3*f^2 - 2*a^11*b*d^3*f^2 - 6*a*b^11*c^2*d*f^2 + 6*a^10*b^2*c*d^2*f^2) + (16*(c + d*tan(e + f*x))^(1/2)*(20*C^2*a^5*b^10*d^11*f^2 - 60*C^2*a^3*b^12*d^11*f^2 + 168*C^2*a^7*b^8*d^11*f^2 + 40*C^2*a^9*b^6*d^11*f^2 - 44*C^2*a^11*b^4*d^11*f^2 + 4*C^2*a^13*b^2*d^11*f^2 - 20*C^2*b^15*c^3*d^8*f^2 - 4*C^2*a^14*b*c*d^10*f^2 - 20*C^2*a*b^14*c^2*d^9*f^2 + 100*C^2*a^2*b^13*c*d^10*f^2 - 300*C^2*a^6*b^9*c*d^10*f^2 - 160*C^2*a^8*b^7*c*d^10*f^2 + 76*C^2*a^10*b^5*c*d^10*f^2 + 32*C^2*a^12*b^3*c*d^10*f^2 + 116*C^2*a^2*b^13*c^3*d^8*f^2 - 124*C^2*a^3*b^12*c^2*d^9*f^2 + 216*C^2*a^4*b^11*c^3*d^8*f^2 - 40*C^2*a^5*b^10*c^2*d^9*f^2 + 8*C^2*a^6*b^9*c^3*d^8*f^2 + 168*C^2*a^7*b^8*c^2*d^9*f^2 - 68*C^2*a^8*b^7*c^3*d^8*f^2 + 60*C^2*a^9*b^6*c^2*d^9*f^2 + 4*C^2*a^10*b^5*c^3*d^8*f^2 - 44*C^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))*((C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 9*C^2*a^4*b^4*d^2 - 6*C^2*a^6*b^2*d^2 - 24*C^2*a^3*b^5*c*d + 8*C^2*a^5*b^3*c*d)*(b^12*c^3*f^2 - a^11*b*d^3*f^2 + 4*a^2*b^10*c^3*f^2 + 6*a^4*b^8*c^3*f^2 + 4*a^6*b^6*c^3*f^2 + a^8*b^4*c^3*f^2 - a^3*b^9*d^3*f^2 - 4*a^5*b^7*d^3*f^2 - 6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*d^3*f^2 - 3*a*b^11*c^2*d*f^2 + 3*a^2*b^10*c*d^2*f^2 - 12*a^3*b^9*c^2*d*f^2 + 12*a^4*b^8*c*d^2*f^2 - 18*a^5*b^7*c^2*d*f^2 + 18*a^6*b^6*c*d^2*f^2 - 12*a^7*b^5*c^2*d*f^2 + 12*a^8*b^4*c*d^2*f^2 - 3*a^9*b^3*c^2*d*f^2 + 3*a^10*b^2*c*d^2*f^2))^(1/2))/(b^10*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^4*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^8*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^6*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^3*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^9*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^5*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^7*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) + 2*b^12*c^3*f^2 - 2*a^11*b*d^3*f^2 - 6*a*b^11*c^2*d*f^2 + 6*a^10*b^2*c*d^2*f^2))*((C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 9*C^2*a^4*b^4*d^2 - 6*C^2*a^6*b^2*d^2 - 24*C^2*a^3*b^5*c*d + 8*C^2*a^5*b

^3*c*d)*(b^12*c^3*f^2 - a^11*b*d^3*f^2 + 4*a^2*b^10*c^3*f^2 + 6*a^4*b^8*c^3
*f^2 + 4*a^6*b^6*c^3*f^2 + a^8*b^4*c^3*f^2 - a^3*b^9*d^3*f^2 - 4*a^5*b^7*d^
3*f^2 - 6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*d^3*f^2 - 3*a*b^11*c^2*d*f^2 + 3*a^2*
b^10*c*d^2*f^2 - 12*a^3*b^9*c^2*d*f^2 + 12*a^4*b^8*c*d^2*f^2 - 18*a^5*b^7*c
^2*d*f^2 + 18*a^6*b^6*c*d^2*f^2 - 12*a^7*b^5*c^2*d*f^2 + 12*a^8*b^4*c*d^2*f
^2 - 3*a^9*b^3*c^2*d*f^2 + 3*a^10*b^2*c*d^2*f^2))^(1/2))/(b^10*(8*a^2*c^3*f
^2 + 6*a^2*c*d^2*f^2) + b^4*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^8*(12*a^
4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^6*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^
3*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^9*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2
) - b^5*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^7*(8*a^5*d^3*f^2 + 36*a^5*c
^2*d*f^2) + 2*b^12*c^3*f^2 - 2*a^11*b*d^3*f^2 - 6*a*b^11*c^2*d*f^2 + 6*a^10
*b^2*c*d^2*f^2) + (16*(c + d*tan(e + f*x))^(1/2)*(2*C^4*a^2*b^9*d^10 - 5*C^
4*a^4*b^7*d^10 + 17*C^4*a^6*b^5*d^10 - 7*C^4*a^8*b^3*d^10 + 2*C^4*b^11*c^2*
d^8 + C^4*a^10*b*d^10 - 12*C^4*a^2*b^9*c^2*d^8 + 18*C^4*a^4*b^7*c^2*d^8 - 4
*C^4*a*b^10*c*d^9 + 16*C^4*a^3*b^8*c*d^9 - 36*C^4*a^5*b^6*c*d^9 + 8*C^4*a^7
*b^4*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c
^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*
d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b
*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))*((C
^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 9*C^2*a^4*b^4*d^2 - 6*C^2*a^6*b^2*d^2 - 2
4*C^2*a^3*b^5*c*d + 8*C^2*a^5*b^3*c*d)*(b^12*c^3*f^2 - a^11*b*d^3*f^2 + 4*a
^2*b^10*c^3*f^2 + 6*a^4*b^8*c^3*f^2 + 4*a^6*b^6*c^3*f^2 + a^8*b^4*c^3*f^2 -
a^3*b^9*d^3*f^2 - 4*a^5*b^7*d^3*f^2 - 6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*d^3*f^
2 - 3*a*b^11*c^2*d*f^2 + 3*a^2*b^10*c*d^2*f^2 - 12*a^3*b^9*c^2*d*f^2 + 12*a
^4*b^8*c*d^2*f^2 - 18*a^5*b^7*c^2*d*f^2 + 18*a^6*b^6*c*d^2*f^2 - 12*a^7*b^5
*c^2*d*f^2 + 12*a^8*b^4*c*d^2*f^2 - 3*a^9*b^3*c^2*d*f^2 + 3*a^10*b^2*c*d^2*
f^2))^(1/2)*1i)/(b^10*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^4*(2*a^8*c^3*f^
2 + 24*a^8*c*d^2*f^2) + b^8*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^6*(8*a^
6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^3*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^9
*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^5*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^
2) - b^7*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) + 2*b^12*c^3*f^2 - 2*a^11*b*d^3
*f^2 - 6*a*b^11*c^2*d*f^2 + 6*a^10*b^2*c*d^2*f^2) - (((((16*(8*C^3*a^6*b^7*
d^11*f^2 - 78*C^3*a^4*b^9*d^11*f^2 + 60*C^3*a^8*b^5*d^11*f^2 - 24*C^3*a^10*
b^3*d^11*f^2 + 2*C^3*a^12*b*d^11*f^2 - 32*C^3*a*b^12*c^3*d^8*f^2 + 152*C^3*
a^3*b^10*c*d^10*f^2 + 128*C^3*a^5*b^8*c*d^10*f^2 - 64*C^3*a^7*b^6*c*d^10*f^
2 - 32*C^3*a^9*b^4*c*d^10*f^2 + 8*C^3*a^11*b^2*c*d^10*f^2 - 40*C^3*a^2*b^11
*c^2*d^9*f^2 + 64*C^3*a^3*b^10*c^3*d^8*f^2 - 216*C^3*a^4*b^9*c^2*d^9*f^2 +
96*C^3*a^5*b^8*c^3*d^8*f^2 - 120*C^3*a^6*b^7*c^2*d^9*f^2 + 56*C^3*a^8*b^5*c
^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c
^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*
d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b
*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - ((
(((16*(40*C*a^3*b^14*d^12*f^4 + 192*C*a^5*b^12*d^12*f^4 + 360*C*a^7*b^10*d^
12*f^4 + 320*C*a^9*b^8*d^12*f^4 + 120*C*a^11*b^6*d^12*f^4 - 8*C*a^15*b^2*d^
12*f^4 + 8*C*b^17*c^3*d^9*f^4 + 40*C*a*b^16*c^2*d^10*f^4 + 32*C*a*b^16*c^4*

d^8*f^4 - 88*C*a^2*b^15*c*d^11*f^4 - 448*C*a^4*b^13*c*d^11*f^4 - 920*C*a^6*
b^11*c*d^11*f^4 - 960*C*a^8*b^9*c*d^11*f^4 - 520*C*a^10*b^7*c*d^11*f^4 - 12
8*C*a^12*b^5*c*d^11*f^4 - 8*C*a^14*b^3*c*d^11*f^4 - 32*C*a^2*b^15*c^3*d^9*f
^4 + 256*C*a^3*b^14*c^2*d^10*f^4 + 160*C*a^3*b^14*c^4*d^8*f^4 - 280*C*a^4*b
^13*c^3*d^9*f^4 + 680*C*a^5*b^12*c^2*d^10*f^4 + 320*C*a^5*b^12*c^4*d^8*f^4
- 640*C*a^6*b^11*c^3*d^9*f^4 + 960*C*a^7*b^10*c^2*d^10*f^4 + 320*C*a^7*b^10
*c^4*d^8*f^4 - 680*C*a^8*b^9*c^3*d^9*f^4 + 760*C*a^9*b^8*c^2*d^10*f^4 + 160
*C*a^9*b^8*c^4*d^8*f^4 - 352*C*a^10*b^7*c^3*d^9*f^4 + 320*C*a^11*b^6*c^2*d^
10*f^4 + 32*C*a^11*b^6*c^4*d^8*f^4 - 72*C*a^12*b^5*c^3*d^9*f^4 + 56*C*a^13*
b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4
*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^
4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2
*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5
) + (16*((C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 9*C^2*a^4*b^4*d^2 - 6*C^2*a^6*
b^2*d^2 - 24*C^2*a^3*b^5*c*d + 8*C^2*a^5*b^3*c*d)*(b^12*c^3*f^2 - a^11*b*d^
3*f^2 + 4*a^2*b^10*c^3*f^2 + 6*a^4*b^8*c^3*f^2 + 4*a^6*b^6*c^3*f^2 + a^8*b^
4*c^3*f^2 - a^3*b^9*d^3*f^2 - 4*a^5*b^7*d^3*f^2 - 6*a^7*b^5*d^3*f^2 - 4*a^9
*b^3*d^3*f^2 - 3*a*b^11*c^2*d*f^2 + 3*a^2*b^10*c*d^2*f^2 - 12*a^3*b^9*c^2*d
*f^2 + 12*a^4*b^8*c*d^2*f^2 - 18*a^5*b^7*c^2*d*f^2 + 18*a^6*b^6*c*d^2*f^2 -
12*a^7*b^5*c^2*d*f^2 + 12*a^8*b^4*c*d^2*f^2 - 3*a^9*b^3*c^2*d*f^2 + 3*a^10
*b^2*c*d^2*f^2))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 1
60*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*
a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16
*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c
^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4
*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 +
112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9
*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*
c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^
11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 +
208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*
f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11
*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^1
1*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*
d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*
c*d^11*f^4))/((b^10*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^4*(2*a^8*c^3*f^2
+ 24*a^8*c*d^2*f^2) + b^8*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^6*(8*a^6*
c^3*f^2 + 36*a^6*c*d^2*f^2) - b^3*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^9*(
2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^5*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2)
- b^7*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) + 2*b^12*c^3*f^2 - 2*a^11*b*d^3*f
^2 - 6*a*b^11*c^2*d*f^2 + 6*a^10*b^2*c*d^2*f^2)*(a^10*d^2*f^4 + b^10*c^2*f^
4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2
*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*
d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^
5*c*d*f^4 - 8*a^7*b^3*c*d*f^4)))*((C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 9*C^2

*a^4*b^4*d^2 - 6*C^2*a^6*b^2*d^2 - 24*C^2*a^3*b^5*c*d + 8*C^2*a^5*b^3*c*d)*
(b^12*c^3*f^2 - a^11*b*d^3*f^2 + 4*a^2*b^10*c^3*f^2 + 6*a^4*b^8*c^3*f^2 + 4
*a^6*b^6*c^3*f^2 + a^8*b^4*c^3*f^2 - a^3*b^9*d^3*f^2 - 4*a^5*b^7*d^3*f^2 -
6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*d^3*f^2 - 3*a*b^11*c^2*d*f^2 + 3*a^2*b^10*c*d
^2*f^2 - 12*a^3*b^9*c^2*d*f^2 + 12*a^4*b^8*c*d^2*f^2 - 18*a^5*b^7*c^2*d*f^2
 + 18*a^6*b^6*c*d^2*f^2 - 12*a^7*b^5*c^2*d*f^2 + 12*a^8*b^4*c*d^2*f^2 - 3*a
^9*b^3*c^2*d*f^2 + 3*a^10*b^2*c*d^2*f^2))^(1/2))/(b^10*(8*a^2*c^3*f^2 + 6*a
^2*c*d^2*f^2) + b^4*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^8*(12*a^4*c^3*f^
2 + 24*a^4*c*d^2*f^2) + b^6*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^3*(8*a^9
*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^9*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^5*
(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^7*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2
) + 2*b^12*c^3*f^2 - 2*a^11*b*d^3*f^2 - 6*a*b^11*c^2*d*f^2 + 6*a^10*b^2*c*d
^2*f^2) - (16*(c + d*tan(e + f*x))^(1/2)*(20*C^2*a^5*b^10*d^11*f^2 - 60*C^2
*a^3*b^12*d^11*f^2 + 168*C^2*a^7*b^8*d^11*f^2 + 40*C^2*a^9*b^6*d^11*f^2 - 4
4*C^2*a^11*b^4*d^11*f^2 + 4*C^2*a^13*b^2*d^11*f^2 - 20*C^2*b^15*c^3*d^8*f^2
 - 4*C^2*a^14*b*c*d^10*f^2 - 20*C^2*a*b^14*c^2*d^9*f^2 + 100*C^2*a^2*b^13*c
*d^10*f^2 - 300*C^2*a^6*b^9*c*d^10*f^2 - 160*C^2*a^8*b^7*c*d^10*f^2 + 76*C^
2*a^10*b^5*c*d^10*f^2 + 32*C^2*a^12*b^3*c*d^10*f^2 + 116*C^2*a^2*b^13*c^3*d
^8*f^2 - 124*C^2*a^3*b^12*c^2*d^9*f^2 + 216*C^2*a^4*b^11*c^3*d^8*f^2 - 40*C
^2*a^5*b^10*c^2*d^9*f^2 + 8*C^2*a^6*b^9*c^3*d^8*f^2 + 168*C^2*a^7*b^8*c^2*d
^9*f^2 - 68*C^2*a^8*b^7*c^3*d^8*f^2 + 60*C^2*a^9*b^6*c^2*d^9*f^2 + 4*C^2*a^
10*b^5*c^3*d^8*f^2 - 44*C^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2
*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*
c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b
^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5
*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))*((C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 9*C
^2*a^4*b^4*d^2 - 6*C^2*a^6*b^2*d^2 - 24*C^2*a^3*b^5*c*d + 8*C^2*a^5*b^3*c*d
)*(b^12*c^3*f^2 - a^11*b*d^3*f^2 + 4*a^2*b^10*c^3*f^2 + 6*a^4*b^8*c^3*f^2 +
 4*a^6*b^6*c^3*f^2 + a^8*b^4*c^3*f^2 - a^3*b^9*d^3*f^2 - 4*a^5*b^7*d^3*f^2
 - 6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*d^3*f^2 - 3*a*b^11*c^2*d*f^2 + 3*a^2*b^10*c
*d^2*f^2 - 12*a^3*b^9*c^2*d*f^2 + 12*a^4*b^8*c*d^2*f^2 - 18*a^5*b^7*c^2*d*f
^2 + 18*a^6*b^6*c*d^2*f^2 - 12*a^7*b^5*c^2*d*f^2 + 12*a^8*b^4*c*d^2*f^2 - 3
*a^9*b^3*c^2*d*f^2 + 3*a^10*b^2*c*d^2*f^2))^(1/2))/(b^10*(8*a^2*c^3*f^2 + 6
*a^2*c*d^2*f^2) + b^4*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^8*(12*a^4*c^3*
f^2 + 24*a^4*c*d^2*f^2) + b^6*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^3*(8*a
^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^9*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^
5*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^7*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f
^2) + 2*b^12*c^3*f^2 - 2*a^11*b*d^3*f^2 - 6*a*b^11*c^2*d*f^2 + 6*a^10*b^2*c
*d^2*f^2))*((C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 9*C^2*a^4*b^4*d^2 - 6*C^2*a
^6*b^2*d^2 - 24*C^2*a^3*b^5*c*d + 8*C^2*a^5*b^3*c*d)*(b^12*c^3*f^2 - a^11*b
*d^3*f^2 + 4*a^2*b^10*c^3*f^2 + 6*a^4*b^8*c^3*f^2 + 4*a^6*b^6*c^3*f^2 + a^8
*b^4*c^3*f^2 - a^3*b^9*d^3*f^2 - 4*a^5*b^7*d^3*f^2 - 6*a^7*b^5*d^3*f^2 - 4*
a^9*b^3*d^3*f^2 - 3*a*b^11*c^2*d*f^2 + 3*a^2*b^10*c*d^2*f^2 - 12*a^3*b^9*c^
2*d*f^2 + 12*a^4*b^8*c*d^2*f^2 - 18*a^5*b^7*c^2*d*f^2 + 18*a^6*b^6*c*d^2*f^
2 - 12*a^7*b^5*c^2*d*f^2 + 12*a^8*b^4*c*d^2*f^2 - 3*a^9*b^3*c^2*d*f^2 + 3*a

^10*b^2*c*d^2*f^2))^(1/2))/(b^10*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^4*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^8*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^6*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^3*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^9*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^5*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^7*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) + 2*b^12*c^3*f^2 - 2*a^11*b*d^3*f^2 - 6*a*b^11*c^2*d*f^2 + 6*a^10*b^2*c*d^2*f^2) - (16*(c + d*tan(e + f*x))^(1/2)*(2*C^4*a^2*b^9*d^10 - 5*C^4*a^4*b^7*d^10 + 17*C^4*a^6*b^5*d^10 - 7*C^4*a^8*b^3*d^10 + 2*C^4*b^11*c^2*d^8 + C^4*a^10*b*d^10 - 12*C^4*a^2*b^9*c^2*d^8 + 18*C^4*a^4*b^7*c^2*d^8 - 4*C^4*a*b^10*c*d^9 + 16*C^4*a^3*b^8*c*d^9 - 36*C^4*a^5*b^6*c*d^9 + 8*C^4*a^7*b^4*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))*((C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 9*C^2*a^4*b^4*d^2 - 6*C^2*a^6*b^2*d^2 - 24*C^2*a^3*b^5*c*d + 8*C^2*a^5*b^3*c*d)*(b^12*c^3*f^2 - a^11*b*d^3*f^2 + 4*a^2*b^10*c^3*f^2 + 6*a^4*b^8*c^3*f^2 + 4*a^6*b^6*c^3*f^2 + a^8*b^4*c^3*f^2 - a^3*b^9*d^3*f^2 - 4*a^5*b^7*d^3*f^2 - 6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*d^3*f^2 - 3*a*b^11*c^2*d*f^2 + 3*a^2*b^10*c*d^2*f^2 - 12*a^3*b^9*c^2*d*f^2 + 12*a^4*b^8*c*d^2*f^2 - 18*a^5*b^7*c^2*d*f^2 + 18*a^6*b^6*c*d^2*f^2 - 12*a^7*b^5*c^2*d*f^2 + 12*a^8*b^4*c*d^2*f^2 - 3*a^9*b^3*c^2*d*f^2 + 3*a^10*b^2*c*d^2*f^2))^(1/2)*1i)/(b^10*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^4*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^8*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^6*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^3*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^9*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^5*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^7*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) + 2*b^12*c^3*f^2 - 2*a^11*b*d^3*f^2 - 6*a*b^11*c^2*d*f^2 + 6*a^10*b^2*c*d^2*f^2))/((((((16*(8*C^3*a^6*b^7*d^11*f^2 - 78*C^3*a^4*b^9*d^11*f^2 + 60*C^3*a^8*b^5*d^11*f^2 - 24*C^3*a^10*b^3*d^11*f^2 + 2*C^3*a^12*b*d^11*f^2 - 32*C^3*a*b^12*c^3*d^8*f^2 + 152*C^3*a^3*b^10*c*d^10*f^2 + 128*C^3*a^5*b^8*c*d^10*f^2 - 64*C^3*a^7*b^6*c*d^10*f^2 - 32*C^3*a^9*b^4*c*d^10*f^2 + 8*C^3*a^11*b^2*c*d^10*f^2 - 40*C^3*a^2*b^11*c^2*d^9*f^2 + 64*C^3*a^3*b^10*c^3*d^8*f^2 - 216*C^3*a^4*b^9*c^2*d^9*f^2 + 96*C^3*a^5*b^8*c^3*d^8*f^2 - 120*C^3*a^6*b^7*c^2*d^9*f^2 + 56*C^3*a^8*b^5*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - (((((16*(40*C*a^3*b^14*d^12*f^4 + 192*C*a^5*b^12*d^12*f^4 + 360*C*a^7*b^10*d^12*f^4 + 320*C*a^9*b^8*d^12*f^4 + 120*C*a^11*b^6*d^12*f^4 - 8*C*a^15*b^2*d^12*f^4 + 8*C*b^17*c^3*d^9*f^4 + 40*C*a*b^16*c^2*d^10*f^4 + 32*C*a*b^16*c^4*d^8*f^4 - 88*C*a^2*b^15*c*d^11*f^4 - 448*C*a^4*b^13*c*d^11*f^4 - 920*C*a^6*b^11*c*d^11*f^4 - 960*C*a^8*b^9*c*d^11*f^4 - 520*C*a^10*b^7*c*d^11*f^4 - 128*C*a^12*b^5*c*d^11*f^4 - 8*C*a^14*b^3*c*d^11*f^4 - 32*C*a^2*b^15*c^3*d^9*f^4 + 256*C*a^3*b^14*c^2*d^10*f^4 + 160*C*a^3*b^14*c^4*d^8*f^4 - 280*C*a^4*b^13*c^3*d^9*f^4 + 680*C*a^5*b^12*c^2*d^10*f^4 + 320*C*a^5*b^12*c^4*d^8*f^4 - 640*C*a^6*b^11*c^3*d^9*f^4 +

960*C*a^7*b^10*c^2*d^10*f^4 + 320*C*a^7*b^10*c^4*d^8*f^4 - 680*C*a^8*b^9*c
^3*d^9*f^4 + 760*C*a^9*b^8*c^2*d^10*f^4 + 160*C*a^9*b^8*c^4*d^8*f^4 - 352*C
*a^10*b^7*c^3*d^9*f^4 + 320*C*a^11*b^6*c^2*d^10*f^4 + 32*C*a^11*b^6*c^4*d^8
*f^4 - 72*C*a^12*b^5*c^3*d^9*f^4 + 56*C*a^13*b^4*c^2*d^10*f^4))/(a^10*d^2*f
^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f
^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*
f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d
*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - (16*((C^2*a^8*d^2 + 16*C^2
*a^2*b^6*c^2 + 9*C^2*a^4*b^4*d^2 - 6*C^2*a^6*b^2*d^2 - 24*C^2*a^3*b^5*c*d +
8*C^2*a^5*b^3*c*d)*(b^12*c^3*f^2 - a^11*b*d^3*f^2 + 4*a^2*b^10*c^3*f^2 + 6
*a^4*b^8*c^3*f^2 + 4*a^6*b^6*c^3*f^2 + a^8*b^4*c^3*f^2 - a^3*b^9*d^3*f^2 -
4*a^5*b^7*d^3*f^2 - 6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*d^3*f^2 - 3*a*b^11*c^2*d*
f^2 + 3*a^2*b^10*c*d^2*f^2 - 12*a^3*b^9*c^2*d*f^2 + 12*a^4*b^8*c*d^2*f^2 -
18*a^5*b^7*c^2*d*f^2 + 18*a^6*b^6*c*d^2*f^2 - 12*a^7*b^5*c^2*d*f^2 + 12*a^8
*b^4*c*d^2*f^2 - 3*a^9*b^3*c^2*d*f^2 + 3*a^10*b^2*c*d^2*f^2))^(1/2)*(c + d*
tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6
*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b
^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^
10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4
*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b
^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 72
0*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f
^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^
2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*
b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304
*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4
- 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 -
304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4
+ 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11
*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((b^10*(8*a^2*c^3
*f^2 + 6*a^2*c*d^2*f^2) + b^4*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^8*(12*
a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^6*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) -
b^3*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^9*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f
^2) - b^5*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^7*(8*a^5*d^3*f^2 + 36*a^5
*c^2*d*f^2) + 2*b^12*c^3*f^2 - 2*a^11*b*d^3*f^2 - 6*a*b^11*c^2*d*f^2 + 6*a^
10*b^2*c*d^2*f^2)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*
b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4
*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*
a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4)
))*((C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 9*C^2*a^4*b^4*d^2 - 6*C^2*a^6*b^2*d
^2 - 24*C^2*a^3*b^5*c*d + 8*C^2*a^5*b^3*c*d)*(b^12*c^3*f^2 - a^11*b*d^3*f^2
+ 4*a^2*b^10*c^3*f^2 + 6*a^4*b^8*c^3*f^2 + 4*a^6*b^6*c^3*f^2 + a^8*b^4*c^3
*f^2 - a^3*b^9*d^3*f^2 - 4*a^5*b^7*d^3*f^2 - 6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*
d^3*f^2 - 3*a*b^11*c^2*d*f^2 + 3*a^2*b^10*c*d^2*f^2 - 12*a^3*b^9*c^2*d*f^2
+ 12*a^4*b^8*c*d^2*f^2 - 18*a^5*b^7*c^2*d*f^2 + 18*a^6*b^6*c*d^2*f^2 - 12*a

^7*b^5*c^2*d*f^2 + 12*a^8*b^4*c*d^2*f^2 - 3*a^9*b^3*c^2*d*f^2 + 3*a^10*b^2*c*d^2*f^2))^(1/2))/(b^10*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^4*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^8*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^6*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^3*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^9*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^5*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^7*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) + 2*b^12*c^3*f^2 - 2*a^11*b*d^3*f^2 - 6*a*b^11*c^2*d*f^2 + 6*a^10*b^2*c*d^2*f^2) + (16*(c + d*tan(e + f*x))^(1/2)*(20*C^2*a^5*b^10*d^11*f^2 - 60*C^2*a^3*b^12*d^11*f^2 + 168*C^2*a^7*b^8*d^11*f^2 + 40*C^2*a^9*b^6*d^11*f^2 - 44*C^2*a^11*b^4*d^11*f^2 + 4*C^2*a^13*b^2*d^11*f^2 - 20*C^2*b^15*c^3*d^8*f^2 - 4*C^2*a^14*b*c*d^10*f^2 - 20*C^2*a*b^14*c^2*d^9*f^2 + 100*C^2*a^2*b^13*c*d^10*f^2 - 300*C^2*a^6*b^9*c*d^10*f^2 - 160*C^2*a^8*b^7*c*d^10*f^2 + 76*C^2*a^10*b^5*c*d^10*f^2 + 32*C^2*a^12*b^3*c*d^10*f^2 + 116*C^2*a^2*b^13*c^3*d^8*f^2 - 124*C^2*a^3*b^12*c^2*d^9*f^2 + 216*C^2*a^4*b^11*c^3*d^8*f^2 - 40*C^2*a^5*b^10*c^2*d^9*f^2 + 8*C^2*a^6*b^9*c^3*d^8*f^2 + 168*C^2*a^7*b^8*c^2*d^9*f^2 - 68*C^2*a^8*b^7*c^3*d^8*f^2 + 60*C^2*a^9*b^6*c^2*d^9*f^2 + 4*C^2*a^10*b^5*c^3*d^8*f^2 - 44*C^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))*((C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 9*C^2*a^4*b^4*d^2 - 6*C^2*a^6*b^2*d^2 - 24*C^2*a^3*b^5*c*d + 8*C^2*a^5*b^3*c*d)*(b^12*c^3*f^2 - a^11*b*d^3*f^2 + 4*a^2*b^10*c^3*f^2 + 6*a^4*b^8*c^3*f^2 + 4*a^6*b^6*c^3*f^2 + a^8*b^4*c^3*f^2 - a^3*b^9*d^3*f^2 - 4*a^5*b^7*d^3*f^2 - 6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*d^3*f^2 - 3*a*b^11*c^2*d*f^2 + 3*a^2*b^10*c*d^2*f^2 - 12*a^3*b^9*c^2*d*f^2 + 12*a^4*b^8*c*d^2*f^2 - 18*a^5*b^7*c^2*d*f^2 + 18*a^6*b^6*c*d^2*f^2 - 12*a^7*b^5*c^2*d*f^2 + 12*a^8*b^4*c*d^2*f^2 - 3*a^9*b^3*c^2*d*f^2 + 3*a^10*b^2*c*d^2*f^2))^(1/2))/(b^10*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^4*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^8*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^6*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^3*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^9*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^5*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^7*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) + 2*b^12*c^3*f^2 - 2*a^11*b*d^3*f^2 - 6*a*b^11*c^2*d*f^2 + 6*a^10*b^2*c*d^2*f^2))*((C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 9*C^2*a^4*b^4*d^2 - 6*C^2*a^6*b^2*d^2 - 24*C^2*a^3*b^5*c*d + 8*C^2*a^5*b^3*c*d)*(b^12*c^3*f^2 - a^11*b*d^3*f^2 + 4*a^2*b^10*c^3*f^2 + 6*a^4*b^8*c^3*f^2 + 4*a^6*b^6*c^3*f^2 + a^8*b^4*c^3*f^2 - a^3*b^9*d^3*f^2 - 4*a^5*b^7*d^3*f^2 - 6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*d^3*f^2 - 3*a*b^11*c^2*d*f^2 + 3*a^2*b^10*c*d^2*f^2 - 12*a^3*b^9*c^2*d*f^2 + 12*a^4*b^8*c*d^2*f^2 - 18*a^5*b^7*c^2*d*f^2 + 18*a^6*b^6*c*d^2*f^2 - 12*a^7*b^5*c^2*d*f^2 + 12*a^8*b^4*c*d^2*f^2 - 3*a^9*b^3*c^2*d*f^2 + 3*a^10*b^2*c*d^2*f^2))^(1/2))/(b^10*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^4*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^8*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^6*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^3*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^9*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^5*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^7*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) + 2*b^12*c^3*f^2 - 2*a^11*b*d^3*f^2 - 6*a*b^11*c^2

*d*f^2 + 6*a^10*b^2*c*d^2*f^2) + (16*(c + d*tan(e + f*x))^(1/2)*(2*C^4*a^2*
b^9*d^10 - 5*C^4*a^4*b^7*d^10 + 17*C^4*a^6*b^5*d^10 - 7*C^4*a^8*b^3*d^10 +
2*C^4*b^11*c^2*d^8 + C^4*a^10*b*d^10 - 12*C^4*a^2*b^9*c^2*d^8 + 18*C^4*a^4*
b^7*c^2*d^8 - 4*C^4*a*b^10*c*d^9 + 16*C^4*a^3*b^8*c*d^9 - 36*C^4*a^5*b^6*c*
d^9 + 8*C^4*a^7*b^4*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^
4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f
^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*
d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^
3*c*d*f^4))*((C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 9*C^2*a^4*b^4*d^2 - 6*C^2*
a^6*b^2*d^2 - 24*C^2*a^3*b^5*c*d + 8*C^2*a^5*b^3*c*d)*(b^12*c^3*f^2 - a^11*
b*d^3*f^2 + 4*a^2*b^10*c^3*f^2 + 6*a^4*b^8*c^3*f^2 + 4*a^6*b^6*c^3*f^2 + a^
8*b^4*c^3*f^2 - a^3*b^9*d^3*f^2 - 4*a^5*b^7*d^3*f^2 - 6*a^7*b^5*d^3*f^2 - 4
*a^9*b^3*d^3*f^2 - 3*a*b^11*c^2*d*f^2 + 3*a^2*b^10*c*d^2*f^2 - 12*a^3*b^9*c
^2*d*f^2 + 12*a^4*b^8*c*d^2*f^2 - 18*a^5*b^7*c^2*d*f^2 + 18*a^6*b^6*c*d^2*f
^2 - 12*a^7*b^5*c^2*d*f^2 + 12*a^8*b^4*c*d^2*f^2 - 3*a^9*b^3*c^2*d*f^2 + 3*
a^10*b^2*c*d^2*f^2))^(1/2))/(b^10*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^4*(
2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^8*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2)
 + b^6*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^3*(8*a^9*d^3*f^2 + 6*a^9*c^2*
d*f^2) - b^9*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^5*(12*a^7*d^3*f^2 + 24*
a^7*c^2*d*f^2) - b^7*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) + 2*b^12*c^3*f^2 -
2*a^11*b*d^3*f^2 - 6*a*b^11*c^2*d*f^2 + 6*a^10*b^2*c*d^2*f^2) - (32*(3*C^5*
a^3*b^6*d^10 - C^5*a^5*b^4*d^10 + 4*C^5*a*b^8*c^2*d^8 - 7*C^5*a^2*b^7*c*d^9
 + C^5*a^4*b^5*c*d^9))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6
*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 +
4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5
 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d
f^5) + (((((16(8*C^3*a^6*b^7*d^11*f^2 - 78*C^3*a^4*b^9*d^11*f^2 + 60*C^3*
a^8*b^5*d^11*f^2 - 24*C^3*a^10*b^3*d^11*f^2 + 2*C^3*a^12*b*d^11*f^2 - 32*C^
3*a*b^12*c^3*d^8*f^2 + 152*C^3*a^3*b^10*c*d^10*f^2 + 128*C^3*a^5*b^8*c*d^10
*f^2 - 64*C^3*a^7*b^6*c*d^10*f^2 - 32*C^3*a^9*b^4*c*d^10*f^2 + 8*C^3*a^11*b
^2*c*d^10*f^2 - 40*C^3*a^2*b^11*c^2*d^9*f^2 + 64*C^3*a^3*b^10*c^3*d^8*f^2 -
 216*C^3*a^4*b^9*c^2*d^9*f^2 + 96*C^3*a^5*b^8*c^3*d^8*f^2 - 120*C^3*a^6*b^7
*c^2*d^9*f^2 + 56*C^3*a^8*b^5*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 +
4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5
 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*
f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*
d*f^5 - 8*a^7*b^3*c*d*f^5) - (((((16*(40*C*a^3*b^14*d^12*f^4 + 192*C*a^5*b^
12*d^12*f^4 + 360*C*a^7*b^10*d^12*f^4 + 320*C*a^9*b^8*d^12*f^4 + 120*C*a^11
*b^6*d^12*f^4 - 8*C*a^15*b^2*d^12*f^4 + 8*C*b^17*c^3*d^9*f^4 + 40*C*a*b^16*
c^2*d^10*f^4 + 32*C*a*b^16*c^4*d^8*f^4 - 88*C*a^2*b^15*c*d^11*f^4 - 448*C*a
^4*b^13*c*d^11*f^4 - 920*C*a^6*b^11*c*d^11*f^4 - 960*C*a^8*b^9*c*d^11*f^4 -
 520*C*a^10*b^7*c*d^11*f^4 - 128*C*a^12*b^5*c*d^11*f^4 - 8*C*a^14*b^3*c*d^1
1*f^4 - 32*C*a^2*b^15*c^3*d^9*f^4 + 256*C*a^3*b^14*c^2*d^10*f^4 + 160*C*a^3
*b^14*c^4*d^8*f^4 - 280*C*a^4*b^13*c^3*d^9*f^4 + 680*C*a^5*b^12*c^2*d^10*f^
4 + 320*C*a^5*b^12*c^4*d^8*f^4 - 640*C*a^6*b^11*c^3*d^9*f^4 + 960*C*a^7*b^1

0*c^2*d^10*f^4 + 320*C*a^7*b^10*c^4*d^8*f^4 - 680*C*a^8*b^9*c^3*d^9*f^4 + 7
60*C*a^9*b^8*c^2*d^10*f^4 + 160*C*a^9*b^8*c^4*d^8*f^4 - 352*C*a^10*b^7*c^3*
d^9*f^4 + 320*C*a^11*b^6*c^2*d^10*f^4 + 32*C*a^11*b^6*c^4*d^8*f^4 - 72*C*a^
12*b^5*c^3*d^9*f^4 + 56*C*a^13*b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*
f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c
^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^
2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*
b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + (16*((C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 +
9*C^2*a^4*b^4*d^2 - 6*C^2*a^6*b^2*d^2 - 24*C^2*a^3*b^5*c*d + 8*C^2*a^5*b^3
*c*d)*(b^12*c^3*f^2 - a^11*b*d^3*f^2 + 4*a^2*b^10*c^3*f^2 + 6*a^4*b^8*c^3*f
^2 + 4*a^6*b^6*c^3*f^2 + a^8*b^4*c^3*f^2 - a^3*b^9*d^3*f^2 - 4*a^5*b^7*d^3*
f^2 - 6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*d^3*f^2 - 3*a*b^11*c^2*d*f^2 + 3*a^2*b^
10*c*d^2*f^2 - 12*a^3*b^9*c^2*d*f^2 + 12*a^4*b^8*c*d^2*f^2 - 18*a^5*b^7*c^2
*d*f^2 + 18*a^6*b^6*c*d^2*f^2 - 12*a^7*b^5*c^2*d*f^2 + 12*a^8*b^4*c*d^2*f^2
- 3*a^9*b^3*c^2*d*f^2 + 3*a^10*b^2*c*d^2*f^2))^(1/2)*(c + d*tan(e + f*x))^
(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4
+ 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 -
160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^
19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432
*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^
4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4
*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b
^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 4
8*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f
^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*
d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*
c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*
c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^1
0*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^1
5*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((b^10*(8*a^2*c^3*f^2 + 6*a^2*c
*d^2*f^2) + b^4*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^8*(12*a^4*c^3*f^2 +
24*a^4*c*d^2*f^2) + b^6*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^3*(8*a^9*d^3
*f^2 + 6*a^9*c^2*d*f^2) - b^9*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^5*(12*
a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^7*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) +
2*b^12*c^3*f^2 - 2*a^11*b*d^3*f^2 - 6*a*b^11*c^2*d*f^2 + 6*a^10*b^2*c*d^2*f
^2)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 +
4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 +
6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4
- 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4)))*((C^2*a^8*d
^2 + 16*C^2*a^2*b^6*c^2 + 9*C^2*a^4*b^4*d^2 - 6*C^2*a^6*b^2*d^2 - 24*C^2*a^
3*b^5*c*d + 8*C^2*a^5*b^3*c*d)*(b^12*c^3*f^2 - a^11*b*d^3*f^2 + 4*a^2*b^10*
c^3*f^2 + 6*a^4*b^8*c^3*f^2 + 4*a^6*b^6*c^3*f^2 + a^8*b^4*c^3*f^2 - a^3*b^9
*d^3*f^2 - 4*a^5*b^7*d^3*f^2 - 6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*d^3*f^2 - 3*a*
b^11*c^2*d*f^2 + 3*a^2*b^10*c*d^2*f^2 - 12*a^3*b^9*c^2*d*f^2 + 12*a^4*b^8*c
*d^2*f^2 - 18*a^5*b^7*c^2*d*f^2 + 18*a^6*b^6*c*d^2*f^2 - 12*a^7*b^5*c^2*d*f

^2 + 12*a^8*b^4*c*d^2*f^2 - 3*a^9*b^3*c^2*d*f^2 + 3*a^10*b^2*c*d^2*f^2))^(1
/2))/(b^10*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^4*(2*a^8*c^3*f^2 + 24*a^8*
c*d^2*f^2) + b^8*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^6*(8*a^6*c^3*f^2 +
36*a^6*c*d^2*f^2) - b^3*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^9*(2*a^3*d^3
*f^2 + 24*a^3*c^2*d*f^2) - b^5*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^7*(8
*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) + 2*b^12*c^3*f^2 - 2*a^11*b*d^3*f^2 - 6*a*
b^11*c^2*d*f^2 + 6*a^10*b^2*c*d^2*f^2) - (16*(c + d*tan(e + f*x))^(1/2)*(20
*C^2*a^5*b^10*d^11*f^2 - 60*C^2*a^3*b^12*d^11*f^2 + 168*C^2*a^7*b^8*d^11*f^
2 + 40*C^2*a^9*b^6*d^11*f^2 - 44*C^2*a^11*b^4*d^11*f^2 + 4*C^2*a^13*b^2*d^1
1*f^2 - 20*C^2*b^15*c^3*d^8*f^2 - 4*C^2*a^14*b*c*d^10*f^2 - 20*C^2*a*b^14*c
^2*d^9*f^2 + 100*C^2*a^2*b^13*c*d^10*f^2 - 300*C^2*a^6*b^9*c*d^10*f^2 - 160
*C^2*a^8*b^7*c*d^10*f^2 + 76*C^2*a^10*b^5*c*d^10*f^2 + 32*C^2*a^12*b^3*c*d^
10*f^2 + 116*C^2*a^2*b^13*c^3*d^8*f^2 - 124*C^2*a^3*b^12*c^2*d^9*f^2 + 216*
C^2*a^4*b^11*c^3*d^8*f^2 - 40*C^2*a^5*b^10*c^2*d^9*f^2 + 8*C^2*a^6*b^9*c^3*
d^8*f^2 + 168*C^2*a^7*b^8*c^2*d^9*f^2 - 68*C^2*a^8*b^7*c^3*d^8*f^2 + 60*C^2
*a^9*b^6*c^2*d^9*f^2 + 4*C^2*a^10*b^5*c^3*d^8*f^2 - 44*C^2*a^11*b^4*c^2*d^9
*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4
+ 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^
4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f
^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))*((C^2*a^8
*d^2 + 16*C^2*a^2*b^6*c^2 + 9*C^2*a^4*b^4*d^2 - 6*C^2*a^6*b^2*d^2 - 24*C^2*
a^3*b^5*c*d + 8*C^2*a^5*b^3*c*d)*(b^12*c^3*f^2 - a^11*b*d^3*f^2 + 4*a^2*b^1
0*c^3*f^2 + 6*a^4*b^8*c^3*f^2 + 4*a^6*b^6*c^3*f^2 + a^8*b^4*c^3*f^2 - a^3*b
^9*d^3*f^2 - 4*a^5*b^7*d^3*f^2 - 6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*d^3*f^2 - 3*
a*b^11*c^2*d*f^2 + 3*a^2*b^10*c*d^2*f^2 - 12*a^3*b^9*c^2*d*f^2 + 12*a^4*b^8
*c*d^2*f^2 - 18*a^5*b^7*c^2*d*f^2 + 18*a^6*b^6*c*d^2*f^2 - 12*a^7*b^5*c^2*d
*f^2 + 12*a^8*b^4*c*d^2*f^2 - 3*a^9*b^3*c^2*d*f^2 + 3*a^10*b^2*c*d^2*f^2))^
(1/2))/(b^10*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^4*(2*a^8*c^3*f^2 + 24*a^
8*c*d^2*f^2) + b^8*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^6*(8*a^6*c^3*f^2
+ 36*a^6*c*d^2*f^2) - b^3*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^9*(2*a^3*d
^3*f^2 + 24*a^3*c^2*d*f^2) - b^5*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^7*
(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) + 2*b^12*c^3*f^2 - 2*a^11*b*d^3*f^2 - 6*
a*b^11*c^2*d*f^2 + 6*a^10*b^2*c*d^2*f^2))*((C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^
2 + 9*C^2*a^4*b^4*d^2 - 6*C^2*a^6*b^2*d^2 - 24*C^2*a^3*b^5*c*d + 8*C^2*a^5*
b^3*c*d)*(b^12*c^3*f^2 - a^11*b*d^3*f^2 + 4*a^2*b^10*c^3*f^2 + 6*a^4*b^8*c^
3*f^2 + 4*a^6*b^6*c^3*f^2 + a^8*b^4*c^3*f^2 - a^3*b^9*d^3*f^2 - 4*a^5*b^7*d
^3*f^2 - 6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*d^3*f^2 - 3*a*b^11*c^2*d*f^2 + 3*a^2
*b^10*c*d^2*f^2 - 12*a^3*b^9*c^2*d*f^2 + 12*a^4*b^8*c*d^2*f^2 - 18*a^5*b^7*
c^2*d*f^2 + 18*a^6*b^6*c*d^2*f^2 - 12*a^7*b^5*c^2*d*f^2 + 12*a^8*b^4*c*d^2*
f^2 - 3*a^9*b^3*c^2*d*f^2 + 3*a^10*b^2*c*d^2*f^2))^(1/2))/(b^10*(8*a^2*c^3*
f^2 + 6*a^2*c*d^2*f^2) + b^4*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^8*(12*a
^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^6*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b
^3*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^9*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^
2) - b^5*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^7*(8*a^5*d^3*f^2 + 36*a^5*
c^2*d*f^2) + 2*b^12*c^3*f^2 - 2*a^11*b*d^3*f^2 - 6*a*b^11*c^2*d*f^2 + 6*a^1

0*b^2*c*d^2*f^2) - (16*(c + d*tan(e + f*x))^(1/2)*(2*C^4*a^2*b^9*d^10 - 5*C
^4*a^4*b^7*d^10 + 17*C^4*a^6*b^5*d^10 - 7*C^4*a^8*b^3*d^10 + 2*C^4*b^11*c^2
*d^8 + C^4*a^10*b*d^10 - 12*C^4*a^2*b^9*c^2*d^8 + 18*C^4*a^4*b^7*c^2*d^8 -
4*C^4*a*b^10*c*d^9 + 16*C^4*a^3*b^8*c*d^9 - 36*C^4*a^5*b^6*c*d^9 + 8*C^4*a^
7*b^4*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*
c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6
*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*
b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))*((
C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 9*C^2*a^4*b^4*d^2 - 6*C^2*a^6*b^2*d^2 -
24*C^2*a^3*b^5*c*d + 8*C^2*a^5*b^3*c*d)*(b^12*c^3*f^2 - a^11*b*d^3*f^2 + 4*
a^2*b^10*c^3*f^2 + 6*a^4*b^8*c^3*f^2 + 4*a^6*b^6*c^3*f^2 + a^8*b^4*c^3*f^2
- a^3*b^9*d^3*f^2 - 4*a^5*b^7*d^3*f^2 - 6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*d^3*f
^2 - 3*a*b^11*c^2*d*f^2 + 3*a^2*b^10*c*d^2*f^2 - 12*a^3*b^9*c^2*d*f^2 + 12*
a^4*b^8*c*d^2*f^2 - 18*a^5*b^7*c^2*d*f^2 + 18*a^6*b^6*c*d^2*f^2 - 12*a^7*b^
5*c^2*d*f^2 + 12*a^8*b^4*c*d^2*f^2 - 3*a^9*b^3*c^2*d*f^2 + 3*a^10*b^2*c*d^2
f^2))^(1/2))/(b^10(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^4*(2*a^8*c^3*f^2
+ 24*a^8*c*d^2*f^2) + b^8*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^6*(8*a^6*
c^3*f^2 + 36*a^6*c*d^2*f^2) - b^3*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^9*(
2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^5*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2)
- b^7*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) + 2*b^12*c^3*f^2 - 2*a^11*b*d^3*f
^2 - 6*a*b^11*c^2*d*f^2 + 6*a^10*b^2*c*d^2*f^2)))*((C^2*a^8*d^2 + 16*C^2*a^
2*b^6*c^2 + 9*C^2*a^4*b^4*d^2 - 6*C^2*a^6*b^2*d^2 - 24*C^2*a^3*b^5*c*d + 8*
C^2*a^5*b^3*c*d)*(b^12*c^3*f^2 - a^11*b*d^3*f^2 + 4*a^2*b^10*c^3*f^2 + 6*a^
4*b^8*c^3*f^2 + 4*a^6*b^6*c^3*f^2 + a^8*b^4*c^3*f^2 - a^3*b^9*d^3*f^2 - 4*a
^5*b^7*d^3*f^2 - 6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*d^3*f^2 - 3*a*b^11*c^2*d*f^2
+ 3*a^2*b^10*c*d^2*f^2 - 12*a^3*b^9*c^2*d*f^2 + 12*a^4*b^8*c*d^2*f^2 - 18*
a^5*b^7*c^2*d*f^2 + 18*a^6*b^6*c*d^2*f^2 - 12*a^7*b^5*c^2*d*f^2 + 12*a^8*b^
4*c*d^2*f^2 - 3*a^9*b^3*c^2*d*f^2 + 3*a^10*b^2*c*d^2*f^2))^(1/2)*2i)/(b^10*
(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^4*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2)
+ b^8*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^6*(8*a^6*c^3*f^2 + 36*a^6*c*d
^2*f^2) - b^3*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^9*(2*a^3*d^3*f^2 + 24*a
^3*c^2*d*f^2) - b^5*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^7*(8*a^5*d^3*f^
2 + 36*a^5*c^2*d*f^2) + 2*b^12*c^3*f^2 - 2*a^11*b*d^3*f^2 - 6*a*b^11*c^2*d*
f^2 + 6*a^10*b^2*c*d^2*f^2) - (atan((((((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b
^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2
*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*
b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a
*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c
*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)*(a^8
*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*
a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4
+ 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(3*B^4*a^2*b^9
*d^10 - 3*B^4*a^4*b^7*d^10 + 17*B^4*a^6*b^5*d^10 - 9*B^4*a^8*b^3*d^10 + 6*B
^4*b^11*c^2*d^8 - 8*B^4*a^2*b^9*c^2*d^8 + 14*B^4*a^4*b^7*c^2*d^8 - 4*B^4*a^
6*b^5*c^2*d^8 - 8*B^4*a*b^10*c*d^9 + 12*B^4*a^3*b^8*c*d^9 - 32*B^4*a^5*b^6*

c*d^9 + 12*B^4*a^7*b^4*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) + ((((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((8*(52*B^3*a^3*b^10*d^11*f^2 - 128*B^3*a^5*b^8*d^11*f^2 - 24*B^3*a^7*b^6*d^11*f^2 + 160*B^3*a^9*b^4*d^11*f^2 + 4*B^3*a^11*b^2*d^11*f^2 + 12*B^3*b^13*c^3*d^8*f^2 + 44*B^3*a*b^12*c^2*d^9*f^2 - 128*B^3*a^2*b^11*c*d^10*f^2 + 48*B^3*a^4*b^9*c*d^10*f^2 + 176*B^3*a^6*b^7*c*d^10*f^2 - 48*B^3*a^8*b^5*c*d^10*f^2 - 48*B^3*a^10*b^3*c*d^10*f^2 - 112*B^3*a^2*b^11*c^3*d^8*f^2 + 192*B^3*a^3*b^10*c^2*d^9*f^2 - 24*B^3*a^4*b^9*c^3*d^8*f^2 - 72*B^3*a^5*b^8*c^2*d^9*f^2 + 80*B^3*a^6*b^7*c^3*d^8*f^2 - 160*B^3*a^7*b^6*c^2*d^9*f^2 - 20*B^3*a^8*b^5*c^3*d^8*f^2 + 60*B^3*a^9*b^4*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - ((((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(68*B^2*a^3*b^12*d^11*f^2 + 20*B^2*a^5*b^10*d^11*f^2 - 88*B^2*a^7*b^8*d^11*f^2 + 40*B^2*a^9*b^6*d^11*f^2 + 84*B^2*a^11*b^4*d^11*f^2 + 4*B^2*a^13*b^2*d^11*f^2 + 36*B^2*b^15*c^3*d^8*f^2 + 36*B^2*a*b^14*c^2*d^9*f^2 - 128*B^2*a^2*b^13*c*d^10*f^2 - 112*B^2*a^4*b^11*c*d^10*f^2 + 128*B^2*a^6*b^9*c*d^10*f^2 + 32*B^2*a^8*b^7*c*d^10*f^2 - 128*B^2*a^10*b^5*c*d^10*f^2 - 48*B^2*a^12*b^3*c*d^10*f^2 - 68*B^2*a^2*b^13*c^3*d^8*f^2 + 204*B^2*a^3*b^12*c^2*d^9*f^2 - 184*B^2*a^4*b^11*c^3*d^8*f^2 + 200*B^2*a^5*b^10*c^2*d^9*f^2 - 40*B^2*a^6*b^9*c^3*d^8*f^2 - 8*B^2*a^7*b^8*c^2*d^9*f^2 + 20*B^2*a^8*b^7*c^3*d^8*f^2 + 20*B^2*a^9*b^6*c^2*d^9*f^2 - 20*B^2*a^10*b^5*c^3*d^8*f^2 + 60*B^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) + ((((512*B^4*

a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8
*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*
b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a
^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) + 4*B
^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 -
24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^
4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d
^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((8*(32*B*a^2*b^15*d
^12*f^4 + 96*B*a^4*b^13*d^12*f^4 - 320*B*a^8*b^9*d^12*f^4 - 480*B*a^10*b^7*
d^12*f^4 - 288*B*a^12*b^5*d^12*f^4 - 64*B*a^14*b^3*d^12*f^4 + 64*B*b^17*c^2
*d^10*f^4 + 48*B*b^17*c^4*d^8*f^4 - 112*B*a*b^16*c^3*d^9*f^4 - 400*B*a^3*b^
14*c*d^11*f^4 - 544*B*a^5*b^12*c*d^11*f^4 - 80*B*a^7*b^10*c*d^11*f^4 + 480*
B*a^9*b^8*c*d^11*f^4 + 464*B*a^11*b^6*c*d^11*f^4 + 160*B*a^13*b^4*c*d^11*f^
4 + 16*B*a^15*b^2*c*d^11*f^4 + 368*B*a^2*b^15*c^2*d^10*f^4 + 224*B*a^2*b^15
*c^4*d^8*f^4 - 512*B*a^3*b^14*c^3*d^9*f^4 + 832*B*a^4*b^13*c^2*d^10*f^4 + 4
00*B*a^4*b^13*c^4*d^8*f^4 - 880*B*a^5*b^12*c^3*d^9*f^4 + 880*B*a^6*b^11*c^2
*d^10*f^4 + 320*B*a^6*b^11*c^4*d^8*f^4 - 640*B*a^7*b^10*c^3*d^9*f^4 + 320*B
*a^8*b^9*c^2*d^10*f^4 + 80*B*a^8*b^9*c^4*d^8*f^4 - 80*B*a^9*b^8*c^3*d^9*f^4
 - 176*B*a^10*b^7*c^2*d^10*f^4 - 32*B*a^10*b^7*c^4*d^8*f^4 + 128*B*a^11*b^6
*c^3*d^9*f^4 - 192*B*a^12*b^5*c^2*d^10*f^4 - 16*B*a^12*b^5*c^4*d^8*f^4 + 48
*B*a^13*b^4*c^3*d^9*f^4 - 48*B*a^14*b^3*c^2*d^10*f^4 - 96*B*a*b^16*c*d^11*f
^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 +
 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5
+ 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5
 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - (4*(((512*
B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4
*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*
a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B
^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) +
 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^
2 - 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^
2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b
^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*tan(e + f
x))^(1/2)(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^1
2*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f
^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 +
48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4
- 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d
^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^1
3*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*
a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^
4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d
^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5
*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16
*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*

b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)))*1i)/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)) + ((((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(3*B^4*a^2*b^9*d^10 - 3*B^4*a^4*b^7*d^10 + 17*B^4*a^6*b^5*d^10 - 9*B^4*a^8*b^3*d^10 + 6*B^4*b^11*c^2*d^8 - 8*B^4*a^2*b^9*c^2*d^8 + 14*B^4*a^4*b^7*c^2*d^8 - 4*B^4*a^6*b^5*c^2*d^8 - 8*B^4*a*b^10*c*d^9 + 12*B^4*a^3*b^8*c*d^9 - 32*B^4*a^5*b^6*c*d^9 + 12*B^4*a^7*b^4*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((8*(52*B^3*a^3*b^10*d^11*f^2 - 128*B^3*a^5*b^8*d^11*f^2 - 24*B^3*a^7*b^6*d^11*f^2 + 160*B^3*a^9*b^4*d^11*f^2 + 4*B^3*a^11*b^2*d^11*f^2 + 12*B^3*b^13*c^3*d^8*f^2 + 44*B^3*a*b^12*c^2*d^9*f^2 - 128*B^3*a^2*b^11*c*

d^10*f^2 + 48*B^3*a^4*b^9*c*d^10*f^2 + 176*B^3*a^6*b^7*c*d^10*f^2 - 48*B^3*
a^8*b^5*c*d^10*f^2 - 48*B^3*a^10*b^3*c*d^10*f^2 - 112*B^3*a^2*b^11*c^3*d^8*
f^2 + 192*B^3*a^3*b^10*c^2*d^9*f^2 - 24*B^3*a^4*b^9*c^3*d^8*f^2 - 72*B^3*a^
5*b^8*c^2*d^9*f^2 + 80*B^3*a^6*b^7*c^3*d^8*f^2 - 160*B^3*a^7*b^6*c^2*d^9*f^
2 - 20*B^3*a^8*b^5*c^3*d^8*f^2 + 60*B^3*a^9*b^4*c^2*d^9*f^2))/(a^10*d^2*f^5
 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5
 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^
5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f
^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + ((((512*B^4*a^4*b^4*c^2*f^4
- 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B
^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 19
2*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 +
 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) + 4*B^2*a^4*c*f^2 + 4
*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c
f^2)(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^
2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b
^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(68*
B^2*a^3*b^12*d^11*f^2 + 20*B^2*a^5*b^10*d^11*f^2 - 88*B^2*a^7*b^8*d^11*f^2
+ 40*B^2*a^9*b^6*d^11*f^2 + 84*B^2*a^11*b^4*d^11*f^2 + 4*B^2*a^13*b^2*d^11*
f^2 + 36*B^2*b^15*c^3*d^8*f^2 + 36*B^2*a*b^14*c^2*d^9*f^2 - 128*B^2*a^2*b^1
3*c*d^10*f^2 - 112*B^2*a^4*b^11*c*d^10*f^2 + 128*B^2*a^6*b^9*c*d^10*f^2 + 3
2*B^2*a^8*b^7*c*d^10*f^2 - 128*B^2*a^10*b^5*c*d^10*f^2 - 48*B^2*a^12*b^3*c*
d^10*f^2 - 68*B^2*a^2*b^13*c^3*d^8*f^2 + 204*B^2*a^3*b^12*c^2*d^9*f^2 - 184
*B^2*a^4*b^11*c^3*d^8*f^2 + 200*B^2*a^5*b^10*c^2*d^9*f^2 - 40*B^2*a^6*b^9*c
^3*d^8*f^2 - 8*B^2*a^7*b^8*c^2*d^9*f^2 + 20*B^2*a^8*b^7*c^3*d^8*f^2 + 20*B^
2*a^9*b^6*c^2*d^9*f^2 - 20*B^2*a^10*b^5*c^3*d^8*f^2 + 60*B^2*a^11*b^4*c^2*d
^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f
^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*
f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d
*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((((51
2*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B
^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^
4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896
*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2)
 + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*
f^2 - 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*
d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2
*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((8*(32*B*a^2*
b^15*d^12*f^4 + 96*B*a^4*b^13*d^12*f^4 - 320*B*a^8*b^9*d^12*f^4 - 480*B*a^1
0*b^7*d^12*f^4 - 288*B*a^12*b^5*d^12*f^4 - 64*B*a^14*b^3*d^12*f^4 + 64*B*b^
17*c^2*d^10*f^4 + 48*B*b^17*c^4*d^8*f^4 - 112*B*a*b^16*c^3*d^9*f^4 - 400*B*
a^3*b^14*c*d^11*f^4 - 544*B*a^5*b^12*c*d^11*f^4 - 80*B*a^7*b^10*c*d^11*f^4
+ 480*B*a^9*b^8*c*d^11*f^4 + 464*B*a^11*b^6*c*d^11*f^4 + 160*B*a^13*b^4*c*d
^11*f^4 + 16*B*a^15*b^2*c*d^11*f^4 + 368*B*a^2*b^15*c^2*d^10*f^4 + 224*B*a^
2*b^15*c^4*d^8*f^4 - 512*B*a^3*b^14*c^3*d^9*f^4 + 832*B*a^4*b^13*c^2*d^10*f

^4 + 400*B*a^4*b^13*c^4*d^8*f^4 - 880*B*a^5*b^12*c^3*d^9*f^4 + 880*B*a^6*b^
11*c^2*d^10*f^4 + 320*B*a^6*b^11*c^4*d^8*f^4 - 640*B*a^7*b^10*c^3*d^9*f^4 +
 320*B*a^8*b^9*c^2*d^10*f^4 + 80*B*a^8*b^9*c^4*d^8*f^4 - 80*B*a^9*b^8*c^3*d
^9*f^4 - 176*B*a^10*b^7*c^2*d^10*f^4 - 32*B*a^10*b^7*c^4*d^8*f^4 + 128*B*a^
11*b^6*c^3*d^9*f^4 - 192*B*a^12*b^5*c^2*d^10*f^4 - 16*B*a^12*b^5*c^4*d^8*f^
4 + 48*B*a^13*b^4*c^3*d^9*f^4 - 48*B*a^14*b^3*c^2*d^10*f^4 - 96*B*a*b^16*c*
d^11*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2
*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^
2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c
*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + (4*(
((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 -
16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 60
8*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 +
 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(
1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*
b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 +
b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4
*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*tan
(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^
13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*
d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*
f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^
8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15
*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a
^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4
+ 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d
^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7
*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^
14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 4
8*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 30
4*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 +
880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^
4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8*
d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4
 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^
2*f^4)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4
 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^
4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f
^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^8
*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*
a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4
 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*
d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2
*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 +
a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2

*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^
2*d^2*f^4)))*1i)/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4
+ 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2
*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)))/((16*(9*B^5*a^6*b^3*d^10 -
B^5*a^2*b^7*d^10 - 4*B^5*a^2*b^7*c^2*d^8 + 4*B^5*a^4*b^5*c^2*d^8 + 2*B^5*a*
b^8*c*d^9 + 6*B^5*a^3*b^6*c*d^9 - 12*B^5*a^5*b^4*c*d^9))/(a^10*d^2*f^5 + b^
10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^
8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4
*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 -
12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + ((((512*B^4*a^4*b^4*c^2*f^4 - 16*
B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^
6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4
*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*
B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*
b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)
*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4
+ 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^
2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(3*B^4*a^
2*b^9*d^10 - 3*B^4*a^4*b^7*d^10 + 17*B^4*a^6*b^5*d^10 - 9*B^4*a^8*b^3*d^10
+ 6*B^4*b^11*c^2*d^8 - 8*B^4*a^2*b^9*c^2*d^8 + 14*B^4*a^4*b^7*c^2*d^8 - 4*B
^4*a^6*b^5*c^2*d^8 - 8*B^4*a*b^10*c*d^9 + 12*B^4*a^3*b^8*c*d^9 - 32*B^4*a^5
*b^6*c*d^9 + 12*B^4*a^7*b^4*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^
8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b
^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*
a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 -
8*a^7*b^3*c*d*f^4) + ((((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256
*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B
^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 89
6*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 1
28*B^4*a^7*b*c*d*f^4)^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 + 16*B^2*a*
b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d
^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4
+ 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2
f^4))^(1/2)((8*(52*B^3*a^3*b^10*d^11*f^2 - 128*B^3*a^5*b^8*d^11*f^2 - 24*
B^3*a^7*b^6*d^11*f^2 + 160*B^3*a^9*b^4*d^11*f^2 + 4*B^3*a^11*b^2*d^11*f^2 +
12*B^3*b^13*c^3*d^8*f^2 + 44*B^3*a*b^12*c^2*d^9*f^2 - 128*B^3*a^2*b^11*c*d
^10*f^2 + 48*B^3*a^4*b^9*c*d^10*f^2 + 176*B^3*a^6*b^7*c*d^10*f^2 - 48*B^3*a
^8*b^5*c*d^10*f^2 - 48*B^3*a^10*b^3*c*d^10*f^2 - 112*B^3*a^2*b^11*c^3*d^8*f
^2 + 192*B^3*a^3*b^10*c^2*d^9*f^2 - 24*B^3*a^4*b^9*c^3*d^8*f^2 - 72*B^3*a^5
*b^8*c^2*d^9*f^2 + 80*B^3*a^6*b^7*c^3*d^8*f^2 - 160*B^3*a^7*b^6*c^2*d^9*f^2
- 20*B^3*a^8*b^5*c^3*d^8*f^2 + 60*B^3*a^9*b^4*c^2*d^9*f^2))/(a^10*d^2*f^5
+ b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5
+ a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5
+ 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^
5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - ((((512*B^4*a^4*b^4*c^2*f^4 -

16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^
4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192
*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 +
128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) + 4*B^2*a^4*c*f^2 + 4*
B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*
f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2
*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^
4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(68*B
^2*a^3*b^12*d^11*f^2 + 20*B^2*a^5*b^10*d^11*f^2 - 88*B^2*a^7*b^8*d^11*f^2 +
40*B^2*a^9*b^6*d^11*f^2 + 84*B^2*a^11*b^4*d^11*f^2 + 4*B^2*a^13*b^2*d^11*f
^2 + 36*B^2*b^15*c^3*d^8*f^2 + 36*B^2*a*b^14*c^2*d^9*f^2 - 128*B^2*a^2*b^13
*c*d^10*f^2 - 112*B^2*a^4*b^11*c*d^10*f^2 + 128*B^2*a^6*b^9*c*d^10*f^2 + 32
*B^2*a^8*b^7*c*d^10*f^2 - 128*B^2*a^10*b^5*c*d^10*f^2 - 48*B^2*a^12*b^3*c*d
^10*f^2 - 68*B^2*a^2*b^13*c^3*d^8*f^2 + 204*B^2*a^3*b^12*c^2*d^9*f^2 - 184*
B^2*a^4*b^11*c^3*d^8*f^2 + 200*B^2*a^5*b^10*c^2*d^9*f^2 - 40*B^2*a^6*b^9*c^
3*d^8*f^2 - 8*B^2*a^7*b^8*c^2*d^9*f^2 + 20*B^2*a^8*b^7*c^3*d^8*f^2 + 20*B^2
*a^9*b^6*c^2*d^9*f^2 - 20*B^2*a^10*b^5*c^3*d^8*f^2 + 60*B^2*a^11*b^4*c^2*d^
9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^
4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f
^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*
f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) + ((((512
*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^
4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4
*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*
B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2)
+ 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f
^2 - 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d
^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*
b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((8*(32*B*a^2*b
^15*d^12*f^4 + 96*B*a^4*b^13*d^12*f^4 - 320*B*a^8*b^9*d^12*f^4 - 480*B*a^10
*b^7*d^12*f^4 - 288*B*a^12*b^5*d^12*f^4 - 64*B*a^14*b^3*d^12*f^4 + 64*B*b^1
7*c^2*d^10*f^4 + 48*B*b^17*c^4*d^8*f^4 - 112*B*a*b^16*c^3*d^9*f^4 - 400*B*a
^3*b^14*c*d^11*f^4 - 544*B*a^5*b^12*c*d^11*f^4 - 80*B*a^7*b^10*c*d^11*f^4 +
480*B*a^9*b^8*c*d^11*f^4 + 464*B*a^11*b^6*c*d^11*f^4 + 160*B*a^13*b^4*c*d^
11*f^4 + 16*B*a^15*b^2*c*d^11*f^4 + 368*B*a^2*b^15*c^2*d^10*f^4 + 224*B*a^2
*b^15*c^4*d^8*f^4 - 512*B*a^3*b^14*c^3*d^9*f^4 + 832*B*a^4*b^13*c^2*d^10*f^
4 + 400*B*a^4*b^13*c^4*d^8*f^4 - 880*B*a^5*b^12*c^3*d^9*f^4 + 880*B*a^6*b^1
1*c^2*d^10*f^4 + 320*B*a^6*b^11*c^4*d^8*f^4 - 640*B*a^7*b^10*c^3*d^9*f^4 +
320*B*a^8*b^9*c^2*d^10*f^4 + 80*B*a^8*b^9*c^4*d^8*f^4 - 80*B*a^9*b^8*c^3*d^
9*f^4 - 176*B*a^10*b^7*c^2*d^10*f^4 - 32*B*a^10*b^7*c^4*d^8*f^4 + 128*B*a^1
1*b^6*c^3*d^9*f^4 - 192*B*a^12*b^5*c^2*d^10*f^4 - 16*B*a^12*b^5*c^4*d^8*f^4
+ 48*B*a^13*b^4*c^3*d^9*f^4 - 48*B*a^14*b^3*c^2*d^10*f^4 - 96*B*a*b^16*c*d
^11*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*
f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2
*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*

d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - (4*((
(512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 1
6*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608
*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 +
896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1
/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b
*d*f^2 - 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b
^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*
a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*tan(
e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^1
3*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d
^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f
^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8
*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*
c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^
6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 +
400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^
10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*
c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^1
4*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48
*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304
*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 8
80*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4
+ 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8*d
^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4
+ 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2
f^4)(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4
+ 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4
+ 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^
4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^8*
c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a
^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4
+ 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d
^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*
b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a
^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*
f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2
*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*
a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4
+ 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)) - ((((512*B^4*a^4*b^4*c^2*f^4 -
16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4
*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*
B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 1
28*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) + 4*B^2*a^4*c*f^2 + 4*B
^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f

^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*
f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4
*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(3*B^4
*a^2*b^9*d^10 - 3*B^4*a^4*b^7*d^10 + 17*B^4*a^6*b^5*d^10 - 9*B^4*a^8*b^3*d^
10 + 6*B^4*b^11*c^2*d^8 - 8*B^4*a^2*b^9*c^2*d^8 + 14*B^4*a^4*b^7*c^2*d^8 -
4*B^4*a^6*b^5*c^2*d^8 - 8*B^4*a*b^10*c*d^9 + 12*B^4*a^3*b^8*c*d^9 - 32*B^4*
a^5*b^6*c*d^9 + 12*B^4*a^7*b^4*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2
*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^
2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 -
 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4
 - 8*a^7*b^3*c*d*f^4) - ((((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 -
256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 19
2*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 -
 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4
- 128*B^4*a^7*b*c*d*f^4)^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 + 16*B^2
*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^
8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f
^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*
d^2*f^4))^(1/2)*((8*(52*B^3*a^3*b^10*d^11*f^2 - 128*B^3*a^5*b^8*d^11*f^2 -
24*B^3*a^7*b^6*d^11*f^2 + 160*B^3*a^9*b^4*d^11*f^2 + 4*B^3*a^11*b^2*d^11*f^
2 + 12*B^3*b^13*c^3*d^8*f^2 + 44*B^3*a*b^12*c^2*d^9*f^2 - 128*B^3*a^2*b^11*
c*d^10*f^2 + 48*B^3*a^4*b^9*c*d^10*f^2 + 176*B^3*a^6*b^7*c*d^10*f^2 - 48*B^
3*a^8*b^5*c*d^10*f^2 - 48*B^3*a^10*b^3*c*d^10*f^2 - 112*B^3*a^2*b^11*c^3*d^
8*f^2 + 192*B^3*a^3*b^10*c^2*d^9*f^2 - 24*B^3*a^4*b^9*c^3*d^8*f^2 - 72*B^3*
a^5*b^8*c^2*d^9*f^2 + 80*B^3*a^6*b^7*c^3*d^8*f^2 - 160*B^3*a^7*b^6*c^2*d^9*
f^2 - 20*B^3*a^8*b^5*c^3*d^8*f^2 + 60*B^3*a^9*b^4*c^2*d^9*f^2))/(a^10*d^2*f
^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f
^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*
f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d
*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + ((((512*B^4*a^4*b^4*c^2*f^
4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256
*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 +
192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4
 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) + 4*B^2*a^4*c*f^2 +
 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2
*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*
c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4
*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(6
8*B^2*a^3*b^12*d^11*f^2 + 20*B^2*a^5*b^10*d^11*f^2 - 88*B^2*a^7*b^8*d^11*f^
2 + 40*B^2*a^9*b^6*d^11*f^2 + 84*B^2*a^11*b^4*d^11*f^2 + 4*B^2*a^13*b^2*d^1
1*f^2 + 36*B^2*b^15*c^3*d^8*f^2 + 36*B^2*a*b^14*c^2*d^9*f^2 - 128*B^2*a^2*b
^13*c*d^10*f^2 - 112*B^2*a^4*b^11*c*d^10*f^2 + 128*B^2*a^6*b^9*c*d^10*f^2 +
 32*B^2*a^8*b^7*c*d^10*f^2 - 128*B^2*a^10*b^5*c*d^10*f^2 - 48*B^2*a^12*b^3*
c*d^10*f^2 - 68*B^2*a^2*b^13*c^3*d^8*f^2 + 204*B^2*a^3*b^12*c^2*d^9*f^2 - 1
84*B^2*a^4*b^11*c^3*d^8*f^2 + 200*B^2*a^5*b^10*c^2*d^9*f^2 - 40*B^2*a^6*b^9

*c^3*d^8*f^2 - 8*B^2*a^7*b^8*c^2*d^9*f^2 + 20*B^2*a^8*b^7*c^3*d^8*f^2 + 20*
B^2*a^9*b^6*c^2*d^9*f^2 - 20*B^2*a^10*b^5*c^3*d^8*f^2 + 60*B^2*a^11*b^4*c^2
*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2
*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^
2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c
*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((((
512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16
*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*
B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 8
96*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/
2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*
d*f^2 - 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^
8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a
^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((8*(32*B*a^
2*b^15*d^12*f^4 + 96*B*a^4*b^13*d^12*f^4 - 320*B*a^8*b^9*d^12*f^4 - 480*B*a
^10*b^7*d^12*f^4 - 288*B*a^12*b^5*d^12*f^4 - 64*B*a^14*b^3*d^12*f^4 + 64*B*
b^17*c^2*d^10*f^4 + 48*B*b^17*c^4*d^8*f^4 - 112*B*a*b^16*c^3*d^9*f^4 - 400*
B*a^3*b^14*c*d^11*f^4 - 544*B*a^5*b^12*c*d^11*f^4 - 80*B*a^7*b^10*c*d^11*f^
4 + 480*B*a^9*b^8*c*d^11*f^4 + 464*B*a^11*b^6*c*d^11*f^4 + 160*B*a^13*b^4*c
*d^11*f^4 + 16*B*a^15*b^2*c*d^11*f^4 + 368*B*a^2*b^15*c^2*d^10*f^4 + 224*B*
a^2*b^15*c^4*d^8*f^4 - 512*B*a^3*b^14*c^3*d^9*f^4 + 832*B*a^4*b^13*c^2*d^10
*f^4 + 400*B*a^4*b^13*c^4*d^8*f^4 - 880*B*a^5*b^12*c^3*d^9*f^4 + 880*B*a^6*
b^11*c^2*d^10*f^4 + 320*B*a^6*b^11*c^4*d^8*f^4 - 640*B*a^7*b^10*c^3*d^9*f^4
 + 320*B*a^8*b^9*c^2*d^10*f^4 + 80*B*a^8*b^9*c^4*d^8*f^4 - 80*B*a^9*b^8*c^3
*d^9*f^4 - 176*B*a^10*b^7*c^2*d^10*f^4 - 32*B*a^10*b^7*c^4*d^8*f^4 + 128*B*
a^11*b^6*c^3*d^9*f^4 - 192*B*a^12*b^5*c^2*d^10*f^4 - 16*B*a^12*b^5*c^4*d^8*
f^4 + 48*B*a^13*b^4*c^3*d^9*f^4 - 48*B*a^14*b^3*c^2*d^10*f^4 - 96*B*a*b^16*
c*d^11*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c
^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*
d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b
*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + (4
*(((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4
- 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 -
608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4
 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)
^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^
3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4
+ b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 +
 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*t
an(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*
b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^
7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^1
0*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*
d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^
15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720

*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))*(((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*1i)/(2*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)) - (atan((((-((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(3*B^4*a^2*b^9*d^10 - 3*B^4*a^4*b^7*d^10 + 17*B^4*a^6*b^5*d^10 - 9*B^4*a^8*b^3*d^10 + 6*B^4*b^11*c^2*d^8 - 8*B^4*a^2*b

^9*c^2*d^8 + 14*B^4*a^4*b^7*c^2*d^8 - 4*B^4*a^6*b^5*c^2*d^8 - 8*B^4*a*b^10*
c*d^9 + 12*B^4*a^3*b^8*c*d^9 - 32*B^4*a^5*b^6*c*d^9 + 12*B^4*a^7*b^4*c*d^9)
)/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*
a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6
*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 -
8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) + ((-((512*B^4*
a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8
*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*
b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a
^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) - 4*B
^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 +
24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^
4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d
^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((8*(52*B^3*a^3*b^10
*d^11*f^2 - 128*B^3*a^5*b^8*d^11*f^2 - 24*B^3*a^7*b^6*d^11*f^2 + 160*B^3*a^
9*b^4*d^11*f^2 + 4*B^3*a^11*b^2*d^11*f^2 + 12*B^3*b^13*c^3*d^8*f^2 + 44*B^3
*a*b^12*c^2*d^9*f^2 - 128*B^3*a^2*b^11*c*d^10*f^2 + 48*B^3*a^4*b^9*c*d^10*f
^2 + 176*B^3*a^6*b^7*c*d^10*f^2 - 48*B^3*a^8*b^5*c*d^10*f^2 - 48*B^3*a^10*b
^3*c*d^10*f^2 - 112*B^3*a^2*b^11*c^3*d^8*f^2 + 192*B^3*a^3*b^10*c^2*d^9*f^2
 - 24*B^3*a^4*b^9*c^3*d^8*f^2 - 72*B^3*a^5*b^8*c^2*d^9*f^2 + 80*B^3*a^6*b^7
*c^3*d^8*f^2 - 160*B^3*a^7*b^6*c^2*d^9*f^2 - 20*B^3*a^8*b^5*c^3*d^8*f^2 + 6
0*B^3*a^9*b^4*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^
5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f
^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*
d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^
3*c*d*f^5) - ((-((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^
2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*
b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a
^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*
a^7*b*c*d*f^4)^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f
^2 + 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4
+ b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6
*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^
(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(68*B^2*a^3*b^12*d^11*f^2 + 20*B^2*a^
5*b^10*d^11*f^2 - 88*B^2*a^7*b^8*d^11*f^2 + 40*B^2*a^9*b^6*d^11*f^2 + 84*B^
2*a^11*b^4*d^11*f^2 + 4*B^2*a^13*b^2*d^11*f^2 + 36*B^2*b^15*c^3*d^8*f^2 + 3
6*B^2*a*b^14*c^2*d^9*f^2 - 128*B^2*a^2*b^13*c*d^10*f^2 - 112*B^2*a^4*b^11*c
*d^10*f^2 + 128*B^2*a^6*b^9*c*d^10*f^2 + 32*B^2*a^8*b^7*c*d^10*f^2 - 128*B^
2*a^10*b^5*c*d^10*f^2 - 48*B^2*a^12*b^3*c*d^10*f^2 - 68*B^2*a^2*b^13*c^3*d^
8*f^2 + 204*B^2*a^3*b^12*c^2*d^9*f^2 - 184*B^2*a^4*b^11*c^3*d^8*f^2 + 200*B
^2*a^5*b^10*c^2*d^9*f^2 - 40*B^2*a^6*b^9*c^3*d^8*f^2 - 8*B^2*a^7*b^8*c^2*d^
9*f^2 + 20*B^2*a^8*b^7*c^3*d^8*f^2 + 20*B^2*a^9*b^6*c^2*d^9*f^2 - 20*B^2*a^
10*b^5*c^3*d^8*f^2 + 60*B^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2
*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*
c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b

^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5
*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) + ((-((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b
^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2
*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*
b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a
*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c
*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)*(a^8
*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*
a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4
 + 4*a^6*b^2*d^2*f^4))^(1/2)*((8*(32*B*a^2*b^15*d^12*f^4 + 96*B*a^4*b^13*d^
12*f^4 - 320*B*a^8*b^9*d^12*f^4 - 480*B*a^10*b^7*d^12*f^4 - 288*B*a^12*b^5*
d^12*f^4 - 64*B*a^14*b^3*d^12*f^4 + 64*B*b^17*c^2*d^10*f^4 + 48*B*b^17*c^4*
d^8*f^4 - 112*B*a*b^16*c^3*d^9*f^4 - 400*B*a^3*b^14*c*d^11*f^4 - 544*B*a^5*
b^12*c*d^11*f^4 - 80*B*a^7*b^10*c*d^11*f^4 + 480*B*a^9*b^8*c*d^11*f^4 + 464
*B*a^11*b^6*c*d^11*f^4 + 160*B*a^13*b^4*c*d^11*f^4 + 16*B*a^15*b^2*c*d^11*f
^4 + 368*B*a^2*b^15*c^2*d^10*f^4 + 224*B*a^2*b^15*c^4*d^8*f^4 - 512*B*a^3*b
^14*c^3*d^9*f^4 + 832*B*a^4*b^13*c^2*d^10*f^4 + 400*B*a^4*b^13*c^4*d^8*f^4
 - 880*B*a^5*b^12*c^3*d^9*f^4 + 880*B*a^6*b^11*c^2*d^10*f^4 + 320*B*a^6*b^11
*c^4*d^8*f^4 - 640*B*a^7*b^10*c^3*d^9*f^4 + 320*B*a^8*b^9*c^2*d^10*f^4 + 80
*B*a^8*b^9*c^4*d^8*f^4 - 80*B*a^9*b^8*c^3*d^9*f^4 - 176*B*a^10*b^7*c^2*d^10
*f^4 - 32*B*a^10*b^7*c^4*d^8*f^4 + 128*B*a^11*b^6*c^3*d^9*f^4 - 192*B*a^12*
b^5*c^2*d^10*f^4 - 16*B*a^12*b^5*c^4*d^8*f^4 + 48*B*a^13*b^4*c^3*d^9*f^4 -
48*B*a^14*b^3*c^2*d^10*f^4 - 96*B*a*b^16*c*d^11*f^4))/(a^10*d^2*f^5 + b^10*
c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b
^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^
8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*
a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - (4*(-((512*B^4*a^4*b^4*c^2*f^4 - 16*
B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^
6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4
*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*
B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*
b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)
*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4
 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^
2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d
^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12
*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^
4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176
*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^
4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3
*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b
^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 24
0*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f
^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4
*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b

^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*
b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5
*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*
a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 1
6*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2
*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^
6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)*(a^10*d^2*f^4 + b^10*c^2
*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*
c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b
^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5
*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^
2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2
*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^
8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6
*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^
4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8
*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^
2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)))*1i)/(4*(a^8*c^2*f^
4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4
*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^
6*b^2*d^2*f^4)) + ((-((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B
^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4
*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*
B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128
*B^4*a^7*b*c*d*f^4)^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 - 16*B^2*a*b^
3*d*f^2 + 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2
*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 +
4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f
^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(3*B^4*a^2*b^9*d^10 - 3*B^4*a^4*
b^7*d^10 + 17*B^4*a^6*b^5*d^10 - 9*B^4*a^8*b^3*d^10 + 6*B^4*b^11*c^2*d^8 -
8*B^4*a^2*b^9*c^2*d^8 + 14*B^4*a^4*b^7*c^2*d^8 - 4*B^4*a^6*b^5*c^2*d^8 - 8*
B^4*a*b^10*c*d^9 + 12*B^4*a^3*b^8*c*d^9 - 32*B^4*a^5*b^6*c*d^9 + 12*B^4*a^7
*b^4*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c
^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*
d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b
*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((
-((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 -
16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 6
08*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4
+ 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^
(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3
*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 +
b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 +
4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((8*(52*B
^3*a^3*b^10*d^11*f^2 - 128*B^3*a^5*b^8*d^11*f^2 - 24*B^3*a^7*b^6*d^11*f^2 +

160*B^3*a^9*b^4*d^11*f^2 + 4*B^3*a^11*b^2*d^11*f^2 + 12*B^3*b^13*c^3*d^8*f
^2 + 44*B^3*a*b^12*c^2*d^9*f^2 - 128*B^3*a^2*b^11*c*d^10*f^2 + 48*B^3*a^4*b
^9*c*d^10*f^2 + 176*B^3*a^6*b^7*c*d^10*f^2 - 48*B^3*a^8*b^5*c*d^10*f^2 - 48
*B^3*a^10*b^3*c*d^10*f^2 - 112*B^3*a^2*b^11*c^3*d^8*f^2 + 192*B^3*a^3*b^10*
c^2*d^9*f^2 - 24*B^3*a^4*b^9*c^3*d^8*f^2 - 72*B^3*a^5*b^8*c^2*d^9*f^2 + 80*
B^3*a^6*b^7*c^3*d^8*f^2 - 160*B^3*a^7*b^6*c^2*d^9*f^2 - 20*B^3*a^8*b^5*c^3*
d^8*f^2 + 60*B^3*a^9*b^4*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2
*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^
2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 -
2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5
- 8*a^7*b^3*c*d*f^5) + ((-((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 -
256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 1
92*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4
- 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4
- 128*B^4*a^7*b*c*d*f^4)^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 - 16*B^
2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a
^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*
f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2
*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(68*B^2*a^3*b^12*d^11*f^2
+ 20*B^2*a^5*b^10*d^11*f^2 - 88*B^2*a^7*b^8*d^11*f^2 + 40*B^2*a^9*b^6*d^11*
f^2 + 84*B^2*a^11*b^4*d^11*f^2 + 4*B^2*a^13*b^2*d^11*f^2 + 36*B^2*b^15*c^3*
d^8*f^2 + 36*B^2*a*b^14*c^2*d^9*f^2 - 128*B^2*a^2*b^13*c*d^10*f^2 - 112*B^2
*a^4*b^11*c*d^10*f^2 + 128*B^2*a^6*b^9*c*d^10*f^2 + 32*B^2*a^8*b^7*c*d^10*f
^2 - 128*B^2*a^10*b^5*c*d^10*f^2 - 48*B^2*a^12*b^3*c*d^10*f^2 - 68*B^2*a^2*
b^13*c^3*d^8*f^2 + 204*B^2*a^3*b^12*c^2*d^9*f^2 - 184*B^2*a^4*b^11*c^3*d^8*
f^2 + 200*B^2*a^5*b^10*c^2*d^9*f^2 - 40*B^2*a^6*b^9*c^3*d^8*f^2 - 8*B^2*a^7
*b^8*c^2*d^9*f^2 + 20*B^2*a^8*b^7*c^3*d^8*f^2 + 20*B^2*a^9*b^6*c^2*d^9*f^2
- 20*B^2*a^10*b^5*c^3*d^8*f^2 + 60*B^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4
+ b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4
+ a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^
4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f
^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((-((512*B^4*a^4*b^4*c^2*f^4
- 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*
B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 1
92*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4
+ 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) - 4*B^2*a^4*c*f^2 -
4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*
c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c
^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*
b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((8*(32*B*a^2*b^15*d^12*f^4 + 96*B*
a^4*b^13*d^12*f^4 - 320*B*a^8*b^9*d^12*f^4 - 480*B*a^10*b^7*d^12*f^4 - 288*
B*a^12*b^5*d^12*f^4 - 64*B*a^14*b^3*d^12*f^4 + 64*B*b^17*c^2*d^10*f^4 + 48*
B*b^17*c^4*d^8*f^4 - 112*B*a*b^16*c^3*d^9*f^4 - 400*B*a^3*b^14*c*d^11*f^4 -
544*B*a^5*b^12*c*d^11*f^4 - 80*B*a^7*b^10*c*d^11*f^4 + 480*B*a^9*b^8*c*d^1
1*f^4 + 464*B*a^11*b^6*c*d^11*f^4 + 160*B*a^13*b^4*c*d^11*f^4 + 16*B*a^15*b

^2*c*d^11*f^4 + 368*B*a^2*b^15*c^2*d^10*f^4 + 224*B*a^2*b^15*c^4*d^8*f^4 - 512*B*a^3*b^14*c^3*d^9*f^4 + 832*B*a^4*b^13*c^2*d^10*f^4 + 400*B*a^4*b^13*c^4*d^8*f^4 - 880*B*a^5*b^12*c^3*d^9*f^4 + 880*B*a^6*b^11*c^2*d^10*f^4 + 320*B*a^6*b^11*c^4*d^8*f^4 - 640*B*a^7*b^10*c^3*d^9*f^4 + 320*B*a^8*b^9*c^2*d^10*f^4 + 80*B*a^8*b^9*c^4*d^8*f^4 - 80*B*a^9*b^8*c^3*d^9*f^4 - 176*B*a^10*b^7*c^2*d^10*f^4 - 32*B*a^10*b^7*c^4*d^8*f^4 + 128*B*a^11*b^6*c^3*d^9*f^4 - 192*B*a^12*b^5*c^2*d^10*f^4 - 16*B*a^12*b^5*c^4*d^8*f^4 + 48*B*a^13*b^4*c^3*d^9*f^4 - 48*B*a^14*b^3*c^2*d^10*f^4 - 96*B*a*b^16*c*d^11*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + (4*(-((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4 *b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)))*1i)/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)))/((16*(9*B^5*a^6*b^3*d^10 - B^5*a^2*b^7*d^10 - 4*B^5*a^2*b^7*c^2*d^8 + 4*B^5*a^4*b^5*c^2*d^8 + 2*B^5*a*b^8*c*d^9 + 6*B^5*a^3*b^6*c*d^9 - 12*B^5*a^5*b^4*c*d^9))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + ((-((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(3*B^4*a^2*b^9*d^10 - 3*B^4*a^4*b^7*d^10 + 17*B^4*a^6*b^5*d^10 - 9*B^4*a^8*b^3*d^10 + 6*B^4*b^11*c^2*d^8 - 8*B^4*a^2*b^9*c^2*d^8 + 14*B^4*a^4*b^7*c^2*d^8 - 4*B^4*a^6*b^5*c^2*d^8 - 8*B^4*a*b^10*c*d^9 + 12*B^4*a^3*b^8*c*d^9 - 32*B^4*a^5*b^6*c*d^9 + 12*B^4*a^7*b^4*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) + ((-((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((8*(52*B^3*a^3*b^10*d^11*f^2 - 128*B^3*a^5*b^8*d^11*f^2 - 24*B^3*a^7*b^6*d^11*f^2 + 160*B^3*a^9*b^4*d^11*f^2 + 4*B^3*a^11*b^2*d^11*f^2 + 12*B^3*b^13*c^3*d^8*f^2 + 44*B^3*a*b^12*c^2*d^9*f^2 - 128*B^3*a^2*b^11*c*d^10*f^2 + 48*B^3*a^4*b^9*c*d^10*f^2 + 176*B^3*a^6*b^7*c*d^10*f^2 - 48*B^3*a^8*b^5*c*d^10*f^2 - 48*B^3*a^10*b^3*c*d^10*f^2 - 112*B^3*a^2*b^11*c^3*d^8*f^2 + 192*B^3*a^3*b^10*c^2*d^9*f^2 - 24*B^3*a^4*b^9*c^3*d^8*f^2 - 72*B^3*a^5*b^8*c^2*d^9*f^2 + 80*B^3*a^6*b^7*c^3*d^8*f^2 - 160*B^3*a^7*b^6*c^2*d^9*f^2 - 20*B^3*a^8*b^5*c^3*d^8*f^2 + 60*B^3*a^9*b^4*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f

^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d
*f^5 - 8*a^7*b^3*c*d*f^5) - ((-((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f
^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4
 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*
f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d
*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 - 1
6*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4
 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*
c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6
*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(68*B^2*a^3*b^12*d^11*
f^2 + 20*B^2*a^5*b^10*d^11*f^2 - 88*B^2*a^7*b^8*d^11*f^2 + 40*B^2*a^9*b^6*d
^11*f^2 + 84*B^2*a^11*b^4*d^11*f^2 + 4*B^2*a^13*b^2*d^11*f^2 + 36*B^2*b^15*
c^3*d^8*f^2 + 36*B^2*a*b^14*c^2*d^9*f^2 - 128*B^2*a^2*b^13*c*d^10*f^2 - 112
*B^2*a^4*b^11*c*d^10*f^2 + 128*B^2*a^6*b^9*c*d^10*f^2 + 32*B^2*a^8*b^7*c*d^
10*f^2 - 128*B^2*a^10*b^5*c*d^10*f^2 - 48*B^2*a^12*b^3*c*d^10*f^2 - 68*B^2*
a^2*b^13*c^3*d^8*f^2 + 204*B^2*a^3*b^12*c^2*d^9*f^2 - 184*B^2*a^4*b^11*c^3*
d^8*f^2 + 200*B^2*a^5*b^10*c^2*d^9*f^2 - 40*B^2*a^6*b^9*c^3*d^8*f^2 - 8*B^2
*a^7*b^8*c^2*d^9*f^2 + 20*B^2*a^8*b^7*c^3*d^8*f^2 + 20*B^2*a^9*b^6*c^2*d^9*
f^2 - 20*B^2*a^10*b^5*c^3*d^8*f^2 + 60*B^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2
*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2
*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^
2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c
*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) + ((-((512*B^4*a^4*b^4*c^2
*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 -
256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4
 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*
f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) - 4*B^2*a^4*c*f^
2 - 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*
b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b
^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*
a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((8*(32*B*a^2*b^15*d^12*f^4 + 9
6*B*a^4*b^13*d^12*f^4 - 320*B*a^8*b^9*d^12*f^4 - 480*B*a^10*b^7*d^12*f^4 -
288*B*a^12*b^5*d^12*f^4 - 64*B*a^14*b^3*d^12*f^4 + 64*B*b^17*c^2*d^10*f^4 +
 48*B*b^17*c^4*d^8*f^4 - 112*B*a*b^16*c^3*d^9*f^4 - 400*B*a^3*b^14*c*d^11*f
^4 - 544*B*a^5*b^12*c*d^11*f^4 - 80*B*a^7*b^10*c*d^11*f^4 + 480*B*a^9*b^8*c
*d^11*f^4 + 464*B*a^11*b^6*c*d^11*f^4 + 160*B*a^13*b^4*c*d^11*f^4 + 16*B*a^
15*b^2*c*d^11*f^4 + 368*B*a^2*b^15*c^2*d^10*f^4 + 224*B*a^2*b^15*c^4*d^8*f^
4 - 512*B*a^3*b^14*c^3*d^9*f^4 + 832*B*a^4*b^13*c^2*d^10*f^4 + 400*B*a^4*b^
13*c^4*d^8*f^4 - 880*B*a^5*b^12*c^3*d^9*f^4 + 880*B*a^6*b^11*c^2*d^10*f^4 +
 320*B*a^6*b^11*c^4*d^8*f^4 - 640*B*a^7*b^10*c^3*d^9*f^4 + 320*B*a^8*b^9*c^
2*d^10*f^4 + 80*B*a^8*b^9*c^4*d^8*f^4 - 80*B*a^9*b^8*c^3*d^9*f^4 - 176*B*a^
10*b^7*c^2*d^10*f^4 - 32*B*a^10*b^7*c^4*d^8*f^4 + 128*B*a^11*b^6*c^3*d^9*f^
4 - 192*B*a^12*b^5*c^2*d^10*f^4 - 16*B*a^12*b^5*c^4*d^8*f^4 + 48*B*a^13*b^4
*c^3*d^9*f^4 - 48*B*a^14*b^3*c^2*d^10*f^4 - 96*B*a*b^16*c*d^11*f^4))/(a^10*
d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*

c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4
*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^
7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - (4*(-((512*B^4*a^4*b^
4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f
^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^
2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3
*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) - 4*B^2*a^4
*c*f^2 - 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 + 24*B^2
*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*
a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4
 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)
*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 16
0*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a
^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^
4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*
b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 9
12*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*
f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c
^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^1
0*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 -
48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*
f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d
^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^1
1*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d
^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4
*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^
2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2
*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)*(a^10*d^2
*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2
*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^
2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c
*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^8*c^2*f^4 + a^8*d
^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4
+ 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2
f^4))))/(4(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*
b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6
*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^
8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2
*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4
*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4
 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^
2*f^4 + 4*a^6*b^2*d^2*f^4)) - ((-((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2
*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f
^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^
2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c

*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 -
16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f
^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^
4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a
^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(3*B^4*a^2*b^9*d^10
- 3*B^4*a^4*b^7*d^10 + 17*B^4*a^6*b^5*d^10 - 9*B^4*a^8*b^3*d^10 + 6*B^4*b^1
1*c^2*d^8 - 8*B^4*a^2*b^9*c^2*d^8 + 14*B^4*a^4*b^7*c^2*d^8 - 4*B^4*a^6*b^5*
c^2*d^8 - 8*B^4*a*b^10*c*d^9 + 12*B^4*a^3*b^8*c*d^9 - 32*B^4*a^5*b^6*c*d^9
+ 12*B^4*a^7*b^4*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 +
6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4
+ 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f
^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c
*d*f^4) - ((-((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b
^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6
*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*
b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7
*b*c*d*f^4)^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2
+ 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b
^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^
2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/
2)*((8*(52*B^3*a^3*b^10*d^11*f^2 - 128*B^3*a^5*b^8*d^11*f^2 - 24*B^3*a^7*b^
6*d^11*f^2 + 160*B^3*a^9*b^4*d^11*f^2 + 4*B^3*a^11*b^2*d^11*f^2 + 12*B^3*b^
13*c^3*d^8*f^2 + 44*B^3*a*b^12*c^2*d^9*f^2 - 128*B^3*a^2*b^11*c*d^10*f^2 +
48*B^3*a^4*b^9*c*d^10*f^2 + 176*B^3*a^6*b^7*c*d^10*f^2 - 48*B^3*a^8*b^5*c*d
^10*f^2 - 48*B^3*a^10*b^3*c*d^10*f^2 - 112*B^3*a^2*b^11*c^3*d^8*f^2 + 192*B
^3*a^3*b^10*c^2*d^9*f^2 - 24*B^3*a^4*b^9*c^3*d^8*f^2 - 72*B^3*a^5*b^8*c^2*d
^9*f^2 + 80*B^3*a^6*b^7*c^3*d^8*f^2 - 160*B^3*a^7*b^6*c^2*d^9*f^2 - 20*B^3*
a^8*b^5*c^3*d^8*f^2 + 60*B^3*a^9*b^4*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2
*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*
c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b
^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5
*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + ((-((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b
^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2
*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*
b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a
*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c
*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)*(a^8
*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*
a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4
+ 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(68*B^2*a^3*b^
12*d^11*f^2 + 20*B^2*a^5*b^10*d^11*f^2 - 88*B^2*a^7*b^8*d^11*f^2 + 40*B^2*a
^9*b^6*d^11*f^2 + 84*B^2*a^11*b^4*d^11*f^2 + 4*B^2*a^13*b^2*d^11*f^2 + 36*B
^2*b^15*c^3*d^8*f^2 + 36*B^2*a*b^14*c^2*d^9*f^2 - 128*B^2*a^2*b^13*c*d^10*f
^2 - 112*B^2*a^4*b^11*c*d^10*f^2 + 128*B^2*a^6*b^9*c*d^10*f^2 + 32*B^2*a^8*
b^7*c*d^10*f^2 - 128*B^2*a^10*b^5*c*d^10*f^2 - 48*B^2*a^12*b^3*c*d^10*f^2 -

68*B^2*a^2*b^13*c^3*d^8*f^2 + 204*B^2*a^3*b^12*c^2*d^9*f^2 - 184*B^2*a^4*b
^11*c^3*d^8*f^2 + 200*B^2*a^5*b^10*c^2*d^9*f^2 - 40*B^2*a^6*b^9*c^3*d^8*f^2
 - 8*B^2*a^7*b^8*c^2*d^9*f^2 + 20*B^2*a^8*b^7*c^3*d^8*f^2 + 20*B^2*a^9*b^6*
c^2*d^9*f^2 - 20*B^2*a^10*b^5*c^3*d^8*f^2 + 60*B^2*a^11*b^4*c^2*d^9*f^2))/(
a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6
*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^
6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a
^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((-((512*B^4*a^4
*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^
2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4
*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*
b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) - 4*B^2*
a^4*c*f^2 - 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 + 24*
B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 +
 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*
f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((8*(32*B*a^2*b^15*d^12
*f^4 + 96*B*a^4*b^13*d^12*f^4 - 320*B*a^8*b^9*d^12*f^4 - 480*B*a^10*b^7*d^1
2*f^4 - 288*B*a^12*b^5*d^12*f^4 - 64*B*a^14*b^3*d^12*f^4 + 64*B*b^17*c^2*d^
10*f^4 + 48*B*b^17*c^4*d^8*f^4 - 112*B*a*b^16*c^3*d^9*f^4 - 400*B*a^3*b^14*
c*d^11*f^4 - 544*B*a^5*b^12*c*d^11*f^4 - 80*B*a^7*b^10*c*d^11*f^4 + 480*B*a
^9*b^8*c*d^11*f^4 + 464*B*a^11*b^6*c*d^11*f^4 + 160*B*a^13*b^4*c*d^11*f^4 +
 16*B*a^15*b^2*c*d^11*f^4 + 368*B*a^2*b^15*c^2*d^10*f^4 + 224*B*a^2*b^15*c^
4*d^8*f^4 - 512*B*a^3*b^14*c^3*d^9*f^4 + 832*B*a^4*b^13*c^2*d^10*f^4 + 400*
B*a^4*b^13*c^4*d^8*f^4 - 880*B*a^5*b^12*c^3*d^9*f^4 + 880*B*a^6*b^11*c^2*d^
10*f^4 + 320*B*a^6*b^11*c^4*d^8*f^4 - 640*B*a^7*b^10*c^3*d^9*f^4 + 320*B*a^
8*b^9*c^2*d^10*f^4 + 80*B*a^8*b^9*c^4*d^8*f^4 - 80*B*a^9*b^8*c^3*d^9*f^4 -
176*B*a^10*b^7*c^2*d^10*f^4 - 32*B*a^10*b^7*c^4*d^8*f^4 + 128*B*a^11*b^6*c^
3*d^9*f^4 - 192*B*a^12*b^5*c^2*d^10*f^4 - 16*B*a^12*b^5*c^4*d^8*f^4 + 48*B*
a^13*b^4*c^3*d^9*f^4 - 48*B*a^14*b^3*c^2*d^10*f^4 - 96*B*a*b^16*c*d^11*f^4)
)/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*
a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6
*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 -
8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + (4*(-((512*B^
4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a
^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^
4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4
*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) - 4
*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2
+ 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*
f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6
*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*tan(e + f*x
))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*
f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4
 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48
*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 -

432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8
*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*
c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^
8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4
+ 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^1
0*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c
^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b
^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^
16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*
b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*
a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8*d^2*f^4
+ b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6
*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)*(
a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6
*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^
6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a
^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^8*c^2*f^4
+ a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*
c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6
*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4
+ 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2
*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*
f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4
*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^
4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6
*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^
4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))*(-((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*
b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^
2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6
*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*
a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*
c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)*(a^
8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6
*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^
4 + 4*a^6*b^2*d^2*f^4))^(1/2)*1i)/(2*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f
^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^
4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)) - (atan((((
-(4*B^2*b^7*c^2 - 8*B^2*a^2*b^5*c^2 + 4*B^2*a^4*b^3*c^2 + B^2*a^2*b^5*d^2 -
6*B^2*a^4*b^3*d^2 + 9*B^2*a^6*b*d^2 + 16*B^2*a^3*b^4*c*d - 12*B^2*a^5*b^2*
c*d - 4*B^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6
*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 +
4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*
f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*
a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^
4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*((16*(c +

d*tan(e + f*x))^(1/2)*(3*B^4*a^2*b^9*d^10 - 3*B^4*a^4*b^7*d^10 + 17*B^4*a^6
*b^5*d^10 - 9*B^4*a^8*b^3*d^10 + 6*B^4*b^11*c^2*d^8 - 8*B^4*a^2*b^9*c^2*d^8
 + 14*B^4*a^4*b^7*c^2*d^8 - 4*B^4*a^6*b^5*c^2*d^8 - 8*B^4*a*b^10*c*d^9 + 12
*B^4*a^3*b^8*c*d^9 - 32*B^4*a^5*b^6*c*d^9 + 12*B^4*a^7*b^4*c*d^9))/(a^10*d^
2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^
2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d
^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*
c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) + ((-(4*B^2*b^7*c^2 - 8*B
^2*a^2*b^5*c^2 + 4*B^2*a^4*b^3*c^2 + B^2*a^2*b^5*d^2 - 6*B^2*a^4*b^3*d^2 +
9*B^2*a^6*b*d^2 + 16*B^2*a^3*b^4*c*d - 12*B^2*a^5*b^2*c*d - 4*B^2*a*b^6*c*d
)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*
a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6
*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*
f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 1
8*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*
b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*((8*(52*B^3*a^3*b^10*d^11*f^2 -
 128*B^3*a^5*b^8*d^11*f^2 - 24*B^3*a^7*b^6*d^11*f^2 + 160*B^3*a^9*b^4*d^11*
f^2 + 4*B^3*a^11*b^2*d^11*f^2 + 12*B^3*b^13*c^3*d^8*f^2 + 44*B^3*a*b^12*c^2
*d^9*f^2 - 128*B^3*a^2*b^11*c*d^10*f^2 + 48*B^3*a^4*b^9*c*d^10*f^2 + 176*B^
3*a^6*b^7*c*d^10*f^2 - 48*B^3*a^8*b^5*c*d^10*f^2 - 48*B^3*a^10*b^3*c*d^10*f
^2 - 112*B^3*a^2*b^11*c^3*d^8*f^2 + 192*B^3*a^3*b^10*c^2*d^9*f^2 - 24*B^3*a
^4*b^9*c^3*d^8*f^2 - 72*B^3*a^5*b^8*c^2*d^9*f^2 + 80*B^3*a^6*b^7*c^3*d^8*f^
2 - 160*B^3*a^7*b^6*c^2*d^9*f^2 - 20*B^3*a^8*b^5*c^3*d^8*f^2 + 60*B^3*a^9*b
^4*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b
^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*
b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a
^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5)
- ((-(4*B^2*b^7*c^2 - 8*B^2*a^2*b^5*c^2 + 4*B^2*a^4*b^3*c^2 + B^2*a^2*b^5*d
^2 - 6*B^2*a^4*b^3*d^2 + 9*B^2*a^6*b*d^2 + 16*B^2*a^3*b^4*c*d - 12*B^2*a^5*
b^2*c*d - 4*B^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2
 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^
2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^
2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 -
 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^
7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*((16*(
c + d*tan(e + f*x))^(1/2)*(68*B^2*a^3*b^12*d^11*f^2 + 20*B^2*a^5*b^10*d^11*
f^2 - 88*B^2*a^7*b^8*d^11*f^2 + 40*B^2*a^9*b^6*d^11*f^2 + 84*B^2*a^11*b^4*d
^11*f^2 + 4*B^2*a^13*b^2*d^11*f^2 + 36*B^2*b^15*c^3*d^8*f^2 + 36*B^2*a*b^14
*c^2*d^9*f^2 - 128*B^2*a^2*b^13*c*d^10*f^2 - 112*B^2*a^4*b^11*c*d^10*f^2 +
128*B^2*a^6*b^9*c*d^10*f^2 + 32*B^2*a^8*b^7*c*d^10*f^2 - 128*B^2*a^10*b^5*c
*d^10*f^2 - 48*B^2*a^12*b^3*c*d^10*f^2 - 68*B^2*a^2*b^13*c^3*d^8*f^2 + 204*
B^2*a^3*b^12*c^2*d^9*f^2 - 184*B^2*a^4*b^11*c^3*d^8*f^2 + 200*B^2*a^5*b^10*
c^2*d^9*f^2 - 40*B^2*a^6*b^9*c^3*d^8*f^2 - 8*B^2*a^7*b^8*c^2*d^9*f^2 + 20*B
^2*a^8*b^7*c^3*d^8*f^2 + 20*B^2*a^9*b^6*c^2*d^9*f^2 - 20*B^2*a^10*b^5*c^3*d
^8*f^2 + 60*B^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2

*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^
2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 -
 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4
 - 8*a^7*b^3*c*d*f^4) + ((-(4*B^2*b^7*c^2 - 8*B^2*a^2*b^5*c^2 + 4*B^2*a^4*b
^3*c^2 + B^2*a^2*b^5*d^2 - 6*B^2*a^4*b^3*d^2 + 9*B^2*a^6*b*d^2 + 16*B^2*a^3
*b^4*c*d - 12*B^2*a^5*b^2*c*d - 4*B^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f
^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^
3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2
*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 +
12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6
*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^
2*d*f^2))^(1/2)*((8*(32*B*a^2*b^15*d^12*f^4 + 96*B*a^4*b^13*d^12*f^4 - 320*
B*a^8*b^9*d^12*f^4 - 480*B*a^10*b^7*d^12*f^4 - 288*B*a^12*b^5*d^12*f^4 - 64
*B*a^14*b^3*d^12*f^4 + 64*B*b^17*c^2*d^10*f^4 + 48*B*b^17*c^4*d^8*f^4 - 112
*B*a*b^16*c^3*d^9*f^4 - 400*B*a^3*b^14*c*d^11*f^4 - 544*B*a^5*b^12*c*d^11*f
^4 - 80*B*a^7*b^10*c*d^11*f^4 + 480*B*a^9*b^8*c*d^11*f^4 + 464*B*a^11*b^6*c
*d^11*f^4 + 160*B*a^13*b^4*c*d^11*f^4 + 16*B*a^15*b^2*c*d^11*f^4 + 368*B*a^
2*b^15*c^2*d^10*f^4 + 224*B*a^2*b^15*c^4*d^8*f^4 - 512*B*a^3*b^14*c^3*d^9*f
^4 + 832*B*a^4*b^13*c^2*d^10*f^4 + 400*B*a^4*b^13*c^4*d^8*f^4 - 880*B*a^5*b
^12*c^3*d^9*f^4 + 880*B*a^6*b^11*c^2*d^10*f^4 + 320*B*a^6*b^11*c^4*d^8*f^4
 - 640*B*a^7*b^10*c^3*d^9*f^4 + 320*B*a^8*b^9*c^2*d^10*f^4 + 80*B*a^8*b^9*c^
4*d^8*f^4 - 80*B*a^9*b^8*c^3*d^9*f^4 - 176*B*a^10*b^7*c^2*d^10*f^4 - 32*B*a
^10*b^7*c^4*d^8*f^4 + 128*B*a^11*b^6*c^3*d^9*f^4 - 192*B*a^12*b^5*c^2*d^10*
f^4 - 16*B*a^12*b^5*c^4*d^8*f^4 + 48*B*a^13*b^4*c^3*d^9*f^4 - 48*B*a^14*b^3
*c^2*d^10*f^4 - 96*B*a*b^16*c*d^11*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a
^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 +
a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5
 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f
^5 - 8*a^7*b^3*c*d*f^5) - (16*(-(4*B^2*b^7*c^2 - 8*B^2*a^2*b^5*c^2 + 4*B^2*
a^4*b^3*c^2 + B^2*a^2*b^5*d^2 - 6*B^2*a^4*b^3*d^2 + 9*B^2*a^6*b*d^2 + 16*B^
2*a^3*b^4*c*d - 12*B^2*a^5*b^2*c*d - 4*B^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*
c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b
^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^
9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f
^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 1
8*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b
^2*c^2*d*f^2))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160
*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^
10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b
^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2
*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b
^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 11
2*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f
^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^
3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11

*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 20
8*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^
4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f
^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*
f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^
11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*
d^11*f^4))/((b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 2
4*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3
*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a
^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) -
b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 -
6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a
^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 +
a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4
 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f
^4 - 8*a^7*b^3*c*d*f^4))))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*
a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) +
 b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*
f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^
7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*
b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2)))/(b^9*(8*a^2*c^3*f
^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^
4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^
2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2
) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c
^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b
*c*d^2*f^2)))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 +
 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c
^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2
*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2)
- b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2
- 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2))*1i)/(b^9*(8*a^2*c^3*f^2 + 6*a^2
*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2
+ 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d
^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(1
2*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2)
- 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2
) + ((-(4*B^2*b^7*c^2 - 8*B^2*a^2*b^5*c^2 + 4*B^2*a^4*b^3*c^2 + B^2*a^2*b^5
*d^2 - 6*B^2*a^4*b^3*d^2 + 9*B^2*a^6*b*d^2 + 16*B^2*a^3*b^4*c*d - 12*B^2*a^
5*b^2*c*d - 4*B^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f
^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*
f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*
c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2
 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*
a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*((16

*(c + d*tan(e + f*x))^(1/2)*(3*B^4*a^2*b^9*d^10 - 3*B^4*a^4*b^7*d^10 + 17*B
^4*a^6*b^5*d^10 - 9*B^4*a^8*b^3*d^10 + 6*B^4*b^11*c^2*d^8 - 8*B^4*a^2*b^9*c
^2*d^8 + 14*B^4*a^4*b^7*c^2*d^8 - 4*B^4*a^6*b^5*c^2*d^8 - 8*B^4*a*b^10*c*d^
9 + 12*B^4*a^3*b^8*c*d^9 - 32*B^4*a^5*b^6*c*d^9 + 12*B^4*a^7*b^4*c*d^9))/(a
^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*
b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6
*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^
3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((-(4*B^2*b^7*c^2
 - 8*B^2*a^2*b^5*c^2 + 4*B^2*a^4*b^3*c^2 + B^2*a^2*b^5*d^2 - 6*B^2*a^4*b^3*
d^2 + 9*B^2*a^6*b*d^2 + 16*B^2*a^3*b^4*c*d - 12*B^2*a^5*b^2*c*d - 4*B^2*a*b
^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^
2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f
^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*
c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f
^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 1
2*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*((8*(52*B^3*a^3*b^10*d^11
*f^2 - 128*B^3*a^5*b^8*d^11*f^2 - 24*B^3*a^7*b^6*d^11*f^2 + 160*B^3*a^9*b^4
*d^11*f^2 + 4*B^3*a^11*b^2*d^11*f^2 + 12*B^3*b^13*c^3*d^8*f^2 + 44*B^3*a*b^
12*c^2*d^9*f^2 - 128*B^3*a^2*b^11*c*d^10*f^2 + 48*B^3*a^4*b^9*c*d^10*f^2 +
176*B^3*a^6*b^7*c*d^10*f^2 - 48*B^3*a^8*b^5*c*d^10*f^2 - 48*B^3*a^10*b^3*c*
d^10*f^2 - 112*B^3*a^2*b^11*c^3*d^8*f^2 + 192*B^3*a^3*b^10*c^2*d^9*f^2 - 24
*B^3*a^4*b^9*c^3*d^8*f^2 - 72*B^3*a^5*b^8*c^2*d^9*f^2 + 80*B^3*a^6*b^7*c^3*
d^8*f^2 - 160*B^3*a^7*b^6*c^2*d^9*f^2 - 20*B^3*a^8*b^5*c^3*d^8*f^2 + 60*B^3
*a^9*b^4*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6
*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 +
4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5
 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d
*f^5) + ((-(4*B^2*b^7*c^2 - 8*B^2*a^2*b^5*c^2 + 4*B^2*a^4*b^3*c^2 + B^2*a^2
*b^5*d^2 - 6*B^2*a^4*b^3*d^2 + 9*B^2*a^6*b*d^2 + 16*B^2*a^3*b^4*c*d - 12*B^
2*a^5*b^2*c*d - 4*B^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c
^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*
d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b
^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d
*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 +
 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*
((16*(c + d*tan(e + f*x))^(1/2)*(68*B^2*a^3*b^12*d^11*f^2 + 20*B^2*a^5*b^10
*d^11*f^2 - 88*B^2*a^7*b^8*d^11*f^2 + 40*B^2*a^9*b^6*d^11*f^2 + 84*B^2*a^11
*b^4*d^11*f^2 + 4*B^2*a^13*b^2*d^11*f^2 + 36*B^2*b^15*c^3*d^8*f^2 + 36*B^2*
a*b^14*c^2*d^9*f^2 - 128*B^2*a^2*b^13*c*d^10*f^2 - 112*B^2*a^4*b^11*c*d^10*
f^2 + 128*B^2*a^6*b^9*c*d^10*f^2 + 32*B^2*a^8*b^7*c*d^10*f^2 - 128*B^2*a^10
*b^5*c*d^10*f^2 - 48*B^2*a^12*b^3*c*d^10*f^2 - 68*B^2*a^2*b^13*c^3*d^8*f^2
+ 204*B^2*a^3*b^12*c^2*d^9*f^2 - 184*B^2*a^4*b^11*c^3*d^8*f^2 + 200*B^2*a^5
*b^10*c^2*d^9*f^2 - 40*B^2*a^6*b^9*c^3*d^8*f^2 - 8*B^2*a^7*b^8*c^2*d^9*f^2
+ 20*B^2*a^8*b^7*c^3*d^8*f^2 + 20*B^2*a^9*b^6*c^2*d^9*f^2 - 20*B^2*a^10*b^5
*c^3*d^8*f^2 + 60*B^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 +

4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((-(4*B^2*b^7*c^2 - 8*B^2*a^2*b^5*c^2 + 4*B^2*a^4*b^3*c^2 + B^2*a^2*b^5*d^2 - 6*B^2*a^4*b^3*d^2 + 9*B^2*a^6*b*d^2 + 16*B^2*a^3*b^4*c*d - 12*B^2*a^5*b^2*c*d - 4*B^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*((8*(32*B*a^2*b^15*d^12*f^4 + 96*B*a^4*b^13*d^12*f^4 - 320*B*a^8*b^9*d^12*f^4 - 480*B*a^10*b^7*d^12*f^4 - 288*B*a^12*b^5*d^12*f^4 - 64*B*a^14*b^3*d^12*f^4 + 64*B*b^17*c^2*d^10*f^4 + 48*B*b^17*c^4*d^8*f^4 - 112*B*a*b^16*c^3*d^9*f^4 - 400*B*a^3*b^14*c*d^11*f^4 - 544*B*a^5*b^12*c*d^11*f^4 - 80*B*a^7*b^10*c*d^11*f^4 + 480*B*a^9*b^8*c*d^11*f^4 + 464*B*a^11*b^6*c*d^11*f^4 + 160*B*a^13*b^4*c*d^11*f^4 + 16*B*a^15*b^2*c*d^11*f^4 + 368*B*a^2*b^15*c^2*d^10*f^4 + 224*B*a^2*b^15*c^4*d^8*f^4 - 512*B*a^3*b^14*c^3*d^9*f^4 + 832*B*a^4*b^13*c^2*d^10*f^4 + 400*B*a^4*b^13*c^4*d^8*f^4 - 880*B*a^5*b^12*c^3*d^9*f^4 + 880*B*a^6*b^11*c^2*d^10*f^4 + 320*B*a^6*b^11*c^4*d^8*f^4 - 640*B*a^7*b^10*c^3*d^9*f^4 + 320*B*a^8*b^9*c^2*d^10*f^4 + 80*B*a^8*b^9*c^4*d^8*f^4 - 80*B*a^9*b^8*c^3*d^9*f^4 - 176*B*a^10*b^7*c^2*d^10*f^4 - 32*B*a^10*b^7*c^4*d^8*f^4 + 128*B*a^11*b^6*c^3*d^9*f^4 - 192*B*a^12*b^5*c^2*d^10*f^4 - 16*B*a^12*b^5*c^4*d^8*f^4 + 48*B*a^13*b^4*c^3*d^9*f^4 - 48*B*a^14*b^3*c^2*d^10*f^4 - 96*B*a*b^16*c*d^11*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + (16*(-(4*B^2*b^7*c^2 - 8*B^2*a^2*b^5*c^2 + 4*B^2*a^4*b^3*c^2 + B^2*a^2*b^5*d^2 - 6*B^2*a^4*b^3*d^2 + 9*B^2*a^6*b*d^2 + 16*B^2*a^3*b^4*c*d - 12*B^2*a^5*b^2*c*d - 4*B^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 24

0*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^
4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*
d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*
d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c
*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^
8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*
b^2*c*d^11*f^4))/((b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f
^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a
^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^
8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f
^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*
f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2)*(a^10*d^2*f^4 + b^10*c^2*f^4
 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*
f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d
^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5
*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b
^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*
f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*
c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 +
 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^
2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2)))/(b^9*(8*a^2
*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*
(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2
) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2
*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36
*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*
a^10*b*c*d^2*f^2)))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3
*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8
*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) -
b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d
f^2) - b^6(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^
3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2))*1i)/(b^9*(8*a^2*c^3*f^2 +
 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^
3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8
*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) -
b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d
*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d
^2*f^2))/((16*(9*B^5*a^6*b^3*d^10 - B^5*a^2*b^7*d^10 - 4*B^5*a^2*b^7*c^2*d^
8 + 4*B^5*a^4*b^5*c^2*d^8 + 2*B^5*a*b^8*c*d^9 + 6*B^5*a^3*b^6*c*d^9 - 12*B^
5*a^5*b^4*c*d^9))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*
b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4
*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*
a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5)
 + ((-(4*B^2*b^7*c^2 - 8*B^2*a^2*b^5*c^2 + 4*B^2*a^4*b^3*c^2 + B^2*a^2*b^5*
d^2 - 6*B^2*a^4*b^3*d^2 + 9*B^2*a^6*b*d^2 + 16*B^2*a^3*b^4*c*d - 12*B^2*a^5

*b^2*c*d - 4*B^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^
2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f
^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c
^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2
- 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a
^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*((16*
(c + d*tan(e + f*x))^(1/2)*(3*B^4*a^2*b^9*d^10 - 3*B^4*a^4*b^7*d^10 + 17*B^
4*a^6*b^5*d^10 - 9*B^4*a^8*b^3*d^10 + 6*B^4*b^11*c^2*d^8 - 8*B^4*a^2*b^9*c^
2*d^8 + 14*B^4*a^4*b^7*c^2*d^8 - 4*B^4*a^6*b^5*c^2*d^8 - 8*B^4*a*b^10*c*d^9
 + 12*B^4*a^3*b^8*c*d^9 - 32*B^4*a^5*b^6*c*d^9 + 12*B^4*a^7*b^4*c*d^9))/(a^
10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b
^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*
b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3
*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) + ((-(4*B^2*b^7*c^2
- 8*B^2*a^2*b^5*c^2 + 4*B^2*a^4*b^3*c^2 + B^2*a^2*b^5*d^2 - 6*B^2*a^4*b^3*d
^2 + 9*B^2*a^6*b*d^2 + 16*B^2*a^3*b^4*c*d - 12*B^2*a^5*b^2*c*d - 4*B^2*a*b^
6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2
 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^
2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c
*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^
2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12
*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*((8*(52*B^3*a^3*b^10*d^11*
f^2 - 128*B^3*a^5*b^8*d^11*f^2 - 24*B^3*a^7*b^6*d^11*f^2 + 160*B^3*a^9*b^4*
d^11*f^2 + 4*B^3*a^11*b^2*d^11*f^2 + 12*B^3*b^13*c^3*d^8*f^2 + 44*B^3*a*b^1
2*c^2*d^9*f^2 - 128*B^3*a^2*b^11*c*d^10*f^2 + 48*B^3*a^4*b^9*c*d^10*f^2 + 1
76*B^3*a^6*b^7*c*d^10*f^2 - 48*B^3*a^8*b^5*c*d^10*f^2 - 48*B^3*a^10*b^3*c*d
^10*f^2 - 112*B^3*a^2*b^11*c^3*d^8*f^2 + 192*B^3*a^3*b^10*c^2*d^9*f^2 - 24*
B^3*a^4*b^9*c^3*d^8*f^2 - 72*B^3*a^5*b^8*c^2*d^9*f^2 + 80*B^3*a^6*b^7*c^3*d
^8*f^2 - 160*B^3*a^7*b^6*c^2*d^9*f^2 - 20*B^3*a^8*b^5*c^3*d^8*f^2 + 60*B^3*
a^9*b^4*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*
a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4
*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5
- 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*
f^5) - ((-(4*B^2*b^7*c^2 - 8*B^2*a^2*b^5*c^2 + 4*B^2*a^4*b^3*c^2 + B^2*a^2*
b^5*d^2 - 6*B^2*a^4*b^3*d^2 + 9*B^2*a^6*b*d^2 + 16*B^2*a^3*b^4*c*d - 12*B^2
*a^5*b^2*c*d - 4*B^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^
3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d
^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^
10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*
f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 +
12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*(
(16*(c + d*tan(e + f*x))^(1/2)*(68*B^2*a^3*b^12*d^11*f^2 + 20*B^2*a^5*b^10*
d^11*f^2 - 88*B^2*a^7*b^8*d^11*f^2 + 40*B^2*a^9*b^6*d^11*f^2 + 84*B^2*a^11*
b^4*d^11*f^2 + 4*B^2*a^13*b^2*d^11*f^2 + 36*B^2*b^15*c^3*d^8*f^2 + 36*B^2*a
*b^14*c^2*d^9*f^2 - 128*B^2*a^2*b^13*c*d^10*f^2 - 112*B^2*a^4*b^11*c*d^10*f

^2 + 128*B^2*a^6*b^9*c*d^10*f^2 + 32*B^2*a^8*b^7*c*d^10*f^2 - 128*B^2*a^10*
b^5*c*d^10*f^2 - 48*B^2*a^12*b^3*c*d^10*f^2 - 68*B^2*a^2*b^13*c^3*d^8*f^2 +
 204*B^2*a^3*b^12*c^2*d^9*f^2 - 184*B^2*a^4*b^11*c^3*d^8*f^2 + 200*B^2*a^5*
b^10*c^2*d^9*f^2 - 40*B^2*a^6*b^9*c^3*d^8*f^2 - 8*B^2*a^7*b^8*c^2*d^9*f^2 +
 20*B^2*a^8*b^7*c^3*d^8*f^2 + 20*B^2*a^9*b^6*c^2*d^9*f^2 - 20*B^2*a^10*b^5*
c^3*d^8*f^2 + 60*B^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 +
4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4
 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*
f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*
d*f^4 - 8*a^7*b^3*c*d*f^4) + ((-(4*B^2*b^7*c^2 - 8*B^2*a^2*b^5*c^2 + 4*B^2*
a^4*b^3*c^2 + B^2*a^2*b^5*d^2 - 6*B^2*a^4*b^3*d^2 + 9*B^2*a^6*b*d^2 + 16*B^
2*a^3*b^4*c*d - 12*B^2*a^5*b^2*c*d - 4*B^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*
c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b
^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^
9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f
^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 1
8*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b
^2*c^2*d*f^2))^(1/2)*((8*(32*B*a^2*b^15*d^12*f^4 + 96*B*a^4*b^13*d^12*f^4 -
 320*B*a^8*b^9*d^12*f^4 - 480*B*a^10*b^7*d^12*f^4 - 288*B*a^12*b^5*d^12*f^4
 - 64*B*a^14*b^3*d^12*f^4 + 64*B*b^17*c^2*d^10*f^4 + 48*B*b^17*c^4*d^8*f^4
- 112*B*a*b^16*c^3*d^9*f^4 - 400*B*a^3*b^14*c*d^11*f^4 - 544*B*a^5*b^12*c*d
^11*f^4 - 80*B*a^7*b^10*c*d^11*f^4 + 480*B*a^9*b^8*c*d^11*f^4 + 464*B*a^11*
b^6*c*d^11*f^4 + 160*B*a^13*b^4*c*d^11*f^4 + 16*B*a^15*b^2*c*d^11*f^4 + 368
*B*a^2*b^15*c^2*d^10*f^4 + 224*B*a^2*b^15*c^4*d^8*f^4 - 512*B*a^3*b^14*c^3*
d^9*f^4 + 832*B*a^4*b^13*c^2*d^10*f^4 + 400*B*a^4*b^13*c^4*d^8*f^4 - 880*B*
a^5*b^12*c^3*d^9*f^4 + 880*B*a^6*b^11*c^2*d^10*f^4 + 320*B*a^6*b^11*c^4*d^8
*f^4 - 640*B*a^7*b^10*c^3*d^9*f^4 + 320*B*a^8*b^9*c^2*d^10*f^4 + 80*B*a^8*b
^9*c^4*d^8*f^4 - 80*B*a^9*b^8*c^3*d^9*f^4 - 176*B*a^10*b^7*c^2*d^10*f^4 - 3
2*B*a^10*b^7*c^4*d^8*f^4 + 128*B*a^11*b^6*c^3*d^9*f^4 - 192*B*a^12*b^5*c^2*
d^10*f^4 - 16*B*a^12*b^5*c^4*d^8*f^4 + 48*B*a^13*b^4*c^3*d^9*f^4 - 48*B*a^1
4*b^3*c^2*d^10*f^4 - 96*B*a*b^16*c*d^11*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5
+ 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f
^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^
2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*
c*d*f^5 - 8*a^7*b^3*c*d*f^5) - (16*(-(4*B^2*b^7*c^2 - 8*B^2*a^2*b^5*c^2 + 4
*B^2*a^4*b^3*c^2 + B^2*a^2*b^5*d^2 - 6*B^2*a^4*b^3*d^2 + 9*B^2*a^6*b*d^2 +
16*B^2*a^3*b^4*c*d - 12*B^2*a^5*b^2*c*d - 4*B^2*a*b^6*c*d)*(a^11*d^3*f^2 -
b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 -
a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 +
 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*
d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^
2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*
a^9*b^2*c^2*d*f^2))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4
+ 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 1
60*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a

^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^1
7*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*
a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4
 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*
d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^
10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240
*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4
 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d
^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d
^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*
d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8
*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b
^2*c*d^11*f^4))/((b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^
2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^
6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8
*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^
2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f
^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2)*(a^10*d^2*f^4 + b^10*c^2*f^4
 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f
^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^
2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*
c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^
3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f
^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c
^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 +
24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2
 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2)))/(b^9*(8*a^2*
c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(
12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2)
 - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*
d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*
a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a
^10*b*c*d^2*f^2)))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*
f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*
a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b
^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*
f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3
*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2)))/(b^9*(8*a^2*c^3*f^2 + 6*a
^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^
2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9
*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*
(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2
) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f
^2) - ((-(4*B^2*b^7*c^2 - 8*B^2*a^2*b^5*c^2 + 4*B^2*a^4*b^3*c^2 + B^2*a^2*b
^5*d^2 - 6*B^2*a^4*b^3*d^2 + 9*B^2*a^6*b*d^2 + 16*B^2*a^3*b^4*c*d - 12*B^2*

a^5*b^2*c*d - 4*B^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3
*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^
3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^1
0*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f
^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 1
2*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*((
16*(c + d*tan(e + f*x))^(1/2)*(3*B^4*a^2*b^9*d^10 - 3*B^4*a^4*b^7*d^10 + 17
*B^4*a^6*b^5*d^10 - 9*B^4*a^8*b^3*d^10 + 6*B^4*b^11*c^2*d^8 - 8*B^4*a^2*b^9
*c^2*d^8 + 14*B^4*a^4*b^7*c^2*d^8 - 4*B^4*a^6*b^5*c^2*d^8 - 8*B^4*a*b^10*c*
d^9 + 12*B^4*a^3*b^8*c*d^9 - 32*B^4*a^5*b^6*c*d^9 + 12*B^4*a^7*b^4*c*d^9))/
(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^
6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a
^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*
a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((-(4*B^2*b^7*c
^2 - 8*B^2*a^2*b^5*c^2 + 4*B^2*a^4*b^3*c^2 + B^2*a^2*b^5*d^2 - 6*B^2*a^4*b^
3*d^2 + 9*B^2*a^6*b*d^2 + 16*B^2*a^3*b^4*c*d - 12*B^2*a^5*b^2*c*d - 4*B^2*a
*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*
f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3
*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*
b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2
*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 -
12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*((8*(52*B^3*a^3*b^10*d^
11*f^2 - 128*B^3*a^5*b^8*d^11*f^2 - 24*B^3*a^7*b^6*d^11*f^2 + 160*B^3*a^9*b
^4*d^11*f^2 + 4*B^3*a^11*b^2*d^11*f^2 + 12*B^3*b^13*c^3*d^8*f^2 + 44*B^3*a*
b^12*c^2*d^9*f^2 - 128*B^3*a^2*b^11*c*d^10*f^2 + 48*B^3*a^4*b^9*c*d^10*f^2
+ 176*B^3*a^6*b^7*c*d^10*f^2 - 48*B^3*a^8*b^5*c*d^10*f^2 - 48*B^3*a^10*b^3*
c*d^10*f^2 - 112*B^3*a^2*b^11*c^3*d^8*f^2 + 192*B^3*a^3*b^10*c^2*d^9*f^2 -
24*B^3*a^4*b^9*c^3*d^8*f^2 - 72*B^3*a^5*b^8*c^2*d^9*f^2 + 80*B^3*a^6*b^7*c^
3*d^8*f^2 - 160*B^3*a^7*b^6*c^2*d^9*f^2 - 20*B^3*a^8*b^5*c^3*d^8*f^2 + 60*B
^3*a^9*b^4*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 +
6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5
+ 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f
^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c
*d*f^5) + ((-(4*B^2*b^7*c^2 - 8*B^2*a^2*b^5*c^2 + 4*B^2*a^4*b^3*c^2 + B^2*a
^2*b^5*d^2 - 6*B^2*a^4*b^3*d^2 + 9*B^2*a^6*b*d^2 + 16*B^2*a^3*b^4*c*d - 12*
B^2*a^5*b^2*c*d - 4*B^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9
*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^
8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a
*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2
*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2
+ 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2
)*((16*(c + d*tan(e + f*x))^(1/2)*(68*B^2*a^3*b^12*d^11*f^2 + 20*B^2*a^5*b^
10*d^11*f^2 - 88*B^2*a^7*b^8*d^11*f^2 + 40*B^2*a^9*b^6*d^11*f^2 + 84*B^2*a^
11*b^4*d^11*f^2 + 4*B^2*a^13*b^2*d^11*f^2 + 36*B^2*b^15*c^3*d^8*f^2 + 36*B^
2*a*b^14*c^2*d^9*f^2 - 128*B^2*a^2*b^13*c*d^10*f^2 - 112*B^2*a^4*b^11*c*d^1

0*f^2 + 128*B^2*a^6*b^9*c*d^10*f^2 + 32*B^2*a^8*b^7*c*d^10*f^2 - 128*B^2*a^10*b^5*c*d^10*f^2 - 48*B^2*a^12*b^3*c*d^10*f^2 - 68*B^2*a^2*b^13*c^3*d^8*f^2 + 204*B^2*a^3*b^12*c^2*d^9*f^2 - 184*B^2*a^4*b^11*c^3*d^8*f^2 + 200*B^2*a^5*b^10*c^2*d^9*f^2 - 40*B^2*a^6*b^9*c^3*d^8*f^2 - 8*B^2*a^7*b^8*c^2*d^9*f^2 + 20*B^2*a^8*b^7*c^3*d^8*f^2 + 20*B^2*a^9*b^6*c^2*d^9*f^2 - 20*B^2*a^10*b^5*c^3*d^8*f^2 + 60*B^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((-(4*B^2*b^7*c^2 - 8*B^2*a^2*b^5*c^2 + 4*B^2*a^4*b^3*c^2 + B^2*a^2*b^5*d^2 - 6*B^2*a^4*b^3*d^2 + 9*B^2*a^6*b*d^2 + 16*B^2*a^3*b^4*c*d - 12*B^2*a^5*b^2*c*d - 4*B^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*((8*(32*B*a^2*b^15*d^12*f^4 + 96*B*a^4*b^13*d^12*f^4 - 320*B*a^8*b^9*d^12*f^4 - 480*B*a^10*b^7*d^12*f^4 - 288*B*a^12*b^5*d^12*f^4 - 64*B*a^14*b^3*d^12*f^4 + 64*B*b^17*c^2*d^10*f^4 + 48*B*b^17*c^4*d^8*f^4 - 112*B*a*b^16*c^3*d^9*f^4 - 400*B*a^3*b^14*c*d^11*f^4 - 544*B*a^5*b^12*c*d^11*f^4 - 80*B*a^7*b^10*c*d^11*f^4 + 480*B*a^9*b^8*c*d^11*f^4 + 464*B*a^11*b^6*c*d^11*f^4 + 160*B*a^13*b^4*c*d^11*f^4 + 16*B*a^15*b^2*c*d^11*f^4 + 368*B*a^2*b^15*c^2*d^10*f^4 + 224*B*a^2*b^15*c^4*d^8*f^4 - 512*B*a^3*b^14*c^3*d^9*f^4 + 832*B*a^4*b^13*c^2*d^10*f^4 + 400*B*a^4*b^13*c^4*d^8*f^4 - 880*B*a^5*b^12*c^3*d^9*f^4 + 880*B*a^6*b^11*c^2*d^10*f^4 + 320*B*a^6*b^11*c^4*d^8*f^4 - 640*B*a^7*b^10*c^3*d^9*f^4 + 320*B*a^8*b^9*c^2*d^10*f^4 + 80*B*a^8*b^9*c^4*d^8*f^4 - 80*B*a^9*b^8*c^3*d^9*f^4 - 176*B*a^10*b^7*c^2*d^10*f^4 - 32*B*a^10*b^7*c^4*d^8*f^4 + 128*B*a^11*b^6*c^3*d^9*f^4 - 192*B*a^12*b^5*c^2*d^10*f^4 - 16*B*a^12*b^5*c^4*d^8*f^4 + 48*B*a^13*b^4*c^3*d^9*f^4 - 48*B*a^14*b^3*c^2*d^10*f^4 - 96*B*a*b^16*c*d^11*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + (16*(-(4*B^2*b^7*c^2 - 8*B^2*a^2*b^5*c^2 + 4*B^2*a^4*b^3*c^2 + B^2*a^2*b^5*d^2 - 6*B^2*a^4*b^3*d^2 + 9*B^2*a^6*b*d^2 + 16*B^2*a^3*b^4*c*d - 12*B^2*a^5*b^2*c*d - 4*B^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 3

2*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*
b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 3
36*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*
f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c
^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9
*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 +
240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*
f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^
4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*
c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14
*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*
b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^1
7*b^2*c*d^11*f^4))/((b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3
*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8
*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) -
b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d
f^2) - b^6(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^
3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2)*(a^10*d^2*f^4 + b^10*c^2*f
^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^
2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2
*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b
^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) +
b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^
2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^
9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2
+ 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*
f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2)))/(b^9*(8*a
^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^
7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f
^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c
^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 +
36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 +
6*a^10*b*c*d^2*f^2)))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c
^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*
(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2)
- b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2
*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*
c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2)))/(b^9*(8*a^2*c^3*f^2 +
6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3
*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*
a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b
^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*
f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^
2*f^2)))*(-(4*B^2*b^7*c^2 - 8*B^2*a^2*b^5*c^2 + 4*B^2*a^4*b^3*c^2 + B^2*a^2
*b^5*d^2 - 6*B^2*a^4*b^3*d^2 + 9*B^2*a^6*b*d^2 + 16*B^2*a^3*b^4*c*d - 12*B^

2*a^5*b^2*c*d - 4*B^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c
^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*
d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b
^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d
*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 +
12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*
2i)/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*
d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 3
6*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f
^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a
^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10
*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2) - (atan((((((512*C^4*a^4*b^4*c^2*f^4 - 16*
C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^
6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4
*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*
C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*
b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)
*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4
+ 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^
2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(2*C^4*a^
2*b^9*d^10 - 5*C^4*a^4*b^7*d^10 + 17*C^4*a^6*b^5*d^10 - 7*C^4*a^8*b^3*d^10
+ 2*C^4*b^11*c^2*d^8 + C^4*a^10*b*d^10 - 12*C^4*a^2*b^9*c^2*d^8 + 18*C^4*a^
4*b^7*c^2*d^8 - 4*C^4*a*b^10*c*d^9 + 16*C^4*a^3*b^8*c*d^9 - 36*C^4*a^5*b^6*
c*d^9 + 8*C^4*a^7*b^4*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*
f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2
*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*
c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*
b^3*c*d*f^4) + ((((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a
^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2
*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*
a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4
*a^7*b*c*d*f^4)^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*
f^2 + 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4
+ b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^
6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))
^(1/2)*((16*(8*C^3*a^6*b^7*d^11*f^2 - 78*C^3*a^4*b^9*d^11*f^2 + 60*C^3*a^8*
b^5*d^11*f^2 - 24*C^3*a^10*b^3*d^11*f^2 + 2*C^3*a^12*b*d^11*f^2 - 32*C^3*a*
b^12*c^3*d^8*f^2 + 152*C^3*a^3*b^10*c*d^10*f^2 + 128*C^3*a^5*b^8*c*d^10*f^2
- 64*C^3*a^7*b^6*c*d^10*f^2 - 32*C^3*a^9*b^4*c*d^10*f^2 + 8*C^3*a^11*b^2*c
*d^10*f^2 - 40*C^3*a^2*b^11*c^2*d^9*f^2 + 64*C^3*a^3*b^10*c^3*d^8*f^2 - 216
*C^3*a^4*b^9*c^2*d^9*f^2 + 96*C^3*a^5*b^8*c^3*d^8*f^2 - 120*C^3*a^6*b^7*c^2
*d^9*f^2 + 56*C^3*a^8*b^5*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^
2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a
^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5
- 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^

5 - 8*a^7*b^3*c*d*f^5) - ((((((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(40*C*a^3*b^14*d^12*f^4 + 192*C*a^5*b^12*d^12*f^4 + 360*C*a^7*b^10*d^12*f^4 + 320*C*a^9*b^8*d^12*f^4 + 120*C*a^11*b^6*d^12*f^4 - 8*C*a^15*b^2*d^12*f^4 + 8*C*b^17*c^3*d^9*f^4 + 40*C*a*b^16*c^2*d^10*f^4 + 32*C*a*b^16*c^4*d^8*f^4 - 88*C*a^2*b^15*c*d^11*f^4 - 448*C*a^4*b^13*c*d^11*f^4 - 920*C*a^6*b^11*c*d^11*f^4 - 960*C*a^8*b^9*c*d^11*f^4 - 520*C*a^10*b^7*c*d^11*f^4 - 128*C*a^12*b^5*c*d^11*f^4 - 8*C*a^14*b^3*c*d^11*f^4 - 32*C*a^2*b^15*c^3*d^9*f^4 + 256*C*a^3*b^14*c^2*d^10*f^4 + 160*C*a^3*b^14*c^4*d^8*f^4 - 280*C*a^4*b^13*c^3*d^9*f^4 + 680*C*a^5*b^12*c^2*d^10*f^4 + 320*C*a^5*b^12*c^4*d^8*f^4 - 640*C*a^6*b^11*c^3*d^9*f^4 + 960*C*a^7*b^10*c^2*d^10*f^4 + 320*C*a^7*b^10*c^4*d^8*f^4 - 680*C*a^8*b^9*c^3*d^9*f^4 + 760*C*a^9*b^8*c^2*d^10*f^4 + 160*C*a^9*b^8*c^4*d^8*f^4 - 352*C*a^10*b^7*c^3*d^9*f^4 + 320*C*a^11*b^6*c^2*d^10*f^4 + 32*C*a^11*b^6*c^4*d^8*f^4 - 72*C*a^12*b^5*c^3*d^9*f^4 + 56*C*a^13*b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - (4*(((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 +

16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)) + (16*(c + d*tan(e + f*x))^(1/2)*(20*C^2*a^5*b^10*d^11*f^2 - 60*C^2*a^3*b^12*d^11*f^2 + 168*C^2*a^7*b^8*d^11*f^2 + 40*C^2*a^9*b^6*d^11*f^2 - 44*C^2*a^11*b^4*d^11*f^2 + 4*C^2*a^13*b^2*d^11*f^2 - 20*C^2*b^15*c^3*d^8*f^2 - 4*C^2*a^14*b*c*d^10*f^2 - 20*C^2*a*b^14*c^2*d^9*f^2 + 100*C^2*a^2*b^13*c*d^10*f^2 - 300*C^2*a^6*b^9*c*d^10*f^2 - 160*C^2*a^8*b^7*c*d^10*f^2 + 76*C^2*a^10*b^5*c*d^10*f^2 + 32*C^2*a^12*b^3*c*d^10*f^2 + 116*C^2*a^2*b^13*c^3*d^8*f^2 - 124*C^2*a^3*b^12*c^2*d^9*f^2 + 216*C^2*a^4*b^11*c^3*d^8*f^2 - 40*C^2*a^5*b^10*c^2*d^9*f^2 + 8*C^2*a^6*b^9*c^3*d^8*f^2 + 168*C^2*a^7*b^8*c^2*d^9*f^2 - 68*C^2*a^8*b^7*c^3*d^8*f^2 + 60*C^2*a^9*b^6*c^2*d^9*f^2 + 4*C^2*a^10*b^5*c^3*d^8*f^2 - 44*C^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))*(((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)))*1i)/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)) + ((((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f

^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*
a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4
))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(2*C^4*a^2*b^9*d^10 - 5*C^4*a^4*b^
7*d^10 + 17*C^4*a^6*b^5*d^10 - 7*C^4*a^8*b^3*d^10 + 2*C^4*b^11*c^2*d^8 + C^
4*a^10*b*d^10 - 12*C^4*a^2*b^9*c^2*d^8 + 18*C^4*a^4*b^7*c^2*d^8 - 4*C^4*a*b
^10*c*d^9 + 16*C^4*a^3*b^8*c*d^9 - 36*C^4*a^5*b^6*c*d^9 + 8*C^4*a^7*b^4*c*d
^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 +
4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4
+ 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4
- 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((((512*C^
4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a
^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^
4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4
*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) - 4
*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2
+ 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*
f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6
*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(8*C^3*a^6*b^
7*d^11*f^2 - 78*C^3*a^4*b^9*d^11*f^2 + 60*C^3*a^8*b^5*d^11*f^2 - 24*C^3*a^1
0*b^3*d^11*f^2 + 2*C^3*a^12*b*d^11*f^2 - 32*C^3*a*b^12*c^3*d^8*f^2 + 152*C^
3*a^3*b^10*c*d^10*f^2 + 128*C^3*a^5*b^8*c*d^10*f^2 - 64*C^3*a^7*b^6*c*d^10*
f^2 - 32*C^3*a^9*b^4*c*d^10*f^2 + 8*C^3*a^11*b^2*c*d^10*f^2 - 40*C^3*a^2*b^
11*c^2*d^9*f^2 + 64*C^3*a^3*b^10*c^3*d^8*f^2 - 216*C^3*a^4*b^9*c^2*d^9*f^2
+ 96*C^3*a^5*b^8*c^3*d^8*f^2 - 120*C^3*a^6*b^7*c^2*d^9*f^2 + 56*C^3*a^8*b^5
*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6
*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^
6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9
*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) -
((((((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^
4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4
- 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f
^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^
4)^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*
a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^
4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4
+ 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(
40*C*a^3*b^14*d^12*f^4 + 192*C*a^5*b^12*d^12*f^4 + 360*C*a^7*b^10*d^12*f^4
+ 320*C*a^9*b^8*d^12*f^4 + 120*C*a^11*b^6*d^12*f^4 - 8*C*a^15*b^2*d^12*f^4
+ 8*C*b^17*c^3*d^9*f^4 + 40*C*a*b^16*c^2*d^10*f^4 + 32*C*a*b^16*c^4*d^8*f^4
- 88*C*a^2*b^15*c*d^11*f^4 - 448*C*a^4*b^13*c*d^11*f^4 - 920*C*a^6*b^11*c*
d^11*f^4 - 960*C*a^8*b^9*c*d^11*f^4 - 520*C*a^10*b^7*c*d^11*f^4 - 128*C*a^1
2*b^5*c*d^11*f^4 - 8*C*a^14*b^3*c*d^11*f^4 - 32*C*a^2*b^15*c^3*d^9*f^4 + 25
6*C*a^3*b^14*c^2*d^10*f^4 + 160*C*a^3*b^14*c^4*d^8*f^4 - 280*C*a^4*b^13*c^3
*d^9*f^4 + 680*C*a^5*b^12*c^2*d^10*f^4 + 320*C*a^5*b^12*c^4*d^8*f^4 - 640*C
*a^6*b^11*c^3*d^9*f^4 + 960*C*a^7*b^10*c^2*d^10*f^4 + 320*C*a^7*b^10*c^4*d^

8*f^4 - 680*C*a^8*b^9*c^3*d^9*f^4 + 760*C*a^9*b^8*c^2*d^10*f^4 + 160*C*a^9*
b^8*c^4*d^8*f^4 - 352*C*a^10*b^7*c^3*d^9*f^4 + 320*C*a^11*b^6*c^2*d^10*f^4
+ 32*C*a^11*b^6*c^4*d^8*f^4 - 72*C*a^12*b^5*c^3*d^9*f^4 + 56*C*a^13*b^4*c^2
*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^
2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d
^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*
c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + (4*
(((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 -
16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 6
08*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4
+ 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^
(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3
*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 +
b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 +
4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*ta
n(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b
^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7
*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10
*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d
^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^1
5*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*
a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4
+ 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*
d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^
7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a
^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 -
48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 3
04*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 +
880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f
^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8
*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^
4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d
^2*f^4)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^
4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f
^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*
f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^
8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6
*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^
4 + 4*a^6*b^2*d^2*f^4)) - (16*(c + d*tan(e + f*x))^(1/2)*(20*C^2*a^5*b^10*d
^11*f^2 - 60*C^2*a^3*b^12*d^11*f^2 + 168*C^2*a^7*b^8*d^11*f^2 + 40*C^2*a^9*
b^6*d^11*f^2 - 44*C^2*a^11*b^4*d^11*f^2 + 4*C^2*a^13*b^2*d^11*f^2 - 20*C^2*
b^15*c^3*d^8*f^2 - 4*C^2*a^14*b*c*d^10*f^2 - 20*C^2*a*b^14*c^2*d^9*f^2 + 10
0*C^2*a^2*b^13*c*d^10*f^2 - 300*C^2*a^6*b^9*c*d^10*f^2 - 160*C^2*a^8*b^7*c*
d^10*f^2 + 76*C^2*a^10*b^5*c*d^10*f^2 + 32*C^2*a^12*b^3*c*d^10*f^2 + 116*C^
2*a^2*b^13*c^3*d^8*f^2 - 124*C^2*a^3*b^12*c^2*d^9*f^2 + 216*C^2*a^4*b^11*c^

3*d^8*f^2 - 40*C^2*a^5*b^10*c^2*d^9*f^2 + 8*C^2*a^6*b^9*c^3*d^8*f^2 + 168*C
^2*a^7*b^8*c^2*d^9*f^2 - 68*C^2*a^8*b^7*c^3*d^8*f^2 + 60*C^2*a^9*b^6*c^2*d^
9*f^2 + 4*C^2*a^10*b^5*c^3*d^8*f^2 - 44*C^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^
2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^
2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d
^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*
c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))*(((512*C^4*a^4*b^4*c^2*f
^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 25
6*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 +
 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^
4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) - 4*C^2*a^4*c*f^2
- 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^
2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6
*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^
4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 +
b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b
^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/
(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f
^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*
d^2*f^4 + 4*a^6*b^2*d^2*f^4)))*1i)/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*
f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f
^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)))/((32*(3*C
^5*a^3*b^6*d^10 - C^5*a^5*b^4*d^10 + 4*C^5*a*b^8*c^2*d^8 - 7*C^5*a^2*b^7*c*
d^9 + C^5*a^4*b^5*c*d^9))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5
+ 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5
 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*
f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*
c*d*f^5) - ((((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b
^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6
*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*
b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7
*b*c*d*f^4)^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2
+ 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b
^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^
2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/
2)*((16*(c + d*tan(e + f*x))^(1/2)*(2*C^4*a^2*b^9*d^10 - 5*C^4*a^4*b^7*d^10
 + 17*C^4*a^6*b^5*d^10 - 7*C^4*a^8*b^3*d^10 + 2*C^4*b^11*c^2*d^8 + C^4*a^10
*b*d^10 - 12*C^4*a^2*b^9*c^2*d^8 + 18*C^4*a^4*b^7*c^2*d^8 - 4*C^4*a*b^10*c*
d^9 + 16*C^4*a^3*b^8*c*d^9 - 36*C^4*a^5*b^6*c*d^9 + 8*C^4*a^7*b^4*c*d^9))/(
a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6
*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^
6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a
^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) + ((((512*C^4*a^4*
b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2
*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*

d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b
^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) - 4*C^2*a
^4*c*f^2 - 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 + 24*C
^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 +
4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f
^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(8*C^3*a^6*b^7*d^11
*f^2 - 78*C^3*a^4*b^9*d^11*f^2 + 60*C^3*a^8*b^5*d^11*f^2 - 24*C^3*a^10*b^3*
d^11*f^2 + 2*C^3*a^12*b*d^11*f^2 - 32*C^3*a*b^12*c^3*d^8*f^2 + 152*C^3*a^3*
b^10*c*d^10*f^2 + 128*C^3*a^5*b^8*c*d^10*f^2 - 64*C^3*a^7*b^6*c*d^10*f^2 -
32*C^3*a^9*b^4*c*d^10*f^2 + 8*C^3*a^11*b^2*c*d^10*f^2 - 40*C^3*a^2*b^11*c^2
*d^9*f^2 + 64*C^3*a^3*b^10*c^3*d^8*f^2 - 216*C^3*a^4*b^9*c^2*d^9*f^2 + 96*C
^3*a^5*b^8*c^3*d^8*f^2 - 120*C^3*a^6*b^7*c^2*d^9*f^2 + 56*C^3*a^8*b^5*c^2*d
^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f
^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*
f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d
*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - ((((((
512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16
*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*
C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 8
96*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/
2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*
d*f^2 + 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^
8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a
^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(40*C*a
^3*b^14*d^12*f^4 + 192*C*a^5*b^12*d^12*f^4 + 360*C*a^7*b^10*d^12*f^4 + 320*
C*a^9*b^8*d^12*f^4 + 120*C*a^11*b^6*d^12*f^4 - 8*C*a^15*b^2*d^12*f^4 + 8*C*
b^17*c^3*d^9*f^4 + 40*C*a*b^16*c^2*d^10*f^4 + 32*C*a*b^16*c^4*d^8*f^4 - 88*
C*a^2*b^15*c*d^11*f^4 - 448*C*a^4*b^13*c*d^11*f^4 - 920*C*a^6*b^11*c*d^11*f
^4 - 960*C*a^8*b^9*c*d^11*f^4 - 520*C*a^10*b^7*c*d^11*f^4 - 128*C*a^12*b^5*
c*d^11*f^4 - 8*C*a^14*b^3*c*d^11*f^4 - 32*C*a^2*b^15*c^3*d^9*f^4 + 256*C*a^
3*b^14*c^2*d^10*f^4 + 160*C*a^3*b^14*c^4*d^8*f^4 - 280*C*a^4*b^13*c^3*d^9*f
^4 + 680*C*a^5*b^12*c^2*d^10*f^4 + 320*C*a^5*b^12*c^4*d^8*f^4 - 640*C*a^6*b
^11*c^3*d^9*f^4 + 960*C*a^7*b^10*c^2*d^10*f^4 + 320*C*a^7*b^10*c^4*d^8*f^4
- 680*C*a^8*b^9*c^3*d^9*f^4 + 760*C*a^9*b^8*c^2*d^10*f^4 + 160*C*a^9*b^8*c^
4*d^8*f^4 - 352*C*a^10*b^7*c^3*d^9*f^4 + 320*C*a^11*b^6*c^2*d^10*f^4 + 32*C
*a^11*b^6*c^4*d^8*f^4 - 72*C*a^12*b^5*c^3*d^9*f^4 + 56*C*a^13*b^4*c^2*d^10*
f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5
+ 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5
+ 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^
5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - (4*(((512
*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^
4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4
*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*
C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2)
- 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f

^2 + 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d ^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2* b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^ 12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12* f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4* d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^ 13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400 *a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f ^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2* d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^ 5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^1 6*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3 *b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a ^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 1 76*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8*d^2*f ^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4* a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4* a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6 *a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^8*c^2* f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b ^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4* a^6*b^2*d^2*f^4)) + (16*(c + d*tan(e + f*x))^(1/2)*(20*C^2*a^5*b^10*d^11*f^ 2 - 60*C^2*a^3*b^12*d^11*f^2 + 168*C^2*a^7*b^8*d^11*f^2 + 40*C^2*a^9*b^6*d^ 11*f^2 - 44*C^2*a^11*b^4*d^11*f^2 + 4*C^2*a^13*b^2*d^11*f^2 - 20*C^2*b^15*c ^3*d^8*f^2 - 4*C^2*a^14*b*c*d^10*f^2 - 20*C^2*a*b^14*c^2*d^9*f^2 + 100*C^2* a^2*b^13*c*d^10*f^2 - 300*C^2*a^6*b^9*c*d^10*f^2 - 160*C^2*a^8*b^7*c*d^10*f ^2 + 76*C^2*a^10*b^5*c*d^10*f^2 + 32*C^2*a^12*b^3*c*d^10*f^2 + 116*C^2*a^2* b^13*c^3*d^8*f^2 - 124*C^2*a^3*b^12*c^2*d^9*f^2 + 216*C^2*a^4*b^11*c^3*d^8* f^2 - 40*C^2*a^5*b^10*c^2*d^9*f^2 + 8*C^2*a^6*b^9*c^3*d^8*f^2 + 168*C^2*a^7 *b^8*c^2*d^9*f^2 - 68*C^2*a^8*b^7*c^3*d^8*f^2 + 60*C^2*a^9*b^6*c^2*d^9*f^2 + 4*C^2*a^10*b^5*c^3*d^8*f^2 - 44*C^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^ 4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))*(((512*C^4*a^4*b^4*c^2*f^4 - 1 6*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4* a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C ^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 12 8*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^ 2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^

2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f
^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*
d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^
2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2
*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^
8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6
*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^
4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8
*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^
2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)) + ((((512*C^4*a^4*b
^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*
f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d
^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^
3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) - 4*C^2*a^
4*c*f^2 - 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 + 24*C^
2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4
*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^
4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))
^(1/2)*(2*C^4*a^2*b^9*d^10 - 5*C^4*a^4*b^7*d^10 + 17*C^4*a^6*b^5*d^10 - 7*C
^4*a^8*b^3*d^10 + 2*C^4*b^11*c^2*d^8 + C^4*a^10*b*d^10 - 12*C^4*a^2*b^9*c^2
*d^8 + 18*C^4*a^4*b^7*c^2*d^8 - 4*C^4*a*b^10*c*d^9 + 16*C^4*a^3*b^8*c*d^9 -
36*C^4*a^5*b^6*c*d^9 + 8*C^4*a^7*b^4*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4
+ 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f
^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^
2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*
c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2
*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f
^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^
2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c
*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 -
16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f
^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^
4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a
^6*b^2*d^2*f^4))^(1/2)*((16*(8*C^3*a^6*b^7*d^11*f^2 - 78*C^3*a^4*b^9*d^11*f
^2 + 60*C^3*a^8*b^5*d^11*f^2 - 24*C^3*a^10*b^3*d^11*f^2 + 2*C^3*a^12*b*d^11
*f^2 - 32*C^3*a*b^12*c^3*d^8*f^2 + 152*C^3*a^3*b^10*c*d^10*f^2 + 128*C^3*a^
5*b^8*c*d^10*f^2 - 64*C^3*a^7*b^6*c*d^10*f^2 - 32*C^3*a^9*b^4*c*d^10*f^2 +
8*C^3*a^11*b^2*c*d^10*f^2 - 40*C^3*a^2*b^11*c^2*d^9*f^2 + 64*C^3*a^3*b^10*c
^3*d^8*f^2 - 216*C^3*a^4*b^9*c^2*d^9*f^2 + 96*C^3*a^5*b^8*c^3*d^8*f^2 - 120
*C^3*a^6*b^7*c^2*d^9*f^2 + 56*C^3*a^8*b^5*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^1
0*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8
*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*
a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 1
2*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - ((((((512*C^4*a^4*b^4*c^2*f^4 - 16
*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a

^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(40*C*a^3*b^14*d^12*f^4 + 192*C*a^5*b^12*d^12*f^4 + 360*C*a^7*b^10*d^12*f^4 + 320*C*a^9*b^8*d^12*f^4 + 120*C*a^11*b^6*d^12*f^4 - 8*C*a^15*b^2*d^12*f^4 + 8*C*b^17*c^3*d^9*f^4 + 40*C*a*b^16*c^2*d^10*f^4 + 32*C*a*b^16*c^4*d^8*f^4 - 88*C*a^2*b^15*c*d^11*f^4 - 448*C*a^4*b^13*c*d^11*f^4 - 920*C*a^6*b^11*c*d^11*f^4 - 960*C*a^8*b^9*c*d^11*f^4 - 520*C*a^10*b^7*c*d^11*f^4 - 128*C*a^12*b^5*c*d^11*f^4 - 8*C*a^14*b^3*c*d^11*f^4 - 32*C*a^2*b^15*c^3*d^9*f^4 + 256*C*a^3*b^14*c^2*d^10*f^4 + 160*C*a^3*b^14*c^4*d^8*f^4 - 280*C*a^4*b^13*c^3*d^9*f^4 + 680*C*a^5*b^12*c^2*d^10*f^4 + 320*C*a^5*b^12*c^4*d^8*f^4 - 640*C*a^6*b^11*c^3*d^9*f^4 + 960*C*a^7*b^10*c^2*d^10*f^4 + 320*C*a^7*b^10*c^4*d^8*f^4 - 680*C*a^8*b^9*c^3*d^9*f^4 + 760*C*a^9*b^8*c^2*d^10*f^4 + 160*C*a^9*b^8*c^4*d^8*f^4 - 352*C*a^10*b^7*c^3*d^9*f^4 + 320*C*a^11*b^6*c^2*d^10*f^4 + 32*C*a^11*b^6*c^4*d^8*f^4 - 72*C*a^12*b^5*c^3*d^9*f^4 + 56*C*a^13*b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + (4*(((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*

a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f
^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*
d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)*(a^10*d^2*f^4 + b^10*c^2*f
^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^
2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2
*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b
^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*
f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f
^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)) - (16*(c +
d*tan(e + f*x))^(1/2)*(20*C^2*a^5*b^10*d^11*f^2 - 60*C^2*a^3*b^12*d^11*f^2
+ 168*C^2*a^7*b^8*d^11*f^2 + 40*C^2*a^9*b^6*d^11*f^2 - 44*C^2*a^11*b^4*d^1
1*f^2 + 4*C^2*a^13*b^2*d^11*f^2 - 20*C^2*b^15*c^3*d^8*f^2 - 4*C^2*a^14*b*c*
d^10*f^2 - 20*C^2*a*b^14*c^2*d^9*f^2 + 100*C^2*a^2*b^13*c*d^10*f^2 - 300*C^
2*a^6*b^9*c*d^10*f^2 - 160*C^2*a^8*b^7*c*d^10*f^2 + 76*C^2*a^10*b^5*c*d^10*
f^2 + 32*C^2*a^12*b^3*c*d^10*f^2 + 116*C^2*a^2*b^13*c^3*d^8*f^2 - 124*C^2*a
^3*b^12*c^2*d^9*f^2 + 216*C^2*a^4*b^11*c^3*d^8*f^2 - 40*C^2*a^5*b^10*c^2*d^
9*f^2 + 8*C^2*a^6*b^9*c^3*d^8*f^2 + 168*C^2*a^7*b^8*c^2*d^9*f^2 - 68*C^2*a^
8*b^7*c^3*d^8*f^2 + 60*C^2*a^9*b^6*c^2*d^9*f^2 + 4*C^2*a^10*b^5*c^3*d^8*f^2
- 44*C^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c
^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*
d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b
^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a
^7*b^3*c*d*f^4))*(((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*
a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^
2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4
*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^
4*a^7*b*c*d*f^4)^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d
*f^2 + 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^
4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a
^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)
)^(1/2))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*
b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6
*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^
8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2
*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4
*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4
+ 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^
2*f^4 + 4*a^6*b^2*d^2*f^4))))*(((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f
^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4
+ 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*
f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d
*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 - 1
6*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4
+ a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*
c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6

*b^2*d^2*f^4))^(1/2)*1i)/(2*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)) - (atan((((-((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(2*C^4*a^2*b^9*d^10 - 5*C^4*a^4*b^7*d^10 + 17*C^4*a^6*b^5*d^10 - 7*C^4*a^8*b^3*d^10 + 2*C^4*b^11*c^2*d^8 + C^4*a^10*b*d^10 - 12*C^4*a^2*b^9*c^2*d^8 + 18*C^4*a^4*b^7*c^2*d^8 - 4*C^4*a*b^10*c*d^9 + 16*C^4*a^3*b^8*c*d^9 - 36*C^4*a^5*b^6*c*d^9 + 8*C^4*a^7*b^4*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) + ((-((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(8*C^3*a^6*b^7*d^11*f^2 - 78*C^3*a^4*b^9*d^11*f^2 + 60*C^3*a^8*b^5*d^11*f^2 - 24*C^3*a^10*b^3*d^11*f^2 + 2*C^3*a^12*b*d^11*f^2 - 32*C^3*a*b^12*c^3*d^8*f^2 + 152*C^3*a^3*b^10*c*d^10*f^2 + 128*C^3*a^5*b^8*c*d^10*f^2 - 64*C^3*a^7*b^6*c*d^10*f^2 - 32*C^3*a^9*b^4*c*d^10*f^2 + 8*C^3*a^11*b^2*c*d^10*f^2 - 40*C^3*a^2*b^11*c^2*d^9*f^2 + 64*C^3*a^3*b^10*c^3*d^8*f^2 - 216*C^3*a^4*b^9*c^2*d^9*f^2 + 96*C^3*a^5*b^8*c^3*d^8*f^2 - 120*C^3*a^6*b^7*c^2*d^9*f^2 + 56*C^3*a^8*b^5*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - ((((-((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(40*C*a^3*b^14*d^12*f^4 +

192*C*a^5*b^12*d^12*f^4 + 360*C*a^7*b^10*d^12*f^4 + 320*C*a^9*b^8*d^12*f^4 + 120*C*a^11*b^6*d^12*f^4 - 8*C*a^15*b^2*d^12*f^4 + 8*C*b^17*c^3*d^9*f^4 + 40*C*a*b^16*c^2*d^10*f^4 + 32*C*a*b^16*c^4*d^8*f^4 - 88*C*a^2*b^15*c*d^11*f^4 - 448*C*a^4*b^13*c*d^11*f^4 - 920*C*a^6*b^11*c*d^11*f^4 - 960*C*a^8*b^9*c*d^11*f^4 - 520*C*a^10*b^7*c*d^11*f^4 - 128*C*a^12*b^5*c*d^11*f^4 - 8*C*a^14*b^3*c*d^11*f^4 - 32*C*a^2*b^15*c^3*d^9*f^4 + 256*C*a^3*b^14*c^2*d^10*f^4 + 160*C*a^3*b^14*c^4*d^8*f^4 - 280*C*a^4*b^13*c^3*d^9*f^4 + 680*C*a^5*b^12*c^2*d^10*f^4 + 320*C*a^5*b^12*c^4*d^8*f^4 - 640*C*a^6*b^11*c^3*d^9*f^4 + 960*C*a^7*b^10*c^2*d^10*f^4 + 320*C*a^7*b^10*c^4*d^8*f^4 - 680*C*a^8*b^9*c^3*d^9*f^4 + 760*C*a^9*b^8*c^2*d^10*f^4 + 160*C*a^9*b^8*c^4*d^8*f^4 - 352*C*a^10*b^7*c^3*d^9*f^4 + 320*C*a^11*b^6*c^2*d^10*f^4 + 32*C*a^11*b^6*c^4*d^8*f^4 - 72*C*a^12*b^5*c^3*d^9*f^4 + 56*C*a^13*b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - (4*(-((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4

+ b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6
*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))
+ (16*(c + d*tan(e + f*x))^(1/2)*(20*C^2*a^5*b^10*d^11*f^2 - 60*C^2*a^3*b^1
2*d^11*f^2 + 168*C^2*a^7*b^8*d^11*f^2 + 40*C^2*a^9*b^6*d^11*f^2 - 44*C^2*a^
11*b^4*d^11*f^2 + 4*C^2*a^13*b^2*d^11*f^2 - 20*C^2*b^15*c^3*d^8*f^2 - 4*C^2
*a^14*b*c*d^10*f^2 - 20*C^2*a*b^14*c^2*d^9*f^2 + 100*C^2*a^2*b^13*c*d^10*f^
2 - 300*C^2*a^6*b^9*c*d^10*f^2 - 160*C^2*a^8*b^7*c*d^10*f^2 + 76*C^2*a^10*b
^5*c*d^10*f^2 + 32*C^2*a^12*b^3*c*d^10*f^2 + 116*C^2*a^2*b^13*c^3*d^8*f^2 -
124*C^2*a^3*b^12*c^2*d^9*f^2 + 216*C^2*a^4*b^11*c^3*d^8*f^2 - 40*C^2*a^5*b
^10*c^2*d^9*f^2 + 8*C^2*a^6*b^9*c^3*d^8*f^2 + 168*C^2*a^7*b^8*c^2*d^9*f^2 -
68*C^2*a^8*b^7*c^3*d^8*f^2 + 60*C^2*a^9*b^6*c^2*d^9*f^2 + 4*C^2*a^10*b^5*c
^3*d^8*f^2 - 44*C^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4
*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4
+ a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f
^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d
*f^4 - 8*a^7*b^3*c*d*f^4))*(-((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4
- 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 +
192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^
4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f
^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 + 16*
C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 +
a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^
2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b
^2*d^2*f^4))^(1/2))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f
^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*
d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d
^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4
+ 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2
*f^4)))*1i)/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a
^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4
+ 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)) + ((-((512*C^4*a^4*b^4*c^2*f^4 -
16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4
*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*
C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 1
28*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C
^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f
^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*
f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4
*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(2*C^4
*a^2*b^9*d^10 - 5*C^4*a^4*b^7*d^10 + 17*C^4*a^6*b^5*d^10 - 7*C^4*a^8*b^3*d^
10 + 2*C^4*b^11*c^2*d^8 + C^4*a^10*b*d^10 - 12*C^4*a^2*b^9*c^2*d^8 + 18*C^4
*a^4*b^7*c^2*d^8 - 4*C^4*a*b^10*c*d^9 + 16*C^4*a^3*b^8*c*d^9 - 36*C^4*a^5*b
^6*c*d^9 + 8*C^4*a^7*b^4*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c
^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*
d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b

^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a
^7*b^3*c*d*f^4) - ((-((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C
^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4
*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*
C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128
*C^4*a^7*b*c*d*f^4)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 + 16*C^2*a*b^
3*d*f^2 - 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2
*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 +
4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f
^4))^(1/2)*((16*(8*C^3*a^6*b^7*d^11*f^2 - 78*C^3*a^4*b^9*d^11*f^2 + 60*C^3*
a^8*b^5*d^11*f^2 - 24*C^3*a^10*b^3*d^11*f^2 + 2*C^3*a^12*b*d^11*f^2 - 32*C^
3*a*b^12*c^3*d^8*f^2 + 152*C^3*a^3*b^10*c*d^10*f^2 + 128*C^3*a^5*b^8*c*d^10
*f^2 - 64*C^3*a^7*b^6*c*d^10*f^2 - 32*C^3*a^9*b^4*c*d^10*f^2 + 8*C^3*a^11*b
^2*c*d^10*f^2 - 40*C^3*a^2*b^11*c^2*d^9*f^2 + 64*C^3*a^3*b^10*c^3*d^8*f^2 -
 216*C^3*a^4*b^9*c^2*d^9*f^2 + 96*C^3*a^5*b^8*c^3*d^8*f^2 - 120*C^3*a^6*b^7
*c^2*d^9*f^2 + 56*C^3*a^8*b^5*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 +
4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5
 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*
f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*
d*f^5 - 8*a^7*b^3*c*d*f^5) - ((((-((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^
2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*
f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d
^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*
c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2
 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*
f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b
^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*
a^6*b^2*d^2*f^4))^(1/2)*((16*(40*C*a^3*b^14*d^12*f^4 + 192*C*a^5*b^12*d^12*
f^4 + 360*C*a^7*b^10*d^12*f^4 + 320*C*a^9*b^8*d^12*f^4 + 120*C*a^11*b^6*d^1
2*f^4 - 8*C*a^15*b^2*d^12*f^4 + 8*C*b^17*c^3*d^9*f^4 + 40*C*a*b^16*c^2*d^10
*f^4 + 32*C*a*b^16*c^4*d^8*f^4 - 88*C*a^2*b^15*c*d^11*f^4 - 448*C*a^4*b^13*
c*d^11*f^4 - 920*C*a^6*b^11*c*d^11*f^4 - 960*C*a^8*b^9*c*d^11*f^4 - 520*C*a
^10*b^7*c*d^11*f^4 - 128*C*a^12*b^5*c*d^11*f^4 - 8*C*a^14*b^3*c*d^11*f^4 -
32*C*a^2*b^15*c^3*d^9*f^4 + 256*C*a^3*b^14*c^2*d^10*f^4 + 160*C*a^3*b^14*c^
4*d^8*f^4 - 280*C*a^4*b^13*c^3*d^9*f^4 + 680*C*a^5*b^12*c^2*d^10*f^4 + 320*
C*a^5*b^12*c^4*d^8*f^4 - 640*C*a^6*b^11*c^3*d^9*f^4 + 960*C*a^7*b^10*c^2*d^
10*f^4 + 320*C*a^7*b^10*c^4*d^8*f^4 - 680*C*a^8*b^9*c^3*d^9*f^4 + 760*C*a^9
*b^8*c^2*d^10*f^4 + 160*C*a^9*b^8*c^4*d^8*f^4 - 352*C*a^10*b^7*c^3*d^9*f^4
 + 320*C*a^11*b^6*c^2*d^10*f^4 + 32*C*a^11*b^6*c^4*d^8*f^4 - 72*C*a^12*b^5*c
^3*d^9*f^4 + 56*C*a^13*b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*
a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 +
 a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^
5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*
f^5 - 8*a^7*b^3*c*d*f^5) + (4*(-((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*
f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^

4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2
*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*
d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 +
16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^
4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4
*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^
6*b^2*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 16
0*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a
^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*
b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^
2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*
b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 1
12*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*
f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c
^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^1
1*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 2
08*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f
^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*
f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11
*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d
^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c
*d^11*f^4))/((a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2
*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 +
6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2
*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^
2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 -
 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4
 - 8*a^7*b^3*c*d*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*
d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2
*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)) - (16*(c + d*tan(e +
 f*x))^(1/2)*(20*C^2*a^5*b^10*d^11*f^2 - 60*C^2*a^3*b^12*d^11*f^2 + 168*C^2
*a^7*b^8*d^11*f^2 + 40*C^2*a^9*b^6*d^11*f^2 - 44*C^2*a^11*b^4*d^11*f^2 + 4*
C^2*a^13*b^2*d^11*f^2 - 20*C^2*b^15*c^3*d^8*f^2 - 4*C^2*a^14*b*c*d^10*f^2 -
 20*C^2*a*b^14*c^2*d^9*f^2 + 100*C^2*a^2*b^13*c*d^10*f^2 - 300*C^2*a^6*b^9*
c*d^10*f^2 - 160*C^2*a^8*b^7*c*d^10*f^2 + 76*C^2*a^10*b^5*c*d^10*f^2 + 32*C
^2*a^12*b^3*c*d^10*f^2 + 116*C^2*a^2*b^13*c^3*d^8*f^2 - 124*C^2*a^3*b^12*c^
2*d^9*f^2 + 216*C^2*a^4*b^11*c^3*d^8*f^2 - 40*C^2*a^5*b^10*c^2*d^9*f^2 + 8*
C^2*a^6*b^9*c^3*d^8*f^2 + 168*C^2*a^7*b^8*c^2*d^9*f^2 - 68*C^2*a^8*b^7*c^3*
d^8*f^2 + 60*C^2*a^9*b^6*c^2*d^9*f^2 + 4*C^2*a^10*b^5*c^3*d^8*f^2 - 44*C^2*
a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6
*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 +
4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4
 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d
f^4))(-((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c
^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2

*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*
c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c
*d*f^4)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16
*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c
^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^
2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2))/
(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f
^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*
d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4
 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4
+ 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)))*1i)/(4*(a^8*
c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a
^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4
+ 4*a^6*b^2*d^2*f^4)))/((32*(3*C^5*a^3*b^6*d^10 - C^5*a^5*b^4*d^10 + 4*C^5*
a*b^8*c^2*d^8 - 7*C^5*a^2*b^7*c*d^9 + C^5*a^4*b^5*c*d^9))/(a^10*d^2*f^5 + b
^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a
^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 +
4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 -
 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - ((-((512*C^4*a^4*b^4*c^2*f^4 - 1
6*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*
a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C
^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 12
8*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^
2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^
2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f
^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*
d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(2*C^4*
a^2*b^9*d^10 - 5*C^4*a^4*b^7*d^10 + 17*C^4*a^6*b^5*d^10 - 7*C^4*a^8*b^3*d^1
0 + 2*C^4*b^11*c^2*d^8 + C^4*a^10*b*d^10 - 12*C^4*a^2*b^9*c^2*d^8 + 18*C^4*
a^4*b^7*c^2*d^8 - 4*C^4*a*b^10*c*d^9 + 16*C^4*a^3*b^8*c*d^9 - 36*C^4*a^5*b^
6*c*d^9 + 8*C^4*a^7*b^4*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^
2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d
^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^
9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^
7*b^3*c*d*f^4) + ((-((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^
4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*
a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C
^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*
C^4*a^7*b*c*d*f^4)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3
*d*f^2 - 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*
f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4
*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^
4))^(1/2)*((16*(8*C^3*a^6*b^7*d^11*f^2 - 78*C^3*a^4*b^9*d^11*f^2 + 60*C^3*a
^8*b^5*d^11*f^2 - 24*C^3*a^10*b^3*d^11*f^2 + 2*C^3*a^12*b*d^11*f^2 - 32*C^3
*a*b^12*c^3*d^8*f^2 + 152*C^3*a^3*b^10*c*d^10*f^2 + 128*C^3*a^5*b^8*c*d^10*

f^2 - 64*C^3*a^7*b^6*c*d^10*f^2 - 32*C^3*a^9*b^4*c*d^10*f^2 + 8*C^3*a^11*b^2*c*d^10*f^2 - 40*C^3*a^2*b^11*c^2*d^9*f^2 + 64*C^3*a^3*b^10*c^3*d^8*f^2 - 216*C^3*a^4*b^9*c^2*d^9*f^2 + 96*C^3*a^5*b^8*c^3*d^8*f^2 - 120*C^3*a^6*b^7*c^2*d^9*f^2 + 56*C^3*a^8*b^5*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - ((((-((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(40*C*a^3*b^14*d^12*f^4 + 192*C*a^5*b^12*d^12*f^4 + 360*C*a^7*b^10*d^12*f^4 + 320*C*a^9*b^8*d^12*f^4 + 120*C*a^11*b^6*d^12*f^4 - 8*C*a^15*b^2*d^12*f^4 + 8*C*b^17*c^3*d^9*f^4 + 40*C*a*b^16*c^2*d^10*f^4 + 32*C*a*b^16*c^4*d^8*f^4 - 88*C*a^2*b^15*c*d^11*f^4 - 448*C*a^4*b^13*c*d^11*f^4 - 920*C*a^6*b^11*c*d^11*f^4 - 960*C*a^8*b^9*c*d^11*f^4 - 520*C*a^10*b^7*c*d^11*f^4 - 128*C*a^12*b^5*c*d^11*f^4 - 8*C*a^14*b^3*c*d^11*f^4 - 32*C*a^2*b^15*c^3*d^9*f^4 + 256*C*a^3*b^14*c^2*d^10*f^4 + 160*C*a^3*b^14*c^4*d^8*f^4 - 280*C*a^4*b^13*c^3*d^9*f^4 + 680*C*a^5*b^12*c^2*d^10*f^4 + 320*C*a^5*b^12*c^4*d^8*f^4 - 640*C*a^6*b^11*c^3*d^9*f^4 + 960*C*a^7*b^10*c^2*d^10*f^4 + 320*C*a^7*b^10*c^4*d^8*f^4 - 680*C*a^8*b^9*c^3*d^9*f^4 + 760*C*a^9*b^8*c^2*d^10*f^4 + 160*C*a^9*b^8*c^4*d^8*f^4 - 352*C*a^10*b^7*c^3*d^9*f^4 + 320*C*a^11*b^6*c^2*d^10*f^4 + 32*C*a^11*b^6*c^4*d^8*f^4 - 72*C*a^12*b^5*c^3*d^9*f^4 + 56*C*a^13*b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - (4*(-((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 11

2*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)) + (16*(c + d*tan(e + f*x))^(1/2)*(20*C^2*a^5*b^10*d^11*f^2 - 60*C^2*a^3*b^12*d^11*f^2 + 168*C^2*a^7*b^8*d^11*f^2 + 40*C^2*a^9*b^6*d^11*f^2 - 44*C^2*a^11*b^4*d^11*f^2 + 4*C^2*a^13*b^2*d^11*f^2 - 20*C^2*b^15*c^3*d^8*f^2 - 4*C^2*a^14*b*c*d^10*f^2 - 20*C^2*a*b^14*c^2*d^9*f^2 + 100*C^2*a^2*b^13*c*d^10*f^2 - 300*C^2*a^6*b^9*c*d^10*f^2 - 160*C^2*a^8*b^7*c*d^10*f^2 + 76*C^2*a^10*b^5*c*d^10*f^2 + 32*C^2*a^12*b^3*c*d^10*f^2 + 116*C^2*a^2*b^13*c^3*d^8*f^2 - 124*C^2*a^3*b^12*c^2*d^9*f^2 + 216*C^2*a^4*b^11*c^3*d^8*f^2 - 40*C^2*a^5*b^10*c^2*d^9*f^2 + 8*C^2*a^6*b^9*c^3*d^8*f^2 + 168*C^2*a^7*b^8*c^2*d^9*f^2 - 68*C^2*a^8*b^7*c^3*d^8*f^2 + 60*C^2*a^9*b^6*c^2*d^9*f^2 + 4*C^2*a^10*b^5*c^3*d^8*f^2 - 44*C^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))*(-((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b

^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*
a^6*b^2*d^2*f^4)) + ((-((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256
*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C
^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 89
6*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 1
28*C^4*a^7*b*c*d*f^4)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 + 16*C^2*a*
b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d
^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4
+ 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2
f^4))^(1/2)((16*(c + d*tan(e + f*x))^(1/2)*(2*C^4*a^2*b^9*d^10 - 5*C^4*a^
4*b^7*d^10 + 17*C^4*a^6*b^5*d^10 - 7*C^4*a^8*b^3*d^10 + 2*C^4*b^11*c^2*d^8
+ C^4*a^10*b*d^10 - 12*C^4*a^2*b^9*c^2*d^8 + 18*C^4*a^4*b^7*c^2*d^8 - 4*C^4
*a*b^10*c*d^9 + 16*C^4*a^3*b^8*c*d^9 - 36*C^4*a^5*b^6*c*d^9 + 8*C^4*a^7*b^4
*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f
^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*
f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d
*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((-((5
12*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*
C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C
^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 89
6*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2
) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d
*f^2 - 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8
*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^
2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(8*C^3*a
^6*b^7*d^11*f^2 - 78*C^3*a^4*b^9*d^11*f^2 + 60*C^3*a^8*b^5*d^11*f^2 - 24*C^
3*a^10*b^3*d^11*f^2 + 2*C^3*a^12*b*d^11*f^2 - 32*C^3*a*b^12*c^3*d^8*f^2 + 1
52*C^3*a^3*b^10*c*d^10*f^2 + 128*C^3*a^5*b^8*c*d^10*f^2 - 64*C^3*a^7*b^6*c*
d^10*f^2 - 32*C^3*a^9*b^4*c*d^10*f^2 + 8*C^3*a^11*b^2*c*d^10*f^2 - 40*C^3*a
^2*b^11*c^2*d^9*f^2 + 64*C^3*a^3*b^10*c^3*d^8*f^2 - 216*C^3*a^4*b^9*c^2*d^9
*f^2 + 96*C^3*a^5*b^8*c^3*d^8*f^2 - 120*C^3*a^6*b^7*c^2*d^9*f^2 + 56*C^3*a^
8*b^5*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^
4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a
^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 -
2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^
5) - ((((-((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*
c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^
2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5
*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*
c*d*f^4)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 1
6*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*
c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c
^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*
((16*(40*C*a^3*b^14*d^12*f^4 + 192*C*a^5*b^12*d^12*f^4 + 360*C*a^7*b^10*d^1
2*f^4 + 320*C*a^9*b^8*d^12*f^4 + 120*C*a^11*b^6*d^12*f^4 - 8*C*a^15*b^2*d^1

2*f^4 + 8*C*b^17*c^3*d^9*f^4 + 40*C*a*b^16*c^2*d^10*f^4 + 32*C*a*b^16*c^4*d
^8*f^4 - 88*C*a^2*b^15*c*d^11*f^4 - 448*C*a^4*b^13*c*d^11*f^4 - 920*C*a^6*b
^11*c*d^11*f^4 - 960*C*a^8*b^9*c*d^11*f^4 - 520*C*a^10*b^7*c*d^11*f^4 - 128
*C*a^12*b^5*c*d^11*f^4 - 8*C*a^14*b^3*c*d^11*f^4 - 32*C*a^2*b^15*c^3*d^9*f^
4 + 256*C*a^3*b^14*c^2*d^10*f^4 + 160*C*a^3*b^14*c^4*d^8*f^4 - 280*C*a^4*b^
13*c^3*d^9*f^4 + 680*C*a^5*b^12*c^2*d^10*f^4 + 320*C*a^5*b^12*c^4*d^8*f^4 -
 640*C*a^6*b^11*c^3*d^9*f^4 + 960*C*a^7*b^10*c^2*d^10*f^4 + 320*C*a^7*b^10*
c^4*d^8*f^4 - 680*C*a^8*b^9*c^3*d^9*f^4 + 760*C*a^9*b^8*c^2*d^10*f^4 + 160*
C*a^9*b^8*c^4*d^8*f^4 - 352*C*a^10*b^7*c^3*d^9*f^4 + 320*C*a^11*b^6*c^2*d^1
0*f^4 + 32*C*a^11*b^6*c^4*d^8*f^4 - 72*C*a^12*b^5*c^3*d^9*f^4 + 56*C*a^13*b
^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*
b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4
*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*
a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5)
 + (4*(-((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^
2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*
f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c
*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*
d*f^4)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*
C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^
2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2
*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c
 + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 28
8*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a
^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c
^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^1
7*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*
a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4
 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d
^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b
^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*
a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4
 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9
*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*
f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^1
1*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c
*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^
4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4
*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^
6*b^2*d^2*f^4)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6
*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^
6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9
*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))
/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*
f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4

*d^2*f^4 + 4*a^6*b^2*d^2*f^4)) - (16*(c + d*tan(e + f*x))^(1/2)*(20*C^2*a^5
*b^10*d^11*f^2 - 60*C^2*a^3*b^12*d^11*f^2 + 168*C^2*a^7*b^8*d^11*f^2 + 40*C
^2*a^9*b^6*d^11*f^2 - 44*C^2*a^11*b^4*d^11*f^2 + 4*C^2*a^13*b^2*d^11*f^2 -
20*C^2*b^15*c^3*d^8*f^2 - 4*C^2*a^14*b*c*d^10*f^2 - 20*C^2*a*b^14*c^2*d^9*f
^2 + 100*C^2*a^2*b^13*c*d^10*f^2 - 300*C^2*a^6*b^9*c*d^10*f^2 - 160*C^2*a^8
*b^7*c*d^10*f^2 + 76*C^2*a^10*b^5*c*d^10*f^2 + 32*C^2*a^12*b^3*c*d^10*f^2 +
116*C^2*a^2*b^13*c^3*d^8*f^2 - 124*C^2*a^3*b^12*c^2*d^9*f^2 + 216*C^2*a^4*
b^11*c^3*d^8*f^2 - 40*C^2*a^5*b^10*c^2*d^9*f^2 + 8*C^2*a^6*b^9*c^3*d^8*f^2
+ 168*C^2*a^7*b^8*c^2*d^9*f^2 - 68*C^2*a^8*b^7*c^3*d^8*f^2 + 60*C^2*a^9*b^6
*c^2*d^9*f^2 + 4*C^2*a^10*b^5*c^3*d^8*f^2 - 44*C^2*a^11*b^4*c^2*d^9*f^2))/(
a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6
*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^
6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a
^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))*(-((512*C^4*a^4*b
^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*
f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d
^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^
3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) + 4*C^2*a^
4*c*f^2 + 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 - 24*C^
2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4
*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^
4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2))/(4*(a^8*c^2*f^4 + a^8*d^
2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 +
4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*
f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b
^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*
a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8
*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*
c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))*(-(
(512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 1
6*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608
*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 +
896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1
/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b
*d*f^2 - 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b
^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*
a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*1i)/(2*(a^8
*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*
a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4
+ 4*a^6*b^2*d^2*f^4)) - (atan(((((16*(c + d*tan(e + f*x))^(1/2)*(A^4*b^11*
d^10 + 7*A^4*a^2*b^9*d^10 + 11*A^4*a^4*b^7*d^10 - 27*A^4*a^6*b^5*d^10 - 2*A
^4*b^11*c^2*d^8 + 12*A^4*a^2*b^9*c^2*d^8 - 18*A^4*a^4*b^7*c^2*d^8 - 4*A^4*a
*b^10*c*d^9 - 24*A^4*a^3*b^8*c*d^9 + 44*A^4*a^5*b^6*c*d^9))/(a^10*d^2*f^4 +
b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 +
a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4

+ 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4
 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((((512*A^4*a^4*b^4*c^2*f^4 -
16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4
*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*
A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 1
28*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) - 4*A^2*a^4*c*f^2 - 4*A
^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f
^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*
f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4
*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(2*A^3*b^13*d^11*f^2 - 24*A^3*a^2
*b^11*d^11*f^2 - 196*A^3*a^4*b^9*d^11*f^2 - 120*A^3*a^6*b^7*d^11*f^2 + 50*A
^3*a^8*b^5*d^11*f^2 + 8*A^3*b^13*c^2*d^9*f^2 - 32*A^3*a*b^12*c^3*d^8*f^2 +
208*A^3*a^3*b^10*c*d^10*f^2 + 288*A^3*a^5*b^8*c*d^10*f^2 + 80*A^3*a^7*b^6*c
*d^10*f^2 - 8*A^3*a^2*b^11*c^2*d^9*f^2 + 64*A^3*a^3*b^10*c^3*d^8*f^2 - 232*
A^3*a^4*b^9*c^2*d^9*f^2 + 96*A^3*a^5*b^8*c^3*d^8*f^2 - 216*A^3*a^6*b^7*c^2*
d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*
f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2
*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*
d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - ((((5
12*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*
A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A
^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 89
6*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2
) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d
*f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8
*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^
2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*t
an(e + f*x))^(1/2)*(36*A^2*a^3*b^12*d^11*f^2 + 316*A^2*a^5*b^10*d^11*f^2 +
552*A^2*a^7*b^8*d^11*f^2 + 256*A^2*a^9*b^6*d^11*f^2 - 12*A^2*a^11*b^4*d^11*
f^2 - 4*A^2*a^13*b^2*d^11*f^2 - 20*A^2*b^15*c^3*d^8*f^2 + 8*A^2*a*b^14*d^11
*f^2 + 4*A^2*b^15*c*d^10*f^2 - 52*A^2*a*b^14*c^2*d^9*f^2 + 80*A^2*a^2*b^13*
c*d^10*f^2 - 156*A^2*a^4*b^11*c*d^10*f^2 - 640*A^2*a^6*b^9*c*d^10*f^2 - 500
*A^2*a^8*b^7*c*d^10*f^2 - 80*A^2*a^10*b^5*c*d^10*f^2 + 12*A^2*a^12*b^3*c*d^
10*f^2 + 116*A^2*a^2*b^13*c^3*d^8*f^2 - 220*A^2*a^3*b^12*c^2*d^9*f^2 + 216*
A^2*a^4*b^11*c^3*d^8*f^2 - 104*A^2*a^5*b^10*c^2*d^9*f^2 + 8*A^2*a^6*b^9*c^3
*d^8*f^2 + 232*A^2*a^7*b^8*c^2*d^9*f^2 - 68*A^2*a^8*b^7*c^3*d^8*f^2 + 156*A
^2*a^9*b^6*c^2*d^9*f^2 + 4*A^2*a^10*b^5*c^3*d^8*f^2 - 12*A^2*a^11*b^4*c^2*d
^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f
^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*
f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d
*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) + ((((51
2*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A
^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^
4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896
*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2)

- 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(16*A*a*b^16*d^12*f^4 - 16*A*b^17*c*d^11*f^4 + 136*A*a^3*b^14*d^12*f^4 + 432*A*a^5*b^12*d^12*f^4 + 680*A*a^7*b^10*d^12*f^4 + 560*A*a^9*b^8*d^12*f^4 + 216*A*a^11*b^6*d^12*f^4 + 16*A*a^13*b^4*d^12*f^4 - 8*A*a^15*b^2*d^12*f^4 - 8*A*b^17*c^3*d^9*f^4 + 56*A*a*b^16*c^2*d^10*f^4 + 32*A*a*b^16*c^4*d^8*f^4 - 184*A*a^2*b^15*c*d^11*f^4 - 688*A*a^4*b^13*c*d^11*f^4 - 1240*A*a^6*b^11*c*d^11*f^4 - 1200*A*a^8*b^9*c*d^11*f^4 - 616*A*a^10*b^7*c*d^11*f^4 - 144*A*a^12*b^5*c*d^11*f^4 - 8*A*a^14*b^3*c*d^11*f^4 - 128*A*a^2*b^15*c^3*d^9*f^4 + 352*A*a^3*b^14*c^2*d^10*f^4 + 160*A*a^3*b^14*c^4*d^8*f^4 - 520*A*a^4*b^13*c^3*d^9*f^4 + 920*A*a^5*b^12*c^2*d^10*f^4 + 320*A*a^5*b^12*c^4*d^8*f^4 - 960*A*a^6*b^11*c^3*d^9*f^4 + 1280*A*a^7*b^10*c^2*d^10*f^4 + 320*A*a^7*b^10*c^4*d^8*f^4 - 920*A*a^8*b^9*c^3*d^9*f^4 + 1000*A*a^9*b^8*c^2*d^10*f^4 + 160*A*a^9*b^8*c^4*d^8*f^4 - 448*A*a^10*b^7*c^3*d^9*f^4 + 416*A*a^11*b^6*c^2*d^10*f^4 + 32*A*a^11*b^6*c^4*d^8*f^4 - 88*A*a^12*b^5*c^3*d^9*f^4 + 72*A*a^13*b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - (4*(((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4

*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^
4)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4
*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 +
6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 -
 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^8*c^2
*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*
b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4
*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*
f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6
*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*
d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4
 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^
2*f^4)))*(((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*
c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^
2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5
*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*
c*d*f^4)^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 1
6*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*
c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c
^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*
1i)/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c
^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*
b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)) + (((16*(c + d*tan(e + f*x))^(1/2)*(A^4*b
^11*d^10 + 7*A^4*a^2*b^9*d^10 + 11*A^4*a^4*b^7*d^10 - 27*A^4*a^6*b^5*d^10 -
 2*A^4*b^11*c^2*d^8 + 12*A^4*a^2*b^9*c^2*d^8 - 18*A^4*a^4*b^7*c^2*d^8 - 4*A
^4*a*b^10*c*d^9 - 24*A^4*a^3*b^8*c*d^9 + 44*A^4*a^5*b^6*c*d^9))/(a^10*d^2*f
^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f
^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*
f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d
*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) + ((((512*A^4*a^4*b^4*c^2*f^
4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256
*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 +
192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4
 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) - 4*A^2*a^4*c*f^2 -
 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2
*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*
c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4
*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(2*A^3*b^13*d^11*f^2 - 24*A^3
*a^2*b^11*d^11*f^2 - 196*A^3*a^4*b^9*d^11*f^2 - 120*A^3*a^6*b^7*d^11*f^2 +
50*A^3*a^8*b^5*d^11*f^2 + 8*A^3*b^13*c^2*d^9*f^2 - 32*A^3*a*b^12*c^3*d^8*f^
2 + 208*A^3*a^3*b^10*c*d^10*f^2 + 288*A^3*a^5*b^8*c*d^10*f^2 + 80*A^3*a^7*b
^6*c*d^10*f^2 - 8*A^3*a^2*b^11*c^2*d^9*f^2 + 64*A^3*a^3*b^10*c^3*d^8*f^2 -
232*A^3*a^4*b^9*c^2*d^9*f^2 + 96*A^3*a^5*b^8*c^3*d^8*f^2 - 216*A^3*a^6*b^7*
c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*
c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6

*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*
b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + (
(((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 -
16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 6
08*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4
+ 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^
(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3
*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 +
b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 +
4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c +
d*tan(e + f*x))^(1/2)*(36*A^2*a^3*b^12*d^11*f^2 + 316*A^2*a^5*b^10*d^11*f^
2 + 552*A^2*a^7*b^8*d^11*f^2 + 256*A^2*a^9*b^6*d^11*f^2 - 12*A^2*a^11*b^4*d
^11*f^2 - 4*A^2*a^13*b^2*d^11*f^2 - 20*A^2*b^15*c^3*d^8*f^2 + 8*A^2*a*b^14*
d^11*f^2 + 4*A^2*b^15*c*d^10*f^2 - 52*A^2*a*b^14*c^2*d^9*f^2 + 80*A^2*a^2*b
^13*c*d^10*f^2 - 156*A^2*a^4*b^11*c*d^10*f^2 - 640*A^2*a^6*b^9*c*d^10*f^2 -
500*A^2*a^8*b^7*c*d^10*f^2 - 80*A^2*a^10*b^5*c*d^10*f^2 + 12*A^2*a^12*b^3*
c*d^10*f^2 + 116*A^2*a^2*b^13*c^3*d^8*f^2 - 220*A^2*a^3*b^12*c^2*d^9*f^2 +
216*A^2*a^4*b^11*c^3*d^8*f^2 - 104*A^2*a^5*b^10*c^2*d^9*f^2 + 8*A^2*a^6*b^9
*c^3*d^8*f^2 + 232*A^2*a^7*b^8*c^2*d^9*f^2 - 68*A^2*a^8*b^7*c^3*d^8*f^2 + 1
56*A^2*a^9*b^6*c^2*d^9*f^2 + 4*A^2*a^10*b^5*c^3*d^8*f^2 - 12*A^2*a^11*b^4*c
^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c
^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*
d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b
*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((
((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 -
16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 60
8*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 +
896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(
1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*
b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 +
b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4
*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(16*A
*a*b^16*d^12*f^4 - 16*A*b^17*c*d^11*f^4 + 136*A*a^3*b^14*d^12*f^4 + 432*A*a
^5*b^12*d^12*f^4 + 680*A*a^7*b^10*d^12*f^4 + 560*A*a^9*b^8*d^12*f^4 + 216*A
*a^11*b^6*d^12*f^4 + 16*A*a^13*b^4*d^12*f^4 - 8*A*a^15*b^2*d^12*f^4 - 8*A*b
^17*c^3*d^9*f^4 + 56*A*a*b^16*c^2*d^10*f^4 + 32*A*a*b^16*c^4*d^8*f^4 - 184*
A*a^2*b^15*c*d^11*f^4 - 688*A*a^4*b^13*c*d^11*f^4 - 1240*A*a^6*b^11*c*d^11*
f^4 - 1200*A*a^8*b^9*c*d^11*f^4 - 616*A*a^10*b^7*c*d^11*f^4 - 144*A*a^12*b^
5*c*d^11*f^4 - 8*A*a^14*b^3*c*d^11*f^4 - 128*A*a^2*b^15*c^3*d^9*f^4 + 352*A
*a^3*b^14*c^2*d^10*f^4 + 160*A*a^3*b^14*c^4*d^8*f^4 - 520*A*a^4*b^13*c^3*d^
9*f^4 + 920*A*a^5*b^12*c^2*d^10*f^4 + 320*A*a^5*b^12*c^4*d^8*f^4 - 960*A*a^
6*b^11*c^3*d^9*f^4 + 1280*A*a^7*b^10*c^2*d^10*f^4 + 320*A*a^7*b^10*c^4*d^8*
f^4 - 920*A*a^8*b^9*c^3*d^9*f^4 + 1000*A*a^9*b^8*c^2*d^10*f^4 + 160*A*a^9*b
^8*c^4*d^8*f^4 - 448*A*a^10*b^7*c^3*d^9*f^4 + 416*A*a^11*b^6*c^2*d^10*f^4 +
32*A*a^11*b^6*c^4*d^8*f^4 - 88*A*a^12*b^5*c^3*d^9*f^4 + 72*A*a^13*b^4*c^2*

d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2 *f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^ 2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c *d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + (4*(((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 60 8*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3* b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4 *a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*tan (e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^ 13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7* d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10* f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^ 8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15 *c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a ^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d ^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7 *c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^ 14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 4 8*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 30 4*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^ 4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8* d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^ 2*f^4)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^ 4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f ^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^8 *c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6* a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8* d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2 *b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2 *f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^ 2*d^2*f^4)))*(((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2* b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^ 6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3 *b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^ 7*b*c*d*f^4)^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2

+ 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*1i)/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)))/((32*(5*A^5*a^3*b^6*d^10 + A^5*a*b^8*d^10 - A^5*b^9*c*d^9 + 4*A^5*a*b^8*c^2*d^8 - 9*A^5*a^2*b^7*c*d^9))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + (((16*(c + d*tan(e + f*x))^(1/2)*(A^4*b^11*d^10 + 7*A^4*a^2*b^9*d^10 + 11*A^4*a^4*b^7*d^10 - 27*A^4*a^6*b^5*d^10 - 2*A^4*b^11*c^2*d^8 + 12*A^4*a^2*b^9*c^2*d^8 - 18*A^4*a^4*b^7*c^2*d^8 - 4*A^4*a*b^10*c*d^9 - 24*A^4*a^3*b^8*c*d^9 + 44*A^4*a^5*b^6*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(2*A^3*b^13*d^11*f^2 - 24*A^3*a^2*b^11*d^11*f^2 - 196*A^3*a^4*b^9*d^11*f^2 - 120*A^3*a^6*b^7*d^11*f^2 + 50*A^3*a^8*b^5*d^11*f^2 + 8*A^3*b^13*c^2*d^9*f^2 - 32*A^3*a*b^12*c^3*d^8*f^2 + 208*A^3*a^3*b^10*c*d^10*f^2 + 288*A^3*a^5*b^8*c*d^10*f^2 + 80*A^3*a^7*b^6*c*d^10*f^2 - 8*A^3*a^2*b^11*c^2*d^9*f^2 + 64*A^3*a^3*b^10*c^3*d^8*f^2 - 232*A^3*a^4*b^9*c^2*d^9*f^2 + 96*A^3*a^5*b^8*c^3*d^8*f^2 - 216*A^3*a^6*b^7*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - ((((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(36*A^2*a^3*b^12*d^11*f^2 + 316*A^2*a^5*b^10*d^11*f^2 + 552*A^2*a^7*b^8*d^11*f^2 + 256*A^2*a^9*b^6*d^11*f^2 -

12*A^2*a^11*b^4*d^11*f^2 - 4*A^2*a^13*b^2*d^11*f^2 - 20*A^2*b^15*c^3*d^8*f^2 + 8*A^2*a*b^14*d^11*f^2 + 4*A^2*b^15*c*d^10*f^2 - 52*A^2*a*b^14*c^2*d^9*f^2 + 80*A^2*a^2*b^13*c*d^10*f^2 - 156*A^2*a^4*b^11*c*d^10*f^2 - 640*A^2*a^6*b^9*c*d^10*f^2 - 500*A^2*a^8*b^7*c*d^10*f^2 - 80*A^2*a^10*b^5*c*d^10*f^2 + 12*A^2*a^12*b^3*c*d^10*f^2 + 116*A^2*a^2*b^13*c^3*d^8*f^2 - 220*A^2*a^3*b^12*c^2*d^9*f^2 + 216*A^2*a^4*b^11*c^3*d^8*f^2 - 104*A^2*a^5*b^10*c^2*d^9*f^2 + 8*A^2*a^6*b^9*c^3*d^8*f^2 + 232*A^2*a^7*b^8*c^2*d^9*f^2 - 68*A^2*a^8*b^7*c^3*d^8*f^2 + 156*A^2*a^9*b^6*c^2*d^9*f^2 + 4*A^2*a^10*b^5*c^3*d^8*f^2 - 12*A^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) + ((((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(16*A*a*b^16*d^12*f^4 - 16*A*b^17*c*d^11*f^4 + 136*A*a^3*b^14*d^12*f^4 + 432*A*a^5*b^12*d^12*f^4 + 680*A*a^7*b^10*d^12*f^4 + 560*A*a^9*b^8*d^12*f^4 + 216*A*a^11*b^6*d^12*f^4 + 16*A*a^13*b^4*d^12*f^4 - 8*A*a^15*b^2*d^12*f^4 - 8*A*b^17*c^3*d^9*f^4 + 56*A*a*b^16*c^2*d^10*f^4 + 32*A*a*b^16*c^4*d^8*f^4 - 184*A*a^2*b^15*c*d^11*f^4 - 688*A*a^4*b^13*c*d^11*f^4 - 1240*A*a^6*b^11*c*d^11*f^4 - 1200*A*a^8*b^9*c*d^11*f^4 - 616*A*a^10*b^7*c*d^11*f^4 - 144*A*a^12*b^5*c*d^11*f^4 - 8*A*a^14*b^3*c*d^11*f^4 - 128*A*a^2*b^15*c^3*d^9*f^4 + 352*A*a^3*b^14*c^2*d^10*f^4 + 160*A*a^3*b^14*c^4*d^8*f^4 - 520*A*a^4*b^13*c^3*d^9*f^4 + 920*A*a^5*b^12*c^2*d^10*f^4 + 320*A*a^5*b^12*c^4*d^8*f^4 - 960*A*a^6*b^11*c^3*d^9*f^4 + 1280*A*a^7*b^10*c^2*d^10*f^4 + 320*A*a^7*b^10*c^4*d^8*f^4 - 920*A*a^8*b^9*c^3*d^9*f^4 + 1000*A*a^9*b^8*c^2*d^10*f^4 + 160*A*a^9*b^8*c^4*d^8*f^4 - 448*A*a^10*b^7*c^3*d^9*f^4 + 416*A*a^11*b^6*c^2*d^10*f^4 + 32*A*a^11*b^6*c^4*d^8*f^4 - 88*A*a^12*b^5*c^3*d^9*f^4 + 72*A*a^13*b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - (4*(((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))

^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12
*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^
4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 +
32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 27
2*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f
^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2
*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b
^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 10
08*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f
^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3
*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^
4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18
*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b
^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a
^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a
^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 +
6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f
^4 + 4*a^6*b^2*d^2*f^4)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 +
6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 +
4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^
4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*
d*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2
*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 +
6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b
^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^
2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(
4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^
4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d
^2*f^4 + 4*a^6*b^2*d^2*f^4)))*(((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f
^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4
+ 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*
f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d
*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 1
6*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4
+ a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*
c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6
*b^2*d^2*f^4))^(1/2))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2
*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^
6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)) - (((16*(c + d*tan(e +
f*x))^(1/2)*(A^4*b^11*d^10 + 7*A^4*a^2*b^9*d^10 + 11*A^4*a^4*b^7*d^10 - 27*
A^4*a^6*b^5*d^10 - 2*A^4*b^11*c^2*d^8 + 12*A^4*a^2*b^9*c^2*d^8 - 18*A^4*a^4
*b^7*c^2*d^8 - 4*A^4*a*b^10*c*d^9 - 24*A^4*a^3*b^8*c*d^9 + 44*A^4*a^5*b^6*c
*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4
+ 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^
4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f

^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) + ((((512*
A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4
*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*
a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A
^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) -
 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^
2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^
2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b
^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(2*A^3*b^13
*d^11*f^2 - 24*A^3*a^2*b^11*d^11*f^2 - 196*A^3*a^4*b^9*d^11*f^2 - 120*A^3*a
^6*b^7*d^11*f^2 + 50*A^3*a^8*b^5*d^11*f^2 + 8*A^3*b^13*c^2*d^9*f^2 - 32*A^3
*a*b^12*c^3*d^8*f^2 + 208*A^3*a^3*b^10*c*d^10*f^2 + 288*A^3*a^5*b^8*c*d^10*
f^2 + 80*A^3*a^7*b^6*c*d^10*f^2 - 8*A^3*a^2*b^11*c^2*d^9*f^2 + 64*A^3*a^3*b
^10*c^3*d^8*f^2 - 232*A^3*a^4*b^9*c^2*d^9*f^2 + 96*A^3*a^5*b^8*c^3*d^8*f^2
- 216*A^3*a^6*b^7*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^
2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d
^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^
9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^
7*b^3*c*d*f^5) + ((((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4
*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a
^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^
4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A
^4*a^7*b*c*d*f^4)^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*
d*f^2 + 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f
^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*
a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4
))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(36*A^2*a^3*b^12*d^11*f^2 + 316*A^
2*a^5*b^10*d^11*f^2 + 552*A^2*a^7*b^8*d^11*f^2 + 256*A^2*a^9*b^6*d^11*f^2 -
 12*A^2*a^11*b^4*d^11*f^2 - 4*A^2*a^13*b^2*d^11*f^2 - 20*A^2*b^15*c^3*d^8*f
^2 + 8*A^2*a*b^14*d^11*f^2 + 4*A^2*b^15*c*d^10*f^2 - 52*A^2*a*b^14*c^2*d^9*
f^2 + 80*A^2*a^2*b^13*c*d^10*f^2 - 156*A^2*a^4*b^11*c*d^10*f^2 - 640*A^2*a^
6*b^9*c*d^10*f^2 - 500*A^2*a^8*b^7*c*d^10*f^2 - 80*A^2*a^10*b^5*c*d^10*f^2
+ 12*A^2*a^12*b^3*c*d^10*f^2 + 116*A^2*a^2*b^13*c^3*d^8*f^2 - 220*A^2*a^3*b
^12*c^2*d^9*f^2 + 216*A^2*a^4*b^11*c^3*d^8*f^2 - 104*A^2*a^5*b^10*c^2*d^9*f
^2 + 8*A^2*a^6*b^9*c^3*d^8*f^2 + 232*A^2*a^7*b^8*c^2*d^9*f^2 - 68*A^2*a^8*b
^7*c^3*d^8*f^2 + 156*A^2*a^9*b^6*c^2*d^9*f^2 + 4*A^2*a^10*b^5*c^3*d^8*f^2 -
 12*A^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2
*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^
2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9
*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7
*b^3*c*d*f^4) - ((((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*
a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^
2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4
*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^
4*a^7*b*c*d*f^4)^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d

*f^2 + 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^
4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a
^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)
)^(1/2)*((16*(16*A*a*b^16*d^12*f^4 - 16*A*b^17*c*d^11*f^4 + 136*A*a^3*b^14*
d^12*f^4 + 432*A*a^5*b^12*d^12*f^4 + 680*A*a^7*b^10*d^12*f^4 + 560*A*a^9*b^
8*d^12*f^4 + 216*A*a^11*b^6*d^12*f^4 + 16*A*a^13*b^4*d^12*f^4 - 8*A*a^15*b^
2*d^12*f^4 - 8*A*b^17*c^3*d^9*f^4 + 56*A*a*b^16*c^2*d^10*f^4 + 32*A*a*b^16*
c^4*d^8*f^4 - 184*A*a^2*b^15*c*d^11*f^4 - 688*A*a^4*b^13*c*d^11*f^4 - 1240*
A*a^6*b^11*c*d^11*f^4 - 1200*A*a^8*b^9*c*d^11*f^4 - 616*A*a^10*b^7*c*d^11*f
^4 - 144*A*a^12*b^5*c*d^11*f^4 - 8*A*a^14*b^3*c*d^11*f^4 - 128*A*a^2*b^15*c
^3*d^9*f^4 + 352*A*a^3*b^14*c^2*d^10*f^4 + 160*A*a^3*b^14*c^4*d^8*f^4 - 520
*A*a^4*b^13*c^3*d^9*f^4 + 920*A*a^5*b^12*c^2*d^10*f^4 + 320*A*a^5*b^12*c^4*
d^8*f^4 - 960*A*a^6*b^11*c^3*d^9*f^4 + 1280*A*a^7*b^10*c^2*d^10*f^4 + 320*A
*a^7*b^10*c^4*d^8*f^4 - 920*A*a^8*b^9*c^3*d^9*f^4 + 1000*A*a^9*b^8*c^2*d^10
*f^4 + 160*A*a^9*b^8*c^4*d^8*f^4 - 448*A*a^10*b^7*c^3*d^9*f^4 + 416*A*a^11*
b^6*c^2*d^10*f^4 + 32*A*a^11*b^6*c^4*d^8*f^4 - 88*A*a^12*b^5*c^3*d^9*f^4 +
72*A*a^13*b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f
^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*
f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c
*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b
^3*c*d*f^5) + (4*(((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*
a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^
2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4
*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^
4*a^7*b*c*d*f^4)^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d
*f^2 + 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^
4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a
^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)
)^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^1
2*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f
^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 +
 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 2
72*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*
f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^
2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*
b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1
008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*
f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^
3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b
^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^1
8*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*
b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*
a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((
a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 +
 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*

f^4 + 4*a^6*b^2*d^2*f^4)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)))*(((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))*(((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*1i)/(2*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)) - (atan(((((16*(c + d*tan(e + f*x))^(1/2)*(A^4*b^11*d^10 + 7*A^4*a^2*b^9*d^10 + 11*A^4*a^4*b^7*d^10 - 27*A^4*a^6*b^5*d^10 - 2*A^4*b^11*c^2*d^8 + 12*A^4*a^2*b^9*c^2*d^8 - 18*A^4*a^4*b^7*c^2*d^8 - 4*A^4*a*b^10*c*d^9 - 24*A^4*a^3*b^8*c*d^9 + 44*A^4*a^5*b^6*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((-((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4

+ a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c
^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*
b^2*d^2*f^4))^(1/2)*((16*(2*A^3*b^13*d^11*f^2 - 24*A^3*a^2*b^11*d^11*f^2 -
196*A^3*a^4*b^9*d^11*f^2 - 120*A^3*a^6*b^7*d^11*f^2 + 50*A^3*a^8*b^5*d^11*f
^2 + 8*A^3*b^13*c^2*d^9*f^2 - 32*A^3*a*b^12*c^3*d^8*f^2 + 208*A^3*a^3*b^10*
c*d^10*f^2 + 288*A^3*a^5*b^8*c*d^10*f^2 + 80*A^3*a^7*b^6*c*d^10*f^2 - 8*A^3
*a^2*b^11*c^2*d^9*f^2 + 64*A^3*a^3*b^10*c^3*d^8*f^2 - 232*A^3*a^4*b^9*c^2*d
^9*f^2 + 96*A^3*a^5*b^8*c^3*d^8*f^2 - 216*A^3*a^6*b^7*c^2*d^9*f^2))/(a^10*d
^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c
^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*
d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7
*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - ((-((512*A^4*a^4*b^4*c
^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4
- 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f
^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*
d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) + 4*A^2*a^4*c*
f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 - 24*A^2*a^
2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2
*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 +
6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/
2)*(36*A^2*a^3*b^12*d^11*f^2 + 316*A^2*a^5*b^10*d^11*f^2 + 552*A^2*a^7*b^8*
d^11*f^2 + 256*A^2*a^9*b^6*d^11*f^2 - 12*A^2*a^11*b^4*d^11*f^2 - 4*A^2*a^13
*b^2*d^11*f^2 - 20*A^2*b^15*c^3*d^8*f^2 + 8*A^2*a*b^14*d^11*f^2 + 4*A^2*b^1
5*c*d^10*f^2 - 52*A^2*a*b^14*c^2*d^9*f^2 + 80*A^2*a^2*b^13*c*d^10*f^2 - 156
*A^2*a^4*b^11*c*d^10*f^2 - 640*A^2*a^6*b^9*c*d^10*f^2 - 500*A^2*a^8*b^7*c*d
^10*f^2 - 80*A^2*a^10*b^5*c*d^10*f^2 + 12*A^2*a^12*b^3*c*d^10*f^2 + 116*A^2
*a^2*b^13*c^3*d^8*f^2 - 220*A^2*a^3*b^12*c^2*d^9*f^2 + 216*A^2*a^4*b^11*c^3
*d^8*f^2 - 104*A^2*a^5*b^10*c^2*d^9*f^2 + 8*A^2*a^6*b^9*c^3*d^8*f^2 + 232*A
^2*a^7*b^8*c^2*d^9*f^2 - 68*A^2*a^8*b^7*c^3*d^8*f^2 + 156*A^2*a^9*b^6*c^2*d
^9*f^2 + 4*A^2*a^10*b^5*c^3*d^8*f^2 - 12*A^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d
^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c
^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*
d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7
*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) + ((-((512*A^4*a^4*b^4*c
^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4
- 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f
^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*
d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) + 4*A^2*a^4*c*
f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 - 24*A^2*a^
2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2
*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 +
6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(16*A*a*b^16*d^12*f^4 -
16*A*b^17*c*d^11*f^4 + 136*A*a^3*b^14*d^12*f^4 + 432*A*a^5*b^12*d^12*f^4 +
680*A*a^7*b^10*d^12*f^4 + 560*A*a^9*b^8*d^12*f^4 + 216*A*a^11*b^6*d^12*f^4
+ 16*A*a^13*b^4*d^12*f^4 - 8*A*a^15*b^2*d^12*f^4 - 8*A*b^17*c^3*d^9*f^4 + 5

6*A*a*b^16*c^2*d^10*f^4 + 32*A*a*b^16*c^4*d^8*f^4 - 184*A*a^2*b^15*c*d^11*f
^4 - 688*A*a^4*b^13*c*d^11*f^4 - 1240*A*a^6*b^11*c*d^11*f^4 - 1200*A*a^8*b^
9*c*d^11*f^4 - 616*A*a^10*b^7*c*d^11*f^4 - 144*A*a^12*b^5*c*d^11*f^4 - 8*A*
a^14*b^3*c*d^11*f^4 - 128*A*a^2*b^15*c^3*d^9*f^4 + 352*A*a^3*b^14*c^2*d^10*
f^4 + 160*A*a^3*b^14*c^4*d^8*f^4 - 520*A*a^4*b^13*c^3*d^9*f^4 + 920*A*a^5*b
^12*c^2*d^10*f^4 + 320*A*a^5*b^12*c^4*d^8*f^4 - 960*A*a^6*b^11*c^3*d^9*f^4
+ 1280*A*a^7*b^10*c^2*d^10*f^4 + 320*A*a^7*b^10*c^4*d^8*f^4 - 920*A*a^8*b^9
*c^3*d^9*f^4 + 1000*A*a^9*b^8*c^2*d^10*f^4 + 160*A*a^9*b^8*c^4*d^8*f^4 - 44
8*A*a^10*b^7*c^3*d^9*f^4 + 416*A*a^11*b^6*c^2*d^10*f^4 + 32*A*a^11*b^6*c^4*
d^8*f^4 - 88*A*a^12*b^5*c^3*d^9*f^4 + 72*A*a^13*b^4*c^2*d^10*f^4))/(a^10*d^
2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^
2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d
^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*
c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - (4*(-((512*A^4*a^4*b^4*
c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4
 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*
f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c
*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) + 4*A^2*a^4*c
*f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 - 24*A^2*a
^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^
2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 +
 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(
32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*
a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^1
4*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*
d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^
16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912
*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^
4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4
*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*
b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48
*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^
4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^1
0*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*
f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^1
1*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c
*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*
f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f
^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)*(a^10*d^2*f
^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f
^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*
f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d
*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2
*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 +
4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f

^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)))*(-((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*1i)/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)) + (((16*(c + d*tan(e + f*x))^(1/2)*(A^4*b^11*d^10 + 7*A^4*a^2*b^9*d^10 + 11*A^4*a^4*b^7*d^10 - 27*A^4*a^6*b^5*d^10 - 2*A^4*b^11*c^2*d^8 + 12*A^4*a^2*b^9*c^2*d^8 - 18*A^4*a^4*b^7*c^2*d^8 - 4*A^4*a*b^10*c*d^9 - 24*A^4*a^3*b^8*c*d^9 + 44*A^4*a^5*b^6*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) + ((-((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(2*A^3*b^13*d^11*f^2 - 24*A^3*a^2*b^11*d^11*f^2 - 196*A^3*a^4*b^9*d^11*f^2 - 120*A^3*a^6*b^7*d^11*f^2 + 50*A^3*a^8*b^5*d^11*f^2 + 8*A^3*b^13*c^2*d^9*f^2 - 32*A^3*a*b^12*c^3*d^8*f^2 + 208*A^3*a^3*b^10*c*d^10*f^2 + 288*A^3*a^5*b^8*c*d^10*f^2 + 80*A^3*a^7*b^6*c*d^10*f^2 - 8*A^3*a^2*b^11*c^2*d^9*f^2 + 64*A^3*a^3*b^10*c^3*d^8*f^2 - 232*A^3*a^4*b^9*c^2*d^9*f^2 + 96*A^3*a^5*b^8*c^3*d^8*f^2 - 216*A^3*a^6*b^7*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + ((-((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 -

24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^
4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d
^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e +
f*x))^(1/2)*(36*A^2*a^3*b^12*d^11*f^2 + 316*A^2*a^5*b^10*d^11*f^2 + 552*A^2
*a^7*b^8*d^11*f^2 + 256*A^2*a^9*b^6*d^11*f^2 - 12*A^2*a^11*b^4*d^11*f^2 - 4
*A^2*a^13*b^2*d^11*f^2 - 20*A^2*b^15*c^3*d^8*f^2 + 8*A^2*a*b^14*d^11*f^2 +
4*A^2*b^15*c*d^10*f^2 - 52*A^2*a*b^14*c^2*d^9*f^2 + 80*A^2*a^2*b^13*c*d^10*
f^2 - 156*A^2*a^4*b^11*c*d^10*f^2 - 640*A^2*a^6*b^9*c*d^10*f^2 - 500*A^2*a^
8*b^7*c*d^10*f^2 - 80*A^2*a^10*b^5*c*d^10*f^2 + 12*A^2*a^12*b^3*c*d^10*f^2
+ 116*A^2*a^2*b^13*c^3*d^8*f^2 - 220*A^2*a^3*b^12*c^2*d^9*f^2 + 216*A^2*a^4
*b^11*c^3*d^8*f^2 - 104*A^2*a^5*b^10*c^2*d^9*f^2 + 8*A^2*a^6*b^9*c^3*d^8*f^
2 + 232*A^2*a^7*b^8*c^2*d^9*f^2 - 68*A^2*a^8*b^7*c^3*d^8*f^2 + 156*A^2*a^9*
b^6*c^2*d^9*f^2 + 4*A^2*a^10*b^5*c^3*d^8*f^2 - 12*A^2*a^11*b^4*c^2*d^9*f^2)
)/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*
a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6
*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 -
8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((-((512*A^4*
a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8
*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*
b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a
^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) + 4*A
^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 -
24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^
4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d
^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(16*A*a*b^16*d^
12*f^4 - 16*A*b^17*c*d^11*f^4 + 136*A*a^3*b^14*d^12*f^4 + 432*A*a^5*b^12*d^
12*f^4 + 680*A*a^7*b^10*d^12*f^4 + 560*A*a^9*b^8*d^12*f^4 + 216*A*a^11*b^6*
d^12*f^4 + 16*A*a^13*b^4*d^12*f^4 - 8*A*a^15*b^2*d^12*f^4 - 8*A*b^17*c^3*d^
9*f^4 + 56*A*a*b^16*c^2*d^10*f^4 + 32*A*a*b^16*c^4*d^8*f^4 - 184*A*a^2*b^15
*c*d^11*f^4 - 688*A*a^4*b^13*c*d^11*f^4 - 1240*A*a^6*b^11*c*d^11*f^4 - 1200
*A*a^8*b^9*c*d^11*f^4 - 616*A*a^10*b^7*c*d^11*f^4 - 144*A*a^12*b^5*c*d^11*f
^4 - 8*A*a^14*b^3*c*d^11*f^4 - 128*A*a^2*b^15*c^3*d^9*f^4 + 352*A*a^3*b^14*
c^2*d^10*f^4 + 160*A*a^3*b^14*c^4*d^8*f^4 - 520*A*a^4*b^13*c^3*d^9*f^4 + 92
0*A*a^5*b^12*c^2*d^10*f^4 + 320*A*a^5*b^12*c^4*d^8*f^4 - 960*A*a^6*b^11*c^3
*d^9*f^4 + 1280*A*a^7*b^10*c^2*d^10*f^4 + 320*A*a^7*b^10*c^4*d^8*f^4 - 920*
A*a^8*b^9*c^3*d^9*f^4 + 1000*A*a^9*b^8*c^2*d^10*f^4 + 160*A*a^9*b^8*c^4*d^8
*f^4 - 448*A*a^10*b^7*c^3*d^9*f^4 + 416*A*a^11*b^6*c^2*d^10*f^4 + 32*A*a^11
*b^6*c^4*d^8*f^4 - 88*A*a^12*b^5*c^3*d^9*f^4 + 72*A*a^13*b^4*c^2*d^10*f^4))
/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a
^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*
a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8
*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + (4*(-((512*A^4
*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^
8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4
*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*

a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)))*(-((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*1i)/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)))/((32*(5*A^5*a^3*b^6*d^10 + A^5*a*b^8*d^10 - A^5*b^9*c*d^9 + 4*A^5*a*b^8*c^2*d^8 - 9*A^5*a^2*b^7*c*d^9))/(a^10*d^2*f^5 +

b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + (((16*(c + d*tan(e + f*x))^(1/2)*(A^4*b^11*d^10 + 7*A^4*a^2*b^9*d^10 + 11*A^4*a^4*b^7*d^10 - 27*A^4*a^6*b^5*d^10 - 2*A^4*b^11*c^2*d^8 + 12*A^4*a^2*b^9*c^2*d^8 - 18*A^4*a^4*b^7*c^2*d^8 - 4*A^4*a*b^10*c*d^9 - 24*A^4*a^3*b^8*c*d^9 + 44*A^4*a^5*b^6*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((-((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(2*A^3*b^13*d^11*f^2 - 24*A^3*a^2*b^11*d^11*f^2 - 196*A^3*a^4*b^9*d^11*f^2 - 120*A^3*a^6*b^7*d^11*f^2 + 50*A^3*a^8*b^5*d^11*f^2 + 8*A^3*b^13*c^2*d^9*f^2 - 32*A^3*a*b^12*c^3*d^8*f^2 + 208*A^3*a^3*b^10*c*d^10*f^2 + 288*A^3*a^5*b^8*c*d^10*f^2 + 80*A^3*a^7*b^6*c*d^10*f^2 - 8*A^3*a^2*b^11*c^2*d^9*f^2 + 64*A^3*a^3*b^10*c^3*d^8*f^2 - 232*A^3*a^4*b^9*c^2*d^9*f^2 + 96*A^3*a^5*b^8*c^3*d^8*f^2 - 216*A^3*a^6*b^7*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - ((-((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(36*A^2*a^3*b^12*d^11*f^2 + 316*A^2*a^5*b^10*d^11*f^2 + 552*A^2*a^7*b^8*d^11*f^2 + 256*A^2*a^9*b^6*d^11*f^2 - 12*A^2*a^11*b^4*d^11*f^2 - 4*A^2*a^13*b^2*d^11*f^2 - 20*A^2*b^15*c^3*d^8*f^2 + 8*A^2*a*b^14*d^11*f^2 + 4*A^2*b^15*c*d^10*f^2 - 52*A^2*a*b^14*c^2*d^9*f^2 + 80*A^2*a^2*b^13*c*d^10*f^2 - 156*A^2*a^4*b^11*c*d^10*f^2 - 640*A^2*a^6*b^9*c*d^10*f^2 - 500*A^2*a^8*b^7*c*d^10*f^2 - 80*A^2*a^10*b^5*c*d^10*f^2 + 12*A^2*a^12*b^3*c*d^10*f^2 + 116*A^2*a^2*b^13*c^3*d^8*f^2 - 220*A^2*a^3*b^12*c^2*d^9*f^2 + 216*A^2*a^4*b^11*c^3*d^8*f^2 - 104*A^2*a^5*b^10*c^2*d^9*f^2 + 8*A^2*a^6*b^9*c^3*d^8*f^2 + 232*A^2*a^7*b^8*c^2*d^9*f^2 - 68*A^2*a^8*b^7*c^3*

d^8*f^2 + 156*A^2*a^9*b^6*c^2*d^9*f^2 + 4*A^2*a^10*b^5*c^3*d^8*f^2 - 12*A^2
*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 +
6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 +
4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^
4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*
d*f^4) + ((-((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^
6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*
d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b
^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*
b*c*d*f^4)^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 -
16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^
8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2
*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2
)*((16*(16*A*a*b^16*d^12*f^4 - 16*A*b^17*c*d^11*f^4 + 136*A*a^3*b^14*d^12*f
^4 + 432*A*a^5*b^12*d^12*f^4 + 680*A*a^7*b^10*d^12*f^4 + 560*A*a^9*b^8*d^12
*f^4 + 216*A*a^11*b^6*d^12*f^4 + 16*A*a^13*b^4*d^12*f^4 - 8*A*a^15*b^2*d^12
*f^4 - 8*A*b^17*c^3*d^9*f^4 + 56*A*a*b^16*c^2*d^10*f^4 + 32*A*a*b^16*c^4*d^
8*f^4 - 184*A*a^2*b^15*c*d^11*f^4 - 688*A*a^4*b^13*c*d^11*f^4 - 1240*A*a^6*
b^11*c*d^11*f^4 - 1200*A*a^8*b^9*c*d^11*f^4 - 616*A*a^10*b^7*c*d^11*f^4 - 1
44*A*a^12*b^5*c*d^11*f^4 - 8*A*a^14*b^3*c*d^11*f^4 - 128*A*a^2*b^15*c^3*d^9
*f^4 + 352*A*a^3*b^14*c^2*d^10*f^4 + 160*A*a^3*b^14*c^4*d^8*f^4 - 520*A*a^4
*b^13*c^3*d^9*f^4 + 920*A*a^5*b^12*c^2*d^10*f^4 + 320*A*a^5*b^12*c^4*d^8*f^
4 - 960*A*a^6*b^11*c^3*d^9*f^4 + 1280*A*a^7*b^10*c^2*d^10*f^4 + 320*A*a^7*b
^10*c^4*d^8*f^4 - 920*A*a^8*b^9*c^3*d^9*f^4 + 1000*A*a^9*b^8*c^2*d^10*f^4 +
160*A*a^9*b^8*c^4*d^8*f^4 - 448*A*a^10*b^7*c^3*d^9*f^4 + 416*A*a^11*b^6*c^
2*d^10*f^4 + 32*A*a^11*b^6*c^4*d^8*f^4 - 88*A*a^12*b^5*c^3*d^9*f^4 + 72*A*a
^13*b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6
*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 +
4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5
- 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d
f^5) - (4(-((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b
^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6
*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*
b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7
*b*c*d*f^4)^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2
- 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b
^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^
2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/
2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4
+ 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 -
288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b
^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^
2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 +
624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^1
0*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*

c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)))*(-((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)) - (((16*(c + d*tan(e + f*x))^(1/2)*(A^4*b^11*d^10 + 7*A^4*a^2*b^9*d^10 + 11*A^4*a^4*b^7*d^10 - 27*A^4*a^6*b^5*d^10 - 2*A^4*b^11*c^2*d^8 + 12*A^4*a^2*b^9*c^2*d^8 - 18*A^4*a^4*b^7*c^2*d^8 - 4*A^4*a*b^10*c*d^9 - 24*A^4*a^3*b^8*c*d^9 + 44*A^4*a^5*b^6*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) + ((-((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2* f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6 *d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(2*A^3*b^13*d ^11*f^2 - 24*A^3*a^2*b^11*d^11*f^2 - 196*A^3*a^4*b^9*d^11*f^2 - 120*A^3*a^6 *b^7*d^11*f^2 + 50*A^3*a^8*b^5*d^11*f^2 + 8*A^3*b^13*c^2*d^9*f^2 - 32*A^3*a *b^12*c^3*d^8*f^2 + 208*A^3*a^3*b^10*c*d^10*f^2 + 288*A^3*a^5*b^8*c*d^10*f^ 2 + 80*A^3*a^7*b^6*c*d^10*f^2 - 8*A^3*a^2*b^11*c^2*d^9*f^2 + 64*A^3*a^3*b^1 0*c^3*d^8*f^2 - 232*A^3*a^4*b^9*c^2*d^9*f^2 + 96*A^3*a^5*b^8*c^3*d^8*f^2 - 216*A^3*a^6*b^7*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2* f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2 *f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9* c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7* b^3*c*d*f^5) + ((-((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4* a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^ 2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4 *a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^ 4*a^7*b*c*d*f^4)^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d *f^2 - 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^ 4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a ^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(36*A^2*a^3*b^12*d^11*f^2 + 316*A^2 *a^5*b^10*d^11*f^2 + 552*A^2*a^7*b^8*d^11*f^2 + 256*A^2*a^9*b^6*d^11*f^2 - 12*A^2*a^11*b^4*d^11*f^2 - 4*A^2*a^13*b^2*d^11*f^2 - 20*A^2*b^15*c^3*d^8*f^ 2 + 8*A^2*a*b^14*d^11*f^2 + 4*A^2*b^15*c*d^10*f^2 - 52*A^2*a*b^14*c^2*d^9*f ^2 + 80*A^2*a^2*b^13*c*d^10*f^2 - 156*A^2*a^4*b^11*c*d^10*f^2 - 640*A^2*a^6 *b^9*c*d^10*f^2 - 500*A^2*a^8*b^7*c*d^10*f^2 - 80*A^2*a^10*b^5*c*d^10*f^2 + 12*A^2*a^12*b^3*c*d^10*f^2 + 116*A^2*a^2*b^13*c^3*d^8*f^2 - 220*A^2*a^3*b^ 12*c^2*d^9*f^2 + 216*A^2*a^4*b^11*c^3*d^8*f^2 - 104*A^2*a^5*b^10*c^2*d^9*f^ 2 + 8*A^2*a^6*b^9*c^3*d^8*f^2 + 232*A^2*a^7*b^8*c^2*d^9*f^2 - 68*A^2*a^8*b^ 7*c^3*d^8*f^2 + 156*A^2*a^9*b^6*c^2*d^9*f^2 + 4*A^2*a^10*b^5*c^3*d^8*f^2 - 12*A^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2* f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2 *f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9* c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7* b^3*c*d*f^4) - ((-((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4* a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^ 2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4 *a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^ 4*a^7*b*c*d*f^4)^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d *f^2 - 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^ 4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a ^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(16*A*a*b^16*d^12*f^4 - 16*A*b^17*c*d^11*f^4 + 136*A*a^3*b^14* d^12*f^4 + 432*A*a^5*b^12*d^12*f^4 + 680*A*a^7*b^10*d^12*f^4 + 560*A*a^9*b^ 8*d^12*f^4 + 216*A*a^11*b^6*d^12*f^4 + 16*A*a^13*b^4*d^12*f^4 - 8*A*a^15*b^ 2*d^12*f^4 - 8*A*b^17*c^3*d^9*f^4 + 56*A*a*b^16*c^2*d^10*f^4 + 32*A*a*b^16* c^4*d^8*f^4 - 184*A*a^2*b^15*c*d^11*f^4 - 688*A*a^4*b^13*c*d^11*f^4 - 1240*A*a^6*b^11*c*d^11*f^4 - 1200*A*a^8*b^9*c*d^11*f^4 - 616*A*a^10*b^7*c*d^11*f^4 - 144*A*a^12*b^5*c*d^11*f^4 - 8*A*a^14*b^3*c*d^11*f^4 - 128*A*a^2*b^15*c^3*d^9*f^4 + 352*A*a^3*b^14*c^2*d^10*f^4 + 160*A*a^3*b^14*c^4*d^8*f^4 - 520*A*a^4*b^13*c^3*d^9*f^4 + 920*A*a^5*b^12*c^2*d^10*f^4 + 320*A*a^5*b^12*c^4*d^8*f^4 - 960*A*a^6*b^11*c^3*d^9*f^4 + 1280*A*a^7*b^10*c^2*d^10*f^4 + 320*A*a^7*b^10*c^4*d^8*f^4 - 920*A*a^8*b^9*c^3*d^9*f^4 + 1000*A*a^9*b^8*c^2*d^10*f^4 + 160*A*a^9*b^8*c^4*d^8*f^4 - 448*A*a^10*b^7*c^3*d^9*f^4 + 416*A*a^11*b^6*c^2*d^10*f^4 + 32*A*a^11*b^6*c^4*d^8*f^4 - 88*A*a^12*b^5*c^3*d^9*f^4 + 72*A*a^13*b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + (4*(-((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 +

b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*
b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))
/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*
f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4
*d^2*f^4 + 4*a^6*b^2*d^2*f^4)))*(-((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^
2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*
f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d
^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*
c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2
+ 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*
f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b
^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*
a^6*b^2*d^2*f^4))^(1/2))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*
d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2
*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))*(-((512*A^4*a^4*b^
4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f
^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^
2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3
*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) + 4*A^2*a^4
*c*f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 - 24*A^2
*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*
a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4
+ 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*1i)/(2*(a^8*c^2*f^4 + a^8*
d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4
+ 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^
2*f^4)) - (atan((((-(A^2*b^7*d^2 + 16*A^2*a^2*b^5*c^2 + 10*A^2*a^2*b^5*d^2
+ 25*A^2*a^4*b^3*d^2 - 40*A^2*a^3*b^4*c*d - 8*A^2*a*b^6*c*d)*(a^11*d^3*f^2
- b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2
- a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2
+ 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*
c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*
f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 +
3*a^9*b^2*c^2*d*f^2))^(1/2)*((((16*(2*A^3*b^13*d^11*f^2 - 24*A^3*a^2*b^11*d
^11*f^2 - 196*A^3*a^4*b^9*d^11*f^2 - 120*A^3*a^6*b^7*d^11*f^2 + 50*A^3*a^8*
b^5*d^11*f^2 + 8*A^3*b^13*c^2*d^9*f^2 - 32*A^3*a*b^12*c^3*d^8*f^2 + 208*A^3
*a^3*b^10*c*d^10*f^2 + 288*A^3*a^5*b^8*c*d^10*f^2 + 80*A^3*a^7*b^6*c*d^10*f
^2 - 8*A^3*a^2*b^11*c^2*d^9*f^2 + 64*A^3*a^3*b^10*c^3*d^8*f^2 - 232*A^3*a^4
*b^9*c^2*d^9*f^2 + 96*A^3*a^5*b^8*c^3*d^8*f^2 - 216*A^3*a^6*b^7*c^2*d^9*f^2
))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4
*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 +
6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 -
8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - ((((-(A^2*b^
7*d^2 + 16*A^2*a^2*b^5*c^2 + 10*A^2*a^2*b^5*d^2 + 25*A^2*a^4*b^3*d^2 - 40*A
^2*a^3*b^4*c*d - 8*A^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*
c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8

*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*
b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*
d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2
+ 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)
((16(16*A*a*b^16*d^12*f^4 - 16*A*b^17*c*d^11*f^4 + 136*A*a^3*b^14*d^12*f^
4 + 432*A*a^5*b^12*d^12*f^4 + 680*A*a^7*b^10*d^12*f^4 + 560*A*a^9*b^8*d^12*
f^4 + 216*A*a^11*b^6*d^12*f^4 + 16*A*a^13*b^4*d^12*f^4 - 8*A*a^15*b^2*d^12*
f^4 - 8*A*b^17*c^3*d^9*f^4 + 56*A*a*b^16*c^2*d^10*f^4 + 32*A*a*b^16*c^4*d^8
*f^4 - 184*A*a^2*b^15*c*d^11*f^4 - 688*A*a^4*b^13*c*d^11*f^4 - 1240*A*a^6*b
^11*c*d^11*f^4 - 1200*A*a^8*b^9*c*d^11*f^4 - 616*A*a^10*b^7*c*d^11*f^4 - 14
4*A*a^12*b^5*c*d^11*f^4 - 8*A*a^14*b^3*c*d^11*f^4 - 128*A*a^2*b^15*c^3*d^9*
f^4 + 352*A*a^3*b^14*c^2*d^10*f^4 + 160*A*a^3*b^14*c^4*d^8*f^4 - 520*A*a^4*
b^13*c^3*d^9*f^4 + 920*A*a^5*b^12*c^2*d^10*f^4 + 320*A*a^5*b^12*c^4*d^8*f^4
 - 960*A*a^6*b^11*c^3*d^9*f^4 + 1280*A*a^7*b^10*c^2*d^10*f^4 + 320*A*a^7*b^
10*c^4*d^8*f^4 - 920*A*a^8*b^9*c^3*d^9*f^4 + 1000*A*a^9*b^8*c^2*d^10*f^4 +
160*A*a^9*b^8*c^4*d^8*f^4 - 448*A*a^10*b^7*c^3*d^9*f^4 + 416*A*a^11*b^6*c^2
*d^10*f^4 + 32*A*a^11*b^6*c^4*d^8*f^4 - 88*A*a^12*b^5*c^3*d^9*f^4 + 72*A*a^
13*b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*
a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4
*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5
- 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*
f^5) - (16*(-(A^2*b^7*d^2 + 16*A^2*a^2*b^5*c^2 + 10*A^2*a^2*b^5*d^2 + 25*A^
2*a^4*b^3*d^2 - 40*A^2*a^3*b^4*c*d - 8*A^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*
c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b
^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^
9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f
^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 1
8*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b
^2*c^2*d*f^2))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160
*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^
10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b
^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2
*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b
^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 11
2*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f
^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^
3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11
*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 20
8*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^
4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f
^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*
f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^
11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*
d^11*f^4))/((b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 2
4*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3

*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2) + (16*(c + d*tan(e + f*x))^(1/2)*(36*A^2*a^3*b^12*d^11*f^2 + 316*A^2*a^5*b^10*d^11*f^2 + 552*A^2*a^7*b^8*d^11*f^2 + 256*A^2*a^9*b^6*d^11*f^2 - 12*A^2*a^11*b^4*d^11*f^2 - 4*A^2*a^13*b^2*d^11*f^2 - 20*A^2*b^15*c^3*d^8*f^2 + 8*A^2*a*b^14*d^11*f^2 + 4*A^2*b^15*c*d^10*f^2 - 52*A^2*a*b^14*c^2*d^9*f^2 + 80*A^2*a^2*b^13*c*d^10*f^2 - 156*A^2*a^4*b^11*c*d^10*f^2 - 640*A^2*a^6*b^9*c*d^10*f^2 - 500*A^2*a^8*b^7*c*d^10*f^2 - 80*A^2*a^10*b^5*c*d^10*f^2 + 12*A^2*a^12*b^3*c*d^10*f^2 + 116*A^2*a^2*b^13*c^3*d^8*f^2 - 220*A^2*a^3*b^12*c^2*d^9*f^2 + 216*A^2*a^4*b^11*c^3*d^8*f^2 - 104*A^2*a^5*b^10*c^2*d^9*f^2 + 8*A^2*a^6*b^9*c^3*d^8*f^2 + 232*A^2*a^7*b^8*c^2*d^9*f^2 - 68*A^2*a^8*b^7*c^3*d^8*f^2 + 156*A^2*a^9*b^6*c^2*d^9*f^2 + 4*A^2*a^10*b^5*c^3*d^8*f^2 - 12*A^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))*(-(A^2*b^7*d^2 + 16*A^2*a^2*b^5*c^2 + 10*A^2*a^2*b^5*d^2 + 25*A^2*a^4*b^3*d^2 - 40*A^2*a^3*b^4*c*d - 8*A^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2))*(-(A^2*b^7*d^2 + 16*A^2*a^2*b^5*c^2 + 10*A^2*a^2*b^5*d^2 + 25*A^2*a^4*b^3*d^2 - 40*A^2*a^3*b^4*c*d - 8*A^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5

*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c
*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f
^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4
*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 +
 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^
3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11
*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2) - (16*
(c + d*tan(e + f*x))^(1/2)*(A^4*b^11*d^10 + 7*A^4*a^2*b^9*d^10 + 11*A^4*a^4
*b^7*d^10 - 27*A^4*a^6*b^5*d^10 - 2*A^4*b^11*c^2*d^8 + 12*A^4*a^2*b^9*c^2*d
^8 - 18*A^4*a^4*b^7*c^2*d^8 - 4*A^4*a*b^10*c*d^9 - 24*A^4*a^3*b^8*c*d^9 + 4
4*A^4*a^5*b^6*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*
a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4
*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4
- 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*
f^4))*1i)/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*
a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f
^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3
*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^
6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*
a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2) - ((-(A^2*b^7*d^2 + 16*A^2*a^2*b^5*c
^2 + 10*A^2*a^2*b^5*d^2 + 25*A^2*a^4*b^3*d^2 - 40*A^2*a^3*b^4*c*d - 8*A^2*a
*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*
f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3
*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*
b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2
*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 -
 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*((((16*(2*A^3*b^13*d^11
*f^2 - 24*A^3*a^2*b^11*d^11*f^2 - 196*A^3*a^4*b^9*d^11*f^2 - 120*A^3*a^6*b^
7*d^11*f^2 + 50*A^3*a^8*b^5*d^11*f^2 + 8*A^3*b^13*c^2*d^9*f^2 - 32*A^3*a*b^
12*c^3*d^8*f^2 + 208*A^3*a^3*b^10*c*d^10*f^2 + 288*A^3*a^5*b^8*c*d^10*f^2 +
 80*A^3*a^7*b^6*c*d^10*f^2 - 8*A^3*a^2*b^11*c^2*d^9*f^2 + 64*A^3*a^3*b^10*c
^3*d^8*f^2 - 232*A^3*a^4*b^9*c^2*d^9*f^2 + 96*A^3*a^5*b^8*c^3*d^8*f^2 - 216
*A^3*a^6*b^7*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5
 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^
5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d
*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3
*c*d*f^5) - ((((-(A^2*b^7*d^2 + 16*A^2*a^2*b^5*c^2 + 10*A^2*a^2*b^5*d^2 + 2
5*A^2*a^4*b^3*d^2 - 40*A^2*a^3*b^4*c*d - 8*A^2*a*b^6*c*d)*(a^11*d^3*f^2 - b
^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a
^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 +
4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d
^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2
 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a
^9*b^2*c^2*d*f^2))^(1/2)*((16*(16*A*a*b^16*d^12*f^4 - 16*A*b^17*c*d^11*f^4
+ 136*A*a^3*b^14*d^12*f^4 + 432*A*a^5*b^12*d^12*f^4 + 680*A*a^7*b^10*d^12*f

^4 + 560*A*a^9*b^8*d^12*f^4 + 216*A*a^11*b^6*d^12*f^4 + 16*A*a^13*b^4*d^12*
f^4 - 8*A*a^15*b^2*d^12*f^4 - 8*A*b^17*c^3*d^9*f^4 + 56*A*a*b^16*c^2*d^10*f
^4 + 32*A*a*b^16*c^4*d^8*f^4 - 184*A*a^2*b^15*c*d^11*f^4 - 688*A*a^4*b^13*c
*d^11*f^4 - 1240*A*a^6*b^11*c*d^11*f^4 - 1200*A*a^8*b^9*c*d^11*f^4 - 616*A*
a^10*b^7*c*d^11*f^4 - 144*A*a^12*b^5*c*d^11*f^4 - 8*A*a^14*b^3*c*d^11*f^4 -
 128*A*a^2*b^15*c^3*d^9*f^4 + 352*A*a^3*b^14*c^2*d^10*f^4 + 160*A*a^3*b^14*
c^4*d^8*f^4 - 520*A*a^4*b^13*c^3*d^9*f^4 + 920*A*a^5*b^12*c^2*d^10*f^4 + 32
0*A*a^5*b^12*c^4*d^8*f^4 - 960*A*a^6*b^11*c^3*d^9*f^4 + 1280*A*a^7*b^10*c^2
*d^10*f^4 + 320*A*a^7*b^10*c^4*d^8*f^4 - 920*A*a^8*b^9*c^3*d^9*f^4 + 1000*A
*a^9*b^8*c^2*d^10*f^4 + 160*A*a^9*b^8*c^4*d^8*f^4 - 448*A*a^10*b^7*c^3*d^9*
f^4 + 416*A*a^11*b^6*c^2*d^10*f^4 + 32*A*a^11*b^6*c^4*d^8*f^4 - 88*A*a^12*b
^5*c^3*d^9*f^4 + 72*A*a^13*b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5
+ 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f
^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^
2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*
c*d*f^5 - 8*a^7*b^3*c*d*f^5) + (16*(-(A^2*b^7*d^2 + 16*A^2*a^2*b^5*c^2 + 10
*A^2*a^2*b^5*d^2 + 25*A^2*a^4*b^3*d^2 - 40*A^2*a^3*b^4*c*d - 8*A^2*a*b^6*c*
d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4
*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 +
6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2
*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 +
18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8
*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32
*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^
8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*
b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^
8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16
*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a
^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4
- 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d
^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^
9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a
^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4
- 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*
f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^
4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*
f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d
^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2)
+ b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d
^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a
^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^
2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3
*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2)*(a^10*d^2*
f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*
f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2

*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*
d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4)))))/(b^9*(8*a^2*c^3*f^2 + 6*
a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f
^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^
9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4
*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^
2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*
f^2) - (16*(c + d*tan(e + f*x))^(1/2)*(36*A^2*a^3*b^12*d^11*f^2 + 316*A^2*a
^5*b^10*d^11*f^2 + 552*A^2*a^7*b^8*d^11*f^2 + 256*A^2*a^9*b^6*d^11*f^2 - 12
*A^2*a^11*b^4*d^11*f^2 - 4*A^2*a^13*b^2*d^11*f^2 - 20*A^2*b^15*c^3*d^8*f^2
+ 8*A^2*a*b^14*d^11*f^2 + 4*A^2*b^15*c*d^10*f^2 - 52*A^2*a*b^14*c^2*d^9*f^2
+ 80*A^2*a^2*b^13*c*d^10*f^2 - 156*A^2*a^4*b^11*c*d^10*f^2 - 640*A^2*a^6*b
^9*c*d^10*f^2 - 500*A^2*a^8*b^7*c*d^10*f^2 - 80*A^2*a^10*b^5*c*d^10*f^2 + 1
2*A^2*a^12*b^3*c*d^10*f^2 + 116*A^2*a^2*b^13*c^3*d^8*f^2 - 220*A^2*a^3*b^12
*c^2*d^9*f^2 + 216*A^2*a^4*b^11*c^3*d^8*f^2 - 104*A^2*a^5*b^10*c^2*d^9*f^2
+ 8*A^2*a^6*b^9*c^3*d^8*f^2 + 232*A^2*a^7*b^8*c^2*d^9*f^2 - 68*A^2*a^8*b^7*
c^3*d^8*f^2 + 156*A^2*a^9*b^6*c^2*d^9*f^2 + 4*A^2*a^10*b^5*c^3*d^8*f^2 - 12
*A^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^
4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f
^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*
d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^
3*c*d*f^4))*(-(A^2*b^7*d^2 + 16*A^2*a^2*b^5*c^2 + 10*A^2*a^2*b^5*d^2 + 25*A
^2*a^4*b^3*d^2 - 40*A^2*a^3*b^4*c*d - 8*A^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11
*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*
b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a
^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*
f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 -
18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*
b^2*c^2*d*f^2))^(1/2))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*
c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5
*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2)
- b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^
2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11
*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2))*(-(A^2*b^7*d^2 + 16*A^
2*a^2*b^5*c^2 + 10*A^2*a^2*b^5*d^2 + 25*A^2*a^4*b^3*d^2 - 40*A^2*a^3*b^4*c*
d - 8*A^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a
^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*
a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^
2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^
4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*
c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2))/(b^9*(8*a^2
*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*
(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2
) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2
*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36

*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*
a^10*b*c*d^2*f^2) + (16*(c + d*tan(e + f*x))^(1/2)*(A^4*b^11*d^10 + 7*A^4*a
^2*b^9*d^10 + 11*A^4*a^4*b^7*d^10 - 27*A^4*a^6*b^5*d^10 - 2*A^4*b^11*c^2*d^
8 + 12*A^4*a^2*b^9*c^2*d^8 - 18*A^4*a^4*b^7*c^2*d^8 - 4*A^4*a*b^10*c*d^9 -
24*A^4*a^3*b^8*c*d^9 + 44*A^4*a^5*b^6*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4
+ 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f
^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^
2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*
c*d*f^4 - 8*a^7*b^3*c*d*f^4))*1i)/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) +
b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2
f^2) + b^5(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9
*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2
+ 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f
^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2))/(((-(A^2*b^
7*d^2 + 16*A^2*a^2*b^5*c^2 + 10*A^2*a^2*b^5*d^2 + 25*A^2*a^4*b^3*d^2 - 40*A
^2*a^3*b^4*c*d - 8*A^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*
c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8
*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*
b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*
d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2
+ 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)
((((16(2*A^3*b^13*d^11*f^2 - 24*A^3*a^2*b^11*d^11*f^2 - 196*A^3*a^4*b^9*d
^11*f^2 - 120*A^3*a^6*b^7*d^11*f^2 + 50*A^3*a^8*b^5*d^11*f^2 + 8*A^3*b^13*c
^2*d^9*f^2 - 32*A^3*a*b^12*c^3*d^8*f^2 + 208*A^3*a^3*b^10*c*d^10*f^2 + 288*
A^3*a^5*b^8*c*d^10*f^2 + 80*A^3*a^7*b^6*c*d^10*f^2 - 8*A^3*a^2*b^11*c^2*d^9
*f^2 + 64*A^3*a^3*b^10*c^3*d^8*f^2 - 232*A^3*a^4*b^9*c^2*d^9*f^2 + 96*A^3*a
^5*b^8*c^3*d^8*f^2 - 216*A^3*a^6*b^7*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2
*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*
c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b
^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5
*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - ((((-(A^2*b^7*d^2 + 16*A^2*a^2*b^5*c^2
+ 10*A^2*a^2*b^5*d^2 + 25*A^2*a^4*b^3*d^2 - 40*A^2*a^3*b^4*c*d - 8*A^2*a*b^
6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2
 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^
2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c
*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^
2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12
*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*((16*(16*A*a*b^16*d^12*f^4
 - 16*A*b^17*c*d^11*f^4 + 136*A*a^3*b^14*d^12*f^4 + 432*A*a^5*b^12*d^12*f^4
 + 680*A*a^7*b^10*d^12*f^4 + 560*A*a^9*b^8*d^12*f^4 + 216*A*a^11*b^6*d^12*f
^4 + 16*A*a^13*b^4*d^12*f^4 - 8*A*a^15*b^2*d^12*f^4 - 8*A*b^17*c^3*d^9*f^4
+ 56*A*a*b^16*c^2*d^10*f^4 + 32*A*a*b^16*c^4*d^8*f^4 - 184*A*a^2*b^15*c*d^1
1*f^4 - 688*A*a^4*b^13*c*d^11*f^4 - 1240*A*a^6*b^11*c*d^11*f^4 - 1200*A*a^8
*b^9*c*d^11*f^4 - 616*A*a^10*b^7*c*d^11*f^4 - 144*A*a^12*b^5*c*d^11*f^4 - 8
*A*a^14*b^3*c*d^11*f^4 - 128*A*a^2*b^15*c^3*d^9*f^4 + 352*A*a^3*b^14*c^2*d^

10*f^4 + 160*A*a^3*b^14*c^4*d^8*f^4 - 520*A*a^4*b^13*c^3*d^9*f^4 + 920*A*a^5*b^12*c^2*d^10*f^4 + 320*A*a^5*b^12*c^4*d^8*f^4 - 960*A*a^6*b^11*c^3*d^9*f^4 + 1280*A*a^7*b^10*c^2*d^10*f^4 + 320*A*a^7*b^10*c^4*d^8*f^4 - 920*A*a^8*b^9*c^3*d^9*f^4 + 1000*A*a^9*b^8*c^2*d^10*f^4 + 160*A*a^9*b^8*c^4*d^8*f^4 - 448*A*a^10*b^7*c^3*d^9*f^4 + 416*A*a^11*b^6*c^2*d^10*f^4 + 32*A*a^11*b^6*c^4*d^8*f^4 - 88*A*a^12*b^5*c^3*d^9*f^4 + 72*A*a^13*b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - (16*(-(A^2*b^7*d^2 + 16*A^2*a^2*b^5*c^2 + 10*A^2*a^2*b^5*d^2 + 25*A^2*a^4*b^3*d^2 - 40*A^2*a^3*b^4*c*d - 8*A^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d

^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2
*d*f^2 + 6*a^10*b*c*d^2*f^2) + (16*(c + d*tan(e + f*x))^(1/2)*(36*A^2*a^3*b
^12*d^11*f^2 + 316*A^2*a^5*b^10*d^11*f^2 + 552*A^2*a^7*b^8*d^11*f^2 + 256*A
^2*a^9*b^6*d^11*f^2 - 12*A^2*a^11*b^4*d^11*f^2 - 4*A^2*a^13*b^2*d^11*f^2 -
20*A^2*b^15*c^3*d^8*f^2 + 8*A^2*a*b^14*d^11*f^2 + 4*A^2*b^15*c*d^10*f^2 - 5
2*A^2*a*b^14*c^2*d^9*f^2 + 80*A^2*a^2*b^13*c*d^10*f^2 - 156*A^2*a^4*b^11*c*
d^10*f^2 - 640*A^2*a^6*b^9*c*d^10*f^2 - 500*A^2*a^8*b^7*c*d^10*f^2 - 80*A^2
*a^10*b^5*c*d^10*f^2 + 12*A^2*a^12*b^3*c*d^10*f^2 + 116*A^2*a^2*b^13*c^3*d^
8*f^2 - 220*A^2*a^3*b^12*c^2*d^9*f^2 + 216*A^2*a^4*b^11*c^3*d^8*f^2 - 104*A
^2*a^5*b^10*c^2*d^9*f^2 + 8*A^2*a^6*b^9*c^3*d^8*f^2 + 232*A^2*a^7*b^8*c^2*d
^9*f^2 - 68*A^2*a^8*b^7*c^3*d^8*f^2 + 156*A^2*a^9*b^6*c^2*d^9*f^2 + 4*A^2*a
^10*b^5*c^3*d^8*f^2 - 12*A^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^
2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2
*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*
b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^
5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))*(-(A^2*b^7*d^2 + 16*A^2*a^2*b^5*c^2 + 1
0*A^2*a^2*b^5*d^2 + 25*A^2*a^4*b^3*d^2 - 40*A^2*a^3*b^4*c*d - 8*A^2*a*b^6*c
d)(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 -
4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 +
6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^
2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 +
18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^
8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*
c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 +
24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^
3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12
*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) -
2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2)
)*(-(A^2*b^7*d^2 + 16*A^2*a^2*b^5*c^2 + 10*A^2*a^2*b^5*d^2 + 25*A^2*a^4*b^3
*d^2 - 40*A^2*a^3*b^4*c*d - 8*A^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 -
4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^
2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3
*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a
^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5
*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*
f^2))^(1/2))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 +
24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^
3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*
a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) -
b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 -
6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2) - (16*(c + d*tan(e + f*x))^(1/2)*
(A^4*b^11*d^10 + 7*A^4*a^2*b^9*d^10 + 11*A^4*a^4*b^7*d^10 - 27*A^4*a^6*b^5*
d^10 - 2*A^4*b^11*c^2*d^8 + 12*A^4*a^2*b^9*c^2*d^8 - 18*A^4*a^4*b^7*c^2*d^8
- 4*A^4*a*b^10*c*d^9 - 24*A^4*a^3*b^8*c*d^9 + 44*A^4*a^5*b^6*c*d^9))/(a^10
*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4

*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^
4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b
^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4)))/(b^9*(8*a^2*c^3*f^2
+ 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c
^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(
8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) -
b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*
d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*
d^2*f^2) - (32*(5*A^5*a^3*b^6*d^10 + A^5*a*b^8*d^10 - A^5*b^9*c*d^9 + 4*A^5
*a*b^8*c^2*d^8 - 9*A^5*a^2*b^7*c*d^9))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2
*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^
2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 -
2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5
- 8*a^7*b^3*c*d*f^5) + ((-(A^2*b^7*d^2 + 16*A^2*a^2*b^5*c^2 + 10*A^2*a^2*b
^5*d^2 + 25*A^2*a^4*b^3*d^2 - 40*A^2*a^3*b^4*c*d - 8*A^2*a*b^6*c*d)*(a^11*d
^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c
^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*
d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a
^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6
*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2
*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*((((16*(2*A^3*b^13*d^11*f^2 - 24*A^3*a^2
*b^11*d^11*f^2 - 196*A^3*a^4*b^9*d^11*f^2 - 120*A^3*a^6*b^7*d^11*f^2 + 50*A
^3*a^8*b^5*d^11*f^2 + 8*A^3*b^13*c^2*d^9*f^2 - 32*A^3*a*b^12*c^3*d^8*f^2 +
208*A^3*a^3*b^10*c*d^10*f^2 + 288*A^3*a^5*b^8*c*d^10*f^2 + 80*A^3*a^7*b^6*c
*d^10*f^2 - 8*A^3*a^2*b^11*c^2*d^9*f^2 + 64*A^3*a^3*b^10*c^3*d^8*f^2 - 232*
A^3*a^4*b^9*c^2*d^9*f^2 + 96*A^3*a^5*b^8*c^3*d^8*f^2 - 216*A^3*a^6*b^7*c^2*
d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*
f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2
*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*
d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - ((((-
(A^2*b^7*d^2 + 16*A^2*a^2*b^5*c^2 + 10*A^2*a^2*b^5*d^2 + 25*A^2*a^4*b^3*d^2
- 40*A^2*a^3*b^4*c*d - 8*A^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a
^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 +
a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2
+ 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b
^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d
^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2)
)^(1/2)*((16*(16*A*a*b^16*d^12*f^4 - 16*A*b^17*c*d^11*f^4 + 136*A*a^3*b^14*
d^12*f^4 + 432*A*a^5*b^12*d^12*f^4 + 680*A*a^7*b^10*d^12*f^4 + 560*A*a^9*b^
8*d^12*f^4 + 216*A*a^11*b^6*d^12*f^4 + 16*A*a^13*b^4*d^12*f^4 - 8*A*a^15*b^
2*d^12*f^4 - 8*A*b^17*c^3*d^9*f^4 + 56*A*a*b^16*c^2*d^10*f^4 + 32*A*a*b^16*
c^4*d^8*f^4 - 184*A*a^2*b^15*c*d^11*f^4 - 688*A*a^4*b^13*c*d^11*f^4 - 1240*
A*a^6*b^11*c*d^11*f^4 - 1200*A*a^8*b^9*c*d^11*f^4 - 616*A*a^10*b^7*c*d^11*f
^4 - 144*A*a^12*b^5*c*d^11*f^4 - 8*A*a^14*b^3*c*d^11*f^4 - 128*A*a^2*b^15*c
^3*d^9*f^4 + 352*A*a^3*b^14*c^2*d^10*f^4 + 160*A*a^3*b^14*c^4*d^8*f^4 - 520

*A*a^4*b^13*c^3*d^9*f^4 + 920*A*a^5*b^12*c^2*d^10*f^4 + 320*A*a^5*b^12*c^4*d^8*f^4 - 960*A*a^6*b^11*c^3*d^9*f^4 + 1280*A*a^7*b^10*c^2*d^10*f^4 + 320*A*a^7*b^10*c^4*d^8*f^4 - 920*A*a^8*b^9*c^3*d^9*f^4 + 1000*A*a^9*b^8*c^2*d^10*f^4 + 160*A*a^9*b^8*c^4*d^8*f^4 - 448*A*a^10*b^7*c^3*d^9*f^4 + 416*A*a^11*b^6*c^2*d^10*f^4 + 32*A*a^11*b^6*c^4*d^8*f^4 - 88*A*a^12*b^5*c^3*d^9*f^4 + 72*A*a^13*b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + (16*(-(A^2*b^7*d^2 + 16*A^2*a^2*b^5*c^2 + 10*A^2*a^2*b^5*d^2 + 25*A^2*a^4*b^3*d^2 - 40*A^2*a^3*b^4*c*d - 8*A^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f

^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2) - (16*(c + d
*tan(e + f*x))^(1/2)*(36*A^2*a^3*b^12*d^11*f^2 + 316*A^2*a^5*b^10*d^11*f^2
+ 552*A^2*a^7*b^8*d^11*f^2 + 256*A^2*a^9*b^6*d^11*f^2 - 12*A^2*a^11*b^4*d^1
1*f^2 - 4*A^2*a^13*b^2*d^11*f^2 - 20*A^2*b^15*c^3*d^8*f^2 + 8*A^2*a*b^14*d^
11*f^2 + 4*A^2*b^15*c*d^10*f^2 - 52*A^2*a*b^14*c^2*d^9*f^2 + 80*A^2*a^2*b^1
3*c*d^10*f^2 - 156*A^2*a^4*b^11*c*d^10*f^2 - 640*A^2*a^6*b^9*c*d^10*f^2 - 5
00*A^2*a^8*b^7*c*d^10*f^2 - 80*A^2*a^10*b^5*c*d^10*f^2 + 12*A^2*a^12*b^3*c*
d^10*f^2 + 116*A^2*a^2*b^13*c^3*d^8*f^2 - 220*A^2*a^3*b^12*c^2*d^9*f^2 + 21
6*A^2*a^4*b^11*c^3*d^8*f^2 - 104*A^2*a^5*b^10*c^2*d^9*f^2 + 8*A^2*a^6*b^9*c
^3*d^8*f^2 + 232*A^2*a^7*b^8*c^2*d^9*f^2 - 68*A^2*a^8*b^7*c^3*d^8*f^2 + 156
*A^2*a^9*b^6*c^2*d^9*f^2 + 4*A^2*a^10*b^5*c^3*d^8*f^2 - 12*A^2*a^11*b^4*c^2
*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2
*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^
2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c
*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))*(-(A^
2*b^7*d^2 + 16*A^2*a^2*b^5*c^2 + 10*A^2*a^2*b^5*d^2 + 25*A^2*a^4*b^3*d^2 -
40*A^2*a^3*b^4*c*d - 8*A^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*
b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3
*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 +
3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*
c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*
f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(
1/2))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*
c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 +
36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3
*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8
*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^
10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2))*(-(A^2*b^7*d^2 + 16*A^2*a^2*b^5*c^2 + 1
0*A^2*a^2*b^5*d^2 + 25*A^2*a^4*b^3*d^2 - 40*A^2*a^3*b^4*c*d - 8*A^2*a*b^6*c
d)(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 -
4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 +
6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^
2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 +
18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^
8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*
c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 +
24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^
3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12
*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) -
2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2)
+ (16*(c + d*tan(e + f*x))^(1/2)*(A^4*b^11*d^10 + 7*A^4*a^2*b^9*d^10 + 11*
A^4*a^4*b^7*d^10 - 27*A^4*a^6*b^5*d^10 - 2*A^4*b^11*c^2*d^8 + 12*A^4*a^2*b^
9*c^2*d^8 - 18*A^4*a^4*b^7*c^2*d^8 - 4*A^4*a*b^10*c*d^9 - 24*A^4*a^3*b^8*c*
d^9 + 44*A^4*a^5*b^6*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f
^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*

f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c
*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b
^3*c*d*f^4)))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 +
 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c
^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2
*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2)
- b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2
- 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2)))*(-(A^2*b^7*d^2 + 16*A^2*a^2*b^
5*c^2 + 10*A^2*a^2*b^5*d^2 + 25*A^2*a^4*b^3*d^2 - 40*A^2*a^3*b^4*c*d - 8*A^
2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c
^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*
d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^
10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*
d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^
2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*2i)/(b^9*(8*a^2*c^3*
f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a
^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b
^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^
2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*
c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*
b*c*d^2*f^2) + (A*b^2*d*(c + d*tan(e + f*x))^(1/2))/((b*f*(c + d*tan(e + f*
x)) + a*d*f - b*c*f)*(a^3*d - b^3*c - a^2*b*c + a*b^2*d)) + (C*a^2*d*(c + d
*tan(e + f*x))^(1/2))/((b*f*(c + d*tan(e + f*x)) + a*d*f - b*c*f)*(a^3*d -
b^3*c - a^2*b*c + a*b^2*d)) - (B*a*b*d*(c + d*tan(e + f*x))^(1/2))/((b*f*(c
 + d*tan(e + f*x)) + a*d*f - b*c*f)*(a^3*d - b^3*c - a^2*b*c + a*b^2*d))

3.116 $\int \frac{(a+b\tan(e+fx))^3(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx$

Optimal result 1637
Rubi [A] (verified) 1638
Mathematica [C] (verified) 1643
Maple [B] (verified) 1644
Fricas [F(-1)] 1645
Sympy [F] 1645
Maxima [F(-1)] 1645
Giac [F(-1)] 1645
Mupad [F(-1)] 1646

Optimal result

Integrand size = 47, antiderivative size = 511

$$\int \frac{(a+b\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{3/2}}\,dx =$$

$$-\frac{(a-ib)^3(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(c-id)^{3/2}f}$$

$$-\frac{(ia-b)^3(A+iB-C)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(c+id)^{3/2}f}$$

$$-\frac{2(c^2C-Bcd+Ad^2)\,(a+b\tan(e+fx))^3}{d\,(c^2+d^2)\,f\sqrt{c+d\tan(e+fx)}}$$

$$+\frac{2b(6a^2d^2(12c^2C-5Bcd+(5A+7C)d^2)-15abd(8c^3C-6Bc^2d+c(3A+5C)d^2-3Bd^3)+b^2(48c^4C-\cdots}{15d^4\,(c^2+d^2)\,f}$$

$$-\frac{2b^2(4(bc-ad)\,(6c^2C-5Bcd+(5A+C)d^2)-5d^2((A-C)(bc-ad)+B(ac+bd)))\tan(e+fx)\sqrt{c+\cdots}}{15d^3\,(c^2+d^2)\,f}$$

$$+\frac{2b(6c^2C-5Bcd+(5A+C)d^2)\,(a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}}{5d^2\,(c^2+d^2)\,f}$$

```
[Out] -(a-I*b)^3*(I*A+B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(c-I*d
)^(3/2)/f-(I*a-b)^3*(A+I*B-C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))
/(c+I*d)^(3/2)/f+2/15*b*(6*a^2*d^2*(12*c^2*C-5*B*c*d+(5*A+7*C)*d^2)-15*a*b*
d*(8*c^3*C-6*B*c^2*d+c*(3*A+5*C)*d^2-3*B*d^3)+b^2*(48*c^4*C-40*B*c^3*d+6*c^
2*(5*A+3*C)*d^2-25*B*c*d^3+15*(A-C)*d^4))*(c+d*tan(f*x+e))^(1/2)/d^4/(c^2+d
^2)/f-2/15*b^2*(4*(-a*d+b*c)*(6*c^2*C-5*B*c*d+(5*A+C)*d^2)-5*d^2*((A-C)*(-a
*d+b*c)+B*(a*c+b*d)))*(c+d*tan(f*x+e))^(1/2)*tan(f*x+e)/d^3/(c^2+d^2)/f+2/5
*b*(6*c^2*C-5*B*c*d+(5*A+C)*d^2)*(c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^2/
```

```
d^2/(c^2+d^2)/f-2*(A*d^2-B*c*d+C*c^2)*(a+b*tan(f*x+e))^3/d/(c^2+d^2)/f/(c+d
*tan(f*x+e))^(1/2)
```

Rubi [A] (verified)

Time = 3.38 (sec) , antiderivative size = 511, normalized size of antiderivative = 1.00, number of steps used = 11, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}} = 0.170$, Rules used = {3726, 3728, 3718, 3711, 3620, 3618, 65, 214}

$$\int \frac{(a+b\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx = \frac{2b\sqrt{c+d\tan(e+fx)}(6a^2d^2(d^2(5A+7C$$

$$-\frac{(-b+ia)^3(A+iB-C)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f(c+id)^{3/2}}$$

$$-\frac{(a-ib)^3(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f(c-id)^{3/2}}$$

$$-\frac{2b^2\tan(e+fx)\sqrt{c+d\tan(e+fx)}(4(bc-ad)\,(d^2(5A+C)-5Bcd+6c^2C)-5d^2((A-C)(bc-ad)+B}{15d^3f\,(c^2+d^2)}$$

$$+\frac{2b(d^2(5A+C)-5Bcd+6c^2C)\,(a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}}{5d^2f\,(c^2+d^2)}$$

$$-\frac{2(Ad^2-Bcd+c^2C)\,(a+b\tan(e+fx))^3}{df\,(c^2+d^2)\sqrt{c+d\tan(e+fx)}}$$

```
[In] Int[((a + b*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c + d
*Tan[e + f*x])^(3/2),x]
```

```
[Out] -(((a - I*b)^3*(I*A + B - I*C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*
d]])/((c - I*d)^(3/2)*f)) - ((I*a - b)^3*(A + I*B - C)*ArcTanh[Sqrt[c + d*T
an[e + f*x]]/Sqrt[c + I*d]])/((c + I*d)^(3/2)*f) - (2*(c^2*C - B*c*d + A*d^
2)*(a + b*Tan[e + f*x])^3)/(d*(c^2 + d^2)*f*Sqrt[c + d*Tan[e + f*x]]) + (2*
b*(6*a^2*d^2*(12*c^2*C - 5*B*c*d + (5*A + 7*C)*d^2) - 15*a*b*d*(8*c^3*C - 6
*B*c^2*d + c*(3*A + 5*C)*d^2 - 3*B*d^3) + b^2*(48*c^4*C - 40*B*c^3*d + 6*c^
2*(5*A + 3*C)*d^2 - 25*B*c*d^3 + 15*(A - C)*d^4))*Sqrt[c + d*Tan[e + f*x]])
/(15*d^4*(c^2 + d^2)*f) - (2*b^2*(4*(b*c - a*d)*(6*c^2*C - 5*B*c*d + (5*A +
 C)*d^2) - 5*d^2*((A - C)*(b*c - a*d) + B*(a*c + b*d)))*Tan[e + f*x]*Sqrt[c
 + d*Tan[e + f*x]])/(15*d^3*(c^2 + d^2)*f) + (2*b*(6*c^2*C - 5*B*c*d + (5*A
 + C)*d^2)*(a + b*Tan[e + f*x])^2*Sqrt[c + d*Tan[e + f*x]])/(5*d^2*(c^2 + d
^2)*f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
```

```
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3726

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
```

```
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*d^2 + c*(c*C - B*d))*(a + b*Tan[e +
 f*x])^m*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\text{integral} = -\frac{2(c^2C-Bcd+Ad^2)\left(a+b\tan(e+fx)\right)^3}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}} + \frac{2\int \frac{(a+b\tan(e+fx))^2\left(\frac{1}{2}\left(Ad(ac+6bd)+2\left(3bc-\frac{ad}{2}\right)(cC-Bd)\right)+\frac{1}{2}d((A-C)(bc-ad)+B(ac+bd))\tan(e+fx)+\frac{1}{2}b\left(6c^2C-5Bcd+(5A+C)d^2\right)\text{ta}}{\sqrt{c+d\tan(e+fx)}}}{d\left(c^2+d^2\right)}$$

$$= -\frac{2(c^2C-Bcd+Ad^2)\left(a+b\tan(e+fx)\right)^3}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}} + \frac{2b(6c^2C-5Bcd+(5A+C)d^2)\left(a+b\tan(e+fx)\right)^2\sqrt{c+d\tan(e+fx)}}{5d^2\left(c^2+d^2\right)f} + \frac{4\int \frac{(a+b\tan(e+fx))\left(\frac{1}{4}\left(-b(4bc+ad)\left(6c^2C-5Bcd+(5A+C)d^2\right)+5ad(Ad(ac+6bd)+(6bc-ad)(cC-Bd))\right)+\frac{5}{4}d^2\left(2ab(Ac-cC+Bd)+\right.}{\sqrt{}}}{5d}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)(a + b\tan(e + fx))^3}{d(c^2 + d^2) f\sqrt{c + d\tan(e + fx)}} - \frac{2b^2(4(bc - ad)(6c^2C - 5Bcd + (5A + C)d^2) - 5d^2((A - C)(bc - ad) + B(ac + bd)))\tan(e +}{15d^3(c^2 + d^2) f} + \frac{2b(6c^2C - 5Bcd + (5A + C)d^2)(a + b\tan(e + fx))^2\sqrt{c + d\tan(e + fx)}}{5d^2(c^2 + d^2) f} - \frac{8\int \frac{\frac{1}{8}(-15a^3d^3(Ac - cC + Bd) - 3a^2bd^2(24c^2C - 25Bcd + (25A - C)d^2) + 30ab^2cd(4c^2C - 3Bcd + (3A + C)d^2) - 2b^3c(24c^3C - 20Bc^2d}{}}{}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)(a + b\tan(e + fx))^3}{d(c^2 + d^2) f\sqrt{c + d\tan(e + fx)}} + \frac{2b(6a^2d^2(12c^2C - 5Bcd + (5A + 7C)d^2) - 15abd(8c^3C - 6Bc^2d + c(3A + 5C)d^2 - 3Bd^3) + b}{15d^4(c^2 + d} - \frac{2b^2(4(bc - ad)(6c^2C - 5Bcd + (5A + C)d^2) - 5d^2((A - C)(bc - ad) + B(ac + bd)))\tan(e +}{15d^3(c^2 + d^2) f} + \frac{2b(6c^2C - 5Bcd + (5A + C)d^2)(a + b\tan(e + fx))^2\sqrt{c + d\tan(e + fx)}}{5d^2(c^2 + d^2) f} - \frac{8\int \frac{-\frac{15}{8}d^3(a^3(Ac - cC + Bd) - 3ab^2(Ac - cC + Bd) - 3a^2b(Bc - (A - C)d) + b^3(Bc - (A - C)d)) - \frac{15}{8}d^3(3a^2b(Ac - cC + Bd) - b^3(Ac - cC}{\sqrt{c + d\tan(e + fx)}}}{15d^3(c^2 + d^2)}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)(a + b\tan(e + fx))^3}{d(c^2 + d^2) f\sqrt{c + d\tan(e + fx)}} + \frac{2b(6a^2d^2(12c^2C - 5Bcd + (5A + 7C)d^2) - 15abd(8c^3C - 6Bc^2d + c(3A + 5C)d^2 - 3Bd^3) + b}{15d^4(c^2 + d} - \frac{2b^2(4(bc - ad)(6c^2C - 5Bcd + (5A + C)d^2) - 5d^2((A - C)(bc - ad) + B(ac + bd)))\tan(e +}{15d^3(c^2 + d^2) f} + \frac{2b(6c^2C - 5Bcd + (5A + C)d^2)(a + b\tan(e + fx))^2\sqrt{c + d\tan(e + fx)}}{5d^2(c^2 + d^2) f} + \frac{((a - ib)^3(A - iB - C))\int \frac{1 + i\tan(e + fx)}{\sqrt{c + d\tan(e + fx)}}\,dx}{2(c - id)} + \frac{((a + ib)^3(A + iB - C))\int \frac{1 - i\tan(e + fx)}{\sqrt{c + d\tan(e + fx)}}\,dx}{2(c + id)}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)(a + b\tan(e + fx))^3}{d(c^2 + d^2) f\sqrt{c + d\tan(e + fx)}}$$
$$+ \frac{2b(6a^2d^2(12c^2C - 5Bcd + (5A + 7C)d^2) - 15abd(8c^3C - 6Bc^2d + c(3A + 5C)d^2 - 3Bd^3) + b^2(}{15d^4(c^2 + d^2)}$$
$$- \frac{2b^2(4(bc - ad)(6c^2C - 5Bcd + (5A + C)d^2) - 5d^2((A - C)(bc - ad) + B(ac + bd)))\tan(e + fx}{15d^3(c^2 + d^2) f}$$
$$+ \frac{2b(6c^2C - 5Bcd + (5A + C)d^2)(a + b\tan(e + fx))^2\sqrt{c + d\tan(e + fx)}}{5d^2(c^2 + d^2) f}$$
$$+ \frac{(i(a - ib)^3(A - iB - C))\,\mathrm{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c-idx}}\,dx, x, i\tan(e + fx)\right)}{2(c - id)f}$$
$$- \frac{(i(a + ib)^3(A + iB - C))\,\mathrm{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c+idx}}\,dx, x, -i\tan(e + fx)\right)}{2(c + id)f}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)(a + b\tan(e + fx))^3}{d(c^2 + d^2) f\sqrt{c + d\tan(e + fx)}}$$
$$+ \frac{2b(6a^2d^2(12c^2C - 5Bcd + (5A + 7C)d^2) - 15abd(8c^3C - 6Bc^2d + c(3A + 5C)d^2 - 3Bd^3) + b^2(}{15d^4(c^2 + d^2)}$$
$$- \frac{2b^2(4(bc - ad)(6c^2C - 5Bcd + (5A + C)d^2) - 5d^2((A - C)(bc - ad) + B(ac + bd)))\tan(e + fx}{15d^3(c^2 + d^2) f}$$
$$+ \frac{2b(6c^2C - 5Bcd + (5A + C)d^2)(a + b\tan(e + fx))^2\sqrt{c + d\tan(e + fx)}}{5d^2(c^2 + d^2) f}$$
$$- \frac{((a - ib)^3(A - iB - C))\,\mathrm{Subst}\left(\int \frac{1}{-1-\frac{ic}{d}+\frac{ix^2}{d}}\,dx, x, \sqrt{c + d\tan(e + fx)}\right)}{(c - id)df}$$
$$- \frac{((a + ib)^3(A + iB - C))\,\mathrm{Subst}\left(\int \frac{1}{-1+\frac{ic}{d}-\frac{ix^2}{d}}\,dx, x, \sqrt{c + d\tan(e + fx)}\right)}{(c + id)df}$$

$$= -\frac{(a-ib)^3(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(c-id)^{3/2}f}$$
$$-\frac{(ia-b)^3(A+iB-C)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(c+id)^{3/2}f}$$
$$-\frac{2(c^2C-Bcd+Ad^2)\,(a+b\tan(e+fx))^3}{d\,(c^2+d^2)\,f\sqrt{c+d\tan(e+fx)}}$$
$$+\frac{2b(6a^2d^2(12c^2C-5Bcd+(5A+7C)d^2)-15abd(8c^3C-6Bc^2d+c(3A+5C)d^2-3Bd^3)+b^2}{15d^4\,(c^2+d^2}$$
$$-\frac{2b^2(4(bc-ad)\,(6c^2C-5Bcd+(5A+C)d^2)-5d^2((A-C)(bc-ad)+B(ac+bd)))\tan(e+}{15d^3\,(c^2+d^2)\,f}$$
$$+\frac{2b(6c^2C-5Bcd+(5A+C)d^2)\,(a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}}{5d^2\,(c^2+d^2)\,f}$$

Mathematica [C] (verified)

Result contains higher order function than in optimal. Order 5 vs. order 3 in optimal.

Time = 6.88 (sec) , antiderivative size = 920, normalized size of antiderivative = 1.80

$$\int \frac{(a+b\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx = \frac{2C(a+b\tan(e+fx))^3}{5df\sqrt{c+d\tan(e+fx)}}$$

$$+\frac{2\left(\frac{(-6bcC+5bBd+6aCd)(a+b\tan(e+fx))^2}{3df\sqrt{c+d\tan(e+fx)}}+\frac{2\left(\frac{\left(15b(Ab+aB-bC)d^2+4(bc-ad)(6bcC-5bBd-6aCd)\right)(a+b\tan(e+fx))}{2df\sqrt{c+d\tan(e+fx)}}+\frac{-\frac{2\left(-48b^3c^3C+40b^3Bc^2d+\right.}{}}{}\right.}{}\right.}{}$$

```
[In] Integrate[((a + b*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(c + d*Tan[e + f*x])^(3/2),x]
```

```
[Out] (2*C*(a + b*Tan[e + f*x])^3)/(5*d*f*Sqrt[c + d*Tan[e + f*x]]) + (2*(((-6*b*
c*C + 5*b*B*d + 6*a*C*d)*(a + b*Tan[e + f*x])^2)/(3*d*f*Sqrt[c + d*Tan[e +
f*x]]) + (2*(((15*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C - 5*b*B*
d - 6*a*C*d))*(a + b*Tan[e + f*x]))/(2*d*f*Sqrt[c + d*Tan[e + f*x]]) + ((-2
*(-48*b^3*c^3*C + 40*b^3*B*c^2*d + 144*a*b^2*c^2*C*d - 30*A*b^3*c*d^2 - 110
*a*b^2*B*c*d^2 - 144*a^2*b*c*C*d^2 + 30*b^3*c*C*d^2 + 60*a*A*b^2*d^3 + 85*a
^2*b*B*d^3 - 15*b^3*B*d^3 + 48*a^3*C*d^3 - 60*a*b^2*C*d^3))/(d*Sqrt[c + d*T
an[e + f*x]]) + (2*(((45*a^2*A*b*d^3 - 15*A*b^3*d^3 + 15*a^3*B*d^3 - 45*a*b
^2*B*d^3 - 45*a^2*b*C*d^3 + 15*b^3*C*d^3)*(((-I)*ArcTanh[Sqrt[c + d*Tan[e +
 f*x]]/Sqrt[c - I*d]])/Sqrt[c - I*d] + (I*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/
Sqrt[c + I*d]])/Sqrt[c + I*d]))/2 + ((-1/2*(c*d*(45*a^2*A*b*d^3 - 15*A*b^3*
d^3 + 15*a^3*B*d^3 - 45*a*b^2*B*d^3 - 45*a^2*b*C*d^3 + 15*b^3*C*d^3)) + d^2
*((-48*b^3*c^3*C + 40*b^3*B*c^2*d + 144*a*b^2*c^2*C*d - 30*A*b^3*c*d^2 - 11
0*a*b^2*B*c*d^2 - 144*a^2*b*c*C*d^2 + 30*b^3*c*C*d^2 + 15*a^3*A*d^3 + 15*a*
A*b^2*d^3 + 40*a^2*b*B*d^3 + 33*a^3*C*d^3 - 15*a*b^2*C*d^3)/2 + (48*b^3*c^3
*C - 40*b^3*B*c^2*d - 144*a*b^2*c^2*C*d + 30*A*b^3*c*d^2 + 110*a*b^2*B*c*d^
2 + 144*a^2*b*c*C*d^2 - 30*b^3*c*C*d^2 - 60*a*A*b^2*d^3 - 85*a^2*b*B*d^3 +
15*b^3*B*d^3 - 48*a^3*C*d^3 + 60*a*b^2*C*d^3)/2))*(-(Hypergeometric2F1[-1/2
, 1, 1/2, (c + d*Tan[e + f*x])/(c - I*d)]/((I*c + d)*Sqrt[c + d*Tan[e + f*x
]])) + Hypergeometric2F1[-1/2, 1, 1/2, (c + d*Tan[e + f*x])/(c + I*d)]/((I*
c - d)*Sqrt[c + d*Tan[e + f*x]])))/d))/d)/(4*d*f)))/(3*d)))/(5*d)
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 11254 vs. 2(476) = 952.

Time = 0.44 (sec) , antiderivative size = 11255, normalized size of antiderivative = 22.03

method	result	size
parts	Expression too large to display	11255
derivativedivides	Expression too large to display	49725
default	Expression too large to display	49725

```
[In] int((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(3/
2),x,method=_RETURNVERBOSE)
```

```
[Out] result too large to display
```

Fricas [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^3 (A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^(3/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int \frac{(a+b\tan(e+fx))^3 (A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx = \int \frac{(a+b\tan{(e+fx)})^3 (A+B\tan{(e+}}{(c+d\tan{(e+}}$$

```
[In] integrate((a+b*tan(f*x+e))**3*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x
+e))**(3/2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**3*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/(
c + d*tan(e + f*x))**(3/2), x)
```

Maxima [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^3 (A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^(3/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

Giac [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^3 (A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^(3/2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^3 (A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx = \text{Hanged}$$

```
[In] int(((a + b*tan(e + f*x))^3*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d
*tan(e + f*x))^(3/2),x)
```

```
[Out] \text{Hanged}
```

3.117 $\int \frac{(a+b\tan(e+fx))^2(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx$

Optimal result 1647
Rubi [A] (verified) 1648
Mathematica [C] (verified) 1652
Maple [B] (verified) 1652
Fricas [F(-1)] 1653
Sympy [F] 1653
Maxima [F(-1)] 1653
Giac [F(-1)] 1654
Mupad [B] (verification not implemented) 1654

Optimal result

Integrand size = 47, antiderivative size = 343

$$\begin{aligned}
&\int \frac{(a+b\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx = \\
&-\frac{(a-ib)^2(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(c-id)^{3/2}f} \\
&-\frac{(a+ib)^2(B-i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(c+id)^{3/2}f} \\
&-\frac{2(c^2C-Bcd+Ad^2)\,(a+b\tan(e+fx))^2}{d\,(c^2+d^2)\,f\sqrt{c+d\tan(e+fx)}} \\
&+\frac{2b(6ad(2c^2C-Bcd+(A+C)d^2)-b(8c^3C-6Bc^2d+c(3A+5C)d^2-3Bd^3))\sqrt{c+d\tan(e+fx)}}{3d^3\,(c^2+d^2)\,f} \\
&+\frac{2b^2(4c^2C-3Bcd+(3A+C)d^2)\tan(e+fx)\sqrt{c+d\tan(e+fx)}}{3d^2\,(c^2+d^2)\,f}
\end{aligned}$$

```
[Out] -(a-I*b)^2*(I*A+B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(c-I*d
)^(3/2)/f-(a+I*b)^2*(B-I*(A-C))*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2
))/(c+I*d)^(3/2)/f+2/3*b*(6*a*d*(2*c^2*C-B*c*d+(A+C)*d^2)-b*(8*c^3*C-6*B*c^
2*d+c*(3*A+5*C)*d^2-3*B*d^3))*(c+d*tan(f*x+e))^(1/2)/d^3/(c^2+d^2)/f+2/3*b^
2*(4*c^2*C-3*B*c*d+(3*A+C)*d^2)*(c+d*tan(f*x+e))^(1/2)*tan(f*x+e)/d^2/(c^2+
d^2)/f-2*(A*d^2-B*c*d+C*c^2)*(a+b*tan(f*x+e))^2/d/(c^2+d^2)/f/(c+d*tan(f*x+
e))^(1/2)
```

Rubi [A] (verified)

Time = 1.52 (sec) , antiderivative size = 343, normalized size of antiderivative = 1.00, number of steps used = 10, number of rules used = 7, $\frac{\text{number of rules}}{\text{integrand size}} = 0.149$, Rules used = {3726, 3718, 3711, 3620, 3618, 65, 214}

$$\int \frac{(a+b\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx =$$

$$-\frac{(a-ib)^2(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f(c-id)^{3/2}} - \frac{(a+ib)^2(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f(c+id)^{3/2}} - \frac{2(Ad^2-Bcd+c^2C)\,(a+b\tan(e+fx))^2}{df\,(c^2+d^2)\,\sqrt{c+d\tan(e+fx)}} + \frac{2b\sqrt{c+d\tan(e+fx)}(6ad(d^2(A+C)-Bcd+2c^2C)-b(cd^2(3A+5C)-6Bc^2d-3Bd^3+8c^3C))}{3d^3f\,(c^2+d^2)} + \frac{2b^2\tan(e+fx)\,(d^2(3A+C)-3Bcd+4c^2C)\,\sqrt{c+d\tan(e+fx)}}{3d^2f\,(c^2+d^2)}$$

```
[In] Int[((a + b*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c + d
*Tan[e + f*x])^(3/2),x]
```

```
[Out] -(((a - I*b)^2*(I*A + B - I*C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*
d]])/((c - I*d)^(3/2)*f)) - ((a + I*b)^2*(B - I*(A - C))*ArcTanh[Sqrt[c + d
*Tan[e + f*x]]/Sqrt[c + I*d]])/((c + I*d)^(3/2)*f) - (2*(c^2*C - B*c*d + A*
d^2)*(a + b*Tan[e + f*x])^2)/(d*(c^2 + d^2)*f*Sqrt[c + d*Tan[e + f*x]]) + (
2*b*(6*a*d*(2*c^2*C - B*c*d + (A + C)*d^2) - b*(8*c^3*C - 6*B*c^2*d + c*(3*
A + 5*C)*d^2 - 3*B*d^3))*Sqrt[c + d*Tan[e + f*x]])/(3*d^3*(c^2 + d^2)*f) +
(2*b^2*(4*c^2*C - 3*B*c*d + (3*A + C)*d^2)*Tan[e + f*x]*Sqrt[c + d*Tan[e +
f*x]])/(3*d^2*(c^2 + d^2)*f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3726

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*d^2 + c*(c*C - B*d))*(a + b*Tan[e +
 f*x])^m*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rubi steps

$$\text{integral} = -\frac{2(c^2C - Bcd + Ad^2)(a + b\tan(e+fx))^2}{d(c^2+d^2) f\sqrt{c+d\tan(e+fx)}} + \frac{2\int \frac{(a+b\tan(e+fx))\left(\frac{1}{2}\left(Ad(ac+4bd)+2\left(2bc-\frac{ad}{2}\right)(cC-Bd)\right)+\frac{1}{2}d((A-C)(bc-ad)+B(ac+bd))\tan(e+fx)+\frac{1}{2}b(4c^2C-3Bcd+(3A+C)d^2)\tan}{\sqrt{c+d\tan(e+fx)}}}{d(c^2+d^2)}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)(a + b\tan(e+fx))^2}{d(c^2+d^2) f\sqrt{c+d\tan(e+fx)}} + \frac{2b^2(4c^2C - 3Bcd + (3A+C)d^2)\tan(e+fx)\sqrt{c+d\tan(e+fx)}}{3d^2(c^2+d^2) f} - \frac{4\int \frac{\frac{1}{4}(2b^2c(4c^2C-3Bcd+(3A+C)d^2)-3ad(Ad(ac+4bd)+(4bc-ad)(cC-Bd)))-\frac{3}{4}d^2(2ab(Ac-cC+Bd)+a^2(Bc-(A-C)d)-b^2(Bc}{\sqrt{c+d\tan(e+fx}}}{3d^2(c^2+d^2)}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)(a + b\tan(e+fx))^2}{d(c^2+d^2) f\sqrt{c+d\tan(e+fx)}} + \frac{2b(6ad(2c^2C - Bcd + (A+C)d^2) - b(8c^3C - 6Bc^2d + c(3A+5C)d^2 - 3Bd^3))\sqrt{c+d\tan(e+}}{3d^3(c^2+d^2) f} + \frac{2b^2(4c^2C - 3Bcd + (3A+C)d^2)\tan(e+fx)\sqrt{c+d\tan(e+fx)}}{3d^2(c^2+d^2) f} - \frac{4\int \frac{-\frac{3}{4}d^2(a^2(Ac-cC+Bd)-b^2(Ac-cC+Bd)-2ab(Bc-(A-C)d))-\frac{3}{4}d^2(2ab(Ac-cC+Bd)+a^2(Bc-(A-C)d)-b^2(Bc-(A-C)d))}{\sqrt{c+d\tan(e+fx)}}}{3d^2(c^2+d^2)}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)(a + b\tan(e+fx))^2}{d(c^2+d^2) f\sqrt{c+d\tan(e+fx)}} + \frac{2b(6ad(2c^2C - Bcd + (A+C)d^2) - b(8c^3C - 6Bc^2d + c(3A+5C)d^2 - 3Bd^3))\sqrt{c+d\tan(e+}}{3d^3(c^2+d^2) f} + \frac{2b^2(4c^2C - 3Bcd + (3A+C)d^2)\tan(e+fx)\sqrt{c+d\tan(e+fx)}}{3d^2(c^2+d^2) f} + \frac{((a-ib)^2(A-iB-C))\int \frac{1+i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(c-id)} + \frac{((a+ib)^2(A+iB-C))\int \frac{1-i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(c+id)}$$

$$
\begin{aligned}
&= -\frac{2(c^2C - Bcd + Ad^2)\,(a + b\tan(e + fx))^2}{d\,(c^2 + d^2)\,f\sqrt{c + d\tan(e + fx)}} \\
&\quad + \frac{2b(6ad(2c^2C - Bcd + (A + C)d^2) - b(8c^3C - 6Bc^2d + c(3A + 5C)d^2 - 3Bd^3))\,\sqrt{c + d\tan(e}{3d^3\,(c^2 + d^2)\,f} \\
&\quad + \frac{2b^2(4c^2C - 3Bcd + (3A + C)d^2)\tan(e + fx)\sqrt{c + d\tan(e + fx)}}{3d^2\,(c^2 + d^2)\,f} \\
&\quad + \frac{((a - ib)^2(iA + B - iC))\,\mathrm{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c-idx}}\,dx, x, i\tan(e + fx)\right)}{2(c - id)f} \\
&\quad - \frac{(i(a + ib)^2(A + iB - C))\,\mathrm{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c+idx}}\,dx, x, -i\tan(e + fx)\right)}{2(c + id)f} \\
&= -\frac{2(c^2C - Bcd + Ad^2)\,(a + b\tan(e + fx))^2}{d\,(c^2 + d^2)\,f\sqrt{c + d\tan(e + fx)}} \\
&\quad + \frac{2b(6ad(2c^2C - Bcd + (A + C)d^2) - b(8c^3C - 6Bc^2d + c(3A + 5C)d^2 - 3Bd^3))\,\sqrt{c + d\tan(e}{3d^3\,(c^2 + d^2)\,f} \\
&\quad + \frac{2b^2(4c^2C - 3Bcd + (3A + C)d^2)\tan(e + fx)\sqrt{c + d\tan(e + fx)}}{3d^2\,(c^2 + d^2)\,f} \\
&\quad - \frac{((a - ib)^2(A - iB - C))\,\mathrm{Subst}\left(\int \frac{1}{-1-\frac{ic}{d}+\frac{ix^2}{d}}\,dx, x, \sqrt{c + d\tan(e + fx)}\right)}{(c - id)df} \\
&\quad - \frac{((a + ib)^2(A + iB - C))\,\mathrm{Subst}\left(\int \frac{1}{-1+\frac{ic}{d}-\frac{ix^2}{d}}\,dx, x, \sqrt{c + d\tan(e + fx)}\right)}{(c + id)df} \\
&= -\frac{(a - ib)^2(iA + B - iC)\,\mathrm{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(c - id)^{3/2}f} \\
&\quad - \frac{(a + ib)^2(B - i(A - C))\,\mathrm{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(c + id)^{3/2}f} \\
&\quad - \frac{2(c^2C - Bcd + Ad^2)\,(a + b\tan(e + fx))^2}{d\,(c^2 + d^2)\,f\sqrt{c + d\tan(e + fx)}} \\
&\quad + \frac{2b(6ad(2c^2C - Bcd + (A + C)d^2) - b(8c^3C - 6Bc^2d + c(3A + 5C)d^2 - 3Bd^3))\,\sqrt{c + d\tan(e}{3d^3\,(c^2 + d^2)\,f} \\
&\quad + \frac{2b^2(4c^2C - 3Bcd + (3A + C)d^2)\tan(e + fx)\sqrt{c + d\tan(e + fx)}}{3d^2\,(c^2 + d^2)\,f}
\end{aligned}
$$

Mathematica [C] (verified)

Result contains higher order function than in optimal. Order 5 vs. order 3 in optimal.

Time = 6.57 (sec) , antiderivative size = 476, normalized size of antiderivative = 1.39

$$\int \frac{(a+b\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{3/2}}\,dx=\frac{2C(a+b\tan(e+fx))^2}{3df\sqrt{c+d\tan(e+fx)}}$$

$$+\frac{2\left(\frac{(-4bcC+3bBd+4aCd)(a+b\tan(e+fx))}{df\sqrt{c+d\tan(e+fx)}}+\frac{-\frac{2\left(8b^2c^2C-6b^2Bcd-16abcCd+3Ab^2d^2+9abBd^2+8a^2Cd^2-3b^2Cd^2\right)}{d\sqrt{c+d\tan(e+fx)}}+\frac{2\left(\frac{3}{2}\left(a^2B-b^2B+2ab(A-C)\right)d^2\left(-\cdots\right.\right.}{\cdots}}{\cdots}\right.}{\cdots}$$

```
[In] Integrate[((a + b*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(c + d*Tan[e + f*x])^(3/2),x]
```

```
[Out] (2*C*(a + b*Tan[e + f*x])^2)/(3*d*f*Sqrt[c + d*Tan[e + f*x]]) + (2*(((-4*b*
c*C + 3*b*B*d + 4*a*C*d)*(a + b*Tan[e + f*x]))/(d*f*Sqrt[c + d*Tan[e + f*x]
]) + ((-2*(8*b^2*c^2*C - 6*b^2*B*c*d - 16*a*b*c*C*d + 3*A*b^2*d^2 + 9*a*b*B
*d^2 + 8*a^2*C*d^2 - 3*b^2*C*d^2))/(d*Sqrt[c + d*Tan[e + f*x]]) + (2*((3*(a
^2*B - b^2*B + 2*a*b*(A - C))*d^2*(((-I)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/S
qrt[c - I*d]])/Sqrt[c - I*d] + (I*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c +
 I*d]])/Sqrt[c + I*d]))/2 + (((-3*c*(a^2*B - b^2*B + 2*a*b*(A - C))*d^3)/2
- (3*(2*a*b*B - a^2*(A - C) + b^2*(A - C))*d^4)/2)*(-(Hypergeometric2F1[-1/
2, 1, 1/2, (c + d*Tan[e + f*x])/(c - I*d)]/((I*c + d)*Sqrt[c + d*Tan[e + f*
x]])) + Hypergeometric2F1[-1/2, 1, 1/2, (c + d*Tan[e + f*x])/(c + I*d)]/((I
*c - d)*Sqrt[c + d*Tan[e + f*x]])))/d))/d)/(2*d*f)))/(3*d)
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 9398 vs. $2(312) = 624$.

Time = 0.22 (sec) , antiderivative size = 9399, normalized size of antiderivative = 27.40

method	result	size
parts	Expression too large to display	9399
derivativedivides	Expression too large to display	36710
default	Expression too large to display	36710

```
[In] int((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(3/
2),x,method=_RETURNVERBOSE)
```

```
[Out] result too large to display
```

Fricas [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^(3/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int \frac{(a+b\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{3/2}}\,dx = \int \frac{(a+b\tan(e+fx))^2\left(A+B\tan(e+\right.}{(c+d\tan(e+}$$

```
[In] integrate((a+b*tan(f*x+e))**2*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x
+e))**(3/2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**2*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/(
c + d*tan(e + f*x))**(3/2), x)
```

Maxima [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^(3/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

Giac [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^(3/2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [B] (verification not implemented)

Time = 63.78 (sec) , antiderivative size = 54886, normalized size of antiderivative = 160.02

$$\int \frac{(a+b\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{3/2}}\,dx = \text{Too large to display}$$

```
[In] int(((a + b*tan(e + f*x))^2*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d
*tan(e + f*x))^(3/2),x)
```

```
[Out] (2*(B*b^2*c^3 + B*a^2*c*d^2 - 2*B*a*b*c^2*d))/(d^2*f*(c^2 + d^2)*(c + d*tan
(e + f*x))^(1/2)) - atan((((-(((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*
B^2*a^2*b^2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*
a^4*c*d^2*f^2 - 24*B^2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*
b*c^2*d*f^2 + 144*B^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 4
8*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*
a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) - 4*B^2*a^4*c^3*f^2 - 4*B^2*b^4*c^3*f^2 + 2
4*B^2*a^2*b^2*c^3*f^2 - 16*B^2*a*b^3*d^3*f^2 + 16*B^2*a^3*b*d^3*f^2 + 12*B^
2*a^4*c*d^2*f^2 + 12*B^2*b^4*c*d^2*f^2 + 48*B^2*a*b^3*c^2*d*f^2 - 48*B^2*a^
3*b*c^2*d*f^2 - 72*B^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^
4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^4*c
^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^2
- 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*B^2*b^4*c*d^2*f^2 - 96*B
^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B^2*a^2*b^2*c*d^2*f^2)^2/
4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(B^4*a^8 +
B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) - 4*B^2*a^4
*c^3*f^2 - 4*B^2*b^4*c^3*f^2 + 24*B^2*a^2*b^2*c^3*f^2 - 16*B^2*a*b^3*d^3*f^
2 + 16*B^2*a^3*b*d^3*f^2 + 12*B^2*a^4*c*d^2*f^2 + 12*B^2*b^4*c*d^2*f^2 + 48
*B^2*a*b^3*c^2*d*f^2 - 48*B^2*a^3*b*c^2*d*f^2 - 72*B^2*a^2*b^2*c*d^2*f^2)/(
16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f
^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5
 + 64*c^11*d^2*f^5) - 32*B*a^2*d^12*f^4 + 32*B*b^2*d^12*f^4 - 96*B*a^2*c^2*
d^10*f^4 - 64*B*a^2*c^4*d^8*f^4 + 64*B*a^2*c^6*d^6*f^4 + 96*B*a^2*c^8*d^4*f
```

^4 + 32*B*a^2*c^10*d^2*f^4 + 96*B*b^2*c^2*d^10*f^4 + 64*B*b^2*c^4*d^8*f^4 -
 64*B*b^2*c^6*d^6*f^4 - 96*B*b^2*c^8*d^4*f^4 - 32*B*b^2*c^10*d^2*f^4 + 128*
B*a*b*c*d^11*f^4 + 512*B*a*b*c^3*d^9*f^4 + 768*B*a*b*c^5*d^7*f^4 + 512*B*a*
b*c^7*d^5*f^4 + 128*B*a*b*c^9*d^3*f^4) + (c + d*tan(e + f*x))^(1/2)*(16*B^2
*a^4*d^10*f^3 + 16*B^2*b^4*d^10*f^3 - 96*B^2*a^2*b^2*d^10*f^3 + 32*B^2*a^4*
c^2*d^8*f^3 - 32*B^2*a^4*c^6*d^4*f^3 - 16*B^2*a^4*c^8*d^2*f^3 + 32*B^2*b^4*
c^2*d^8*f^3 - 32*B^2*b^4*c^6*d^4*f^3 - 16*B^2*b^4*c^8*d^2*f^3 + 128*B^2*a*b
^3*c*d^9*f^3 - 128*B^2*a^3*b*c*d^9*f^3 + 384*B^2*a*b^3*c^3*d^7*f^3 + 384*B^
2*a*b^3*c^5*d^5*f^3 + 128*B^2*a*b^3*c^7*d^3*f^3 - 384*B^2*a^3*b*c^3*d^7*f^3
 - 384*B^2*a^3*b*c^5*d^5*f^3 - 128*B^2*a^3*b*c^7*d^3*f^3 - 192*B^2*a^2*b^2*
c^2*d^8*f^3 + 192*B^2*a^2*b^2*c^6*d^4*f^3 + 96*B^2*a^2*b^2*c^8*d^2*f^3))*(-
(((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2 + 32*B^2*
a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*B^2*b^4*c*
d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B^2*a^2*b^2
*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^
4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/
2) - 4*B^2*a^4*c^3*f^2 - 4*B^2*b^4*c^3*f^2 + 24*B^2*a^2*b^2*c^3*f^2 - 16*B^
2*a*b^3*d^3*f^2 + 16*B^2*a^3*b*d^3*f^2 + 12*B^2*a^4*c*d^2*f^2 + 12*B^2*b^4*
c*d^2*f^2 + 48*B^2*a*b^3*c^2*d*f^2 - 48*B^2*a^3*b*c^2*d*f^2 - 72*B^2*a^2*b^
2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2
)*1i - ((-(((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2
 + 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*
B^2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B
^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*
c^4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6
*b^2))^(1/2) - 4*B^2*a^4*c^3*f^2 - 4*B^2*b^4*c^3*f^2 + 24*B^2*a^2*b^2*c^3*f
^2 - 16*B^2*a*b^3*d^3*f^2 + 16*B^2*a^3*b*d^3*f^2 + 12*B^2*a^4*c*d^2*f^2 + 1
2*B^2*b^4*c*d^2*f^2 + 48*B^2*a*b^3*c^2*d*f^2 - 48*B^2*a^3*b*c^2*d*f^2 - 72*
B^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f
^4)))^(1/2)*(32*B*b^2*d^12*f^4 - 32*B*a^2*d^12*f^4 - (c + d*tan(e + f*x))^(
1/2)*(-(((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2 +
32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*B^2
*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B^2*
a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4
*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^
2))^(1/2) - 4*B^2*a^4*c^3*f^2 - 4*B^2*b^4*c^3*f^2 + 24*B^2*a^2*b^2*c^3*f^2
- 16*B^2*a*b^3*d^3*f^2 + 16*B^2*a^3*b*d^3*f^2 + 12*B^2*a^4*c*d^2*f^2 + 12*B
^2*b^4*c*d^2*f^2 + 48*B^2*a*b^3*c^2*d*f^2 - 48*B^2*a^3*b*c^2*d*f^2 - 72*B^2
*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)
))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*
f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 96*B*a^2*c^2*d^10*f^4 - 64*B*a^2
*c^4*d^8*f^4 + 64*B*a^2*c^6*d^6*f^4 + 96*B*a^2*c^8*d^4*f^4 + 32*B*a^2*c^10*
d^2*f^4 + 96*B*b^2*c^2*d^10*f^4 + 64*B*b^2*c^4*d^8*f^4 - 64*B*b^2*c^6*d^6*f
^4 - 96*B*b^2*c^8*d^4*f^4 - 32*B*b^2*c^10*d^2*f^4 + 128*B*a*b*c*d^11*f^4 +
512*B*a*b*c^3*d^9*f^4 + 768*B*a*b*c^5*d^7*f^4 + 512*B*a*b*c^7*d^5*f^4 + 128

*B*a*b*c^9*d^3*f^4) - (c + d*tan(e + f*x))^(1/2)*(16*B^2*a^4*d^10*f^3 + 16*B^2*b^4*d^10*f^3 - 96*B^2*a^2*b^2*d^10*f^3 + 32*B^2*a^4*c^2*d^8*f^3 - 32*B^2*a^4*c^6*d^4*f^3 - 16*B^2*a^4*c^8*d^2*f^3 + 32*B^2*b^4*c^2*d^8*f^3 - 32*B^2*b^4*c^6*d^4*f^3 - 16*B^2*b^4*c^8*d^2*f^3 + 128*B^2*a*b^3*c*d^9*f^3 - 128*B^2*a^3*b*c*d^9*f^3 + 384*B^2*a*b^3*c^3*d^7*f^3 + 384*B^2*a*b^3*c^5*d^5*f^3 + 128*B^2*a*b^3*c^7*d^3*f^3 - 384*B^2*a^3*b*c^3*d^7*f^3 - 384*B^2*a^3*b*c^5*d^5*f^3 - 128*B^2*a^3*b*c^7*d^3*f^3 - 192*B^2*a^2*b^2*c^2*d^8*f^3 + 192*B^2*a^2*b^2*c^6*d^4*f^3 + 96*B^2*a^2*b^2*c^8*d^2*f^3))*(-(((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*B^2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) - 4*B^2*a^4*c^3*f^2 - 4*B^2*b^4*c^3*f^2 + 24*B^2*a^2*b^2*c^3*f^2 - 16*B^2*a*b^3*d^3*f^2 + 16*B^2*a^3*b*d^3*f^2 + 12*B^2*a^4*c*d^2*f^2 + 12*B^2*b^4*c*d^2*f^2 + 48*B^2*a*b^3*c^2*d*f^2 - 48*B^2*a^3*b*c^2*d*f^2 - 72*B^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*1i)/(64*B^3*a^3*b^3*d^9*f^2 - ((-(((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*B^2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) - 4*B^2*a^4*c^3*f^2 - 4*B^2*b^4*c^3*f^2 + 24*B^2*a^2*b^2*c^3*f^2 - 16*B^2*a*b^3*d^3*f^2 + 16*B^2*a^3*b*d^3*f^2 + 12*B^2*a^4*c*d^2*f^2 + 12*B^2*b^4*c*d^2*f^2 + 48*B^2*a*b^3*c^2*d*f^2 - 48*B^2*a^3*b*c^2*d*f^2 - 72*B^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(32*B*b^2*d^12*f^4 - 32*B*a^2*d^12*f^4 - (c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*B^2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) - 4*B^2*a^4*c^3*f^2 - 4*B^2*b^4*c^3*f^2 + 24*B^2*a^2*b^2*c^3*f^2 - 16*B^2*a*b^3*d^3*f^2 + 16*B^2*a^3*b*d^3*f^2 + 12*B^2*a^4*c*d^2*f^2 + 12*B^2*b^4*c*d^2*f^2 + 48*B^2*a*b^3*c^2*d*f^2 - 48*B^2*a^3*b*c^2*d*f^2 - 72*B^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 96*B*a^2*c^2*d^10*f^4 - 64*B*a^2*c^4*d^8*f^4 + 64*B*a^2*c^6*d^6*f^4 + 96*B*a^2*c^8*d^4*f^4 + 32*B*a^2*c^10*d^2*f^4 + 96*B*b^2*c^2*d^10*f^4 + 64*B*b^2*c^4*d^8*f^4 - 64*B*b^2*c^6*d^6*f^4 - 96*B*b^2*c^8*d^4*f^4 - 32*B*b^2*c^10*d^2*f^4 + 128*B*a*b*c*d^11*f^4 + 512*B*a*b*c^3*d^9*f^4 + 768*B*a*b*c^5*d^7*f^4 + 512*B*a*b*c^7*d^5*f^4 + 128*B*a*b*c^9*d^3*f^4) - (c + d*tan(e + f*x))^(1/2)*(16*B^2*a^4*d^10*f^3 + 16*B^2*b^4*d^10*f^3 - 96*B^2*a^2*b^2*d^10*f^3 + 32*B^2*a^4*c^2*d^8*f^3 - 32*B^2*a^4*c^6*d^4*f^3 - 16*B^2*a^4*c^8*d^2*f^3 + 32*B^2*b^4*c^2*d^8*f^3 -

32*B^2*b^4*c^6*d^4*f^3 - 16*B^2*b^4*c^8*d^2*f^3 + 128*B^2*a*b^3*c*d^9*f^3 - 128*B^2*a^3*b*c*d^9*f^3 + 384*B^2*a*b^3*c^3*d^7*f^3 + 384*B^2*a*b^3*c^5*d^5*f^3 + 128*B^2*a*b^3*c^7*d^3*f^3 - 384*B^2*a^3*b*c^3*d^7*f^3 - 384*B^2*a^3*b*c^5*d^5*f^3 - 128*B^2*a^3*b*c^7*d^3*f^3 - 192*B^2*a^2*b^2*c^2*d^8*f^3 + 192*B^2*a^2*b^2*c^6*d^4*f^3 + 96*B^2*a^2*b^2*c^8*d^2*f^3))*(-(((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*B^2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) - 4*B^2*a^4*c^3*f^2 - 4*B^2*b^4*c^3*f^2 + 24*B^2*a^2*b^2*c^3*f^2 - 16*B^2*a*b^3*d^3*f^2 + 16*B^2*a^3*b*d^3*f^2 + 12*B^2*a^4*c*d^2*f^2 + 12*B^2*b^4*c*d^2*f^2 + 48*B^2*a*b^3*c^2*d*f^2 - 48*B^2*a^3*b*c^2*d*f^2 - 72*B^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2) - ((-(((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*B^2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) - 4*B^2*a^4*c^3*f^2 - 4*B^2*b^4*c^3*f^2 + 24*B^2*a^2*b^2*c^3*f^2 - 16*B^2*a*b^3*d^3*f^2 + 16*B^2*a^3*b*d^3*f^2 + 12*B^2*a^4*c*d^2*f^2 + 12*B^2*b^4*c*d^2*f^2 + 48*B^2*a*b^3*c^2*d*f^2 - 48*B^2*a^3*b*c^2*d*f^2 - 72*B^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*B^2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) - 4*B^2*a^4*c^3*f^2 - 4*B^2*b^4*c^3*f^2 + 24*B^2*a^2*b^2*c^3*f^2 - 16*B^2*a*b^3*d^3*f^2 + 16*B^2*a^3*b*d^3*f^2 + 12*B^2*a^4*c*d^2*f^2 + 12*B^2*b^4*c*d^2*f^2 + 48*B^2*a*b^3*c^2*d*f^2 - 48*B^2*a^3*b*c^2*d*f^2 - 72*B^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 32*B*a^2*d^12*f^4 + 32*B*b^2*d^12*f^4 - 96*B*a^2*c^2*d^10*f^4 - 64*B*a^2*c^4*d^8*f^4 + 64*B*a^2*c^6*d^6*f^4 + 96*B*a^2*c^8*d^4*f^4 + 32*B*a^2*c^10*d^2*f^4 + 96*B*b^2*c^2*d^10*f^4 + 64*B*b^2*c^4*d^8*f^4 - 64*B*b^2*c^6*d^6*f^4 - 96*B*b^2*c^8*d^4*f^4 - 32*B*b^2*c^10*d^2*f^4 + 128*B*a*b*c*d^11*f^4 + 512*B*a*b*c^3*d^9*f^4 + 768*B*a*b*c^5*d^7*f^4 + 512*B*a*b*c^7*d^5*f^4 + 128*B*a*b*c^9*d^3*f^4) + (c + d*tan(e + f*x))^(1/2)*(16*B^2*a^4*d^10*f^3 + 16*B^2*b^4*d^10*f^3 - 96*B^2*a^2*b^2*d^10*f^3 + 32*B^2*a^4*c^2*d^8*f^3 - 32*B^2*a^4*c^6*d^4*f^3 - 16*B^2*a^4*c^8*d^2*f^3 + 32*B^2*b^4*c^2*d^8*f^3 - 32*B^2*b^4*c^6*d^4*f^3 - 16*B^2*b^4*c^8*d^2*f^3 + 128*B^2*a*b^3*c*d^9*f^3 - 128*B^2*a^3*b*c*d^9*f^3 + 384*B^2*a*b^3*c^3*d^7*f^3 + 384*B^2*a*b^3*c^5*d^5*f^3 + 128*B^2*a*b^3*c^7*d^3*f^3 - 384*B^2*a^3*b*c^3*d^7*f^3 - 384*B^2*a^3*b*c^5*d^5*f^3 - 128*

B^2*a^3*b*c^7*d^3*f^3 - 192*B^2*a^2*b^2*c^2*d^8*f^3 + 192*B^2*a^2*b^2*c^6*d
^4*f^3 + 96*B^2*a^2*b^2*c^8*d^2*f^3))*(-(((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^
3*f^2 - 48*B^2*a^2*b^2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^
2 - 24*B^2*a^4*c*d^2*f^2 - 24*B^2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 +
96*B^2*a^3*b*c^2*d*f^2 + 144*B^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*
d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b
^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) - 4*B^2*a^4*c^3*f^2 - 4*B^2*b^4*
c^3*f^2 + 24*B^2*a^2*b^2*c^3*f^2 - 16*B^2*a*b^3*d^3*f^2 + 16*B^2*a^3*b*d^3*
f^2 + 12*B^2*a^4*c*d^2*f^2 + 12*B^2*b^4*c*d^2*f^2 + 48*B^2*a*b^3*c^2*d*f^2
- 48*B^2*a^3*b*c^2*d*f^2 - 72*B^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4
 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2) + 48*B^3*a^6*c^3*d^6*f^2 + 48*B^3
*a^6*c^5*d^4*f^2 + 16*B^3*a^6*c^7*d^2*f^2 - 48*B^3*b^6*c^3*d^6*f^2 - 48*B^3
*b^6*c^5*d^4*f^2 - 16*B^3*b^6*c^7*d^2*f^2 + 32*B^3*a*b^5*d^9*f^2 + 32*B^3*a
^5*b*d^9*f^2 + 16*B^3*a^6*c*d^8*f^2 - 16*B^3*b^6*c*d^8*f^2 + 96*B^3*a*b^5*c
^2*d^7*f^2 + 96*B^3*a*b^5*c^4*d^5*f^2 + 32*B^3*a*b^5*c^6*d^3*f^2 - 16*B^3*a
^2*b^4*c*d^8*f^2 + 16*B^3*a^4*b^2*c*d^8*f^2 + 96*B^3*a^5*b*c^2*d^7*f^2 + 96
*B^3*a^5*b*c^4*d^5*f^2 + 32*B^3*a^5*b*c^6*d^3*f^2 - 48*B^3*a^2*b^4*c^3*d^6*
f^2 - 48*B^3*a^2*b^4*c^5*d^4*f^2 - 16*B^3*a^2*b^4*c^7*d^2*f^2 + 192*B^3*a^3
*b^3*c^2*d^7*f^2 + 192*B^3*a^3*b^3*c^4*d^5*f^2 + 64*B^3*a^3*b^3*c^6*d^3*f^2
 + 48*B^3*a^4*b^2*c^3*d^6*f^2 + 48*B^3*a^4*b^2*c^5*d^4*f^2 + 16*B^3*a^4*b^2
*c^7*d^2*f^2))*(-(((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*
c^3*f^2 + 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^
2 - 24*B^2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2
+ 144*B^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^
4 + 48*c^4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*
B^4*a^6*b^2))^(1/2) - 4*B^2*a^4*c^3*f^2 - 4*B^2*b^4*c^3*f^2 + 24*B^2*a^2*b^
2*c^3*f^2 - 16*B^2*a*b^3*d^3*f^2 + 16*B^2*a^3*b*d^3*f^2 + 12*B^2*a^4*c*d^2*
f^2 + 12*B^2*b^4*c*d^2*f^2 + 48*B^2*a*b^3*c^2*d*f^2 - 48*B^2*a^3*b*c^2*d*f^
2 - 72*B^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^
4*d^2*f^4)))^(1/2)*2i - atan((((((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 -
48*B^2*a^2*b^2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B
^2*a^4*c*d^2*f^2 - 24*B^2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a
^3*b*c^2*d*f^2 + 144*B^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4
+ 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B
^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) + 4*B^2*a^4*c^3*f^2 + 4*B^2*b^4*c^3*f^2
- 24*B^2*a^2*b^2*c^3*f^2 + 16*B^2*a*b^3*d^3*f^2 - 16*B^2*a^3*b*d^3*f^2 - 12
*B^2*a^4*c*d^2*f^2 - 12*B^2*b^4*c*d^2*f^2 - 48*B^2*a*b^3*c^2*d*f^2 + 48*B^2
*a^3*b*c^2*d*f^2 + 72*B^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2
*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^4
*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^
2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*B^2*b^4*c*d^2*f^2 - 96
*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B^2*a^2*b^2*c*d^2*f^2)^
2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(B^4*a^8
+ B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) + 4*B^2*a
^4*c^3*f^2 + 4*B^2*b^4*c^3*f^2 - 24*B^2*a^2*b^2*c^3*f^2 + 16*B^2*a*b^3*d^3*

f^2 - 16*B^2*a^3*b*d^3*f^2 - 12*B^2*a^4*c*d^2*f^2 - 12*B^2*b^4*c*d^2*f^2 - 48*B^2*a*b^3*c^2*d*f^2 + 48*B^2*a^3*b*c^2*d*f^2 + 72*B^2*a^2*b^2*c*d^2*f^2) /(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12 *f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f ^5 + 64*c^11*d^2*f^5) - 32*B*a^2*d^12*f^4 + 32*B*b^2*d^12*f^4 - 96*B*a^2*c^ 2*d^10*f^4 - 64*B*a^2*c^4*d^8*f^4 + 64*B*a^2*c^6*d^6*f^4 + 96*B*a^2*c^8*d^4 *f^4 + 32*B*a^2*c^10*d^2*f^4 + 96*B*b^2*c^2*d^10*f^4 + 64*B*b^2*c^4*d^8*f^4 - 64*B*b^2*c^6*d^6*f^4 - 96*B*b^2*c^8*d^4*f^4 - 32*B*b^2*c^10*d^2*f^4 + 12 8*B*a*b*c*d^11*f^4 + 512*B*a*b*c^3*d^9*f^4 + 768*B*a*b*c^5*d^7*f^4 + 512*B* a*b*c^7*d^5*f^4 + 128*B*a*b*c^9*d^3*f^4) + (c + d*tan(e + f*x))^(1/2)*(16*B ^2*a^4*d^10*f^3 + 16*B^2*b^4*d^10*f^3 - 96*B^2*a^2*b^2*d^10*f^3 + 32*B^2*a^ 4*c^2*d^8*f^3 - 32*B^2*a^4*c^6*d^4*f^3 - 16*B^2*a^4*c^8*d^2*f^3 + 32*B^2*b^ 4*c^2*d^8*f^3 - 32*B^2*b^4*c^6*d^4*f^3 - 16*B^2*b^4*c^8*d^2*f^3 + 128*B^2*a *b^3*c*d^9*f^3 - 128*B^2*a^3*b*c*d^9*f^3 + 384*B^2*a*b^3*c^3*d^7*f^3 + 384* B^2*a*b^3*c^5*d^5*f^3 + 128*B^2*a*b^3*c^7*d^3*f^3 - 384*B^2*a^3*b*c^3*d^7*f ^3 - 384*B^2*a^3*b*c^5*d^5*f^3 - 128*B^2*a^3*b*c^7*d^3*f^3 - 192*B^2*a^2*b^ 2*c^2*d^8*f^3 + 192*B^2*a^2*b^2*c^6*d^4*f^3 + 96*B^2*a^2*b^2*c^8*d^2*f^3))* ((((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2 + 32*B^2 *a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*B^2*b^4*c *d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B^2*a^2*b^ 2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f ^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1 /2) + 4*B^2*a^4*c^3*f^2 + 4*B^2*b^4*c^3*f^2 - 24*B^2*a^2*b^2*c^3*f^2 + 16*B ^2*a*b^3*d^3*f^2 - 16*B^2*a^3*b*d^3*f^2 - 12*B^2*a^4*c*d^2*f^2 - 12*B^2*b^4 *c*d^2*f^2 - 48*B^2*a*b^3*c^2*d*f^2 + 48*B^2*a^3*b*c^2*d*f^2 + 72*B^2*a^2*b ^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/ 2)*1i - (((((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24* B^2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B ^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48* c^4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6 *b^2))^(1/2) + 4*B^2*a^4*c^3*f^2 + 4*B^2*b^4*c^3*f^2 - 24*B^2*a^2*b^2*c^3*f ^2 + 16*B^2*a*b^3*d^3*f^2 - 16*B^2*a^3*b*d^3*f^2 - 12*B^2*a^4*c*d^2*f^2 - 1 2*B^2*b^4*c*d^2*f^2 - 48*B^2*a*b^3*c^2*d*f^2 + 48*B^2*a^3*b*c^2*d*f^2 + 72* B^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f ^4)))^(1/2)*(32*B*b^2*d^12*f^4 - 32*B*a^2*d^12*f^4 - (c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2 + 3 2*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*B^2* b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B^2*a ^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4* d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) + 4*B^2*a^4*c^3*f^2 + 4*B^2*b^4*c^3*f^2 - 24*B^2*a^2*b^2*c^3*f^2 + 16*B^2*a*b^3*d^3*f^2 - 16*B^2*a^3*b*d^3*f^2 - 12*B^2*a^4*c*d^2*f^2 - 12*B^ 2*b^4*c*d^2*f^2 - 48*B^2*a*b^3*c^2*d*f^2 + 48*B^2*a^3*b*c^2*d*f^2 + 72*B^2* a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))

)^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f
^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 96*B*a^2*c^2*d^10*f^4 - 64*B*a^2*
c^4*d^8*f^4 + 64*B*a^2*c^6*d^6*f^4 + 96*B*a^2*c^8*d^4*f^4 + 32*B*a^2*c^10*d
^2*f^4 + 96*B*b^2*c^2*d^10*f^4 + 64*B*b^2*c^4*d^8*f^4 - 64*B*b^2*c^6*d^6*f^
4 - 96*B*b^2*c^8*d^4*f^4 - 32*B*b^2*c^10*d^2*f^4 + 128*B*a*b*c*d^11*f^4 + 5
12*B*a*b*c^3*d^9*f^4 + 768*B*a*b*c^5*d^7*f^4 + 512*B*a*b*c^7*d^5*f^4 + 128*
B*a*b*c^9*d^3*f^4) - (c + d*tan(e + f*x))^(1/2)*(16*B^2*a^4*d^10*f^3 + 16*B
^2*b^4*d^10*f^3 - 96*B^2*a^2*b^2*d^10*f^3 + 32*B^2*a^4*c^2*d^8*f^3 - 32*B^2
*a^4*c^6*d^4*f^3 - 16*B^2*a^4*c^8*d^2*f^3 + 32*B^2*b^4*c^2*d^8*f^3 - 32*B^2
*b^4*c^6*d^4*f^3 - 16*B^2*b^4*c^8*d^2*f^3 + 128*B^2*a*b^3*c*d^9*f^3 - 128*B
^2*a^3*b*c*d^9*f^3 + 384*B^2*a*b^3*c^3*d^7*f^3 + 384*B^2*a*b^3*c^5*d^5*f^3
+ 128*B^2*a*b^3*c^7*d^3*f^3 - 384*B^2*a^3*b*c^3*d^7*f^3 - 384*B^2*a^3*b*c^5
*d^5*f^3 - 128*B^2*a^3*b*c^7*d^3*f^3 - 192*B^2*a^2*b^2*c^2*d^8*f^3 + 192*B^
2*a^2*b^2*c^6*d^4*f^3 + 96*B^2*a^2*b^2*c^8*d^2*f^3))*((((8*B^2*a^4*c^3*f^2
+ 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^2 - 32*B^
2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*B^2*b^4*c*d^2*f^2 - 96*B^2*a*b^
3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B^2*a^2*b^2*c*d^2*f^2)^2/4 - (16
*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(B^4*a^8 + B^4*b^8
 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) + 4*B^2*a^4*c^3*f^
2 + 4*B^2*b^4*c^3*f^2 - 24*B^2*a^2*b^2*c^3*f^2 + 16*B^2*a*b^3*d^3*f^2 - 16*
B^2*a^3*b*d^3*f^2 - 12*B^2*a^4*c*d^2*f^2 - 12*B^2*b^4*c*d^2*f^2 - 48*B^2*a*
b^3*c^2*d*f^2 + 48*B^2*a^3*b*c^2*d*f^2 + 72*B^2*a^2*b^2*c*d^2*f^2)/(16*(c^6
*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*1i)/(64*B^3*a^3*b^3
*d^9*f^2 - (((((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*
f^2 + 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 -
24*B^2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 14
4*B^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 +
48*c^4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*
a^6*b^2))^(1/2) + 4*B^2*a^4*c^3*f^2 + 4*B^2*b^4*c^3*f^2 - 24*B^2*a^2*b^2*c^
3*f^2 + 16*B^2*a*b^3*d^3*f^2 - 16*B^2*a^3*b*d^3*f^2 - 12*B^2*a^4*c*d^2*f^2
- 12*B^2*b^4*c*d^2*f^2 - 48*B^2*a*b^3*c^2*d*f^2 + 48*B^2*a^3*b*c^2*d*f^2 +
72*B^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^
2*f^4)))^(1/2)*(32*B*b^2*d^12*f^4 - 32*B*a^2*d^12*f^4 - (c + d*tan(e + f*x)
)^(1/2)*((((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2
+ 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*B
^2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B^
2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c
^4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*
b^2))^(1/2) + 4*B^2*a^4*c^3*f^2 + 4*B^2*b^4*c^3*f^2 - 24*B^2*a^2*b^2*c^3*f^
2 + 16*B^2*a*b^3*d^3*f^2 - 16*B^2*a^3*b*d^3*f^2 - 12*B^2*a^4*c*d^2*f^2 - 12
*B^2*b^4*c*d^2*f^2 - 48*B^2*a*b^3*c^2*d*f^2 + 48*B^2*a^3*b*c^2*d*f^2 + 72*B
^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^
4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^
6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 96*B*a^2*c^2*d^10*f^4 - 64*B*a
^2*c^4*d^8*f^4 + 64*B*a^2*c^6*d^6*f^4 + 96*B*a^2*c^8*d^4*f^4 + 32*B*a^2*c^1

0*d^2*f^4 + 96*B*b^2*c^2*d^10*f^4 + 64*B*b^2*c^4*d^8*f^4 - 64*B*b^2*c^6*d^6
*f^4 - 96*B*b^2*c^8*d^4*f^4 - 32*B*b^2*c^10*d^2*f^4 + 128*B*a*b*c*d^11*f^4
+ 512*B*a*b*c^3*d^9*f^4 + 768*B*a*b*c^5*d^7*f^4 + 512*B*a*b*c^7*d^5*f^4 + 1
28*B*a*b*c^9*d^3*f^4) - (c + d*tan(e + f*x))^(1/2)*(16*B^2*a^4*d^10*f^3 + 1
6*B^2*b^4*d^10*f^3 - 96*B^2*a^2*b^2*d^10*f^3 + 32*B^2*a^4*c^2*d^8*f^3 - 32*
B^2*a^4*c^6*d^4*f^3 - 16*B^2*a^4*c^8*d^2*f^3 + 32*B^2*b^4*c^2*d^8*f^3 - 32*
B^2*b^4*c^6*d^4*f^3 - 16*B^2*b^4*c^8*d^2*f^3 + 128*B^2*a*b^3*c*d^9*f^3 - 12
8*B^2*a^3*b*c*d^9*f^3 + 384*B^2*a*b^3*c^3*d^7*f^3 + 384*B^2*a*b^3*c^5*d^5*f
^3 + 128*B^2*a*b^3*c^7*d^3*f^3 - 384*B^2*a^3*b*c^3*d^7*f^3 - 384*B^2*a^3*b*
c^5*d^5*f^3 - 128*B^2*a^3*b*c^7*d^3*f^3 - 192*B^2*a^2*b^2*c^2*d^8*f^3 + 192
*B^2*a^2*b^2*c^6*d^4*f^3 + 96*B^2*a^2*b^2*c^8*d^2*f^3))*((((8*B^2*a^4*c^3*f
^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^2 - 32
*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*B^2*b^4*c*d^2*f^2 - 96*B^2*a
*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B^2*a^2*b^2*c*d^2*f^2)^2/4 -
(16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(B^4*a^8 + B^4*
b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) + 4*B^2*a^4*c^3
*f^2 + 4*B^2*b^4*c^3*f^2 - 24*B^2*a^2*b^2*c^3*f^2 + 16*B^2*a*b^3*d^3*f^2 -
16*B^2*a^3*b*d^3*f^2 - 12*B^2*a^4*c*d^2*f^2 - 12*B^2*b^4*c*d^2*f^2 - 48*B^2
*a*b^3*c^2*d*f^2 + 48*B^2*a^3*b*c^2*d*f^2 + 72*B^2*a^2*b^2*c*d^2*f^2)/(16*(
c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2) - (((((8*B^2*a^4
*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^
2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*B^2*b^4*c*d^2*f^2 - 96
*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B^2*a^2*b^2*c*d^2*f^2)^
2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(B^4*a^8
+ B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) + 4*B^2*a
^4*c^3*f^2 + 4*B^2*b^4*c^3*f^2 - 24*B^2*a^2*b^2*c^3*f^2 + 16*B^2*a*b^3*d^3*
f^2 - 16*B^2*a^3*b*d^3*f^2 - 12*B^2*a^4*c*d^2*f^2 - 12*B^2*b^4*c*d^2*f^2 -
48*B^2*a*b^3*c^2*d*f^2 + 48*B^2*a^3*b*c^2*d*f^2 + 72*B^2*a^2*b^2*c*d^2*f^2)
/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*ta
n(e + f*x))^(1/2)*((((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^
2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*
f^2 - 24*B^2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^
2 + 144*B^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*
f^4 + 48*c^4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 +
4*B^4*a^6*b^2))^(1/2) + 4*B^2*a^4*c^3*f^2 + 4*B^2*b^4*c^3*f^2 - 24*B^2*a^2*
b^2*c^3*f^2 + 16*B^2*a*b^3*d^3*f^2 - 16*B^2*a^3*b*d^3*f^2 - 12*B^2*a^4*c*d^
2*f^2 - 12*B^2*b^4*c*d^2*f^2 - 48*B^2*a*b^3*c^2*d*f^2 + 48*B^2*a^3*b*c^2*d*
f^2 + 72*B^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*
c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 +
640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 32*B*a^2*d^12*f^4 +
32*B*b^2*d^12*f^4 - 96*B*a^2*c^2*d^10*f^4 - 64*B*a^2*c^4*d^8*f^4 + 64*B*a^2
*c^6*d^6*f^4 + 96*B*a^2*c^8*d^4*f^4 + 32*B*a^2*c^10*d^2*f^4 + 96*B*b^2*c^2*
d^10*f^4 + 64*B*b^2*c^4*d^8*f^4 - 64*B*b^2*c^6*d^6*f^4 - 96*B*b^2*c^8*d^4*f
^4 - 32*B*b^2*c^10*d^2*f^4 + 128*B*a*b*c*d^11*f^4 + 512*B*a*b*c^3*d^9*f^4 +
768*B*a*b*c^5*d^7*f^4 + 512*B*a*b*c^7*d^5*f^4 + 128*B*a*b*c^9*d^3*f^4) + (

c + d*tan(e + f*x))^(1/2)*(16*B^2*a^4*d^10*f^3 + 16*B^2*b^4*d^10*f^3 - 96*B
^2*a^2*b^2*d^10*f^3 + 32*B^2*a^4*c^2*d^8*f^3 - 32*B^2*a^4*c^6*d^4*f^3 - 16*
B^2*a^4*c^8*d^2*f^3 + 32*B^2*b^4*c^2*d^8*f^3 - 32*B^2*b^4*c^6*d^4*f^3 - 16*
B^2*b^4*c^8*d^2*f^3 + 128*B^2*a*b^3*c*d^9*f^3 - 128*B^2*a^3*b*c*d^9*f^3 + 3
84*B^2*a*b^3*c^3*d^7*f^3 + 384*B^2*a*b^3*c^5*d^5*f^3 + 128*B^2*a*b^3*c^7*d^
3*f^3 - 384*B^2*a^3*b*c^3*d^7*f^3 - 384*B^2*a^3*b*c^5*d^5*f^3 - 128*B^2*a^3
*b*c^7*d^3*f^3 - 192*B^2*a^2*b^2*c^2*d^8*f^3 + 192*B^2*a^2*b^2*c^6*d^4*f^3
+ 96*B^2*a^2*b^2*c^8*d^2*f^3))*((((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 -
48*B^2*a^2*b^2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B
^2*a^4*c*d^2*f^2 - 24*B^2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a
^3*b*c^2*d*f^2 + 144*B^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4
+ 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B
^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) + 4*B^2*a^4*c^3*f^2 + 4*B^2*b^4*c^3*f^2
- 24*B^2*a^2*b^2*c^3*f^2 + 16*B^2*a*b^3*d^3*f^2 - 16*B^2*a^3*b*d^3*f^2 - 12
*B^2*a^4*c*d^2*f^2 - 12*B^2*b^4*c*d^2*f^2 - 48*B^2*a*b^3*c^2*d*f^2 + 48*B^2
*a^3*b*c^2*d*f^2 + 72*B^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2
*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2) + 48*B^3*a^6*c^3*d^6*f^2 + 48*B^3*a^6*c^5
*d^4*f^2 + 16*B^3*a^6*c^7*d^2*f^2 - 48*B^3*b^6*c^3*d^6*f^2 - 48*B^3*b^6*c^5
*d^4*f^2 - 16*B^3*b^6*c^7*d^2*f^2 + 32*B^3*a*b^5*d^9*f^2 + 32*B^3*a^5*b*d^9
*f^2 + 16*B^3*a^6*c*d^8*f^2 - 16*B^3*b^6*c*d^8*f^2 + 96*B^3*a*b^5*c^2*d^7*f
^2 + 96*B^3*a*b^5*c^4*d^5*f^2 + 32*B^3*a*b^5*c^6*d^3*f^2 - 16*B^3*a^2*b^4*c
*d^8*f^2 + 16*B^3*a^4*b^2*c*d^8*f^2 + 96*B^3*a^5*b*c^2*d^7*f^2 + 96*B^3*a^5
*b*c^4*d^5*f^2 + 32*B^3*a^5*b*c^6*d^3*f^2 - 48*B^3*a^2*b^4*c^3*d^6*f^2 - 48
*B^3*a^2*b^4*c^5*d^4*f^2 - 16*B^3*a^2*b^4*c^7*d^2*f^2 + 192*B^3*a^3*b^3*c^2
*d^7*f^2 + 192*B^3*a^3*b^3*c^4*d^5*f^2 + 64*B^3*a^3*b^3*c^6*d^3*f^2 + 48*B^
3*a^4*b^2*c^3*d^6*f^2 + 48*B^3*a^4*b^2*c^5*d^4*f^2 + 16*B^3*a^4*b^2*c^7*d^2
f^2))((((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2 +
32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*B^
2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B^2
*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^
4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b
^2))^(1/2) + 4*B^2*a^4*c^3*f^2 + 4*B^2*b^4*c^3*f^2 - 24*B^2*a^2*b^2*c^3*f^2
+ 16*B^2*a*b^3*d^3*f^2 - 16*B^2*a^3*b*d^3*f^2 - 12*B^2*a^4*c*d^2*f^2 - 12*
B^2*b^4*c*d^2*f^2 - 48*B^2*a*b^3*c^2*d*f^2 + 48*B^2*a^3*b*c^2*d*f^2 + 72*B^
2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4
)))^(1/2)*2i - atan((((c + d*tan(e + f*x))^(1/2)*(16*C^2*a^4*d^10*f^3 + 16*
C^2*b^4*d^10*f^3 - 96*C^2*a^2*b^2*d^10*f^3 + 32*C^2*a^4*c^2*d^8*f^3 - 32*C^
2*a^4*c^6*d^4*f^3 - 16*C^2*a^4*c^8*d^2*f^3 + 32*C^2*b^4*c^2*d^8*f^3 - 32*C^
2*b^4*c^6*d^4*f^3 - 16*C^2*b^4*c^8*d^2*f^3 + 128*C^2*a*b^3*c*d^9*f^3 - 128*
C^2*a^3*b*c*d^9*f^3 + 384*C^2*a*b^3*c^3*d^7*f^3 + 384*C^2*a*b^3*c^5*d^5*f^3
+ 128*C^2*a*b^3*c^7*d^3*f^3 - 384*C^2*a^3*b*c^3*d^7*f^3 - 384*C^2*a^3*b*c^
5*d^5*f^3 - 128*C^2*a^3*b*c^7*d^3*f^3 - 192*C^2*a^2*b^2*c^2*d^8*f^3 + 192*C
^2*a^2*b^2*c^6*d^4*f^3 + 96*C^2*a^2*b^2*c^8*d^2*f^3) - ((((8*C^2*a^4*c^3*f^
2 + 8*C^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32*C^2*a*b^3*d^3*f^2 - 32*
C^2*a^3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*b^4*c*d^2*f^2 - 96*C^2*a*

b^3*c^2*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a^2*b^2*c*d^2*f^2)^2/4 - (
16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(C^4*a^8 + C^4*b
^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) - 4*C^2*a^4*c^3*
f^2 - 4*C^2*b^4*c^3*f^2 + 24*C^2*a^2*b^2*c^3*f^2 - 16*C^2*a*b^3*d^3*f^2 + 1
6*C^2*a^3*b*d^3*f^2 + 12*C^2*a^4*c*d^2*f^2 + 12*C^2*b^4*c*d^2*f^2 + 48*C^2*
a*b^3*c^2*d*f^2 - 48*C^2*a^3*b*c^2*d*f^2 - 72*C^2*a^2*b^2*c*d^2*f^2)/(16*(c
^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f
x))^(1/2)((((8*C^2*a^4*c^3*f^2 + 8*C^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f
^2 + 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 2
4*C^2*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144
*C^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 4
8*c^4*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a
^6*b^2))^(1/2) - 4*C^2*a^4*c^3*f^2 - 4*C^2*b^4*c^3*f^2 + 24*C^2*a^2*b^2*c^3
*f^2 - 16*C^2*a*b^3*d^3*f^2 + 16*C^2*a^3*b*d^3*f^2 + 12*C^2*a^4*c*d^2*f^2 +
 12*C^2*b^4*c*d^2*f^2 + 48*C^2*a*b^3*c^2*d*f^2 - 48*C^2*a^3*b*c^2*d*f^2 - 7
2*C^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2
f^4)))^(1/2)(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7
*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 64*C*a^2*c*d^11*f^4 + 64*C*
b^2*c*d^11*f^4 - 256*C*a^2*c^3*d^9*f^4 - 384*C*a^2*c^5*d^7*f^4 - 256*C*a^2*
c^7*d^5*f^4 - 64*C*a^2*c^9*d^3*f^4 + 256*C*b^2*c^3*d^9*f^4 + 384*C*b^2*c^5*
d^7*f^4 + 256*C*b^2*c^7*d^5*f^4 + 64*C*b^2*c^9*d^3*f^4 - 64*C*a*b*d^12*f^4
- 192*C*a*b*c^2*d^10*f^4 - 128*C*a*b*c^4*d^8*f^4 + 128*C*a*b*c^6*d^6*f^4 +
192*C*a*b*c^8*d^4*f^4 + 64*C*a*b*c^10*d^2*f^4))*((((8*C^2*a^4*c^3*f^2 + 8*C
^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3
*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2
*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*
f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*
C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) - 4*C^2*a^4*c^3*f^2 - 4
*C^2*b^4*c^3*f^2 + 24*C^2*a^2*b^2*c^3*f^2 - 16*C^2*a*b^3*d^3*f^2 + 16*C^2*a
^3*b*d^3*f^2 + 12*C^2*a^4*c*d^2*f^2 + 12*C^2*b^4*c*d^2*f^2 + 48*C^2*a*b^3*c
^2*d*f^2 - 48*C^2*a^3*b*c^2*d*f^2 - 72*C^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4
+ d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*1i + ((c + d*tan(e + f*x
))^(1/2)*(16*C^2*a^4*d^10*f^3 + 16*C^2*b^4*d^10*f^3 - 96*C^2*a^2*b^2*d^10*f
^3 + 32*C^2*a^4*c^2*d^8*f^3 - 32*C^2*a^4*c^6*d^4*f^3 - 16*C^2*a^4*c^8*d^2*f
^3 + 32*C^2*b^4*c^2*d^8*f^3 - 32*C^2*b^4*c^6*d^4*f^3 - 16*C^2*b^4*c^8*d^2*f
^3 + 128*C^2*a*b^3*c*d^9*f^3 - 128*C^2*a^3*b*c*d^9*f^3 + 384*C^2*a*b^3*c^3*
d^7*f^3 + 384*C^2*a*b^3*c^5*d^5*f^3 + 128*C^2*a*b^3*c^7*d^3*f^3 - 384*C^2*a
^3*b*c^3*d^7*f^3 - 384*C^2*a^3*b*c^5*d^5*f^3 - 128*C^2*a^3*b*c^7*d^3*f^3 -
192*C^2*a^2*b^2*c^2*d^8*f^3 + 192*C^2*a^2*b^2*c^6*d^4*f^3 + 96*C^2*a^2*b^2*
c^8*d^2*f^3) - ((((8*C^2*a^4*c^3*f^2 + 8*C^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c
^3*f^2 + 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2
 - 24*C^2*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 +
 144*C^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4
 + 48*c^4*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C
^4*a^6*b^2))^(1/2) - 4*C^2*a^4*c^3*f^2 - 4*C^2*b^4*c^3*f^2 + 24*C^2*a^2*b^2

*c^3*f^2 - 16*C^2*a*b^3*d^3*f^2 + 16*C^2*a^3*b*d^3*f^2 + 12*C^2*a^4*c*d^2*f
^2 + 12*C^2*b^4*c*d^2*f^2 + 48*C^2*a*b^3*c^2*d*f^2 - 48*C^2*a^3*b*c^2*d*f^2
 - 72*C^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4
*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*C^2*a^4*c^3*f^2 + 8*C^2
*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b
*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d
*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^
4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^
4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) - 4*C^2*a^4*c^3*f^2 - 4*C
^2*b^4*c^3*f^2 + 24*C^2*a^2*b^2*c^3*f^2 - 16*C^2*a*b^3*d^3*f^2 + 16*C^2*a^3
*b*d^3*f^2 + 12*C^2*a^4*c*d^2*f^2 + 12*C^2*b^4*c*d^2*f^2 + 48*C^2*a*b^3*c^2
*d*f^2 - 48*C^2*a^3*b*c^2*d*f^2 - 72*C^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 +
d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d
^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2
*f^5) + 64*C*a^2*c*d^11*f^4 - 64*C*b^2*c*d^11*f^4 + 256*C*a^2*c^3*d^9*f^4 +
 384*C*a^2*c^5*d^7*f^4 + 256*C*a^2*c^7*d^5*f^4 + 64*C*a^2*c^9*d^3*f^4 - 256
*C*b^2*c^3*d^9*f^4 - 384*C*b^2*c^5*d^7*f^4 - 256*C*b^2*c^7*d^5*f^4 - 64*C*b
^2*c^9*d^3*f^4 + 64*C*a*b*d^12*f^4 + 192*C*a*b*c^2*d^10*f^4 + 128*C*a*b*c^4
*d^8*f^4 - 128*C*a*b*c^6*d^6*f^4 - 192*C*a*b*c^8*d^4*f^4 - 64*C*a*b*c^10*d^
2*f^4))*((((8*C^2*a^4*c^3*f^2 + 8*C^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2
+ 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C
^2*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^
2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c
^4*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*
b^2))^(1/2) - 4*C^2*a^4*c^3*f^2 - 4*C^2*b^4*c^3*f^2 + 24*C^2*a^2*b^2*c^3*f^
2 - 16*C^2*a*b^3*d^3*f^2 + 16*C^2*a^3*b*d^3*f^2 + 12*C^2*a^4*c*d^2*f^2 + 12
*C^2*b^4*c*d^2*f^2 + 48*C^2*a*b^3*c^2*d*f^2 - 48*C^2*a^3*b*c^2*d*f^2 - 72*C
^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^
4)))^(1/2)*1i)/(((c + d*tan(e + f*x))^(1/2)*(16*C^2*a^4*d^10*f^3 + 16*C^2*b
^4*d^10*f^3 - 96*C^2*a^2*b^2*d^10*f^3 + 32*C^2*a^4*c^2*d^8*f^3 - 32*C^2*a^4
*c^6*d^4*f^3 - 16*C^2*a^4*c^8*d^2*f^3 + 32*C^2*b^4*c^2*d^8*f^3 - 32*C^2*b^4
*c^6*d^4*f^3 - 16*C^2*b^4*c^8*d^2*f^3 + 128*C^2*a*b^3*c*d^9*f^3 - 128*C^2*a
^3*b*c*d^9*f^3 + 384*C^2*a*b^3*c^3*d^7*f^3 + 384*C^2*a*b^3*c^5*d^5*f^3 + 12
8*C^2*a*b^3*c^7*d^3*f^3 - 384*C^2*a^3*b*c^3*d^7*f^3 - 384*C^2*a^3*b*c^5*d^5
*f^3 - 128*C^2*a^3*b*c^7*d^3*f^3 - 192*C^2*a^2*b^2*c^2*d^8*f^3 + 192*C^2*a^
2*b^2*c^6*d^4*f^3 + 96*C^2*a^2*b^2*c^8*d^2*f^3) - ((((8*C^2*a^4*c^3*f^2 + 8
*C^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a
^3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c
^2*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^
6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(C^4*a^8 + C^4*b^8 +
4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) - 4*C^2*a^4*c^3*f^2 -
 4*C^2*b^4*c^3*f^2 + 24*C^2*a^2*b^2*c^3*f^2 - 16*C^2*a*b^3*d^3*f^2 + 16*C^2
*a^3*b*d^3*f^2 + 12*C^2*a^4*c*d^2*f^2 + 12*C^2*b^4*c*d^2*f^2 + 48*C^2*a*b^3
*c^2*d*f^2 - 48*C^2*a^3*b*c^2*d*f^2 - 72*C^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^
4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^

(1/2)*((((8*C^2*a^4*c^3*f^2 + 8*C^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2 *b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2* a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4 *d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^ 2))^(1/2) - 4*C^2*a^4*c^3*f^2 - 4*C^2*b^4*c^3*f^2 + 24*C^2*a^2*b^2*c^3*f^2 - 16*C^2*a*b^3*d^3*f^2 + 16*C^2*a^3*b*d^3*f^2 + 12*C^2*a^4*c*d^2*f^2 + 12*C ^2*b^4*c*d^2*f^2 + 48*C^2*a*b^3*c^2*d*f^2 - 48*C^2*a^3*b*c^2*d*f^2 - 72*C^2 *a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6* f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 64*C*a^2*c*d^11*f^4 + 64*C*b^2*c *d^11*f^4 - 256*C*a^2*c^3*d^9*f^4 - 384*C*a^2*c^5*d^7*f^4 - 256*C*a^2*c^7*d ^5*f^4 - 64*C*a^2*c^9*d^3*f^4 + 256*C*b^2*c^3*d^9*f^4 + 384*C*b^2*c^5*d^7*f ^4 + 256*C*b^2*c^7*d^5*f^4 + 64*C*b^2*c^9*d^3*f^4 - 64*C*a*b*d^12*f^4 - 192 *C*a*b*c^2*d^10*f^4 - 128*C*a*b*c^4*d^8*f^4 + 128*C*a*b*c^6*d^6*f^4 + 192*C *a*b*c^8*d^4*f^4 + 64*C*a*b*c^10*d^2*f^4))*((((8*C^2*a^4*c^3*f^2 + 8*C^2*b^ 4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^ 3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^ 2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a ^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) - 4*C^2*a^4*c^3*f^2 - 4*C^2* b^4*c^3*f^2 + 24*C^2*a^2*b^2*c^3*f^2 - 16*C^2*a*b^3*d^3*f^2 + 16*C^2*a^3*b* d^3*f^2 + 12*C^2*a^4*c*d^2*f^2 + 12*C^2*b^4*c*d^2*f^2 + 48*C^2*a*b^3*c^2*d* f^2 - 48*C^2*a^3*b*c^2*d*f^2 - 72*C^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6 *f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2) - ((c + d*tan(e + f*x))^(1/2) *(16*C^2*a^4*d^10*f^3 + 16*C^2*b^4*d^10*f^3 - 96*C^2*a^2*b^2*d^10*f^3 + 32* C^2*a^4*c^2*d^8*f^3 - 32*C^2*a^4*c^6*d^4*f^3 - 16*C^2*a^4*c^8*d^2*f^3 + 32* C^2*b^4*c^2*d^8*f^3 - 32*C^2*b^4*c^6*d^4*f^3 - 16*C^2*b^4*c^8*d^2*f^3 + 128 *C^2*a*b^3*c*d^9*f^3 - 128*C^2*a^3*b*c*d^9*f^3 + 384*C^2*a*b^3*c^3*d^7*f^3 + 384*C^2*a*b^3*c^5*d^5*f^3 + 128*C^2*a*b^3*c^7*d^3*f^3 - 384*C^2*a^3*b*c^3 *d^7*f^3 - 384*C^2*a^3*b*c^5*d^5*f^3 - 128*C^2*a^3*b*c^7*d^3*f^3 - 192*C^2* a^2*b^2*c^2*d^8*f^3 + 192*C^2*a^2*b^2*c^6*d^4*f^3 + 96*C^2*a^2*b^2*c^8*d^2* f^3) - ((((8*C^2*a^4*c^3*f^2 + 8*C^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^ 2*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2 *a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^ 4*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b ^2))^(1/2) - 4*C^2*a^4*c^3*f^2 - 4*C^2*b^4*c^3*f^2 + 24*C^2*a^2*b^2*c^3*f^2 - 16*C^2*a*b^3*d^3*f^2 + 16*C^2*a^3*b*d^3*f^2 + 12*C^2*a^4*c*d^2*f^2 + 12* C^2*b^4*c*d^2*f^2 + 48*C^2*a*b^3*c^2*d*f^2 - 48*C^2*a^3*b*c^2*d*f^2 - 72*C^ 2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*C^2*a^4*c^3*f^2 + 8*C^2*b^4*c^3 *f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^2 + 9 6*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d

^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^
6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) - 4*C^2*a^4*c^3*f^2 - 4*C^2*b^4*c
^3*f^2 + 24*C^2*a^2*b^2*c^3*f^2 - 16*C^2*a*b^3*d^3*f^2 + 16*C^2*a^3*b*d^3*f
^2 + 12*C^2*a^4*c*d^2*f^2 + 12*C^2*b^4*c*d^2*f^2 + 48*C^2*a*b^3*c^2*d*f^2 -
48*C^2*a^3*b*c^2*d*f^2 - 72*C^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4
+ 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5
+ 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) +
64*C*a^2*c*d^11*f^4 - 64*C*b^2*c*d^11*f^4 + 256*C*a^2*c^3*d^9*f^4 + 384*C*a
^2*c^5*d^7*f^4 + 256*C*a^2*c^7*d^5*f^4 + 64*C*a^2*c^9*d^3*f^4 - 256*C*b^2*c
^3*d^9*f^4 - 384*C*b^2*c^5*d^7*f^4 - 256*C*b^2*c^7*d^5*f^4 - 64*C*b^2*c^9*d
^3*f^4 + 64*C*a*b*d^12*f^4 + 192*C*a*b*c^2*d^10*f^4 + 128*C*a*b*c^4*d^8*f^4
- 128*C*a*b*c^6*d^6*f^4 - 192*C*a*b*c^8*d^4*f^4 - 64*C*a*b*c^10*d^2*f^4))*
((((8*C^2*a^4*c^3*f^2 + 8*C^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32*C^2
*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*b^4*c
*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a^2*b^
2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f
^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1
/2) - 4*C^2*a^4*c^3*f^2 - 4*C^2*b^4*c^3*f^2 + 24*C^2*a^2*b^2*c^3*f^2 - 16*C
^2*a*b^3*d^3*f^2 + 16*C^2*a^3*b*d^3*f^2 + 12*C^2*a^4*c*d^2*f^2 + 12*C^2*b^4
*c*d^2*f^2 + 48*C^2*a*b^3*c^2*d*f^2 - 48*C^2*a^3*b*c^2*d*f^2 - 72*C^2*a^2*b
^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/
2) - 16*C^3*a^6*d^9*f^2 + 16*C^3*b^6*d^9*f^2 + 16*C^3*a^2*b^4*d^9*f^2 - 16*
C^3*a^4*b^2*d^9*f^2 - 48*C^3*a^6*c^2*d^7*f^2 - 48*C^3*a^6*c^4*d^5*f^2 - 16*
C^3*a^6*c^6*d^3*f^2 + 48*C^3*b^6*c^2*d^7*f^2 + 48*C^3*b^6*c^4*d^5*f^2 + 16*
C^3*b^6*c^6*d^3*f^2 + 32*C^3*a*b^5*c*d^8*f^2 + 32*C^3*a^5*b*c*d^8*f^2 + 96*
C^3*a*b^5*c^3*d^6*f^2 + 96*C^3*a*b^5*c^5*d^4*f^2 + 32*C^3*a*b^5*c^7*d^2*f^2
+ 64*C^3*a^3*b^3*c*d^8*f^2 + 96*C^3*a^5*b*c^3*d^6*f^2 + 96*C^3*a^5*b*c^5*d
^4*f^2 + 32*C^3*a^5*b*c^7*d^2*f^2 + 48*C^3*a^2*b^4*c^2*d^7*f^2 + 48*C^3*a^2
*b^4*c^4*d^5*f^2 + 16*C^3*a^2*b^4*c^6*d^3*f^2 + 192*C^3*a^3*b^3*c^3*d^6*f^2
+ 192*C^3*a^3*b^3*c^5*d^4*f^2 + 64*C^3*a^3*b^3*c^7*d^2*f^2 - 48*C^3*a^4*b^
2*c^2*d^7*f^2 - 48*C^3*a^4*b^2*c^4*d^5*f^2 - 16*C^3*a^4*b^2*c^6*d^3*f^2))*(
(((8*C^2*a^4*c^3*f^2 + 8*C^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32*C^2*
a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*b^4*c*
d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a^2*b^2
*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^
4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/
2) - 4*C^2*a^4*c^3*f^2 - 4*C^2*b^4*c^3*f^2 + 24*C^2*a^2*b^2*c^3*f^2 - 16*C^
2*a*b^3*d^3*f^2 + 16*C^2*a^3*b*d^3*f^2 + 12*C^2*a^4*c*d^2*f^2 + 12*C^2*b^4*
c*d^2*f^2 + 48*C^2*a*b^3*c^2*d*f^2 - 48*C^2*a^3*b*c^2*d*f^2 - 72*C^2*a^2*b^
2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2
)*2i - atan((((c + d*tan(e + f*x))^(1/2)*(16*C^2*a^4*d^10*f^3 + 16*C^2*b^4*
d^10*f^3 - 96*C^2*a^2*b^2*d^10*f^3 + 32*C^2*a^4*c^2*d^8*f^3 - 32*C^2*a^4*c^
6*d^4*f^3 - 16*C^2*a^4*c^8*d^2*f^3 + 32*C^2*b^4*c^2*d^8*f^3 - 32*C^2*b^4*c^
6*d^4*f^3 - 16*C^2*b^4*c^8*d^2*f^3 + 128*C^2*a*b^3*c*d^9*f^3 - 128*C^2*a^3*
b*c*d^9*f^3 + 384*C^2*a*b^3*c^3*d^7*f^3 + 384*C^2*a*b^3*c^5*d^5*f^3 + 128*C

^2*a*b^3*c^7*d^3*f^3 - 384*C^2*a^3*b*c^3*d^7*f^3 - 384*C^2*a^3*b*c^5*d^5*f^
3 - 128*C^2*a^3*b*c^7*d^3*f^3 - 192*C^2*a^2*b^2*c^2*d^8*f^3 + 192*C^2*a^2*b
^2*c^6*d^4*f^3 + 96*C^2*a^2*b^2*c^8*d^2*f^3) - (-(((8*C^2*a^4*c^3*f^2 + 8*C
^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3
*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2
*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*
f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*
C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) + 4*C^2*a^4*c^3*f^2 + 4
*C^2*b^4*c^3*f^2 - 24*C^2*a^2*b^2*c^3*f^2 + 16*C^2*a*b^3*d^3*f^2 - 16*C^2*a
^3*b*d^3*f^2 - 12*C^2*a^4*c*d^2*f^2 - 12*C^2*b^4*c*d^2*f^2 - 48*C^2*a*b^3*c
^2*d*f^2 + 48*C^2*a^3*b*c^2*d*f^2 + 72*C^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4
+ d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1
/2)*(-(((8*C^2*a^4*c^3*f^2 + 8*C^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 3
2*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*
b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a
^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*
d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2
))^(1/2) + 4*C^2*a^4*c^3*f^2 + 4*C^2*b^4*c^3*f^2 - 24*C^2*a^2*b^2*c^3*f^2 +
16*C^2*a*b^3*d^3*f^2 - 16*C^2*a^3*b*d^3*f^2 - 12*C^2*a^4*c*d^2*f^2 - 12*C^
2*b^4*c*d^2*f^2 - 48*C^2*a*b^3*c^2*d*f^2 + 48*C^2*a^3*b*c^2*d*f^2 + 72*C^2*
a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))
)^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f
^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 64*C*a^2*c*d^11*f^4 + 64*C*b^2*c*
d^11*f^4 - 256*C*a^2*c^3*d^9*f^4 - 384*C*a^2*c^5*d^7*f^4 - 256*C*a^2*c^7*d^
5*f^4 - 64*C*a^2*c^9*d^3*f^4 + 256*C*b^2*c^3*d^9*f^4 + 384*C*b^2*c^5*d^7*f^
4 + 256*C*b^2*c^7*d^5*f^4 + 64*C*b^2*c^9*d^3*f^4 - 64*C*a*b*d^12*f^4 - 192*
C*a*b*c^2*d^10*f^4 - 128*C*a*b*c^4*d^8*f^4 + 128*C*a*b*c^6*d^6*f^4 + 192*C*
a*b*c^8*d^4*f^4 + 64*C*a*b*c^10*d^2*f^4))*(-(((8*C^2*a^4*c^3*f^2 + 8*C^2*b^
4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^
3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^
2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 +
16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a
^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) + 4*C^2*a^4*c^3*f^2 + 4*C^2*
b^4*c^3*f^2 - 24*C^2*a^2*b^2*c^3*f^2 + 16*C^2*a*b^3*d^3*f^2 - 16*C^2*a^3*b*
d^3*f^2 - 12*C^2*a^4*c*d^2*f^2 - 12*C^2*b^4*c*d^2*f^2 - 48*C^2*a*b^3*c^2*d*
f^2 + 48*C^2*a^3*b*c^2*d*f^2 + 72*C^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6
*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*1i + ((c + d*tan(e + f*x))^(1
/2)*(16*C^2*a^4*d^10*f^3 + 16*C^2*b^4*d^10*f^3 - 96*C^2*a^2*b^2*d^10*f^3 +
32*C^2*a^4*c^2*d^8*f^3 - 32*C^2*a^4*c^6*d^4*f^3 - 16*C^2*a^4*c^8*d^2*f^3 +
32*C^2*b^4*c^2*d^8*f^3 - 32*C^2*b^4*c^6*d^4*f^3 - 16*C^2*b^4*c^8*d^2*f^3 +
128*C^2*a*b^3*c*d^9*f^3 - 128*C^2*a^3*b*c*d^9*f^3 + 384*C^2*a*b^3*c^3*d^7*f
^3 + 384*C^2*a*b^3*c^5*d^5*f^3 + 128*C^2*a*b^3*c^7*d^3*f^3 - 384*C^2*a^3*b*
c^3*d^7*f^3 - 384*C^2*a^3*b*c^5*d^5*f^3 - 128*C^2*a^3*b*c^7*d^3*f^3 - 192*C
^2*a^2*b^2*c^2*d^8*f^3 + 192*C^2*a^2*b^2*c^6*d^4*f^3 + 96*C^2*a^2*b^2*c^8*d
^2*f^3) - (-(((8*C^2*a^4*c^3*f^2 + 8*C^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f

```
^2 + 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 2
4*C^2*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144
*C^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 4
8*c^4*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a
^6*b^2))^(1/2) + 4*C^2*a^4*c^3*f^2 + 4*C^2*b^4*c^3*f^2 - 24*C^2*a^2*b^2*c^3
*f^2 + 16*C^2*a*b^3*d^3*f^2 - 16*C^2*a^3*b*d^3*f^2 - 12*C^2*a^4*c*d^2*f^2 -
 12*C^2*b^4*c*d^2*f^2 - 48*C^2*a*b^3*c^2*d*f^2 + 48*C^2*a^3*b*c^2*d*f^2 + 7
2*C^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2
*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^4*c^3*f^2 + 8*C^2*b^
4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^
3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^
2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 +
 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a
^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) + 4*C^2*a^4*c^3*f^2 + 4*C^2*
b^4*c^3*f^2 - 24*C^2*a^2*b^2*c^3*f^2 + 16*C^2*a*b^3*d^3*f^2 - 16*C^2*a^3*b*
d^3*f^2 - 12*C^2*a^4*c*d^2*f^2 - 12*C^2*b^4*c*d^2*f^2 - 48*C^2*a*b^3*c^2*d*
f^2 + 48*C^2*a^3*b*c^2*d*f^2 + 72*C^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6
*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10
*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^
5) + 64*C*a^2*c*d^11*f^4 - 64*C*b^2*c*d^11*f^4 + 256*C*a^2*c^3*d^9*f^4 + 38
4*C*a^2*c^5*d^7*f^4 + 256*C*a^2*c^7*d^5*f^4 + 64*C*a^2*c^9*d^3*f^4 - 256*C*
b^2*c^3*d^9*f^4 - 384*C*b^2*c^5*d^7*f^4 - 256*C*b^2*c^7*d^5*f^4 - 64*C*b^2*
c^9*d^3*f^4 + 64*C*a*b*d^12*f^4 + 192*C*a*b*c^2*d^10*f^4 + 128*C*a*b*c^4*d^
8*f^4 - 128*C*a*b*c^6*d^6*f^4 - 192*C*a*b*c^8*d^4*f^4 - 64*C*a*b*c^10*d^2*f
^4))*(-(((8*C^2*a^4*c^3*f^2 + 8*C^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 +
32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2
*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*
a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4
*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^
2))^(1/2) + 4*C^2*a^4*c^3*f^2 + 4*C^2*b^4*c^3*f^2 - 24*C^2*a^2*b^2*c^3*f^2
+ 16*C^2*a*b^3*d^3*f^2 - 16*C^2*a^3*b*d^3*f^2 - 12*C^2*a^4*c*d^2*f^2 - 12*C
^2*b^4*c*d^2*f^2 - 48*C^2*a*b^3*c^2*d*f^2 + 48*C^2*a^3*b*c^2*d*f^2 + 72*C^2
*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)
))^(1/2)*1i)/(((c + d*tan(e + f*x))^(1/2)*(16*C^2*a^4*d^10*f^3 + 16*C^2*b^4
*d^10*f^3 - 96*C^2*a^2*b^2*d^10*f^3 + 32*C^2*a^4*c^2*d^8*f^3 - 32*C^2*a^4*c
^6*d^4*f^3 - 16*C^2*a^4*c^8*d^2*f^3 + 32*C^2*b^4*c^2*d^8*f^3 - 32*C^2*b^4*c
^6*d^4*f^3 - 16*C^2*b^4*c^8*d^2*f^3 + 128*C^2*a*b^3*c*d^9*f^3 - 128*C^2*a^3
*b*c*d^9*f^3 + 384*C^2*a*b^3*c^3*d^7*f^3 + 384*C^2*a*b^3*c^5*d^5*f^3 + 128*
C^2*a*b^3*c^7*d^3*f^3 - 384*C^2*a^3*b*c^3*d^7*f^3 - 384*C^2*a^3*b*c^5*d^5*f
^3 - 128*C^2*a^3*b*c^7*d^3*f^3 - 192*C^2*a^2*b^2*c^2*d^8*f^3 + 192*C^2*a^2*
b^2*c^6*d^4*f^3 + 96*C^2*a^2*b^2*c^8*d^2*f^3) - (-(((8*C^2*a^4*c^3*f^2 + 8*
C^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^
3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^
2*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6
*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4
```

*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) + 4*C^2*a^4*c^3*f^2 + 4*C^2*b^4*c^3*f^2 - 24*C^2*a^2*b^2*c^3*f^2 + 16*C^2*a*b^3*d^3*f^2 - 16*C^2*a^3*b*d^3*f^2 - 12*C^2*a^4*c*d^2*f^2 - 12*C^2*b^4*c*d^2*f^2 - 48*C^2*a*b^3*c^2*d*f^2 + 48*C^2*a^3*b*c^2*d*f^2 + 72*C^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^4*c^3*f^2 + 8*C^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) + 4*C^2*a^4*c^3*f^2 + 4*C^2*b^4*c^3*f^2 - 24*C^2*a^2*b^2*c^3*f^2 + 16*C^2*a*b^3*d^3*f^2 - 16*C^2*a^3*b*d^3*f^2 - 12*C^2*a^4*c*d^2*f^2 - 12*C^2*b^4*c*d^2*f^2 - 48*C^2*a*b^3*c^2*d*f^2 + 48*C^2*a^3*b*c^2*d*f^2 + 72*C^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 64*C*a^2*c*d^11*f^4 + 64*C*b^2*c*d^11*f^4 - 256*C*a^2*c^3*d^9*f^4 - 384*C*a^2*c^5*d^7*f^4 - 256*C*a^2*c^7*d^5*f^4 - 64*C*a^2*c^9*d^3*f^4 + 256*C*b^2*c^3*d^9*f^4 + 384*C*b^2*c^5*d^7*f^4 + 256*C*b^2*c^7*d^5*f^4 + 64*C*b^2*c^9*d^3*f^4 - 64*C*a*b*d^12*f^4 - 192*C*a*b*c^2*d^10*f^4 - 128*C*a*b*c^4*d^8*f^4 + 128*C*a*b*c^6*d^6*f^4 + 192*C*a*b*c^8*d^4*f^4 + 64*C*a*b*c^10*d^2*f^4))*(-(((8*C^2*a^4*c^3*f^2 + 8*C^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) + 4*C^2*a^4*c^3*f^2 + 4*C^2*b^4*c^3*f^2 - 24*C^2*a^2*b^2*c^3*f^2 + 16*C^2*a*b^3*d^3*f^2 - 16*C^2*a^3*b*d^3*f^2 - 12*C^2*a^4*c*d^2*f^2 - 12*C^2*b^4*c*d^2*f^2 - 48*C^2*a*b^3*c^2*d*f^2 + 48*C^2*a^3*b*c^2*d*f^2 + 72*C^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2) - ((c + d*tan(e + f*x))^(1/2)*(16*C^2*a^4*d^10*f^3 + 16*C^2*b^4*d^10*f^3 - 96*C^2*a^2*b^2*d^10*f^3 + 32*C^2*a^4*c^2*d^8*f^3 - 32*C^2*a^4*c^6*d^4*f^3 - 16*C^2*a^4*c^8*d^2*f^3 + 32*C^2*b^4*c^2*d^8*f^3 - 32*C^2*b^4*c^6*d^4*f^3 - 16*C^2*b^4*c^8*d^2*f^3 + 128*C^2*a*b^3*c*d^9*f^3 - 128*C^2*a^3*b*c*d^9*f^3 + 384*C^2*a*b^3*c^3*d^7*f^3 + 384*C^2*a*b^3*c^5*d^5*f^3 + 128*C^2*a*b^3*c^7*d^3*f^3 - 384*C^2*a^3*b*c^3*d^7*f^3 - 384*C^2*a^3*b*c^5*d^5*f^3 - 128*C^2*a^3*b*c^7*d^3*f^3 - 192*C^2*a^2*b^2*c^2*d^8*f^3 + 192*C^2*a^2*b^2*c^6*d^4*f^3 + 96*C^2*a^2*b^2*c^8*d^2*f^3) - (-(((8*C^2*a^4*c^3*f^2 + 8*C^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) + 4*C^2*a^4*c^3*f^2 + 4*C^2*b^4*c^3*f^2 - 24*C^2*a^2*b^2*c^3*f^2 + 16*C^2*a*b^3*d^3*f^2 - 16*C^2*a^3*b*d^3*f^2 - 12*C^2*a^4*c*d^2*f^2 - 12*C^2*b^4*c*d^2*f^2 - 48*C^2*a*b^3*c^2*d*f^2 + 48*C^2*a^3*b*c^2*d*f^2 + 72*

C^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f
^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^4*c^3*f^2 + 8*C^2*b^4*
c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^3*
f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^2
+ 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 1
6*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2
*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) + 4*C^2*a^4*c^3*f^2 + 4*C^2*b^
4*c^3*f^2 - 24*C^2*a^2*b^2*c^3*f^2 + 16*C^2*a*b^3*d^3*f^2 - 16*C^2*a^3*b*d^
3*f^2 - 12*C^2*a^4*c*d^2*f^2 - 12*C^2*b^4*c*d^2*f^2 - 48*C^2*a*b^3*c^2*d*f^
2 + 48*C^2*a^3*b*c^2*d*f^2 + 72*C^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f
^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f
^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5)
 + 64*C*a^2*c*d^11*f^4 - 64*C*b^2*c*d^11*f^4 + 256*C*a^2*c^3*d^9*f^4 + 384*
C*a^2*c^5*d^7*f^4 + 256*C*a^2*c^7*d^5*f^4 + 64*C*a^2*c^9*d^3*f^4 - 256*C*b^
2*c^3*d^9*f^4 - 384*C*b^2*c^5*d^7*f^4 - 256*C*b^2*c^7*d^5*f^4 - 64*C*b^2*c^
9*d^3*f^4 + 64*C*a*b*d^12*f^4 + 192*C*a*b*c^2*d^10*f^4 + 128*C*a*b*c^4*d^8*
f^4 - 128*C*a*b*c^6*d^6*f^4 - 192*C*a*b*c^8*d^4*f^4 - 64*C*a*b*c^10*d^2*f^4
))*(-(((8*C^2*a^4*c^3*f^2 + 8*C^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32
*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*b
^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a^
2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d
^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)
)^(1/2) + 4*C^2*a^4*c^3*f^2 + 4*C^2*b^4*c^3*f^2 - 24*C^2*a^2*b^2*c^3*f^2 +
16*C^2*a*b^3*d^3*f^2 - 16*C^2*a^3*b*d^3*f^2 - 12*C^2*a^4*c*d^2*f^2 - 12*C^2
*b^4*c*d^2*f^2 - 48*C^2*a*b^3*c^2*d*f^2 + 48*C^2*a^3*b*c^2*d*f^2 + 72*C^2*a
^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))
^(1/2) - 16*C^3*a^6*d^9*f^2 + 16*C^3*b^6*d^9*f^2 + 16*C^3*a^2*b^4*d^9*f^2 -
 16*C^3*a^4*b^2*d^9*f^2 - 48*C^3*a^6*c^2*d^7*f^2 - 48*C^3*a^6*c^4*d^5*f^2 -
 16*C^3*a^6*c^6*d^3*f^2 + 48*C^3*b^6*c^2*d^7*f^2 + 48*C^3*b^6*c^4*d^5*f^2 +
 16*C^3*b^6*c^6*d^3*f^2 + 32*C^3*a*b^5*c*d^8*f^2 + 32*C^3*a^5*b*c*d^8*f^2 +
 96*C^3*a*b^5*c^3*d^6*f^2 + 96*C^3*a*b^5*c^5*d^4*f^2 + 32*C^3*a*b^5*c^7*d^2
*f^2 + 64*C^3*a^3*b^3*c*d^8*f^2 + 96*C^3*a^5*b*c^3*d^6*f^2 + 96*C^3*a^5*b*c
^5*d^4*f^2 + 32*C^3*a^5*b*c^7*d^2*f^2 + 48*C^3*a^2*b^4*c^2*d^7*f^2 + 48*C^3
*a^2*b^4*c^4*d^5*f^2 + 16*C^3*a^2*b^4*c^6*d^3*f^2 + 192*C^3*a^3*b^3*c^3*d^6
*f^2 + 192*C^3*a^3*b^3*c^5*d^4*f^2 + 64*C^3*a^3*b^3*c^7*d^2*f^2 - 48*C^3*a^
4*b^2*c^2*d^7*f^2 - 48*C^3*a^4*b^2*c^4*d^5*f^2 - 16*C^3*a^4*b^2*c^6*d^3*f^2
))*(-(((8*C^2*a^4*c^3*f^2 + 8*C^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32
*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*b
^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a^
2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d
^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)
)^(1/2) + 4*C^2*a^4*c^3*f^2 + 4*C^2*b^4*c^3*f^2 - 24*C^2*a^2*b^2*c^3*f^2 +
16*C^2*a*b^3*d^3*f^2 - 16*C^2*a^3*b*d^3*f^2 - 12*C^2*a^4*c*d^2*f^2 - 12*C^2
*b^4*c*d^2*f^2 - 48*C^2*a*b^3*c^2*d*f^2 + 48*C^2*a^3*b*c^2*d*f^2 + 72*C^2*a
^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))

^(1/2)*2i - atan(-(((c + d*tan(e + f*x))^(1/2)*(16*A^2*a^4*d^10*f^3 + 16*A^2*b^4*d^10*f^3 - 96*A^2*a^2*b^2*d^10*f^3 + 32*A^2*a^4*c^2*d^8*f^3 - 32*A^2*a^4*c^6*d^4*f^3 - 16*A^2*a^4*c^8*d^2*f^3 + 32*A^2*b^4*c^2*d^8*f^3 - 32*A^2*b^4*c^6*d^4*f^3 - 16*A^2*b^4*c^8*d^2*f^3 + 128*A^2*a*b^3*c*d^9*f^3 - 128*A^2*a^3*b*c*d^9*f^3 + 384*A^2*a*b^3*c^3*d^7*f^3 + 384*A^2*a*b^3*c^5*d^5*f^3 + 128*A^2*a*b^3*c^7*d^3*f^3 - 384*A^2*a^3*b*c^3*d^7*f^3 - 384*A^2*a^3*b*c^5*d^5*f^3 - 128*A^2*a^3*b*c^7*d^3*f^3 - 192*A^2*a^2*b^2*c^2*d^8*f^3 + 192*A^2*a^2*b^2*c^6*d^4*f^3 + 96*A^2*a^2*b^2*c^8*d^2*f^3) - ((((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) - 4*A^2*a^4*c^3*f^2 - 4*A^2*b^4*c^3*f^2 + 24*A^2*a^2*b^2*c^3*f^2 - 16*A^2*a*b^3*d^3*f^2 + 16*A^2*a^3*b*d^3*f^2 + 12*A^2*a^4*c*d^2*f^2 + 12*A^2*b^4*c*d^2*f^2 + 48*A^2*a*b^3*c^2*d*f^2 - 48*A^2*a^3*b*c^2*d*f^2 - 72*A^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) - 4*A^2*a^4*c^3*f^2 - 4*A^2*b^4*c^3*f^2 + 24*A^2*a^2*b^2*c^3*f^2 - 16*A^2*a*b^3*d^3*f^2 + 16*A^2*a^3*b*d^3*f^2 + 12*A^2*a^4*c*d^2*f^2 + 12*A^2*b^4*c*d^2*f^2 + 48*A^2*a*b^3*c^2*d*f^2 - 48*A^2*a^3*b*c^2*d*f^2 - 72*A^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 64*A*a^2*c*d^11*f^4 + 64*A*b^2*c*d^11*f^4 - 256*A*a^2*c^3*d^9*f^4 - 384*A*a^2*c^5*d^7*f^4 - 256*A*a^2*c^7*d^5*f^4 - 64*A*a^2*c^9*d^3*f^4 + 256*A*b^2*c^3*d^9*f^4 + 384*A*b^2*c^5*d^7*f^4 + 256*A*b^2*c^7*d^5*f^4 + 64*A*b^2*c^9*d^3*f^4 - 64*A*a*b*d^12*f^4 - 192*A*a*b*c^2*d^10*f^4 - 128*A*a*b*c^4*d^8*f^4 + 128*A*a*b*c^6*d^6*f^4 + 192*A*a*b*c^8*d^4*f^4 + 64*A*a*b*c^10*d^2*f^4))*((((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) - 4*A^2*a^4*c^3*f^2 - 4*A^2*b^4*c^3*f^2 + 24*A^2*a^2*b^2*c^3*f^2 - 16*A^2*a*b^3*d^3*f^2 + 16*A^2*a^3*b*d^3*f^2 + 12*A^2*a^4*c*d^2*f^2 + 12*A^2*b^4*c*d^2*f^2 + 48*A^2*a*b^3*c^2*d*f^2 - 48*A^2*a^3*b*c^2*d*f^2 - 72*A^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*1i + ((c + d*tan(e + f*x))^(1/2)*(16*A^2*a^4*d^10*f^3 + 16*A^2*b^4*d^10*f^3 - 96*A^2*a^2*b^2*d^10*f^3 + 32*A^2*a^4*c^2*d^8*f^3 - 32*A^2*a^4*c^6*d^4*f^3 - 16*A^2*a^4*c^8*d^2*f^3 + 32*A^2*b^4*c^2*d^8*f^3 - 32*A^2*b^4*c^6*d^4*f^3 - 16*A^2*b^4*c^8*d^2*f^3

+ 128*A^2*a*b^3*c*d^9*f^3 - 128*A^2*a^3*b*c*d^9*f^3 + 384*A^2*a*b^3*c^3*d^
7*f^3 + 384*A^2*a*b^3*c^5*d^5*f^3 + 128*A^2*a*b^3*c^7*d^3*f^3 - 384*A^2*a^3
*b*c^3*d^7*f^3 - 384*A^2*a^3*b*c^5*d^5*f^3 - 128*A^2*a^3*b*c^7*d^3*f^3 - 19
2*A^2*a^2*b^2*c^2*d^8*f^3 + 192*A^2*a^2*b^2*c^6*d^4*f^3 + 96*A^2*a^2*b^2*c^
8*d^2*f^3) - ((((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3
*f^2 + 32*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 -
24*A^2*b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 1
44*A^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 +
48*c^4*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4
*a^6*b^2))^(1/2) - 4*A^2*a^4*c^3*f^2 - 4*A^2*b^4*c^3*f^2 + 24*A^2*a^2*b^2*c
^3*f^2 - 16*A^2*a*b^3*d^3*f^2 + 16*A^2*a^3*b*d^3*f^2 + 12*A^2*a^4*c*d^2*f^2
+ 12*A^2*b^4*c*d^2*f^2 + 48*A^2*a*b^3*c^2*d*f^2 - 48*A^2*a^3*b*c^2*d*f^2 -
72*A^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d
^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^4*c^3*f^2 + 8*A^2*b
^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d
^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f
^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4
+ 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*
a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) - 4*A^2*a^4*c^3*f^2 - 4*A^2
*b^4*c^3*f^2 + 24*A^2*a^2*b^2*c^3*f^2 - 16*A^2*a*b^3*d^3*f^2 + 16*A^2*a^3*b
*d^3*f^2 + 12*A^2*a^4*c*d^2*f^2 + 12*A^2*b^4*c*d^2*f^2 + 48*A^2*a*b^3*c^2*d
*f^2 - 48*A^2*a^3*b*c^2*d*f^2 - 72*A^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^
6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^1
0*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f
^5) + 64*A*a^2*c*d^11*f^4 - 64*A*b^2*c*d^11*f^4 + 256*A*a^2*c^3*d^9*f^4 + 3
84*A*a^2*c^5*d^7*f^4 + 256*A*a^2*c^7*d^5*f^4 + 64*A*a^2*c^9*d^3*f^4 - 256*A
*b^2*c^3*d^9*f^4 - 384*A*b^2*c^5*d^7*f^4 - 256*A*b^2*c^7*d^5*f^4 - 64*A*b^2
*c^9*d^3*f^4 + 64*A*a*b*d^12*f^4 + 192*A*a*b*c^2*d^10*f^4 + 128*A*a*b*c^4*d
^8*f^4 - 128*A*a*b*c^6*d^6*f^4 - 192*A*a*b*c^8*d^4*f^4 - 64*A*a*b*c^10*d^2*
f^4))*((((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 +
32*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2
*b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*
a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4
*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^
2))^(1/2) - 4*A^2*a^4*c^3*f^2 - 4*A^2*b^4*c^3*f^2 + 24*A^2*a^2*b^2*c^3*f^2
- 16*A^2*a*b^3*d^3*f^2 + 16*A^2*a^3*b*d^3*f^2 + 12*A^2*a^4*c*d^2*f^2 + 12*A
^2*b^4*c*d^2*f^2 + 48*A^2*a*b^3*c^2*d*f^2 - 48*A^2*a^3*b*c^2*d*f^2 - 72*A^2
*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)
))^(1/2)*1i)/(((c + d*tan(e + f*x))^(1/2)*(16*A^2*a^4*d^10*f^3 + 16*A^2*b^4
*d^10*f^3 - 96*A^2*a^2*b^2*d^10*f^3 + 32*A^2*a^4*c^2*d^8*f^3 - 32*A^2*a^4*c
^6*d^4*f^3 - 16*A^2*a^4*c^8*d^2*f^3 + 32*A^2*b^4*c^2*d^8*f^3 - 32*A^2*b^4*c
^6*d^4*f^3 - 16*A^2*b^4*c^8*d^2*f^3 + 128*A^2*a*b^3*c*d^9*f^3 - 128*A^2*a^3
*b*c*d^9*f^3 + 384*A^2*a*b^3*c^3*d^7*f^3 + 384*A^2*a*b^3*c^5*d^5*f^3 + 128*
A^2*a*b^3*c^7*d^3*f^3 - 384*A^2*a^3*b*c^3*d^7*f^3 - 384*A^2*a^3*b*c^5*d^5*f
^3 - 128*A^2*a^3*b*c^7*d^3*f^3 - 192*A^2*a^2*b^2*c^2*d^8*f^3 + 192*A^2*a^2*

b^2*c^6*d^4*f^3 + 96*A^2*a^2*b^2*c^8*d^2*f^3) - ((((8*A^2*a^4*c^3*f^2 + 8*A
^2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3
*b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2
*d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*
f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*
A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) - 4*A^2*a^4*c^3*f^2 - 4
*A^2*b^4*c^3*f^2 + 24*A^2*a^2*b^2*c^3*f^2 - 16*A^2*a*b^3*d^3*f^2 + 16*A^2*a
^3*b*d^3*f^2 + 12*A^2*a^4*c*d^2*f^2 + 12*A^2*b^4*c*d^2*f^2 + 48*A^2*a*b^3*c
^2*d*f^2 - 48*A^2*a^3*b*c^2*d*f^2 - 72*A^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4
+ d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1
/2)*((((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32
*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*b
^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^
2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d
^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)
)^(1/2) - 4*A^2*a^4*c^3*f^2 - 4*A^2*b^4*c^3*f^2 + 24*A^2*a^2*b^2*c^3*f^2 -
16*A^2*a*b^3*d^3*f^2 + 16*A^2*a^3*b*d^3*f^2 + 12*A^2*a^4*c*d^2*f^2 + 12*A^2
*b^4*c*d^2*f^2 + 48*A^2*a*b^3*c^2*d*f^2 - 48*A^2*a^3*b*c^2*d*f^2 - 72*A^2*a
^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))
^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^
5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 64*A*a^2*c*d^11*f^4 + 64*A*b^2*c*d
^11*f^4 - 256*A*a^2*c^3*d^9*f^4 - 384*A*a^2*c^5*d^7*f^4 - 256*A*a^2*c^7*d^5
*f^4 - 64*A*a^2*c^9*d^3*f^4 + 256*A*b^2*c^3*d^9*f^4 + 384*A*b^2*c^5*d^7*f^4
 + 256*A*b^2*c^7*d^5*f^4 + 64*A*b^2*c^9*d^3*f^4 - 64*A*a*b*d^12*f^4 - 192*A
*a*b*c^2*d^10*f^4 - 128*A*a*b*c^4*d^8*f^4 + 128*A*a*b*c^6*d^6*f^4 + 192*A*a
*b*c^8*d^4*f^4 + 64*A*a*b*c^10*d^2*f^4))*((((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*
c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*
f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2
+ 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 1
6*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2
*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) - 4*A^2*a^4*c^3*f^2 - 4*A^2*b^
4*c^3*f^2 + 24*A^2*a^2*b^2*c^3*f^2 - 16*A^2*a*b^3*d^3*f^2 + 16*A^2*a^3*b*d^
3*f^2 + 12*A^2*a^4*c*d^2*f^2 + 12*A^2*b^4*c*d^2*f^2 + 48*A^2*a*b^3*c^2*d*f^
2 - 48*A^2*a^3*b*c^2*d*f^2 - 72*A^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f
^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2) - ((c + d*tan(e + f*x))^(1/2)*(
16*A^2*a^4*d^10*f^3 + 16*A^2*b^4*d^10*f^3 - 96*A^2*a^2*b^2*d^10*f^3 + 32*A^
2*a^4*c^2*d^8*f^3 - 32*A^2*a^4*c^6*d^4*f^3 - 16*A^2*a^4*c^8*d^2*f^3 + 32*A^
2*b^4*c^2*d^8*f^3 - 32*A^2*b^4*c^6*d^4*f^3 - 16*A^2*b^4*c^8*d^2*f^3 + 128*A
^2*a*b^3*c*d^9*f^3 - 128*A^2*a^3*b*c*d^9*f^3 + 384*A^2*a*b^3*c^3*d^7*f^3 +
384*A^2*a*b^3*c^5*d^5*f^3 + 128*A^2*a*b^3*c^7*d^3*f^3 - 384*A^2*a^3*b*c^3*d
^7*f^3 - 384*A^2*a^3*b*c^5*d^5*f^3 - 128*A^2*a^3*b*c^7*d^3*f^3 - 192*A^2*a^
2*b^2*c^2*d^8*f^3 + 192*A^2*a^2*b^2*c^6*d^4*f^3 + 96*A^2*a^2*b^2*c^8*d^2*f^
3) - ((((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 3
2*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*
b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a

^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*
d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2
))^(1/2) - 4*A^2*a^4*c^3*f^2 - 4*A^2*b^4*c^3*f^2 + 24*A^2*a^2*b^2*c^3*f^2 -
16*A^2*a*b^3*d^3*f^2 + 16*A^2*a^3*b*d^3*f^2 + 12*A^2*a^4*c*d^2*f^2 + 12*A^
2*b^4*c*d^2*f^2 + 48*A^2*a*b^3*c^2*d*f^2 - 48*A^2*a^3*b*c^2*d*f^2 - 72*A^2*
a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))
)^(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*c^3*f
^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*f^2 -
24*A^2*a^4*c*d^2*f^2 - 24*A^2*b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2 + 96*
A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6
*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6
+ 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) - 4*A^2*a^4*c^3*f^2 - 4*A^2*b^4*c^3
*f^2 + 24*A^2*a^2*b^2*c^3*f^2 - 16*A^2*a*b^3*d^3*f^2 + 16*A^2*a^3*b*d^3*f^2
+ 12*A^2*a^4*c*d^2*f^2 + 12*A^2*b^4*c*d^2*f^2 + 48*A^2*a*b^3*c^2*d*f^2 - 4
8*A^2*a^3*b*c^2*d*f^2 - 72*A^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 +
3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 +
640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) + 64
*A*a^2*c*d^11*f^4 - 64*A*b^2*c*d^11*f^4 + 256*A*a^2*c^3*d^9*f^4 + 384*A*a^2
*c^5*d^7*f^4 + 256*A*a^2*c^7*d^5*f^4 + 64*A*a^2*c^9*d^3*f^4 - 256*A*b^2*c^3
*d^9*f^4 - 384*A*b^2*c^5*d^7*f^4 - 256*A*b^2*c^7*d^5*f^4 - 64*A*b^2*c^9*d^3
*f^4 + 64*A*a*b*d^12*f^4 + 192*A*a*b*c^2*d^10*f^4 + 128*A*a*b*c^4*d^8*f^4 -
128*A*a*b*c^6*d^6*f^4 - 192*A*a*b*c^8*d^4*f^4 - 64*A*a*b*c^10*d^2*f^4))*((
((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32*A^2*a
*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*b^4*c*d
^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^2*b^2*
c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4
)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2
) - 4*A^2*a^4*c^3*f^2 - 4*A^2*b^4*c^3*f^2 + 24*A^2*a^2*b^2*c^3*f^2 - 16*A^2
*a*b^3*d^3*f^2 + 16*A^2*a^3*b*d^3*f^2 + 12*A^2*a^4*c*d^2*f^2 + 12*A^2*b^4*c
*d^2*f^2 + 48*A^2*a*b^3*c^2*d*f^2 - 48*A^2*a^3*b*c^2*d*f^2 - 72*A^2*a^2*b^2
*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)
- 16*A^3*a^6*d^9*f^2 + 16*A^3*b^6*d^9*f^2 + 16*A^3*a^2*b^4*d^9*f^2 - 16*A^
3*a^4*b^2*d^9*f^2 - 48*A^3*a^6*c^2*d^7*f^2 - 48*A^3*a^6*c^4*d^5*f^2 - 16*A^
3*a^6*c^6*d^3*f^2 + 48*A^3*b^6*c^2*d^7*f^2 + 48*A^3*b^6*c^4*d^5*f^2 + 16*A^
3*b^6*c^6*d^3*f^2 + 32*A^3*a*b^5*c*d^8*f^2 + 32*A^3*a^5*b*c*d^8*f^2 + 96*A^
3*a*b^5*c^3*d^6*f^2 + 96*A^3*a*b^5*c^5*d^4*f^2 + 32*A^3*a*b^5*c^7*d^2*f^2 +
64*A^3*a^3*b^3*c*d^8*f^2 + 96*A^3*a^5*b*c^3*d^6*f^2 + 96*A^3*a^5*b*c^5*d^4
*f^2 + 32*A^3*a^5*b*c^7*d^2*f^2 + 48*A^3*a^2*b^4*c^2*d^7*f^2 + 48*A^3*a^2*b
^4*c^4*d^5*f^2 + 16*A^3*a^2*b^4*c^6*d^3*f^2 + 192*A^3*a^3*b^3*c^3*d^6*f^2 +
192*A^3*a^3*b^3*c^5*d^4*f^2 + 64*A^3*a^3*b^3*c^7*d^2*f^2 - 48*A^3*a^4*b^2*
c^2*d^7*f^2 - 48*A^3*a^4*b^2*c^4*d^5*f^2 - 16*A^3*a^4*b^2*c^6*d^3*f^2))*(((
(8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32*A^2*a*
b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*b^4*c*d^
2*f^2 - 96*A^2*a*b^3*c^2*d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^2*b^2*c
*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)

*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2)
 - 4*A^2*a^4*c^3*f^2 - 4*A^2*b^4*c^3*f^2 + 24*A^2*a^2*b^2*c^3*f^2 - 16*A^2*
a*b^3*d^3*f^2 + 16*A^2*a^3*b*d^3*f^2 + 12*A^2*a^4*c*d^2*f^2 + 12*A^2*b^4*c*
d^2*f^2 + 48*A^2*a*b^3*c^2*d*f^2 - 48*A^2*a^3*b*c^2*d*f^2 - 72*A^2*a^2*b^2*
c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*
2i - atan(-(((c + d*tan(e + f*x))^(1/2)*(16*A^2*a^4*d^10*f^3 + 16*A^2*b^4*d
^10*f^3 - 96*A^2*a^2*b^2*d^10*f^3 + 32*A^2*a^4*c^2*d^8*f^3 - 32*A^2*a^4*c^6
*d^4*f^3 - 16*A^2*a^4*c^8*d^2*f^3 + 32*A^2*b^4*c^2*d^8*f^3 - 32*A^2*b^4*c^6
*d^4*f^3 - 16*A^2*b^4*c^8*d^2*f^3 + 128*A^2*a*b^3*c*d^9*f^3 - 128*A^2*a^3*b
*c*d^9*f^3 + 384*A^2*a*b^3*c^3*d^7*f^3 + 384*A^2*a*b^3*c^5*d^5*f^3 + 128*A^
2*a*b^3*c^7*d^3*f^3 - 384*A^2*a^3*b*c^3*d^7*f^3 - 384*A^2*a^3*b*c^5*d^5*f^3
 - 128*A^2*a^3*b*c^7*d^3*f^3 - 192*A^2*a^2*b^2*c^2*d^8*f^3 + 192*A^2*a^2*b^
2*c^6*d^4*f^3 + 96*A^2*a^2*b^2*c^8*d^2*f^3) - (-(((8*A^2*a^4*c^3*f^2 + 8*A^
2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*
b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*
d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f
^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A
^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) + 4*A^2*a^4*c^3*f^2 + 4*
A^2*b^4*c^3*f^2 - 24*A^2*a^2*b^2*c^3*f^2 + 16*A^2*a*b^3*d^3*f^2 - 16*A^2*a^
3*b*d^3*f^2 - 12*A^2*a^4*c*d^2*f^2 - 12*A^2*b^4*c*d^2*f^2 - 48*A^2*a*b^3*c^
2*d*f^2 + 48*A^2*a^3*b*c^2*d*f^2 + 72*A^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 +
 d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/
2)*(-(((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32
*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*b
^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^
2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d
^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)
)^(1/2) + 4*A^2*a^4*c^3*f^2 + 4*A^2*b^4*c^3*f^2 - 24*A^2*a^2*b^2*c^3*f^2 +
16*A^2*a*b^3*d^3*f^2 - 16*A^2*a^3*b*d^3*f^2 - 12*A^2*a^4*c*d^2*f^2 - 12*A^2
*b^4*c*d^2*f^2 - 48*A^2*a*b^3*c^2*d*f^2 + 48*A^2*a^3*b*c^2*d*f^2 + 72*A^2*a
^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))
^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^
5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 64*A*a^2*c*d^11*f^4 + 64*A*b^2*c*d
^11*f^4 - 256*A*a^2*c^3*d^9*f^4 - 384*A*a^2*c^5*d^7*f^4 - 256*A*a^2*c^7*d^5
*f^4 - 64*A*a^2*c^9*d^3*f^4 + 256*A*b^2*c^3*d^9*f^4 + 384*A*b^2*c^5*d^7*f^4
 + 256*A*b^2*c^7*d^5*f^4 + 64*A*b^2*c^9*d^3*f^4 - 64*A*a*b*d^12*f^4 - 192*A
*a*b*c^2*d^10*f^4 - 128*A*a*b*c^4*d^8*f^4 + 128*A*a*b*c^6*d^6*f^4 + 192*A*a
*b*c^8*d^4*f^4 + 64*A*a*b*c^10*d^2*f^4))*(-(((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4
*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3
*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2
 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 +
16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^
2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) + 4*A^2*a^4*c^3*f^2 + 4*A^2*b
^4*c^3*f^2 - 24*A^2*a^2*b^2*c^3*f^2 + 16*A^2*a*b^3*d^3*f^2 - 16*A^2*a^3*b*d
^3*f^2 - 12*A^2*a^4*c*d^2*f^2 - 12*A^2*b^4*c*d^2*f^2 - 48*A^2*a*b^3*c^2*d*f

^2 + 48*A^2*a^3*b*c^2*d*f^2 + 72*A^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*
f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*1i + ((c + d*tan(e + f*x))^(1/
2)*(16*A^2*a^4*d^10*f^3 + 16*A^2*b^4*d^10*f^3 - 96*A^2*a^2*b^2*d^10*f^3 + 3
2*A^2*a^4*c^2*d^8*f^3 - 32*A^2*a^4*c^6*d^4*f^3 - 16*A^2*a^4*c^8*d^2*f^3 + 3
2*A^2*b^4*c^2*d^8*f^3 - 32*A^2*b^4*c^6*d^4*f^3 - 16*A^2*b^4*c^8*d^2*f^3 + 1
28*A^2*a*b^3*c*d^9*f^3 - 128*A^2*a^3*b*c*d^9*f^3 + 384*A^2*a*b^3*c^3*d^7*f^
3 + 384*A^2*a*b^3*c^5*d^5*f^3 + 128*A^2*a*b^3*c^7*d^3*f^3 - 384*A^2*a^3*b*c
^3*d^7*f^3 - 384*A^2*a^3*b*c^5*d^5*f^3 - 128*A^2*a^3*b*c^7*d^3*f^3 - 192*A^
2*a^2*b^2*c^2*d^8*f^3 + 192*A^2*a^2*b^2*c^6*d^4*f^3 + 96*A^2*a^2*b^2*c^8*d^
2*f^3) - (-(((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^
2 + 32*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24
*A^2*b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*
A^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48
*c^4*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^
6*b^2))^(1/2) + 4*A^2*a^4*c^3*f^2 + 4*A^2*b^4*c^3*f^2 - 24*A^2*a^2*b^2*c^3*
f^2 + 16*A^2*a*b^3*d^3*f^2 - 16*A^2*a^3*b*d^3*f^2 - 12*A^2*a^4*c*d^2*f^2 -
12*A^2*b^4*c*d^2*f^2 - 48*A^2*a*b^3*c^2*d*f^2 + 48*A^2*a^3*b*c^2*d*f^2 + 72
*A^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*
f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4
*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3
*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2
 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 +
16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^
2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) + 4*A^2*a^4*c^3*f^2 + 4*A^2*b
^4*c^3*f^2 - 24*A^2*a^2*b^2*c^3*f^2 + 16*A^2*a*b^3*d^3*f^2 - 16*A^2*a^3*b*d
^3*f^2 - 12*A^2*a^4*c*d^2*f^2 - 12*A^2*b^4*c*d^2*f^2 - 48*A^2*a*b^3*c^2*d*f
^2 + 48*A^2*a^3*b*c^2*d*f^2 + 72*A^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*
f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*
f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5
) + 64*A*a^2*c*d^11*f^4 - 64*A*b^2*c*d^11*f^4 + 256*A*a^2*c^3*d^9*f^4 + 384
*A*a^2*c^5*d^7*f^4 + 256*A*a^2*c^7*d^5*f^4 + 64*A*a^2*c^9*d^3*f^4 - 256*A*b
^2*c^3*d^9*f^4 - 384*A*b^2*c^5*d^7*f^4 - 256*A*b^2*c^7*d^5*f^4 - 64*A*b^2*c
^9*d^3*f^4 + 64*A*a*b*d^12*f^4 + 192*A*a*b*c^2*d^10*f^4 + 128*A*a*b*c^4*d^8
*f^4 - 128*A*a*b*c^6*d^6*f^4 - 192*A*a*b*c^8*d^4*f^4 - 64*A*a*b*c^10*d^2*f^
4))*(-(((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 3
2*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*
b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a
^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*
d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2
))^(1/2) + 4*A^2*a^4*c^3*f^2 + 4*A^2*b^4*c^3*f^2 - 24*A^2*a^2*b^2*c^3*f^2 +
 16*A^2*a*b^3*d^3*f^2 - 16*A^2*a^3*b*d^3*f^2 - 12*A^2*a^4*c*d^2*f^2 - 12*A^
2*b^4*c*d^2*f^2 - 48*A^2*a*b^3*c^2*d*f^2 + 48*A^2*a^3*b*c^2*d*f^2 + 72*A^2*
a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))
)^(1/2)*1i)/(((c + d*tan(e + f*x))^(1/2)*(16*A^2*a^4*d^10*f^3 + 16*A^2*b^4*
d^10*f^3 - 96*A^2*a^2*b^2*d^10*f^3 + 32*A^2*a^4*c^2*d^8*f^3 - 32*A^2*a^4*c^

6*d^4*f^3 - 16*A^2*a^4*c^8*d^2*f^3 + 32*A^2*b^4*c^2*d^8*f^3 - 32*A^2*b^4*c^6*d^4*f^3 - 16*A^2*b^4*c^8*d^2*f^3 + 128*A^2*a*b^3*c*d^9*f^3 - 128*A^2*a^3*b*c*d^9*f^3 + 384*A^2*a*b^3*c^3*d^7*f^3 + 384*A^2*a*b^3*c^5*d^5*f^3 + 128*A^2*a*b^3*c^7*d^3*f^3 - 384*A^2*a^3*b*c^3*d^7*f^3 - 384*A^2*a^3*b*c^5*d^5*f^3 - 128*A^2*a^3*b*c^7*d^3*f^3 - 192*A^2*a^2*b^2*c^2*d^8*f^3 + 192*A^2*a^2*b^2*c^6*d^4*f^3 + 96*A^2*a^2*b^2*c^8*d^2*f^3) - (-(((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) + 4*A^2*a^4*c^3*f^2 + 4*A^2*b^4*c^3*f^2 - 24*A^2*a^2*b^2*c^3*f^2 + 16*A^2*a*b^3*d^3*f^2 - 16*A^2*a^3*b*d^3*f^2 - 12*A^2*a^4*c*d^2*f^2 - 12*A^2*b^4*c*d^2*f^2 - 48*A^2*a*b^3*c^2*d*f^2 + 48*A^2*a^3*b*c^2*d*f^2 + 72*A^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) + 4*A^2*a^4*c^3*f^2 + 4*A^2*b^4*c^3*f^2 - 24*A^2*a^2*b^2*c^3*f^2 + 16*A^2*a*b^3*d^3*f^2 - 16*A^2*a^3*b*d^3*f^2 - 12*A^2*a^4*c*d^2*f^2 - 12*A^2*b^4*c*d^2*f^2 - 48*A^2*a*b^3*c^2*d*f^2 + 48*A^2*a^3*b*c^2*d*f^2 + 72*A^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 64*A*a^2*c*d^11*f^4 + 64*A*b^2*c*d^11*f^4 - 256*A*a^2*c^3*d^9*f^4 - 384*A*a^2*c^5*d^7*f^4 - 256*A*a^2*c^7*d^5*f^4 - 64*A*a^2*c^9*d^3*f^4 + 256*A*b^2*c^3*d^9*f^4 + 384*A*b^2*c^5*d^7*f^4 + 256*A*b^2*c^7*d^5*f^4 + 64*A*b^2*c^9*d^3*f^4 - 64*A*a*b*d^12*f^4 - 192*A*a*b*c^2*d^10*f^4 - 128*A*a*b*c^4*d^8*f^4 + 128*A*a*b*c^6*d^6*f^4 + 192*A*a*b*c^8*d^4*f^4 + 64*A*a*b*c^10*d^2*f^4))*(-(((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) + 4*A^2*a^4*c^3*f^2 + 4*A^2*b^4*c^3*f^2 - 24*A^2*a^2*b^2*c^3*f^2 + 16*A^2*a*b^3*d^3*f^2 - 16*A^2*a^3*b*d^3*f^2 - 12*A^2*a^4*c*d^2*f^2 - 12*A^2*b^4*c*d^2*f^2 - 48*A^2*a*b^3*c^2*d*f^2 + 48*A^2*a^3*b*c^2*d*f^2 + 72*A^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2) - ((c + d*tan(e + f*x))^(1/2)*(16*A^2*a^4*d^10*f^3 + 16*A^2*b^4*d^10*f^3 - 96*A^2*a^2*b^2*d^10*f^3 + 32*A^2*a^4*c^2*d^8*f^3 - 32*A^2*a^4*c^6*d^4*f^3 - 16*A^2*a^4*c^8*d^2*f^3 + 32*A^2*b^4*c^2*d^8*f^3 - 32*A^2*b^4*c^6*d^4*f^3 - 16*A^2*b^4*c^8*d^2*f^3 + 128*A^2*a*b^3*c*d^9*f^3 - 128*A^2*a^3*b*c*d^9*f^3 + 384*A^2*a*b^3*c^3*d^7*f^3 + 384*A^2*a*b^3*c^5*d^5*f^3 + 128*A^2*a*b^3*c^7*d^3*f^3 - 384*A^2*a^3*b*c^3

*d^7*f^3 - 384*A^2*a^3*b*c^5*d^5*f^3 - 128*A^2*a^3*b*c^7*d^3*f^3 - 192*A^2*
a^2*b^2*c^2*d^8*f^3 + 192*A^2*a^2*b^2*c^6*d^4*f^3 + 96*A^2*a^2*b^2*c^8*d^2*
f^3) - (-(((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2
+ 32*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A
^2*b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^
2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c
^4*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*
b^2))^(1/2) + 4*A^2*a^4*c^3*f^2 + 4*A^2*b^4*c^3*f^2 - 24*A^2*a^2*b^2*c^3*f^
2 + 16*A^2*a*b^3*d^3*f^2 - 16*A^2*a^3*b*d^3*f^2 - 12*A^2*a^4*c*d^2*f^2 - 12
*A^2*b^4*c*d^2*f^2 - 48*A^2*a*b^3*c^2*d*f^2 + 48*A^2*a^3*b*c^2*d*f^2 + 72*A
^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^
4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*c
^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*f
^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2 +
96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16
*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*
b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) + 4*A^2*a^4*c^3*f^2 + 4*A^2*b^4
*c^3*f^2 - 24*A^2*a^2*b^2*c^3*f^2 + 16*A^2*a*b^3*d^3*f^2 - 16*A^2*a^3*b*d^3
*f^2 - 12*A^2*a^4*c*d^2*f^2 - 12*A^2*b^4*c*d^2*f^2 - 48*A^2*a*b^3*c^2*d*f^2
+ 48*A^2*a^3*b*c^2*d*f^2 + 72*A^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^
4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^
5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5)
+ 64*A*a^2*c*d^11*f^4 - 64*A*b^2*c*d^11*f^4 + 256*A*a^2*c^3*d^9*f^4 + 384*A
*a^2*c^5*d^7*f^4 + 256*A*a^2*c^7*d^5*f^4 + 64*A*a^2*c^9*d^3*f^4 - 256*A*b^2
*c^3*d^9*f^4 - 384*A*b^2*c^5*d^7*f^4 - 256*A*b^2*c^7*d^5*f^4 - 64*A*b^2*c^9
*d^3*f^4 + 64*A*a*b*d^12*f^4 + 192*A*a*b*c^2*d^10*f^4 + 128*A*a*b*c^4*d^8*f
^4 - 128*A*a*b*c^6*d^6*f^4 - 192*A*a*b*c^8*d^4*f^4 - 64*A*a*b*c^10*d^2*f^4)
)*(-(((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32*
A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*b^
4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^2
*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^
2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))
^(1/2) + 4*A^2*a^4*c^3*f^2 + 4*A^2*b^4*c^3*f^2 - 24*A^2*a^2*b^2*c^3*f^2 + 1
6*A^2*a*b^3*d^3*f^2 - 16*A^2*a^3*b*d^3*f^2 - 12*A^2*a^4*c*d^2*f^2 - 12*A^2*
b^4*c*d^2*f^2 - 48*A^2*a*b^3*c^2*d*f^2 + 48*A^2*a^3*b*c^2*d*f^2 + 72*A^2*a^
2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^
(1/2) - 16*A^3*a^6*d^9*f^2 + 16*A^3*b^6*d^9*f^2 + 16*A^3*a^2*b^4*d^9*f^2 -
16*A^3*a^4*b^2*d^9*f^2 - 48*A^3*a^6*c^2*d^7*f^2 - 48*A^3*a^6*c^4*d^5*f^2 -
16*A^3*a^6*c^6*d^3*f^2 + 48*A^3*b^6*c^2*d^7*f^2 + 48*A^3*b^6*c^4*d^5*f^2 +
16*A^3*b^6*c^6*d^3*f^2 + 32*A^3*a*b^5*c*d^8*f^2 + 32*A^3*a^5*b*c*d^8*f^2 +
96*A^3*a*b^5*c^3*d^6*f^2 + 96*A^3*a*b^5*c^5*d^4*f^2 + 32*A^3*a*b^5*c^7*d^2*
f^2 + 64*A^3*a^3*b^3*c*d^8*f^2 + 96*A^3*a^5*b*c^3*d^6*f^2 + 96*A^3*a^5*b*c^
5*d^4*f^2 + 32*A^3*a^5*b*c^7*d^2*f^2 + 48*A^3*a^2*b^4*c^2*d^7*f^2 + 48*A^3*
a^2*b^4*c^4*d^5*f^2 + 16*A^3*a^2*b^4*c^6*d^3*f^2 + 192*A^3*a^3*b^3*c^3*d^6*
f^2 + 192*A^3*a^3*b^3*c^5*d^4*f^2 + 64*A^3*a^3*b^3*c^7*d^2*f^2 - 48*A^3*a^4

*b^2*c^2*d^7*f^2 - 48*A^3*a^4*b^2*c^4*d^5*f^2 - 16*A^3*a^4*b^2*c^6*d^3*f^2)
)*(-(((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32*
A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*b^
4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^2
*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^
2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))
^(1/2) + 4*A^2*a^4*c^3*f^2 + 4*A^2*b^4*c^3*f^2 - 24*A^2*a^2*b^2*c^3*f^2 + 1
6*A^2*a*b^3*d^3*f^2 - 16*A^2*a^3*b*d^3*f^2 - 12*A^2*a^4*c*d^2*f^2 - 12*A^2*
b^4*c*d^2*f^2 - 48*A^2*a*b^3*c^2*d*f^2 + 48*A^2*a^3*b*c^2*d*f^2 + 72*A^2*a^
2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^
(1/2)*2i - ((8*C*b^2*c - 4*C*a*b*d)/(d^3*f) - (4*C*b^2*c)/(d^3*f))*(c + d*t
an(e + f*x))^(1/2) + (2*B*b^2*(c + d*tan(e + f*x))^(1/2))/(d^2*f) + (2*C*b^
2*(c + d*tan(e + f*x))^(3/2))/(3*d^3*f) - (2*(A*a^2*d^2 + A*b^2*c^2 - 2*A*a
*b*c*d))/(d*f*(c^2 + d^2)*(c + d*tan(e + f*x))^(1/2)) - (2*(C*b^2*c^4 + C*a
^2*c^2*d^2 - 2*C*a*b*c^3*d))/(d^3*f*(c^2 + d^2)*(c + d*tan(e + f*x))^(1/2))

3.118 $\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{3/2}}\,dx$

Optimal result 1680
Rubi [A] (verified) 1680
Mathematica [C] (verified) 1683
Maple [B] (verified) 1684
Fricas [B] (verification not implemented) 1684
Sympy [F] 1684
Maxima [F(-1)] 1685
Giac [F(-1)] 1685
Mupad [B] (verification not implemented) 1685

Optimal result

Integrand size = 45, antiderivative size = 201

$$\begin{aligned}\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{3/2}}\,dx =\\ -\frac{(ia+b)(A-iB-C)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(c-id)^{3/2}f}\\ +\frac{(ia-b)(A+iB-C)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(c+id)^{3/2}f}\\ +\frac{2(bc-ad)\left(c^2C-Bcd+Ad^2\right)}{d^2\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}+\frac{2bC\sqrt{c+d\tan(e+fx)}}{d^2f}\end{aligned}$$

```
[Out] -(I*a+b)*(A-I*B-C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(c-I*d)^(3
/2)/f+(I*a-b)*(A+I*B-C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))/(c+I*
d)^(3/2)/f+2*(-a*d+b*c)*(A*d^2-B*c*d+C*c^2)/d^2/(c^2+d^2)/f/(c+d*tan(f*x+e)
)^(1/2)+2*b*C*(c+d*tan(f*x+e))^(1/2)/d^2/f
```

Rubi [A] (verified)

Time = 0.64 (sec) , antiderivative size = 201, normalized size of antiderivative = 1.00, number of steps used = 9, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.133, Rules used

$= \{3716, 3711, 3620, 3618, 65, 214\}$

$$\int \frac{(a+b\tan(e+fx))\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx = -\frac{(b+ia)(A-iB-C)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f(c-id)^{3/2}} + \frac{(-b+ia)(A+iB-C)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f(c+id)^{3/2}} + \frac{2(bc-ad)\,(Ad^2-Bcd+c^2C)}{d^2f\,(c^2+d^2)\,\sqrt{c+d\tan(e+fx)}} + \frac{2bC\sqrt{c+d\tan(e+fx)}}{d^2f}$$

```
[In] Int[((a + b*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c + d*T
an[e + f*x])^(3/2),x]
```

```
[Out] -(((I*a + b)*(A - I*B - C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]])
/((c - I*d)^(3/2)*f)) + ((I*a - b)*(A + I*B - C)*ArcTanh[Sqrt[c + d*Tan[e +
 f*x]]/Sqrt[c + I*d]])/((c + I*d)^(3/2)*f) + (2*(b*c - a*d)*(c^2*C - B*c*d
+ A*d^2))/(d^2*(c^2 + d^2)*f*Sqrt[c + d*Tan[e + f*x]]) + (2*b*C*Sqrt[c + d*
Tan[e + f*x]])/(d^2*f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
```

```
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3716

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.
)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(-(b*c - a*d))*(c^2*C - B*c*d + A*d^2)*((c
+ d*Tan[e + f*x])^(n + 1)/(d^2*f*(n + 1)*(c^2 + d^2))), x] + Dist[1/(d*(c^2
 + d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)*Simp[a*d*(A*c - c*C + B*d) + b*(
c^2*C - B*c*d + A*d^2) + d*(A*b*c + a*B*c - b*c*C - a*A*d + b*B*d + a*C*d)*
Tan[e + f*x] + b*C*(c^2 + d^2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c
, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] && LtQ[n,
 -1]
```

Rubi steps

$$\text{integral} = \frac{2(bc-ad)\left(c^2C-Bcd+Ad^2\right)}{d^2\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}} + \frac{\int \frac{ad(Ac-cC+Bd)+b\left(c^2C-Bcd+Ad^2\right)+d(Abc+aBc-bcC-aAd+bBd+aCd)\tan(e+fx)+bC\left(c^2+d^2\right)\tan^2(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{d\left(c^2+d^2\right)}$$

$$= \frac{2(bc-ad)\left(c^2C-Bcd+Ad^2\right)}{d^2\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}} + \frac{2bC\sqrt{c+d\tan(e+fx)}}{d^2f} + \frac{\int \frac{d(a(Ac-cC+Bd)-b(Bc-(A-C)d))+d(Abc+aBc-bcC-aAd+bBd+aCd)\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{d\left(c^2+d^2\right)}$$

$$= \frac{2(bc-ad)\left(c^2C-Bcd+Ad^2\right)}{d^2\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}} + \frac{2bC\sqrt{c+d\tan(e+fx)}}{d^2f} + \frac{((a-ib)(A-iB-C))\int \frac{1+i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(c-id)} + \frac{((a+ib)(A+iB-C))\int \frac{1-i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(c+id)}$$

$$=\frac{2(bc-ad)\left(c^2C-Bcd+Ad^2\right)}{d^2\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}+\frac{2bC\sqrt{c+d\tan(e+fx)}}{d^2f}+\frac{(i(a-ib)(A-iB-C))\text{Subst}\left(\int\frac{1}{(-1+x)\sqrt{c-idx}}\,dx,x,i\tan(e+fx)\right)}{2(c-id)f}-\frac{((ia-b)(A+iB-C))\text{Subst}\left(\int\frac{1}{(-1+x)\sqrt{c+idx}}\,dx,x,-i\tan(e+fx)\right)}{2(c+id)f}$$

$$=\frac{2(bc-ad)\left(c^2C-Bcd+Ad^2\right)}{d^2\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}+\frac{2bC\sqrt{c+d\tan(e+fx)}}{d^2f}-\frac{((a-ib)(A-iB-C))\text{Subst}\left(\int\frac{1}{-1-\frac{ic}{d}+\frac{ix^2}{d}}\,dx,x,\sqrt{c+d\tan(e+fx)}\right)}{(c-id)df}-\frac{((a+ib)(A+iB-C))\text{Subst}\left(\int\frac{1}{-1+\frac{ic}{d}-\frac{ix^2}{d}}\,dx,x,\sqrt{c+d\tan(e+fx)}\right)}{(c+id)df}$$

$$=-\frac{(ia+b)(A-iB-C)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(c-id)^{3/2}f}+\frac{(ia-b)(A+iB-C)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(c+id)^{3/2}f}+\frac{2(bc-ad)\left(c^2C-Bcd+Ad^2\right)}{d^2\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}+\frac{2bC\sqrt{c+d\tan(e+fx)}}{d^2f}$$

Mathematica [C] (verified)

Result contains higher order function than in optimal. Order 5 vs. order 3 in optimal.

Time = 2.78 (sec) , antiderivative size = 290, normalized size of antiderivative = 1.44

$$\int\frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{3/2}}\,dx=\frac{(Ab+aB-bC)\left(-\frac{i\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+}}{\sqrt{c-id}}\right.}{\sqrt{c-id}}\right.}{}$$

```
[In] Integrate[((a + b*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c
 + d*Tan[e + f*x])^(3/2),x]
```

```
[Out] ((A*b + a*B - b*C)*(((-I)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]])/
Sqrt[c - I*d] + (I*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]])/Sqrt[c
+ I*d]) - (2*(-2*b*c*C + b*B*d + 2*a*C*d))/(d*Sqrt[c + d*Tan[e + f*x]]) + (
(A*b*c + a*B*c - b*c*C - a*A*d + b*B*d + a*C*d)*(((-I)*c + d)*Hypergeometri
c2F1[-1/2, 1, 1/2, (c + d*Tan[e + f*x])/(c - I*d)] + (I*c + d)*Hypergeometr
```

```
ic2F1[-1/2, 1, 1/2, (c + d*Tan[e + f*x])/(c + I*d)]))/((c^2 + d^2)*Sqrt[c +
 d*Tan[e + f*x]]) + (2*C*(a + b*Tan[e + f*x]))/Sqrt[c + d*Tan[e + f*x]])/(d
*f)
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 7395 vs. 2(177) = 354.

Time = 0.16 (sec) , antiderivative size = 7396, normalized size of antiderivative = 36.80

method	result	size
parts	Expression too large to display	7396
derivativedivides	Expression too large to display	23472
default	Expression too large to display	23472

```
[In] int((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(3/2)
,x,method=_RETURNVERBOSE)
```

```
[Out] result too large to display
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 31879 vs. 2(170) = 340.

Time = 61.21 (sec) , antiderivative size = 31879, normalized size of antiderivative = 158.60

$$\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{3/2}}\,dx = \text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(3/2),x, algorithm="fricas")
```

```
[Out] Too large to include
```

Sympy [F]

$$\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{3/2}}\,dx = \int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+f\right.}{(c+d\tan(e+fx)}$$

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x+e)
)**(3/2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/(c +
 d*tan(e + f*x))**(3/2), x)
```

Maxima [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(3/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

Giac [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(3/2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [B] (verification not implemented)

Time = 38.60 (sec) , antiderivative size = 40542, normalized size of antiderivative = 201.70

$$\int \frac{(a+b\tan(e+fx))\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx = \text{Too large to display}$$

```
[In] int(((a + b*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d*t
an(e + f*x))^(3/2),x)
```

```
[Out] atan((((c + d*tan(e + f*x))^(1/2)*(16*A^2*a^2*d^10*f^3 - 16*B^2*a^2*d^10*f^
3 + 16*C^2*a^2*d^10*f^3 + 32*A^2*a^2*c^2*d^8*f^3 - 32*A^2*a^2*c^6*d^4*f^3 -
 16*A^2*a^2*c^8*d^2*f^3 - 32*B^2*a^2*c^2*d^8*f^3 + 32*B^2*a^2*c^6*d^4*f^3 +
 16*B^2*a^2*c^8*d^2*f^3 + 32*C^2*a^2*c^2*d^8*f^3 - 32*C^2*a^2*c^6*d^4*f^3 -
 16*C^2*a^2*c^8*d^2*f^3 - 32*A*C*a^2*d^10*f^3 - 64*A*B*a^2*c*d^9*f^3 + 64*B
*C*a^2*c*d^9*f^3 - 192*A*B*a^2*c^3*d^7*f^3 - 192*A*B*a^2*c^5*d^5*f^3 - 64*A
*B*a^2*c^7*d^3*f^3 - 64*A*C*a^2*c^2*d^8*f^3 + 64*A*C*a^2*c^6*d^4*f^3 + 32*A
*C*a^2*c^8*d^2*f^3 + 192*B*C*a^2*c^3*d^7*f^3 + 192*B*C*a^2*c^5*d^5*f^3 + 64
*B*C*a^2*c^7*d^3*f^3) - ((((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*a
^2*c^3*f^2 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2 -
 24*A^2*a^2*c*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A*
B*a^2*c^2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c^
```

6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2 *C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*B^2*a^2*c^3*f^2 - 4*C^2*a^2*c^3*f^2 + 8*A*B*a^2*d^3*f^2 + 8*A*C*a^2*c^3*f^2 - 8*B*C*a^2*d^3*f ^2 + 12*A^2*a^2*c*d^2*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*C^2*a^2*c*d^2*f^2 - 2 4*A*B*a^2*c^2*d*f^2 - 24*A*C*a^2*c*d^2*f^2 + 24*B*C*a^2*c^2*d*f^2)/(16*(c^6 *f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*a^2*c^3*f^2 - 16 *A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2 - 24*A^2*a^2*c*d ^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A*B*a^2*c^2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6* f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C ^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A* B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*B^2*a^2*c^3*f^2 - 4*C^2*a^2*c^3*f ^2 + 8*A*B*a^2*d^3*f^2 + 8*A*C*a^2*c^3*f^2 - 8*B*C*a^2*d^3*f^2 + 12*A^2*a^2 *c*d^2*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*C^2*a^2*c*d^2*f^2 - 24*A*B*a^2*c^2*d *f^2 - 24*A*C*a^2*c*d^2*f^2 + 24*B*C*a^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 32*B*a*d^12*f^4 - 256*A*a*c^3*d^9*f^4 - 384*A*a*c^5*d^7*f^4 - 256*A*a*c^7*d ^5*f^4 - 64*A*a*c^9*d^3*f^4 - 96*B*a*c^2*d^10*f^4 - 64*B*a*c^4*d^8*f^4 + 64 *B*a*c^6*d^6*f^4 + 96*B*a*c^8*d^4*f^4 + 32*B*a*c^10*d^2*f^4 + 256*C*a*c^3*d ^9*f^4 + 384*C*a*c^5*d^7*f^4 + 256*C*a*c^7*d^5*f^4 + 64*C*a*c^9*d^3*f^4 - 6 4*A*a*c*d^11*f^4 + 64*C*a*c*d^11*f^4))*((((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^ 3*f^2 + 8*C^2*a^2*c^3*f^2 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B* C*a^2*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c* d^2*f^2 + 48*A*B*a^2*c^2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^ 2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a ^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2* C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*B^2 *a^2*c^3*f^2 - 4*C^2*a^2*c^3*f^2 + 8*A*B*a^2*d^3*f^2 + 8*A*C*a^2*c^3*f^2 - 8*B*C*a^2*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*C^2*a^ 2*c*d^2*f^2 - 24*A*B*a^2*c^2*d*f^2 - 24*A*C*a^2*c*d^2*f^2 + 24*B*C*a^2*c^2* d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*1i + ((c + d*tan(e + f*x))^(1/2)*(16*A^2*a^2*d^10*f^3 - 16*B^2*a^2*d^10*f^3 + 1 6*C^2*a^2*d^10*f^3 + 32*A^2*a^2*c^2*d^8*f^3 - 32*A^2*a^2*c^6*d^4*f^3 - 16*A ^2*a^2*c^8*d^2*f^3 - 32*B^2*a^2*c^2*d^8*f^3 + 32*B^2*a^2*c^6*d^4*f^3 + 16*B ^2*a^2*c^8*d^2*f^3 + 32*C^2*a^2*c^2*d^8*f^3 - 32*C^2*a^2*c^6*d^4*f^3 - 16*C ^2*a^2*c^8*d^2*f^3 - 32*A*C*a^2*d^10*f^3 - 64*A*B*a^2*c*d^9*f^3 + 64*B*C*a^ 2*c*d^9*f^3 - 192*A*B*a^2*c^3*d^7*f^3 - 192*A*B*a^2*c^5*d^5*f^3 - 64*A*B*a^ 2*c^7*d^3*f^3 - 64*A*C*a^2*c^2*d^8*f^3 + 64*A*C*a^2*c^6*d^4*f^3 + 32*A*C*a^ 2*c^8*d^2*f^3 + 192*B*C*a^2*c^3*d^7*f^3 + 192*B*C*a^2*c^5*d^5*f^3 + 64*B*C* a^2*c^7*d^3*f^3) - ((((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*a^2*c^ 3*f^2 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2 - 24*A ^2*a^2*c*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A*B*a^2

*c^2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c^6*f^4
 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a
^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*
a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*B^2*a^2*c^3*f^2 - 4*C^2
*a^2*c^3*f^2 + 8*A*B*a^2*d^3*f^2 + 8*A*C*a^2*c^3*f^2 - 8*B*C*a^2*d^3*f^2 +
12*A^2*a^2*c*d^2*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*C^2*a^2*c*d^2*f^2 - 24*A*B
*a^2*c^2*d*f^2 - 24*A*C*a^2*c*d^2*f^2 + 24*B*C*a^2*c^2*d*f^2)/(16*(c^6*f^4
+ d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1
/2)*((((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*a^2*c^3*f^2 - 16*A*B*
a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2 - 24*A^2*a^2*c*d^2*f^
2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A*B*a^2*c^2*d*f^2 + 48
*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 +
 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^
4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C
*a^4))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*B^2*a^2*c^3*f^2 - 4*C^2*a^2*c^3*f^2 +
8*A*B*a^2*d^3*f^2 + 8*A*C*a^2*c^3*f^2 - 8*B*C*a^2*d^3*f^2 + 12*A^2*a^2*c*d^
2*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*C^2*a^2*c*d^2*f^2 - 24*A*B*a^2*c^2*d*f^2
- 24*A*C*a^2*c*d^2*f^2 + 24*B*C*a^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c
^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640
*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) + 32*B*
a*d^12*f^4 + 256*A*a*c^3*d^9*f^4 + 384*A*a*c^5*d^7*f^4 + 256*A*a*c^7*d^5*f^
4 + 64*A*a*c^9*d^3*f^4 + 96*B*a*c^2*d^10*f^4 + 64*B*a*c^4*d^8*f^4 - 64*B*a*
c^6*d^6*f^4 - 96*B*a*c^8*d^4*f^4 - 32*B*a*c^10*d^2*f^4 - 256*C*a*c^3*d^9*f^
4 - 384*C*a*c^5*d^7*f^4 - 256*C*a*c^7*d^5*f^4 - 64*C*a*c^9*d^3*f^4 + 64*A*a
*c*d^11*f^4 - 64*C*a*c*d^11*f^4))*((((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2
 + 8*C^2*a^2*c^3*f^2 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2
*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f
^2 + 48*A*B*a^2*c^2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/
4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 +
B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a
^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*B^2*a^2*
c^3*f^2 - 4*C^2*a^2*c^3*f^2 + 8*A*B*a^2*d^3*f^2 + 8*A*C*a^2*c^3*f^2 - 8*B*C
*a^2*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*C^2*a^2*c*d
^2*f^2 - 24*A*B*a^2*c^2*d*f^2 - 24*A*C*a^2*c*d^2*f^2 + 24*B*C*a^2*c^2*d*f^2
)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*1i)/(((c
+ d*tan(e + f*x))^(1/2)*(16*A^2*a^2*d^10*f^3 - 16*B^2*a^2*d^10*f^3 + 16*C^2
*a^2*d^10*f^3 + 32*A^2*a^2*c^2*d^8*f^3 - 32*A^2*a^2*c^6*d^4*f^3 - 16*A^2*a^
2*c^8*d^2*f^3 - 32*B^2*a^2*c^2*d^8*f^3 + 32*B^2*a^2*c^6*d^4*f^3 + 16*B^2*a^
2*c^8*d^2*f^3 + 32*C^2*a^2*c^2*d^8*f^3 - 32*C^2*a^2*c^6*d^4*f^3 - 16*C^2*a^
2*c^8*d^2*f^3 - 32*A*C*a^2*d^10*f^3 - 64*A*B*a^2*c*d^9*f^3 + 64*B*C*a^2*c*d
^9*f^3 - 192*A*B*a^2*c^3*d^7*f^3 - 192*A*B*a^2*c^5*d^5*f^3 - 64*A*B*a^2*c^7
*d^3*f^3 - 64*A*C*a^2*c^2*d^8*f^3 + 64*A*C*a^2*c^6*d^4*f^3 + 32*A*C*a^2*c^8
*d^2*f^3 + 192*B*C*a^2*c^3*d^7*f^3 + 192*B*C*a^2*c^5*d^5*f^3 + 64*B*C*a^2*c
^7*d^3*f^3) - ((((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*a^2*c^3*f^2
 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2 - 24*A^2*a^

2*c*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A*B*a^2*c^2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*B^2*a^2*c^3*f^2 - 4*C^2*a^2*c^3*f^2 + 8*A*B*a^2*d^3*f^2 + 8*A*C*a^2*c^3*f^2 - 8*B*C*a^2*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*C^2*a^2*c*d^2*f^2 - 24*A*B*a^2*c^2*d*f^2 - 24*A*C*a^2*c*d^2*f^2 + 24*B*C*a^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*a^2*c^3*f^2 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A*B*a^2*c^2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*B^2*a^2*c^3*f^2 - 4*C^2*a^2*c^3*f^2 + 8*A*B*a^2*d^3*f^2 + 8*A*C*a^2*c^3*f^2 - 8*B*C*a^2*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*C^2*a^2*c*d^2*f^2 - 24*A*B*a^2*c^2*d*f^2 - 24*A*C*a^2*c*d^2*f^2 + 24*B*C*a^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 32*B*a*d^12*f^4 - 256*A*a*c^3*d^9*f^4 - 384*A*a*c^5*d^7*f^4 - 256*A*a*c^7*d^5*f^4 - 64*A*a*c^9*d^3*f^4 - 96*B*a*c^2*d^10*f^4 - 64*B*a*c^4*d^8*f^4 + 64*B*a*c^6*d^6*f^4 + 96*B*a*c^8*d^4*f^4 + 32*B*a*c^10*d^2*f^4 + 256*C*a*c^3*d^9*f^4 + 384*C*a*c^5*d^7*f^4 + 256*C*a*c^7*d^5*f^4 + 64*C*a*c^9*d^3*f^4 - 64*A*a*c*d^11*f^4 + 64*C*a*c*d^11*f^4))*((((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*a^2*c^3*f^2 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A*B*a^2*c^2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*B^2*a^2*c^3*f^2 - 4*C^2*a^2*c^3*f^2 + 8*A*B*a^2*d^3*f^2 + 8*A*C*a^2*c^3*f^2 - 8*B*C*a^2*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*C^2*a^2*c*d^2*f^2 - 24*A*B*a^2*c^2*d*f^2 - 24*A*C*a^2*c*d^2*f^2 + 24*B*C*a^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2) - ((c + d*tan(e + f*x))^(1/2)*(16*A^2*a^2*d^10*f^3 - 16*B^2*a^2*d^10*f^3 + 16*C^2*a^2*d^10*f^3 + 32*A^2*a^2*c^2*d^8*f^3 - 32*A^2*a^2*c^6*d^4*f^3 - 16*A^2*a^2*c^8*d^2*f^3 - 32*B^2*a^2*c^2*d^8*f^3 + 32*B^2*a^2*c^6*d^4*f^3 + 16*B^2*a^2*c^8*d^2*f^3 + 32*C^2*a^2*c^2*d^8*f^3 - 32*C^2*a^2*c^6*d^4*f^3 - 16*C^2*a^2*c^8*d^2*f^3 - 32*A*C*a^2*d^10*f^3 - 64*A*B*a^2*c*d^9*f^3 + 64*B*C*a^2*c*d^9*f^3 - 192*A*B*a^2*c^3*d^7*f^3 - 192*A*B*a^2*c^5*d^5*f^3 - 64*A*B*a^2*c^7*d^3*f^3 - 64*A*C*a^2*c^2*d^8*f^3 + 64*A*C*a^2*c^6*d^4*f^3 + 32*A*C*a^2*c^8*d^2*f^3 + 192*B*C*a^2*c^3*d^7*f^3 + 192*B*C*a^2*c^5*d^5*f^3 + 64*B*C*a^2*c^7*d^3*f^3) - ((((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*a^2*c^3*f^2 - 16*A*

B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2 - 24*A^2*a^2*c*d^2*
f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A*B*a^2*c^2*d*f^2 +
48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4
 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*
a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2
*C*a^4))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*B^2*a^2*c^3*f^2 - 4*C^2*a^2*c^3*f^2
+ 8*A*B*a^2*d^3*f^2 + 8*A*C*a^2*c^3*f^2 - 8*B*C*a^2*d^3*f^2 + 12*A^2*a^2*c*
d^2*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*C^2*a^2*c*d^2*f^2 - 24*A*B*a^2*c^2*d*f^
2 - 24*A*C*a^2*c*d^2*f^2 + 24*B*C*a^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3
*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*A^2
*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*a^2*c^3*f^2 - 16*A*B*a^2*d^3*f^2 -
 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*B^2*a^
2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A*B*a^2*c^2*d*f^2 + 48*A*C*a^2*c*d^
2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f
^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a
^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2)
- 4*A^2*a^2*c^3*f^2 + 4*B^2*a^2*c^3*f^2 - 4*C^2*a^2*c^3*f^2 + 8*A*B*a^2*d^3
*f^2 + 8*A*C*a^2*c^3*f^2 - 8*B*C*a^2*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*B^
2*a^2*c*d^2*f^2 + 12*C^2*a^2*c*d^2*f^2 - 24*A*B*a^2*c^2*d*f^2 - 24*A*C*a^2*
c*d^2*f^2 + 24*B*C*a^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 +
3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5
+ 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) + 32*B*a*d^12*f^4 +
256*A*a*c^3*d^9*f^4 + 384*A*a*c^5*d^7*f^4 + 256*A*a*c^7*d^5*f^4 + 64*A*a*c^
9*d^3*f^4 + 96*B*a*c^2*d^10*f^4 + 64*B*a*c^4*d^8*f^4 - 64*B*a*c^6*d^6*f^4 -
 96*B*a*c^8*d^4*f^4 - 32*B*a*c^10*d^2*f^4 - 256*C*a*c^3*d^9*f^4 - 384*C*a*c
^5*d^7*f^4 - 256*C*a*c^7*d^5*f^4 - 64*C*a*c^9*d^3*f^4 + 64*A*a*c*d^11*f^4 -
 64*C*a*c*d^11*f^4))*((((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*a^2*
c^3*f^2 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2 - 24
*A^2*a^2*c*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A*B*a
^2*c^2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c^6*f
^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4
*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^
2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*B^2*a^2*c^3*f^2 - 4*C
^2*a^2*c^3*f^2 + 8*A*B*a^2*d^3*f^2 + 8*A*C*a^2*c^3*f^2 - 8*B*C*a^2*d^3*f^2
+ 12*A^2*a^2*c*d^2*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*C^2*a^2*c*d^2*f^2 - 24*A
*B*a^2*c^2*d*f^2 - 24*A*C*a^2*c*d^2*f^2 + 24*B*C*a^2*c^2*d*f^2)/(16*(c^6*f^
4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2) - 16*A^3*a^3*d^9*f^2 +
 16*C^3*a^3*d^9*f^2 - 48*A^3*a^3*c^2*d^7*f^2 - 48*A^3*a^3*c^4*d^5*f^2 - 16*
A^3*a^3*c^6*d^3*f^2 + 48*B^3*a^3*c^3*d^6*f^2 + 48*B^3*a^3*c^5*d^4*f^2 + 16*
B^3*a^3*c^7*d^2*f^2 + 48*C^3*a^3*c^2*d^7*f^2 + 48*C^3*a^3*c^4*d^5*f^2 + 16*
C^3*a^3*c^6*d^3*f^2 - 16*A*B^2*a^3*d^9*f^2 - 48*A*C^2*a^3*d^9*f^2 + 48*A^2*
C*a^3*d^9*f^2 + 16*B^2*C*a^3*d^9*f^2 + 16*B^3*a^3*c*d^8*f^2 - 48*A*B^2*a^3*
c^2*d^7*f^2 - 48*A*B^2*a^3*c^4*d^5*f^2 - 16*A*B^2*a^3*c^6*d^3*f^2 + 48*A^2*
B*a^3*c^3*d^6*f^2 + 48*A^2*B*a^3*c^5*d^4*f^2 + 16*A^2*B*a^3*c^7*d^2*f^2 - 1
44*A*C^2*a^3*c^2*d^7*f^2 - 144*A*C^2*a^3*c^4*d^5*f^2 - 48*A*C^2*a^3*c^6*d^3

*f^2 + 144*A^2*C*a^3*c^2*d^7*f^2 + 144*A^2*C*a^3*c^4*d^5*f^2 + 48*A^2*C*a^3*c^6*d^3*f^2 + 48*B*C^2*a^3*c^3*d^6*f^2 + 48*B*C^2*a^3*c^5*d^4*f^2 + 16*B*C^2*a^3*c^7*d^2*f^2 + 48*B^2*C*a^3*c^2*d^7*f^2 + 48*B^2*C*a^3*c^4*d^5*f^2 + 16*B^2*C*a^3*c^6*d^3*f^2 + 16*A^2*B*a^3*c*d^8*f^2 + 16*B*C^2*a^3*c*d^8*f^2 - 96*A*B*C*a^3*c^3*d^6*f^2 - 96*A*B*C*a^3*c^5*d^4*f^2 - 32*A*B*C*a^3*c^7*d^2*f^2 - 32*A*B*C*a^3*c*d^8*f^2))*((((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*a^2*c^3*f^2 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A*B*a^2*c^2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*B^2*a^2*c^3*f^2 - 4*C^2*a^2*c^3*f^2 + 8*A*B*a^2*d^3*f^2 + 8*A*C*a^2*c^3*f^2 - 8*B*C*a^2*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*C^2*a^2*c*d^2*f^2 - 24*A*B*a^2*c^2*d*f^2 - 24*A*C*a^2*c*d^2*f^2 + 24*B*C*a^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*2i - atan((((((8*A^2*b^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 - 16*A*C*b^2*c^3*f^2 + 16*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 - 24*C^2*b^2*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2*f^2 - 48*B*C*b^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 - 8*A*B*b^2*d^3*f^2 - 8*A*C*b^2*c^3*f^2 + 8*B*C*b^2*d^3*f^2 - 12*A^2*b^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 + 24*A*B*b^2*c^2*d*f^2 + 24*A*C*b^2*c*d^2*f^2 - 24*B*C*b^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*A^2*b^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 - 16*A*C*b^2*c^3*f^2 + 16*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 - 24*C^2*b^2*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2*f^2 - 48*B*C*b^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 - 8*A*B*b^2*d^3*f^2 - 8*A*C*b^2*c^3*f^2 + 8*B*C*b^2*d^3*f^2 - 12*A^2*b^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 + 24*A*B*b^2*c^2*d*f^2 + 24*A*C*b^2*c*d^2*f^2 - 24*B*C*b^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 32*A*b*d^12*f^4 + 32*C*b*d^12*f^4 - 96*A*b*c^2*d^10*f^4 - 64*A*b*c^4*d^8*f^4 + 64*A*b*c^6*d^6*f^4 + 96*A*b*c^8*d^4*f^4 + 32*A*b*c^10*d^2*f^4 + 256*B*b*c^3*d^9*f^4 + 384*B*b*c^5*d^7*f^4 + 256*B*b*c^7*d^5*f^4 + 64*B*b*c^9*d^3*f^4 + 96*C*b*c^2*d^10*f^4 + 64*C*b*c^4*d^8*f^4 - 64*C*b*c^6*d^6*f^4 - 96*C*b*c^8*d^4*f^4 - 32*C*b*c^10*d^2*f^4 + 64*B*b*c*d^11*f^4) + (c + d*tan(e + f*x))^(1/2)*(16*A^2*b^2*d^10*f^3 - 16*B^2*b^2*d^10*f^3 + 16*C^2*b^2*d^10*f^3 + 32*A^2*b^2*c^2*d^8*f

^3 - 32*A^2*b^2*c^6*d^4*f^3 - 16*A^2*b^2*c^8*d^2*f^3 - 32*B^2*b^2*c^2*d^8*f
^3 + 32*B^2*b^2*c^6*d^4*f^3 + 16*B^2*b^2*c^8*d^2*f^3 + 32*C^2*b^2*c^2*d^8*f
^3 - 32*C^2*b^2*c^6*d^4*f^3 - 16*C^2*b^2*c^8*d^2*f^3 - 32*A*C*b^2*d^10*f^3
- 64*A*B*b^2*c*d^9*f^3 + 64*B*C*b^2*c*d^9*f^3 - 192*A*B*b^2*c^3*d^7*f^3 - 1
92*A*B*b^2*c^5*d^5*f^3 - 64*A*B*b^2*c^7*d^3*f^3 - 64*A*C*b^2*c^2*d^8*f^3 +
64*A*C*b^2*c^6*d^4*f^3 + 32*A*C*b^2*c^8*d^2*f^3 + 192*B*C*b^2*c^3*d^7*f^3 +
192*B*C*b^2*c^5*d^5*f^3 + 64*B*C*b^2*c^7*d^3*f^3))*((((8*A^2*b^2*c^3*f^2 -
8*B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 - 16*A*C*b^2*c^
3*f^2 + 16*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 -
24*C^2*b^2*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2*f^2 - 48*B*C
*b^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^
2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2
*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^
3*f^2 - 4*B^2*b^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 - 8*A*B*b^2*d^3*f^2 - 8*A*C*b
^2*c^3*f^2 + 8*B*C*b^2*d^3*f^2 - 12*A^2*b^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^
2 - 12*C^2*b^2*c*d^2*f^2 + 24*A*B*b^2*c^2*d*f^2 + 24*A*C*b^2*c*d^2*f^2 - 24
*B*C*b^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)
))^(1/2)*1i - (((((8*A^2*b^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^
2 - 16*A*B*b^2*d^3*f^2 - 16*A*C*b^2*c^3*f^2 + 16*B*C*b^2*d^3*f^2 - 24*A^2*b
^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 - 24*C^2*b^2*c*d^2*f^2 + 48*A*B*b^2*c^2
*d*f^2 + 48*A*C*b^2*c*d^2*f^2 - 48*B*C*b^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 1
6*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 -
4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4
- 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 + 4*C^2*b^2
*c^3*f^2 - 8*A*B*b^2*d^3*f^2 - 8*A*C*b^2*c^3*f^2 + 8*B*C*b^2*d^3*f^2 - 12*A
^2*b^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 + 24*A*B*b^2
*c^2*d*f^2 + 24*A*C*b^2*c*d^2*f^2 - 24*B*C*b^2*c^2*d*f^2)/(16*(c^6*f^4 + d^
6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(32*C*b*d^12*f^4 - 32*A*b*d^
12*f^4 - (c + d*tan(e + f*x))^(1/2)*((((8*A^2*b^2*c^3*f^2 - 8*B^2*b^2*c^3*f
^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 - 16*A*C*b^2*c^3*f^2 + 16*B*C*b
^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 - 24*C^2*b^2*c*d^2
*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2*f^2 - 48*B*C*b^2*c^2*d*f^2)^
2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*b^4
+ B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2
*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^3*f^2 - 4*B^2*b^
2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 - 8*A*B*b^2*d^3*f^2 - 8*A*C*b^2*c^3*f^2 + 8*B
*C*b^2*d^3*f^2 - 12*A^2*b^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 - 12*C^2*b^2*c
*d^2*f^2 + 24*A*B*b^2*c^2*d*f^2 + 24*A*C*b^2*c*d^2*f^2 - 24*B*C*b^2*c^2*d*f
^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d
^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^
4*f^5 + 64*c^11*d^2*f^5) - 96*A*b*c^2*d^10*f^4 - 64*A*b*c^4*d^8*f^4 + 64*A*
b*c^6*d^6*f^4 + 96*A*b*c^8*d^4*f^4 + 32*A*b*c^10*d^2*f^4 + 256*B*b*c^3*d^9*
f^4 + 384*B*b*c^5*d^7*f^4 + 256*B*b*c^7*d^5*f^4 + 64*B*b*c^9*d^3*f^4 + 96*C
*b*c^2*d^10*f^4 + 64*C*b*c^4*d^8*f^4 - 64*C*b*c^6*d^6*f^4 - 96*C*b*c^8*d^4*
f^4 - 32*C*b*c^10*d^2*f^4 + 64*B*b*c*d^11*f^4) - (c + d*tan(e + f*x))^(1/2)

*(16*A^2*b^2*d^10*f^3 - 16*B^2*b^2*d^10*f^3 + 16*C^2*b^2*d^10*f^3 + 32*A^2*
b^2*c^2*d^8*f^3 - 32*A^2*b^2*c^6*d^4*f^3 - 16*A^2*b^2*c^8*d^2*f^3 - 32*B^2*
b^2*c^2*d^8*f^3 + 32*B^2*b^2*c^6*d^4*f^3 + 16*B^2*b^2*c^8*d^2*f^3 + 32*C^2*
b^2*c^2*d^8*f^3 - 32*C^2*b^2*c^6*d^4*f^3 - 16*C^2*b^2*c^8*d^2*f^3 - 32*A*C*
b^2*d^10*f^3 - 64*A*B*b^2*c*d^9*f^3 + 64*B*C*b^2*c*d^9*f^3 - 192*A*B*b^2*c^
3*d^7*f^3 - 192*A*B*b^2*c^5*d^5*f^3 - 64*A*B*b^2*c^7*d^3*f^3 - 64*A*C*b^2*c
^2*d^8*f^3 + 64*A*C*b^2*c^6*d^4*f^3 + 32*A*C*b^2*c^8*d^2*f^3 + 192*B*C*b^2*
c^3*d^7*f^3 + 192*B*C*b^2*c^5*d^5*f^3 + 64*B*C*b^2*c^7*d^3*f^3))*((((8*A^2*
b^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 -
16*A*C*b^2*c^3*f^2 + 16*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2
*c*d^2*f^2 - 24*C^2*b^2*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2
*f^2 - 48*B*C*b^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^
4 + 48*c^4*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^
4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) +
4*A^2*b^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 - 8*A*B*b^2*d^3*
f^2 - 8*A*C*b^2*c^3*f^2 + 8*B*C*b^2*d^3*f^2 - 12*A^2*b^2*c*d^2*f^2 + 12*B^2
*b^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 + 24*A*B*b^2*c^2*d*f^2 + 24*A*C*b^2*c
*d^2*f^2 - 24*B*C*b^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3
*c^4*d^2*f^4)))^(1/2)*1i)/(16*B^3*b^3*d^9*f^2 - (((((8*A^2*b^2*c^3*f^2 - 8*
B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 - 16*A*C*b^2*c^3*f
^2 + 16*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 - 24*
C^2*b^2*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2*f^2 - 48*B*C*b^
2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f
^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^
4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^3*f
^2 - 4*B^2*b^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 - 8*A*B*b^2*d^3*f^2 - 8*A*C*b^2*
c^3*f^2 + 8*B*C*b^2*d^3*f^2 - 12*A^2*b^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 -
12*C^2*b^2*c*d^2*f^2 + 24*A*B*b^2*c^2*d*f^2 + 24*A*C*b^2*c*d^2*f^2 - 24*B*
C*b^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^
(1/2)*(32*C*b*d^12*f^4 - 32*A*b*d^12*f^4 - (c + d*tan(e + f*x))^(1/2)*((((8
*A^2*b^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f
^2 - 16*A*C*b^2*c^3*f^2 + 16*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^
2*b^2*c*d^2*f^2 - 24*C^2*b^2*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*
c*d^2*f^2 - 48*B*C*b^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d
^4*f^4 + 48*c^4*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3
*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1
/2) + 4*A^2*b^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 - 8*A*B*b^2
*d^3*f^2 - 8*A*C*b^2*c^3*f^2 + 8*B*C*b^2*d^3*f^2 - 12*A^2*b^2*c*d^2*f^2 + 1
2*B^2*b^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 + 24*A*B*b^2*c^2*d*f^2 + 24*A*C*
b^2*c*d^2*f^2 - 24*B*C*b^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^
4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*
f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 96*A*b*c^2*d^1
0*f^4 - 64*A*b*c^4*d^8*f^4 + 64*A*b*c^6*d^6*f^4 + 96*A*b*c^8*d^4*f^4 + 32*A
*b*c^10*d^2*f^4 + 256*B*b*c^3*d^9*f^4 + 384*B*b*c^5*d^7*f^4 + 256*B*b*c^7*d
^5*f^4 + 64*B*b*c^9*d^3*f^4 + 96*C*b*c^2*d^10*f^4 + 64*C*b*c^4*d^8*f^4 - 64

*C*b*c^6*d^6*f^4 - 96*C*b*c^8*d^4*f^4 - 32*C*b*c^10*d^2*f^4 + 64*B*b*c*d^11
*f^4) - (c + d*tan(e + f*x))^(1/2)*(16*A^2*b^2*d^10*f^3 - 16*B^2*b^2*d^10*f
^3 + 16*C^2*b^2*d^10*f^3 + 32*A^2*b^2*c^2*d^8*f^3 - 32*A^2*b^2*c^6*d^4*f^3
- 16*A^2*b^2*c^8*d^2*f^3 - 32*B^2*b^2*c^2*d^8*f^3 + 32*B^2*b^2*c^6*d^4*f^3
+ 16*B^2*b^2*c^8*d^2*f^3 + 32*C^2*b^2*c^2*d^8*f^3 - 32*C^2*b^2*c^6*d^4*f^3
- 16*C^2*b^2*c^8*d^2*f^3 - 32*A*C*b^2*d^10*f^3 - 64*A*B*b^2*c*d^9*f^3 + 64*
B*C*b^2*c*d^9*f^3 - 192*A*B*b^2*c^3*d^7*f^3 - 192*A*B*b^2*c^5*d^5*f^3 - 64*
A*B*b^2*c^7*d^3*f^3 - 64*A*C*b^2*c^2*d^8*f^3 + 64*A*C*b^2*c^6*d^4*f^3 + 32*
A*C*b^2*c^8*d^2*f^3 + 192*B*C*b^2*c^3*d^7*f^3 + 192*B*C*b^2*c^5*d^5*f^3 + 6
4*B*C*b^2*c^7*d^3*f^3))*((((8*A^2*b^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 + 8*C^2*b
^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 - 16*A*C*b^2*c^3*f^2 + 16*B*C*b^2*d^3*f^2 -
24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 - 24*C^2*b^2*c*d^2*f^2 + 48*A*
B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2*f^2 - 48*B*C*b^2*c^2*d*f^2)^2/4 - (16*c^
6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*b^4 + B^4*b^4 +
C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2
*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 +
4*C^2*b^2*c^3*f^2 - 8*A*B*b^2*d^3*f^2 - 8*A*C*b^2*c^3*f^2 + 8*B*C*b^2*d^3*f
^2 - 12*A^2*b^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 + 2
4*A*B*b^2*c^2*d*f^2 + 24*A*C*b^2*c*d^2*f^2 - 24*B*C*b^2*c^2*d*f^2)/(16*(c^6
*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2) - (((((8*A^2*b^2*c^
3*f^2 - 8*B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 - 16*A*C
*b^2*c^3*f^2 + 16*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2
*f^2 - 24*C^2*b^2*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2*f^2 -
48*B*C*b^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48
*c^4*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*
A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2
*b^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 - 8*A*B*b^2*d^3*f^2 -
8*A*C*b^2*c^3*f^2 + 8*B*C*b^2*d^3*f^2 - 12*A^2*b^2*c*d^2*f^2 + 12*B^2*b^2*c
*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 + 24*A*B*b^2*c^2*d*f^2 + 24*A*C*b^2*c*d^2*f
^2 - 24*B*C*b^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d
^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*A^2*b^2*c^3*f^2 - 8*B^2*b
^2*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 - 16*A*C*b^2*c^3*f^2 +
16*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 - 24*C^2*b
^2*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2*f^2 - 48*B*C*b^2*c^2
*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(
A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6
*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^3*f^2 -
4*B^2*b^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 - 8*A*B*b^2*d^3*f^2 - 8*A*C*b^2*c^3*f
^2 + 8*B*C*b^2*d^3*f^2 - 12*A^2*b^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 - 12*C
^2*b^2*c*d^2*f^2 + 24*A*B*b^2*c^2*d*f^2 + 24*A*C*b^2*c*d^2*f^2 - 24*B*C*b^2
*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)
*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 32
0*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 32*A*b*d^12*f^4 + 32*C*b*d^12*f^4 - 96*A
*b*c^2*d^10*f^4 - 64*A*b*c^4*d^8*f^4 + 64*A*b*c^6*d^6*f^4 + 96*A*b*c^8*d^4*
f^4 + 32*A*b*c^10*d^2*f^4 + 256*B*b*c^3*d^9*f^4 + 384*B*b*c^5*d^7*f^4 + 256

*B*b*c^7*d^5*f^4 + 64*B*b*c^9*d^3*f^4 + 96*C*b*c^2*d^10*f^4 + 64*C*b*c^4*d^8*f^4 - 64*C*b*c^6*d^6*f^4 - 96*C*b*c^8*d^4*f^4 - 32*C*b*c^10*d^2*f^4 + 64*B*b*c*d^11*f^4) + (c + d*tan(e + f*x))^(1/2)*(16*A^2*b^2*d^10*f^3 - 16*B^2*b^2*d^10*f^3 + 16*C^2*b^2*d^10*f^3 + 32*A^2*b^2*c^2*d^8*f^3 - 32*A^2*b^2*c^6*d^4*f^3 - 16*A^2*b^2*c^8*d^2*f^3 - 32*B^2*b^2*c^2*d^8*f^3 + 32*B^2*b^2*c^6*d^4*f^3 + 16*B^2*b^2*c^8*d^2*f^3 + 32*C^2*b^2*c^2*d^8*f^3 - 32*C^2*b^2*c^6*d^4*f^3 - 16*C^2*b^2*c^8*d^2*f^3 - 32*A*C*b^2*d^10*f^3 - 64*A*B*b^2*c*d^9*f^3 + 64*B*C*b^2*c*d^9*f^3 - 192*A*B*b^2*c^3*d^7*f^3 - 192*A*B*b^2*c^5*d^5*f^3 - 64*A*B*b^2*c^7*d^3*f^3 - 64*A*C*b^2*c^2*d^8*f^3 + 64*A*C*b^2*c^6*d^4*f^3 + 32*A*C*b^2*c^8*d^2*f^3 + 192*B*C*b^2*c^3*d^7*f^3 + 192*B*C*b^2*c^5*d^5*f^3 + 64*B*C*b^2*c^7*d^3*f^3))*((((8*A^2*b^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 - 16*A*C*b^2*c^3*f^2 + 16*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 - 24*C^2*b^2*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2*f^2 - 48*B*C*b^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 - 8*A*B*b^2*d^3*f^2 - 8*A*C*b^2*c^3*f^2 + 8*B*C*b^2*d^3*f^2 - 12*A^2*b^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 + 24*A*B*b^2*c^2*d*f^2 + 24*A*C*b^2*c*d^2*f^2 - 24*B*C*b^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2) + 48*A^3*b^3*c^3*d^6*f^2 + 48*A^3*b^3*c^5*d^4*f^2 + 16*A^3*b^3*c^7*d^2*f^2 + 48*B^3*b^3*c^2*d^7*f^2 + 48*B^3*b^3*c^4*d^5*f^2 + 16*B^3*b^3*c^6*d^3*f^2 - 48*C^3*b^3*c^3*d^6*f^2 - 48*C^3*b^3*c^5*d^4*f^2 - 16*C^3*b^3*c^7*d^2*f^2 + 16*A^2*B*b^3*d^9*f^2 + 16*B*C^2*b^3*d^9*f^2 + 16*A^3*b^3*c*d^8*f^2 - 16*C^3*b^3*c*d^8*f^2 + 48*A*B^2*b^3*c^3*d^6*f^2 + 48*A*B^2*b^3*c^5*d^4*f^2 + 16*A*B^2*b^3*c^7*d^2*f^2 + 48*A^2*B*b^3*c^2*d^7*f^2 + 48*A^2*B*b^3*c^4*d^5*f^2 + 16*A^2*B*b^3*c^6*d^3*f^2 + 144*A*C^2*b^3*c^3*d^6*f^2 + 144*A*C^2*b^3*c^5*d^4*f^2 + 48*A*C^2*b^3*c^7*d^2*f^2 - 144*A^2*C*b^3*c^3*d^6*f^2 - 144*A^2*C*b^3*c^5*d^4*f^2 - 48*A^2*C*b^3*c^7*d^2*f^2 + 48*B*C^2*b^3*c^2*d^7*f^2 + 48*B*C^2*b^3*c^4*d^5*f^2 + 16*B*C^2*b^3*c^6*d^3*f^2 - 48*B^2*C*b^3*c^3*d^6*f^2 - 48*B^2*C*b^3*c^5*d^4*f^2 - 16*B^2*C*b^3*c^7*d^2*f^2 - 32*A*B*C*b^3*d^9*f^2 + 16*A*B^2*b^3*c*d^8*f^2 + 48*A*C^2*b^3*c*d^8*f^2 - 48*A^2*C*b^3*c*d^8*f^2 - 16*B^2*C*b^3*c*d^8*f^2 - 96*A*B*C*b^3*c^2*d^7*f^2 - 96*A*B*C*b^3*c^4*d^5*f^2 - 32*A*B*C*b^3*c^6*d^3*f^2))*((((8*A^2*b^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 - 16*A*C*b^2*c^3*f^2 + 16*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 - 24*C^2*b^2*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2*f^2 - 48*B*C*b^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 - 8*A*B*b^2*d^3*f^2 - 8*A*C*b^2*c^3*f^2 + 8*B*C*b^2*d^3*f^2 - 12*A^2*b^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 + 24*A*B*b^2*c^2*d*f^2 + 24*A*C*b^2*c*d^2*f^2 - 24*B*C*b^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*2i - atan((((-

(((8*A^2*b^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d
^3*f^2 - 16*A*C*b^2*c^3*f^2 + 16*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 2
4*B^2*b^2*c*d^2*f^2 - 24*C^2*b^2*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*
b^2*c*d^2*f^2 - 48*B*C*b^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c
^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4
*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4)
)^(1/2) - 4*A^2*b^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 + 8*A*B
*b^2*d^3*f^2 + 8*A*C*b^2*c^3*f^2 - 8*B*C*b^2*d^3*f^2 + 12*A^2*b^2*c*d^2*f^2
 - 12*B^2*b^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 - 24*A*B*b^2*c^2*d*f^2 - 24*
A*C*b^2*c*d^2*f^2 + 24*B*C*b^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^
4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*b^2*c
^3*f^2 - 8*B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 - 16*A*
C*b^2*c^3*f^2 + 16*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2*c*d^
2*f^2 - 24*C^2*b^2*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2*f^2
- 48*B*C*b^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 4
8*c^4*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2
*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^
2*b^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 + 8*A*B*b^2*d^3*f^2 +
 8*A*C*b^2*c^3*f^2 - 8*B*C*b^2*d^3*f^2 + 12*A^2*b^2*c*d^2*f^2 - 12*B^2*b^2*
c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 - 24*A*B*b^2*c^2*d*f^2 - 24*A*C*b^2*c*d^2*
f^2 + 24*B*C*b^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*
d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*
c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 32*A*b*d^12*f^4 + 32*C*b
*d^12*f^4 - 96*A*b*c^2*d^10*f^4 - 64*A*b*c^4*d^8*f^4 + 64*A*b*c^6*d^6*f^4 +
 96*A*b*c^8*d^4*f^4 + 32*A*b*c^10*d^2*f^4 + 256*B*b*c^3*d^9*f^4 + 384*B*b*c
^5*d^7*f^4 + 256*B*b*c^7*d^5*f^4 + 64*B*b*c^9*d^3*f^4 + 96*C*b*c^2*d^10*f^4
 + 64*C*b*c^4*d^8*f^4 - 64*C*b*c^6*d^6*f^4 - 96*C*b*c^8*d^4*f^4 - 32*C*b*c^
10*d^2*f^4 + 64*B*b*c*d^11*f^4) + (c + d*tan(e + f*x))^(1/2)*(16*A^2*b^2*d^
10*f^3 - 16*B^2*b^2*d^10*f^3 + 16*C^2*b^2*d^10*f^3 + 32*A^2*b^2*c^2*d^8*f^3
 - 32*A^2*b^2*c^6*d^4*f^3 - 16*A^2*b^2*c^8*d^2*f^3 - 32*B^2*b^2*c^2*d^8*f^3
 + 32*B^2*b^2*c^6*d^4*f^3 + 16*B^2*b^2*c^8*d^2*f^3 + 32*C^2*b^2*c^2*d^8*f^3
 - 32*C^2*b^2*c^6*d^4*f^3 - 16*C^2*b^2*c^8*d^2*f^3 - 32*A*C*b^2*d^10*f^3 -
64*A*B*b^2*c*d^9*f^3 + 64*B*C*b^2*c*d^9*f^3 - 192*A*B*b^2*c^3*d^7*f^3 - 192
*A*B*b^2*c^5*d^5*f^3 - 64*A*B*b^2*c^7*d^3*f^3 - 64*A*C*b^2*c^2*d^8*f^3 + 64
*A*C*b^2*c^6*d^4*f^3 + 32*A*C*b^2*c^8*d^2*f^3 + 192*B*C*b^2*c^3*d^7*f^3 + 1
92*B*C*b^2*c^5*d^5*f^3 + 64*B*C*b^2*c^7*d^3*f^3))*(-(((8*A^2*b^2*c^3*f^2 -
8*B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 - 16*A*C*b^2*c^3
*f^2 + 16*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 - 2
4*C^2*b^2*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2*f^2 - 48*B*C*
b^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2
f^4)(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*
b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c^3
*f^2 + 4*B^2*b^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 + 8*A*B*b^2*d^3*f^2 + 8*A*C*b^
2*c^3*f^2 - 8*B*C*b^2*d^3*f^2 + 12*A^2*b^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2
 + 12*C^2*b^2*c*d^2*f^2 - 24*A*B*b^2*c^2*d*f^2 - 24*A*C*b^2*c*d^2*f^2 + 24*

B*C*b^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))
)^(1/2)*1i - ((-(((8*A^2*b^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^
2 - 16*A*B*b^2*d^3*f^2 - 16*A*C*b^2*c^3*f^2 + 16*B*C*b^2*d^3*f^2 - 24*A^2*b
^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 - 24*C^2*b^2*c*d^2*f^2 + 48*A*B*b^2*c^2
*d*f^2 + 48*A*C*b^2*c*d^2*f^2 - 48*B*C*b^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 1
6*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 -
 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4
- 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 - 4*C^2*b^2
*c^3*f^2 + 8*A*B*b^2*d^3*f^2 + 8*A*C*b^2*c^3*f^2 - 8*B*C*b^2*d^3*f^2 + 12*A
^2*b^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 - 24*A*B*b^2
*c^2*d*f^2 - 24*A*C*b^2*c*d^2*f^2 + 24*B*C*b^2*c^2*d*f^2)/(16*(c^6*f^4 + d^
6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(32*C*b*d^12*f^4 - 32*A*b*d^
12*f^4 - (c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*b^2*c^3*f^2 - 8*B^2*b^2*c^3*
f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 - 16*A*C*b^2*c^3*f^2 + 16*B*C*
b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 - 24*C^2*b^2*c*d^
2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2*f^2 - 48*B*C*b^2*c^2*d*f^2)
^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*b^4
 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^
2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c^3*f^2 + 4*B^2*b
^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 + 8*A*B*b^2*d^3*f^2 + 8*A*C*b^2*c^3*f^2 - 8*
B*C*b^2*d^3*f^2 + 12*A^2*b^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 + 12*C^2*b^2*
c*d^2*f^2 - 24*A*B*b^2*c^2*d*f^2 - 24*A*C*b^2*c*d^2*f^2 + 24*B*C*b^2*c^2*d*
f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*
d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d
^4*f^5 + 64*c^11*d^2*f^5) - 96*A*b*c^2*d^10*f^4 - 64*A*b*c^4*d^8*f^4 + 64*A
*b*c^6*d^6*f^4 + 96*A*b*c^8*d^4*f^4 + 32*A*b*c^10*d^2*f^4 + 256*B*b*c^3*d^9
*f^4 + 384*B*b*c^5*d^7*f^4 + 256*B*b*c^7*d^5*f^4 + 64*B*b*c^9*d^3*f^4 + 96*
C*b*c^2*d^10*f^4 + 64*C*b*c^4*d^8*f^4 - 64*C*b*c^6*d^6*f^4 - 96*C*b*c^8*d^4
*f^4 - 32*C*b*c^10*d^2*f^4 + 64*B*b*c*d^11*f^4) - (c + d*tan(e + f*x))^(1/2
)*(16*A^2*b^2*d^10*f^3 - 16*B^2*b^2*d^10*f^3 + 16*C^2*b^2*d^10*f^3 + 32*A^2
*b^2*c^2*d^8*f^3 - 32*A^2*b^2*c^6*d^4*f^3 - 16*A^2*b^2*c^8*d^2*f^3 - 32*B^2
*b^2*c^2*d^8*f^3 + 32*B^2*b^2*c^6*d^4*f^3 + 16*B^2*b^2*c^8*d^2*f^3 + 32*C^2
*b^2*c^2*d^8*f^3 - 32*C^2*b^2*c^6*d^4*f^3 - 16*C^2*b^2*c^8*d^2*f^3 - 32*A*C
*b^2*d^10*f^3 - 64*A*B*b^2*c*d^9*f^3 + 64*B*C*b^2*c*d^9*f^3 - 192*A*B*b^2*c
^3*d^7*f^3 - 192*A*B*b^2*c^5*d^5*f^3 - 64*A*B*b^2*c^7*d^3*f^3 - 64*A*C*b^2*
c^2*d^8*f^3 + 64*A*C*b^2*c^6*d^4*f^3 + 32*A*C*b^2*c^8*d^2*f^3 + 192*B*C*b^2
*c^3*d^7*f^3 + 192*B*C*b^2*c^5*d^5*f^3 + 64*B*C*b^2*c^7*d^3*f^3))*(-(((8*A^
2*b^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2
- 16*A*C*b^2*c^3*f^2 + 16*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b
^2*c*d^2*f^2 - 24*C^2*b^2*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d
^2*f^2 - 48*B*C*b^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*
f^4 + 48*c^4*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*
b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2)
 - 4*A^2*b^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 + 8*A*B*b^2*d^
3*f^2 + 8*A*C*b^2*c^3*f^2 - 8*B*C*b^2*d^3*f^2 + 12*A^2*b^2*c*d^2*f^2 - 12*B

^2*b^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 - 24*A*B*b^2*c^2*d*f^2 - 24*A*C*b^2
*c*d^2*f^2 + 24*B*C*b^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 +
3*c^4*d^2*f^4)))^(1/2)*1i)/(16*B^3*b^3*d^9*f^2 - ((-(((8*A^2*b^2*c^3*f^2 -
8*B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 - 16*A*C*b^2*c^
3*f^2 + 16*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 -
24*C^2*b^2*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2*f^2 - 48*B*C
*b^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^
2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2
*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c^
3*f^2 + 4*B^2*b^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 + 8*A*B*b^2*d^3*f^2 + 8*A*C*b
^2*c^3*f^2 - 8*B*C*b^2*d^3*f^2 + 12*A^2*b^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^
2 + 12*C^2*b^2*c*d^2*f^2 - 24*A*B*b^2*c^2*d*f^2 - 24*A*C*b^2*c*d^2*f^2 + 24
*B*C*b^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)
))^(1/2)*(32*C*b*d^12*f^4 - 32*A*b*d^12*f^4 - (c + d*tan(e + f*x))^(1/2)*(-
(((8*A^2*b^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d
^3*f^2 - 16*A*C*b^2*c^3*f^2 + 16*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 2
4*B^2*b^2*c*d^2*f^2 - 24*C^2*b^2*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*
b^2*c*d^2*f^2 - 48*B*C*b^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c
^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4
*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4)
)^(1/2) - 4*A^2*b^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 + 8*A*B
*b^2*d^3*f^2 + 8*A*C*b^2*c^3*f^2 - 8*B*C*b^2*d^3*f^2 + 12*A^2*b^2*c*d^2*f^2
- 12*B^2*b^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 - 24*A*B*b^2*c^2*d*f^2 - 24*
A*C*b^2*c*d^2*f^2 + 24*B*C*b^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^
4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*
d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 96*A*b*c^2
*d^10*f^4 - 64*A*b*c^4*d^8*f^4 + 64*A*b*c^6*d^6*f^4 + 96*A*b*c^8*d^4*f^4 +
32*A*b*c^10*d^2*f^4 + 256*B*b*c^3*d^9*f^4 + 384*B*b*c^5*d^7*f^4 + 256*B*b*c
^7*d^5*f^4 + 64*B*b*c^9*d^3*f^4 + 96*C*b*c^2*d^10*f^4 + 64*C*b*c^4*d^8*f^4
- 64*C*b*c^6*d^6*f^4 - 96*C*b*c^8*d^4*f^4 - 32*C*b*c^10*d^2*f^4 + 64*B*b*c*
d^11*f^4) - (c + d*tan(e + f*x))^(1/2)*(16*A^2*b^2*d^10*f^3 - 16*B^2*b^2*d^
10*f^3 + 16*C^2*b^2*d^10*f^3 + 32*A^2*b^2*c^2*d^8*f^3 - 32*A^2*b^2*c^6*d^4*
f^3 - 16*A^2*b^2*c^8*d^2*f^3 - 32*B^2*b^2*c^2*d^8*f^3 + 32*B^2*b^2*c^6*d^4*
f^3 + 16*B^2*b^2*c^8*d^2*f^3 + 32*C^2*b^2*c^2*d^8*f^3 - 32*C^2*b^2*c^6*d^4*
f^3 - 16*C^2*b^2*c^8*d^2*f^3 - 32*A*C*b^2*d^10*f^3 - 64*A*B*b^2*c*d^9*f^3 +
64*B*C*b^2*c*d^9*f^3 - 192*A*B*b^2*c^3*d^7*f^3 - 192*A*B*b^2*c^5*d^5*f^3 -
64*A*B*b^2*c^7*d^3*f^3 - 64*A*C*b^2*c^2*d^8*f^3 + 64*A*C*b^2*c^6*d^4*f^3 +
32*A*C*b^2*c^8*d^2*f^3 + 192*B*C*b^2*c^3*d^7*f^3 + 192*B*C*b^2*c^5*d^5*f^3
+ 64*B*C*b^2*c^7*d^3*f^3))*(-(((8*A^2*b^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 + 8*
C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 - 16*A*C*b^2*c^3*f^2 + 16*B*C*b^2*d^3*
f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 - 24*C^2*b^2*c*d^2*f^2 +
48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2*f^2 - 48*B*C*b^2*c^2*d*f^2)^2/4 - (
16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*b^4 + B^4*b
^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 +
2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c^3*f^2 + 4*B^2*b^2*c^3*f

^2 - 4*C^2*b^2*c^3*f^2 + 8*A*B*b^2*d^3*f^2 + 8*A*C*b^2*c^3*f^2 - 8*B*C*b^2*
d^3*f^2 + 12*A^2*b^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^
2 - 24*A*B*b^2*c^2*d*f^2 - 24*A*C*b^2*c*d^2*f^2 + 24*B*C*b^2*c^2*d*f^2)/(16
*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2) - ((-(((8*A^2*
b^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 -
16*A*C*b^2*c^3*f^2 + 16*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2
*c*d^2*f^2 - 24*C^2*b^2*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2
*f^2 - 48*B*C*b^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^
4 + 48*c^4*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^
4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) -
 4*A^2*b^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 + 8*A*B*b^2*d^3*
f^2 + 8*A*C*b^2*c^3*f^2 - 8*B*C*b^2*d^3*f^2 + 12*A^2*b^2*c*d^2*f^2 - 12*B^2
*b^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 - 24*A*B*b^2*c^2*d*f^2 - 24*A*C*b^2*c
*d^2*f^2 + 24*B*C*b^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3
*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*b^2*c^3*f^2 -
8*B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 - 16*A*C*b^2*c^3
*f^2 + 16*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 - 2
4*C^2*b^2*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2*f^2 - 48*B*C*
b^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2
f^4)(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*
b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c^3
*f^2 + 4*B^2*b^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 + 8*A*B*b^2*d^3*f^2 + 8*A*C*b^
2*c^3*f^2 - 8*B*C*b^2*d^3*f^2 + 12*A^2*b^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2
 + 12*C^2*b^2*c*d^2*f^2 - 24*A*B*b^2*c^2*d*f^2 - 24*A*C*b^2*c*d^2*f^2 + 24*
B*C*b^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))
)^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f
^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 32*A*b*d^12*f^4 + 32*C*b*d^12*f^4
 - 96*A*b*c^2*d^10*f^4 - 64*A*b*c^4*d^8*f^4 + 64*A*b*c^6*d^6*f^4 + 96*A*b*c
^8*d^4*f^4 + 32*A*b*c^10*d^2*f^4 + 256*B*b*c^3*d^9*f^4 + 384*B*b*c^5*d^7*f^
4 + 256*B*b*c^7*d^5*f^4 + 64*B*b*c^9*d^3*f^4 + 96*C*b*c^2*d^10*f^4 + 64*C*b
*c^4*d^8*f^4 - 64*C*b*c^6*d^6*f^4 - 96*C*b*c^8*d^4*f^4 - 32*C*b*c^10*d^2*f^
4 + 64*B*b*c*d^11*f^4) + (c + d*tan(e + f*x))^(1/2)*(16*A^2*b^2*d^10*f^3 -
16*B^2*b^2*d^10*f^3 + 16*C^2*b^2*d^10*f^3 + 32*A^2*b^2*c^2*d^8*f^3 - 32*A^2
*b^2*c^6*d^4*f^3 - 16*A^2*b^2*c^8*d^2*f^3 - 32*B^2*b^2*c^2*d^8*f^3 + 32*B^2
*b^2*c^6*d^4*f^3 + 16*B^2*b^2*c^8*d^2*f^3 + 32*C^2*b^2*c^2*d^8*f^3 - 32*C^2
*b^2*c^6*d^4*f^3 - 16*C^2*b^2*c^8*d^2*f^3 - 32*A*C*b^2*d^10*f^3 - 64*A*B*b^
2*c*d^9*f^3 + 64*B*C*b^2*c*d^9*f^3 - 192*A*B*b^2*c^3*d^7*f^3 - 192*A*B*b^2*
c^5*d^5*f^3 - 64*A*B*b^2*c^7*d^3*f^3 - 64*A*C*b^2*c^2*d^8*f^3 + 64*A*C*b^2*
c^6*d^4*f^3 + 32*A*C*b^2*c^8*d^2*f^3 + 192*B*C*b^2*c^3*d^7*f^3 + 192*B*C*b^
2*c^5*d^5*f^3 + 64*B*C*b^2*c^7*d^3*f^3))*(-(((8*A^2*b^2*c^3*f^2 - 8*B^2*b^2
*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 - 16*A*C*b^2*c^3*f^2 + 16
*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 - 24*C^2*b^2
*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2*f^2 - 48*B*C*b^2*c^2*d
*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^
4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A

^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c^3*f^2 + 4*
B^2*b^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 + 8*A*B*b^2*d^3*f^2 + 8*A*C*b^2*c^3*f^2
 - 8*B*C*b^2*d^3*f^2 + 12*A^2*b^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 + 12*C^2
*b^2*c*d^2*f^2 - 24*A*B*b^2*c^2*d*f^2 - 24*A*C*b^2*c*d^2*f^2 + 24*B*C*b^2*c
^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2) +
 48*A^3*b^3*c^3*d^6*f^2 + 48*A^3*b^3*c^5*d^4*f^2 + 16*A^3*b^3*c^7*d^2*f^2 +
 48*B^3*b^3*c^2*d^7*f^2 + 48*B^3*b^3*c^4*d^5*f^2 + 16*B^3*b^3*c^6*d^3*f^2 -
 48*C^3*b^3*c^3*d^6*f^2 - 48*C^3*b^3*c^5*d^4*f^2 - 16*C^3*b^3*c^7*d^2*f^2 +
 16*A^2*B*b^3*d^9*f^2 + 16*B*C^2*b^3*d^9*f^2 + 16*A^3*b^3*c*d^8*f^2 - 16*C^
3*b^3*c*d^8*f^2 + 48*A*B^2*b^3*c^3*d^6*f^2 + 48*A*B^2*b^3*c^5*d^4*f^2 + 16*
A*B^2*b^3*c^7*d^2*f^2 + 48*A^2*B*b^3*c^2*d^7*f^2 + 48*A^2*B*b^3*c^4*d^5*f^2
 + 16*A^2*B*b^3*c^6*d^3*f^2 + 144*A*C^2*b^3*c^3*d^6*f^2 + 144*A*C^2*b^3*c^5
*d^4*f^2 + 48*A*C^2*b^3*c^7*d^2*f^2 - 144*A^2*C*b^3*c^3*d^6*f^2 - 144*A^2*C
*b^3*c^5*d^4*f^2 - 48*A^2*C*b^3*c^7*d^2*f^2 + 48*B*C^2*b^3*c^2*d^7*f^2 + 48
*B*C^2*b^3*c^4*d^5*f^2 + 16*B*C^2*b^3*c^6*d^3*f^2 - 48*B^2*C*b^3*c^3*d^6*f^
2 - 48*B^2*C*b^3*c^5*d^4*f^2 - 16*B^2*C*b^3*c^7*d^2*f^2 - 32*A*B*C*b^3*d^9*
f^2 + 16*A*B^2*b^3*c*d^8*f^2 + 48*A*C^2*b^3*c*d^8*f^2 - 48*A^2*C*b^3*c*d^8*
f^2 - 16*B^2*C*b^3*c*d^8*f^2 - 96*A*B*C*b^3*c^2*d^7*f^2 - 96*A*B*C*b^3*c^4*
d^5*f^2 - 32*A*B*C*b^3*c^6*d^3*f^2))*(-(((8*A^2*b^2*c^3*f^2 - 8*B^2*b^2*c^3
*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 - 16*A*C*b^2*c^3*f^2 + 16*B*C
*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 - 24*C^2*b^2*c*d
^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2*f^2 - 48*B*C*b^2*c^2*d*f^2
)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*b^
4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C
^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c^3*f^2 + 4*B^2*
b^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 + 8*A*B*b^2*d^3*f^2 + 8*A*C*b^2*c^3*f^2 - 8
*B*C*b^2*d^3*f^2 + 12*A^2*b^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 + 12*C^2*b^2
*c*d^2*f^2 - 24*A*B*b^2*c^2*d*f^2 - 24*A*C*b^2*c*d^2*f^2 + 24*B*C*b^2*c^2*d
f^2)/(16(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*2i +
atan((((c + d*tan(e + f*x))^(1/2)*(16*A^2*a^2*d^10*f^3 - 16*B^2*a^2*d^10*f^
3 + 16*C^2*a^2*d^10*f^3 + 32*A^2*a^2*c^2*d^8*f^3 - 32*A^2*a^2*c^6*d^4*f^3 -
 16*A^2*a^2*c^8*d^2*f^3 - 32*B^2*a^2*c^2*d^8*f^3 + 32*B^2*a^2*c^6*d^4*f^3 +
 16*B^2*a^2*c^8*d^2*f^3 + 32*C^2*a^2*c^2*d^8*f^3 - 32*C^2*a^2*c^6*d^4*f^3 -
 16*C^2*a^2*c^8*d^2*f^3 - 32*A*C*a^2*d^10*f^3 - 64*A*B*a^2*c*d^9*f^3 + 64*B
*C*a^2*c*d^9*f^3 - 192*A*B*a^2*c^3*d^7*f^3 - 192*A*B*a^2*c^5*d^5*f^3 - 64*A
*B*a^2*c^7*d^3*f^3 - 64*A*C*a^2*c^2*d^8*f^3 + 64*A*C*a^2*c^6*d^4*f^3 + 32*A
*C*a^2*c^8*d^2*f^3 + 192*B*C*a^2*c^3*d^7*f^3 + 192*B*C*a^2*c^5*d^5*f^3 + 64
*B*C*a^2*c^7*d^3*f^3) - (-(((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*
a^2*c^3*f^2 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2
- 24*A^2*a^2*c*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A
*B*a^2*c^2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c
^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 +
 C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^
2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*B^2*a^2*c^3*f^2 +
 4*C^2*a^2*c^3*f^2 - 8*A*B*a^2*d^3*f^2 - 8*A*C*a^2*c^3*f^2 + 8*B*C*a^2*d^3*

f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*C^2*a^2*c*d^2*f^2 + 24*A*B*a^2*c^2*d*f^2 + 24*A*C*a^2*c*d^2*f^2 - 24*B*C*a^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*a^2*c^3*f^2 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A*B*a^2*c^2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*B^2*a^2*c^3*f^2 + 4*C^2*a^2*c^3*f^2 - 8*A*B*a^2*d^3*f^2 - 8*A*C*a^2*c^3*f^2 + 8*B*C*a^2*d^3*f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*C^2*a^2*c*d^2*f^2 + 24*A*B*a^2*c^2*d*f^2 + 24*A*C*a^2*c*d^2*f^2 - 24*B*C*a^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 32*B*a*d^12*f^4 - 256*A*a*c^3*d^9*f^4 - 384*A*a*c^5*d^7*f^4 - 256*A*a*c^7*d^5*f^4 - 64*A*a*c^9*d^3*f^4 - 96*B*a*c^2*d^10*f^4 - 64*B*a*c^4*d^8*f^4 + 64*B*a*c^6*d^6*f^4 + 96*B*a*c^8*d^4*f^4 + 32*B*a*c^10*d^2*f^4 + 256*C*a*c^3*d^9*f^4 + 384*C*a*c^5*d^7*f^4 + 256*C*a*c^7*d^5*f^4 + 64*C*a*c^9*d^3*f^4 - 64*A*a*c*d^11*f^4 + 64*C*a*c*d^11*f^4))*(-(((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*a^2*c^3*f^2 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A*B*a^2*c^2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*B^2*a^2*c^3*f^2 + 4*C^2*a^2*c^3*f^2 - 8*A*B*a^2*d^3*f^2 - 8*A*C*a^2*c^3*f^2 + 8*B*C*a^2*d^3*f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*C^2*a^2*c*d^2*f^2 + 24*A*B*a^2*c^2*d*f^2 + 24*A*C*a^2*c*d^2*f^2 - 24*B*C*a^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*1i + ((c + d*tan(e + f*x))^(1/2)*(16*A^2*a^2*d^10*f^3 - 16*B^2*a^2*d^10*f^3 + 16*C^2*a^2*d^10*f^3 + 32*A^2*a^2*c^2*d^8*f^3 - 32*A^2*a^2*c^6*d^4*f^3 - 16*A^2*a^2*c^8*d^2*f^3 - 32*B^2*a^2*c^2*d^8*f^3 + 32*B^2*a^2*c^6*d^4*f^3 + 16*B^2*a^2*c^8*d^2*f^3 + 32*C^2*a^2*c^2*d^8*f^3 - 32*C^2*a^2*c^6*d^4*f^3 - 16*C^2*a^2*c^8*d^2*f^3 - 32*A*C*a^2*d^10*f^3 - 64*A*B*a^2*c*d^9*f^3 + 64*B*C*a^2*c*d^9*f^3 - 192*A*B*a^2*c^3*d^7*f^3 - 192*A*B*a^2*c^5*d^5*f^3 - 64*A*B*a^2*c^7*d^3*f^3 - 64*A*C*a^2*c^2*d^8*f^3 + 64*A*C*a^2*c^6*d^4*f^3 + 32*A*C*a^2*c^8*d^2*f^3 + 192*B*C*a^2*c^3*d^7*f^3 + 192*B*C*a^2*c^5*d^5*f^3 + 64*B*C*a^2*c^7*d^3*f^3) - (-(((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*a^2*c^3*f^2 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A*B*a^2*c^2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*B^2*a^2*c^3*f^2 + 4

```
*C^2*a^2*c^3*f^2 - 8*A*B*a^2*d^3*f^2 - 8*A*C*a^2*c^3*f^2 + 8*B*C*a^2*d^3*f^
2 - 12*A^2*a^2*c*d^2*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*C^2*a^2*c*d^2*f^2 + 24
*A*B*a^2*c^2*d*f^2 + 24*A*C*a^2*c*d^2*f^2 - 24*B*C*a^2*c^2*d*f^2)/(16*(c^6*
f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x)
)^(1/2)*(-(((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*a^2*c^3*f^2 - 16
*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2 - 24*A^2*a^2*c*d
^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A*B*a^2*c^2*d*f^2
 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*
f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C
^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*
B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*B^2*a^2*c^3*f^2 + 4*C^2*a^2*c^3*f
^2 - 8*A*B*a^2*d^3*f^2 - 8*A*C*a^2*c^3*f^2 + 8*B*C*a^2*d^3*f^2 - 12*A^2*a^2
*c*d^2*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*C^2*a^2*c*d^2*f^2 + 24*A*B*a^2*c^2*d
*f^2 + 24*A*C*a^2*c*d^2*f^2 - 24*B*C*a^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4
+ 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5
+ 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) +
32*B*a*d^12*f^4 + 256*A*a*c^3*d^9*f^4 + 384*A*a*c^5*d^7*f^4 + 256*A*a*c^7*d
^5*f^4 + 64*A*a*c^9*d^3*f^4 + 96*B*a*c^2*d^10*f^4 + 64*B*a*c^4*d^8*f^4 - 64
*B*a*c^6*d^6*f^4 - 96*B*a*c^8*d^4*f^4 - 32*B*a*c^10*d^2*f^4 - 256*C*a*c^3*d
^9*f^4 - 384*C*a*c^5*d^7*f^4 - 256*C*a*c^7*d^5*f^4 - 64*C*a*c^9*d^3*f^4 + 6
4*A*a*c*d^11*f^4 - 64*C*a*c*d^11*f^4))*(-(((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c
^3*f^2 + 8*C^2*a^2*c^3*f^2 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B
*C*a^2*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c
*d^2*f^2 + 48*A*B*a^2*c^2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f
^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*
a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2
*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*B^
2*a^2*c^3*f^2 + 4*C^2*a^2*c^3*f^2 - 8*A*B*a^2*d^3*f^2 - 8*A*C*a^2*c^3*f^2 +
 8*B*C*a^2*d^3*f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*C^2*a
^2*c*d^2*f^2 + 24*A*B*a^2*c^2*d*f^2 + 24*A*C*a^2*c*d^2*f^2 - 24*B*C*a^2*c^2
*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*1i)
/(((c + d*tan(e + f*x))^(1/2)*(16*A^2*a^2*d^10*f^3 - 16*B^2*a^2*d^10*f^3 +
16*C^2*a^2*d^10*f^3 + 32*A^2*a^2*c^2*d^8*f^3 - 32*A^2*a^2*c^6*d^4*f^3 - 16*
A^2*a^2*c^8*d^2*f^3 - 32*B^2*a^2*c^2*d^8*f^3 + 32*B^2*a^2*c^6*d^4*f^3 + 16*
B^2*a^2*c^8*d^2*f^3 + 32*C^2*a^2*c^2*d^8*f^3 - 32*C^2*a^2*c^6*d^4*f^3 - 16*
C^2*a^2*c^8*d^2*f^3 - 32*A*C*a^2*d^10*f^3 - 64*A*B*a^2*c*d^9*f^3 + 64*B*C*a
^2*c*d^9*f^3 - 192*A*B*a^2*c^3*d^7*f^3 - 192*A*B*a^2*c^5*d^5*f^3 - 64*A*B*a
^2*c^7*d^3*f^3 - 64*A*C*a^2*c^2*d^8*f^3 + 64*A*C*a^2*c^6*d^4*f^3 + 32*A*C*a
^2*c^8*d^2*f^3 + 192*B*C*a^2*c^3*d^7*f^3 + 192*B*C*a^2*c^5*d^5*f^3 + 64*B*C
*a^2*c^7*d^3*f^3) - (-(((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*a^2*
c^3*f^2 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2 - 24
*A^2*a^2*c*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A*B*a
^2*c^2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c^6*f
^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4
*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^
```

2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*B^2*a^2*c^3*f^2 + 4*C
^2*a^2*c^3*f^2 - 8*A*B*a^2*d^3*f^2 - 8*A*C*a^2*c^3*f^2 + 8*B*C*a^2*d^3*f^2
- 12*A^2*a^2*c*d^2*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*C^2*a^2*c*d^2*f^2 + 24*A
*B*a^2*c^2*d*f^2 + 24*A*C*a^2*c*d^2*f^2 - 24*B*C*a^2*c^2*d*f^2)/(16*(c^6*f^
4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^
(1/2)*(-(((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*a^2*c^3*f^2 - 16*A
*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2 - 24*A^2*a^2*c*d^2
*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A*B*a^2*c^2*d*f^2 +
48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^
4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3
*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^
2*C*a^4))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*B^2*a^2*c^3*f^2 + 4*C^2*a^2*c^3*f^2
- 8*A*B*a^2*d^3*f^2 - 8*A*C*a^2*c^3*f^2 + 8*B*C*a^2*d^3*f^2 - 12*A^2*a^2*c
*d^2*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*C^2*a^2*c*d^2*f^2 + 24*A*B*a^2*c^2*d*f
^2 + 24*A*C*a^2*c*d^2*f^2 - 24*B*C*a^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 +
3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 +
640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 32
*B*a*d^12*f^4 - 256*A*a*c^3*d^9*f^4 - 384*A*a*c^5*d^7*f^4 - 256*A*a*c^7*d^5
*f^4 - 64*A*a*c^9*d^3*f^4 - 96*B*a*c^2*d^10*f^4 - 64*B*a*c^4*d^8*f^4 + 64*B
*a*c^6*d^6*f^4 + 96*B*a*c^8*d^4*f^4 + 32*B*a*c^10*d^2*f^4 + 256*C*a*c^3*d^9
*f^4 + 384*C*a*c^5*d^7*f^4 + 256*C*a*c^7*d^5*f^4 + 64*C*a*c^9*d^3*f^4 - 64*
A*a*c*d^11*f^4 + 64*C*a*c*d^11*f^4))*(-(((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3
*f^2 + 8*C^2*a^2*c^3*f^2 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C
*a^2*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d
^2*f^2 + 48*A*B*a^2*c^2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2
)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^
4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C
^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*B^2*
a^2*c^3*f^2 + 4*C^2*a^2*c^3*f^2 - 8*A*B*a^2*d^3*f^2 - 8*A*C*a^2*c^3*f^2 + 8
*B*C*a^2*d^3*f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*C^2*a^2
*c*d^2*f^2 + 24*A*B*a^2*c^2*d*f^2 + 24*A*C*a^2*c*d^2*f^2 - 24*B*C*a^2*c^2*d
f^2)/(16(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2) - ((c
+ d*tan(e + f*x))^(1/2)*(16*A^2*a^2*d^10*f^3 - 16*B^2*a^2*d^10*f^3 + 16*C^
2*a^2*d^10*f^3 + 32*A^2*a^2*c^2*d^8*f^3 - 32*A^2*a^2*c^6*d^4*f^3 - 16*A^2*a
^2*c^8*d^2*f^3 - 32*B^2*a^2*c^2*d^8*f^3 + 32*B^2*a^2*c^6*d^4*f^3 + 16*B^2*a
^2*c^8*d^2*f^3 + 32*C^2*a^2*c^2*d^8*f^3 - 32*C^2*a^2*c^6*d^4*f^3 - 16*C^2*a
^2*c^8*d^2*f^3 - 32*A*C*a^2*d^10*f^3 - 64*A*B*a^2*c*d^9*f^3 + 64*B*C*a^2*c*
d^9*f^3 - 192*A*B*a^2*c^3*d^7*f^3 - 192*A*B*a^2*c^5*d^5*f^3 - 64*A*B*a^2*c^
7*d^3*f^3 - 64*A*C*a^2*c^2*d^8*f^3 + 64*A*C*a^2*c^6*d^4*f^3 + 32*A*C*a^2*c^
8*d^2*f^3 + 192*B*C*a^2*c^3*d^7*f^3 + 192*B*C*a^2*c^5*d^5*f^3 + 64*B*C*a^2*
c^7*d^3*f^3) - (-(((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*a^2*c^3*f
^2 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2 - 24*A^2*
a^2*c*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A*B*a^2*c^
2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 +
16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4

- 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4
 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*B^2*a^2*c^3*f^2 + 4*C^2*a^
2*c^3*f^2 - 8*A*B*a^2*d^3*f^2 - 8*A*C*a^2*c^3*f^2 + 8*B*C*a^2*d^3*f^2 - 12*
A^2*a^2*c*d^2*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*C^2*a^2*c*d^2*f^2 + 24*A*B*a^
2*c^2*d*f^2 + 24*A*C*a^2*c*d^2*f^2 - 24*B*C*a^2*c^2*d*f^2)/(16*(c^6*f^4 + d
^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)
*(-(((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*a^2*c^3*f^2 - 16*A*B*a^
2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2
+ 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A*B*a^2*c^2*d*f^2 + 48*A
*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 4
8*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4
- 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a
^4))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*B^2*a^2*c^3*f^2 + 4*C^2*a^2*c^3*f^2 - 8*
A*B*a^2*d^3*f^2 - 8*A*C*a^2*c^3*f^2 + 8*B*C*a^2*d^3*f^2 - 12*A^2*a^2*c*d^2*
f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*C^2*a^2*c*d^2*f^2 + 24*A*B*a^2*c^2*d*f^2 +
24*A*C*a^2*c*d^2*f^2 - 24*B*C*a^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2
*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c
^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) + 32*B*a*
d^12*f^4 + 256*A*a*c^3*d^9*f^4 + 384*A*a*c^5*d^7*f^4 + 256*A*a*c^7*d^5*f^4
+ 64*A*a*c^9*d^3*f^4 + 96*B*a*c^2*d^10*f^4 + 64*B*a*c^4*d^8*f^4 - 64*B*a*c^
6*d^6*f^4 - 96*B*a*c^8*d^4*f^4 - 32*B*a*c^10*d^2*f^4 - 256*C*a*c^3*d^9*f^4
- 384*C*a*c^5*d^7*f^4 - 256*C*a*c^7*d^5*f^4 - 64*C*a*c^9*d^3*f^4 + 64*A*a*c
*d^11*f^4 - 64*C*a*c*d^11*f^4))*(-(((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2
+ 8*C^2*a^2*c^3*f^2 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*
d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^
2 + 48*A*B*a^2*c^2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4
 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B
^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^
4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*B^2*a^2*c
^3*f^2 + 4*C^2*a^2*c^3*f^2 - 8*A*B*a^2*d^3*f^2 - 8*A*C*a^2*c^3*f^2 + 8*B*C*
a^2*d^3*f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*C^2*a^2*c*d^
2*f^2 + 24*A*B*a^2*c^2*d*f^2 + 24*A*C*a^2*c*d^2*f^2 - 24*B*C*a^2*c^2*d*f^2)
/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2) - 16*A^3*a
^3*d^9*f^2 + 16*C^3*a^3*d^9*f^2 - 48*A^3*a^3*c^2*d^7*f^2 - 48*A^3*a^3*c^4*d
^5*f^2 - 16*A^3*a^3*c^6*d^3*f^2 + 48*B^3*a^3*c^3*d^6*f^2 + 48*B^3*a^3*c^5*d
^4*f^2 + 16*B^3*a^3*c^7*d^2*f^2 + 48*C^3*a^3*c^2*d^7*f^2 + 48*C^3*a^3*c^4*d
^5*f^2 + 16*C^3*a^3*c^6*d^3*f^2 - 16*A*B^2*a^3*d^9*f^2 - 48*A*C^2*a^3*d^9*f
^2 + 48*A^2*C*a^3*d^9*f^2 + 16*B^2*C*a^3*d^9*f^2 + 16*B^3*a^3*c*d^8*f^2 - 4
8*A*B^2*a^3*c^2*d^7*f^2 - 48*A*B^2*a^3*c^4*d^5*f^2 - 16*A*B^2*a^3*c^6*d^3*f
^2 + 48*A^2*B*a^3*c^3*d^6*f^2 + 48*A^2*B*a^3*c^5*d^4*f^2 + 16*A^2*B*a^3*c^7
*d^2*f^2 - 144*A*C^2*a^3*c^2*d^7*f^2 - 144*A*C^2*a^3*c^4*d^5*f^2 - 48*A*C^2
*a^3*c^6*d^3*f^2 + 144*A^2*C*a^3*c^2*d^7*f^2 + 144*A^2*C*a^3*c^4*d^5*f^2 +
48*A^2*C*a^3*c^6*d^3*f^2 + 48*B*C^2*a^3*c^3*d^6*f^2 + 48*B*C^2*a^3*c^5*d^4*
f^2 + 16*B*C^2*a^3*c^7*d^2*f^2 + 48*B^2*C*a^3*c^2*d^7*f^2 + 48*B^2*C*a^3*c^
4*d^5*f^2 + 16*B^2*C*a^3*c^6*d^3*f^2 + 16*A^2*B*a^3*c*d^8*f^2 + 16*B*C^2*a^

3*c*d^8*f^2 - 96*A*B*C*a^3*c^3*d^6*f^2 - 96*A*B*C*a^3*c^5*d^4*f^2 - 32*A*B*C*a^3*c^7*d^2*f^2 - 32*A*B*C*a^3*c*d^8*f^2))*(-(((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*a^2*c^3*f^2 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A*B*a^2*c^2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*B^2*a^2*c^3*f^2 + 4*C^2*a^2*c^3*f^2 - 8*A*B*a^2*d^3*f^2 - 8*A*C*a^2*c^3*f^2 + 8*B*C*a^2*d^3*f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*C^2*a^2*c*d^2*f^2 + 24*A*B*a^2*c^2*d*f^2 + 24*A*C*a^2*c*d^2*f^2 - 24*B*C*a^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*2i + (2*(C*b*c^3 + A*b*c*d^2 - B*b*c^2*d))/(d^2*f*(c^2 + d^2)*(c + d*tan(e + f*x))^(1/2)) - (2*(A*a*d^2 + C*a*c^2 - B*a*c*d))/(d*f*(c^2 + d^2)*(c + d*tan(e + f*x))^(1/2)) + (2*C*b*(c + d*tan(e + f*x))^(1/2))/(d^2*f)

3.119 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^{3/2}}\,dx$

Optimal result 1705
Rubi [A] (verified) 1705
Mathematica [C] (verified) 1707
Maple [B] (verified) 1708
Fricas [B] (verification not implemented) 1708
Sympy [F] 1708
Maxima [F(-1)] 1709
Giac [F(-1)] 1709
Mupad [B] (verification not implemented) 1709

Optimal result

Integrand size = 35, antiderivative size = 157

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^{3/2}}\,dx = -\frac{(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(c-id)^{3/2}f} - \frac{(B-i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(c+id)^{3/2}f} - \frac{2(c^2C-Bcd+Ad^2)}{d\,(c^2+d^2)\,f\sqrt{c+d\tan(e+fx)}}$$

```
[Out] -(I*A+B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(c-I*d)^(3/2)/f-
(B-I*(A-C))*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))/(c+I*d)^(3/2)/f-2
*(A*d^2-B*c*d+C*c^2)/d/(c^2+d^2)/f/(c+d*tan(f*x+e))^(1/2)
```

Rubi [A] (verified)

Time = 0.33 (sec) , antiderivative size = 157, normalized size of antiderivative = 1.00, number of steps used = 8, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.143, Rules used = {3709, 3620, 3618, 65, 214}

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^{3/2}}\,dx = -\frac{(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f(c-id)^{3/2}} - \frac{(B-i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f(c+id)^{3/2}} - \frac{2(Ad^2-Bcd+c^2C)}{df\,(c^2+d^2)\,\sqrt{c+d\tan(e+fx)}}$$

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/(c + d*Tan[e + f*x])^(3/2),x]
```

```
[Out] -(((I*A + B - I*C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]])/((c - I
*d)^(3/2)*f)) - ((B - I*(A - C))*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c +
```

```
I*d]])/((c + I*d)^(3/2)*f) - (2*(c^2*C - B*c*d + A*d^2))/(d*(c^2 + d^2)*f*S
qrt[c + d*Tan[e + f*x]])
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3709

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 -
 a*b*B + a^2*C)*((a + b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1)*(a^2 + b^2))), x
] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[b*B + a*(A -
C) - (A*b - a*B - b*C)*Tan[e + f*x], x], x], x] /; FreeQ[{a, b, e, f, A, B,
 C}, x] && NeQ[A*b^2 - a*b*B + a^2*C, 0] && LtQ[m, -1] && NeQ[a^2 + b^2, 0]
```

Rubi steps

$$\text{integral} = -\frac{2(c^2C - Bcd + Ad^2)}{d\,(c^2+d^2)\,f\sqrt{c+d\tan(e+fx)}} + \frac{\int \frac{Ac-cC+Bd+(Bc-(A-C)d)\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{c^2+d^2}$$

$$
\begin{aligned}
&= -\frac{2(c^2C-Bcd+Ad^2)}{d\,(c^2+d^2)\,f\sqrt{c+d\tan(e+fx)}}+\frac{(A-iB-C)\int\frac{1+i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(c-id)} \\
&\quad+\frac{(A+iB-C)\int\frac{1-i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(c+id)} \\
&= -\frac{2(c^2C-Bcd+Ad^2)}{d\,(c^2+d^2)\,f\sqrt{c+d\tan(e+fx)}} \\
&\quad+\frac{(iA+B-iC)\text{Subst}\left(\int\frac{1}{(-1+x)\sqrt{c-idx}}\,dx,x,i\tan(e+fx)\right)}{2(c-id)f} \\
&\quad-\frac{(i(A+iB-C))\text{Subst}\left(\int\frac{1}{(-1+x)\sqrt{c+idx}}\,dx,x,-i\tan(e+fx)\right)}{2(c+id)f} \\
&= -\frac{2(c^2C-Bcd+Ad^2)}{d\,(c^2+d^2)\,f\sqrt{c+d\tan(e+fx)}} \\
&\quad-\frac{(A-iB-C)\text{Subst}\left(\int\frac{1}{-1-\frac{ic}{d}+\frac{ix^2}{d}}\,dx,x,\sqrt{c+d\tan(e+fx)}\right)}{(c-id)df} \\
&\quad-\frac{(A+iB-C)\text{Subst}\left(\int\frac{1}{-1+\frac{ic}{d}-\frac{ix^2}{d}}\,dx,x,\sqrt{c+d\tan(e+fx)}\right)}{(c+id)df} \\
&= -\frac{(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(c-id)^{3/2}f} \\
&\quad-\frac{(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(c+id)^{3/2}f}-\frac{2(c^2C-Bcd+Ad^2)}{d\,(c^2+d^2)\,f\sqrt{c+d\tan(e+fx)}}
\end{aligned}
$$

Mathematica [C] (verified)

Result contains higher order function than in optimal. Order 5 vs. order 3 in optimal.

Time = 1.11 (sec) , antiderivative size = 218, normalized size of antiderivative = 1.39

$$
\int\frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^{3/2}}\,dx=\frac{-iB\left(\frac{\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{\sqrt{c-id}}-\frac{\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{\sqrt{c+id}}\right)-\frac{}{\sqrt{}}
$$

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/(c + d*Tan[e + f*x])^(3/2
),x]
```

```
[Out] ((-I)*B*(ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]]/Sqrt[c - I*d] - Ar
cTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]]/Sqrt[c + I*d]) - (2*C)/Sqrt[c
```

```
 + d*Tan[e + f*x]] + ((B*c + (-A + C)*d)*(((-I)*c + d)*Hypergeometric2F1[-1
/2, 1, 1/2, (c + d*Tan[e + f*x])/(c - I*d)] + (I*c + d)*Hypergeometric2F1[-
1/2, 1, 1/2, (c + d*Tan[e + f*x])/(c + I*d)]))/((c^2 + d^2)*Sqrt[c + d*Tan[
e + f*x]]))/(d*f)
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 5612 vs. $2(136) = 272$.

Time = 0.13 (sec) , antiderivative size = 5613, normalized size of antiderivative = 35.75

method	result	size
parts	Expression too large to display	5613
derivativedivides	Expression too large to display	11427
default	Expression too large to display	11427

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(3/2),x,method=_RETURN
VERBOSE)
```

```
[Out] result too large to display
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 7982 vs. $2(129) = 258$.

Time = 1.83 (sec) , antiderivative size = 7982, normalized size of antiderivative = 50.84

$$\int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{(c + d\tan(e + fx))^{3/2}}\,dx = \text{Too large to display}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(3/2),x, algorit
hm="fricas")
```

```
[Out] Too large to include
```

Sympy [F]

$$\int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{(c + d\tan(e + fx))^{3/2}}\,dx = \int \frac{A + B\tan\left(e + fx\right) + C\tan^2\left(e + fx\right)}{\left(c + d\tan\left(e + fx\right)\right)^{\frac{3}{2}}}\,dx$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x+e))**(3/2),x)
```

```
[Out] Integral((A + B*tan(e + f*x) + C*tan(e + f*x)**2)/(c + d*tan(e + f*x))**(3/
2), x)
```

Maxima [F(-1)]

Timed out.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(3/2),x, algorit
hm="maxima")
[Out] Timed out
```

Giac [F(-1)]

Timed out.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(3/2),x, algorit
hm="giac")
[Out] Timed out
```

Mupad [B] (verification not implemented)

Time = 18.19 (sec) , antiderivative size = 8588, normalized size of antiderivative = 54.70

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^{3/2}}\,dx = \text{Too large to display}$$

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/(c + d*tan(e + f*x))^(3/2),x)
[Out] (log((((((96*C^4*c^2*d^4*f^4 - 16*C^4*d^6*f^4 - 144*C^4*c^4*d^2*f^4)^(1/2)
 - 4*C^2*c^3*f^2 + 12*C^2*c*d^2*f^2)/(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3
*c^4*d^2*f^4))^(1/2)*(64*C*c*d^11*f^4 - ((c + d*tan(e + f*x))^(1/2)*(((96*C
^4*c^2*d^4*f^4 - 16*C^4*d^6*f^4 - 144*C^4*c^4*d^2*f^4)^(1/2) - 4*C^2*c^3*f^
2 + 12*C^2*c*d^2*f^2)/(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))^
(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5
 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5))/4 + 256*C*c^3*d^9*f^4 + 384*C*c^5*d^
7*f^4 + 256*C*c^7*d^5*f^4 + 64*C*c^9*d^3*f^4))/4 + (c + d*tan(e + f*x))^(1/
2)*(16*C^2*d^10*f^3 + 32*C^2*c^2*d^8*f^3 - 32*C^2*c^6*d^4*f^3 - 16*C^2*c^8*
d^2*f^3))*(((96*C^4*c^2*d^4*f^4 - 16*C^4*d^6*f^4 - 144*C^4*c^4*d^2*f^4)^(1/
2) - 4*C^2*c^3*f^2 + 12*C^2*c*d^2*f^2)/(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 +
 3*c^4*d^2*f^4))^(1/2))/4 - 8*C^3*d^9*f^2 - 24*C^3*c^2*d^7*f^2 - 24*C^3*c^4
*d^5*f^2 - 8*C^3*c^6*d^3*f^2)*(((96*C^4*c^2*d^4*f^4 - 16*C^4*d^6*f^4 - 144*
C^4*c^4*d^2*f^4)^(1/2) - 4*C^2*c^3*f^2 + 12*C^2*c*d^2*f^2)/(c^6*f^4 + d^6*f
```

```
^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))^(1/2))/4 + (log(((((-((96*C^4*c^2*d^4*
f^4 - 16*C^4*d^6*f^4 - 144*C^4*c^4*d^2*f^4)^(1/2) + 4*C^2*c^3*f^2 - 12*C^2*
c*d^2*f^2)/(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))^(1/2)*(64*C
*c*d^11*f^4 - ((c + d*tan(e + f*x))^(1/2)*(-((96*C^4*c^2*d^4*f^4 - 16*C^4*d
^6*f^4 - 144*C^4*c^4*d^2*f^4)^(1/2) + 4*C^2*c^3*f^2 - 12*C^2*c*d^2*f^2)/(c^
6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))^(1/2)*(64*c*d^12*f^5 + 32
0*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c
^11*d^2*f^5))/4 + 256*C*c^3*d^9*f^4 + 384*C*c^5*d^7*f^4 + 256*C*c^7*d^5*f^4
 + 64*C*c^9*d^3*f^4))/4 + (c + d*tan(e + f*x))^(1/2)*(16*C^2*d^10*f^3 + 32*
C^2*c^2*d^8*f^3 - 32*C^2*c^6*d^4*f^3 - 16*C^2*c^8*d^2*f^3))*(-((96*C^4*c^2*
d^4*f^4 - 16*C^4*d^6*f^4 - 144*C^4*c^4*d^2*f^4)^(1/2) + 4*C^2*c^3*f^2 - 12*
C^2*c*d^2*f^2)/(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))^(1/2))/
4 - 8*C^3*d^9*f^2 - 24*C^3*c^2*d^7*f^2 - 24*C^3*c^4*d^5*f^2 - 8*C^3*c^6*d^3
*f^2)*(-((96*C^4*c^2*d^4*f^4 - 16*C^4*d^6*f^4 - 144*C^4*c^4*d^2*f^4)^(1/2)
+ 4*C^2*c^3*f^2 - 12*C^2*c*d^2*f^2)/(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*
c^4*d^2*f^4))^(1/2))/4 - log(((((96*C^4*c^2*d^4*f^4 - 16*C^4*d^6*f^4 - 144*
C^4*c^4*d^2*f^4)^(1/2) - 4*C^2*c^3*f^2 + 12*C^2*c*d^2*f^2)/(16*c^6*f^4 + 16
*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4))^(1/2)*((c + d*tan(e + f*x))^(1
/2)*(((96*C^4*c^2*d^4*f^4 - 16*C^4*d^6*f^4 - 144*C^4*c^4*d^2*f^4)^(1/2) - 4
*C^2*c^3*f^2 + 12*C^2*c*d^2*f^2)/(16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4
+ 48*c^4*d^2*f^4))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^
5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) + 64*C*c*d^11*f^4
+ 256*C*c^3*d^9*f^4 + 384*C*c^5*d^7*f^4 + 256*C*c^7*d^5*f^4 + 64*C*c^9*d^3*
f^4) - (c + d*tan(e + f*x))^(1/2)*(16*C^2*d^10*f^3 + 32*C^2*c^2*d^8*f^3 - 3
2*C^2*c^6*d^4*f^3 - 16*C^2*c^8*d^2*f^3))*(((96*C^4*c^2*d^4*f^4 - 16*C^4*d^6
*f^4 - 144*C^4*c^4*d^2*f^4)^(1/2) - 4*C^2*c^3*f^2 + 12*C^2*c*d^2*f^2)/(16*c
^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4))^(1/2) - 8*C^3*d^9*f
^2 - 24*C^3*c^2*d^7*f^2 - 24*C^3*c^4*d^5*f^2 - 8*C^3*c^6*d^3*f^2)*(((96*C^4
*c^2*d^4*f^4 - 16*C^4*d^6*f^4 - 144*C^4*c^4*d^2*f^4)^(1/2) - 4*C^2*c^3*f^2
+ 12*C^2*c*d^2*f^2)/(16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*
f^4))^(1/2) - log(((-((96*C^4*c^2*d^4*f^4 - 16*C^4*d^6*f^4 - 144*C^4*c^4*d^
2*f^4)^(1/2) + 4*C^2*c^3*f^2 - 12*C^2*c*d^2*f^2)/(16*c^6*f^4 + 16*d^6*f^4 +
 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(-((96
*C^4*c^2*d^4*f^4 - 16*C^4*d^6*f^4 - 144*C^4*c^4*d^2*f^4)^(1/2) + 4*C^2*c^3*
f^2 - 12*C^2*c*d^2*f^2)/(16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*
d^2*f^4))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c
^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) + 64*C*c*d^11*f^4 + 256*C*c
^3*d^9*f^4 + 384*C*c^5*d^7*f^4 + 256*C*c^7*d^5*f^4 + 64*C*c^9*d^3*f^4) - (c
 + d*tan(e + f*x))^(1/2)*(16*C^2*d^10*f^3 + 32*C^2*c^2*d^8*f^3 - 32*C^2*c^6
*d^4*f^3 - 16*C^2*c^8*d^2*f^3))*(-((96*C^4*c^2*d^4*f^4 - 16*C^4*d^6*f^4 - 1
44*C^4*c^4*d^2*f^4)^(1/2) + 4*C^2*c^3*f^2 - 12*C^2*c*d^2*f^2)/(16*c^6*f^4 +
 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4))^(1/2) - 8*C^3*d^9*f^2 - 24*
C^3*c^2*d^7*f^2 - 24*C^3*c^4*d^5*f^2 - 8*C^3*c^6*d^3*f^2)*(-((96*C^4*c^2*d^
4*f^4 - 16*C^4*d^6*f^4 - 144*C^4*c^4*d^2*f^4)^(1/2) + 4*C^2*c^3*f^2 - 12*C^
2*c*d^2*f^2)/(16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4))^(
```

1/2) + (log(8*A^3*d^9*f^2 - ((((((96*A^4*c^2*d^4*f^4 - 16*A^4*d^6*f^4 - 144
*A^4*c^4*d^2*f^4)^(1/2) - 4*A^2*c^3*f^2 + 12*A^2*c*d^2*f^2)/(c^6*f^4 + d^6*
f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))^(1/2)*(((((96*A^4*c^2*d^4*f^4 - 16*A^
4*d^6*f^4 - 144*A^4*c^4*d^2*f^4)^(1/2) - 4*A^2*c^3*f^2 + 12*A^2*c*d^2*f^2)/
(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))^(1/2)*(c + d*tan(e + f
x))^(1/2)(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^
6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5))/4 + 64*A*c*d^11*f^4 + 256*A*c^3
*d^9*f^4 + 384*A*c^5*d^7*f^4 + 256*A*c^7*d^5*f^4 + 64*A*c^9*d^3*f^4))/4 - (
c + d*tan(e + f*x))^(1/2)*(16*A^2*d^10*f^3 + 32*A^2*c^2*d^8*f^3 - 32*A^2*c^
6*d^4*f^3 - 16*A^2*c^8*d^2*f^3))*(((96*A^4*c^2*d^4*f^4 - 16*A^4*d^6*f^4 - 1
44*A^4*c^4*d^2*f^4)^(1/2) - 4*A^2*c^3*f^2 + 12*A^2*c*d^2*f^2)/(c^6*f^4 + d^
6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))^(1/2))/4 + 24*A^3*c^2*d^7*f^2 + 24*
A^3*c^4*d^5*f^2 + 8*A^3*c^6*d^3*f^2)*(((96*A^4*c^2*d^4*f^4 - 16*A^4*d^6*f^4
 - 144*A^4*c^4*d^2*f^4)^(1/2) - 4*A^2*c^3*f^2 + 12*A^2*c*d^2*f^2)/(c^6*f^4
+ d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))^(1/2))/4 + (log(8*A^3*d^9*f^2 -
 ((((-((96*A^4*c^2*d^4*f^4 - 16*A^4*d^6*f^4 - 144*A^4*c^4*d^2*f^4)^(1/2) +
4*A^2*c^3*f^2 - 12*A^2*c*d^2*f^2)/(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^
4*d^2*f^4))^(1/2)*(((-((96*A^4*c^2*d^4*f^4 - 16*A^4*d^6*f^4 - 144*A^4*c^4*d
^2*f^4)^(1/2) + 4*A^2*c^3*f^2 - 12*A^2*c*d^2*f^2)/(c^6*f^4 + d^6*f^4 + 3*c^
2*d^4*f^4 + 3*c^4*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(64*c*d^12*f^5
 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 +
 64*c^11*d^2*f^5))/4 + 64*A*c*d^11*f^4 + 256*A*c^3*d^9*f^4 + 384*A*c^5*d^7*
f^4 + 256*A*c^7*d^5*f^4 + 64*A*c^9*d^3*f^4))/4 - (c + d*tan(e + f*x))^(1/2)
*(16*A^2*d^10*f^3 + 32*A^2*c^2*d^8*f^3 - 32*A^2*c^6*d^4*f^3 - 16*A^2*c^8*d^
2*f^3))*(-((96*A^4*c^2*d^4*f^4 - 16*A^4*d^6*f^4 - 144*A^4*c^4*d^2*f^4)^(1/2
) + 4*A^2*c^3*f^2 - 12*A^2*c*d^2*f^2)/(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 +
3*c^4*d^2*f^4))^(1/2))/4 + 24*A^3*c^2*d^7*f^2 + 24*A^3*c^4*d^5*f^2 + 8*A^3*
c^6*d^3*f^2)*(-((96*A^4*c^2*d^4*f^4 - 16*A^4*d^6*f^4 - 144*A^4*c^4*d^2*f^4)
^(1/2) + 4*A^2*c^3*f^2 - 12*A^2*c*d^2*f^2)/(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f
^4 + 3*c^4*d^2*f^4))^(1/2))/4 - log(8*A^3*d^9*f^2 - ((((96*A^4*c^2*d^4*f^4
 - 16*A^4*d^6*f^4 - 144*A^4*c^4*d^2*f^4)^(1/2) - 4*A^2*c^3*f^2 + 12*A^2*c*d^
2*f^2)/(16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4))^(1/2)*(
64*A*c*d^11*f^4 - (((96*A^4*c^2*d^4*f^4 - 16*A^4*d^6*f^4 - 144*A^4*c^4*d^2*
f^4)^(1/2) - 4*A^2*c^3*f^2 + 12*A^2*c*d^2*f^2)/(16*c^6*f^4 + 16*d^6*f^4 + 4
8*c^2*d^4*f^4 + 48*c^4*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(64*c*d^1
2*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*
f^5 + 64*c^11*d^2*f^5) + 256*A*c^3*d^9*f^4 + 384*A*c^5*d^7*f^4 + 256*A*c^7*
d^5*f^4 + 64*A*c^9*d^3*f^4) + (c + d*tan(e + f*x))^(1/2)*(16*A^2*d^10*f^3 +
 32*A^2*c^2*d^8*f^3 - 32*A^2*c^6*d^4*f^3 - 16*A^2*c^8*d^2*f^3))*(((96*A^4*c
^2*d^4*f^4 - 16*A^4*d^6*f^4 - 144*A^4*c^4*d^2*f^4)^(1/2) - 4*A^2*c^3*f^2 +
12*A^2*c*d^2*f^2)/(16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^
4))^(1/2) + 24*A^3*c^2*d^7*f^2 + 24*A^3*c^4*d^5*f^2 + 8*A^3*c^6*d^3*f^2)*((
(96*A^4*c^2*d^4*f^4 - 16*A^4*d^6*f^4 - 144*A^4*c^4*d^2*f^4)^(1/2) - 4*A^2*c
^3*f^2 + 12*A^2*c*d^2*f^2)/(16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c
^4*d^2*f^4))^(1/2) - log(8*A^3*d^9*f^2 - ((-((96*A^4*c^2*d^4*f^4 - 16*A^4*d

^6*f^4 - 144*A^4*c^4*d^2*f^4)^(1/2) + 4*A^2*c^3*f^2 - 12*A^2*c*d^2*f^2)/(16
*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4))^(1/2)*(64*A*c*d^1
1*f^4 - (-((96*A^4*c^2*d^4*f^4 - 16*A^4*d^6*f^4 - 144*A^4*c^4*d^2*f^4)^(1/2
) + 4*A^2*c^3*f^2 - 12*A^2*c*d^2*f^2)/(16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4
*f^4 + 48*c^4*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(64*c*d^12*f^5 + 3
20*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*
c^11*d^2*f^5) + 256*A*c^3*d^9*f^4 + 384*A*c^5*d^7*f^4 + 256*A*c^7*d^5*f^4 +
 64*A*c^9*d^3*f^4) + (c + d*tan(e + f*x))^(1/2)*(16*A^2*d^10*f^3 + 32*A^2*c
^2*d^8*f^3 - 32*A^2*c^6*d^4*f^3 - 16*A^2*c^8*d^2*f^3))*(-((96*A^4*c^2*d^4*f
^4 - 16*A^4*d^6*f^4 - 144*A^4*c^4*d^2*f^4)^(1/2) + 4*A^2*c^3*f^2 - 12*A^2*c
*d^2*f^2)/(16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4))^(1/2
) + 24*A^3*c^2*d^7*f^2 + 24*A^3*c^4*d^5*f^2 + 8*A^3*c^6*d^3*f^2)*(-((96*A^4
*c^2*d^4*f^4 - 16*A^4*d^6*f^4 - 144*A^4*c^4*d^2*f^4)^(1/2) + 4*A^2*c^3*f^2
- 12*A^2*c*d^2*f^2)/(16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*
f^4))^(1/2) + (log(- (((c + d*tan(e + f*x))^(1/2)*(16*B^2*d^10*f^3 + 32*B^2
*c^2*d^8*f^3 - 32*B^2*c^6*d^4*f^3 - 16*B^2*c^8*d^2*f^3) + ((((96*B^4*c^2*d^
4*f^4 - 16*B^4*d^6*f^4 - 144*B^4*c^4*d^2*f^4)^(1/2) + 4*B^2*c^3*f^2 - 12*B^
2*c*d^2*f^2)/(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))^(1/2)*(((
((96*B^4*c^2*d^4*f^4 - 16*B^4*d^6*f^4 - 144*B^4*c^4*d^2*f^4)^(1/2) + 4*B^2*
c^3*f^2 - 12*B^2*c*d^2*f^2)/(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*
f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 +
640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5))/4 +
 32*B*d^12*f^4 + 96*B*c^2*d^10*f^4 + 64*B*c^4*d^8*f^4 - 64*B*c^6*d^6*f^4 -
96*B*c^8*d^4*f^4 - 32*B*c^10*d^2*f^4))/4)*(((96*B^4*c^2*d^4*f^4 - 16*B^4*d^
6*f^4 - 144*B^4*c^4*d^2*f^4)^(1/2) + 4*B^2*c^3*f^2 - 12*B^2*c*d^2*f^2)/(c^6
*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))^(1/2))/4 - 24*B^3*c^3*d^6*
f^2 - 24*B^3*c^5*d^4*f^2 - 8*B^3*c^7*d^2*f^2 - 8*B^3*c*d^8*f^2)*(((96*B^4*c
^2*d^4*f^4 - 16*B^4*d^6*f^4 - 144*B^4*c^4*d^2*f^4)^(1/2) + 4*B^2*c^3*f^2 -
12*B^2*c*d^2*f^2)/(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))^(1/2
))/4 + (log(- (((c + d*tan(e + f*x))^(1/2)*(16*B^2*d^10*f^3 + 32*B^2*c^2*d^
8*f^3 - 32*B^2*c^6*d^4*f^3 - 16*B^2*c^8*d^2*f^3) + ((-((96*B^4*c^2*d^4*f^4
- 16*B^4*d^6*f^4 - 144*B^4*c^4*d^2*f^4)^(1/2) - 4*B^2*c^3*f^2 + 12*B^2*c*d^
2*f^2)/(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))^(1/2)*(((-((96*
B^4*c^2*d^4*f^4 - 16*B^4*d^6*f^4 - 144*B^4*c^4*d^2*f^4)^(1/2) - 4*B^2*c^3*f
^2 + 12*B^2*c*d^2*f^2)/(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))
^(1/2)*(c + d*tan(e + f*x))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c
^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5))/4 + 32*B
*d^12*f^4 + 96*B*c^2*d^10*f^4 + 64*B*c^4*d^8*f^4 - 64*B*c^6*d^6*f^4 - 96*B*
c^8*d^4*f^4 - 32*B*c^10*d^2*f^4))/4)*(-((96*B^4*c^2*d^4*f^4 - 16*B^4*d^6*f^
4 - 144*B^4*c^4*d^2*f^4)^(1/2) - 4*B^2*c^3*f^2 + 12*B^2*c*d^2*f^2)/(c^6*f^4
 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))^(1/2))/4 - 24*B^3*c^3*d^6*f^2
- 24*B^3*c^5*d^4*f^2 - 8*B^3*c^7*d^2*f^2 - 8*B^3*c*d^8*f^2)*(-((96*B^4*c^2*
d^4*f^4 - 16*B^4*d^6*f^4 - 144*B^4*c^4*d^2*f^4)^(1/2) - 4*B^2*c^3*f^2 + 12*
B^2*c*d^2*f^2)/(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))^(1/2))/
4 - log(((c + d*tan(e + f*x))^(1/2)*(16*B^2*d^10*f^3 + 32*B^2*c^2*d^8*f^3 -

32*B^2*c^6*d^4*f^3 - 16*B^2*c^8*d^2*f^3) + (((96*B^4*c^2*d^4*f^4 - 16*B^4*d^6*f^4 - 144*B^4*c^4*d^2*f^4)^(1/2) + 4*B^2*c^3*f^2 - 12*B^2*c*d^2*f^2)/(16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4))^(1/2)*((((96*B^4*c^2*d^4*f^4 - 16*B^4*d^6*f^4 - 144*B^4*c^4*d^2*f^4)^(1/2) + 4*B^2*c^3*f^2 - 12*B^2*c*d^2*f^2)/(16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 32*B*d^12*f^4 - 96*B*c^2*d^10*f^4 - 64*B*c^4*d^8*f^4 + 64*B*c^6*d^6*f^4 + 96*B*c^8*d^4*f^4 + 32*B*c^10*d^2*f^4))*(((96*B^4*c^2*d^4*f^4 - 16*B^4*d^6*f^4 - 144*B^4*c^4*d^2*f^4)^(1/2) + 4*B^2*c^3*f^2 - 12*B^2*c*d^2*f^2)/(16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4))^(1/2) - 24*B^3*c^3*d^6*f^2 - 24*B^3*c^5*d^4*f^2 - 8*B^3*c^7*d^2*f^2 - 8*B^3*c*d^8*f^2)*(((96*B^4*c^2*d^4*f^4 - 16*B^4*d^6*f^4 - 144*B^4*c^4*d^2*f^4)^(1/2) + 4*B^2*c^3*f^2 - 12*B^2*c*d^2*f^2)/(16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4))^(1/2) - log(((c + d*tan(e + f*x))^(1/2)*(16*B^2*d^10*f^3 + 32*B^2*c^2*d^8*f^3 - 32*B^2*c^6*d^4*f^3 - 16*B^2*c^8*d^2*f^3) + (-((96*B^4*c^2*d^4*f^4 - 16*B^4*d^6*f^4 - 144*B^4*c^4*d^2*f^4)^(1/2) - 4*B^2*c^3*f^2 + 12*B^2*c*d^2*f^2)/(16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4))^(1/2)*((-((96*B^4*c^2*d^4*f^4 - 16*B^4*d^6*f^4 - 144*B^4*c^4*d^2*f^4)^(1/2) - 4*B^2*c^3*f^2 + 12*B^2*c*d^2*f^2)/(16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 32*B*d^12*f^4 - 96*B*c^2*d^10*f^4 - 64*B*c^4*d^8*f^4 + 64*B*c^6*d^6*f^4 + 96*B*c^8*d^4*f^4 + 32*B*c^10*d^2*f^4))*(-((96*B^4*c^2*d^4*f^4 - 16*B^4*d^6*f^4 - 144*B^4*c^4*d^2*f^4)^(1/2) - 4*B^2*c^3*f^2 + 12*B^2*c*d^2*f^2)/(16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4))^(1/2) - 24*B^3*c^3*d^6*f^2 - 24*B^3*c^5*d^4*f^2 - 8*B^3*c^7*d^2*f^2 - 8*B^3*c*d^8*f^2)*(-((96*B^4*c^2*d^4*f^4 - 16*B^4*d^6*f^4 - 144*B^4*c^4*d^2*f^4)^(1/2) - 4*B^2*c^3*f^2 + 12*B^2*c*d^2*f^2)/(16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4))^(1/2) - (2*A*d)/(f*(c^2 + d^2)*(c + d*tan(e + f*x))^(1/2)) + (2*B*c)/(f*(c^2 + d^2)*(c + d*tan(e + f*x))^(1/2)) - (2*C*c^2)/(d*f*(c^2 + d^2)*(c + d*tan(e + f*x))^(1/2))

3.120 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}}\,dx$

Optimal result 1714
Rubi [A] (verified) 1715
Mathematica [A] (verified) 1718
Maple [B] (verified) 1718
Fricas [F(-1)] 1719
Sympy [F] 1719
Maxima [F(-2)] 1719
Giac [F(-1)] 1720
Mupad [F(-1)] 1720

Optimal result

Integrand size = 47, antiderivative size = 262

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}}\,dx = \frac{(A-iB-C)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(ia+b)(c-id)^{3/2}f} + \frac{(iA-B-iC)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(a+ib)(c+id)^{3/2}f} - \frac{2\sqrt{b}(Ab^2-a(bB-aC))\,\text{arctanh}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{(a^2+b^2)\,(bc-ad)^{3/2}f} + \frac{2(c^2C-Bcd+Ad^2)}{(bc-ad)\,(c^2+d^2)\,f\sqrt{c+d\tan(e+fx)}}$$

```
[Out] (A-I*B-C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(I*a+b)/(c-I*d)^(3/
2)/f+(I*A-B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))/(a+I*b)/(c+I
*d)^(3/2)/f-2*(A*b^2-a*(B*b-C*a))*arctanh(b^(1/2)*(c+d*tan(f*x+e))^(1/2)/(-
a*d+b*c)^(1/2))*b^(1/2)/(a^2+b^2)/(-a*d+b*c)^(3/2)/f+2*(A*d^2-B*c*d+C*c^2)/
(-a*d+b*c)/(c^2+d^2)/f/(c+d*tan(f*x+e))^(1/2)
```

Rubi [A] (verified)

Time = 1.41 (sec) , antiderivative size = 262, normalized size of antiderivative = 1.00, number of steps used = 12, number of rules used = 7, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.149, Rules used = {3730, 3734, 3620, 3618, 65, 214, 3715}

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}}\,dx = -\frac{2\sqrt{b}(Ab^2-a(bB-aC))\,\text{arctanh}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{f\,(a^2+b^2)\,(bc-ad)^{3/2}} + \frac{(A-iB-C)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f(b+ia)(c-id)^{3/2}} + \frac{(iA-B-iC)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f(a+ib)(c+id)^{3/2}} + \frac{2(Ad^2-Bcd+c^2C)}{f\,(c^2+d^2)\,(bc-ad)\sqrt{c+d\tan(e+fx)}}$$

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])*(c + d*Ta
n[e + f*x])^(3/2)),x]
```

```
[Out] ((A - I*B - C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]])/((I*a + b)*
(c - I*d)^(3/2)*f) + ((I*A - B - I*C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt
[c + I*d]])/((a + I*b)*(c + I*d)^(3/2)*f) - (2*Sqrt[b]*(A*b^2 - a*(b*B - a*
C))*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*d]])/((a^2 + b^
2)*(b*c - a*d)^(3/2)*f) + (2*(c^2*C - B*c*d + A*d^2))/((b*c - a*d)*(c^2 + d
^2)*f*Sqrt[c + d*Tan[e + f*x]])
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3715

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_) + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :>
 Dist[A/f, Subst[Int[(a + b*x)^m*(c + d*x)^n, x], x, Tan[e + f*x]], x] /; F
reeQ[{a, b, c, d, e, f, A, C, m, n}, x] && EqQ[A, C]
```

Rule 3730

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 - a*(b*B - a*C))*(a + b*Tan[e +
 f*x])^(m + 1)*((c + d*Tan[e + f*x])^(n + 1)/(f*(m + 1)*(b*c - a*d)*(a^2 +
b^2))), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3734

```
Int[(((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2))/((a_.) + (b_.)*tan[(e_.)
+ (f_.)*(x_)]), x_Symbol] :> Dist[1/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n
*Simp[b*B + a*(A - C) + (a*B - b*(A - C))*Tan[e + f*x], x], x], x] + Dist[(
A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n*((1 + Tan[e
+ f*x]^2)/(a + b*Tan[e + f*x])), x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C
, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&
 !GtQ[n, 0] &&  !LeQ[n, -1]
```

Rubi steps

$$\text{integral} = \frac{2(c^2C - Bcd + Ad^2)}{(bc-ad)\,(c^2+d^2)\,f\sqrt{c+d\tan(e+fx)}} + \frac{2\int \frac{\frac{1}{2}\left(-aAcd+ad(cC-Bd)+Ab\left(c^2+d^2\right)\right)+\frac{1}{2}(bc-ad)(Bc-(A-C)d)\tan(e+fx)+\frac{1}{2}b\left(c^2C-Bcd+Ad^2\right)\tan^2(e+fx)}{(a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}}\,dx}{(bc-ad)\,(c^2+d^2)}$$

$$= \frac{2(c^2C - Bcd + Ad^2)}{(bc-ad)(c^2+d^2)f\sqrt{c+d\tan(e+fx)}} + \frac{(b(Ab^2 - a(bB-aC)))\int \frac{1+\tan^2(e+fx)}{(a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}}\,dx}{(a^2+b^2)(bc-ad)} + \frac{2\int \frac{\frac{1}{2}(bc-ad)(bBc-b(A-C)d+a(Ac-cC+Bd))+\frac{1}{2}(bc-ad)(aBc+bcC-bBd+aCd-A(bc+ad))\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{(a^2+b^2)(bc-ad)(c^2+d^2)}$$

$$= \frac{2(c^2C - Bcd + Ad^2)}{(bc-ad)(c^2+d^2)f\sqrt{c+d\tan(e+fx)}} + \frac{(A-iB-C)\int \frac{1+i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(a-ib)(c-id)} + \frac{(A+iB-C)\int \frac{1-i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(a+ib)(c+id)} + \frac{(b(Ab^2 - a(bB-aC)))\,\mathrm{Subst}\left(\int \frac{1}{(a+bx)\sqrt{c+dx}}\,dx, x, \tan(e+fx)\right)}{(a^2+b^2)(bc-ad)f}$$

$$= \frac{2(c^2C - Bcd + Ad^2)}{(bc-ad)(c^2+d^2)f\sqrt{c+d\tan(e+fx)}} + \frac{(iA+B-iC)\mathrm{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c-idx}}\,dx, x, i\tan(e+fx)\right)}{2(a-ib)(c-id)f} - \frac{(i(A+iB-C))\mathrm{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c+idx}}\,dx, x, -i\tan(e+fx)\right)}{2(a+ib)(c+id)f} + \frac{(2b(Ab^2 - a(bB-aC)))\,\mathrm{Subst}\left(\int \frac{1}{a-\frac{bc}{d}+\frac{bx^2}{d}}\,dx, x, \sqrt{c+d\tan(e+fx)}\right)}{(a^2+b^2)d(bc-ad)f}$$

$$= -\frac{2\sqrt{b}(Ab^2 - a(bB-aC))\,\mathrm{arctanh}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{(a^2+b^2)(bc-ad)^{3/2}f} + \frac{2(c^2C - Bcd + Ad^2)}{(bc-ad)(c^2+d^2)f\sqrt{c+d\tan(e+fx)}} - \frac{(A-iB-C)\mathrm{Subst}\left(\int \frac{1}{-1-\frac{ic}{d}+\frac{ix^2}{d}}\,dx, x, \sqrt{c+d\tan(e+fx)}\right)}{(a-ib)(c-id)df} - \frac{(A+iB-C)\mathrm{Subst}\left(\int \frac{1}{-1+\frac{ic}{d}-\frac{ix^2}{d}}\,dx, x, \sqrt{c+d\tan(e+fx)}\right)}{(a+ib)(c+id)df}$$

$$= -\frac{(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(a-ib)(c-id)^{3/2}f} - \frac{(A+iB-C)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(ia-b)(c+id)^{3/2}f}$$
$$- \frac{2\sqrt{b}(Ab^2-a(bB-aC))\,\text{arctanh}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{(a^2+b^2)\,(bc-ad)^{3/2}f}$$
$$+ \frac{2(c^2C-Bcd+Ad^2)}{(bc-ad)\,(c^2+d^2)\,f\sqrt{c+d\tan(e+fx)}}$$

Mathematica [A] (verified)

Time = 5.35 (sec) , antiderivative size = 296, normalized size of antiderivative = 1.13

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}}\,dx = -\frac{\frac{i\left(\frac{(a+ib)(A-iB-C)(c+id)(-bc+ad)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{\sqrt{c-id}}+\frac{(a-ib)(A+iB\cdots}{}\right)}{a^2+b^2}}{}$$

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])*(c
+ d*Tan[e + f*x])^(3/2)),x]
```

```
[Out] (((-I)*(((a + I*b)*(A - I*B - C)*(c + I*d)*(-(b*c) + a*d)*ArcTanh[Sqrt[c +
d*Tan[e + f*x]]/Sqrt[c - I*d]])/Sqrt[c - I*d] + ((a - I*b)*(A + I*B - C)*(c
 - I*d)*(b*c - a*d)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]])/Sqrt[c
 + I*d]))/(a^2 + b^2) + (2*Sqrt[b]*(A*b^2 + a*(-(b*B) + a*C))*(c^2 + d^2)*A
rcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*d]])/((a^2 + b^2)*Sq
rt[b*c - a*d]) - (2*(c^2*C - B*c*d + A*d^2))/Sqrt[c + d*Tan[e + f*x]])/((-(
b*c) + a*d)*(c^2 + d^2)*f)
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 26342 vs. 2(229) = 458.

Time = 0.15 (sec) , antiderivative size = 26343, normalized size of antiderivative = 100.55

method	result	size
derivativedivides	Expression too large to display	26343
default	Expression too large to display	26343

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))^(3/2)
,x,method=_RETURNVERBOSE)
```

```
[Out] result too large to display
```

Fricas [F(-1)]

Timed out.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))
^(3/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}}\,dx = \int \frac{A+B\tan{\left(e+fx\right)}+C\tan^2{\left(e+fx\right)}}{\left(a+b\tan{\left(e+fx\right)}\right)\left(c+d\tan{\left(e+fx\right)}\right)^{\frac{3}{2}}}\,dx$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e)
)**(3/2),x)
```

```
[Out] Integral((A + B*tan(e + f*x) + C*tan(e + f*x)**2)/((a + b*tan(e + f*x))*(c
+ d*tan(e + f*x))**(3/2)), x)
```

Maxima [F(-2)]

Exception generated.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}}\,dx = \text{Exception raised: ValueError}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))
^(3/2),x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 detail
```

Giac [F(-1)]

Timed out.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))
^(3/2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}}\,dx = \text{Hanged}$$

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))*(c + d*ta
n(e + f*x))^(3/2)),x)
```

```
[Out] \text{Hanged}
```

3.121 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}}\,dx$

Optimal result 1721
Rubi [A] (verified) 1722
Mathematica [B] (verified) 1726
Maple [B] (verified) 1727
Fricas [F(-1)] 1727
Sympy [F] 1728
Maxima [F(-2)] 1728
Giac [F(-1)] 1728
Mupad [F(-1)] 1729

Optimal result

Integrand size = 47, antiderivative size = 447

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}}\,dx = -\frac{(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(a-ib)^2(c-id)^{3/2}f} - \frac{(B-i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(a+ib)^2(c+id)^{3/2}f}$$
$$-\frac{\sqrt{b}(5a^3bBd-3a^4Cd+b^4(2Bc-3Ad)+ab^3(4Ac-4cC+Bd)-a^2b^2(2Bc+(7A-C)d))\operatorname{arctanh}\Big(\frac{\sqrt{b}\sqrt{}}{}}{(a^2+b^2)^2(bc-ad)^{5/2}f}$$
$$-\frac{d(2b^2c(cC-Bd)-abB(c^2+d^2)+a^2(3c^2C-2Bcd+Cd^2)+A(2a^2d^2+b^2(c^2+3d^2)))}{(a^2+b^2)(bc-ad)^2(c^2+d^2)f\sqrt{c+d\tan(e+fx)}}$$
$$-\frac{Ab^2-a(bB-aC)}{(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}}$$

```
[Out] -(I*A+B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(a-I*b)^2/(c-I*d
)^(3/2)/f-(B-I*(A-C))*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))/(a+I*b)
^2/(c+I*d)^(3/2)/f-(5*a^3*b*B*d-3*a^4*C*d+b^4*(-3*A*d+2*B*c)+a*b^3*(4*A*c+B
*d-4*C*c)-a^2*b^2*(2*B*c+(7*A-C)*d))*arctanh(b^(1/2)*(c+d*tan(f*x+e))^(1/2)
/(-a*d+b*c)^(1/2))*b^(1/2)/(a^2+b^2)^2/(-a*d+b*c)^(5/2)/f-d*(2*b^2*c*(-B*d+
C*c)-a*b*B*(c^2+d^2)+a^2*(-2*B*c*d+3*C*c^2+C*d^2)+A*(2*a^2*d^2+b^2*(c^2+3*d
^2)))/(a^2+b^2)/(-a*d+b*c)^2/(c^2+d^2)/f/(c+d*tan(f*x+e))^(1/2)+(-A*b^2+a*(
B*b-C*a))/(a^2+b^2)/(-a*d+b*c)/f/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f*x+e))
```

Rubi [A] (verified)

Time = 3.13 (sec) , antiderivative size = 446, normalized size of antiderivative = 1.00, number of steps used = 13, number of rules used = 7, $\frac{\text{number of rules}}{\text{integrand size}} = 0.149$, Rules used = {3730, 3734, 3620, 3618, 65, 214, 3715}

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}}\,dx =$$
$$-\frac{d(2a^2Ad^2+a^2(-2Bcd+3c^2C+Cd^2)-abB(c^2+d^2)+Ab^2(c^2+3d^2)+2b^2c(cC-Bd))}{f(a^2+b^2)(c^2+d^2)(bc-ad)^2\sqrt{c+d\tan(e+fx)}}$$
$$-\frac{Ab^2-a(bB-aC)}{f(a^2+b^2)(bc-ad)(a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}}$$
$$-\frac{\sqrt{b}(-3a^4Cd+5a^3bBd-a^2b^2(d(7A-C)+2Bc)+ab^3(4Ac+Bd-4cC)+b^4(2Bc-3Ad))\operatorname{arctanh}\left(\frac{\sqrt{b}\sqrt{\ldots}}{\ldots}\right)}{f(a^2+b^2)^2(bc-ad)^{5/2}}$$
$$-\frac{(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f(a-ib)^2(c-id)^{3/2}}-\frac{(B-i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f(a+ib)^2(c+id)^{3/2}}$$

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^2*(c + d*
Tan[e + f*x])^(3/2)),x]
```

```
[Out] -(((I*A + B - I*C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]])/((a - I
*b)^2*(c - I*d)^(3/2)*f)) - ((B - I*(A - C))*ArcTanh[Sqrt[c + d*Tan[e + f*x
]]/Sqrt[c + I*d]])/((a + I*b)^2*(c + I*d)^(3/2)*f) - (Sqrt[b]*(5*a^3*b*B*d
 - 3*a^4*C*d + b^4*(2*B*c - 3*A*d) + a*b^3*(4*A*c - 4*c*C + B*d) - a^2*b^2*(
2*B*c + (7*A - C)*d))*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c -
 a*d]])/((a^2 + b^2)^2*(b*c - a*d)^(5/2)*f) - (d*(2*a^2*A*d^2 + 2*b^2*c*(c*
C - B*d) - a*b*B*(c^2 + d^2) + A*b^2*(c^2 + 3*d^2) + a^2*(3*c^2*C - 2*B*c*d
 + C*d^2)))/((a^2 + b^2)*(b*c - a*d)^2*(c^2 + d^2)*f*Sqrt[c + d*Tan[e + f*x
]]) - (A*b^2 - a*(b*B - a*C))/((a^2 + b^2)*(b*c - a*d)*f*(a + b*Tan[e + f*x
])*Sqrt[c + d*Tan[e + f*x]])
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3715

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_) + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :>
 Dist[A/f, Subst[Int[(a + b*x)^m*(c + d*x)^n, x], x, Tan[e + f*x]], x] /; F
reeQ[{a, b, c, d, e, f, A, C, m, n}, x] && EqQ[A, C]
```

Rule 3730

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 - a*(b*B - a*C))*(a + b*Tan[e +
 f*x])^(m + 1)*((c + d*Tan[e + f*x])^(n + 1)/(f*(m + 1)*(b*c - a*d)*(a^2 +
b^2))), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3734

```
Int[(((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2))/((a_.) + (b_.)*tan[(e_.)
+ (f_.)*(x_)]), x_Symbol] :> Dist[1/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n
*Simp[b*B + a*(A - C) + (a*B - b*(A - C))*Tan[e + f*x], x], x], x] + Dist[(
A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n*((1 + Tan[e
+ f*x]^2)/(a + b*Tan[e + f*x])), x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C
, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&
 !GtQ[n, 0] &&  !LeQ[n, -1]
```

Rubi steps

$$\text{integral} = -\frac{Ab^2 - a(bB - aC)}{(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}} - \frac{\int \frac{\frac{1}{2}(3Ab^2d - 2aA(bc-ad) - (bB-aC)(2bc+ad)) + (Ab-aB-bC)(bc-ad)\tan(e+fx) + \frac{3}{2}(Ab^2 - a(bB-aC))d\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}}\,dx}{(a^2+b^2)(bc-ad)}$$

$$= -\frac{d(2a^2Ad^2 + 2b^2c(cC - Bd) - abB(c^2+d^2) + Ab^2(c^2+3d^2) + a^2(3c^2C - 2Bcd + Cd^2))}{(a^2+b^2)(bc-ad)^2(c^2+d^2)f\sqrt{c+d\tan(e+fx)}} - \frac{Ab^2 - a(bB-aC)}{(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}} - \frac{2\int \frac{\frac{1}{4}(-2a^3d^2(Ac-cC+Bd) + a^2b(4A-C)d(c^2+d^2) - b^3(2Bc-3Ad)(c^2+d^2) + ab^2(2c^3C + Bc^2d + 4cCd^2 - Bd^3 - 2A(c^3+2cd^2))) + \frac{1}{2}(\cdots}{(a+\cdots}}{(a^2+\cdots}$$

$$= -\frac{d(2a^2Ad^2 + 2b^2c(cC - Bd) - abB(c^2+d^2) + Ab^2(c^2+3d^2) + a^2(3c^2C - 2Bcd + Cd^2))}{(a^2+b^2)(bc-ad)^2(c^2+d^2)f\sqrt{c+d\tan(e+fx)}} - \frac{Ab^2 - a(bB-aC)}{(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}} - \frac{2\int \frac{-\frac{1}{2}(bc-ad)^2(a^2(Ac-cC+Bd) - b^2(Ac-cC+Bd) + 2ab(Bc-(A-C)d)) + \frac{1}{2}(bc-ad)^2(2ab(Ac-cC+Bd) - a^2(Bc-(A-C)d) + b^2(B\cdots}{\sqrt{c+d\tan(e+fx)}}}{(a^2+b^2)^2(bc-ad)^2(c^2+d^2)} - \frac{(2(-\frac{1}{2}ab(bc-ad)^2(Abc - aBc - bcC + aAd + bBd - aCd) + \frac{1}{4}a^2bd(2a^2Ad^2 + 2b^2c(cC - Bd) - \cdots}{\cdots}$$

$$= -\frac{d(2a^2Ad^2 + 2b^2c(cC - Bd) - abB(c^2+d^2) + Ab^2(c^2+3d^2) + a^2(3c^2C - 2Bcd + Cd^2))}{(a^2+b^2)(bc-ad)^2(c^2+d^2)f\sqrt{c+d\tan(e+fx)}} - \frac{Ab^2 - a(bB-aC)}{(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}} + \frac{(A-iB-C)\int \frac{1+i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(a-ib)^2(c-id)} + \frac{(A+iB-C)\int \frac{1-i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(a+ib)^2(c+id)} - \frac{(2(-\frac{1}{2}ab(bc-ad)^2(Abc - aBc - bcC + aAd + bBd - aCd) + \frac{1}{4}a^2bd(2a^2Ad^2 + 2b^2c(cC - Bd) - \cdots}{\cdots}$$

$$=$$

$$-\frac{d(2a^2Ad^2+2b^2c(cC-Bd)-abB(c^2+d^2)+Ab^2(c^2+3d^2)+a^2(3c^2C-2Bcd+Cd^2))}{(a^2+b^2)(bc-ad)^2(c^2+d^2)f\sqrt{c+d\tan(e+fx)}}$$
$$-\frac{Ab^2-a(bB-aC)}{(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}}$$
$$+\frac{(iA+B-iC)\text{Subst}\left(\int\frac{1}{(-1+x)\sqrt{c-idx}}\,dx,x,i\tan(e+fx)\right)}{2(a-ib)^2(c-id)f}$$
$$-\frac{(i(A+iB-C))\text{Subst}\left(\int\frac{1}{(-1+x)\sqrt{c+idx}}\,dx,x,-i\tan(e+fx)\right)}{2(a+ib)^2(c+id)f}$$
$$-\frac{\left(4\left(-\frac{1}{2}ab(bc-ad)^2(Abc-aBc-bcC+aAd+bBd-aCd)+\frac{1}{4}a^2bd(2a^2Ad^2+2b^2c(cC-Bd)\right.\right.}{}$$

$$=$$

$$-\frac{\sqrt{b}(5a^3bBd-3a^4Cd+b^4(2Bc-3Ad)+ab^3(4Ac-4cC+Bd)-a^2b^2(2Bc+7Ad-Cd))\,\text{arct}}{(a^2+b^2)^2(bc-ad)^{5/2}f}$$
$$-\frac{d(2a^2Ad^2+2b^2c(cC-Bd)-abB(c^2+d^2)+Ab^2(c^2+3d^2)+a^2(3c^2C-2Bcd+Cd^2))}{(a^2+b^2)(bc-ad)^2(c^2+d^2)f\sqrt{c+d\tan(e+fx)}}$$
$$-\frac{Ab^2-a(bB-aC)}{(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}}$$
$$-\frac{(A-iB-C)\text{Subst}\left(\int\frac{1}{-1-\frac{ic}{d}+\frac{ix^2}{d}}\,dx,x,\sqrt{c+d\tan(e+fx)}\right)}{(a-ib)^2(c-id)df}$$
$$-\frac{(A+iB-C)\text{Subst}\left(\int\frac{1}{-1+\frac{ic}{d}-\frac{ix^2}{d}}\,dx,x,\sqrt{c+d\tan(e+fx)}\right)}{(a+ib)^2(c+id)df}$$

$$=-\frac{(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(a-ib)^2(c-id)^{3/2}f}-\frac{(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(a+ib)^2(c+id)^{3/2}f}$$
$$-\frac{\sqrt{b}(5a^3bBd-3a^4Cd+b^4(2Bc-3Ad)+ab^3(4Ac-4cC+Bd)-a^2b^2(2Bc+7Ad-Cd))\,\text{arct}}{(a^2+b^2)^2(bc-ad)^{5/2}f}$$
$$-\frac{d(2a^2Ad^2+2b^2c(cC-Bd)-abB(c^2+d^2)+Ab^2(c^2+3d^2)+a^2(3c^2C-2Bcd+Cd^2))}{(a^2+b^2)(bc-ad)^2(c^2+d^2)f\sqrt{c+d\tan(e+fx)}}$$
$$-\frac{Ab^2-a(bB-aC)}{(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}}$$

Mathematica [B] (verified)

Both result and optimal contain complex but leaf count is larger than twice the leaf count of optimal. 2078 vs. $2(447) = 894$.

Time = 6.43 (sec) , antiderivative size = 2078, normalized size of antiderivative = 4.65

$$\int \frac{A + B\tan(e+fx) + C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}}\,dx = \text{Result too large to show}$$

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^2*(
c + d*Tan[e + f*x])^(3/2)),x]
```

```
[Out] -((A*b^2 - a*(b*B - a*C))/((a^2 + b^2)*(b*c - a*d)*f*(a + b*Tan[e + f*x])*S
qrt[c + d*Tan[e + f*x]])) - ((-2*(((I*Sqrt[c - I*d]*((b*(-(b*c) + a*d)*((-3
*(A*b^2 - a*(b*B - a*C))*d^2)/2 - c*(A*b - a*B - b*C)*(b*c - a*d) + (d*(3*A
*b^2*d - 2*a*A*(b*c - a*d) - (b*B - a*C)*(2*b*c + a*d)))/2))/2 + a*(-1/2*(a
*d*((-3*c*(A*b^2 - a*(b*B - a*C))*d)/2 + (A*b - a*B - b*C)*d*(b*c - a*d)))
+ (((b*d^2)/2 - (c*(-(b*c) + a*d))/2)*(3*A*b^2*d - 2*a*A*(b*c - a*d) - (b*B
 - a*C)*(2*b*c + a*d)))/2 - (b*(-(c*((-3*c*(A*b^2 - a*(b*B - a*C))*d)/2 + (
A*b - a*B - b*C)*d*(b*c - a*d))) + (d^2*(3*A*b^2*d - 2*a*A*(b*c - a*d) - (b
*B - a*C)*(2*b*c + a*d)))/2))/2) - I*((a*(-(b*c) + a*d)*((-3*(A*b^2 - a*(b*
B - a*C))*d^2)/2 - c*(A*b - a*B - b*C)*(b*c - a*d) + (d*(3*A*b^2*d - 2*a*A*
(b*c - a*d) - (b*B - a*C)*(2*b*c + a*d)))/2))/2 - b*(-1/2*(a*d*((-3*c*(A*b^
2 - a*(b*B - a*C))*d)/2 + (A*b - a*B - b*C)*d*(b*c - a*d))) + (((b*d^2)/2 -
 (c*(-(b*c) + a*d))/2)*(3*A*b^2*d - 2*a*A*(b*c - a*d) - (b*B - a*C)*(2*b*c
+ a*d)))/2 - (b*(-(c*((-3*c*(A*b^2 - a*(b*B - a*C))*d)/2 + (A*b - a*B - b*C
)*d*(b*c - a*d))) + (d^2*(3*A*b^2*d - 2*a*A*(b*c - a*d) - (b*B - a*C)*(2*b*
c + a*d)))/2))/2)))*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]])/((-c +
 I*d)*f) - (I*Sqrt[c + I*d]*((b*(-(b*c) + a*d)*((-3*(A*b^2 - a*(b*B - a*C))
*d^2)/2 - c*(A*b - a*B - b*C)*(b*c - a*d) + (d*(3*A*b^2*d - 2*a*A*(b*c - a*
d) - (b*B - a*C)*(2*b*c + a*d)))/2))/2 + a*(-1/2*(a*d*((-3*c*(A*b^2 - a*(b*
B - a*C))*d)/2 + (A*b - a*B - b*C)*d*(b*c - a*d))) + (((b*d^2)/2 - (c*(-(b*
c) + a*d))/2)*(3*A*b^2*d - 2*a*A*(b*c - a*d) - (b*B - a*C)*(2*b*c + a*d)))/
2 - (b*(-(c*((-3*c*(A*b^2 - a*(b*B - a*C))*d)/2 + (A*b - a*B - b*C)*d*(b*c
- a*d))) + (d^2*(3*A*b^2*d - 2*a*A*(b*c - a*d) - (b*B - a*C)*(2*b*c + a*d))
)/2))/2) + I*((a*(-(b*c) + a*d)*((-3*(A*b^2 - a*(b*B - a*C))*d^2)/2 - c*(A*
b - a*B - b*C)*(b*c - a*d) + (d*(3*A*b^2*d - 2*a*A*(b*c - a*d) - (b*B - a*C
)*(2*b*c + a*d)))/2))/2 - b*(-1/2*(a*d*((-3*c*(A*b^2 - a*(b*B - a*C))*d)/2
+ (A*b - a*B - b*C)*d*(b*c - a*d))) + (((b*d^2)/2 - (c*(-(b*c) + a*d))/2)*(
3*A*b^2*d - 2*a*A*(b*c - a*d) - (b*B - a*C)*(2*b*c + a*d)))/2 - (b*(-(c*((-
3*c*(A*b^2 - a*(b*B - a*C))*d)/2 + (A*b - a*B - b*C)*d*(b*c - a*d))) + (d^2
*(3*A*b^2*d - 2*a*A*(b*c - a*d) - (b*B - a*C)*(2*b*c + a*d)))/2))/2)))*ArcT
anh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]])/((-c - I*d)*f))/(a^2 + b^2) +
(2*Sqrt[b*c - a*d]*(-1/2*(a*b*(-(b*c) + a*d)*((-3*(A*b^2 - a*(b*B - a*C))*d
^2)/2 - c*(A*b - a*B - b*C)*(b*c - a*d) + (d*(3*A*b^2*d - 2*a*A*(b*c - a*d)
```

```
 - (b*B - a*C)*(2*b*c + a*d)))/2)) + (a^2*b*(-(c*((-3*c*(A*b^2 - a*(b*B - a
*C))*d)/2 + (A*b - a*B - b*C)*d*(b*c - a*d))) + (d^2*(3*A*b^2*d - 2*a*A*(b*
c - a*d) - (b*B - a*C)*(2*b*c + a*d)))/2))/2 + b^2*(-1/2*(a*d*((-3*c*(A*b^2
 - a*(b*B - a*C))*d)/2 + (A*b - a*B - b*C)*d*(b*c - a*d))) + (((b*d^2)/2 -
(c*(-(b*c) + a*d))/2)*(3*A*b^2*d - 2*a*A*(b*c - a*d) - (b*B - a*C)*(2*b*c +
 a*d)))/2))*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*d]])/(S
qrt[b]*(a^2 + b^2)*(-(b*c) + a*d)*f)))/((-(b*c) + a*d)*(c^2 + d^2)) - (2*(-
(c*((-3*c*(A*b^2 - a*(b*B - a*C))*d)/2 + (A*b - a*B - b*C)*d*(b*c - a*d)))
+ (d^2*(3*A*b^2*d - 2*a*A*(b*c - a*d) - (b*B - a*C)*(2*b*c + a*d)))/2))/((-
(b*c) + a*d)*(c^2 + d^2)*f*Sqrt[c + d*Tan[e + f*x]]))/((a^2 + b^2)*(b*c - a
*d))
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 40618 vs. $2(411) = 822$.

Time = 0.24 (sec) , antiderivative size = 40619, normalized size of antiderivative = 90.87

method	result	size
derivativedivides	Expression too large to display	40619
default	Expression too large to display	40619

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e))^(3/
2),x,method=_RETURNVERBOSE)
```

```
[Out] result too large to display
```

Fricas [F(-1)]

Timed out.

$$\int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{(a + b\tan(e + fx))^2(c + d\tan(e + fx))^{3/2}}\, dx = \text{Timed out}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e
))^(3/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}}\,dx = \int \frac{A+B\tan{(e+fx)}+C\tan^2{(e+fx)}}{(a+b\tan{(e+fx)})^2\,(c+d\tan{(e+fx)})^{\frac{3}{2}}}\,dx$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e))**2/(c+d*tan(f*x
+e))**(3/2),x)
```

```
[Out] Integral((A + B*tan(e + f*x) + C*tan(e + f*x)**2)/((a + b*tan(e + f*x))**2*
(c + d*tan(e + f*x))**(3/2)), x)
```

Maxima [F(-2)]

Exception generated.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}}\,dx = \text{Exception raised: ValueError}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e
))^(3/2),x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 detail
```

Giac [F(-1)]

Timed out.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e
))^(3/2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}}\,dx=\text{Hanged}$$

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))^2*(c + d*
tan(e + f*x))^(3/2)),x)
```

```
[Out] \text{Hanged}
```

3.122 $\int \frac{(a+b\tan(e+fx))^3(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx$

Optimal result 1730
Rubi [A] (verified) 1731
Mathematica [C] (verified) 1736
Maple [B] (verified) 1737
Fricas [F(-1)] 1737
Sympy [F] 1738
Maxima [F(-1)] 1738
Giac [F(-1)] 1738
Mupad [F(-1)] 1738

Optimal result

Integrand size = 47, antiderivative size = 585

$$\int \frac{(a+b\tan(e+fx))^3\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx =$$

$$-\frac{(a-ib)^3(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(c-id)^{5/2}f}$$

$$-\frac{(ia-b)^3(A+iB-C)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(c+id)^{5/2}f}$$

$$-\frac{2(c^2C-Bcd+Ad^2)\,(a+b\tan(e+fx))^3}{3d\,(c^2+d^2)\,f(c+d\tan(e+fx))^{3/2}}$$

$$-\frac{2(b(2c^4C-Bc^3d+4c^2Cd^2-3Bcd^3+2Ad^4)+ad^2(2c(A-C)d-B(c^2-d^2)))\,(a+b\tan(e+fx))^2}{d^2\,(c^2+d^2)^2\,f\sqrt{c+d\tan(e+fx)}}$$

$$+\frac{2b(3abd(8c^4C-2Bc^3d-c^2(A-17C)d^2-8Bcd^3+(5A+3C)d^4)-b^2(16c^5C-8Bc^4d+2c^3(A+15C)d^2 \ldots}{3d^4\,(c^2+d^2)^2\,\ldots}$$

$$+\frac{2b^2(b(8c^4C-4Bc^3d+c^2(A+15C)d^2-10Bcd^3+(7A+C)d^4)+3ad^2(2c(A-C)d-B(c^2-d^2)))\tan(e+\ldots}{3d^3\,(c^2+d^2)^2\,f}$$

```
[Out] -(a-I*b)^3*(I*A+B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(c-I*d
)^(5/2)/f-(I*a-b)^3*(A+I*B-C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))
/(c+I*d)^(5/2)/f+2/3*b*(3*a*b*d*(8*c^4*C-2*B*c^3*d-c^2*(A-17*C)*d^2-8*B*c*d
^3+(5*A+3*C)*d^4)-b^2*(16*c^5*C-8*B*c^4*d+2*c^3*(A+15*C)*d^2-17*B*c^2*d^3+8
*c*(A+C)*d^4-3*B*d^5)+6*a^2*d^3*(2*c*(A-C)*d-B*(c^2-d^2)))*(c+d*tan(f*x+e))
^(1/2)/d^4/(c^2+d^2)^2/f+2/3*b^2*(b*(8*c^4*C-4*B*c^3*d+c^2*(A+15*C)*d^2-10*
B*c*d^3+(7*A+C)*d^4)+3*a*d^2*(2*c*(A-C)*d-B*(c^2-d^2)))*(c+d*tan(f*x+e))^(1
/2)*tan(f*x+e)/d^3/(c^2+d^2)^2/f-2*(b*(2*A*d^4-B*c^3*d-3*B*c*d^3+2*C*c^4+4*
C*c^2*d^2)+a*d^2*(2*c*(A-C)*d-B*(c^2-d^2)))*(a+b*tan(f*x+e))^2/d^2/(c^2+d^2
```

```
)^2/f/(c+d*tan(f*x+e))^(1/2)-2/3*(A*d^2-B*c*d+C*c^2)*(a+b*tan(f*x+e))^3/d/(
c^2+d^2)/f/(c+d*tan(f*x+e))^(3/2)
```

Rubi [A] (verified)

Time = 3.36 (sec) , antiderivative size = 585, normalized size of antiderivative = 1.00, number of steps used = 11, number of rules used = 7, $\frac{\text{number of rules}}{\text{integrand size}} = 0.149$, Rules used = {3726, 3718, 3711, 3620, 3618, 65, 214}

$$\int \frac{(a+b\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{5/2}}\,dx = \frac{2b\sqrt{c+d\tan(e+fx)}(6a^2d^3(2cd(A-$$

$$-\frac{(-b+ia)^3(A+iB-C)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f(c+id)^{5/2}}$$

$$-\frac{(a-ib)^3(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f(c-id)^{5/2}}$$

$$+\frac{2b^2\tan(e+fx)\sqrt{c+d\tan(e+fx)}(3ad^2(2cd(A-C)-B(c^2-d^2))+b(c^2d^2(A+15C)+d^4(7A+C)-}{3d^3f\left(c^2+d^2\right)^2}$$

$$-\frac{2(Ad^2-Bcd+c^2C)\left(a+b\tan(e+fx)\right)^3}{3df\left(c^2+d^2\right)\left(c+d\tan(e+fx)\right)^{3/2}}$$

$$-\frac{2(a+b\tan(e+fx))^2\left(ad^2(2cd(A-C)-B(c^2-d^2))+b(2Ad^4-Bc^3d-3Bcd^3+2c^4C+4c^2Cd^2)\right)}{d^2f\left(c^2+d^2\right)^2\sqrt{c+d\tan(e+fx)}}$$

```
[In] Int[((a + b*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c + d
*Tan[e + f*x])^(5/2),x]
```

```
[Out] -(((a - I*b)^3*(I*A + B - I*C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*
d]])/((c - I*d)^(5/2)*f)) - ((I*a - b)^3*(A + I*B - C)*ArcTanh[Sqrt[c + d*T
an[e + f*x]]/Sqrt[c + I*d]])/((c + I*d)^(5/2)*f) - (2*(c^2*C - B*c*d + A*d^
2)*(a + b*Tan[e + f*x])^3)/(3*d*(c^2 + d^2)*f*(c + d*Tan[e + f*x])^(3/2)) -
 (2*(b*(2*c^4*C - B*c^3*d + 4*c^2*C*d^2 - 3*B*c*d^3 + 2*A*d^4) + a*d^2*(2*c
*(A - C)*d - B*(c^2 - d^2)))*(a + b*Tan[e + f*x])^2)/(d^2*(c^2 + d^2)^2*f*S
qrt[c + d*Tan[e + f*x]]) + (2*b*(3*a*b*d*(8*c^4*C - 2*B*c^3*d - c^2*(A - 17
*C)*d^2 - 8*B*c*d^3 + (5*A + 3*C)*d^4) - b^2*(16*c^5*C - 8*B*c^4*d + 2*c^3*
(A + 15*C)*d^2 - 17*B*c^2*d^3 + 8*c*(A + C)*d^4 - 3*B*d^5) + 6*a^2*d^3*(2*c
*(A - C)*d - B*(c^2 - d^2)))*Sqrt[c + d*Tan[e + f*x]])/(3*d^4*(c^2 + d^2)^2
*f) + (2*b^2*(b*(8*c^4*C - 4*B*c^3*d + c^2*(A + 15*C)*d^2 - 10*B*c*d^3 + (7
*A + C)*d^4) + 3*a*d^2*(2*c*(A - C)*d - B*(c^2 - d^2)))*Tan[e + f*x]*Sqrt[c
 + d*Tan[e + f*x]])/(3*d^3*(c^2 + d^2)^2*f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
```

```
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3726

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
```

```
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*d^2 + c*(c*C - B*d))*(a + b*Tan[e +
 f*x])^m*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rubi steps

$$\text{integral} = -\frac{2(c^2C - Bcd + Ad^2)\,(a + b\tan(e+fx))^3}{3d\,(c^2+d^2)\,f(c+d\tan(e+fx))^{3/2}} + \frac{2\int \frac{(a+b\tan(e+fx))^2\left(\frac{3}{2}\left(Ad(ac+2bd)+\frac{2}{3}\left(3bc-\frac{3ad}{2}\right)(cC-Bd)\right)+\frac{3}{2}d((A-C)(bc-ad)+B(ac+bd))\tan(e+fx)+\frac{3}{2}b(2c^2C-Bcd+(A+C)d^2)}{(c+d\tan(e+fx))^{3/2}}}{3d\,(c^2+d^2)}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)\,(a + b\tan(e+fx))^3}{3d\,(c^2+d^2)\,f(c+d\tan(e+fx))^{3/2}} - \frac{2(b(2c^4C - Bc^3d + 4c^2Cd^2 - 3Bcd^3 + 2Ad^4) + ad^2(2c(A-C)d - B(c^2-d^2)))\,(a+b\tan(e+}{d^2\,(c^2+d^2)^2\,f\sqrt{c+d\tan(e+fx)}} + \frac{4\int \frac{(a+b\tan(e+fx))\left(-\frac{3}{4}((4bc-ad)(ad^2(Bc-(A-C)d)-b(2cC-Bd)(c^2+d^2))-d(ac+4bd)(ad(Ac-cC+Bd)+2b(c^2C-Bcd+Ad^2}{}}{}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)\,(a + b\tan(e+fx))^3}{3d\,(c^2+d^2)\,f(c+d\tan(e+fx))^{3/2}} - \frac{2(b(2c^4C - Bc^3d + 4c^2Cd^2 - 3Bcd^3 + 2Ad^4) + ad^2(2c(A-C)d - B(c^2-d^2)))\,(a+b\tan(e+}{d^2\,(c^2+d^2)^2\,f\sqrt{c+d\tan(e+fx)}} + \frac{2b^2(b(8c^4C - 4Bc^3d + c^2(A+15C)d^2 - 10Bcd^3 + (7A+C)d^4) + 3ad^2(2c(A-C)d - B(c^2-}{3d^3\,(c^2+d^2)^2\,f} - \frac{8\int \frac{-\frac{3}{8}(6ab^2d(4c^4C-Bc^3d-2c^2(A-5C)d^2-7Bcd^3+4Ad^4)-2b^3c(8c^4C-4Bc^3d+c^2(A+15C)d^2-10Bcd^3+(7A+C)d^4)-3a^3d^3(}{}}{}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)(a + b\tan(e + fx))^3}{3d(c^2 + d^2)f(c + d\tan(e + fx))^{3/2}}$$
$$-\frac{2(b(2c^4C - Bc^3d + 4c^2Cd^2 - 3Bcd^3 + 2Ad^4) + ad^2(2c(A - C)d - B(c^2 - d^2)))(a + b\tan(e + f}{d^2(c^2 + d^2)^2 f\sqrt{c + d\tan(e + fx)}}$$
$$+\frac{2b(3abd(8c^4C - 2Bc^3d - c^2(A - 17C)d^2 - 8Bcd^3 + (5A + 3C)d^4) - b^2(16c^5C - 8Bc^4d + 2c^3(A}{3d^4}$$
$$+\frac{2b^2(b(8c^4C - 4Bc^3d + c^2(A + 15C)d^2 - 10Bcd^3 + (7A + C)d^4) + 3ad^2(2c(A - C)d - B(c^2 - d}{3d^3(c^2 + d^2)^2 f}$$
$$-\frac{8\int \frac{9}{8}d^3(a^3(c^2C - 2Bcd - Cd^2 - A(c^2 - d^2)) - 3ab^2(c^2C - 2Bcd - Cd^2 - A(c^2 - d^2)) - 3a^2b(2c(A - C)d - B(c^2 - d^2)) + b^3(2c(A - C)d - B}{}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)(a + b\tan(e + fx))^3}{3d(c^2 + d^2)f(c + d\tan(e + fx))^{3/2}}$$
$$-\frac{2(b(2c^4C - Bc^3d + 4c^2Cd^2 - 3Bcd^3 + 2Ad^4) + ad^2(2c(A - C)d - B(c^2 - d^2)))(a + b\tan(e + f}{d^2(c^2 + d^2)^2 f\sqrt{c + d\tan(e + fx)}}$$
$$+\frac{2b(3abd(8c^4C - 2Bc^3d - c^2(A - 17C)d^2 - 8Bcd^3 + (5A + 3C)d^4) - b^2(16c^5C - 8Bc^4d + 2c^3(A}{3d^4}$$
$$+\frac{2b^2(b(8c^4C - 4Bc^3d + c^2(A + 15C)d^2 - 10Bcd^3 + (7A + C)d^4) + 3ad^2(2c(A - C)d - B(c^2 - d}{3d^3(c^2 + d^2)^2 f}$$
$$+\frac{((a - ib)^3(A - iB - C))\int \frac{1 + i\tan(e + fx)}{\sqrt{c + d\tan(e + fx)}}\,dx}{2(c - id)^2}$$
$$+\frac{((a + ib)^3(A + iB - C))\int \frac{1 - i\tan(e + fx)}{\sqrt{c + d\tan(e + fx)}}\,dx}{2(c + id)^2}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)(a + b\tan(e + fx))^3}{3d(c^2 + d^2)f(c + d\tan(e + fx))^{3/2}}$$
$$-\frac{2(b(2c^4C - Bc^3d + 4c^2Cd^2 - 3Bcd^3 + 2Ad^4) + ad^2(2c(A - C)d - B(c^2 - d^2)))(a + b\tan(e + f}{d^2(c^2 + d^2)^2 f\sqrt{c + d\tan(e + fx)}}$$
$$+\frac{2b(3abd(8c^4C - 2Bc^3d - c^2(A - 17C)d^2 - 8Bcd^3 + (5A + 3C)d^4) - b^2(16c^5C - 8Bc^4d + 2c^3(A}{3d^4}$$
$$+\frac{2b^2(b(8c^4C - 4Bc^3d + c^2(A + 15C)d^2 - 10Bcd^3 + (7A + C)d^4) + 3ad^2(2c(A - C)d - B(c^2 - d}{3d^3(c^2 + d^2)^2 f}$$
$$+\frac{(i(a - ib)^3(A - iB - C))\,\mathrm{Subst}\left(\int \frac{1}{(-1 + x)\sqrt{c - idx}}\,dx, x, i\tan(e + fx)\right)}{2(c - id)^2 f}$$
$$-\frac{(i(a + ib)^3(A + iB - C))\,\mathrm{Subst}\left(\int \frac{1}{(-1 + x)\sqrt{c + idx}}\,dx, x, -i\tan(e + fx)\right)}{2(c + id)^2 f}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)(a + b\tan(e + fx))^3}{3d(c^2 + d^2)f(c + d\tan(e + fx))^{3/2}}$$
$$-\frac{2(b(2c^4C - Bc^3d + 4c^2Cd^2 - 3Bcd^3 + 2Ad^4) + ad^2(2c(A - C)d - B(c^2 - d^2)))(a + b\tan(e +}{d^2(c^2 + d^2)^2 f\sqrt{c + d\tan(e + fx)}}$$
$$+\frac{2b(3abd(8c^4C - 2Bc^3d - c^2(A - 17C)d^2 - 8Bcd^3 + (5A + 3C)d^4) - b^2(16c^5C - 8Bc^4d + 2c^3(}{3d}$$
$$+\frac{2b^2(b(8c^4C - 4Bc^3d + c^2(A + 15C)d^2 - 10Bcd^3 + (7A + C)d^4) + 3ad^2(2c(A - C)d - B(c^2 -}{3d^3(c^2 + d^2)^2 f}$$
$$-\frac{((a - ib)^3(A - iB - C))\,\mathrm{Subst}\left(\int \frac{1}{-1 - \frac{ic}{d} + \frac{ix^2}{d}}\,dx, x, \sqrt{c + d\tan(e + fx)}\right)}{(c - id)^2df}$$
$$-\frac{((a + ib)^3(A + iB - C))\,\mathrm{Subst}\left(\int \frac{1}{-1 + \frac{ic}{d} - \frac{ix^2}{d}}\,dx, x, \sqrt{c + d\tan(e + fx)}\right)}{(c + id)^2df}$$

$$= -\frac{(a - ib)^3(iA + B - iC)\mathrm{arctanh}\left(\frac{\sqrt{c + d\tan(e + fx)}}{\sqrt{c - id}}\right)}{(c - id)^{5/2}f}$$
$$-\frac{(ia - b)^3(A + iB - C)\mathrm{arctanh}\left(\frac{\sqrt{c + d\tan(e + fx)}}{\sqrt{c + id}}\right)}{(c + id)^{5/2}f}$$
$$-\frac{2(c^2C - Bcd + Ad^2)(a + b\tan(e + fx))^3}{3d(c^2 + d^2)f(c + d\tan(e + fx))^{3/2}}$$
$$-\frac{2(b(2c^4C - Bc^3d + 4c^2Cd^2 - 3Bcd^3 + 2Ad^4) + ad^2(2c(A - C)d - B(c^2 - d^2)))(a + b\tan(e +}{d^2(c^2 + d^2)^2 f\sqrt{c + d\tan(e + fx)}}$$
$$+\frac{2b(3abd(8c^4C - 2Bc^3d - c^2(A - 17C)d^2 - 8Bcd^3 + (5A + 3C)d^4) - b^2(16c^5C - 8Bc^4d + 2c^3(}{3d}$$
$$+\frac{2b^2(b(8c^4C - 4Bc^3d + c^2(A + 15C)d^2 - 10Bcd^3 + (7A + C)d^4) + 3ad^2(2c(A - C)d - B(c^2 -}{3d^3(c^2 + d^2)^2 f}$$

Mathematica [C] (verified)

Result contains higher order function than in optimal. Order 5 vs. order 3 in optimal.

Time = 6.93 (sec) , antiderivative size = 670, normalized size of antiderivative = 1.15

$$\int \frac{(a+b\tan(e+fx))^3 \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{5/2}}\,dx = \frac{2C(a+b\tan(e+fx))^3}{3df(c+d\tan(e+fx))^{3/2}}$$

$$+\frac{2\left(-\frac{3(2bcC-bBd-2aCd)(a+b\tan(e+fx))^2}{df(c+d\tan(e+fx))^{3/2}}+\frac{2\left(-\frac{3\left(b(Ab+aB-bC)d^2+4(bc-ad)(2bcC-bBd-2aCd)\right)(a+b\tan(e+fx))}{2df(c+d\tan(e+fx))^{3/2}}-\frac{3\left(-\frac{2\left(-16b^3c^3C+8b^3Bc^2d+\cdots\right.}{\cdots}\cdots\right)}{\cdots}\right)}{\cdots}\right)}{\cdots}$$

[In] Integrate[((a + b*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(c + d*Tan[e + f*x])^(5/2),x]

```
[Out] (2*C*(a + b*Tan[e + f*x])^3)/(3*d*f*(c + d*Tan[e + f*x])^(3/2)) + (2*((-3*(
2*b*c*C - b*B*d - 2*a*C*d)*(a + b*Tan[e + f*x])^2)/(d*f*(c + d*Tan[e + f*x]
)^(3/2)) + (2*((-3*(b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(2*b*c*C - b*B*
d - 2*a*C*d))*(a + b*Tan[e + f*x]))/(2*d*f*(c + d*Tan[e + f*x])^(3/2)) - (3
*((-2*(-16*b^3*c^3*C + 8*b^3*B*c^2*d + 48*a*b^2*c^2*C*d - 2*A*b^3*c*d^2 - 1
8*a*b^2*B*c*d^2 - 48*a^2*b*c*C*d^2 + 2*b^3*c*C*d^2 + 9*a^2*b*B*d^3 + b^3*B*
d^3 + 16*a^3*C*d^3))/(3*d*(c + d*Tan[e + f*x])^(3/2)) + (2*((((3*c*(a^3*B -
 3*a*b^2*B + 3*a^2*b*(A - C) - b^3*(A - C))*d^4)/2 + (3*(3*a^2*b*B - b^3*B
- a^3*(A - C) + 3*a*b^2*(A - C))*d^5)/2)*(-1/3*Hypergeometric2F1[-3/2, 1, -
1/2, (c + d*Tan[e + f*x])/(c - I*d)]/((I*c + d)*(c + d*Tan[e + f*x])^(3/2))
 + Hypergeometric2F1[-3/2, 1, -1/2, (c + d*Tan[e + f*x])/(c + I*d)]/(3*(I*c
 - d)*(c + d*Tan[e + f*x])^(3/2))))/d - (3*(a^3*B - 3*a*b^2*B + 3*a^2*b*(A
- C) - b^3*(A - C))*d^3*(-(Hypergeometric2F1[-1/2, 1, 1/2, (c + d*Tan[e + f
*x])/(c - I*d)]/((I*c + d)*Sqrt[c + d*Tan[e + f*x]])) + Hypergeometric2F1[-
1/2, 1, 1/2, (c + d*Tan[e + f*x])/(c + I*d)]/((I*c - d)*Sqrt[c + d*Tan[e +
f*x]])))/2))/(3*d)))/(4*d*f)))/d))/(3*d)
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 13585 vs. $2(550) = 1100$.

Time = 0.48 (sec) , antiderivative size = 13586, normalized size of antiderivative = 23.22

method	result	size
parts	Expression too large to display	13586
derivativedivides	Expression too large to display	85156
default	Expression too large to display	85156

```
[In] int((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(5/
2),x,method=_RETURNVERBOSE)
```

```
[Out] result too large to display
```

Fricas [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{5/2}}\,dx=\text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^(5/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int \frac{(a+b\tan(e+fx))^3 (A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx = \int \frac{(a+b\tan(e+fx))^3 (A+B\tan(e+$$

```
[In] integrate((a+b*tan(f*x+e))**3*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x
+e))**(5/2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**3*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/(
c + d*tan(e + f*x))**(5/2), x)
```

Maxima [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^3 (A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^(5/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

Giac [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^3 (A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^(5/2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^3 (A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx = \text{Hanged}$$

```
[In] int(((a + b*tan(e + f*x))^3*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d
*tan(e + f*x))^(5/2),x)
```

```
[Out] \text{Hanged}
```

3.123 $\int \frac{(a+b\tan(e+fx))^2(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx$

Optimal result 1739
Rubi [A] (verified) 1740
Mathematica [C] (verified) 1744
Maple [B] (verified) 1744
Fricas [F(-1)] 1745
Sympy [F] 1745
Maxima [F(-1)] 1745
Giac [F(-1)] 1746
Mupad [B] (verification not implemented) 1746

Optimal result

Integrand size = 47, antiderivative size = 358

$$\int \frac{(a+b\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx = -\frac{(a-ib)^2(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(c-id)^{5/2}f} - \frac{(a+ib)^2(B-i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(c+id)^{5/2}f} - \frac{2(c^2C-Bcd+Ad^2)\,(a+b\tan(e+fx))^2}{3d\,(c^2+d^2)\,f(c+d\tan(e+fx))^{3/2}} + \frac{2(bc-ad)\,(b(4c^4C-Bc^3d-2c^2(A-5C)d^2-7Bcd^3+4Ad^4)+3ad^2(2c(A-C)d-B(c^2-d^2)))}{3d^3\,(c^2+d^2)^2\,f\sqrt{c+d\tan(e+fx)}} + \frac{2b^2(4c^2C-Bcd+(A+3C)d^2)\,\sqrt{c+d\tan(e+fx)}}{3d^3\,(c^2+d^2)\,f}$$

```
[Out] -(a-I*b)^2*(I*A+B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(c-I*d
)^(5/2)/f-(a+I*b)^2*(B-I*(A-C))*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2
))/(c+I*d)^(5/2)/f+2/3*(-a*d+b*c)*(b*(4*c^4*C-B*c^3*d-2*c^2*(A-5*C)*d^2-7*B
*c*d^3+4*A*d^4)+3*a*d^2*(2*c*(A-C)*d-B*(c^2-d^2)))/d^3/(c^2+d^2)^2/f/(c+d*t
an(f*x+e))^(1/2)+2/3*b^2*(4*c^2*C-B*c*d+(A+3*C)*d^2)*(c+d*tan(f*x+e))^(1/2)
/d^3/(c^2+d^2)/f-2/3*(A*d^2-B*c*d+C*c^2)*(a+b*tan(f*x+e))^2/d/(c^2+d^2)/f/(
c+d*tan(f*x+e))^(3/2)
```

Rubi [A] (verified)

Time = 1.94 (sec) , antiderivative size = 358, normalized size of antiderivative = 1.00, number of steps used = 10, number of rules used = 7, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.149, Rules used = {3726, 3716, 3711, 3620, 3618, 65, 214}

$$\int \frac{(a+b\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx =$$
$$-\frac{(a-ib)^2(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f(c-id)^{5/2}}$$
$$-\frac{(a+ib)^2(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f(c+id)^{5/2}}$$
$$-\frac{2(Ad^2-Bcd+c^2C)\,(a+b\tan(e+fx))^2}{3df\,(c^2+d^2)\,(c+d\tan(e+fx))^{3/2}}$$
$$+\frac{2(bc-ad)\,(3ad^2(2cd(A-C)-B(c^2-d^2))+b(-2c^2d^2(A-5C)+4Ad^4-Bc^3d-7Bcd^3+4c^4C))}{3d^3f\,(c^2+d^2)^2\,\sqrt{c+d\tan(e+fx)}}$$
$$+\frac{2b^2(d^2(A+3C)-Bcd+4c^2C)\,\sqrt{c+d\tan(e+fx)}}{3d^3f\,(c^2+d^2)}$$

```
[In] Int[((a + b*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c + d
*Tan[e + f*x])^(5/2),x]
```

```
[Out] -(((a - I*b)^2*(I*A + B - I*C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*
d]])/((c - I*d)^(5/2)*f)) - ((a + I*b)^2*(B - I*(A - C))*ArcTanh[Sqrt[c + d
*Tan[e + f*x]]/Sqrt[c + I*d]])/((c + I*d)^(5/2)*f) - (2*(c^2*C - B*c*d + A*
d^2)*(a + b*Tan[e + f*x])^2)/(3*d*(c^2 + d^2)*f*(c + d*Tan[e + f*x])^(3/2))
 + (2*(b*c - a*d)*(b*(4*c^4*C - B*c^3*d - 2*c^2*(A - 5*C)*d^2 - 7*B*c*d^3 +
 4*A*d^4) + 3*a*d^2*(2*c*(A - C)*d - B*(c^2 - d^2))))/(3*d^3*(c^2 + d^2)^2*
f*Sqrt[c + d*Tan[e + f*x]]) + (2*b^2*(4*c^2*C - B*c*d + (A + 3*C)*d^2)*Sqrt
[c + d*Tan[e + f*x]])/(3*d^3*(c^2 + d^2)*f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3716

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.
)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(-(b*c - a*d))*(c^2*C - B*c*d + A*d^2)*((c
+ d*Tan[e + f*x])^(n + 1)/(d^2*f*(n + 1)*(c^2 + d^2))), x] + Dist[1/(d*(c^2
 + d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)*Simp[a*d*(A*c - c*C + B*d) + b*(
c^2*C - B*c*d + A*d^2) + d*(A*b*c + a*B*c - b*c*C - a*A*d + b*B*d + a*C*d)*
Tan[e + f*x] + b*C*(c^2 + d^2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c
, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] && LtQ[n,
 -1]
```

Rule 3726

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*d^2 + c*(c*C - B*d))*(a + b*Tan[e +
 f*x])^m*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
```

```
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rubi steps

$$\text{integral} = -\frac{2(c^2C - Bcd + Ad^2)(a + b\tan(e + fx))^2}{3d(c^2 + d^2) f(c + d\tan(e + fx))^{3/2}} + \frac{2\int \frac{(a+b\tan(e+fx))\left(\frac{1}{2}\left(2Ad\left(\frac{3ac}{2}+2bd\right)+2\left(2bc-\frac{3ad}{2}\right)(cC-Bd)\right)+\frac{3}{2}d((A-C)(bc-ad)+B(ac+bd))\tan(e+fx)+\frac{1}{2}b(4c^2C-Bcd+(A+3C)d^2)}{(c+d\tan(e+fx))^{3/2}}}{3d(c^2+d^2)}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)(a + b\tan(e + fx))^2}{3d(c^2 + d^2) f(c + d\tan(e + fx))^{3/2}} + \frac{2(bc - ad)(b(4c^4C - Bc^3d - 2c^2(A - 5C)d^2 - 7Bcd^3 + 4Ad^4) + 3ad^2(2c(A - C)d - B(c^2 - d^2)}{3d^3(c^2 + d^2)^2 f\sqrt{c + d\tan(e + fx)}} + \frac{2\int \frac{1}{2}(b^2(4c^4C-Bc^3d-2c^2(A-5C)d^2-7Bcd^3+4Ad^4)-3a^2d^2(c^2C-2Bcd-Cd^2-A(c^2-d^2))+6abd^2(2c(A-C)d-B(c^2-d^2)))-\frac{3}{2}}{}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)(a + b\tan(e + fx))^2}{3d(c^2 + d^2) f(c + d\tan(e + fx))^{3/2}} + \frac{2(bc - ad)(b(4c^4C - Bc^3d - 2c^2(A - 5C)d^2 - 7Bcd^3 + 4Ad^4) + 3ad^2(2c(A - C)d - B(c^2 - d^2)}{3d^3(c^2 + d^2)^2 f\sqrt{c + d\tan(e + fx)}} + \frac{2b^2(4c^2C - Bcd + (A + 3C)d^2)\sqrt{c + d\tan(e + fx)}}{3d^3(c^2 + d^2) f} + \frac{2\int \frac{-\frac{3}{2}d^2(a^2(c^2C-2Bcd-Cd^2-A(c^2-d^2))-b^2(c^2C-2Bcd-Cd^2-A(c^2-d^2))-2ab(2c(A-C)d-B(c^2-d^2)))-\frac{3}{2}d^2(2ab(c^2C-2B}{\sqrt{c+d\tan(e+fx)}}}{3d^2(c^2+d^2)^2}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)(a + b\tan(e + fx))^2}{3d(c^2 + d^2) f(c + d\tan(e + fx))^{3/2}} + \frac{2(bc - ad)(b(4c^4C - Bc^3d - 2c^2(A - 5C)d^2 - 7Bcd^3 + 4Ad^4) + 3ad^2(2c(A - C)d - B(c^2 - d^2)}{3d^3(c^2 + d^2)^2 f\sqrt{c + d\tan(e + fx)}} + \frac{2b^2(4c^2C - Bcd + (A + 3C)d^2)\sqrt{c + d\tan(e + fx)}}{3d^3(c^2 + d^2) f} + \frac{((a - ib)^2(A - iB - C))\int \frac{1+i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(c - id)^2} + \frac{((a + ib)^2(A + iB - C))\int \frac{1-i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(c + id)^2}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)(a + b\tan(e + fx))^2}{3d(c^2 + d^2)f(c + d\tan(e + fx))^{3/2}}$$
$$+ \frac{2(bc - ad)(b(4c^4C - Bc^3d - 2c^2(A - 5C)d^2 - 7Bcd^3 + 4Ad^4) + 3ad^2(2c(A - C)d - B(c^2 - d}{3d^3(c^2 + d^2)^2 f\sqrt{c + d\tan(e + fx)}}$$
$$+ \frac{2b^2(4c^2C - Bcd + (A + 3C)d^2)\sqrt{c + d\tan(e + fx)}}{3d^3(c^2 + d^2)f}$$
$$+ \frac{((a - ib)^2(iA + B - iC))\,\mathrm{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c-idx}}\,dx, x, i\tan(e + fx)\right)}{2(c - id)^2 f}$$
$$- \frac{(i(a + ib)^2(A + iB - C))\,\mathrm{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c+idx}}\,dx, x, -i\tan(e + fx)\right)}{2(c + id)^2 f}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)(a + b\tan(e + fx))^2}{3d(c^2 + d^2)f(c + d\tan(e + fx))^{3/2}}$$
$$+ \frac{2(bc - ad)(b(4c^4C - Bc^3d - 2c^2(A - 5C)d^2 - 7Bcd^3 + 4Ad^4) + 3ad^2(2c(A - C)d - B(c^2 - d}{3d^3(c^2 + d^2)^2 f\sqrt{c + d\tan(e + fx)}}$$
$$+ \frac{2b^2(4c^2C - Bcd + (A + 3C)d^2)\sqrt{c + d\tan(e + fx)}}{3d^3(c^2 + d^2)f}$$
$$- \frac{((a - ib)^2(A - iB - C))\,\mathrm{Subst}\left(\int \frac{1}{-1-\frac{ic}{d}+\frac{ix^2}{d}}\,dx, x, \sqrt{c + d\tan(e + fx)}\right)}{(c - id)^2 df}$$
$$- \frac{((a + ib)^2(A + iB - C))\,\mathrm{Subst}\left(\int \frac{1}{-1+\frac{ic}{d}-\frac{ix^2}{d}}\,dx, x, \sqrt{c + d\tan(e + fx)}\right)}{(c + id)^2 df}$$

$$= -\frac{(a - ib)^2(iA + B - iC)\mathrm{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(c - id)^{5/2} f}$$
$$- \frac{(a + ib)^2(B - i(A - C))\mathrm{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(c + id)^{5/2} f}$$
$$- \frac{2(c^2C - Bcd + Ad^2)(a + b\tan(e + fx))^2}{3d(c^2 + d^2)f(c + d\tan(e + fx))^{3/2}}$$
$$+ \frac{2(bc - ad)(b(4c^4C - Bc^3d - 2c^2(A - 5C)d^2 - 7Bcd^3 + 4Ad^4) + 3ad^2(2c(A - C)d - B(c^2 - d}{3d^3(c^2 + d^2)^2 f\sqrt{c + d\tan(e + fx)}}$$
$$+ \frac{2b^2(4c^2C - Bcd + (A + 3C)d^2)\sqrt{c + d\tan(e + fx)}}{3d^3(c^2 + d^2)f}$$

Mathematica [C] (verified)

Result contains higher order function than in optimal. Order 5 vs. order 3 in optimal.

Time = 6.63 (sec) , antiderivative size = 502, normalized size of antiderivative = 1.40

$$\int \frac{(a+b\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx=\frac{2C(a+b\tan(e+fx))^2}{df(c+d\tan(e+fx))^{3/2}}$$

$$+\frac{2\left(-\frac{(-4bcC+bBd+4aCd)(a+b\tan(e+fx))}{df(c+d\tan(e+fx))^{3/2}}-\frac{-\frac{2\left(8b^2c^2C-2b^2Bcd-16abcCd-Ab^2d^2+abBd^2+8a^2Cd^2+b^2Cd^2\right)}{3d(c+d\tan(e+fx))^{3/2}}+\frac{2\left(\frac{\left(\frac{3}{2}c\left(a^2B-b^2B+2ab(A-C)\right)d^3+\cdots\right.}{\cdots}\right.}{\cdots}}{\cdots}\right)}{\cdots}$$

```
[In] Integrate[((a + b*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(c + d*Tan[e + f*x])^(5/2),x]
```

```
[Out] (2*C*(a + b*Tan[e + f*x])^2)/(d*f*(c + d*Tan[e + f*x])^(3/2)) + (2*(-(((-4*
b*c*C + b*B*d + 4*a*C*d)*(a + b*Tan[e + f*x]))/(d*f*(c + d*Tan[e + f*x])^(3
/2))) - ((-2*(8*b^2*c^2*C - 2*b^2*B*c*d - 16*a*b*c*C*d - A*b^2*d^2 + a*b*B*
d^2 + 8*a^2*C*d^2 + b^2*C*d^2))/(3*d*(c + d*Tan[e + f*x])^(3/2)) + (2*((((3
*c*(a^2*B - b^2*B + 2*a*b*(A - C))*d^3)/2 + (3*(2*a*b*B - a^2*(A - C) + b^2
*(A - C))*d^4)/2)*(-1/3*Hypergeometric2F1[-3/2, 1, -1/2, (c + d*Tan[e + f*x
])/(c - I*d)]/((I*c + d)*(c + d*Tan[e + f*x])^(3/2)) + Hypergeometric2F1[-3
/2, 1, -1/2, (c + d*Tan[e + f*x])/(c + I*d)]/(3*(I*c - d)*(c + d*Tan[e + f*
x])^(3/2))))/d - (3*(a^2*B - b^2*B + 2*a*b*(A - C))*d^2*(-(Hypergeometric2F
1[-1/2, 1, 1/2, (c + d*Tan[e + f*x])/(c - I*d)]/((I*c + d)*Sqrt[c + d*Tan[e
 + f*x]])) + Hypergeometric2F1[-1/2, 1, 1/2, (c + d*Tan[e + f*x])/(c + I*d)
]/((I*c - d)*Sqrt[c + d*Tan[e + f*x]])))/2))/(3*d))/(2*d*f)))/d
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 11359 vs. 2(325) = 650.

Time = 0.22 (sec) , antiderivative size = 11360, normalized size of antiderivative = 31.73

method	result	size
parts	Expression too large to display	11360
derivativedivides	Expression too large to display	61833
default	Expression too large to display	61833

```
[In] int((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(5/
2),x,method=_RETURNVERBOSE)
```

```
[Out] result too large to display
```

Fricas [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^2 (A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^(5/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int \frac{(a+b\tan(e+fx))^2 (A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx = \int \frac{(a+b\tan{(e+fx)})^2 (A+B\tan{(e+fx)}+C\tan^2{(e+fx)})}{(c+d\tan{(e+fx)})^{\frac{5}{2}}}\,dx$$

```
[In] integrate((a+b*tan(f*x+e))**2*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x
+e))**(5/2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**2*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/(
c + d*tan(e + f*x))**(5/2), x)
```

Maxima [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^2 (A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^(5/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

Giac [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^(5/2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [B] (verification not implemented)

Time = 109.69 (sec) , antiderivative size = 88684, normalized size of antiderivative = 247.72

$$\int \frac{(a+b\tan(e+fx))^2\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx = \text{Too large to display}$$

```
[In] int(((a + b*tan(e + f*x))^2*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d
*tan(e + f*x))^(5/2),x)
```

```
[Out] atan((((c + d*tan(e + f*x))^(1/2)*(96*A^2*a^2*b^2*d^18*f^3 - 16*A^2*b^4*d^1
8*f^3 - 16*A^2*a^4*d^18*f^3 + 320*A^2*a^4*c^4*d^14*f^3 + 1024*A^2*a^4*c^6*d
^12*f^3 + 1440*A^2*a^4*c^8*d^10*f^3 + 1024*A^2*a^4*c^10*d^8*f^3 + 320*A^2*a
^4*c^12*d^6*f^3 - 16*A^2*a^4*c^16*d^2*f^3 + 320*A^2*b^4*c^4*d^14*f^3 + 1024
*A^2*b^4*c^6*d^12*f^3 + 1440*A^2*b^4*c^8*d^10*f^3 + 1024*A^2*b^4*c^10*d^8*f
^3 + 320*A^2*b^4*c^12*d^6*f^3 - 16*A^2*b^4*c^16*d^2*f^3 - 256*A^2*a*b^3*c*d
^17*f^3 + 256*A^2*a^3*b*c*d^17*f^3 - 1280*A^2*a*b^3*c^3*d^15*f^3 - 2304*A^2
*a*b^3*c^5*d^13*f^3 - 1280*A^2*a*b^3*c^7*d^11*f^3 + 1280*A^2*a*b^3*c^9*d^9*
f^3 + 2304*A^2*a*b^3*c^11*d^7*f^3 + 1280*A^2*a*b^3*c^13*d^5*f^3 + 256*A^2*a
*b^3*c^15*d^3*f^3 + 1280*A^2*a^3*b*c^3*d^15*f^3 + 2304*A^2*a^3*b*c^5*d^13*f
^3 + 1280*A^2*a^3*b*c^7*d^11*f^3 - 1280*A^2*a^3*b*c^9*d^9*f^3 - 2304*A^2*a^
3*b*c^11*d^7*f^3 - 1280*A^2*a^3*b*c^13*d^5*f^3 - 256*A^2*a^3*b*c^15*d^3*f^3
 - 1920*A^2*a^2*b^2*c^4*d^14*f^3 - 6144*A^2*a^2*b^2*c^6*d^12*f^3 - 8640*A^2
*a^2*b^2*c^8*d^10*f^3 - 6144*A^2*a^2*b^2*c^10*d^8*f^3 - 1920*A^2*a^2*b^2*c^
12*d^6*f^3 + 96*A^2*a^2*b^2*c^16*d^2*f^3) + ((((8*A^2*a^4*c^5*f^2 + 8*A^2*b
^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^4*c
^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*d^4
*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4*d
*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^3*
b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4*A
^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80
*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) -
 4*A^2*a^4*c^5*f^2 - 4*A^2*b^4*c^5*f^2 + 24*A^2*a^2*b^2*c^5*f^2 + 40*A^2*a^
```

4*c^3*d^2*f^2 + 40*A^2*b^4*c^3*d^2*f^2 + 16*A^2*a*b^3*d^5*f^2 - 16*A^2*a^3*
b*d^5*f^2 - 20*A^2*a^4*c*d^4*f^2 - 20*A^2*b^4*c*d^4*f^2 + 80*A^2*a*b^3*c^4*
d*f^2 - 80*A^2*a^3*b*c^4*d*f^2 - 160*A^2*a*b^3*c^2*d^3*f^2 + 120*A^2*a^2*b^
2*c*d^4*f^2 + 160*A^2*a^3*b*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*
(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*
c^8*d^2*f^4)))^(1/2)*(32*A*b^2*d^21*f^4 - 32*A*a^2*d^21*f^4 - (c + d*tan(e
+ f*x))^(1/2)*((((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^
5*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*
f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^4*f^2 -
160*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f
^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^
2*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4
*A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4
 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*A^2*a^4*c^5*f^2 - 4*A^2*b^4
*c^5*f^2 + 24*A^2*a^2*b^2*c^5*f^2 + 40*A^2*a^4*c^3*d^2*f^2 + 40*A^2*b^4*c^3
*d^2*f^2 + 16*A^2*a*b^3*d^5*f^2 - 16*A^2*a^3*b*d^5*f^2 - 20*A^2*a^4*c*d^4*f
^2 - 20*A^2*b^4*c*d^4*f^2 + 80*A^2*a*b^3*c^4*d*f^2 - 80*A^2*a^3*b*c^4*d*f^2
 - 160*A^2*a*b^3*c^2*d^3*f^2 + 120*A^2*a^2*b^2*c*d^4*f^2 + 160*A^2*a^3*b*c^
2*d^3*f^2 - 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d
^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^2
2*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^
9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5
+ 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 160*A*a^2*c^2*d
^19*f^4 - 128*A*a^2*c^4*d^17*f^4 + 896*A*a^2*c^6*d^15*f^4 + 3136*A*a^2*c^8*
d^13*f^4 + 4928*A*a^2*c^10*d^11*f^4 + 4480*A*a^2*c^12*d^9*f^4 + 2432*A*a^2*
c^14*d^7*f^4 + 736*A*a^2*c^16*d^5*f^4 + 96*A*a^2*c^18*d^3*f^4 + 160*A*b^2*c
^2*d^19*f^4 + 128*A*b^2*c^4*d^17*f^4 - 896*A*b^2*c^6*d^15*f^4 - 3136*A*b^2*
c^8*d^13*f^4 - 4928*A*b^2*c^10*d^11*f^4 - 4480*A*b^2*c^12*d^9*f^4 - 2432*A*
b^2*c^14*d^7*f^4 - 736*A*b^2*c^16*d^5*f^4 - 96*A*b^2*c^18*d^3*f^4 + 192*A*a
*b*c*d^20*f^4 + 1472*A*a*b*c^3*d^18*f^4 + 4864*A*a*b*c^5*d^16*f^4 + 8960*A*
a*b*c^7*d^14*f^4 + 9856*A*a*b*c^9*d^12*f^4 + 6272*A*a*b*c^11*d^10*f^4 + 179
2*A*a*b*c^13*d^8*f^4 - 256*A*a*b*c^15*d^6*f^4 - 320*A*a*b*c^17*d^4*f^4 - 64
*A*a*b*c^19*d^2*f^4))*((((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4*c^5*f^2 - 48*A^2*a^
2*b^2*c^5*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*
b^3*d^5*f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^
4*f^2 - 160*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2 + 320*A^2*a*b^3*c
^2*d^3*f^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^2*d^3*f^2 + 480*A^
2*a^2*b^2*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4
*b^4 + 4*A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4
*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*A^2*a^4*c^5*f^2 - 4
*A^2*b^4*c^5*f^2 + 24*A^2*a^2*b^2*c^5*f^2 + 40*A^2*a^4*c^3*d^2*f^2 + 40*A^2
*b^4*c^3*d^2*f^2 + 16*A^2*a*b^3*d^5*f^2 - 16*A^2*a^3*b*d^5*f^2 - 20*A^2*a^4
*c*d^4*f^2 - 20*A^2*b^4*c*d^4*f^2 + 80*A^2*a*b^3*c^4*d*f^2 - 80*A^2*a^3*b*c
^4*d*f^2 - 160*A^2*a*b^3*c^2*d^3*f^2 + 120*A^2*a^2*b^2*c*d^4*f^2 + 160*A^2*
a^3*b*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 +

5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*1
i + ((c + d*tan(e + f*x))^(1/2)*(96*A^2*a^2*b^2*d^18*f^3 - 16*A^2*b^4*d^18*
f^3 - 16*A^2*a^4*d^18*f^3 + 320*A^2*a^4*c^4*d^14*f^3 + 1024*A^2*a^4*c^6*d^1
2*f^3 + 1440*A^2*a^4*c^8*d^10*f^3 + 1024*A^2*a^4*c^10*d^8*f^3 + 320*A^2*a^4
*c^12*d^6*f^3 - 16*A^2*a^4*c^16*d^2*f^3 + 320*A^2*b^4*c^4*d^14*f^3 + 1024*A
^2*b^4*c^6*d^12*f^3 + 1440*A^2*b^4*c^8*d^10*f^3 + 1024*A^2*b^4*c^10*d^8*f^3
 + 320*A^2*b^4*c^12*d^6*f^3 - 16*A^2*b^4*c^16*d^2*f^3 - 256*A^2*a*b^3*c*d^1
7*f^3 + 256*A^2*a^3*b*c*d^17*f^3 - 1280*A^2*a*b^3*c^3*d^15*f^3 - 2304*A^2*a
*b^3*c^5*d^13*f^3 - 1280*A^2*a*b^3*c^7*d^11*f^3 + 1280*A^2*a*b^3*c^9*d^9*f^
3 + 2304*A^2*a*b^3*c^11*d^7*f^3 + 1280*A^2*a*b^3*c^13*d^5*f^3 + 256*A^2*a*b
^3*c^15*d^3*f^3 + 1280*A^2*a^3*b*c^3*d^15*f^3 + 2304*A^2*a^3*b*c^5*d^13*f^3
 + 1280*A^2*a^3*b*c^7*d^11*f^3 - 1280*A^2*a^3*b*c^9*d^9*f^3 - 2304*A^2*a^3*
b*c^11*d^7*f^3 - 1280*A^2*a^3*b*c^13*d^5*f^3 - 256*A^2*a^3*b*c^15*d^3*f^3 -
 1920*A^2*a^2*b^2*c^4*d^14*f^3 - 6144*A^2*a^2*b^2*c^6*d^12*f^3 - 8640*A^2*a
^2*b^2*c^8*d^10*f^3 - 6144*A^2*a^2*b^2*c^10*d^8*f^3 - 1920*A^2*a^2*b^2*c^12
*d^6*f^3 + 96*A^2*a^2*b^2*c^16*d^2*f^3) - ((((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4
*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3
*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f
^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f
^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*
c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4
*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c
^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4
*A^2*a^4*c^5*f^2 - 4*A^2*b^4*c^5*f^2 + 24*A^2*a^2*b^2*c^5*f^2 + 40*A^2*a^4*
c^3*d^2*f^2 + 40*A^2*b^4*c^3*d^2*f^2 + 16*A^2*a*b^3*d^5*f^2 - 16*A^2*a^3*b*
d^5*f^2 - 20*A^2*a^4*c*d^4*f^2 - 20*A^2*b^4*c*d^4*f^2 + 80*A^2*a*b^3*c^4*d*
f^2 - 80*A^2*a^3*b*c^4*d*f^2 - 160*A^2*a*b^3*c^2*d^3*f^2 + 120*A^2*a^2*b^2*
c*d^4*f^2 + 160*A^2*a^3*b*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c
^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^
8*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^4*c^5*f^2 + 8*A^
2*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^
4*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*
d^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^
4*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a
^3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*b^8 +
4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 +
 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2
) - 4*A^2*a^4*c^5*f^2 - 4*A^2*b^4*c^5*f^2 + 24*A^2*a^2*b^2*c^5*f^2 + 40*A^2
*a^4*c^3*d^2*f^2 + 40*A^2*b^4*c^3*d^2*f^2 + 16*A^2*a*b^3*d^5*f^2 - 16*A^2*a
^3*b*d^5*f^2 - 20*A^2*a^4*c*d^4*f^2 - 20*A^2*b^4*c*d^4*f^2 + 80*A^2*a*b^3*c
^4*d*f^2 - 80*A^2*a^3*b*c^4*d*f^2 - 160*A^2*a*b^3*c^2*d^3*f^2 + 120*A^2*a^2
*b^2*c*d^4*f^2 + 160*A^2*a^3*b*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c^3*d^2*f^2)/(
16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 +
 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f
^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c

^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 6
4*c^21*d^2*f^5) - 32*A*a^2*d^21*f^4 + 32*A*b^2*d^21*f^4 - 160*A*a^2*c^2*d^1
9*f^4 - 128*A*a^2*c^4*d^17*f^4 + 896*A*a^2*c^6*d^15*f^4 + 3136*A*a^2*c^8*d^
13*f^4 + 4928*A*a^2*c^10*d^11*f^4 + 4480*A*a^2*c^12*d^9*f^4 + 2432*A*a^2*c^
14*d^7*f^4 + 736*A*a^2*c^16*d^5*f^4 + 96*A*a^2*c^18*d^3*f^4 + 160*A*b^2*c^2
*d^19*f^4 + 128*A*b^2*c^4*d^17*f^4 - 896*A*b^2*c^6*d^15*f^4 - 3136*A*b^2*c^
8*d^13*f^4 - 4928*A*b^2*c^10*d^11*f^4 - 4480*A*b^2*c^12*d^9*f^4 - 2432*A*b^
2*c^14*d^7*f^4 - 736*A*b^2*c^16*d^5*f^4 - 96*A*b^2*c^18*d^3*f^4 + 192*A*a*b
*c*d^20*f^4 + 1472*A*a*b*c^3*d^18*f^4 + 4864*A*a*b*c^5*d^16*f^4 + 8960*A*a*
b*c^7*d^14*f^4 + 9856*A*a*b*c^9*d^12*f^4 + 6272*A*a*b*c^11*d^10*f^4 + 1792*
A*a*b*c^13*d^8*f^4 - 256*A*a*b*c^15*d^6*f^4 - 320*A*a*b*c^17*d^4*f^4 - 64*A
*a*b*c^19*d^2*f^4))*((((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4*c^5*f^2 - 48*A^2*a^2*
b^2*c^5*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*b^
3*d^5*f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^4*
f^2 - 160*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2
*d^3*f^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^2*d^3*f^2 + 480*A^2*
a^2*b^2*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b
^4 + 4*A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d
^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*A^2*a^4*c^5*f^2 - 4*A
^2*b^4*c^5*f^2 + 24*A^2*a^2*b^2*c^5*f^2 + 40*A^2*a^4*c^3*d^2*f^2 + 40*A^2*b
^4*c^3*d^2*f^2 + 16*A^2*a*b^3*d^5*f^2 - 16*A^2*a^3*b*d^5*f^2 - 20*A^2*a^4*c
*d^4*f^2 - 20*A^2*b^4*c*d^4*f^2 + 80*A^2*a*b^3*c^4*d*f^2 - 80*A^2*a^3*b*c^4
*d*f^2 - 160*A^2*a*b^3*c^2*d^3*f^2 + 120*A^2*a^2*b^2*c*d^4*f^2 + 160*A^2*a^
3*b*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5
*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*1i)
/(((c + d*tan(e + f*x))^(1/2)*(96*A^2*a^2*b^2*d^18*f^3 - 16*A^2*b^4*d^18*f^
3 - 16*A^2*a^4*d^18*f^3 + 320*A^2*a^4*c^4*d^14*f^3 + 1024*A^2*a^4*c^6*d^12*
f^3 + 1440*A^2*a^4*c^8*d^10*f^3 + 1024*A^2*a^4*c^10*d^8*f^3 + 320*A^2*a^4*c
^12*d^6*f^3 - 16*A^2*a^4*c^16*d^2*f^3 + 320*A^2*b^4*c^4*d^14*f^3 + 1024*A^2
*b^4*c^6*d^12*f^3 + 1440*A^2*b^4*c^8*d^10*f^3 + 1024*A^2*b^4*c^10*d^8*f^3 +
320*A^2*b^4*c^12*d^6*f^3 - 16*A^2*b^4*c^16*d^2*f^3 - 256*A^2*a*b^3*c*d^17*
f^3 + 256*A^2*a^3*b*c*d^17*f^3 - 1280*A^2*a*b^3*c^3*d^15*f^3 - 2304*A^2*a*b
^3*c^5*d^13*f^3 - 1280*A^2*a*b^3*c^7*d^11*f^3 + 1280*A^2*a*b^3*c^9*d^9*f^3
+ 2304*A^2*a*b^3*c^11*d^7*f^3 + 1280*A^2*a*b^3*c^13*d^5*f^3 + 256*A^2*a*b^3
*c^15*d^3*f^3 + 1280*A^2*a^3*b*c^3*d^15*f^3 + 2304*A^2*a^3*b*c^5*d^13*f^3 +
1280*A^2*a^3*b*c^7*d^11*f^3 - 1280*A^2*a^3*b*c^9*d^9*f^3 - 2304*A^2*a^3*b*
c^11*d^7*f^3 - 1280*A^2*a^3*b*c^13*d^5*f^3 - 256*A^2*a^3*b*c^15*d^3*f^3 - 1
920*A^2*a^2*b^2*c^4*d^14*f^3 - 6144*A^2*a^2*b^2*c^6*d^12*f^3 - 8640*A^2*a^2
*b^2*c^8*d^10*f^3 - 6144*A^2*a^2*b^2*c^10*d^8*f^3 - 1920*A^2*a^2*b^2*c^12*d
^6*f^3 + 96*A^2*a^2*b^2*c^16*d^2*f^3) + ((((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4*c
^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3*d
^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f^2
+ 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2
+ 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^
2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a

^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2
*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*A
^2*a^4*c^5*f^2 - 4*A^2*b^4*c^5*f^2 + 24*A^2*a^2*b^2*c^5*f^2 + 40*A^2*a^4*c^
3*d^2*f^2 + 40*A^2*b^4*c^3*d^2*f^2 + 16*A^2*a*b^3*d^5*f^2 - 16*A^2*a^3*b*d^
5*f^2 - 20*A^2*a^4*c*d^4*f^2 - 20*A^2*b^4*c*d^4*f^2 + 80*A^2*a*b^3*c^4*d*f^
2 - 80*A^2*a^3*b*c^4*d*f^2 - 160*A^2*a*b^3*c^2*d^3*f^2 + 120*A^2*a^2*b^2*c*
d^4*f^2 + 160*A^2*a^3*b*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^1
0*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*
d^2*f^4)))^(1/2)*(32*A*b^2*d^21*f^4 - 32*A*a^2*d^21*f^4 - (c + d*tan(e + f*
x))^(1/2)*((((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^
2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2
+ 32*A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160*
A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 -
240*A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^
3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4
*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 1
60*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*A^2*a^4*c^5*f^2 - 4*A^2*b^4*c^5
*f^2 + 24*A^2*a^2*b^2*c^5*f^2 + 40*A^2*a^4*c^3*d^2*f^2 + 40*A^2*b^4*c^3*d^2
*f^2 + 16*A^2*a*b^3*d^5*f^2 - 16*A^2*a^3*b*d^5*f^2 - 20*A^2*a^4*c*d^4*f^2 -
20*A^2*b^4*c*d^4*f^2 + 80*A^2*a*b^3*c^4*d*f^2 - 80*A^2*a^3*b*c^4*d*f^2 - 1
60*A^2*a*b^3*c^2*d^3*f^2 + 120*A^2*a^2*b^2*c*d^4*f^2 + 160*A^2*a^3*b*c^2*d^
3*f^2 - 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f
^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^
5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^
14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 28
80*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 160*A*a^2*c^2*d^19*
f^4 - 128*A*a^2*c^4*d^17*f^4 + 896*A*a^2*c^6*d^15*f^4 + 3136*A*a^2*c^8*d^13
*f^4 + 4928*A*a^2*c^10*d^11*f^4 + 4480*A*a^2*c^12*d^9*f^4 + 2432*A*a^2*c^14
*d^7*f^4 + 736*A*a^2*c^16*d^5*f^4 + 96*A*a^2*c^18*d^3*f^4 + 160*A*b^2*c^2*d
^19*f^4 + 128*A*b^2*c^4*d^17*f^4 - 896*A*b^2*c^6*d^15*f^4 - 3136*A*b^2*c^8*
d^13*f^4 - 4928*A*b^2*c^10*d^11*f^4 - 4480*A*b^2*c^12*d^9*f^4 - 2432*A*b^2*
c^14*d^7*f^4 - 736*A*b^2*c^16*d^5*f^4 - 96*A*b^2*c^18*d^3*f^4 + 192*A*a*b*c
*d^20*f^4 + 1472*A*a*b*c^3*d^18*f^4 + 4864*A*a*b*c^5*d^16*f^4 + 8960*A*a*b*
c^7*d^14*f^4 + 9856*A*a*b*c^9*d^12*f^4 + 6272*A*a*b*c^11*d^10*f^4 + 1792*A*
a*b*c^13*d^8*f^4 - 256*A*a*b*c^15*d^6*f^4 - 320*A*a*b*c^17*d^4*f^4 - 64*A*a
*b*c^19*d^2*f^4))*((((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4*c^5*f^2 - 48*A^2*a^2*b^
2*c^5*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*b^3*
d^5*f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^4*f^
2 - 160*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2*d
^3*f^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^2*d^3*f^2 + 480*A^2*a^
2*b^2*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4
+ 4*A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6
*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*A^2*a^4*c^5*f^2 - 4*A^2
*b^4*c^5*f^2 + 24*A^2*a^2*b^2*c^5*f^2 + 40*A^2*a^4*c^3*d^2*f^2 + 40*A^2*b^4
*c^3*d^2*f^2 + 16*A^2*a*b^3*d^5*f^2 - 16*A^2*a^3*b*d^5*f^2 - 20*A^2*a^4*c*d

^4*f^2 - 20*A^2*b^4*c*d^4*f^2 + 80*A^2*a*b^3*c^4*d*f^2 - 80*A^2*a^3*b*c^4*d
*f^2 - 160*A^2*a*b^3*c^2*d^3*f^2 + 120*A^2*a^2*b^2*c*d^4*f^2 + 160*A^2*a^3*
b*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c
^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2) - ((c
 + d*tan(e + f*x))^(1/2)*(96*A^2*a^2*b^2*d^18*f^3 - 16*A^2*b^4*d^18*f^3 - 1
6*A^2*a^4*d^18*f^3 + 320*A^2*a^4*c^4*d^14*f^3 + 1024*A^2*a^4*c^6*d^12*f^3 +
 1440*A^2*a^4*c^8*d^10*f^3 + 1024*A^2*a^4*c^10*d^8*f^3 + 320*A^2*a^4*c^12*d
^6*f^3 - 16*A^2*a^4*c^16*d^2*f^3 + 320*A^2*b^4*c^4*d^14*f^3 + 1024*A^2*b^4*
c^6*d^12*f^3 + 1440*A^2*b^4*c^8*d^10*f^3 + 1024*A^2*b^4*c^10*d^8*f^3 + 320*
A^2*b^4*c^12*d^6*f^3 - 16*A^2*b^4*c^16*d^2*f^3 - 256*A^2*a*b^3*c*d^17*f^3 +
 256*A^2*a^3*b*c*d^17*f^3 - 1280*A^2*a*b^3*c^3*d^15*f^3 - 2304*A^2*a*b^3*c^
5*d^13*f^3 - 1280*A^2*a*b^3*c^7*d^11*f^3 + 1280*A^2*a*b^3*c^9*d^9*f^3 + 230
4*A^2*a*b^3*c^11*d^7*f^3 + 1280*A^2*a*b^3*c^13*d^5*f^3 + 256*A^2*a*b^3*c^15
*d^3*f^3 + 1280*A^2*a^3*b*c^3*d^15*f^3 + 2304*A^2*a^3*b*c^5*d^13*f^3 + 1280
*A^2*a^3*b*c^7*d^11*f^3 - 1280*A^2*a^3*b*c^9*d^9*f^3 - 2304*A^2*a^3*b*c^11*
d^7*f^3 - 1280*A^2*a^3*b*c^13*d^5*f^3 - 256*A^2*a^3*b*c^15*d^3*f^3 - 1920*A
^2*a^2*b^2*c^4*d^14*f^3 - 6144*A^2*a^2*b^2*c^6*d^12*f^3 - 8640*A^2*a^2*b^2*
c^8*d^10*f^3 - 6144*A^2*a^2*b^2*c^10*d^8*f^3 - 1920*A^2*a^2*b^2*c^12*d^6*f^
3 + 96*A^2*a^2*b^2*c^16*d^2*f^3) - ((((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4*c^5*f^
2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3*d^2*f^
2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f^2 + 40
*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2 + 32
0*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^2*d^3
*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^
6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*
f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*A^2*a^
4*c^5*f^2 - 4*A^2*b^4*c^5*f^2 + 24*A^2*a^2*b^2*c^5*f^2 + 40*A^2*a^4*c^3*d^2
*f^2 + 40*A^2*b^4*c^3*d^2*f^2 + 16*A^2*a*b^3*d^5*f^2 - 16*A^2*a^3*b*d^5*f^2
 - 20*A^2*a^4*c*d^4*f^2 - 20*A^2*b^4*c*d^4*f^2 + 80*A^2*a*b^3*c^4*d*f^2 - 8
0*A^2*a^3*b*c^4*d*f^2 - 160*A^2*a*b^3*c^2*d^3*f^2 + 120*A^2*a^2*b^2*c*d^4*f
^2 + 160*A^2*a^3*b*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4
 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f
^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4*c
^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3*d
^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f^2
 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2
 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^
2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a
^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2
*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*A
^2*a^4*c^5*f^2 - 4*A^2*b^4*c^5*f^2 + 24*A^2*a^2*b^2*c^5*f^2 + 40*A^2*a^4*c^
3*d^2*f^2 + 40*A^2*b^4*c^3*d^2*f^2 + 16*A^2*a*b^3*d^5*f^2 - 16*A^2*a^3*b*d^
5*f^2 - 20*A^2*a^4*c*d^4*f^2 - 20*A^2*b^4*c*d^4*f^2 + 80*A^2*a*b^3*c^4*d*f^
2 - 80*A^2*a^3*b*c^4*d*f^2 - 160*A^2*a*b^3*c^2*d^3*f^2 + 120*A^2*a^2*b^2*c*
d^4*f^2 + 160*A^2*a^3*b*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^1

0*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*
d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 76
80*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^1
0*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*
d^2*f^5) - 32*A*a^2*d^21*f^4 + 32*A*b^2*d^21*f^4 - 160*A*a^2*c^2*d^19*f^4 -
128*A*a^2*c^4*d^17*f^4 + 896*A*a^2*c^6*d^15*f^4 + 3136*A*a^2*c^8*d^13*f^4
+ 4928*A*a^2*c^10*d^11*f^4 + 4480*A*a^2*c^12*d^9*f^4 + 2432*A*a^2*c^14*d^7*
f^4 + 736*A*a^2*c^16*d^5*f^4 + 96*A*a^2*c^18*d^3*f^4 + 160*A*b^2*c^2*d^19*f
^4 + 128*A*b^2*c^4*d^17*f^4 - 896*A*b^2*c^6*d^15*f^4 - 3136*A*b^2*c^8*d^13*
f^4 - 4928*A*b^2*c^10*d^11*f^4 - 4480*A*b^2*c^12*d^9*f^4 - 2432*A*b^2*c^14*
d^7*f^4 - 736*A*b^2*c^16*d^5*f^4 - 96*A*b^2*c^18*d^3*f^4 + 192*A*a*b*c*d^20
*f^4 + 1472*A*a*b*c^3*d^18*f^4 + 4864*A*a*b*c^5*d^16*f^4 + 8960*A*a*b*c^7*d
^14*f^4 + 9856*A*a*b*c^9*d^12*f^4 + 6272*A*a*b*c^11*d^10*f^4 + 1792*A*a*b*c
^13*d^8*f^4 - 256*A*a*b*c^15*d^6*f^4 - 320*A*a*b*c^17*d^4*f^4 - 64*A*a*b*c^
19*d^2*f^4))*((((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5
*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f
^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 1
60*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^
2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^2
*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*
A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4
+ 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*A^2*a^4*c^5*f^2 - 4*A^2*b^4*
c^5*f^2 + 24*A^2*a^2*b^2*c^5*f^2 + 40*A^2*a^4*c^3*d^2*f^2 + 40*A^2*b^4*c^3*
d^2*f^2 + 16*A^2*a*b^3*d^5*f^2 - 16*A^2*a^3*b*d^5*f^2 - 20*A^2*a^4*c*d^4*f^
2 - 20*A^2*b^4*c*d^4*f^2 + 80*A^2*a*b^3*c^4*d*f^2 - 80*A^2*a^3*b*c^4*d*f^2
- 160*A^2*a*b^3*c^2*d^3*f^2 + 120*A^2*a^2*b^2*c*d^4*f^2 + 160*A^2*a^3*b*c^2
*d^3*f^2 - 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^
8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2) - 64*A^3*a
^3*b^3*d^16*f^2 - 192*A^3*a^6*c^3*d^13*f^2 - 480*A^3*a^6*c^5*d^11*f^2 - 640
*A^3*a^6*c^7*d^9*f^2 - 480*A^3*a^6*c^9*d^7*f^2 - 192*A^3*a^6*c^11*d^5*f^2 -
32*A^3*a^6*c^13*d^3*f^2 + 192*A^3*b^6*c^3*d^13*f^2 + 480*A^3*b^6*c^5*d^11*
f^2 + 640*A^3*b^6*c^7*d^9*f^2 + 480*A^3*b^6*c^9*d^7*f^2 + 192*A^3*b^6*c^11*
d^5*f^2 + 32*A^3*b^6*c^13*d^3*f^2 - 32*A^3*a*b^5*d^16*f^2 - 32*A^3*a^5*b*d^
16*f^2 - 32*A^3*a^6*c*d^15*f^2 + 32*A^3*b^6*c*d^15*f^2 - 160*A^3*a*b^5*c^2*
d^14*f^2 - 288*A^3*a*b^5*c^4*d^12*f^2 - 160*A^3*a*b^5*c^6*d^10*f^2 + 160*A^
3*a*b^5*c^8*d^8*f^2 + 288*A^3*a*b^5*c^10*d^6*f^2 + 160*A^3*a*b^5*c^12*d^4*f
^2 + 32*A^3*a*b^5*c^14*d^2*f^2 + 32*A^3*a^2*b^4*c*d^15*f^2 - 32*A^3*a^4*b^2
*c*d^15*f^2 - 160*A^3*a^5*b*c^2*d^14*f^2 - 288*A^3*a^5*b*c^4*d^12*f^2 - 160
*A^3*a^5*b*c^6*d^10*f^2 + 160*A^3*a^5*b*c^8*d^8*f^2 + 288*A^3*a^5*b*c^10*d^
6*f^2 + 160*A^3*a^5*b*c^12*d^4*f^2 + 32*A^3*a^5*b*c^14*d^2*f^2 + 192*A^3*a^
2*b^4*c^3*d^13*f^2 + 480*A^3*a^2*b^4*c^5*d^11*f^2 + 640*A^3*a^2*b^4*c^7*d^9
*f^2 + 480*A^3*a^2*b^4*c^9*d^7*f^2 + 192*A^3*a^2*b^4*c^11*d^5*f^2 + 32*A^3*
a^2*b^4*c^13*d^3*f^2 - 320*A^3*a^3*b^3*c^2*d^14*f^2 - 576*A^3*a^3*b^3*c^4*d
^12*f^2 - 320*A^3*a^3*b^3*c^6*d^10*f^2 + 320*A^3*a^3*b^3*c^8*d^8*f^2 + 576*
A^3*a^3*b^3*c^10*d^6*f^2 + 320*A^3*a^3*b^3*c^12*d^4*f^2 + 64*A^3*a^3*b^3*c^

14*d^2*f^2 - 192*A^3*a^4*b^2*c^3*d^13*f^2 - 480*A^3*a^4*b^2*c^5*d^11*f^2 -
640*A^3*a^4*b^2*c^7*d^9*f^2 - 480*A^3*a^4*b^2*c^9*d^7*f^2 - 192*A^3*a^4*b^2
*c^11*d^5*f^2 - 32*A^3*a^4*b^2*c^13*d^3*f^2))*((((8*A^2*a^4*c^5*f^2 + 8*A^2
*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^4
*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*d
^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4
*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^
3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4
*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 +
80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2)
 - 4*A^2*a^4*c^5*f^2 - 4*A^2*b^4*c^5*f^2 + 24*A^2*a^2*b^2*c^5*f^2 + 40*A^2*
a^4*c^3*d^2*f^2 + 40*A^2*b^4*c^3*d^2*f^2 + 16*A^2*a*b^3*d^5*f^2 - 16*A^2*a^
3*b*d^5*f^2 - 20*A^2*a^4*c*d^4*f^2 - 20*A^2*b^4*c*d^4*f^2 + 80*A^2*a*b^3*c^
4*d*f^2 - 80*A^2*a^3*b*c^4*d*f^2 - 160*A^2*a*b^3*c^2*d^3*f^2 + 120*A^2*a^2*
b^2*c*d^4*f^2 + 160*A^2*a^3*b*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c^3*d^2*f^2)/(1
6*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 +
5*c^8*d^2*f^4)))^(1/2)*2i + atan((((c + d*tan(e + f*x))^(1/2)*(96*A^2*a^2*b
^2*d^18*f^3 - 16*A^2*b^4*d^18*f^3 - 16*A^2*a^4*d^18*f^3 + 320*A^2*a^4*c^4*d
^14*f^3 + 1024*A^2*a^4*c^6*d^12*f^3 + 1440*A^2*a^4*c^8*d^10*f^3 + 1024*A^2*
a^4*c^10*d^8*f^3 + 320*A^2*a^4*c^12*d^6*f^3 - 16*A^2*a^4*c^16*d^2*f^3 + 320
*A^2*b^4*c^4*d^14*f^3 + 1024*A^2*b^4*c^6*d^12*f^3 + 1440*A^2*b^4*c^8*d^10*f
^3 + 1024*A^2*b^4*c^10*d^8*f^3 + 320*A^2*b^4*c^12*d^6*f^3 - 16*A^2*b^4*c^16
*d^2*f^3 - 256*A^2*a*b^3*c*d^17*f^3 + 256*A^2*a^3*b*c*d^17*f^3 - 1280*A^2*a
*b^3*c^3*d^15*f^3 - 2304*A^2*a*b^3*c^5*d^13*f^3 - 1280*A^2*a*b^3*c^7*d^11*f
^3 + 1280*A^2*a*b^3*c^9*d^9*f^3 + 2304*A^2*a*b^3*c^11*d^7*f^3 + 1280*A^2*a*
b^3*c^13*d^5*f^3 + 256*A^2*a*b^3*c^15*d^3*f^3 + 1280*A^2*a^3*b*c^3*d^15*f^3
 + 2304*A^2*a^3*b*c^5*d^13*f^3 + 1280*A^2*a^3*b*c^7*d^11*f^3 - 1280*A^2*a^3
*b*c^9*d^9*f^3 - 2304*A^2*a^3*b*c^11*d^7*f^3 - 1280*A^2*a^3*b*c^13*d^5*f^3
- 256*A^2*a^3*b*c^15*d^3*f^3 - 1920*A^2*a^2*b^2*c^4*d^14*f^3 - 6144*A^2*a^2
*b^2*c^6*d^12*f^3 - 8640*A^2*a^2*b^2*c^8*d^10*f^3 - 6144*A^2*a^2*b^2*c^10*d
^8*f^3 - 1920*A^2*a^2*b^2*c^12*d^6*f^3 + 96*A^2*a^2*b^2*c^16*d^2*f^3) + (-(
((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2*a
^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^3
*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*c^
4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^2
*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)^2
/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*(1
6*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f
^4 + 80*c^8*d^2*f^4))^(1/2) + 4*A^2*a^4*c^5*f^2 + 4*A^2*b^4*c^5*f^2 - 24*A^
2*a^2*b^2*c^5*f^2 - 40*A^2*a^4*c^3*d^2*f^2 - 40*A^2*b^4*c^3*d^2*f^2 - 16*A^
2*a*b^3*d^5*f^2 + 16*A^2*a^3*b*d^5*f^2 + 20*A^2*a^4*c*d^4*f^2 + 20*A^2*b^4*
c*d^4*f^2 - 80*A^2*a*b^3*c^4*d*f^2 + 80*A^2*a^3*b*c^4*d*f^2 + 160*A^2*a*b^3
*c^2*d^3*f^2 - 120*A^2*a^2*b^2*c*d^4*f^2 - 160*A^2*a^3*b*c^2*d^3*f^2 + 240*
A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*
d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(32*A*b^2*d^21*f^4 - 32*A

*a^2*d^21*f^4 - (c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^4*c^5*f^2 + 8*A^2*b
^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^4*c
^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*d^4
*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4*d
*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^3*
b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4*A
^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80
*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) +
 4*A^2*a^4*c^5*f^2 + 4*A^2*b^4*c^5*f^2 - 24*A^2*a^2*b^2*c^5*f^2 - 40*A^2*a^
4*c^3*d^2*f^2 - 40*A^2*b^4*c^3*d^2*f^2 - 16*A^2*a*b^3*d^5*f^2 + 16*A^2*a^3*
b*d^5*f^2 + 20*A^2*a^4*c*d^4*f^2 + 20*A^2*b^4*c*d^4*f^2 - 80*A^2*a*b^3*c^4*
d*f^2 + 80*A^2*a^3*b*c^4*d*f^2 + 160*A^2*a*b^3*c^2*d^3*f^2 - 120*A^2*a^2*b^
2*c*d^4*f^2 - 160*A^2*a^3*b*c^2*d^3*f^2 + 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*
(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*
c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5
+ 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13
*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c
^21*d^2*f^5) - 160*A*a^2*c^2*d^19*f^4 - 128*A*a^2*c^4*d^17*f^4 + 896*A*a^2*
c^6*d^15*f^4 + 3136*A*a^2*c^8*d^13*f^4 + 4928*A*a^2*c^10*d^11*f^4 + 4480*A*
a^2*c^12*d^9*f^4 + 2432*A*a^2*c^14*d^7*f^4 + 736*A*a^2*c^16*d^5*f^4 + 96*A*
a^2*c^18*d^3*f^4 + 160*A*b^2*c^2*d^19*f^4 + 128*A*b^2*c^4*d^17*f^4 - 896*A*
b^2*c^6*d^15*f^4 - 3136*A*b^2*c^8*d^13*f^4 - 4928*A*b^2*c^10*d^11*f^4 - 448
0*A*b^2*c^12*d^9*f^4 - 2432*A*b^2*c^14*d^7*f^4 - 736*A*b^2*c^16*d^5*f^4 - 9
6*A*b^2*c^18*d^3*f^4 + 192*A*a*b*c*d^20*f^4 + 1472*A*a*b*c^3*d^18*f^4 + 486
4*A*a*b*c^5*d^16*f^4 + 8960*A*a*b*c^7*d^14*f^4 + 9856*A*a*b*c^9*d^12*f^4 +
6272*A*a*b*c^11*d^10*f^4 + 1792*A*a*b*c^13*d^8*f^4 - 256*A*a*b*c^15*d^6*f^4
 - 320*A*a*b*c^17*d^4*f^4 - 64*A*a*b*c^19*d^2*f^4))*(-(((8*A^2*a^4*c^5*f^2
+ 8*A^2*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*
A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*
a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^
3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320
*A^2*a^3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*
b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10
*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)
)^(1/2) + 4*A^2*a^4*c^5*f^2 + 4*A^2*b^4*c^5*f^2 - 24*A^2*a^2*b^2*c^5*f^2 -
40*A^2*a^4*c^3*d^2*f^2 - 40*A^2*b^4*c^3*d^2*f^2 - 16*A^2*a*b^3*d^5*f^2 + 16
*A^2*a^3*b*d^5*f^2 + 20*A^2*a^4*c*d^4*f^2 + 20*A^2*b^4*c*d^4*f^2 - 80*A^2*a
*b^3*c^4*d*f^2 + 80*A^2*a^3*b*c^4*d*f^2 + 160*A^2*a*b^3*c^2*d^3*f^2 - 120*A
^2*a^2*b^2*c*d^4*f^2 - 160*A^2*a^3*b*c^2*d^3*f^2 + 240*A^2*a^2*b^2*c^3*d^2*
f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4
*f^4 + 5*c^8*d^2*f^4)))^(1/2)*1i + ((c + d*tan(e + f*x))^(1/2)*(96*A^2*a^2*
b^2*d^18*f^3 - 16*A^2*b^4*d^18*f^3 - 16*A^2*a^4*d^18*f^3 + 320*A^2*a^4*c^4*
d^14*f^3 + 1024*A^2*a^4*c^6*d^12*f^3 + 1440*A^2*a^4*c^8*d^10*f^3 + 1024*A^2
*a^4*c^10*d^8*f^3 + 320*A^2*a^4*c^12*d^6*f^3 - 16*A^2*a^4*c^16*d^2*f^3 + 32
0*A^2*b^4*c^4*d^14*f^3 + 1024*A^2*b^4*c^6*d^12*f^3 + 1440*A^2*b^4*c^8*d^10*

f^3 + 1024*A^2*b^4*c^10*d^8*f^3 + 320*A^2*b^4*c^12*d^6*f^3 - 16*A^2*b^4*c^1
6*d^2*f^3 - 256*A^2*a*b^3*c*d^17*f^3 + 256*A^2*a^3*b*c*d^17*f^3 - 1280*A^2*
a*b^3*c^3*d^15*f^3 - 2304*A^2*a*b^3*c^5*d^13*f^3 - 1280*A^2*a*b^3*c^7*d^11*
f^3 + 1280*A^2*a*b^3*c^9*d^9*f^3 + 2304*A^2*a*b^3*c^11*d^7*f^3 + 1280*A^2*a
*b^3*c^13*d^5*f^3 + 256*A^2*a*b^3*c^15*d^3*f^3 + 1280*A^2*a^3*b*c^3*d^15*f^
3 + 2304*A^2*a^3*b*c^5*d^13*f^3 + 1280*A^2*a^3*b*c^7*d^11*f^3 - 1280*A^2*a^
3*b*c^9*d^9*f^3 - 2304*A^2*a^3*b*c^11*d^7*f^3 - 1280*A^2*a^3*b*c^13*d^5*f^3
 - 256*A^2*a^3*b*c^15*d^3*f^3 - 1920*A^2*a^2*b^2*c^4*d^14*f^3 - 6144*A^2*a^
2*b^2*c^6*d^12*f^3 - 8640*A^2*a^2*b^2*c^8*d^10*f^3 - 6144*A^2*a^2*b^2*c^10*
d^8*f^3 - 1920*A^2*a^2*b^2*c^12*d^6*f^3 + 96*A^2*a^2*b^2*c^16*d^2*f^3) - (-
(((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2*
a^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^
3*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*c
^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^
2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)^
2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*(
16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*
f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*A^2*a^4*c^5*f^2 + 4*A^2*b^4*c^5*f^2 - 24*A
^2*a^2*b^2*c^5*f^2 - 40*A^2*a^4*c^3*d^2*f^2 - 40*A^2*b^4*c^3*d^2*f^2 - 16*A
^2*a*b^3*d^5*f^2 + 16*A^2*a^3*b*d^5*f^2 + 20*A^2*a^4*c*d^4*f^2 + 20*A^2*b^4
*c*d^4*f^2 - 80*A^2*a*b^3*c^4*d*f^2 + 80*A^2*a^3*b*c^4*d*f^2 + 160*A^2*a*b^
3*c^2*d^3*f^2 - 120*A^2*a^2*b^2*c*d^4*f^2 - 160*A^2*a^3*b*c^2*d^3*f^2 + 240
*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4
*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1
/2)*(-(((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 8
0*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*
A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a
*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*
A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2
*f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*
b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^
6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*A^2*a^4*c^5*f^2 + 4*A^2*b^4*c^5*f^2
 - 24*A^2*a^2*b^2*c^5*f^2 - 40*A^2*a^4*c^3*d^2*f^2 - 40*A^2*b^4*c^3*d^2*f^2
 - 16*A^2*a*b^3*d^5*f^2 + 16*A^2*a^3*b*d^5*f^2 + 20*A^2*a^4*c*d^4*f^2 + 20*A
^2*b^4*c*d^4*f^2 - 80*A^2*a*b^3*c^4*d*f^2 + 80*A^2*a^3*b*c^4*d*f^2 + 160*A^
2*a*b^3*c^2*d^3*f^2 - 120*A^2*a^2*b^2*c*d^4*f^2 - 160*A^2*a^3*b*c^2*d^3*f^2
 + 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 +
10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 6
40*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^
5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^
17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 32*A*a^2*d^21*f^4 + 32*A
*b^2*d^21*f^4 - 160*A*a^2*c^2*d^19*f^4 - 128*A*a^2*c^4*d^17*f^4 + 896*A*a^2
*c^6*d^15*f^4 + 3136*A*a^2*c^8*d^13*f^4 + 4928*A*a^2*c^10*d^11*f^4 + 4480*A
*a^2*c^12*d^9*f^4 + 2432*A*a^2*c^14*d^7*f^4 + 736*A*a^2*c^16*d^5*f^4 + 96*A
*a^2*c^18*d^3*f^4 + 160*A*b^2*c^2*d^19*f^4 + 128*A*b^2*c^4*d^17*f^4 - 896*A

*b^2*c^6*d^15*f^4 - 3136*A*b^2*c^8*d^13*f^4 - 4928*A*b^2*c^10*d^11*f^4 - 44
80*A*b^2*c^12*d^9*f^4 - 2432*A*b^2*c^14*d^7*f^4 - 736*A*b^2*c^16*d^5*f^4 -
96*A*b^2*c^18*d^3*f^4 + 192*A*a*b*c*d^20*f^4 + 1472*A*a*b*c^3*d^18*f^4 + 48
64*A*a*b*c^5*d^16*f^4 + 8960*A*a*b*c^7*d^14*f^4 + 9856*A*a*b*c^9*d^12*f^4 +
6272*A*a*b*c^11*d^10*f^4 + 1792*A*a*b*c^13*d^8*f^4 - 256*A*a*b*c^15*d^6*f^
4 - 320*A*a*b*c^17*d^4*f^4 - 64*A*a*b*c^19*d^2*f^4))*(-(((8*A^2*a^4*c^5*f^2
+ 8*A^2*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80
*A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2
*a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a
^3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 32
0*A^2*a^3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4
*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^1
0*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4
))^(1/2) + 4*A^2*a^4*c^5*f^2 + 4*A^2*b^4*c^5*f^2 - 24*A^2*a^2*b^2*c^5*f^2 -
40*A^2*a^4*c^3*d^2*f^2 - 40*A^2*b^4*c^3*d^2*f^2 - 16*A^2*a*b^3*d^5*f^2 + 1
6*A^2*a^3*b*d^5*f^2 + 20*A^2*a^4*c*d^4*f^2 + 20*A^2*b^4*c*d^4*f^2 - 80*A^2*
a*b^3*c^4*d*f^2 + 80*A^2*a^3*b*c^4*d*f^2 + 160*A^2*a*b^3*c^2*d^3*f^2 - 120*
A^2*a^2*b^2*c*d^4*f^2 - 160*A^2*a^3*b*c^2*d^3*f^2 + 240*A^2*a^2*b^2*c^3*d^2
f^2)/(16(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^
4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*1i)/(((c + d*tan(e + f*x))^(1/2)*(96*A^2*a^2
*b^2*d^18*f^3 - 16*A^2*b^4*d^18*f^3 - 16*A^2*a^4*d^18*f^3 + 320*A^2*a^4*c^4
*d^14*f^3 + 1024*A^2*a^4*c^6*d^12*f^3 + 1440*A^2*a^4*c^8*d^10*f^3 + 1024*A^
2*a^4*c^10*d^8*f^3 + 320*A^2*a^4*c^12*d^6*f^3 - 16*A^2*a^4*c^16*d^2*f^3 + 3
20*A^2*b^4*c^4*d^14*f^3 + 1024*A^2*b^4*c^6*d^12*f^3 + 1440*A^2*b^4*c^8*d^10
*f^3 + 1024*A^2*b^4*c^10*d^8*f^3 + 320*A^2*b^4*c^12*d^6*f^3 - 16*A^2*b^4*c^
16*d^2*f^3 - 256*A^2*a*b^3*c*d^17*f^3 + 256*A^2*a^3*b*c*d^17*f^3 - 1280*A^2
*a*b^3*c^3*d^15*f^3 - 2304*A^2*a*b^3*c^5*d^13*f^3 - 1280*A^2*a*b^3*c^7*d^11
*f^3 + 1280*A^2*a*b^3*c^9*d^9*f^3 + 2304*A^2*a*b^3*c^11*d^7*f^3 + 1280*A^2*
a*b^3*c^13*d^5*f^3 + 256*A^2*a*b^3*c^15*d^3*f^3 + 1280*A^2*a^3*b*c^3*d^15*f
^3 + 2304*A^2*a^3*b*c^5*d^13*f^3 + 1280*A^2*a^3*b*c^7*d^11*f^3 - 1280*A^2*a
^3*b*c^9*d^9*f^3 - 2304*A^2*a^3*b*c^11*d^7*f^3 - 1280*A^2*a^3*b*c^13*d^5*f^
3 - 256*A^2*a^3*b*c^15*d^3*f^3 - 1920*A^2*a^2*b^2*c^4*d^14*f^3 - 6144*A^2*a
^2*b^2*c^6*d^12*f^3 - 8640*A^2*a^2*b^2*c^8*d^10*f^3 - 6144*A^2*a^2*b^2*c^10
*d^8*f^3 - 1920*A^2*a^2*b^2*c^12*d^6*f^3 + 96*A^2*a^2*b^2*c^16*d^2*f^3) + (
-(((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2
*a^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a
^3*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*
c^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a
^2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)
^2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*
(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4
*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*A^2*a^4*c^5*f^2 + 4*A^2*b^4*c^5*f^2 - 24*
A^2*a^2*b^2*c^5*f^2 - 40*A^2*a^4*c^3*d^2*f^2 - 40*A^2*b^4*c^3*d^2*f^2 - 16*
A^2*a*b^3*d^5*f^2 + 16*A^2*a^3*b*d^5*f^2 + 20*A^2*a^4*c*d^4*f^2 + 20*A^2*b^
4*c*d^4*f^2 - 80*A^2*a*b^3*c^4*d*f^2 + 80*A^2*a^3*b*c^4*d*f^2 + 160*A^2*a*b

^3*c^2*d^3*f^2 - 120*A^2*a^2*b^2*c*d^4*f^2 - 160*A^2*a^3*b*c^2*d^3*f^2 + 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(32*A*b^2*d^21*f^4 - 32*A*a^2*d^21*f^4 - (c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*A^2*a^4*c^5*f^2 + 4*A^2*b^4*c^5*f^2 - 24*A^2*a^2*b^2*c^5*f^2 - 40*A^2*a^4*c^3*d^2*f^2 - 40*A^2*b^4*c^3*d^2*f^2 - 16*A^2*a*b^3*d^5*f^2 + 16*A^2*a^3*b*d^5*f^2 + 20*A^2*a^4*c*d^4*f^2 + 20*A^2*b^4*c*d^4*f^2 - 80*A^2*a*b^3*c^4*d*f^2 + 80*A^2*a^3*b*c^4*d*f^2 + 160*A^2*a*b^3*c^2*d^3*f^2 - 120*A^2*a^2*b^2*c*d^4*f^2 - 160*A^2*a^3*b*c^2*d^3*f^2 + 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 160*A*a^2*c^2*d^19*f^4 - 128*A*a^2*c^4*d^17*f^4 + 896*A*a^2*c^6*d^15*f^4 + 3136*A*a^2*c^8*d^13*f^4 + 4928*A*a^2*c^10*d^11*f^4 + 4480*A*a^2*c^12*d^9*f^4 + 2432*A*a^2*c^14*d^7*f^4 + 736*A*a^2*c^16*d^5*f^4 + 96*A*a^2*c^18*d^3*f^4 + 160*A*b^2*c^2*d^19*f^4 + 128*A*b^2*c^4*d^17*f^4 - 896*A*b^2*c^6*d^15*f^4 - 3136*A*b^2*c^8*d^13*f^4 - 4928*A*b^2*c^10*d^11*f^4 - 4480*A*b^2*c^12*d^9*f^4 - 2432*A*b^2*c^14*d^7*f^4 - 736*A*b^2*c^16*d^5*f^4 - 96*A*b^2*c^18*d^3*f^4 + 192*A*a*b*c*d^20*f^4 + 1472*A*a*b*c^3*d^18*f^4 + 4864*A*a*b*c^5*d^16*f^4 + 8960*A*a*b*c^7*d^14*f^4 + 9856*A*a*b*c^9*d^12*f^4 + 6272*A*a*b*c^11*d^10*f^4 + 1792*A*a*b*c^13*d^8*f^4 - 256*A*a*b*c^15*d^6*f^4 - 320*A*a*b*c^17*d^4*f^4 - 64*A*a*b*c^19*d^2*f^4))*(-(((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*A^2*a^4*c^5*f^2 + 4*A^2*b^4*c^5*f^2 - 24*A^2*a^2*b^2*c^5*f^2 - 40*A^2*a^4*c^3*d^2*f^2 - 40*A^2*b^4*c^3*d^2*f^2 - 16*A^2*a*b^3*d^5*f^2 + 16*A^2*a^3*b*d^5*f^2 + 20*A^2*a^4*c*d^4*f^2 + 20*A^2*b^4*c*d^4*f^2 - 80*A^2*a*b^3*c^4*d*f^2 + 80*A^2*a^3*b*c^4*d*f^2 + 160*A^2*a*b^3*c^2*d^3*f^2 - 120*A^2*a^2*b^2*c*d^4*f^2 - 160*A^2*a^3*b*c^2*d^3*f^2 + 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2) - ((c + d*tan(e + f*x))^(1/2)*(96*A^2*a^2*b^2*d^18*f^3 - 16*A^2*b^4*d^18*f^3 - 16*A^2*a^4*d^18*f^3 + 320*A^2*a^4*c^4*d

```
^14*f^3 + 1024*A^2*a^4*c^6*d^12*f^3 + 1440*A^2*a^4*c^8*d^10*f^3 + 1024*A^2*
a^4*c^10*d^8*f^3 + 320*A^2*a^4*c^12*d^6*f^3 - 16*A^2*a^4*c^16*d^2*f^3 + 320
*A^2*b^4*c^4*d^14*f^3 + 1024*A^2*b^4*c^6*d^12*f^3 + 1440*A^2*b^4*c^8*d^10*f
^3 + 1024*A^2*b^4*c^10*d^8*f^3 + 320*A^2*b^4*c^12*d^6*f^3 - 16*A^2*b^4*c^16
*d^2*f^3 - 256*A^2*a*b^3*c*d^17*f^3 + 256*A^2*a^3*b*c*d^17*f^3 - 1280*A^2*a
*b^3*c^3*d^15*f^3 - 2304*A^2*a*b^3*c^5*d^13*f^3 - 1280*A^2*a*b^3*c^7*d^11*f
^3 + 1280*A^2*a*b^3*c^9*d^9*f^3 + 2304*A^2*a*b^3*c^11*d^7*f^3 + 1280*A^2*a*
b^3*c^13*d^5*f^3 + 256*A^2*a*b^3*c^15*d^3*f^3 + 1280*A^2*a^3*b*c^3*d^15*f^3
 + 2304*A^2*a^3*b*c^5*d^13*f^3 + 1280*A^2*a^3*b*c^7*d^11*f^3 - 1280*A^2*a^3
*b*c^9*d^9*f^3 - 2304*A^2*a^3*b*c^11*d^7*f^3 - 1280*A^2*a^3*b*c^13*d^5*f^3
- 256*A^2*a^3*b*c^15*d^3*f^3 - 1920*A^2*a^2*b^2*c^4*d^14*f^3 - 6144*A^2*a^2
*b^2*c^6*d^12*f^3 - 8640*A^2*a^2*b^2*c^8*d^10*f^3 - 6144*A^2*a^2*b^2*c^10*d
^8*f^3 - 1920*A^2*a^2*b^2*c^12*d^6*f^3 + 96*A^2*a^2*b^2*c^16*d^2*f^3) - (-(
((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2*a
^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^3
*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*c^
4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^2
*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)^2
/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*(1
6*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f
^4 + 80*c^8*d^2*f^4))^(1/2) + 4*A^2*a^4*c^5*f^2 + 4*A^2*b^4*c^5*f^2 - 24*A^
2*a^2*b^2*c^5*f^2 - 40*A^2*a^4*c^3*d^2*f^2 - 40*A^2*b^4*c^3*d^2*f^2 - 16*A^
2*a*b^3*d^5*f^2 + 16*A^2*a^3*b*d^5*f^2 + 20*A^2*a^4*c*d^4*f^2 + 20*A^2*b^4*
c*d^4*f^2 - 80*A^2*a*b^3*c^4*d*f^2 + 80*A^2*a^3*b*c^4*d*f^2 + 160*A^2*a*b^3
*c^2*d^3*f^2 - 120*A^2*a^2*b^2*c*d^4*f^2 - 160*A^2*a^3*b*c^2*d^3*f^2 + 240*
A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*
d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/
2)*(-(((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80
*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A
^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*
b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A
^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*
f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b
^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6
*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*A^2*a^4*c^5*f^2 + 4*A^2*b^4*c^5*f^2 -
 24*A^2*a^2*b^2*c^5*f^2 - 40*A^2*a^4*c^3*d^2*f^2 - 40*A^2*b^4*c^3*d^2*f^2 -
 16*A^2*a*b^3*d^5*f^2 + 16*A^2*a^3*b*d^5*f^2 + 20*A^2*a^4*c*d^4*f^2 + 20*A^
2*b^4*c*d^4*f^2 - 80*A^2*a*b^3*c^4*d*f^2 + 80*A^2*a^3*b*c^4*d*f^2 + 160*A^2
*a*b^3*c^2*d^3*f^2 - 120*A^2*a^2*b^2*c*d^4*f^2 - 160*A^2*a^3*b*c^2*d^3*f^2
+ 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 1
0*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 64
0*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5
 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^1
7*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 32*A*a^2*d^21*f^4 + 32*A*
b^2*d^21*f^4 - 160*A*a^2*c^2*d^19*f^4 - 128*A*a^2*c^4*d^17*f^4 + 896*A*a^2*
```

c^6*d^15*f^4 + 3136*A*a^2*c^8*d^13*f^4 + 4928*A*a^2*c^10*d^11*f^4 + 4480*A*
a^2*c^12*d^9*f^4 + 2432*A*a^2*c^14*d^7*f^4 + 736*A*a^2*c^16*d^5*f^4 + 96*A*
a^2*c^18*d^3*f^4 + 160*A*b^2*c^2*d^19*f^4 + 128*A*b^2*c^4*d^17*f^4 - 896*A*
b^2*c^6*d^15*f^4 - 3136*A*b^2*c^8*d^13*f^4 - 4928*A*b^2*c^10*d^11*f^4 - 448
0*A*b^2*c^12*d^9*f^4 - 2432*A*b^2*c^14*d^7*f^4 - 736*A*b^2*c^16*d^5*f^4 - 9
6*A*b^2*c^18*d^3*f^4 + 192*A*a*b*c*d^20*f^4 + 1472*A*a*b*c^3*d^18*f^4 + 486
4*A*a*b*c^5*d^16*f^4 + 8960*A*a*b*c^7*d^14*f^4 + 9856*A*a*b*c^9*d^12*f^4 +
6272*A*a*b*c^11*d^10*f^4 + 1792*A*a*b*c^13*d^8*f^4 - 256*A*a*b*c^15*d^6*f^4
 - 320*A*a*b*c^17*d^4*f^4 - 64*A*a*b*c^19*d^2*f^4))*(-(((8*A^2*a^4*c^5*f^2
+ 8*A^2*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*
A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*
a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^
3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320
*A^2*a^3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*
b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10
*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)
)^(1/2) + 4*A^2*a^4*c^5*f^2 + 4*A^2*b^4*c^5*f^2 - 24*A^2*a^2*b^2*c^5*f^2 -
40*A^2*a^4*c^3*d^2*f^2 - 40*A^2*b^4*c^3*d^2*f^2 - 16*A^2*a*b^3*d^5*f^2 + 16
*A^2*a^3*b*d^5*f^2 + 20*A^2*a^4*c*d^4*f^2 + 20*A^2*b^4*c*d^4*f^2 - 80*A^2*a
*b^3*c^4*d*f^2 + 80*A^2*a^3*b*c^4*d*f^2 + 160*A^2*a*b^3*c^2*d^3*f^2 - 120*A
^2*a^2*b^2*c*d^4*f^2 - 160*A^2*a^3*b*c^2*d^3*f^2 + 240*A^2*a^2*b^2*c^3*d^2*
f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4
*f^4 + 5*c^8*d^2*f^4)))^(1/2) - 64*A^3*a^3*b^3*d^16*f^2 - 192*A^3*a^6*c^3*d
^13*f^2 - 480*A^3*a^6*c^5*d^11*f^2 - 640*A^3*a^6*c^7*d^9*f^2 - 480*A^3*a^6*
c^9*d^7*f^2 - 192*A^3*a^6*c^11*d^5*f^2 - 32*A^3*a^6*c^13*d^3*f^2 + 192*A^3*
b^6*c^3*d^13*f^2 + 480*A^3*b^6*c^5*d^11*f^2 + 640*A^3*b^6*c^7*d^9*f^2 + 480
*A^3*b^6*c^9*d^7*f^2 + 192*A^3*b^6*c^11*d^5*f^2 + 32*A^3*b^6*c^13*d^3*f^2 -
 32*A^3*a*b^5*d^16*f^2 - 32*A^3*a^5*b*d^16*f^2 - 32*A^3*a^6*c*d^15*f^2 + 32
*A^3*b^6*c*d^15*f^2 - 160*A^3*a*b^5*c^2*d^14*f^2 - 288*A^3*a*b^5*c^4*d^12*f
^2 - 160*A^3*a*b^5*c^6*d^10*f^2 + 160*A^3*a*b^5*c^8*d^8*f^2 + 288*A^3*a*b^5
*c^10*d^6*f^2 + 160*A^3*a*b^5*c^12*d^4*f^2 + 32*A^3*a*b^5*c^14*d^2*f^2 + 32
*A^3*a^2*b^4*c*d^15*f^2 - 32*A^3*a^4*b^2*c*d^15*f^2 - 160*A^3*a^5*b*c^2*d^1
4*f^2 - 288*A^3*a^5*b*c^4*d^12*f^2 - 160*A^3*a^5*b*c^6*d^10*f^2 + 160*A^3*a
^5*b*c^8*d^8*f^2 + 288*A^3*a^5*b*c^10*d^6*f^2 + 160*A^3*a^5*b*c^12*d^4*f^2
+ 32*A^3*a^5*b*c^14*d^2*f^2 + 192*A^3*a^2*b^4*c^3*d^13*f^2 + 480*A^3*a^2*b^
4*c^5*d^11*f^2 + 640*A^3*a^2*b^4*c^7*d^9*f^2 + 480*A^3*a^2*b^4*c^9*d^7*f^2
+ 192*A^3*a^2*b^4*c^11*d^5*f^2 + 32*A^3*a^2*b^4*c^13*d^3*f^2 - 320*A^3*a^3*
b^3*c^2*d^14*f^2 - 576*A^3*a^3*b^3*c^4*d^12*f^2 - 320*A^3*a^3*b^3*c^6*d^10*
f^2 + 320*A^3*a^3*b^3*c^8*d^8*f^2 + 576*A^3*a^3*b^3*c^10*d^6*f^2 + 320*A^3*
a^3*b^3*c^12*d^4*f^2 + 64*A^3*a^3*b^3*c^14*d^2*f^2 - 192*A^3*a^4*b^2*c^3*d^
13*f^2 - 480*A^3*a^4*b^2*c^5*d^11*f^2 - 640*A^3*a^4*b^2*c^7*d^9*f^2 - 480*A
^3*a^4*b^2*c^9*d^7*f^2 - 192*A^3*a^4*b^2*c^11*d^5*f^2 - 32*A^3*a^4*b^2*c^13
*d^3*f^2))*(-(((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*
f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^
2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 16

0*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2
 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*
c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A
^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 +
 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*A^2*a^4*c^5*f^2 + 4*A^2*b^4*c
^5*f^2 - 24*A^2*a^2*b^2*c^5*f^2 - 40*A^2*a^4*c^3*d^2*f^2 - 40*A^2*b^4*c^3*d
^2*f^2 - 16*A^2*a*b^3*d^5*f^2 + 16*A^2*a^3*b*d^5*f^2 + 20*A^2*a^4*c*d^4*f^2
 + 20*A^2*b^4*c*d^4*f^2 - 80*A^2*a*b^3*c^4*d*f^2 + 80*A^2*a^3*b*c^4*d*f^2 +
 160*A^2*a*b^3*c^2*d^3*f^2 - 120*A^2*a^2*b^2*c*d^4*f^2 - 160*A^2*a^3*b*c^2*
d^3*f^2 + 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8
*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*2i - atan(-
((((((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*C
^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^2
*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b^
3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^2
*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^
2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2
)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d
^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*C^2*a^4*c^5*f^2 - 4*C^2*b^4*c^5*f^2 + 2
4*C^2*a^2*b^2*c^5*f^2 + 40*C^2*a^4*c^3*d^2*f^2 + 40*C^2*b^4*c^3*d^2*f^2 + 1
6*C^2*a*b^3*d^5*f^2 - 16*C^2*a^3*b*d^5*f^2 - 20*C^2*a^4*c*d^4*f^2 - 20*C^2*
b^4*c*d^4*f^2 + 80*C^2*a*b^3*c^4*d*f^2 - 80*C^2*a^3*b*c^4*d*f^2 - 160*C^2*a
*b^3*c^2*d^3*f^2 + 120*C^2*a^2*b^2*c*d^4*f^2 + 160*C^2*a^3*b*c^2*d^3*f^2 -
240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*
c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))
^(1/2)*((((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2 -
 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 + 3
2*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 - 160*C^2
*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 24
0*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*d
^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^
6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*
c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*C^2*a^4*c^5*f^2 - 4*C^2*b^4*c^5*f^
2 + 24*C^2*a^2*b^2*c^5*f^2 + 40*C^2*a^4*c^3*d^2*f^2 + 40*C^2*b^4*c^3*d^2*f^
2 + 16*C^2*a*b^3*d^5*f^2 - 16*C^2*a^3*b*d^5*f^2 - 20*C^2*a^4*c*d^4*f^2 - 20
*C^2*b^4*c*d^4*f^2 + 80*C^2*a*b^3*c^4*d*f^2 - 80*C^2*a^3*b*c^4*d*f^2 - 160*
C^2*a*b^3*c^2*d^3*f^2 + 120*C^2*a^2*b^2*c*d^4*f^2 + 160*C^2*a^3*b*c^2*d^3*f
^2 - 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4
+ 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 +
 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*
f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*
c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 32*C*a^2*d^21*f^4 + 32
*C*b^2*d^21*f^4 - 160*C*a^2*c^2*d^19*f^4 - 128*C*a^2*c^4*d^17*f^4 + 896*C*a
^2*c^6*d^15*f^4 + 3136*C*a^2*c^8*d^13*f^4 + 4928*C*a^2*c^10*d^11*f^4 + 4480
*C*a^2*c^12*d^9*f^4 + 2432*C*a^2*c^14*d^7*f^4 + 736*C*a^2*c^16*d^5*f^4 + 96

*C*a^2*c^18*d^3*f^4 + 160*C*b^2*c^2*d^19*f^4 + 128*C*b^2*c^4*d^17*f^4 - 896
*C*b^2*c^6*d^15*f^4 - 3136*C*b^2*c^8*d^13*f^4 - 4928*C*b^2*c^10*d^11*f^4 -
4480*C*b^2*c^12*d^9*f^4 - 2432*C*b^2*c^14*d^7*f^4 - 736*C*b^2*c^16*d^5*f^4
- 96*C*b^2*c^18*d^3*f^4 + 192*C*a*b*c*d^20*f^4 + 1472*C*a*b*c^3*d^18*f^4 +
4864*C*a*b*c^5*d^16*f^4 + 8960*C*a*b*c^7*d^14*f^4 + 9856*C*a*b*c^9*d^12*f^4
 + 6272*C*a*b*c^11*d^10*f^4 + 1792*C*a*b*c^13*d^8*f^4 - 256*C*a*b*c^15*d^6*
f^4 - 320*C*a*b*c^17*d^4*f^4 - 64*C*a*b*c^19*d^2*f^4) - (c + d*tan(e + f*x)
)^(1/2)*(96*C^2*a^2*b^2*d^18*f^3 - 16*C^2*b^4*d^18*f^3 - 16*C^2*a^4*d^18*f^
3 + 320*C^2*a^4*c^4*d^14*f^3 + 1024*C^2*a^4*c^6*d^12*f^3 + 1440*C^2*a^4*c^8
*d^10*f^3 + 1024*C^2*a^4*c^10*d^8*f^3 + 320*C^2*a^4*c^12*d^6*f^3 - 16*C^2*a
^4*c^16*d^2*f^3 + 320*C^2*b^4*c^4*d^14*f^3 + 1024*C^2*b^4*c^6*d^12*f^3 + 14
40*C^2*b^4*c^8*d^10*f^3 + 1024*C^2*b^4*c^10*d^8*f^3 + 320*C^2*b^4*c^12*d^6*
f^3 - 16*C^2*b^4*c^16*d^2*f^3 - 256*C^2*a*b^3*c*d^17*f^3 + 256*C^2*a^3*b*c*
d^17*f^3 - 1280*C^2*a*b^3*c^3*d^15*f^3 - 2304*C^2*a*b^3*c^5*d^13*f^3 - 1280
*C^2*a*b^3*c^7*d^11*f^3 + 1280*C^2*a*b^3*c^9*d^9*f^3 + 2304*C^2*a*b^3*c^11*
d^7*f^3 + 1280*C^2*a*b^3*c^13*d^5*f^3 + 256*C^2*a*b^3*c^15*d^3*f^3 + 1280*C
^2*a^3*b*c^3*d^15*f^3 + 2304*C^2*a^3*b*c^5*d^13*f^3 + 1280*C^2*a^3*b*c^7*d^
11*f^3 - 1280*C^2*a^3*b*c^9*d^9*f^3 - 2304*C^2*a^3*b*c^11*d^7*f^3 - 1280*C^
2*a^3*b*c^13*d^5*f^3 - 256*C^2*a^3*b*c^15*d^3*f^3 - 1920*C^2*a^2*b^2*c^4*d^
14*f^3 - 6144*C^2*a^2*b^2*c^6*d^12*f^3 - 8640*C^2*a^2*b^2*c^8*d^10*f^3 - 61
44*C^2*a^2*b^2*c^10*d^8*f^3 - 1920*C^2*a^2*b^2*c^12*d^6*f^3 + 96*C^2*a^2*b^
2*c^16*d^2*f^3))*((((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2
*c^5*f^2 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d
^5*f^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2
 - 160*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^
3*f^2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2
*b^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4
+ 4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*
f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*C^2*a^4*c^5*f^2 - 4*C^2*
b^4*c^5*f^2 + 24*C^2*a^2*b^2*c^5*f^2 + 40*C^2*a^4*c^3*d^2*f^2 + 40*C^2*b^4*
c^3*d^2*f^2 + 16*C^2*a*b^3*d^5*f^2 - 16*C^2*a^3*b*d^5*f^2 - 20*C^2*a^4*c*d^
4*f^2 - 20*C^2*b^4*c*d^4*f^2 + 80*C^2*a*b^3*c^4*d*f^2 - 80*C^2*a^3*b*c^4*d*
f^2 - 160*C^2*a*b^3*c^2*d^3*f^2 + 120*C^2*a^2*b^2*c*d^4*f^2 + 160*C^2*a^3*b
*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^
2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*1i - (
((((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*C^2
*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a
^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*
c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a
^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)
^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*
(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4
*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*C^2*a^4*c^5*f^2 - 4*C^2*b^4*c^5*f^2 + 24*
C^2*a^2*b^2*c^5*f^2 + 40*C^2*a^4*c^3*d^2*f^2 + 40*C^2*b^4*c^3*d^2*f^2 + 16*
C^2*a*b^3*d^5*f^2 - 16*C^2*a^3*b*d^5*f^2 - 20*C^2*a^4*c*d^4*f^2 - 20*C^2*b^

4*c*d^4*f^2 + 80*C^2*a*b^3*c^4*d*f^2 - 80*C^2*a^3*b*c^4*d*f^2 - 160*C^2*a*b^3*c^2*d^3*f^2 + 120*C^2*a^2*b^2*c*d^4*f^2 + 160*C^2*a^3*b*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(32*C*b^2*d^21*f^4 - 32*C*a^2*d^21*f^4 - (c + d*tan(e + f*x))^(1/2)*((((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*C^2*a^4*c^5*f^2 - 4*C^2*b^4*c^5*f^2 + 24*C^2*a^2*b^2*c^5*f^2 + 40*C^2*a^4*c^3*d^2*f^2 + 40*C^2*b^4*c^3*d^2*f^2 + 16*C^2*a*b^3*d^5*f^2 - 16*C^2*a^3*b*d^5*f^2 - 20*C^2*a^4*c*d^4*f^2 - 20*C^2*b^4*c*d^4*f^2 + 80*C^2*a*b^3*c^4*d*f^2 - 80*C^2*a^3*b*c^4*d*f^2 - 160*C^2*a*b^3*c^2*d^3*f^2 + 120*C^2*a^2*b^2*c*d^4*f^2 + 160*C^2*a^3*b*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 160*C*a^2*c^2*d^19*f^4 - 128*C*a^2*c^4*d^17*f^4 + 896*C*a^2*c^6*d^15*f^4 + 3136*C*a^2*c^8*d^13*f^4 + 4928*C*a^2*c^10*d^11*f^4 + 4480*C*a^2*c^12*d^9*f^4 + 2432*C*a^2*c^14*d^7*f^4 + 736*C*a^2*c^16*d^5*f^4 + 96*C*a^2*c^18*d^3*f^4 + 160*C*b^2*c^2*d^19*f^4 + 128*C*b^2*c^4*d^17*f^4 - 896*C*b^2*c^6*d^15*f^4 - 3136*C*b^2*c^8*d^13*f^4 - 4928*C*b^2*c^10*d^11*f^4 - 4480*C*b^2*c^12*d^9*f^4 - 2432*C*b^2*c^14*d^7*f^4 - 736*C*b^2*c^16*d^5*f^4 - 96*C*b^2*c^18*d^3*f^4 + 192*C*a*b*c*d^20*f^4 + 1472*C*a*b*c^3*d^18*f^4 + 4864*C*a*b*c^5*d^16*f^4 + 8960*C*a*b*c^7*d^14*f^4 + 9856*C*a*b*c^9*d^12*f^4 + 6272*C*a*b*c^11*d^10*f^4 + 1792*C*a*b*c^13*d^8*f^4 - 256*C*a*b*c^15*d^6*f^4 - 320*C*a*b*c^17*d^4*f^4 - 64*C*a*b*c^19*d^2*f^4) + (c + d*tan(e + f*x))^(1/2)*(96*C^2*a^2*b^2*d^18*f^3 - 16*C^2*b^4*d^18*f^3 - 16*C^2*a^4*d^18*f^3 + 320*C^2*a^4*c^4*d^14*f^3 + 1024*C^2*a^4*c^6*d^12*f^3 + 1440*C^2*a^4*c^8*d^10*f^3 + 1024*C^2*a^4*c^10*d^8*f^3 + 320*C^2*a^4*c^12*d^6*f^3 - 16*C^2*a^4*c^16*d^2*f^3 + 320*C^2*b^4*c^4*d^14*f^3 + 1024*C^2*b^4*c^6*d^12*f^3 + 1440*C^2*b^4*c^8*d^10*f^3 + 1024*C^2*b^4*c^10*d^8*f^3 + 320*C^2*b^4*c^12*d^6*f^3 - 16*C^2*b^4*c^16*d^2*f^3 - 256*C^2*a*b^3*c*d^17*f^3 + 256*C^2*a^3*b*c*d^17*f^3 - 1280*C^2*a*b^3*c^3*d^15*f^3 - 2304*C^2*a*b^3*c^5*d^13*f^3 - 1280*C^2*a*b^3*c^7*d^11*f^3 + 1280*C^2*a*b^3*c^9*d^9*f^3 + 2304*C^2*a*b^3*c^11*d^7*f^3 + 1280*C^2*a*b^3*c^13*d^5*f^3 + 256*C^2*a*b^3*c^15*d^3*f^3 + 1280*C^2*a^3*b*c^3*d^15*f^3 + 2304*C^2*a^3*b*c^5*d^13*f^3 + 1280*C^2*a^3*b*c^7*d^11*f^3 - 1280*C^2*a^3*b*c^9*d^9*f^3 - 2304*C^2*a^3*b*c^11*d^7*f^3 - 1280*C^2*a^3*b*c^13*d^5*f^3 - 256*C^2*a^3*b*c^15*d^3*f^3 - 1920*C^2*a^2*b^2*c^4*d^14*f^3 - 6144*C^2*a^2*b^2*c^6*d^12*f^3 - 8640*C^2*a^2*b^2*c^8*d^10*f^3 - 6144*C^2*a^2*b^2*c^10*d^8*f^3 - 1920*C^2*a^2*b^2*c^12*d^6*f^3 + 96*C^2*a^2*b^2*

c^16*d^2*f^3))*((((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c
^5*f^2 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5
*f^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 -
160*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*
f^2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b
^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 +
4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^
4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*C^2*a^4*c^5*f^2 - 4*C^2*b^
4*c^5*f^2 + 24*C^2*a^2*b^2*c^5*f^2 + 40*C^2*a^4*c^3*d^2*f^2 + 40*C^2*b^4*c^
3*d^2*f^2 + 16*C^2*a*b^3*d^5*f^2 - 16*C^2*a^3*b*d^5*f^2 - 20*C^2*a^4*c*d^4*
f^2 - 20*C^2*b^4*c*d^4*f^2 + 80*C^2*a*b^3*c^4*d*f^2 - 80*C^2*a^3*b*c^4*d*f^
2 - 160*C^2*a*b^3*c^2*d^3*f^2 + 120*C^2*a^2*b^2*c*d^4*f^2 + 160*C^2*a^3*b*c
^2*d^3*f^2 - 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*
d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*1i)/((((
((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*C^2*a
^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3
*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^
4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2
*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2
/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(1
6*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f
^4 + 80*c^8*d^2*f^4))^(1/2) - 4*C^2*a^4*c^5*f^2 - 4*C^2*b^4*c^5*f^2 + 24*C^
2*a^2*b^2*c^5*f^2 + 40*C^2*a^4*c^3*d^2*f^2 + 40*C^2*b^4*c^3*d^2*f^2 + 16*C^
2*a*b^3*d^5*f^2 - 16*C^2*a^3*b*d^5*f^2 - 20*C^2*a^4*c*d^4*f^2 - 20*C^2*b^4*
c*d^4*f^2 + 80*C^2*a*b^3*c^4*d*f^2 - 80*C^2*a^3*b*c^4*d*f^2 - 160*C^2*a*b^3
*c^2*d^3*f^2 + 120*C^2*a^2*b^2*c*d^4*f^2 + 160*C^2*a^3*b*c^2*d^3*f^2 - 240*
C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*
d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/
2)*((((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*
C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^
2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b
^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^
2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f
^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^
2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*
d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*C^2*a^4*c^5*f^2 - 4*C^2*b^4*c^5*f^2 +
24*C^2*a^2*b^2*c^5*f^2 + 40*C^2*a^4*c^3*d^2*f^2 + 40*C^2*b^4*c^3*d^2*f^2 +
16*C^2*a*b^3*d^5*f^2 - 16*C^2*a^3*b*d^5*f^2 - 20*C^2*a^4*c*d^4*f^2 - 20*C^2
*b^4*c*d^4*f^2 + 80*C^2*a*b^3*c^4*d*f^2 - 80*C^2*a^3*b*c^4*d*f^2 - 160*C^2*
a*b^3*c^2*d^3*f^2 + 120*C^2*a^2*b^2*c*d^4*f^2 + 160*C^2*a^3*b*c^2*d^3*f^2 -
240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10
*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640
*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5
+ 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17
*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 32*C*a^2*d^21*f^4 + 32*C*b

^2*d^21*f^4 - 160*C*a^2*c^2*d^19*f^4 - 128*C*a^2*c^4*d^17*f^4 + 896*C*a^2*c
^6*d^15*f^4 + 3136*C*a^2*c^8*d^13*f^4 + 4928*C*a^2*c^10*d^11*f^4 + 4480*C*a
^2*c^12*d^9*f^4 + 2432*C*a^2*c^14*d^7*f^4 + 736*C*a^2*c^16*d^5*f^4 + 96*C*a
^2*c^18*d^3*f^4 + 160*C*b^2*c^2*d^19*f^4 + 128*C*b^2*c^4*d^17*f^4 - 896*C*b
^2*c^6*d^15*f^4 - 3136*C*b^2*c^8*d^13*f^4 - 4928*C*b^2*c^10*d^11*f^4 - 4480
*C*b^2*c^12*d^9*f^4 - 2432*C*b^2*c^14*d^7*f^4 - 736*C*b^2*c^16*d^5*f^4 - 96
*C*b^2*c^18*d^3*f^4 + 192*C*a*b*c*d^20*f^4 + 1472*C*a*b*c^3*d^18*f^4 + 4864
*C*a*b*c^5*d^16*f^4 + 8960*C*a*b*c^7*d^14*f^4 + 9856*C*a*b*c^9*d^12*f^4 + 6
272*C*a*b*c^11*d^10*f^4 + 1792*C*a*b*c^13*d^8*f^4 - 256*C*a*b*c^15*d^6*f^4
- 320*C*a*b*c^17*d^4*f^4 - 64*C*a*b*c^19*d^2*f^4) - (c + d*tan(e + f*x))^(1
/2)*(96*C^2*a^2*b^2*d^18*f^3 - 16*C^2*b^4*d^18*f^3 - 16*C^2*a^4*d^18*f^3 +
320*C^2*a^4*c^4*d^14*f^3 + 1024*C^2*a^4*c^6*d^12*f^3 + 1440*C^2*a^4*c^8*d^1
0*f^3 + 1024*C^2*a^4*c^10*d^8*f^3 + 320*C^2*a^4*c^12*d^6*f^3 - 16*C^2*a^4*c
^16*d^2*f^3 + 320*C^2*b^4*c^4*d^14*f^3 + 1024*C^2*b^4*c^6*d^12*f^3 + 1440*C
^2*b^4*c^8*d^10*f^3 + 1024*C^2*b^4*c^10*d^8*f^3 + 320*C^2*b^4*c^12*d^6*f^3
- 16*C^2*b^4*c^16*d^2*f^3 - 256*C^2*a*b^3*c*d^17*f^3 + 256*C^2*a^3*b*c*d^17
*f^3 - 1280*C^2*a*b^3*c^3*d^15*f^3 - 2304*C^2*a*b^3*c^5*d^13*f^3 - 1280*C^2
*a*b^3*c^7*d^11*f^3 + 1280*C^2*a*b^3*c^9*d^9*f^3 + 2304*C^2*a*b^3*c^11*d^7*
f^3 + 1280*C^2*a*b^3*c^13*d^5*f^3 + 256*C^2*a*b^3*c^15*d^3*f^3 + 1280*C^2*a
^3*b*c^3*d^15*f^3 + 2304*C^2*a^3*b*c^5*d^13*f^3 + 1280*C^2*a^3*b*c^7*d^11*f
^3 - 1280*C^2*a^3*b*c^9*d^9*f^3 - 2304*C^2*a^3*b*c^11*d^7*f^3 - 1280*C^2*a^
3*b*c^13*d^5*f^3 - 256*C^2*a^3*b*c^15*d^3*f^3 - 1920*C^2*a^2*b^2*c^4*d^14*f
^3 - 6144*C^2*a^2*b^2*c^6*d^12*f^3 - 8640*C^2*a^2*b^2*c^8*d^10*f^3 - 6144*C
^2*a^2*b^2*c^10*d^8*f^3 - 1920*C^2*a^2*b^2*c^12*d^6*f^3 + 96*C^2*a^2*b^2*c^
16*d^2*f^3))*((((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5
*f^2 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f
^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 - 1
60*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^
2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2
*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*
C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4
+ 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*C^2*a^4*c^5*f^2 - 4*C^2*b^4*
c^5*f^2 + 24*C^2*a^2*b^2*c^5*f^2 + 40*C^2*a^4*c^3*d^2*f^2 + 40*C^2*b^4*c^3*
d^2*f^2 + 16*C^2*a*b^3*d^5*f^2 - 16*C^2*a^3*b*d^5*f^2 - 20*C^2*a^4*c*d^4*f^
2 - 20*C^2*b^4*c*d^4*f^2 + 80*C^2*a*b^3*c^4*d*f^2 - 80*C^2*a^3*b*c^4*d*f^2
- 160*C^2*a*b^3*c^2*d^3*f^2 + 120*C^2*a^2*b^2*c*d^4*f^2 + 160*C^2*a^3*b*c^2
*d^3*f^2 - 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^
8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2) + (((((8*C
^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*C^2*a^4*c^
3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3*b*d^
5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^4*d*f
^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2*b^2*
c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2/4 -
(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(16*c^1
0*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 +

80*c^8*d^2*f^4))^(1/2) - 4*C^2*a^4*c^5*f^2 - 4*C^2*b^4*c^5*f^2 + 24*C^2*a^2
*b^2*c^5*f^2 + 40*C^2*a^4*c^3*d^2*f^2 + 40*C^2*b^4*c^3*d^2*f^2 + 16*C^2*a*b
^3*d^5*f^2 - 16*C^2*a^3*b*d^5*f^2 - 20*C^2*a^4*c*d^4*f^2 - 20*C^2*b^4*c*d^4
*f^2 + 80*C^2*a*b^3*c^4*d*f^2 - 80*C^2*a^3*b*c^4*d*f^2 - 160*C^2*a*b^3*c^2*
d^3*f^2 + 120*C^2*a^2*b^2*c*d^4*f^2 + 160*C^2*a^3*b*c^2*d^3*f^2 - 240*C^2*a
^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f
^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(32*C*b^2*d^21*f^4 - 32*C*a^2*
d^21*f^4 - (c + d*tan(e + f*x))^(1/2)*((((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5
*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2
*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 +
40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 +
320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*
d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2
*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d
^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*C^2
*a^4*c^5*f^2 - 4*C^2*b^4*c^5*f^2 + 24*C^2*a^2*b^2*c^5*f^2 + 40*C^2*a^4*c^3*
d^2*f^2 + 40*C^2*b^4*c^3*d^2*f^2 + 16*C^2*a*b^3*d^5*f^2 - 16*C^2*a^3*b*d^5*
f^2 - 20*C^2*a^4*c*d^4*f^2 - 20*C^2*b^4*c*d^4*f^2 + 80*C^2*a*b^3*c^4*d*f^2
- 80*C^2*a^3*b*c^4*d*f^2 - 160*C^2*a*b^3*c^2*d^3*f^2 + 120*C^2*a^2*b^2*c*d^
4*f^2 + 160*C^2*a^3*b*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*
f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^
2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680
*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*
f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^
2*f^5) - 160*C*a^2*c^2*d^19*f^4 - 128*C*a^2*c^4*d^17*f^4 + 896*C*a^2*c^6*d^
15*f^4 + 3136*C*a^2*c^8*d^13*f^4 + 4928*C*a^2*c^10*d^11*f^4 + 4480*C*a^2*c^
12*d^9*f^4 + 2432*C*a^2*c^14*d^7*f^4 + 736*C*a^2*c^16*d^5*f^4 + 96*C*a^2*c^
18*d^3*f^4 + 160*C*b^2*c^2*d^19*f^4 + 128*C*b^2*c^4*d^17*f^4 - 896*C*b^2*c^
6*d^15*f^4 - 3136*C*b^2*c^8*d^13*f^4 - 4928*C*b^2*c^10*d^11*f^4 - 4480*C*b^
2*c^12*d^9*f^4 - 2432*C*b^2*c^14*d^7*f^4 - 736*C*b^2*c^16*d^5*f^4 - 96*C*b^
2*c^18*d^3*f^4 + 192*C*a*b*c*d^20*f^4 + 1472*C*a*b*c^3*d^18*f^4 + 4864*C*a*
b*c^5*d^16*f^4 + 8960*C*a*b*c^7*d^14*f^4 + 9856*C*a*b*c^9*d^12*f^4 + 6272*C
*a*b*c^11*d^10*f^4 + 1792*C*a*b*c^13*d^8*f^4 - 256*C*a*b*c^15*d^6*f^4 - 320
*C*a*b*c^17*d^4*f^4 - 64*C*a*b*c^19*d^2*f^4) + (c + d*tan(e + f*x))^(1/2)*(
96*C^2*a^2*b^2*d^18*f^3 - 16*C^2*b^4*d^18*f^3 - 16*C^2*a^4*d^18*f^3 + 320*C
^2*a^4*c^4*d^14*f^3 + 1024*C^2*a^4*c^6*d^12*f^3 + 1440*C^2*a^4*c^8*d^10*f^3
+ 1024*C^2*a^4*c^10*d^8*f^3 + 320*C^2*a^4*c^12*d^6*f^3 - 16*C^2*a^4*c^16*d
^2*f^3 + 320*C^2*b^4*c^4*d^14*f^3 + 1024*C^2*b^4*c^6*d^12*f^3 + 1440*C^2*b^
4*c^8*d^10*f^3 + 1024*C^2*b^4*c^10*d^8*f^3 + 320*C^2*b^4*c^12*d^6*f^3 - 16*
C^2*b^4*c^16*d^2*f^3 - 256*C^2*a*b^3*c*d^17*f^3 + 256*C^2*a^3*b*c*d^17*f^3
- 1280*C^2*a*b^3*c^3*d^15*f^3 - 2304*C^2*a*b^3*c^5*d^13*f^3 - 1280*C^2*a*b^
3*c^7*d^11*f^3 + 1280*C^2*a*b^3*c^9*d^9*f^3 + 2304*C^2*a*b^3*c^11*d^7*f^3 +
1280*C^2*a*b^3*c^13*d^5*f^3 + 256*C^2*a*b^3*c^15*d^3*f^3 + 1280*C^2*a^3*b*
c^3*d^15*f^3 + 2304*C^2*a^3*b*c^5*d^13*f^3 + 1280*C^2*a^3*b*c^7*d^11*f^3 -
1280*C^2*a^3*b*c^9*d^9*f^3 - 2304*C^2*a^3*b*c^11*d^7*f^3 - 1280*C^2*a^3*b*c

^13*d^5*f^3 - 256*C^2*a^3*b*c^15*d^3*f^3 - 1920*C^2*a^2*b^2*c^4*d^14*f^3 - 6144*C^2*a^2*b^2*c^6*d^12*f^3 - 8640*C^2*a^2*b^2*c^8*d^10*f^3 - 6144*C^2*a^2*b^2*c^10*d^8*f^3 - 1920*C^2*a^2*b^2*c^12*d^6*f^3 + 96*C^2*a^2*b^2*c^16*d^2*f^3))*((((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*C^2*a^4*c^5*f^2 - 4*C^2*b^4*c^5*f^2 + 24*C^2*a^2*b^2*c^5*f^2 + 40*C^2*a^4*c^3*d^2*f^2 + 40*C^2*b^4*c^3*d^2*f^2 + 16*C^2*a*b^3*d^5*f^2 - 16*C^2*a^3*b*d^5*f^2 - 20*C^2*a^4*c*d^4*f^2 - 20*C^2*b^4*c*d^4*f^2 + 80*C^2*a*b^3*c^4*d*f^2 - 80*C^2*a^3*b*c^4*d*f^2 - 160*C^2*a*b^3*c^2*d^3*f^2 + 120*C^2*a^2*b^2*c*d^4*f^2 + 160*C^2*a^3*b*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2) - 64*C^3*a^3*b^3*d^16*f^2 - 192*C^3*a^6*c^3*d^13*f^2 - 480*C^3*a^6*c^5*d^11*f^2 - 640*C^3*a^6*c^7*d^9*f^2 - 480*C^3*a^6*c^9*d^7*f^2 - 192*C^3*a^6*c^11*d^5*f^2 - 32*C^3*a^6*c^13*d^3*f^2 + 192*C^3*b^6*c^3*d^13*f^2 + 480*C^3*b^6*c^5*d^11*f^2 + 640*C^3*b^6*c^7*d^9*f^2 + 480*C^3*b^6*c^9*d^7*f^2 + 192*C^3*b^6*c^11*d^5*f^2 + 32*C^3*b^6*c^13*d^3*f^2 - 32*C^3*a*b^5*d^16*f^2 - 32*C^3*a^5*b*d^16*f^2 - 32*C^3*a^6*c*d^15*f^2 + 32*C^3*b^6*c*d^15*f^2 - 160*C^3*a*b^5*c^2*d^14*f^2 - 288*C^3*a*b^5*c^4*d^12*f^2 - 160*C^3*a*b^5*c^6*d^10*f^2 + 160*C^3*a*b^5*c^8*d^8*f^2 + 288*C^3*a*b^5*c^10*d^6*f^2 + 160*C^3*a*b^5*c^12*d^4*f^2 + 32*C^3*a*b^5*c^14*d^2*f^2 + 32*C^3*a^2*b^4*c*d^15*f^2 - 32*C^3*a^4*b^2*c*d^15*f^2 - 160*C^3*a^5*b*c^2*d^14*f^2 - 288*C^3*a^5*b*c^4*d^12*f^2 - 160*C^3*a^5*b*c^6*d^10*f^2 + 160*C^3*a^5*b*c^8*d^8*f^2 + 288*C^3*a^5*b*c^10*d^6*f^2 + 160*C^3*a^5*b*c^12*d^4*f^2 + 32*C^3*a^5*b*c^14*d^2*f^2 + 192*C^3*a^2*b^4*c^3*d^13*f^2 + 480*C^3*a^2*b^4*c^5*d^11*f^2 + 640*C^3*a^2*b^4*c^7*d^9*f^2 + 480*C^3*a^2*b^4*c^9*d^7*f^2 + 192*C^3*a^2*b^4*c^11*d^5*f^2 + 32*C^3*a^2*b^4*c^13*d^3*f^2 - 320*C^3*a^3*b^3*c^2*d^14*f^2 - 576*C^3*a^3*b^3*c^4*d^12*f^2 - 320*C^3*a^3*b^3*c^6*d^10*f^2 + 320*C^3*a^3*b^3*c^8*d^8*f^2 + 576*C^3*a^3*b^3*c^10*d^6*f^2 + 320*C^3*a^3*b^3*c^12*d^4*f^2 + 64*C^3*a^3*b^3*c^14*d^2*f^2 - 192*C^3*a^4*b^2*c^3*d^13*f^2 - 480*C^3*a^4*b^2*c^5*d^11*f^2 - 640*C^3*a^4*b^2*c^7*d^9*f^2 - 480*C^3*a^4*b^2*c^9*d^7*f^2 - 192*C^3*a^4*b^2*c^11*d^5*f^2 - 32*C^3*a^4*b^2*c^13*d^3*f^2))*((((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*C^2*a^4*c^5*f^2 - 4*C^2*b^4*c^5*f^2 + 24*C^2*a^2*b^2*c^5*f^2 + 40*C^2*a^4*c

^3*d^2*f^2 + 40*C^2*b^4*c^3*d^2*f^2 + 16*C^2*a*b^3*d^5*f^2 - 16*C^2*a^3*b*d
^5*f^2 - 20*C^2*a^4*c*d^4*f^2 - 20*C^2*b^4*c*d^4*f^2 + 80*C^2*a*b^3*c^4*d*f
^2 - 80*C^2*a^3*b*c^4*d*f^2 - 160*C^2*a*b^3*c^2*d^3*f^2 + 120*C^2*a^2*b^2*c
*d^4*f^2 + 160*C^2*a^3*b*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^
10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8
*d^2*f^4)))^(1/2)*2i - atan(-(((-(((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 -
48*C^2*a^2*b^2*c^5*f^2 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 -
32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^
2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C
^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^
2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 +
6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4
+ 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*C^2*a^4*c
^5*f^2 + 4*C^2*b^4*c^5*f^2 - 24*C^2*a^2*b^2*c^5*f^2 - 40*C^2*a^4*c^3*d^2*f^
2 - 40*C^2*b^4*c^3*d^2*f^2 - 16*C^2*a*b^3*d^5*f^2 + 16*C^2*a^3*b*d^5*f^2 +
20*C^2*a^4*c*d^4*f^2 + 20*C^2*b^4*c*d^4*f^2 - 80*C^2*a*b^3*c^4*d*f^2 + 80*C
^2*a^3*b*c^4*d*f^2 + 160*C^2*a*b^3*c^2*d^3*f^2 - 120*C^2*a^2*b^2*c*d^4*f^2
- 160*C^2*a^3*b*c^2*d^3*f^2 + 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 +
d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)
))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5
*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2
*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 +
40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 +
320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*
d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2
*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d
^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*C^2
*a^4*c^5*f^2 + 4*C^2*b^4*c^5*f^2 - 24*C^2*a^2*b^2*c^5*f^2 - 40*C^2*a^4*c^3*
d^2*f^2 - 40*C^2*b^4*c^3*d^2*f^2 - 16*C^2*a*b^3*d^5*f^2 + 16*C^2*a^3*b*d^5*
f^2 + 20*C^2*a^4*c*d^4*f^2 + 20*C^2*b^4*c*d^4*f^2 - 80*C^2*a*b^3*c^4*d*f^2
+ 80*C^2*a^3*b*c^4*d*f^2 + 160*C^2*a*b^3*c^2*d^3*f^2 - 120*C^2*a^2*b^2*c*d^
4*f^2 - 160*C^2*a^3*b*c^2*d^3*f^2 + 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*
f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^
2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680
*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*
f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^
2*f^5) - 32*C*a^2*d^21*f^4 + 32*C*b^2*d^21*f^4 - 160*C*a^2*c^2*d^19*f^4 - 1
28*C*a^2*c^4*d^17*f^4 + 896*C*a^2*c^6*d^15*f^4 + 3136*C*a^2*c^8*d^13*f^4 +
4928*C*a^2*c^10*d^11*f^4 + 4480*C*a^2*c^12*d^9*f^4 + 2432*C*a^2*c^14*d^7*f^
4 + 736*C*a^2*c^16*d^5*f^4 + 96*C*a^2*c^18*d^3*f^4 + 160*C*b^2*c^2*d^19*f^4
+ 128*C*b^2*c^4*d^17*f^4 - 896*C*b^2*c^6*d^15*f^4 - 3136*C*b^2*c^8*d^13*f^
4 - 4928*C*b^2*c^10*d^11*f^4 - 4480*C*b^2*c^12*d^9*f^4 - 2432*C*b^2*c^14*d^
7*f^4 - 736*C*b^2*c^16*d^5*f^4 - 96*C*b^2*c^18*d^3*f^4 + 192*C*a*b*c*d^20*f
^4 + 1472*C*a*b*c^3*d^18*f^4 + 4864*C*a*b*c^5*d^16*f^4 + 8960*C*a*b*c^7*d^1
4*f^4 + 9856*C*a*b*c^9*d^12*f^4 + 6272*C*a*b*c^11*d^10*f^4 + 1792*C*a*b*c^1

3*d^8*f^4 - 256*C*a*b*c^15*d^6*f^4 - 320*C*a*b*c^17*d^4*f^4 - 64*C*a*b*c^19
*d^2*f^4) - (c + d*tan(e + f*x))^(1/2)*(96*C^2*a^2*b^2*d^18*f^3 - 16*C^2*b^
4*d^18*f^3 - 16*C^2*a^4*d^18*f^3 + 320*C^2*a^4*c^4*d^14*f^3 + 1024*C^2*a^4*
c^6*d^12*f^3 + 1440*C^2*a^4*c^8*d^10*f^3 + 1024*C^2*a^4*c^10*d^8*f^3 + 320*
C^2*a^4*c^12*d^6*f^3 - 16*C^2*a^4*c^16*d^2*f^3 + 320*C^2*b^4*c^4*d^14*f^3 +
1024*C^2*b^4*c^6*d^12*f^3 + 1440*C^2*b^4*c^8*d^10*f^3 + 1024*C^2*b^4*c^10*
d^8*f^3 + 320*C^2*b^4*c^12*d^6*f^3 - 16*C^2*b^4*c^16*d^2*f^3 - 256*C^2*a*b^
3*c*d^17*f^3 + 256*C^2*a^3*b*c*d^17*f^3 - 1280*C^2*a*b^3*c^3*d^15*f^3 - 230
4*C^2*a*b^3*c^5*d^13*f^3 - 1280*C^2*a*b^3*c^7*d^11*f^3 + 1280*C^2*a*b^3*c^9
*d^9*f^3 + 2304*C^2*a*b^3*c^11*d^7*f^3 + 1280*C^2*a*b^3*c^13*d^5*f^3 + 256*
C^2*a*b^3*c^15*d^3*f^3 + 1280*C^2*a^3*b*c^3*d^15*f^3 + 2304*C^2*a^3*b*c^5*d
^13*f^3 + 1280*C^2*a^3*b*c^7*d^11*f^3 - 1280*C^2*a^3*b*c^9*d^9*f^3 - 2304*C
^2*a^3*b*c^11*d^7*f^3 - 1280*C^2*a^3*b*c^13*d^5*f^3 - 256*C^2*a^3*b*c^15*d^
3*f^3 - 1920*C^2*a^2*b^2*c^4*d^14*f^3 - 6144*C^2*a^2*b^2*c^6*d^12*f^3 - 864
0*C^2*a^2*b^2*c^8*d^10*f^3 - 6144*C^2*a^2*b^2*c^10*d^8*f^3 - 1920*C^2*a^2*b
^2*c^12*d^6*f^3 + 96*C^2*a^2*b^2*c^16*d^2*f^3))*(-(((8*C^2*a^4*c^5*f^2 + 8*
C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*
b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*
c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*
c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2
*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8
+ 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4
+ 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1
/2) + 4*C^2*a^4*c^5*f^2 + 4*C^2*b^4*c^5*f^2 - 24*C^2*a^2*b^2*c^5*f^2 - 40*C
^2*a^4*c^3*d^2*f^2 - 40*C^2*b^4*c^3*d^2*f^2 - 16*C^2*a*b^3*d^5*f^2 + 16*C^2
*a^3*b*d^5*f^2 + 20*C^2*a^4*c*d^4*f^2 + 20*C^2*b^4*c*d^4*f^2 - 80*C^2*a*b^3
*c^4*d*f^2 + 80*C^2*a^3*b*c^4*d*f^2 + 160*C^2*a*b^3*c^2*d^3*f^2 - 120*C^2*a
^2*b^2*c*d^4*f^2 - 160*C^2*a^3*b*c^2*d^3*f^2 + 240*C^2*a^2*b^2*c^3*d^2*f^2)
/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4
+ 5*c^8*d^2*f^4)))^(1/2)*1i - ((-(((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2
- 48*C^2*a^2*b^2*c^5*f^2 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2
- 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C
^2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*
C^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f
^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6
+ 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^
4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*C^2*a^4*
c^5*f^2 + 4*C^2*b^4*c^5*f^2 - 24*C^2*a^2*b^2*c^5*f^2 - 40*C^2*a^4*c^3*d^2*f
^2 - 40*C^2*b^4*c^3*d^2*f^2 - 16*C^2*a*b^3*d^5*f^2 + 16*C^2*a^3*b*d^5*f^2 +
20*C^2*a^4*c*d^4*f^2 + 20*C^2*b^4*c*d^4*f^2 - 80*C^2*a*b^3*c^4*d*f^2 + 80*
C^2*a^3*b*c^4*d*f^2 + 160*C^2*a*b^3*c^2*d^3*f^2 - 120*C^2*a^2*b^2*c*d^4*f^2
- 160*C^2*a^3*b*c^2*d^3*f^2 + 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 +
d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4
)))^(1/2)*(32*C*b^2*d^21*f^4 - 32*C*a^2*d^21*f^4 - (c + d*tan(e + f*x))^(1/
2)*(-(((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80

*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C
^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*
b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 240*C
^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*
f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b
^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6
*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*C^2*a^4*c^5*f^2 + 4*C^2*b^4*c^5*f^2 -
24*C^2*a^2*b^2*c^5*f^2 - 40*C^2*a^4*c^3*d^2*f^2 - 40*C^2*b^4*c^3*d^2*f^2 -
16*C^2*a*b^3*d^5*f^2 + 16*C^2*a^3*b*d^5*f^2 + 20*C^2*a^4*c*d^4*f^2 + 20*C^
2*b^4*c*d^4*f^2 - 80*C^2*a*b^3*c^4*d*f^2 + 80*C^2*a^3*b*c^4*d*f^2 + 160*C^2
*a*b^3*c^2*d^3*f^2 - 120*C^2*a^2*b^2*c*d^4*f^2 - 160*C^2*a^3*b*c^2*d^3*f^2
+ 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 1
0*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 64
0*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5
+ 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^1
7*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 160*C*a^2*c^2*d^19*f^4 -
128*C*a^2*c^4*d^17*f^4 + 896*C*a^2*c^6*d^15*f^4 + 3136*C*a^2*c^8*d^13*f^4 +
4928*C*a^2*c^10*d^11*f^4 + 4480*C*a^2*c^12*d^9*f^4 + 2432*C*a^2*c^14*d^7*f
^4 + 736*C*a^2*c^16*d^5*f^4 + 96*C*a^2*c^18*d^3*f^4 + 160*C*b^2*c^2*d^19*f^
4 + 128*C*b^2*c^4*d^17*f^4 - 896*C*b^2*c^6*d^15*f^4 - 3136*C*b^2*c^8*d^13*f
^4 - 4928*C*b^2*c^10*d^11*f^4 - 4480*C*b^2*c^12*d^9*f^4 - 2432*C*b^2*c^14*d
^7*f^4 - 736*C*b^2*c^16*d^5*f^4 - 96*C*b^2*c^18*d^3*f^4 + 192*C*a*b*c*d^20*
f^4 + 1472*C*a*b*c^3*d^18*f^4 + 4864*C*a*b*c^5*d^16*f^4 + 8960*C*a*b*c^7*d^
14*f^4 + 9856*C*a*b*c^9*d^12*f^4 + 6272*C*a*b*c^11*d^10*f^4 + 1792*C*a*b*c^
13*d^8*f^4 - 256*C*a*b*c^15*d^6*f^4 - 320*C*a*b*c^17*d^4*f^4 - 64*C*a*b*c^1
9*d^2*f^4) + (c + d*tan(e + f*x))^(1/2)*(96*C^2*a^2*b^2*d^18*f^3 - 16*C^2*b
^4*d^18*f^3 - 16*C^2*a^4*d^18*f^3 + 320*C^2*a^4*c^4*d^14*f^3 + 1024*C^2*a^4
*c^6*d^12*f^3 + 1440*C^2*a^4*c^8*d^10*f^3 + 1024*C^2*a^4*c^10*d^8*f^3 + 320
*C^2*a^4*c^12*d^6*f^3 - 16*C^2*a^4*c^16*d^2*f^3 + 320*C^2*b^4*c^4*d^14*f^3
+ 1024*C^2*b^4*c^6*d^12*f^3 + 1440*C^2*b^4*c^8*d^10*f^3 + 1024*C^2*b^4*c^10
*d^8*f^3 + 320*C^2*b^4*c^12*d^6*f^3 - 16*C^2*b^4*c^16*d^2*f^3 - 256*C^2*a*b
^3*c*d^17*f^3 + 256*C^2*a^3*b*c*d^17*f^3 - 1280*C^2*a*b^3*c^3*d^15*f^3 - 23
04*C^2*a*b^3*c^5*d^13*f^3 - 1280*C^2*a*b^3*c^7*d^11*f^3 + 1280*C^2*a*b^3*c^
9*d^9*f^3 + 2304*C^2*a*b^3*c^11*d^7*f^3 + 1280*C^2*a*b^3*c^13*d^5*f^3 + 256
*C^2*a*b^3*c^15*d^3*f^3 + 1280*C^2*a^3*b*c^3*d^15*f^3 + 2304*C^2*a^3*b*c^5*
d^13*f^3 + 1280*C^2*a^3*b*c^7*d^11*f^3 - 1280*C^2*a^3*b*c^9*d^9*f^3 - 2304*
C^2*a^3*b*c^11*d^7*f^3 - 1280*C^2*a^3*b*c^13*d^5*f^3 - 256*C^2*a^3*b*c^15*d
^3*f^3 - 1920*C^2*a^2*b^2*c^4*d^14*f^3 - 6144*C^2*a^2*b^2*c^6*d^12*f^3 - 86
40*C^2*a^2*b^2*c^8*d^10*f^3 - 6144*C^2*a^2*b^2*c^10*d^8*f^3 - 1920*C^2*a^2*
b^2*c^12*d^6*f^3 + 96*C^2*a^2*b^2*c^16*d^2*f^3))*(-(((8*C^2*a^4*c^5*f^2 + 8
*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2
*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4
*c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b
*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C^
2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8

+ 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^
4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(
1/2) + 4*C^2*a^4*c^5*f^2 + 4*C^2*b^4*c^5*f^2 - 24*C^2*a^2*b^2*c^5*f^2 - 40*
C^2*a^4*c^3*d^2*f^2 - 40*C^2*b^4*c^3*d^2*f^2 - 16*C^2*a*b^3*d^5*f^2 + 16*C^
2*a^3*b*d^5*f^2 + 20*C^2*a^4*c*d^4*f^2 + 20*C^2*b^4*c*d^4*f^2 - 80*C^2*a*b^
3*c^4*d*f^2 + 80*C^2*a^3*b*c^4*d*f^2 + 160*C^2*a*b^3*c^2*d^3*f^2 - 120*C^2*
a^2*b^2*c*d^4*f^2 - 160*C^2*a^3*b*c^2*d^3*f^2 + 240*C^2*a^2*b^2*c^3*d^2*f^2
)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^
4 + 5*c^8*d^2*f^4)))^(1/2)*1i)/(((-(((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2
 - 48*C^2*a^2*b^2*c^5*f^2 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2
 - 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*
C^2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320
*C^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*
f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6
 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f
^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*C^2*a^4
*c^5*f^2 + 4*C^2*b^4*c^5*f^2 - 24*C^2*a^2*b^2*c^5*f^2 - 40*C^2*a^4*c^3*d^2*
f^2 - 40*C^2*b^4*c^3*d^2*f^2 - 16*C^2*a*b^3*d^5*f^2 + 16*C^2*a^3*b*d^5*f^2
+ 20*C^2*a^4*c*d^4*f^2 + 20*C^2*b^4*c*d^4*f^2 - 80*C^2*a*b^3*c^4*d*f^2 + 80
*C^2*a^3*b*c^4*d*f^2 + 160*C^2*a*b^3*c^2*d^3*f^2 - 120*C^2*a^2*b^2*c*d^4*f^
2 - 160*C^2*a^3*b*c^2*d^3*f^2 + 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4
+ d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^
4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c
^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d
^2*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2
 + 40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2
 + 320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^
2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a
^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2
*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*C
^2*a^4*c^5*f^2 + 4*C^2*b^4*c^5*f^2 - 24*C^2*a^2*b^2*c^5*f^2 - 40*C^2*a^4*c^
3*d^2*f^2 - 40*C^2*b^4*c^3*d^2*f^2 - 16*C^2*a*b^3*d^5*f^2 + 16*C^2*a^3*b*d^
5*f^2 + 20*C^2*a^4*c*d^4*f^2 + 20*C^2*b^4*c*d^4*f^2 - 80*C^2*a*b^3*c^4*d*f^
2 + 80*C^2*a^3*b*c^4*d*f^2 + 160*C^2*a*b^3*c^2*d^3*f^2 - 120*C^2*a^2*b^2*c*
d^4*f^2 - 160*C^2*a^3*b*c^2*d^3*f^2 + 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^1
0*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*
d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 76
80*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^1
0*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*
d^2*f^5) - 32*C*a^2*d^21*f^4 + 32*C*b^2*d^21*f^4 - 160*C*a^2*c^2*d^19*f^4 -
 128*C*a^2*c^4*d^17*f^4 + 896*C*a^2*c^6*d^15*f^4 + 3136*C*a^2*c^8*d^13*f^4
+ 4928*C*a^2*c^10*d^11*f^4 + 4480*C*a^2*c^12*d^9*f^4 + 2432*C*a^2*c^14*d^7*
f^4 + 736*C*a^2*c^16*d^5*f^4 + 96*C*a^2*c^18*d^3*f^4 + 160*C*b^2*c^2*d^19*f
^4 + 128*C*b^2*c^4*d^17*f^4 - 896*C*b^2*c^6*d^15*f^4 - 3136*C*b^2*c^8*d^13*
f^4 - 4928*C*b^2*c^10*d^11*f^4 - 4480*C*b^2*c^12*d^9*f^4 - 2432*C*b^2*c^14*

d^7*f^4 - 736*C*b^2*c^16*d^5*f^4 - 96*C*b^2*c^18*d^3*f^4 + 192*C*a*b*c*d^20
*f^4 + 1472*C*a*b*c^3*d^18*f^4 + 4864*C*a*b*c^5*d^16*f^4 + 8960*C*a*b*c^7*d
^14*f^4 + 9856*C*a*b*c^9*d^12*f^4 + 6272*C*a*b*c^11*d^10*f^4 + 1792*C*a*b*c
^13*d^8*f^4 - 256*C*a*b*c^15*d^6*f^4 - 320*C*a*b*c^17*d^4*f^4 - 64*C*a*b*c^
19*d^2*f^4) - (c + d*tan(e + f*x))^(1/2)*(96*C^2*a^2*b^2*d^18*f^3 - 16*C^2*
b^4*d^18*f^3 - 16*C^2*a^4*d^18*f^3 + 320*C^2*a^4*c^4*d^14*f^3 + 1024*C^2*a^
4*c^6*d^12*f^3 + 1440*C^2*a^4*c^8*d^10*f^3 + 1024*C^2*a^4*c^10*d^8*f^3 + 32
0*C^2*a^4*c^12*d^6*f^3 - 16*C^2*a^4*c^16*d^2*f^3 + 320*C^2*b^4*c^4*d^14*f^3
 + 1024*C^2*b^4*c^6*d^12*f^3 + 1440*C^2*b^4*c^8*d^10*f^3 + 1024*C^2*b^4*c^1
0*d^8*f^3 + 320*C^2*b^4*c^12*d^6*f^3 - 16*C^2*b^4*c^16*d^2*f^3 - 256*C^2*a*
b^3*c*d^17*f^3 + 256*C^2*a^3*b*c*d^17*f^3 - 1280*C^2*a*b^3*c^3*d^15*f^3 - 2
304*C^2*a*b^3*c^5*d^13*f^3 - 1280*C^2*a*b^3*c^7*d^11*f^3 + 1280*C^2*a*b^3*c
^9*d^9*f^3 + 2304*C^2*a*b^3*c^11*d^7*f^3 + 1280*C^2*a*b^3*c^13*d^5*f^3 + 25
6*C^2*a*b^3*c^15*d^3*f^3 + 1280*C^2*a^3*b*c^3*d^15*f^3 + 2304*C^2*a^3*b*c^5
*d^13*f^3 + 1280*C^2*a^3*b*c^7*d^11*f^3 - 1280*C^2*a^3*b*c^9*d^9*f^3 - 2304
*C^2*a^3*b*c^11*d^7*f^3 - 1280*C^2*a^3*b*c^13*d^5*f^3 - 256*C^2*a^3*b*c^15*
d^3*f^3 - 1920*C^2*a^2*b^2*c^4*d^14*f^3 - 6144*C^2*a^2*b^2*c^6*d^12*f^3 - 8
640*C^2*a^2*b^2*c^8*d^10*f^3 - 6144*C^2*a^2*b^2*c^10*d^8*f^3 - 1920*C^2*a^2
*b^2*c^12*d^6*f^3 + 96*C^2*a^2*b^2*c^16*d^2*f^3))*(-(((8*C^2*a^4*c^5*f^2 +
8*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^
2*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^
4*c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*
b*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C
^2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^
8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f
^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^
(1/2) + 4*C^2*a^4*c^5*f^2 + 4*C^2*b^4*c^5*f^2 - 24*C^2*a^2*b^2*c^5*f^2 - 40
*C^2*a^4*c^3*d^2*f^2 - 40*C^2*b^4*c^3*d^2*f^2 - 16*C^2*a*b^3*d^5*f^2 + 16*C
^2*a^3*b*d^5*f^2 + 20*C^2*a^4*c*d^4*f^2 + 20*C^2*b^4*c*d^4*f^2 - 80*C^2*a*b
^3*c^4*d*f^2 + 80*C^2*a^3*b*c^4*d*f^2 + 160*C^2*a*b^3*c^2*d^3*f^2 - 120*C^2
*a^2*b^2*c*d^4*f^2 - 160*C^2*a^3*b*c^2*d^3*f^2 + 240*C^2*a^2*b^2*c^3*d^2*f^
2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f
^4 + 5*c^8*d^2*f^4)))^(1/2) + ((-(((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 -
 48*C^2*a^2*b^2*c^5*f^2 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 -
 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^
2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C
^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^
2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 +
 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4
 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*C^2*a^4*c
^5*f^2 + 4*C^2*b^4*c^5*f^2 - 24*C^2*a^2*b^2*c^5*f^2 - 40*C^2*a^4*c^3*d^2*f^
2 - 40*C^2*b^4*c^3*d^2*f^2 - 16*C^2*a*b^3*d^5*f^2 + 16*C^2*a^3*b*d^5*f^2 +
20*C^2*a^4*c*d^4*f^2 + 20*C^2*b^4*c*d^4*f^2 - 80*C^2*a*b^3*c^4*d*f^2 + 80*C
^2*a^3*b*c^4*d*f^2 + 160*C^2*a*b^3*c^2*d^3*f^2 - 120*C^2*a^2*b^2*c*d^4*f^2
- 160*C^2*a^3*b*c^2*d^3*f^2 + 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 +

d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)
))^(1/2)*(32*C*b^2*d^21*f^4 - 32*C*a^2*d^21*f^4 - (c + d*tan(e + f*x))^(1/2
)*(-(((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*
C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^
2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b
^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^
2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f
^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^
2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*
d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*C^2*a^4*c^5*f^2 + 4*C^2*b^4*c^5*f^2 -
24*C^2*a^2*b^2*c^5*f^2 - 40*C^2*a^4*c^3*d^2*f^2 - 40*C^2*b^4*c^3*d^2*f^2 -
16*C^2*a*b^3*d^5*f^2 + 16*C^2*a^3*b*d^5*f^2 + 20*C^2*a^4*c*d^4*f^2 + 20*C^2
*b^4*c*d^4*f^2 - 80*C^2*a*b^3*c^4*d*f^2 + 80*C^2*a^3*b*c^4*d*f^2 + 160*C^2*
a*b^3*c^2*d^3*f^2 - 120*C^2*a^2*b^2*c*d^4*f^2 - 160*C^2*a^3*b*c^2*d^3*f^2 +
 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10
*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640
*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5
+ 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17
*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 160*C*a^2*c^2*d^19*f^4 - 1
28*C*a^2*c^4*d^17*f^4 + 896*C*a^2*c^6*d^15*f^4 + 3136*C*a^2*c^8*d^13*f^4 +
4928*C*a^2*c^10*d^11*f^4 + 4480*C*a^2*c^12*d^9*f^4 + 2432*C*a^2*c^14*d^7*f^
4 + 736*C*a^2*c^16*d^5*f^4 + 96*C*a^2*c^18*d^3*f^4 + 160*C*b^2*c^2*d^19*f^4
 + 128*C*b^2*c^4*d^17*f^4 - 896*C*b^2*c^6*d^15*f^4 - 3136*C*b^2*c^8*d^13*f^
4 - 4928*C*b^2*c^10*d^11*f^4 - 4480*C*b^2*c^12*d^9*f^4 - 2432*C*b^2*c^14*d^
7*f^4 - 736*C*b^2*c^16*d^5*f^4 - 96*C*b^2*c^18*d^3*f^4 + 192*C*a*b*c*d^20*f
^4 + 1472*C*a*b*c^3*d^18*f^4 + 4864*C*a*b*c^5*d^16*f^4 + 8960*C*a*b*c^7*d^1
4*f^4 + 9856*C*a*b*c^9*d^12*f^4 + 6272*C*a*b*c^11*d^10*f^4 + 1792*C*a*b*c^1
3*d^8*f^4 - 256*C*a*b*c^15*d^6*f^4 - 320*C*a*b*c^17*d^4*f^4 - 64*C*a*b*c^19
*d^2*f^4) + (c + d*tan(e + f*x))^(1/2)*(96*C^2*a^2*b^2*d^18*f^3 - 16*C^2*b^
4*d^18*f^3 - 16*C^2*a^4*d^18*f^3 + 320*C^2*a^4*c^4*d^14*f^3 + 1024*C^2*a^4*
c^6*d^12*f^3 + 1440*C^2*a^4*c^8*d^10*f^3 + 1024*C^2*a^4*c^10*d^8*f^3 + 320*
C^2*a^4*c^12*d^6*f^3 - 16*C^2*a^4*c^16*d^2*f^3 + 320*C^2*b^4*c^4*d^14*f^3 +
 1024*C^2*b^4*c^6*d^12*f^3 + 1440*C^2*b^4*c^8*d^10*f^3 + 1024*C^2*b^4*c^10*
d^8*f^3 + 320*C^2*b^4*c^12*d^6*f^3 - 16*C^2*b^4*c^16*d^2*f^3 - 256*C^2*a*b^
3*c*d^17*f^3 + 256*C^2*a^3*b*c*d^17*f^3 - 1280*C^2*a*b^3*c^3*d^15*f^3 - 230
4*C^2*a*b^3*c^5*d^13*f^3 - 1280*C^2*a*b^3*c^7*d^11*f^3 + 1280*C^2*a*b^3*c^9
*d^9*f^3 + 2304*C^2*a*b^3*c^11*d^7*f^3 + 1280*C^2*a*b^3*c^13*d^5*f^3 + 256*
C^2*a*b^3*c^15*d^3*f^3 + 1280*C^2*a^3*b*c^3*d^15*f^3 + 2304*C^2*a^3*b*c^5*d
^13*f^3 + 1280*C^2*a^3*b*c^7*d^11*f^3 - 1280*C^2*a^3*b*c^9*d^9*f^3 - 2304*C
^2*a^3*b*c^11*d^7*f^3 - 1280*C^2*a^3*b*c^13*d^5*f^3 - 256*C^2*a^3*b*c^15*d^
3*f^3 - 1920*C^2*a^2*b^2*c^4*d^14*f^3 - 6144*C^2*a^2*b^2*c^6*d^12*f^3 - 864
0*C^2*a^2*b^2*c^8*d^10*f^3 - 6144*C^2*a^2*b^2*c^10*d^8*f^3 - 1920*C^2*a^2*b
^2*c^12*d^6*f^3 + 96*C^2*a^2*b^2*c^16*d^2*f^3))*(-(((8*C^2*a^4*c^5*f^2 + 8*
C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*
b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*

c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*
c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2
*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8
+ 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4
+ 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1
/2) + 4*C^2*a^4*c^5*f^2 + 4*C^2*b^4*c^5*f^2 - 24*C^2*a^2*b^2*c^5*f^2 - 40*C
^2*a^4*c^3*d^2*f^2 - 40*C^2*b^4*c^3*d^2*f^2 - 16*C^2*a*b^3*d^5*f^2 + 16*C^2
*a^3*b*d^5*f^2 + 20*C^2*a^4*c*d^4*f^2 + 20*C^2*b^4*c*d^4*f^2 - 80*C^2*a*b^3
*c^4*d*f^2 + 80*C^2*a^3*b*c^4*d*f^2 + 160*C^2*a*b^3*c^2*d^3*f^2 - 120*C^2*a
^2*b^2*c*d^4*f^2 - 160*C^2*a^3*b*c^2*d^3*f^2 + 240*C^2*a^2*b^2*c^3*d^2*f^2)
/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4
+ 5*c^8*d^2*f^4)))^(1/2) - 64*C^3*a^3*b^3*d^16*f^2 - 192*C^3*a^6*c^3*d^13*
f^2 - 480*C^3*a^6*c^5*d^11*f^2 - 640*C^3*a^6*c^7*d^9*f^2 - 480*C^3*a^6*c^9*
d^7*f^2 - 192*C^3*a^6*c^11*d^5*f^2 - 32*C^3*a^6*c^13*d^3*f^2 + 192*C^3*b^6*
c^3*d^13*f^2 + 480*C^3*b^6*c^5*d^11*f^2 + 640*C^3*b^6*c^7*d^9*f^2 + 480*C^3
*b^6*c^9*d^7*f^2 + 192*C^3*b^6*c^11*d^5*f^2 + 32*C^3*b^6*c^13*d^3*f^2 - 32*
C^3*a*b^5*d^16*f^2 - 32*C^3*a^5*b*d^16*f^2 - 32*C^3*a^6*c*d^15*f^2 + 32*C^3
*b^6*c*d^15*f^2 - 160*C^3*a*b^5*c^2*d^14*f^2 - 288*C^3*a*b^5*c^4*d^12*f^2 -
160*C^3*a*b^5*c^6*d^10*f^2 + 160*C^3*a*b^5*c^8*d^8*f^2 + 288*C^3*a*b^5*c^1
0*d^6*f^2 + 160*C^3*a*b^5*c^12*d^4*f^2 + 32*C^3*a*b^5*c^14*d^2*f^2 + 32*C^3
*a^2*b^4*c*d^15*f^2 - 32*C^3*a^4*b^2*c*d^15*f^2 - 160*C^3*a^5*b*c^2*d^14*f^
2 - 288*C^3*a^5*b*c^4*d^12*f^2 - 160*C^3*a^5*b*c^6*d^10*f^2 + 160*C^3*a^5*b
*c^8*d^8*f^2 + 288*C^3*a^5*b*c^10*d^6*f^2 + 160*C^3*a^5*b*c^12*d^4*f^2 + 32
*C^3*a^5*b*c^14*d^2*f^2 + 192*C^3*a^2*b^4*c^3*d^13*f^2 + 480*C^3*a^2*b^4*c^
5*d^11*f^2 + 640*C^3*a^2*b^4*c^7*d^9*f^2 + 480*C^3*a^2*b^4*c^9*d^7*f^2 + 19
2*C^3*a^2*b^4*c^11*d^5*f^2 + 32*C^3*a^2*b^4*c^13*d^3*f^2 - 320*C^3*a^3*b^3*
c^2*d^14*f^2 - 576*C^3*a^3*b^3*c^4*d^12*f^2 - 320*C^3*a^3*b^3*c^6*d^10*f^2
+ 320*C^3*a^3*b^3*c^8*d^8*f^2 + 576*C^3*a^3*b^3*c^10*d^6*f^2 + 320*C^3*a^3*
b^3*c^12*d^4*f^2 + 64*C^3*a^3*b^3*c^14*d^2*f^2 - 192*C^3*a^4*b^2*c^3*d^13*f
^2 - 480*C^3*a^4*b^2*c^5*d^11*f^2 - 640*C^3*a^4*b^2*c^7*d^9*f^2 - 480*C^3*a
^4*b^2*c^9*d^7*f^2 - 192*C^3*a^4*b^2*c^11*d^5*f^2 - 32*C^3*a^4*b^2*c^13*d^3
f^2))(-(((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2
- 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 +
32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 - 160*C^
2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 2
40*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*
d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a
^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160
*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*C^2*a^4*c^5*f^2 + 4*C^2*b^4*c^5*f
^2 - 24*C^2*a^2*b^2*c^5*f^2 - 40*C^2*a^4*c^3*d^2*f^2 - 40*C^2*b^4*c^3*d^2*f
^2 - 16*C^2*a*b^3*d^5*f^2 + 16*C^2*a^3*b*d^5*f^2 + 20*C^2*a^4*c*d^4*f^2 + 2
0*C^2*b^4*c*d^4*f^2 - 80*C^2*a*b^3*c^4*d*f^2 + 80*C^2*a^3*b*c^4*d*f^2 + 160
*C^2*a*b^3*c^2*d^3*f^2 - 120*C^2*a^2*b^2*c*d^4*f^2 - 160*C^2*a^3*b*c^2*d^3*
f^2 + 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4
+ 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*2i - atan((((-(

((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a
^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3
*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^
4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2
*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2
/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(1
6*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f
^4 + 80*c^8*d^2*f^4))^(1/2) - 4*B^2*a^4*c^5*f^2 - 4*B^2*b^4*c^5*f^2 + 24*B^
2*a^2*b^2*c^5*f^2 + 40*B^2*a^4*c^3*d^2*f^2 + 40*B^2*b^4*c^3*d^2*f^2 + 16*B^
2*a*b^3*d^5*f^2 - 16*B^2*a^3*b*d^5*f^2 - 20*B^2*a^4*c*d^4*f^2 - 20*B^2*b^4*
c*d^4*f^2 + 80*B^2*a*b^3*c^4*d*f^2 - 80*B^2*a^3*b*c^4*d*f^2 - 160*B^2*a*b^3
*c^2*d^3*f^2 + 120*B^2*a^2*b^2*c*d^4*f^2 + 160*B^2*a^3*b*c^2*d^3*f^2 - 240*
B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*
d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/
2)*(-(((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80
*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B
^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*
b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B
^2*a^2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*
f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b
^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6
*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*B^2*a^4*c^5*f^2 - 4*B^2*b^4*c^5*f^2 +
24*B^2*a^2*b^2*c^5*f^2 + 40*B^2*a^4*c^3*d^2*f^2 + 40*B^2*b^4*c^3*d^2*f^2 +
16*B^2*a*b^3*d^5*f^2 - 16*B^2*a^3*b*d^5*f^2 - 20*B^2*a^4*c*d^4*f^2 - 20*B^
2*b^4*c*d^4*f^2 + 80*B^2*a*b^3*c^4*d*f^2 - 80*B^2*a^3*b*c^4*d*f^2 - 160*B^2
*a*b^3*c^2*d^3*f^2 + 120*B^2*a^2*b^2*c*d^4*f^2 + 160*B^2*a^3*b*c^2*d^3*f^2
- 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 1
0*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 64
0*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5
+ 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^1
7*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 96*B*a^2*c*d^20*f^4 + 96*
B*b^2*c*d^20*f^4 - 736*B*a^2*c^3*d^18*f^4 - 2432*B*a^2*c^5*d^16*f^4 - 4480*
B*a^2*c^7*d^14*f^4 - 4928*B*a^2*c^9*d^12*f^4 - 3136*B*a^2*c^11*d^10*f^4 - 8
96*B*a^2*c^13*d^8*f^4 + 128*B*a^2*c^15*d^6*f^4 + 160*B*a^2*c^17*d^4*f^4 + 3
2*B*a^2*c^19*d^2*f^4 + 736*B*b^2*c^3*d^18*f^4 + 2432*B*b^2*c^5*d^16*f^4 + 4
480*B*b^2*c^7*d^14*f^4 + 4928*B*b^2*c^9*d^12*f^4 + 3136*B*b^2*c^11*d^10*f^4
+ 896*B*b^2*c^13*d^8*f^4 - 128*B*b^2*c^15*d^6*f^4 - 160*B*b^2*c^17*d^4*f^4
- 32*B*b^2*c^19*d^2*f^4 - 64*B*a*b*d^21*f^4 - 320*B*a*b*c^2*d^19*f^4 - 256
*B*a*b*c^4*d^17*f^4 + 1792*B*a*b*c^6*d^15*f^4 + 6272*B*a*b*c^8*d^13*f^4 + 9
856*B*a*b*c^10*d^11*f^4 + 8960*B*a*b*c^12*d^9*f^4 + 4864*B*a*b*c^14*d^7*f^4
+ 1472*B*a*b*c^16*d^5*f^4 + 192*B*a*b*c^18*d^3*f^4) + (c + d*tan(e + f*x))
^(1/2)*(96*B^2*a^2*b^2*d^18*f^3 - 16*B^2*b^4*d^18*f^3 - 16*B^2*a^4*d^18*f^3
+ 320*B^2*a^4*c^4*d^14*f^3 + 1024*B^2*a^4*c^6*d^12*f^3 + 1440*B^2*a^4*c^8*
d^10*f^3 + 1024*B^2*a^4*c^10*d^8*f^3 + 320*B^2*a^4*c^12*d^6*f^3 - 16*B^2*a^
4*c^16*d^2*f^3 + 320*B^2*b^4*c^4*d^14*f^3 + 1024*B^2*b^4*c^6*d^12*f^3 + 144

0*B^2*b^4*c^8*d^10*f^3 + 1024*B^2*b^4*c^10*d^8*f^3 + 320*B^2*b^4*c^12*d^6*f
^3 - 16*B^2*b^4*c^16*d^2*f^3 - 256*B^2*a*b^3*c*d^17*f^3 + 256*B^2*a^3*b*c*d
^17*f^3 - 1280*B^2*a*b^3*c^3*d^15*f^3 - 2304*B^2*a*b^3*c^5*d^13*f^3 - 1280*
B^2*a*b^3*c^7*d^11*f^3 + 1280*B^2*a*b^3*c^9*d^9*f^3 + 2304*B^2*a*b^3*c^11*d
^7*f^3 + 1280*B^2*a*b^3*c^13*d^5*f^3 + 256*B^2*a*b^3*c^15*d^3*f^3 + 1280*B^
2*a^3*b*c^3*d^15*f^3 + 2304*B^2*a^3*b*c^5*d^13*f^3 + 1280*B^2*a^3*b*c^7*d^1
1*f^3 - 1280*B^2*a^3*b*c^9*d^9*f^3 - 2304*B^2*a^3*b*c^11*d^7*f^3 - 1280*B^2
*a^3*b*c^13*d^5*f^3 - 256*B^2*a^3*b*c^15*d^3*f^3 - 1920*B^2*a^2*b^2*c^4*d^1
4*f^3 - 6144*B^2*a^2*b^2*c^6*d^12*f^3 - 8640*B^2*a^2*b^2*c^8*d^10*f^3 - 614
4*B^2*a^2*b^2*c^10*d^8*f^3 - 1920*B^2*a^2*b^2*c^12*d^6*f^3 + 96*B^2*a^2*b^2
*c^16*d^2*f^3))*(-(((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2
*c^5*f^2 - 80*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d
^5*f^2 + 32*B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2
 - 160*B^2*a*b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^
3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2
*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4
 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*
f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*B^2*a^4*c^5*f^2 - 4*B^2*
b^4*c^5*f^2 + 24*B^2*a^2*b^2*c^5*f^2 + 40*B^2*a^4*c^3*d^2*f^2 + 40*B^2*b^4*
c^3*d^2*f^2 + 16*B^2*a*b^3*d^5*f^2 - 16*B^2*a^3*b*d^5*f^2 - 20*B^2*a^4*c*d^
4*f^2 - 20*B^2*b^4*c*d^4*f^2 + 80*B^2*a*b^3*c^4*d*f^2 - 80*B^2*a^3*b*c^4*d*
f^2 - 160*B^2*a*b^3*c^2*d^3*f^2 + 120*B^2*a^2*b^2*c*d^4*f^2 + 160*B^2*a^3*b
*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^
2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*1i - (
(-(((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^
2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*
a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3
*c^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*
a^2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2
)^2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)
*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^
4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*B^2*a^4*c^5*f^2 - 4*B^2*b^4*c^5*f^2 + 24
*B^2*a^2*b^2*c^5*f^2 + 40*B^2*a^4*c^3*d^2*f^2 + 40*B^2*b^4*c^3*d^2*f^2 + 16
*B^2*a*b^3*d^5*f^2 - 16*B^2*a^3*b*d^5*f^2 - 20*B^2*a^4*c*d^4*f^2 - 20*B^2*b
^4*c*d^4*f^2 + 80*B^2*a*b^3*c^4*d*f^2 - 80*B^2*a^3*b*c^4*d*f^2 - 160*B^2*a*
b^3*c^2*d^3*f^2 + 120*B^2*a^2*b^2*c*d^4*f^2 + 160*B^2*a^3*b*c^2*d^3*f^2 - 2
40*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c
^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(96*B*b^2*c*d^20*f^4 -
 96*B*a^2*c*d^20*f^4 - (c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^4*c^5*f^2 +
8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2*f^2 - 80*B^
2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2 + 40*B^2*a^
4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^2 + 160*B^2*a^3*
b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 320*B
^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4*b^
8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f

^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^
(1/2) - 4*B^2*a^4*c^5*f^2 - 4*B^2*b^4*c^5*f^2 + 24*B^2*a^2*b^2*c^5*f^2 + 40
*B^2*a^4*c^3*d^2*f^2 + 40*B^2*b^4*c^3*d^2*f^2 + 16*B^2*a*b^3*d^5*f^2 - 16*B
^2*a^3*b*d^5*f^2 - 20*B^2*a^4*c*d^4*f^2 - 20*B^2*b^4*c*d^4*f^2 + 80*B^2*a*b
^3*c^4*d*f^2 - 80*B^2*a^3*b*c^4*d*f^2 - 160*B^2*a*b^3*c^2*d^3*f^2 + 120*B^2
*a^2*b^2*c*d^4*f^2 + 160*B^2*a^3*b*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c^3*d^2*f^
2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f
^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^
18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 134
40*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5
 + 64*c^21*d^2*f^5) - 736*B*a^2*c^3*d^18*f^4 - 2432*B*a^2*c^5*d^16*f^4 - 44
80*B*a^2*c^7*d^14*f^4 - 4928*B*a^2*c^9*d^12*f^4 - 3136*B*a^2*c^11*d^10*f^4
- 896*B*a^2*c^13*d^8*f^4 + 128*B*a^2*c^15*d^6*f^4 + 160*B*a^2*c^17*d^4*f^4
+ 32*B*a^2*c^19*d^2*f^4 + 736*B*b^2*c^3*d^18*f^4 + 2432*B*b^2*c^5*d^16*f^4
+ 4480*B*b^2*c^7*d^14*f^4 + 4928*B*b^2*c^9*d^12*f^4 + 3136*B*b^2*c^11*d^10*
f^4 + 896*B*b^2*c^13*d^8*f^4 - 128*B*b^2*c^15*d^6*f^4 - 160*B*b^2*c^17*d^4*
f^4 - 32*B*b^2*c^19*d^2*f^4 - 64*B*a*b*d^21*f^4 - 320*B*a*b*c^2*d^19*f^4 -
256*B*a*b*c^4*d^17*f^4 + 1792*B*a*b*c^6*d^15*f^4 + 6272*B*a*b*c^8*d^13*f^4
+ 9856*B*a*b*c^10*d^11*f^4 + 8960*B*a*b*c^12*d^9*f^4 + 4864*B*a*b*c^14*d^7*
f^4 + 1472*B*a*b*c^16*d^5*f^4 + 192*B*a*b*c^18*d^3*f^4) - (c + d*tan(e + f*
x))^(1/2)*(96*B^2*a^2*b^2*d^18*f^3 - 16*B^2*b^4*d^18*f^3 - 16*B^2*a^4*d^18*
f^3 + 320*B^2*a^4*c^4*d^14*f^3 + 1024*B^2*a^4*c^6*d^12*f^3 + 1440*B^2*a^4*c
^8*d^10*f^3 + 1024*B^2*a^4*c^10*d^8*f^3 + 320*B^2*a^4*c^12*d^6*f^3 - 16*B^2
*a^4*c^16*d^2*f^3 + 320*B^2*b^4*c^4*d^14*f^3 + 1024*B^2*b^4*c^6*d^12*f^3 +
1440*B^2*b^4*c^8*d^10*f^3 + 1024*B^2*b^4*c^10*d^8*f^3 + 320*B^2*b^4*c^12*d^
6*f^3 - 16*B^2*b^4*c^16*d^2*f^3 - 256*B^2*a*b^3*c*d^17*f^3 + 256*B^2*a^3*b*
c*d^17*f^3 - 1280*B^2*a*b^3*c^3*d^15*f^3 - 2304*B^2*a*b^3*c^5*d^13*f^3 - 12
80*B^2*a*b^3*c^7*d^11*f^3 + 1280*B^2*a*b^3*c^9*d^9*f^3 + 2304*B^2*a*b^3*c^1
1*d^7*f^3 + 1280*B^2*a*b^3*c^13*d^5*f^3 + 256*B^2*a*b^3*c^15*d^3*f^3 + 1280
*B^2*a^3*b*c^3*d^15*f^3 + 2304*B^2*a^3*b*c^5*d^13*f^3 + 1280*B^2*a^3*b*c^7*
d^11*f^3 - 1280*B^2*a^3*b*c^9*d^9*f^3 - 2304*B^2*a^3*b*c^11*d^7*f^3 - 1280*
B^2*a^3*b*c^13*d^5*f^3 - 256*B^2*a^3*b*c^15*d^3*f^3 - 1920*B^2*a^2*b^2*c^4*
d^14*f^3 - 6144*B^2*a^2*b^2*c^6*d^12*f^3 - 8640*B^2*a^2*b^2*c^8*d^10*f^3 -
6144*B^2*a^2*b^2*c^10*d^8*f^3 - 1920*B^2*a^2*b^2*c^12*d^6*f^3 + 96*B^2*a^2*
b^2*c^16*d^2*f^3))*(-(((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*
b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^
3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*
f^2 - 160*B^2*a*b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2
*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*
a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b
^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d
^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*B^2*a^4*c^5*f^2 - 4*B
^2*b^4*c^5*f^2 + 24*B^2*a^2*b^2*c^5*f^2 + 40*B^2*a^4*c^3*d^2*f^2 + 40*B^2*b
^4*c^3*d^2*f^2 + 16*B^2*a*b^3*d^5*f^2 - 16*B^2*a^3*b*d^5*f^2 - 20*B^2*a^4*c
*d^4*f^2 - 20*B^2*b^4*c*d^4*f^2 + 80*B^2*a*b^3*c^4*d*f^2 - 80*B^2*a^3*b*c^4

*d*f^2 - 160*B^2*a*b^3*c^2*d^3*f^2 + 120*B^2*a^2*b^2*c*d^4*f^2 + 160*B^2*a^3*b*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*1i)/(16*B^3*b^6*d^16*f^2 - ((-(((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*B^2*a^4*c^5*f^2 - 4*B^2*b^4*c^5*f^2 + 24*B^2*a^2*b^2*c^5*f^2 + 40*B^2*a^4*c^3*d^2*f^2 + 40*B^2*b^4*c^3*d^2*f^2 + 16*B^2*a*b^3*d^5*f^2 - 16*B^2*a^3*b*d^5*f^2 - 20*B^2*a^4*c*d^4*f^2 - 20*B^2*b^4*c*d^4*f^2 + 80*B^2*a*b^3*c^4*d*f^2 - 80*B^2*a^3*b*c^4*d*f^2 - 160*B^2*a*b^3*c^2*d^3*f^2 + 120*B^2*a^2*b^2*c*d^4*f^2 + 160*B^2*a^3*b*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(96*B*b^2*c*d^20*f^4 - 96*B*a^2*c*d^20*f^4 - (c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*B^2*a^4*c^5*f^2 - 4*B^2*b^4*c^5*f^2 + 24*B^2*a^2*b^2*c^5*f^2 + 40*B^2*a^4*c^3*d^2*f^2 + 40*B^2*b^4*c^3*d^2*f^2 + 16*B^2*a*b^3*d^5*f^2 - 16*B^2*a^3*b*d^5*f^2 - 20*B^2*a^4*c*d^4*f^2 - 20*B^2*b^4*c*d^4*f^2 + 80*B^2*a*b^3*c^4*d*f^2 - 80*B^2*a^3*b*c^4*d*f^2 - 160*B^2*a*b^3*c^2*d^3*f^2 + 120*B^2*a^2*b^2*c*d^4*f^2 + 160*B^2*a^3*b*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 736*B*a^2*c^3*d^18*f^4 - 2432*B*a^2*c^5*d^16*f^4 - 4480*B*a^2*c^7*d^14*f^4 - 4928*B*a^2*c^9*d^12*f^4 - 3136*B*a^2*c^11*d^10*f^4 - 896*B*a^2*c^13*d^8*f^4 + 128*B*a^2*c^15*d^6*f^4 + 160*B*a^2*c^17*d^4*f^4 + 32*B*a^2*c^19*d^2*f^4 + 736*B*b^2*c^3*d^18*f^4 + 2432*B*b^2*c^5*d^16*f^4 + 4480*B*b^2*c^7*d^14*f^4 + 4928*B*b^2*c^9*d^12*f^4 + 3136*B*b^2*c^11*d^10*f^4 + 896*B*b^2*c^13*d^8*f^4 - 128*B*b^2*c^15*d^6*f^4 - 160*B*b^2*c^17*d^4*f^4 - 32*B*b^2*c^19*d^2*f^4 - 64*B*a*b*d^21*f^4 - 320*B*a*b*c^2*d^19*f^4 - 256*B*a*b*c^4*d^17*f^4 + 1792*B*a*b*c^6*d^15*f^4 + 6272*B*a*b*c^8*d^13*f^4 + 9856*B*a*b*c^10*d^11*f^4 + 8960*B*a*b*c^12*d^9*f^4 + 4864*B*a*b*c^14*d^7*f^4 + 1472*B*a*b*c^16*d^5*f^4 + 192*B*a*b*c^18*d^3*f^4) - (c + d*tan(e + f*x))^(1/2)*(96*B^2*a^2*b^2*d^18*f^3 - 16*B^2*b^4*d^

18*f^3 - 16*B^2*a^4*d^18*f^3 + 320*B^2*a^4*c^4*d^14*f^3 + 1024*B^2*a^4*c^6*
d^12*f^3 + 1440*B^2*a^4*c^8*d^10*f^3 + 1024*B^2*a^4*c^10*d^8*f^3 + 320*B^2*
a^4*c^12*d^6*f^3 - 16*B^2*a^4*c^16*d^2*f^3 + 320*B^2*b^4*c^4*d^14*f^3 + 102
4*B^2*b^4*c^6*d^12*f^3 + 1440*B^2*b^4*c^8*d^10*f^3 + 1024*B^2*b^4*c^10*d^8*
f^3 + 320*B^2*b^4*c^12*d^6*f^3 - 16*B^2*b^4*c^16*d^2*f^3 - 256*B^2*a*b^3*c*
d^17*f^3 + 256*B^2*a^3*b*c*d^17*f^3 - 1280*B^2*a*b^3*c^3*d^15*f^3 - 2304*B^
2*a*b^3*c^5*d^13*f^3 - 1280*B^2*a*b^3*c^7*d^11*f^3 + 1280*B^2*a*b^3*c^9*d^9
*f^3 + 2304*B^2*a*b^3*c^11*d^7*f^3 + 1280*B^2*a*b^3*c^13*d^5*f^3 + 256*B^2*
a*b^3*c^15*d^3*f^3 + 1280*B^2*a^3*b*c^3*d^15*f^3 + 2304*B^2*a^3*b*c^5*d^13*
f^3 + 1280*B^2*a^3*b*c^7*d^11*f^3 - 1280*B^2*a^3*b*c^9*d^9*f^3 - 2304*B^2*a
^3*b*c^11*d^7*f^3 - 1280*B^2*a^3*b*c^13*d^5*f^3 - 256*B^2*a^3*b*c^15*d^3*f^
3 - 1920*B^2*a^2*b^2*c^4*d^14*f^3 - 6144*B^2*a^2*b^2*c^6*d^12*f^3 - 8640*B^
2*a^2*b^2*c^8*d^10*f^3 - 6144*B^2*a^2*b^2*c^10*d^8*f^3 - 1920*B^2*a^2*b^2*c
^12*d^6*f^3 + 96*B^2*a^2*b^2*c^16*d^2*f^3))*(-(((8*B^2*a^4*c^5*f^2 + 8*B^2*
b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*
c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^
4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4*
d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 320*B^2*a^3
*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4*
B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 8
0*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2)
- 4*B^2*a^4*c^5*f^2 - 4*B^2*b^4*c^5*f^2 + 24*B^2*a^2*b^2*c^5*f^2 + 40*B^2*a
^4*c^3*d^2*f^2 + 40*B^2*b^4*c^3*d^2*f^2 + 16*B^2*a*b^3*d^5*f^2 - 16*B^2*a^3
*b*d^5*f^2 - 20*B^2*a^4*c*d^4*f^2 - 20*B^2*b^4*c*d^4*f^2 + 80*B^2*a*b^3*c^4
*d*f^2 - 80*B^2*a^3*b*c^4*d*f^2 - 160*B^2*a*b^3*c^2*d^3*f^2 + 120*B^2*a^2*b
^2*c*d^4*f^2 + 160*B^2*a^3*b*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16
*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5
*c^8*d^2*f^4)))^(1/2) - 16*B^3*a^6*d^16*f^2 - ((-(((8*B^2*a^4*c^5*f^2 + 8*B
^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b
^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c
*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^2 + 160*B^2*a^3*b*c
^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 320*B^2*
a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4*b^8 +
4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4
+ 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/
2) - 4*B^2*a^4*c^5*f^2 - 4*B^2*b^4*c^5*f^2 + 24*B^2*a^2*b^2*c^5*f^2 + 40*B^
2*a^4*c^3*d^2*f^2 + 40*B^2*b^4*c^3*d^2*f^2 + 16*B^2*a*b^3*d^5*f^2 - 16*B^2*
a^3*b*d^5*f^2 - 20*B^2*a^4*c*d^4*f^2 - 20*B^2*b^4*c*d^4*f^2 + 80*B^2*a*b^3*
c^4*d*f^2 - 80*B^2*a^3*b*c^4*d*f^2 - 160*B^2*a*b^3*c^2*d^3*f^2 + 120*B^2*a^
2*b^2*c*d^4*f^2 + 160*B^2*a^3*b*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c^3*d^2*f^2)/
(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4
+ 5*c^8*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^4*c^5*f^2
+ 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2*f^2 - 80
*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2 + 40*B^2
*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^2 + 160*B^2*a

^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 32
0*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4
*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^1
0*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4
))^(1/2) - 4*B^2*a^4*c^5*f^2 - 4*B^2*b^4*c^5*f^2 + 24*B^2*a^2*b^2*c^5*f^2 +
40*B^2*a^4*c^3*d^2*f^2 + 40*B^2*b^4*c^3*d^2*f^2 + 16*B^2*a*b^3*d^5*f^2 - 1
6*B^2*a^3*b*d^5*f^2 - 20*B^2*a^4*c*d^4*f^2 - 20*B^2*b^4*c*d^4*f^2 + 80*B^2*
a*b^3*c^4*d*f^2 - 80*B^2*a^3*b*c^4*d*f^2 - 160*B^2*a*b^3*c^2*d^3*f^2 + 120*
B^2*a^2*b^2*c*d^4*f^2 + 160*B^2*a^3*b*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c^3*d^2
f^2)/(16(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^
4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5
*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 +
13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*
f^5 + 64*c^21*d^2*f^5) - 96*B*a^2*c*d^20*f^4 + 96*B*b^2*c*d^20*f^4 - 736*B*
a^2*c^3*d^18*f^4 - 2432*B*a^2*c^5*d^16*f^4 - 4480*B*a^2*c^7*d^14*f^4 - 4928
*B*a^2*c^9*d^12*f^4 - 3136*B*a^2*c^11*d^10*f^4 - 896*B*a^2*c^13*d^8*f^4 + 1
28*B*a^2*c^15*d^6*f^4 + 160*B*a^2*c^17*d^4*f^4 + 32*B*a^2*c^19*d^2*f^4 + 73
6*B*b^2*c^3*d^18*f^4 + 2432*B*b^2*c^5*d^16*f^4 + 4480*B*b^2*c^7*d^14*f^4 +
4928*B*b^2*c^9*d^12*f^4 + 3136*B*b^2*c^11*d^10*f^4 + 896*B*b^2*c^13*d^8*f^4
- 128*B*b^2*c^15*d^6*f^4 - 160*B*b^2*c^17*d^4*f^4 - 32*B*b^2*c^19*d^2*f^4
- 64*B*a*b*d^21*f^4 - 320*B*a*b*c^2*d^19*f^4 - 256*B*a*b*c^4*d^17*f^4 + 179
2*B*a*b*c^6*d^15*f^4 + 6272*B*a*b*c^8*d^13*f^4 + 9856*B*a*b*c^10*d^11*f^4 +
8960*B*a*b*c^12*d^9*f^4 + 4864*B*a*b*c^14*d^7*f^4 + 1472*B*a*b*c^16*d^5*f^
4 + 192*B*a*b*c^18*d^3*f^4) + (c + d*tan(e + f*x))^(1/2)*(96*B^2*a^2*b^2*d^
18*f^3 - 16*B^2*b^4*d^18*f^3 - 16*B^2*a^4*d^18*f^3 + 320*B^2*a^4*c^4*d^14*f
^3 + 1024*B^2*a^4*c^6*d^12*f^3 + 1440*B^2*a^4*c^8*d^10*f^3 + 1024*B^2*a^4*c
^10*d^8*f^3 + 320*B^2*a^4*c^12*d^6*f^3 - 16*B^2*a^4*c^16*d^2*f^3 + 320*B^2*
b^4*c^4*d^14*f^3 + 1024*B^2*b^4*c^6*d^12*f^3 + 1440*B^2*b^4*c^8*d^10*f^3 +
1024*B^2*b^4*c^10*d^8*f^3 + 320*B^2*b^4*c^12*d^6*f^3 - 16*B^2*b^4*c^16*d^2*
f^3 - 256*B^2*a*b^3*c*d^17*f^3 + 256*B^2*a^3*b*c*d^17*f^3 - 1280*B^2*a*b^3*
c^3*d^15*f^3 - 2304*B^2*a*b^3*c^5*d^13*f^3 - 1280*B^2*a*b^3*c^7*d^11*f^3 +
1280*B^2*a*b^3*c^9*d^9*f^3 + 2304*B^2*a*b^3*c^11*d^7*f^3 + 1280*B^2*a*b^3*c
^13*d^5*f^3 + 256*B^2*a*b^3*c^15*d^3*f^3 + 1280*B^2*a^3*b*c^3*d^15*f^3 + 23
04*B^2*a^3*b*c^5*d^13*f^3 + 1280*B^2*a^3*b*c^7*d^11*f^3 - 1280*B^2*a^3*b*c^
9*d^9*f^3 - 2304*B^2*a^3*b*c^11*d^7*f^3 - 1280*B^2*a^3*b*c^13*d^5*f^3 - 256
*B^2*a^3*b*c^15*d^3*f^3 - 1920*B^2*a^2*b^2*c^4*d^14*f^3 - 6144*B^2*a^2*b^2*
c^6*d^12*f^3 - 8640*B^2*a^2*b^2*c^8*d^10*f^3 - 6144*B^2*a^2*b^2*c^10*d^8*f^
3 - 1920*B^2*a^2*b^2*c^12*d^6*f^3 + 96*B^2*a^2*b^2*c^16*d^2*f^3))*(-(((8*B^
2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3
*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5
*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^
2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c
*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (
B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10
*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 8

0*c^8*d^2*f^4))^(1/2) - 4*B^2*a^4*c^5*f^2 - 4*B^2*b^4*c^5*f^2 + 24*B^2*a^2*
b^2*c^5*f^2 + 40*B^2*a^4*c^3*d^2*f^2 + 40*B^2*b^4*c^3*d^2*f^2 + 16*B^2*a*b^
3*d^5*f^2 - 16*B^2*a^3*b*d^5*f^2 - 20*B^2*a^4*c*d^4*f^2 - 20*B^2*b^4*c*d^4*
f^2 + 80*B^2*a*b^3*c^4*d*f^2 - 80*B^2*a^3*b*c^4*d*f^2 - 160*B^2*a*b^3*c^2*d
^3*f^2 + 120*B^2*a^2*b^2*c*d^4*f^2 + 160*B^2*a^3*b*c^2*d^3*f^2 - 240*B^2*a^
2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^
4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2) + 16*B^3*a^2*b^4*d^16*f^2 - 16*
B^3*a^4*b^2*d^16*f^2 - 80*B^3*a^6*c^2*d^14*f^2 - 144*B^3*a^6*c^4*d^12*f^2 -
80*B^3*a^6*c^6*d^10*f^2 + 80*B^3*a^6*c^8*d^8*f^2 + 144*B^3*a^6*c^10*d^6*f^
2 + 80*B^3*a^6*c^12*d^4*f^2 + 16*B^3*a^6*c^14*d^2*f^2 + 80*B^3*b^6*c^2*d^14
*f^2 + 144*B^3*b^6*c^4*d^12*f^2 + 80*B^3*b^6*c^6*d^10*f^2 - 80*B^3*b^6*c^8*
d^8*f^2 - 144*B^3*b^6*c^10*d^6*f^2 - 80*B^3*b^6*c^12*d^4*f^2 - 16*B^3*b^6*c
^14*d^2*f^2 + 64*B^3*a*b^5*c*d^15*f^2 + 64*B^3*a^5*b*c*d^15*f^2 + 384*B^3*a
*b^5*c^3*d^13*f^2 + 960*B^3*a*b^5*c^5*d^11*f^2 + 1280*B^3*a*b^5*c^7*d^9*f^2
+ 960*B^3*a*b^5*c^9*d^7*f^2 + 384*B^3*a*b^5*c^11*d^5*f^2 + 64*B^3*a*b^5*c^
13*d^3*f^2 + 128*B^3*a^3*b^3*c*d^15*f^2 + 384*B^3*a^5*b*c^3*d^13*f^2 + 960*
B^3*a^5*b*c^5*d^11*f^2 + 1280*B^3*a^5*b*c^7*d^9*f^2 + 960*B^3*a^5*b*c^9*d^7
*f^2 + 384*B^3*a^5*b*c^11*d^5*f^2 + 64*B^3*a^5*b*c^13*d^3*f^2 + 80*B^3*a^2*
b^4*c^2*d^14*f^2 + 144*B^3*a^2*b^4*c^4*d^12*f^2 + 80*B^3*a^2*b^4*c^6*d^10*f
^2 - 80*B^3*a^2*b^4*c^8*d^8*f^2 - 144*B^3*a^2*b^4*c^10*d^6*f^2 - 80*B^3*a^2
*b^4*c^12*d^4*f^2 - 16*B^3*a^2*b^4*c^14*d^2*f^2 + 768*B^3*a^3*b^3*c^3*d^13*
f^2 + 1920*B^3*a^3*b^3*c^5*d^11*f^2 + 2560*B^3*a^3*b^3*c^7*d^9*f^2 + 1920*B
^3*a^3*b^3*c^9*d^7*f^2 + 768*B^3*a^3*b^3*c^11*d^5*f^2 + 128*B^3*a^3*b^3*c^1
3*d^3*f^2 - 80*B^3*a^4*b^2*c^2*d^14*f^2 - 144*B^3*a^4*b^2*c^4*d^12*f^2 - 80
*B^3*a^4*b^2*c^6*d^10*f^2 + 80*B^3*a^4*b^2*c^8*d^8*f^2 + 144*B^3*a^4*b^2*c^
10*d^6*f^2 + 80*B^3*a^4*b^2*c^12*d^4*f^2 + 16*B^3*a^4*b^2*c^14*d^2*f^2))*(-
(((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*
a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^
3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c
^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^
2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^
2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(
16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*
f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*B^2*a^4*c^5*f^2 - 4*B^2*b^4*c^5*f^2 + 24*B
^2*a^2*b^2*c^5*f^2 + 40*B^2*a^4*c^3*d^2*f^2 + 40*B^2*b^4*c^3*d^2*f^2 + 16*B
^2*a*b^3*d^5*f^2 - 16*B^2*a^3*b*d^5*f^2 - 20*B^2*a^4*c*d^4*f^2 - 20*B^2*b^4
*c*d^4*f^2 + 80*B^2*a*b^3*c^4*d*f^2 - 80*B^2*a^3*b*c^4*d*f^2 - 160*B^2*a*b^
3*c^2*d^3*f^2 + 120*B^2*a^2*b^2*c*d^4*f^2 + 160*B^2*a^3*b*c^2*d^3*f^2 - 240
*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4
*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*2i - atan((((((8*B^2*a^
4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2
*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2
+ 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^2 +
160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4
*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*

a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*B^2*a^4*c^5*f^2 + 4*B^2*b^4*c^5*f^2 - 24*B^2*a^2*b^2*c^5*f^2 - 40*B^2*a^4*c^3*d^2*f^2 - 40*B^2*b^4*c^3*d^2*f^2 - 16*B^2*a*b^3*d^5*f^2 + 16*B^2*a^3*b*d^5*f^2 + 20*B^2*a^4*c*d^4*f^2 + 20*B^2*b^4*c*d^4*f^2 - 80*B^2*a*b^3*c^4*d*f^2 + 80*B^2*a^3*b*c^4*d*f^2 + 160*B^2*a*b^3*c^2*d^3*f^2 - 120*B^2*a^2*b^2*c*d^4*f^2 - 160*B^2*a^3*b*c^2*d^3*f^2 + 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*B^2*a^4*c^5*f^2 + 4*B^2*b^4*c^5*f^2 - 24*B^2*a^2*b^2*c^5*f^2 - 40*B^2*a^4*c^3*d^2*f^2 - 40*B^2*b^4*c^3*d^2*f^2 - 16*B^2*a*b^3*d^5*f^2 + 16*B^2*a^3*b*d^5*f^2 + 20*B^2*a^4*c*d^4*f^2 + 20*B^2*b^4*c*d^4*f^2 - 80*B^2*a*b^3*c^4*d*f^2 + 80*B^2*a^3*b*c^4*d*f^2 + 160*B^2*a*b^3*c^2*d^3*f^2 - 120*B^2*a^2*b^2*c*d^4*f^2 - 160*B^2*a^3*b*c^2*d^3*f^2 + 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 96*B*a^2*c*d^20*f^4 + 96*B*b^2*c*d^20*f^4 - 736*B*a^2*c^3*d^18*f^4 - 2432*B*a^2*c^5*d^16*f^4 - 4480*B*a^2*c^7*d^14*f^4 - 4928*B*a^2*c^9*d^12*f^4 - 3136*B*a^2*c^11*d^10*f^4 - 896*B*a^2*c^13*d^8*f^4 + 128*B*a^2*c^15*d^6*f^4 + 160*B*a^2*c^17*d^4*f^4 + 32*B*a^2*c^19*d^2*f^4 + 736*B*b^2*c^3*d^18*f^4 + 2432*B*b^2*c^5*d^16*f^4 + 4480*B*b^2*c^7*d^14*f^4 + 4928*B*b^2*c^9*d^12*f^4 + 3136*B*b^2*c^11*d^10*f^4 + 896*B*b^2*c^13*d^8*f^4 - 128*B*b^2*c^15*d^6*f^4 - 160*B*b^2*c^17*d^4*f^4 - 32*B*b^2*c^19*d^2*f^4 - 64*B*a*b*d^21*f^4 - 320*B*a*b*c^2*d^19*f^4 - 256*B*a*b*c^4*d^17*f^4 + 1792*B*a*b*c^6*d^15*f^4 + 6272*B*a*b*c^8*d^13*f^4 + 9856*B*a*b*c^10*d^11*f^4 + 8960*B*a*b*c^12*d^9*f^4 + 4864*B*a*b*c^14*d^7*f^4 + 1472*B*a*b*c^16*d^5*f^4 + 192*B*a*b*c^18*d^3*f^4) + (c + d*tan(e + f*x))^(1/2)*(96*B^2*a^2*b^2*d^18*f^3 - 16*B^2*b^4*d^18*f^3 - 16*B^2*a^4*d^18*f^3 + 320*B^2*a^4*c^4*d^14*f^3 + 1024*B^2*a^4*c^6*d^12*f^3 + 1440*B^2*a^4*c^8*d^10*f^3 + 1024*B^2*a^4*c^10*d^8*f^3 + 320*B^2*a^4*c^12*d^6*f^3 - 16*B^2*a^4*c^16*d^2*f^3 + 320*B^2*b^4*c^4*d^14*f^3 + 1024*B^2*b^4*c^6*d^12*f^3 + 1440*B^2*b^4*c^8*d^10*f^3 + 1024*B^2*b^4*c^10*d^8*f^3 + 320*B^2*b^4*c^12*d^6*f^3 - 16*B^2*b^4*c^16*d^2*f^3 - 256*B^2*a*b^3*c*d^17*f^3 + 256*B^2*a^3*b*c*d^17*f^3 - 1280*B^2*a*b^3*c^3*d^15*f^3 - 2304*B^2*a*b^3*c^5*d^13*f^3 - 1280*B^2*a*b^3*c^7*d^11*f^3 + 1280*B^2*a*b^3*c^9*d^9*f^3 + 2304*B^2*a*b^3*c^11*d^7*f^3 + 1280*B^2*a*b^3*c^13*d^5*f^3 + 256*B^2*a*b^3*c^15*d^3*f^3 + 1280*B^2*a^3*b*c^3

*d^15*f^3 + 2304*B^2*a^3*b*c^5*d^13*f^3 + 1280*B^2*a^3*b*c^7*d^11*f^3 - 128
0*B^2*a^3*b*c^9*d^9*f^3 - 2304*B^2*a^3*b*c^11*d^7*f^3 - 1280*B^2*a^3*b*c^13
*d^5*f^3 - 256*B^2*a^3*b*c^15*d^3*f^3 - 1920*B^2*a^2*b^2*c^4*d^14*f^3 - 614
4*B^2*a^2*b^2*c^6*d^12*f^3 - 8640*B^2*a^2*b^2*c^8*d^10*f^3 - 6144*B^2*a^2*b
^2*c^10*d^8*f^3 - 1920*B^2*a^2*b^2*c^12*d^6*f^3 + 96*B^2*a^2*b^2*c^16*d^2*f
^3))*((((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 8
0*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*
B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a
*b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*
B^2*a^2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2
*f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*
b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^
6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*B^2*a^4*c^5*f^2 + 4*B^2*b^4*c^5*f^2
- 24*B^2*a^2*b^2*c^5*f^2 - 40*B^2*a^4*c^3*d^2*f^2 - 40*B^2*b^4*c^3*d^2*f^2
- 16*B^2*a*b^3*d^5*f^2 + 16*B^2*a^3*b*d^5*f^2 + 20*B^2*a^4*c*d^4*f^2 + 20*B
^2*b^4*c*d^4*f^2 - 80*B^2*a*b^3*c^4*d*f^2 + 80*B^2*a^3*b*c^4*d*f^2 + 160*B^
2*a*b^3*c^2*d^3*f^2 - 120*B^2*a^2*b^2*c*d^4*f^2 - 160*B^2*a^3*b*c^2*d^3*f^2
 + 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 +
10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*1i - (((((8*B^2*a^
4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2
*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2
 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^2 +
160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4
*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*
a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4
 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^
8*d^2*f^4))^(1/2) + 4*B^2*a^4*c^5*f^2 + 4*B^2*b^4*c^5*f^2 - 24*B^2*a^2*b^2*
c^5*f^2 - 40*B^2*a^4*c^3*d^2*f^2 - 40*B^2*b^4*c^3*d^2*f^2 - 16*B^2*a*b^3*d^
5*f^2 + 16*B^2*a^3*b*d^5*f^2 + 20*B^2*a^4*c*d^4*f^2 + 20*B^2*b^4*c*d^4*f^2
- 80*B^2*a*b^3*c^4*d*f^2 + 80*B^2*a^3*b*c^4*d*f^2 + 160*B^2*a*b^3*c^2*d^3*f
^2 - 120*B^2*a^2*b^2*c*d^4*f^2 - 160*B^2*a^3*b*c^2*d^3*f^2 + 240*B^2*a^2*b^
2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 +
10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(96*B*b^2*c*d^20*f^4 - 96*B*a^2*c*d
^20*f^4 - (c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*
f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*
f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 +
40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 +
320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d
^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*
b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^
8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*B^2*
a^4*c^5*f^2 + 4*B^2*b^4*c^5*f^2 - 24*B^2*a^2*b^2*c^5*f^2 - 40*B^2*a^4*c^3*d
^2*f^2 - 40*B^2*b^4*c^3*d^2*f^2 - 16*B^2*a*b^3*d^5*f^2 + 16*B^2*a^3*b*d^5*f
^2 + 20*B^2*a^4*c*d^4*f^2 + 20*B^2*b^4*c*d^4*f^2 - 80*B^2*a*b^3*c^4*d*f^2 +
 80*B^2*a^3*b*c^4*d*f^2 + 160*B^2*a*b^3*c^2*d^3*f^2 - 120*B^2*a^2*b^2*c*d^4

*f^2 - 160*B^2*a^3*b*c^2*d^3*f^2 + 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f
^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2
f^4)))^(1/2)(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*
c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f
^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2
*f^5) - 736*B*a^2*c^3*d^18*f^4 - 2432*B*a^2*c^5*d^16*f^4 - 4480*B*a^2*c^7*d
^14*f^4 - 4928*B*a^2*c^9*d^12*f^4 - 3136*B*a^2*c^11*d^10*f^4 - 896*B*a^2*c^
13*d^8*f^4 + 128*B*a^2*c^15*d^6*f^4 + 160*B*a^2*c^17*d^4*f^4 + 32*B*a^2*c^1
9*d^2*f^4 + 736*B*b^2*c^3*d^18*f^4 + 2432*B*b^2*c^5*d^16*f^4 + 4480*B*b^2*c
^7*d^14*f^4 + 4928*B*b^2*c^9*d^12*f^4 + 3136*B*b^2*c^11*d^10*f^4 + 896*B*b^
2*c^13*d^8*f^4 - 128*B*b^2*c^15*d^6*f^4 - 160*B*b^2*c^17*d^4*f^4 - 32*B*b^2
*c^19*d^2*f^4 - 64*B*a*b*d^21*f^4 - 320*B*a*b*c^2*d^19*f^4 - 256*B*a*b*c^4*
d^17*f^4 + 1792*B*a*b*c^6*d^15*f^4 + 6272*B*a*b*c^8*d^13*f^4 + 9856*B*a*b*c
^10*d^11*f^4 + 8960*B*a*b*c^12*d^9*f^4 + 4864*B*a*b*c^14*d^7*f^4 + 1472*B*a
*b*c^16*d^5*f^4 + 192*B*a*b*c^18*d^3*f^4) - (c + d*tan(e + f*x))^(1/2)*(96*
B^2*a^2*b^2*d^18*f^3 - 16*B^2*b^4*d^18*f^3 - 16*B^2*a^4*d^18*f^3 + 320*B^2*
a^4*c^4*d^14*f^3 + 1024*B^2*a^4*c^6*d^12*f^3 + 1440*B^2*a^4*c^8*d^10*f^3 +
1024*B^2*a^4*c^10*d^8*f^3 + 320*B^2*a^4*c^12*d^6*f^3 - 16*B^2*a^4*c^16*d^2*
f^3 + 320*B^2*b^4*c^4*d^14*f^3 + 1024*B^2*b^4*c^6*d^12*f^3 + 1440*B^2*b^4*c
^8*d^10*f^3 + 1024*B^2*b^4*c^10*d^8*f^3 + 320*B^2*b^4*c^12*d^6*f^3 - 16*B^2
*b^4*c^16*d^2*f^3 - 256*B^2*a*b^3*c*d^17*f^3 + 256*B^2*a^3*b*c*d^17*f^3 - 1
280*B^2*a*b^3*c^3*d^15*f^3 - 2304*B^2*a*b^3*c^5*d^13*f^3 - 1280*B^2*a*b^3*c
^7*d^11*f^3 + 1280*B^2*a*b^3*c^9*d^9*f^3 + 2304*B^2*a*b^3*c^11*d^7*f^3 + 12
80*B^2*a*b^3*c^13*d^5*f^3 + 256*B^2*a*b^3*c^15*d^3*f^3 + 1280*B^2*a^3*b*c^3
*d^15*f^3 + 2304*B^2*a^3*b*c^5*d^13*f^3 + 1280*B^2*a^3*b*c^7*d^11*f^3 - 128
0*B^2*a^3*b*c^9*d^9*f^3 - 2304*B^2*a^3*b*c^11*d^7*f^3 - 1280*B^2*a^3*b*c^13
*d^5*f^3 - 256*B^2*a^3*b*c^15*d^3*f^3 - 1920*B^2*a^2*b^2*c^4*d^14*f^3 - 614
4*B^2*a^2*b^2*c^6*d^12*f^3 - 8640*B^2*a^2*b^2*c^8*d^10*f^3 - 6144*B^2*a^2*b
^2*c^10*d^8*f^3 - 1920*B^2*a^2*b^2*c^12*d^6*f^3 + 96*B^2*a^2*b^2*c^16*d^2*f
^3))*((((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 8
0*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*
B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a
*b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*
B^2*a^2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2
*f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*
b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^
6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*B^2*a^4*c^5*f^2 + 4*B^2*b^4*c^5*f^2
- 24*B^2*a^2*b^2*c^5*f^2 - 40*B^2*a^4*c^3*d^2*f^2 - 40*B^2*b^4*c^3*d^2*f^2
- 16*B^2*a*b^3*d^5*f^2 + 16*B^2*a^3*b*d^5*f^2 + 20*B^2*a^4*c*d^4*f^2 + 20*B
^2*b^4*c*d^4*f^2 - 80*B^2*a*b^3*c^4*d*f^2 + 80*B^2*a^3*b*c^4*d*f^2 + 160*B^
2*a*b^3*c^2*d^3*f^2 - 120*B^2*a^2*b^2*c*d^4*f^2 - 160*B^2*a^3*b*c^2*d^3*f^2
+ 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 +
10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*1i)/(16*B^3*b^6*d^
16*f^2 - (((((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^
2 - 80*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2

+ 32*B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*B^2*a^4*c^5*f^2 + 4*B^2*b^4*c^5*f^2 - 24*B^2*a^2*b^2*c^5*f^2 - 40*B^2*a^4*c^3*d^2*f^2 - 40*B^2*b^4*c^3*d^2*f^2 - 16*B^2*a*b^3*d^5*f^2 + 16*B^2*a^3*b*d^5*f^2 + 20*B^2*a^4*c*d^4*f^2 + 20*B^2*b^4*c*d^4*f^2 - 80*B^2*a*b^3*c^4*d*f^2 + 80*B^2*a^3*b*c^4*d*f^2 + 160*B^2*a*b^3*c^2*d^3*f^2 - 120*B^2*a^2*b^2*c*d^4*f^2 - 160*B^2*a^3*b*c^2*d^3*f^2 + 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(96*B*b^2*c*d^20*f^4 - 96*B*a^2*c*d^20*f^4 - (c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*B^2*a^4*c^5*f^2 + 4*B^2*b^4*c^5*f^2 - 24*B^2*a^2*b^2*c^5*f^2 - 40*B^2*a^4*c^3*d^2*f^2 - 40*B^2*b^4*c^3*d^2*f^2 - 16*B^2*a*b^3*d^5*f^2 + 16*B^2*a^3*b*d^5*f^2 + 20*B^2*a^4*c*d^4*f^2 + 20*B^2*b^4*c*d^4*f^2 - 80*B^2*a*b^3*c^4*d*f^2 + 80*B^2*a^3*b*c^4*d*f^2 + 160*B^2*a*b^3*c^2*d^3*f^2 - 120*B^2*a^2*b^2*c*d^4*f^2 - 160*B^2*a^3*b*c^2*d^3*f^2 + 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 736*B*a^2*c^3*d^18*f^4 - 2432*B*a^2*c^5*d^16*f^4 - 4480*B*a^2*c^7*d^14*f^4 - 4928*B*a^2*c^9*d^12*f^4 - 3136*B*a^2*c^11*d^10*f^4 - 896*B*a^2*c^13*d^8*f^4 + 128*B*a^2*c^15*d^6*f^4 + 160*B*a^2*c^17*d^4*f^4 + 32*B*a^2*c^19*d^2*f^4 + 736*B*b^2*c^3*d^18*f^4 + 2432*B*b^2*c^5*d^16*f^4 + 4480*B*b^2*c^7*d^14*f^4 + 4928*B*b^2*c^9*d^12*f^4 + 3136*B*b^2*c^11*d^10*f^4 + 896*B*b^2*c^13*d^8*f^4 - 128*B*b^2*c^15*d^6*f^4 - 160*B*b^2*c^17*d^4*f^4 - 32*B*b^2*c^19*d^2*f^4 - 64*B*a*b*d^21*f^4 - 320*B*a*b*c^2*d^19*f^4 - 256*B*a*b*c^4*d^17*f^4 + 1792*B*a*b*c^6*d^15*f^4 + 6272*B*a*b*c^8*d^13*f^4 + 9856*B*a*b*c^10*d^11*f^4 + 8960*B*a*b*c^12*d^9*f^4 + 4864*B*a*b*c^14*d^7*f^4 + 1472*B*a*b*c^16*d^5*f^4 + 192*B*a*b*c^18*d^3*f^4) - (c + d*tan(e + f*x))^(1/2)*(96*B^2*a^2*b^2*d^18*f^3 - 16*B^2*b^4*d^18*f^3 - 16*B^2*a^4*d^18*f^3 + 320*B^2*a^4*c^4*d^14*f^3 + 1024*B^2*a^4*c^6*d^12*f^3 + 1440*B^2*a^4*c^8*d^10*f^3 + 1024*B^2*a^4*c^10*d^8*f^3 + 320*B^2*a^4*c^12*d^6*f^3 - 16*B^2*a^4*c^16*d^2*f^3 + 320*B^2*b^4*c^4*d^14*f^3 + 1024*B^2*b^4*c^6*d^12*f^3 + 1440*B^2*b^4*c^8*d^10*f^3 + 1024*B^2*b^4*c^10*d^8*f^3 + 320*B^2*b^4*c^12*d^6*f^3 - 16*B^2*b^4*c^16*d^2*f^3 - 256*B^2*a*b^3*c*d^17*f^3 + 256*B^

2*a^3*b*c*d^17*f^3 - 1280*B^2*a*b^3*c^3*d^15*f^3 - 2304*B^2*a*b^3*c^5*d^13*
f^3 - 1280*B^2*a*b^3*c^7*d^11*f^3 + 1280*B^2*a*b^3*c^9*d^9*f^3 + 2304*B^2*a
*b^3*c^11*d^7*f^3 + 1280*B^2*a*b^3*c^13*d^5*f^3 + 256*B^2*a*b^3*c^15*d^3*f^
3 + 1280*B^2*a^3*b*c^3*d^15*f^3 + 2304*B^2*a^3*b*c^5*d^13*f^3 + 1280*B^2*a^
3*b*c^7*d^11*f^3 - 1280*B^2*a^3*b*c^9*d^9*f^3 - 2304*B^2*a^3*b*c^11*d^7*f^3
 - 1280*B^2*a^3*b*c^13*d^5*f^3 - 256*B^2*a^3*b*c^15*d^3*f^3 - 1920*B^2*a^2*
b^2*c^4*d^14*f^3 - 6144*B^2*a^2*b^2*c^6*d^12*f^3 - 8640*B^2*a^2*b^2*c^8*d^1
0*f^3 - 6144*B^2*a^2*b^2*c^10*d^8*f^3 - 1920*B^2*a^2*b^2*c^12*d^6*f^3 + 96*
B^2*a^2*b^2*c^16*d^2*f^3))*((((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B
^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B
^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4
*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*
b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 4
80*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^
4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 16
0*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*B^2*a^4*c^5*f^
2 + 4*B^2*b^4*c^5*f^2 - 24*B^2*a^2*b^2*c^5*f^2 - 40*B^2*a^4*c^3*d^2*f^2 - 4
0*B^2*b^4*c^3*d^2*f^2 - 16*B^2*a*b^3*d^5*f^2 + 16*B^2*a^3*b*d^5*f^2 + 20*B^
2*a^4*c*d^4*f^2 + 20*B^2*b^4*c*d^4*f^2 - 80*B^2*a*b^3*c^4*d*f^2 + 80*B^2*a^
3*b*c^4*d*f^2 + 160*B^2*a*b^3*c^2*d^3*f^2 - 120*B^2*a^2*b^2*c*d^4*f^2 - 160
*B^2*a^3*b*c^2*d^3*f^2 + 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*
f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1
/2) - 16*B^3*a^6*d^16*f^2 - (((((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48
*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32
*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b
^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*
a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 +
 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*
B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 +
160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*B^2*a^4*c^5*
f^2 + 4*B^2*b^4*c^5*f^2 - 24*B^2*a^2*b^2*c^5*f^2 - 40*B^2*a^4*c^3*d^2*f^2 -
 40*B^2*b^4*c^3*d^2*f^2 - 16*B^2*a*b^3*d^5*f^2 + 16*B^2*a^3*b*d^5*f^2 + 20*
B^2*a^4*c*d^4*f^2 + 20*B^2*b^4*c*d^4*f^2 - 80*B^2*a*b^3*c^4*d*f^2 + 80*B^2*
a^3*b*c^4*d*f^2 + 160*B^2*a*b^3*c^2*d^3*f^2 - 120*B^2*a^2*b^2*c*d^4*f^2 - 1
60*B^2*a^3*b*c^2*d^3*f^2 + 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^1
0*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^
(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2
 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2
 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*
B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320
*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*
f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6
 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f
^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*B^2*a^4
*c^5*f^2 + 4*B^2*b^4*c^5*f^2 - 24*B^2*a^2*b^2*c^5*f^2 - 40*B^2*a^4*c^3*d^2*

f^2 - 40*B^2*b^4*c^3*d^2*f^2 - 16*B^2*a*b^3*d^5*f^2 + 16*B^2*a^3*b*d^5*f^2
+ 20*B^2*a^4*c*d^4*f^2 + 20*B^2*b^4*c*d^4*f^2 - 80*B^2*a*b^3*c^4*d*f^2 + 80
*B^2*a^3*b*c^4*d*f^2 + 160*B^2*a*b^3*c^2*d^3*f^2 - 120*B^2*a^2*b^2*c*d^4*f^
2 - 160*B^2*a^3*b*c^2*d^3*f^2 + 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4
+ d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^
4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7
*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5
+ 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^
5) - 96*B*a^2*c*d^20*f^4 + 96*B*b^2*c*d^20*f^4 - 736*B*a^2*c^3*d^18*f^4 - 2
432*B*a^2*c^5*d^16*f^4 - 4480*B*a^2*c^7*d^14*f^4 - 4928*B*a^2*c^9*d^12*f^4
- 3136*B*a^2*c^11*d^10*f^4 - 896*B*a^2*c^13*d^8*f^4 + 128*B*a^2*c^15*d^6*f^
4 + 160*B*a^2*c^17*d^4*f^4 + 32*B*a^2*c^19*d^2*f^4 + 736*B*b^2*c^3*d^18*f^4
 + 2432*B*b^2*c^5*d^16*f^4 + 4480*B*b^2*c^7*d^14*f^4 + 4928*B*b^2*c^9*d^12*
f^4 + 3136*B*b^2*c^11*d^10*f^4 + 896*B*b^2*c^13*d^8*f^4 - 128*B*b^2*c^15*d^
6*f^4 - 160*B*b^2*c^17*d^4*f^4 - 32*B*b^2*c^19*d^2*f^4 - 64*B*a*b*d^21*f^4
- 320*B*a*b*c^2*d^19*f^4 - 256*B*a*b*c^4*d^17*f^4 + 1792*B*a*b*c^6*d^15*f^4
 + 6272*B*a*b*c^8*d^13*f^4 + 9856*B*a*b*c^10*d^11*f^4 + 8960*B*a*b*c^12*d^9
*f^4 + 4864*B*a*b*c^14*d^7*f^4 + 1472*B*a*b*c^16*d^5*f^4 + 192*B*a*b*c^18*d
^3*f^4) + (c + d*tan(e + f*x))^(1/2)*(96*B^2*a^2*b^2*d^18*f^3 - 16*B^2*b^4*
d^18*f^3 - 16*B^2*a^4*d^18*f^3 + 320*B^2*a^4*c^4*d^14*f^3 + 1024*B^2*a^4*c^
6*d^12*f^3 + 1440*B^2*a^4*c^8*d^10*f^3 + 1024*B^2*a^4*c^10*d^8*f^3 + 320*B^
2*a^4*c^12*d^6*f^3 - 16*B^2*a^4*c^16*d^2*f^3 + 320*B^2*b^4*c^4*d^14*f^3 + 1
024*B^2*b^4*c^6*d^12*f^3 + 1440*B^2*b^4*c^8*d^10*f^3 + 1024*B^2*b^4*c^10*d^
8*f^3 + 320*B^2*b^4*c^12*d^6*f^3 - 16*B^2*b^4*c^16*d^2*f^3 - 256*B^2*a*b^3*
c*d^17*f^3 + 256*B^2*a^3*b*c*d^17*f^3 - 1280*B^2*a*b^3*c^3*d^15*f^3 - 2304*
B^2*a*b^3*c^5*d^13*f^3 - 1280*B^2*a*b^3*c^7*d^11*f^3 + 1280*B^2*a*b^3*c^9*d
^9*f^3 + 2304*B^2*a*b^3*c^11*d^7*f^3 + 1280*B^2*a*b^3*c^13*d^5*f^3 + 256*B^
2*a*b^3*c^15*d^3*f^3 + 1280*B^2*a^3*b*c^3*d^15*f^3 + 2304*B^2*a^3*b*c^5*d^1
3*f^3 + 1280*B^2*a^3*b*c^7*d^11*f^3 - 1280*B^2*a^3*b*c^9*d^9*f^3 - 2304*B^2
*a^3*b*c^11*d^7*f^3 - 1280*B^2*a^3*b*c^13*d^5*f^3 - 256*B^2*a^3*b*c^15*d^3*
f^3 - 1920*B^2*a^2*b^2*c^4*d^14*f^3 - 6144*B^2*a^2*b^2*c^6*d^12*f^3 - 8640*
B^2*a^2*b^2*c^8*d^10*f^3 - 6144*B^2*a^2*b^2*c^10*d^8*f^3 - 1920*B^2*a^2*b^2
*c^12*d^6*f^3 + 96*B^2*a^2*b^2*c^16*d^2*f^3))*((((8*B^2*a^4*c^5*f^2 + 8*B^2
*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4
*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d
^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4
*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 320*B^2*a^
3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4
*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 +
80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2)
 + 4*B^2*a^4*c^5*f^2 + 4*B^2*b^4*c^5*f^2 - 24*B^2*a^2*b^2*c^5*f^2 - 40*B^2*
a^4*c^3*d^2*f^2 - 40*B^2*b^4*c^3*d^2*f^2 - 16*B^2*a*b^3*d^5*f^2 + 16*B^2*a^
3*b*d^5*f^2 + 20*B^2*a^4*c*d^4*f^2 + 20*B^2*b^4*c*d^4*f^2 - 80*B^2*a*b^3*c^
4*d*f^2 + 80*B^2*a^3*b*c^4*d*f^2 + 160*B^2*a*b^3*c^2*d^3*f^2 - 120*B^2*a^2*
b^2*c*d^4*f^2 - 160*B^2*a^3*b*c^2*d^3*f^2 + 240*B^2*a^2*b^2*c^3*d^2*f^2)/(1

6*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2) + 16*B^3*a^2*b^4*d^16*f^2 - 16*B^3*a^4*b^2*d^16*f^2 - 80*B^3*a^6*c^2*d^14*f^2 - 144*B^3*a^6*c^4*d^12*f^2 - 80*B^3*a^6*c^6*d^10*f^2 + 80*B^3*a^6*c^8*d^8*f^2 + 144*B^3*a^6*c^10*d^6*f^2 + 80*B^3*a^6*c^12*d^4*f^2 + 16*B^3*a^6*c^14*d^2*f^2 + 80*B^3*b^6*c^2*d^14*f^2 + 144*B^3*b^6*c^4*d^12*f^2 + 80*B^3*b^6*c^6*d^10*f^2 - 80*B^3*b^6*c^8*d^8*f^2 - 144*B^3*b^6*c^10*d^6*f^2 - 80*B^3*b^6*c^12*d^4*f^2 - 16*B^3*b^6*c^14*d^2*f^2 + 64*B^3*a*b^5*c*d^15*f^2 + 64*B^3*a^5*b*c*d^15*f^2 + 384*B^3*a*b^5*c^3*d^13*f^2 + 960*B^3*a*b^5*c^5*d^11*f^2 + 1280*B^3*a*b^5*c^7*d^9*f^2 + 960*B^3*a*b^5*c^9*d^7*f^2 + 384*B^3*a*b^5*c^11*d^5*f^2 + 64*B^3*a*b^5*c^13*d^3*f^2 + 128*B^3*a^3*b^3*c*d^15*f^2 + 384*B^3*a^5*b*c^3*d^13*f^2 + 960*B^3*a^5*b*c^5*d^11*f^2 + 1280*B^3*a^5*b*c^7*d^9*f^2 + 960*B^3*a^5*b*c^9*d^7*f^2 + 384*B^3*a^5*b*c^11*d^5*f^2 + 64*B^3*a^5*b*c^13*d^3*f^2 + 80*B^3*a^2*b^4*c^2*d^14*f^2 + 144*B^3*a^2*b^4*c^4*d^12*f^2 + 80*B^3*a^2*b^4*c^6*d^10*f^2 - 80*B^3*a^2*b^4*c^8*d^8*f^2 - 144*B^3*a^2*b^4*c^10*d^6*f^2 - 80*B^3*a^2*b^4*c^12*d^4*f^2 - 16*B^3*a^2*b^4*c^14*d^2*f^2 + 768*B^3*a^3*b^3*c^3*d^13*f^2 + 1920*B^3*a^3*b^3*c^5*d^11*f^2 + 2560*B^3*a^3*b^3*c^7*d^9*f^2 + 1920*B^3*a^3*b^3*c^9*d^7*f^2 + 768*B^3*a^3*b^3*c^11*d^5*f^2 + 128*B^3*a^3*b^3*c^13*d^3*f^2 - 80*B^3*a^4*b^2*c^2*d^14*f^2 - 144*B^3*a^4*b^2*c^4*d^12*f^2 - 80*B^3*a^4*b^2*c^6*d^10*f^2 + 80*B^3*a^4*b^2*c^8*d^8*f^2 + 144*B^3*a^4*b^2*c^10*d^6*f^2 + 80*B^3*a^4*b^2*c^12*d^4*f^2 + 16*B^3*a^4*b^2*c^14*d^2*f^2))*((((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*B^2*a^4*c^5*f^2 + 4*B^2*b^4*c^5*f^2 - 24*B^2*a^2*b^2*c^5*f^2 - 40*B^2*a^4*c^3*d^2*f^2 - 40*B^2*b^4*c^3*d^2*f^2 - 16*B^2*a*b^3*d^5*f^2 + 16*B^2*a^3*b*d^5*f^2 + 20*B^2*a^4*c*d^4*f^2 + 20*B^2*b^4*c*d^4*f^2 - 80*B^2*a*b^3*c^4*d*f^2 + 80*B^2*a^3*b*c^4*d*f^2 + 160*B^2*a*b^3*c^2*d^3*f^2 - 120*B^2*a^2*b^2*c*d^4*f^2 - 160*B^2*a^3*b*c^2*d^3*f^2 + 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*2i - ((2*(A*a^2*d^2 + A*b^2*c^2 - 2*A*a*b*c*d))/(3*(c^2 + d^2)) - (4*d*(c + d*tan(e + f*x))*(A*a*b*c^2 - A*a*b*d^2 - A*a^2*c*d + A*b^2*c*d))/(c^2 + d^2)^2)/(d*f*(c + d*tan(e + f*x))^(3/2)) - ((2*(C*b^2*c^4 + C*a^2*c^2*d^2 - 2*C*a*b*c^3*d))/(3*(c^2 + d^2)) - (4*(c + d*tan(e + f*x))*(C*b^2*c^5 + C*a^2*c*d^4 + 2*C*b^2*c^3*d^2 - C*a*b*c^4*d - 3*C*a*b*c^2*d^3))/(c^2 + d^2)^2)/(d^3*f*(c + d*tan(e + f*x))^(3/2)) + ((2*(B*b^2*c^3 + B*a^2*c*d^2 - 2*B*a*b*c^2*d))/(3*(c^2 + d^2)) - (2*(c + d*tan(e + f*x))*(B*a^2*d^4 + B*b^2*c^4 - B*a^2*c^2*d^2 + 3*B*b^2*c^2*d^2 - 4*B*a*b*c*d^3))/(c^2 + d^2)^2)/(d^2*f*(c + d*tan(e + f*x))^(3/2)) + (2*C*b^2*(c + d*tan(e + f*x))^(1/2))/(d^3*f)

3.124 $\int \frac{(a+b\tan(e+fx))(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx$

Optimal result . . . 1788
Rubi [A] (verified) . . . 1789
Mathematica [C] (verified) . . . 1792
Maple [B] (verified) . . . 1792
Fricas [B] (verification not implemented) . . . 1793
Sympy [F] . . . 1793
Maxima [F(-1)] . . . 1793
Giac [F(-1)] . . . 1794
Mupad [B] (verification not implemented) . . . 1794

Optimal result

Integrand size = 45, antiderivative size = 273

$$\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{5/2}}\,dx = -\frac{(a-ib)(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(c-id)^{5/2}f} + \frac{(ia-b)(A+iB-C)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(c+id)^{5/2}f} + \frac{2(bc-ad)\left(c^2C-Bcd+Ad^2\right)}{3d^2\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}} - \frac{2(b(c^4C-c^2(A-3C)d^2-2Bcd^3+Ad^4)+ad^2(2c(A-C)d-B(c^2-d^2)))}{d^2\left(c^2+d^2\right)^2 f\sqrt{c+d\tan(e+fx)}}$$

```
[Out] -(a-I*b)*(I*A+B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(c-I*d)^
(5/2)/f+(I*a-b)*(A+I*B-C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))/(c+
I*d)^(5/2)/f-2*(b*(c^4*C-c^2*(A-3*C)*d^2-2*B*c*d^3+A*d^4)+a*d^2*(2*c*(A-C)*
d-B*(c^2-d^2)))/d^2/(c^2+d^2)^2/f/(c+d*tan(f*x+e))^(1/2)+2/3*(-a*d+b*c)*(A*
d^2-B*c*d+C*c^2)/d^2/(c^2+d^2)/f/(c+d*tan(f*x+e))^(3/2)
```

Rubi [A] (verified)

Time = 1.14 (sec) , antiderivative size = 271, normalized size of antiderivative = 0.99, number of steps used = 9, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.133, Rules used = {3716, 3709, 3620, 3618, 65, 214}

$$\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{5/2}}\,dx = -\frac{(b+ia)(A-iB-C)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f(c-id)^{5/2}} + \frac{(-b+ia)(A+iB-C)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f(c+id)^{5/2}} + \frac{2(bc-ad)\left(Ad^2-Bcd+c^2C\right)}{3d^2f\left(c^2+d^2\right)(c+d\tan(e+fx))^{3/2}} - \frac{2(ad^2(2cd(A-C)-B(c^2-d^2))+b(-c^2d^2(A-3C)+Ad^4-2Bcd^3+c^4C))}{d^2f\left(c^2+d^2\right)^2\sqrt{c+d\tan(e+fx)}}$$

```
[In] Int[((a + b*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c + d*T
an[e + f*x])^(5/2),x]
```

```
[Out] -(((I*a + b)*(A - I*B - C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]])
/((c - I*d)^(5/2)*f)) + ((I*a - b)*(A + I*B - C)*ArcTanh[Sqrt[c + d*Tan[e +
 f*x]]/Sqrt[c + I*d]])/((c + I*d)^(5/2)*f) + (2*(b*c - a*d)*(c^2*C - B*c*d
+ A*d^2))/(3*d^2*(c^2 + d^2)*f*(c + d*Tan[e + f*x])^(3/2)) - (2*(b*(c^4*C -
 c^2*(A - 3*C)*d^2 - 2*B*c*d^3 + A*d^4) + a*d^2*(2*c*(A - C)*d - B*(c^2 - d
^2))))/(d^2*(c^2 + d^2)^2*f*Sqrt[c + d*Tan[e + f*x]])
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
```

```
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3709

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 -
 a*b*B + a^2*C)*((a + b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1)*(a^2 + b^2))), x
] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[b*B + a*(A -
C) - (A*b - a*B - b*C)*Tan[e + f*x], x], x], x] /; FreeQ[{a, b, e, f, A, B,
 C}, x] && NeQ[A*b^2 - a*b*B + a^2*C, 0] && LtQ[m, -1] && NeQ[a^2 + b^2, 0]
```

Rule 3716

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.
)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(-(b*c - a*d))*(c^2*C - B*c*d + A*d^2)*((c
+ d*Tan[e + f*x])^(n + 1)/(d^2*f*(n + 1)*(c^2 + d^2))), x] + Dist[1/(d*(c^2
 + d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)*Simp[a*d*(A*c - c*C + B*d) + b*(
c^2*C - B*c*d + A*d^2) + d*(A*b*c + a*B*c - b*c*C - a*A*d + b*B*d + a*C*d)*
Tan[e + f*x] + b*C*(c^2 + d^2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c
, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] && LtQ[n,
 -1]
```

Rubi steps

$$\text{integral} = \frac{2(bc-ad)\left(c^2C-Bcd+Ad^2\right)}{3d^2\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}} + \frac{\int \frac{ad(Ac-cC+Bd)+b\left(c^2C-Bcd+Ad^2\right)+d(Abc+aBc-bcC-aAd+bBd+aCd)\tan(e+fx)+bC\left(c^2+d^2\right)\tan^2(e+fx)}{(c+d\tan(e+fx))^{3/2}}\,dx}{d\left(c^2+d^2\right)}$$

$$= \frac{2(bc-ad)\left(c^2C-Bcd+Ad^2\right)}{3d^2\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}} - \frac{2(b(c^4C-c^2(A-3C)d^2-2Bcd^3+Ad^4)+ad^2(2c(A-C)d-B(c^2-d^2)))}{d^2\left(c^2+d^2\right)^2 f\sqrt{c+d\tan(e+fx)}} + \frac{\int \frac{-d\left(a\left(c^2C-2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-b\left(2c(A-C)d-B\left(c^2-d^2\right)\right)\right)-d\left(2aAcd-2acCd-Ab\left(c^2-d^2\right)-aB\left(c^2-d^2\right)+b\left(c^2C-2Bcd-C\right.\right.}{\sqrt{c+d\tan(e+fx)}}}{d\left(c^2+d^2\right)^2}$$

$$
\begin{aligned}
&= \frac{2(bc-ad)\left(c^2C-Bcd+Ad^2\right)}{3d^2\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}} \\
&\quad - \frac{2(b(c^4C-c^2(A-3C)d^2-2Bcd^3+Ad^4)+ad^2(2c(A-C)d-B(c^2-d^2)))}{d^2\left(c^2+d^2\right)^2f\sqrt{c+d\tan(e+fx)}} \\
&\quad + \frac{((a-ib)(A-iB-C))\int\frac{1+i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(c-id)^2} \\
&\quad + \frac{((a+ib)(A+iB-C))\int\frac{1-i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(c+id)^2} \\
&= \frac{2(bc-ad)\left(c^2C-Bcd+Ad^2\right)}{3d^2\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}} \\
&\quad - \frac{2(b(c^4C-c^2(A-3C)d^2-2Bcd^3+Ad^4)+ad^2(2c(A-C)d-B(c^2-d^2)))}{d^2\left(c^2+d^2\right)^2f\sqrt{c+d\tan(e+fx)}} \\
&\quad + \frac{((ia+b)(A-iB-C))\text{Subst}\left(\int\frac{1}{(-1+x)\sqrt{c-idx}}\,dx,x,i\tan(e+fx)\right)}{2(c-id)^2f} \\
&\quad - \frac{((ia-b)(A+iB-C))\text{Subst}\left(\int\frac{1}{(-1+x)\sqrt{c+idx}}\,dx,x,-i\tan(e+fx)\right)}{2(c+id)^2f} \\
&= \frac{2(bc-ad)\left(c^2C-Bcd+Ad^2\right)}{3d^2\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}} \\
&\quad - \frac{2(b(c^4C-c^2(A-3C)d^2-2Bcd^3+Ad^4)+ad^2(2c(A-C)d-B(c^2-d^2)))}{d^2\left(c^2+d^2\right)^2f\sqrt{c+d\tan(e+fx)}} \\
&\quad + \frac{((a+ib)(A+iB-C))\text{Subst}\left(\int\frac{1}{-1+\frac{ic}{d}-\frac{ix^2}{d}}\,dx,x,\sqrt{c+d\tan(e+fx)}\right)}{(ic-d)^2df} \\
&\quad + \frac{((a-ib)(A-iB-C))\text{Subst}\left(\int\frac{1}{-1-\frac{ic}{d}+\frac{ix^2}{d}}\,dx,x,\sqrt{c+d\tan(e+fx)}\right)}{d(ic+d)^2f} \\
&= -\frac{(ia+b)(A-iB-C)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(c-id)^{5/2}f} \\
&\quad + \frac{(ia-b)(A+iB-C)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(c+id)^{5/2}f} \\
&\quad + \frac{2(bc-ad)\left(c^2C-Bcd+Ad^2\right)}{3d^2\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}} \\
&\quad - \frac{2(b(c^4C-c^2(A-3C)d^2-2Bcd^3+Ad^4)+ad^2(2c(A-C)d-B(c^2-d^2)))}{d^2\left(c^2+d^2\right)^2f\sqrt{c+d\tan(e+fx)}}
\end{aligned}
$$

Mathematica [C] (verified)

Result contains higher order function than in optimal. Order 5 vs. order 3 in optimal.

Time = 3.16 (sec) , antiderivative size = 300, normalized size of antiderivative = 1.10

$$\int \frac{(a+b\tan(e+fx))\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx =$$

$$-\frac{2(c-id)(c+id)(2bcC+bBd-2aCd)+d(Abc+aBc-bcC-aAd+bBd+aCd)\left(i(c+id)\,\text{Hypergeom}\right.}{}$$

```
[In] Integrate[((a + b*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c
 + d*Tan[e + f*x])^(5/2),x]
```

```
[Out] -1/3*(2*(c - I*d)*(c + I*d)*(2*b*c*C + b*B*d - 2*a*C*d) + d*(A*b*c + a*B*c
- b*c*C - a*A*d + b*B*d + a*C*d)*(I*(c + I*d)*Hypergeometric2F1[-3/2, 1, -1
/2, (c + d*Tan[e + f*x])/(c - I*d)] - (I*c + d)*Hypergeometric2F1[-3/2, 1,
-1/2, (c + d*Tan[e + f*x])/(c + I*d)]) + 6*C*(c - I*d)*(c + I*d)*d*(a + b*T
an[e + f*x]) - 3*(A*b + a*B - b*C)*d*(I*(c + I*d)*Hypergeometric2F1[-1/2, 1
, 1/2, (c + d*Tan[e + f*x])/(c - I*d)] - (I*c + d)*Hypergeometric2F1[-1/2,
1, 1/2, (c + d*Tan[e + f*x])/(c + I*d)])*(c + d*Tan[e + f*x]))/(d^2*(c^2 +
d^2)*f*(c + d*Tan[e + f*x])^(3/2))
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 8962 vs. 2(246) = 492.

Time = 0.25 (sec) , antiderivative size = 8963, normalized size of antiderivative = 32.83

method	result	size
parts	Expression too large to display	8963
derivativedivides	Expression too large to display	40201
default	Expression too large to display	40201

```
[In] int((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(5/2)
,x,method=_RETURNVERBOSE)
```

```
[Out] result too large to display
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 50755 vs. 2(237) = 474.

Time = 297.23 (sec) , antiderivative size = 50755, normalized size of antiderivative = 185.92

$$\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{5/2}}\,dx = \text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(5/2),x, algorithm="fricas")
```

```
[Out] Too large to include
```

Sympy [F]

$$\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{5/2}}\,dx = \int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+$$

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x+e)
)**(5/2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/(c +
 d*tan(e + f*x))**(5/2), x)
```

Maxima [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{5/2}}\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(5/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

Giac [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{5/2}}\,dx=\text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(5/2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [B] (verification not implemented)

Time = 85.53 (sec) , antiderivative size = 64641, normalized size of antiderivative = 236.78

$$\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{5/2}}\,dx=\text{Too large to display}$$

```
[In] int(((a + b*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d*t
an(e + f*x))^(5/2),x)
```

```
[Out] ((2*(C*b*c^3 + A*b*c*d^2 - B*b*c^2*d))/(3*(c^2 + d^2)) - (2*(c + d*tan(e +
f*x))*(A*b*d^4 + C*b*c^4 - 2*B*b*c*d^3 - A*b*c^2*d^2 + 3*C*b*c^2*d^2))/(c^2
 + d^2)^2)/(d^2*f*(c + d*tan(e + f*x))^(3/2)) - atan(-(((c + d*tan(e + f*x)
)^(1/2)*(16*A^2*b^2*d^18*f^3 - 16*B^2*b^2*d^18*f^3 + 16*C^2*b^2*d^18*f^3 -
320*A^2*b^2*c^4*d^14*f^3 - 1024*A^2*b^2*c^6*d^12*f^3 - 1440*A^2*b^2*c^8*d^1
0*f^3 - 1024*A^2*b^2*c^10*d^8*f^3 - 320*A^2*b^2*c^12*d^6*f^3 + 16*A^2*b^2*c
^16*d^2*f^3 + 320*B^2*b^2*c^4*d^14*f^3 + 1024*B^2*b^2*c^6*d^12*f^3 + 1440*B
^2*b^2*c^8*d^10*f^3 + 1024*B^2*b^2*c^10*d^8*f^3 + 320*B^2*b^2*c^12*d^6*f^3
- 16*B^2*b^2*c^16*d^2*f^3 - 320*C^2*b^2*c^4*d^14*f^3 - 1024*C^2*b^2*c^6*d^1
2*f^3 - 1440*C^2*b^2*c^8*d^10*f^3 - 1024*C^2*b^2*c^10*d^8*f^3 - 320*C^2*b^2
*c^12*d^6*f^3 + 16*C^2*b^2*c^16*d^2*f^3 - 32*A*C*b^2*d^18*f^3 - 128*A*B*b^2
*c*d^17*f^3 + 128*B*C*b^2*c*d^17*f^3 - 640*A*B*b^2*c^3*d^15*f^3 - 1152*A*B*
b^2*c^5*d^13*f^3 - 640*A*B*b^2*c^7*d^11*f^3 + 640*A*B*b^2*c^9*d^9*f^3 + 115
2*A*B*b^2*c^11*d^7*f^3 + 640*A*B*b^2*c^13*d^5*f^3 + 128*A*B*b^2*c^15*d^3*f^
3 + 640*A*C*b^2*c^4*d^14*f^3 + 2048*A*C*b^2*c^6*d^12*f^3 + 2880*A*C*b^2*c^8
*d^10*f^3 + 2048*A*C*b^2*c^10*d^8*f^3 + 640*A*C*b^2*c^12*d^6*f^3 - 32*A*C*b
^2*c^16*d^2*f^3 + 640*B*C*b^2*c^3*d^15*f^3 + 1152*B*C*b^2*c^5*d^13*f^3 + 64
0*B*C*b^2*c^7*d^11*f^3 - 640*B*C*b^2*c^9*d^9*f^3 - 1152*B*C*b^2*c^11*d^7*f^
3 - 640*B*C*b^2*c^13*d^5*f^3 - 128*B*C*b^2*c^15*d^3*f^3) + ((((8*A^2*b^2*c^
5*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80
*B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C
*b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4
*f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 -
```

80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^5*f^2 - 4*B^2*b^2*c^5*f^2 + 4*C^2*b^2*c^5*f^2 - 40*A^2*b^2*c^3*d^2*f^2 + 40*B^2*b^2*c^3*d^2*f^2 - 40*C^2*b^2*c^3*d^2*f^2 + 8*A*B*b^2*d^5*f^2 - 8*A*C*b^2*c^5*f^2 - 8*B*C*b^2*d^5*f^2 + 20*A^2*b^2*c*d^4*f^2 - 20*B^2*b^2*c*d^4*f^2 + 20*C^2*b^2*c*d^4*f^2 + 40*A*B*b^2*c^4*d*f^2 - 40*A*C*b^2*c*d^4*f^2 - 40*B*C*b^2*c^4*d*f^2 - 80*A*B*b^2*c^2*d^3*f^2 + 80*A*C*b^2*c^3*d^2*f^2 + 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(128*A*b*c^15*d^6*f^4 - 32*B*b*d^21*f^4 - 736*A*b*c^3*d^18*f^4 - 2432*A*b*c^5*d^16*f^4 - 4480*A*b*c^7*d^14*f^4 - 4928*A*b*c^9*d^12*f^4 - 3136*A*b*c^11*d^10*f^4 - 896*A*b*c^13*d^8*f^4 - (c + d*tan(e + f*x))^(1/2)*((((8*A^2*b^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^5*f^2 - 4*B^2*b^2*c^5*f^2 + 4*C^2*b^2*c^5*f^2 - 40*A^2*b^2*c^3*d^2*f^2 + 40*B^2*b^2*c^3*d^2*f^2 - 40*C^2*b^2*c^3*d^2*f^2 + 8*A*B*b^2*d^5*f^2 - 8*A*C*b^2*c^5*f^2 - 8*B*C*b^2*d^5*f^2 + 20*A^2*b^2*c*d^4*f^2 - 20*B^2*b^2*c*d^4*f^2 + 20*C^2*b^2*c*d^4*f^2 + 40*A*B*b^2*c^4*d*f^2 - 40*A*C*b^2*c*d^4*f^2 - 40*B*C*b^2*c^4*d*f^2 - 80*A*B*b^2*c^2*d^3*f^2 + 80*A*C*b^2*c^3*d^2*f^2 + 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) + 160*A*b*c^17*d^4*f^4 + 32*A*b*c^19*d^2*f^4 - 160*B*b*c^2*d^19*f^4 - 128*B*b*c^4*d^17*f^4 + 896*B*b*c^6*d^15*f^4 + 3136*B*b*c^8*d^13*f^4 + 4928*B*b*c^10*d^11*f^4 + 4480*B*b*c^12*d^9*f^4 + 2432*B*b*c^14*d^7*f^4 + 736*B*b*c^16*d^5*f^4 + 96*B*b*c^18*d^3*f^4 + 736*C*b*c^3*d^18*f^4 + 2432*C*b*c^5*d^16*f^4 + 4480*C*b*c^7*d^14*f^4 + 4928*C*b*c^9*d^12*f^4 + 3136*C*b*c^11*d^10*f^4 + 896*C*b*c^13*d^8*f^4 - 128*C*b*c^15*d^6*f^4 - 160*C*b*c^17*d^4*f^4 - 32*C*b*c^19*d^2*f^4 - 96*A*b*c*d^20*f^4 + 96*C*b*c*d^20*f^4))*((((8*A^2*b^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4

+ 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^
8*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2
*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^
2*c^5*f^2 - 4*B^2*b^2*c^5*f^2 + 4*C^2*b^2*c^5*f^2 - 40*A^2*b^2*c^3*d^2*f^2
+ 40*B^2*b^2*c^3*d^2*f^2 - 40*C^2*b^2*c^3*d^2*f^2 + 8*A*B*b^2*d^5*f^2 - 8*A
*C*b^2*c^5*f^2 - 8*B*C*b^2*d^5*f^2 + 20*A^2*b^2*c*d^4*f^2 - 20*B^2*b^2*c*d^
4*f^2 + 20*C^2*b^2*c*d^4*f^2 + 40*A*B*b^2*c^4*d*f^2 - 40*A*C*b^2*c*d^4*f^2
- 40*B*C*b^2*c^4*d*f^2 - 80*A*B*b^2*c^2*d^3*f^2 + 80*A*C*b^2*c^3*d^2*f^2 +
80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d
^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*1i + ((c + d*tan(e + f*x))
^(1/2)*(16*A^2*b^2*d^18*f^3 - 16*B^2*b^2*d^18*f^3 + 16*C^2*b^2*d^18*f^3 - 3
20*A^2*b^2*c^4*d^14*f^3 - 1024*A^2*b^2*c^6*d^12*f^3 - 1440*A^2*b^2*c^8*d^10
*f^3 - 1024*A^2*b^2*c^10*d^8*f^3 - 320*A^2*b^2*c^12*d^6*f^3 + 16*A^2*b^2*c^
16*d^2*f^3 + 320*B^2*b^2*c^4*d^14*f^3 + 1024*B^2*b^2*c^6*d^12*f^3 + 1440*B^
2*b^2*c^8*d^10*f^3 + 1024*B^2*b^2*c^10*d^8*f^3 + 320*B^2*b^2*c^12*d^6*f^3 -
16*B^2*b^2*c^16*d^2*f^3 - 320*C^2*b^2*c^4*d^14*f^3 - 1024*C^2*b^2*c^6*d^12
*f^3 - 1440*C^2*b^2*c^8*d^10*f^3 - 1024*C^2*b^2*c^10*d^8*f^3 - 320*C^2*b^2*
c^12*d^6*f^3 + 16*C^2*b^2*c^16*d^2*f^3 - 32*A*C*b^2*d^18*f^3 - 128*A*B*b^2*
c*d^17*f^3 + 128*B*C*b^2*c*d^17*f^3 - 640*A*B*b^2*c^3*d^15*f^3 - 1152*A*B*b
^2*c^5*d^13*f^3 - 640*A*B*b^2*c^7*d^11*f^3 + 640*A*B*b^2*c^9*d^9*f^3 + 1152
*A*B*b^2*c^11*d^7*f^3 + 640*A*B*b^2*c^13*d^5*f^3 + 128*A*B*b^2*c^15*d^3*f^3
+ 640*A*C*b^2*c^4*d^14*f^3 + 2048*A*C*b^2*c^6*d^12*f^3 + 2880*A*C*b^2*c^8*
d^10*f^3 + 2048*A*C*b^2*c^10*d^8*f^3 + 640*A*C*b^2*c^12*d^6*f^3 - 32*A*C*b^
2*c^16*d^2*f^3 + 640*B*C*b^2*c^3*d^15*f^3 + 1152*B*C*b^2*c^5*d^13*f^3 + 640
*B*C*b^2*c^7*d^11*f^3 - 640*B*C*b^2*c^9*d^9*f^3 - 1152*B*C*b^2*c^11*d^7*f^3
- 640*B*C*b^2*c^13*d^5*f^3 - 128*B*C*b^2*c^15*d^3*f^3) - ((((8*A^2*b^2*c^5
*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*
B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*
b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*
f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 -
80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 +
160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4
+ 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 +
C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2
*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^5*f^2 - 4*B^2*b^2*c^5*f^2 +
4*C^2*b^2*c^5*f^2 - 40*A^2*b^2*c^3*d^2*f^2 + 40*B^2*b^2*c^3*d^2*f^2 - 40*C^
2*b^2*c^3*d^2*f^2 + 8*A*B*b^2*d^5*f^2 - 8*A*C*b^2*c^5*f^2 - 8*B*C*b^2*d^5*f
^2 + 20*A^2*b^2*c*d^4*f^2 - 20*B^2*b^2*c*d^4*f^2 + 20*C^2*b^2*c*d^4*f^2 + 4
0*A*B*b^2*c^4*d*f^2 - 40*A*C*b^2*c*d^4*f^2 - 40*B*C*b^2*c^4*d*f^2 - 80*A*B*
b^2*c^2*d^3*f^2 + 80*A*C*b^2*c^3*d^2*f^2 + 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^1
0*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*
d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*A^2*b^2*c^5*f^2 - 8*B^2*
b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b^2*c^3*d
^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c^5*f^2 -
16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f^2 + 40*C^2*

b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^5*f^2 - 4*B^2*b^2*c^5*f^2 + 4*C^2*b^2*c^5*f^2 - 40*A^2*b^2*c^3*d^2*f^2 + 40*B^2*b^2*c^3*d^2*f^2 - 40*C^2*b^2*c^3*d^2*f^2 + 8*A*B*b^2*d^5*f^2 - 8*A*C*b^2*c^5*f^2 - 8*B*C*b^2*d^5*f^2 + 20*A^2*b^2*c*d^4*f^2 - 20*B^2*b^2*c*d^4*f^2 + 20*C^2*b^2*c*d^4*f^2 + 40*A*B*b^2*c^4*d*f^2 - 40*A*C*b^2*c*d^4*f^2 - 40*B*C*b^2*c^4*d*f^2 - 80*A*B*b^2*c^2*d^3*f^2 + 80*A*C*b^2*c^3*d^2*f^2 + 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 32*B*b*d^21*f^4 - 736*A*b*c^3*d^18*f^4 - 2432*A*b*c^5*d^16*f^4 - 4480*A*b*c^7*d^14*f^4 - 4928*A*b*c^9*d^12*f^4 - 3136*A*b*c^11*d^10*f^4 - 896*A*b*c^13*d^8*f^4 + 128*A*b*c^15*d^6*f^4 + 160*A*b*c^17*d^4*f^4 + 32*A*b*c^19*d^2*f^4 - 160*B*b*c^2*d^19*f^4 - 128*B*b*c^4*d^17*f^4 + 896*B*b*c^6*d^15*f^4 + 3136*B*b*c^8*d^13*f^4 + 4928*B*b*c^10*d^11*f^4 + 4480*B*b*c^12*d^9*f^4 + 2432*B*b*c^14*d^7*f^4 + 736*B*b*c^16*d^5*f^4 + 96*B*b*c^18*d^3*f^4 + 736*C*b*c^3*d^18*f^4 + 2432*C*b*c^5*d^16*f^4 + 4480*C*b*c^7*d^14*f^4 + 4928*C*b*c^9*d^12*f^4 + 3136*C*b*c^11*d^10*f^4 + 896*C*b*c^13*d^8*f^4 - 128*C*b*c^15*d^6*f^4 - 160*C*b*c^17*d^4*f^4 - 32*C*b*c^19*d^2*f^4 - 96*A*b*c*d^20*f^4 + 96*C*b*c*d^20*f^4))*((((8*A^2*b^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^5*f^2 - 4*B^2*b^2*c^5*f^2 + 4*C^2*b^2*c^5*f^2 - 40*A^2*b^2*c^3*d^2*f^2 + 40*B^2*b^2*c^3*d^2*f^2 - 40*C^2*b^2*c^3*d^2*f^2 + 8*A*B*b^2*d^5*f^2 - 8*A*C*b^2*c^5*f^2 - 8*B*C*b^2*d^5*f^2 + 20*A^2*b^2*c*d^4*f^2 - 20*B^2*b^2*c*d^4*f^2 + 20*C^2*b^2*c*d^4*f^2 + 40*A*B*b^2*c^4*d*f^2 - 40*A*C*b^2*c*d^4*f^2 - 40*B*C*b^2*c^4*d*f^2 - 80*A*B*b^2*c^2*d^3*f^2 + 80*A*C*b^2*c^3*d^2*f^2 + 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*1i)/(((c + d*tan(e + f*x))^(1/2)*(16*A^2*b^2*d^18*f^3 - 16*B^2*b^2*d^18*f^3 + 16*C^2*b^2*d^18*f^3 - 320*A^2*b^2*c^4*d^14*f^3 - 1024*A^2*b^2*c^6*d^12*f^3 - 1440*A^2*b^2*c^8*d^10*f^3 - 1024*A^2*b^2*c^10*d^8*f^3 - 320*A^2*b^2*c^12*d^6*f^3 + 16*A^2*b^2*c^16*d^2*f^3 + 320*B^2*b^2*c^4*d^14*f^3 + 1024*B^2*b^2*c^6*d^12*f^3 + 1440*B^2*b^2*c^8*d^10*f^3 + 1024*B^2*b^2*c^10*d^8*f^3 + 320*B^2*b^2*c^12*d^6*f^3 -

16*B^2*b^2*c^16*d^2*f^3 - 320*C^2*b^2*c^4*d^14*f^3 - 1024*C^2*b^2*c^6*d^12*f^3 - 1440*C^2*b^2*c^8*d^10*f^3 - 1024*C^2*b^2*c^10*d^8*f^3 - 320*C^2*b^2*c^12*d^6*f^3 + 16*C^2*b^2*c^16*d^2*f^3 - 32*A*C*b^2*d^18*f^3 - 128*A*B*b^2*c*d^17*f^3 + 128*B*C*b^2*c*d^17*f^3 - 640*A*B*b^2*c^3*d^15*f^3 - 1152*A*B*b^2*c^5*d^13*f^3 - 640*A*B*b^2*c^7*d^11*f^3 + 640*A*B*b^2*c^9*d^9*f^3 + 1152*A*B*b^2*c^11*d^7*f^3 + 640*A*B*b^2*c^13*d^5*f^3 + 128*A*B*b^2*c^15*d^3*f^3 + 640*A*C*b^2*c^4*d^14*f^3 + 2048*A*C*b^2*c^6*d^12*f^3 + 2880*A*C*b^2*c^8*d^10*f^3 + 2048*A*C*b^2*c^10*d^8*f^3 + 640*A*C*b^2*c^12*d^6*f^3 - 32*A*C*b^2*c^16*d^2*f^3 + 640*B*C*b^2*c^3*d^15*f^3 + 1152*B*C*b^2*c^5*d^13*f^3 + 640*B*C*b^2*c^7*d^11*f^3 - 640*B*C*b^2*c^9*d^9*f^3 - 1152*B*C*b^2*c^11*d^7*f^3 - 640*B*C*b^2*c^13*d^5*f^3 - 128*B*C*b^2*c^15*d^3*f^3) - ((((8*A^2*b^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^5*f^2 - 4*B^2*b^2*c^5*f^2 + 4*C^2*b^2*c^5*f^2 - 40*A^2*b^2*c^3*d^2*f^2 + 40*B^2*b^2*c^3*d^2*f^2 - 40*C^2*b^2*c^3*d^2*f^2 + 8*A*B*b^2*d^5*f^2 - 8*A*C*b^2*c^5*f^2 - 8*B*C*b^2*d^5*f^2 + 20*A^2*b^2*c*d^4*f^2 - 20*B^2*b^2*c*d^4*f^2 + 20*C^2*b^2*c*d^4*f^2 + 40*A*B*b^2*c^4*d*f^2 - 40*A*C*b^2*c*d^4*f^2 - 40*B*C*b^2*c^4*d*f^2 - 80*A*B*b^2*c^2*d^3*f^2 + 80*A*C*b^2*c^3*d^2*f^2 + 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*A^2*b^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^5*f^2 - 4*B^2*b^2*c^5*f^2 + 4*C^2*b^2*c^5*f^2 - 40*A^2*b^2*c^3*d^2*f^2 + 40*B^2*b^2*c^3*d^2*f^2 - 40*C^2*b^2*c^3*d^2*f^2 + 8*A*B*b^2*d^5*f^2 - 8*A*C*b^2*c^5*f^2 - 8*B*C*b^2*d^5*f^2 + 20*A^2*b^2*c*d^4*f^2 - 20*B^2*b^2*c*d^4*f^2 + 20*C^2*b^2*c*d^4*f^2 + 40*A*B*b^2*c^4*d*f^2 - 40*A*C*b^2*c*d^4*f^2 - 40*B*C*b^2*c^4*d*f^2 - 80*A*B*b^2*c^2*d^3*f^2 + 80*A*C*b^2*c^3*d^2*f^2 + 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 32*

B*b*d^21*f^4 - 736*A*b*c^3*d^18*f^4 - 2432*A*b*c^5*d^16*f^4 - 4480*A*b*c^7*
d^14*f^4 - 4928*A*b*c^9*d^12*f^4 - 3136*A*b*c^11*d^10*f^4 - 896*A*b*c^13*d^
8*f^4 + 128*A*b*c^15*d^6*f^4 + 160*A*b*c^17*d^4*f^4 + 32*A*b*c^19*d^2*f^4 -
160*B*b*c^2*d^19*f^4 - 128*B*b*c^4*d^17*f^4 + 896*B*b*c^6*d^15*f^4 + 3136*
B*b*c^8*d^13*f^4 + 4928*B*b*c^10*d^11*f^4 + 4480*B*b*c^12*d^9*f^4 + 2432*B*
b*c^14*d^7*f^4 + 736*B*b*c^16*d^5*f^4 + 96*B*b*c^18*d^3*f^4 + 736*C*b*c^3*d
^18*f^4 + 2432*C*b*c^5*d^16*f^4 + 4480*C*b*c^7*d^14*f^4 + 4928*C*b*c^9*d^12
*f^4 + 3136*C*b*c^11*d^10*f^4 + 896*C*b*c^13*d^8*f^4 - 128*C*b*c^15*d^6*f^4
- 160*C*b*c^17*d^4*f^4 - 32*C*b*c^19*d^2*f^4 - 96*A*b*c*d^20*f^4 + 96*C*b*
c*d^20*f^4))*((((8*A^2*b^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2
- 80*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2
+ 16*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2
*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d
*f^2 - 80*A*C*b^2*c*d^4*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^
2 + 160*A*C*b^2*c^3*d^2*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 +
16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*
d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B
^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*
c^5*f^2 - 4*B^2*b^2*c^5*f^2 + 4*C^2*b^2*c^5*f^2 - 40*A^2*b^2*c^3*d^2*f^2 +
40*B^2*b^2*c^3*d^2*f^2 - 40*C^2*b^2*c^3*d^2*f^2 + 8*A*B*b^2*d^5*f^2 - 8*A*C
*b^2*c^5*f^2 - 8*B*C*b^2*d^5*f^2 + 20*A^2*b^2*c*d^4*f^2 - 20*B^2*b^2*c*d^4*
f^2 + 20*C^2*b^2*c*d^4*f^2 + 40*A*B*b^2*c^4*d*f^2 - 40*A*C*b^2*c*d^4*f^2 -
40*B*C*b^2*c^4*d*f^2 - 80*A*B*b^2*c^2*d^3*f^2 + 80*A*C*b^2*c^3*d^2*f^2 + 80
*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6
*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2) - ((c + d*tan(e + f*x))^(1/2
)*(16*A^2*b^2*d^18*f^3 - 16*B^2*b^2*d^18*f^3 + 16*C^2*b^2*d^18*f^3 - 320*A^
2*b^2*c^4*d^14*f^3 - 1024*A^2*b^2*c^6*d^12*f^3 - 1440*A^2*b^2*c^8*d^10*f^3
- 1024*A^2*b^2*c^10*d^8*f^3 - 320*A^2*b^2*c^12*d^6*f^3 + 16*A^2*b^2*c^16*d^
2*f^3 + 320*B^2*b^2*c^4*d^14*f^3 + 1024*B^2*b^2*c^6*d^12*f^3 + 1440*B^2*b^2
*c^8*d^10*f^3 + 1024*B^2*b^2*c^10*d^8*f^3 + 320*B^2*b^2*c^12*d^6*f^3 - 16*B
^2*b^2*c^16*d^2*f^3 - 320*C^2*b^2*c^4*d^14*f^3 - 1024*C^2*b^2*c^6*d^12*f^3
- 1440*C^2*b^2*c^8*d^10*f^3 - 1024*C^2*b^2*c^10*d^8*f^3 - 320*C^2*b^2*c^12*
d^6*f^3 + 16*C^2*b^2*c^16*d^2*f^3 - 32*A*C*b^2*d^18*f^3 - 128*A*B*b^2*c*d^1
7*f^3 + 128*B*C*b^2*c*d^17*f^3 - 640*A*B*b^2*c^3*d^15*f^3 - 1152*A*B*b^2*c^
5*d^13*f^3 - 640*A*B*b^2*c^7*d^11*f^3 + 640*A*B*b^2*c^9*d^9*f^3 + 1152*A*B*
b^2*c^11*d^7*f^3 + 640*A*B*b^2*c^13*d^5*f^3 + 128*A*B*b^2*c^15*d^3*f^3 + 64
0*A*C*b^2*c^4*d^14*f^3 + 2048*A*C*b^2*c^6*d^12*f^3 + 2880*A*C*b^2*c^8*d^10*
f^3 + 2048*A*C*b^2*c^10*d^8*f^3 + 640*A*C*b^2*c^12*d^6*f^3 - 32*A*C*b^2*c^1
6*d^2*f^3 + 640*B*C*b^2*c^3*d^15*f^3 + 1152*B*C*b^2*c^5*d^13*f^3 + 640*B*C*
b^2*c^7*d^11*f^3 - 640*B*C*b^2*c^9*d^9*f^3 - 1152*B*C*b^2*c^11*d^7*f^3 - 64
0*B*C*b^2*c^13*d^5*f^3 - 128*B*C*b^2*c^15*d^3*f^3) + ((((8*A^2*b^2*c^5*f^2
- 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b
^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c
^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f^2 +
40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 - 80*B*

C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 + 160*B
*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160
*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b
^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*
b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^5*f^2 - 4*B^2*b^2*c^5*f^2 + 4*C^2
*b^2*c^5*f^2 - 40*A^2*b^2*c^3*d^2*f^2 + 40*B^2*b^2*c^3*d^2*f^2 - 40*C^2*b^2
*c^3*d^2*f^2 + 8*A*B*b^2*d^5*f^2 - 8*A*C*b^2*c^5*f^2 - 8*B*C*b^2*d^5*f^2 +
20*A^2*b^2*c*d^4*f^2 - 20*B^2*b^2*c*d^4*f^2 + 20*C^2*b^2*c*d^4*f^2 + 40*A*B
*b^2*c^4*d*f^2 - 40*A*C*b^2*c*d^4*f^2 - 40*B*C*b^2*c^4*d*f^2 - 80*A*B*b^2*c
^2*d^3*f^2 + 80*A*C*b^2*c^3*d^2*f^2 + 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4
 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f
^4)))^(1/2)*(128*A*b*c^15*d^6*f^4 - 32*B*b*d^21*f^4 - 736*A*b*c^3*d^18*f^4
- 2432*A*b*c^5*d^16*f^4 - 4480*A*b*c^7*d^14*f^4 - 4928*A*b*c^9*d^12*f^4 - 3
136*A*b*c^11*d^10*f^4 - 896*A*b*c^13*d^8*f^4 - (c + d*tan(e + f*x))^(1/2)*(
(((8*A^2*b^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c
^3*d^2*f^2 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d
^5*f^2 - 16*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d^4*f^2 - 4
0*B^2*b^2*c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2 - 80*A*C*
b^2*c*d^4*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 + 160*A*C*b^
2*c^3*d^2*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 +
 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*
b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2
*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^5*f^2 - 4*B^
2*b^2*c^5*f^2 + 4*C^2*b^2*c^5*f^2 - 40*A^2*b^2*c^3*d^2*f^2 + 40*B^2*b^2*c^3
*d^2*f^2 - 40*C^2*b^2*c^3*d^2*f^2 + 8*A*B*b^2*d^5*f^2 - 8*A*C*b^2*c^5*f^2 -
 8*B*C*b^2*d^5*f^2 + 20*A^2*b^2*c*d^4*f^2 - 20*B^2*b^2*c*d^4*f^2 + 20*C^2*b
^2*c*d^4*f^2 + 40*A*B*b^2*c^4*d*f^2 - 40*A*C*b^2*c*d^4*f^2 - 40*B*C*b^2*c^4
*d*f^2 - 80*A*B*b^2*c^2*d^3*f^2 + 80*A*C*b^2*c^3*d^2*f^2 + 80*B*C*b^2*c^2*d
^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*
d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c
^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5
+ 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^
4*f^5 + 64*c^21*d^2*f^5) + 160*A*b*c^17*d^4*f^4 + 32*A*b*c^19*d^2*f^4 - 160
*B*b*c^2*d^19*f^4 - 128*B*b*c^4*d^17*f^4 + 896*B*b*c^6*d^15*f^4 + 3136*B*b*
c^8*d^13*f^4 + 4928*B*b*c^10*d^11*f^4 + 4480*B*b*c^12*d^9*f^4 + 2432*B*b*c^
14*d^7*f^4 + 736*B*b*c^16*d^5*f^4 + 96*B*b*c^18*d^3*f^4 + 736*C*b*c^3*d^18*
f^4 + 2432*C*b*c^5*d^16*f^4 + 4480*C*b*c^7*d^14*f^4 + 4928*C*b*c^9*d^12*f^4
 + 3136*C*b*c^11*d^10*f^4 + 896*C*b*c^13*d^8*f^4 - 128*C*b*c^15*d^6*f^4 - 1
60*C*b*c^17*d^4*f^4 - 32*C*b*c^19*d^2*f^4 - 96*A*b*c*d^20*f^4 + 96*C*b*c*d^
20*f^4))*((((8*A^2*b^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80
*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16
*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d
^4*f^2 - 40*B^2*b^2*c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2
 - 80*A*C*b^2*c*d^4*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 +
160*A*C*b^2*c^3*d^2*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*

d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*
f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b
^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^5*
f^2 - 4*B^2*b^2*c^5*f^2 + 4*C^2*b^2*c^5*f^2 - 40*A^2*b^2*c^3*d^2*f^2 + 40*B
^2*b^2*c^3*d^2*f^2 - 40*C^2*b^2*c^3*d^2*f^2 + 8*A*B*b^2*d^5*f^2 - 8*A*C*b^2
*c^5*f^2 - 8*B*C*b^2*d^5*f^2 + 20*A^2*b^2*c*d^4*f^2 - 20*B^2*b^2*c*d^4*f^2
+ 20*C^2*b^2*c*d^4*f^2 + 40*A*B*b^2*c^4*d*f^2 - 40*A*C*b^2*c*d^4*f^2 - 40*B
*C*b^2*c^4*d*f^2 - 80*A*B*b^2*c^2*d^3*f^2 + 80*A*C*b^2*c^3*d^2*f^2 + 80*B*C
*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4
+ 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2) - 16*A^3*b^3*d^16*f^2 + 16*C^3*b
^3*d^16*f^2 - 80*A^3*b^3*c^2*d^14*f^2 - 144*A^3*b^3*c^4*d^12*f^2 - 80*A^3*b
^3*c^6*d^10*f^2 + 80*A^3*b^3*c^8*d^8*f^2 + 144*A^3*b^3*c^10*d^6*f^2 + 80*A^
3*b^3*c^12*d^4*f^2 + 16*A^3*b^3*c^14*d^2*f^2 + 192*B^3*b^3*c^3*d^13*f^2 + 4
80*B^3*b^3*c^5*d^11*f^2 + 640*B^3*b^3*c^7*d^9*f^2 + 480*B^3*b^3*c^9*d^7*f^2
+ 192*B^3*b^3*c^11*d^5*f^2 + 32*B^3*b^3*c^13*d^3*f^2 + 80*C^3*b^3*c^2*d^14
*f^2 + 144*C^3*b^3*c^4*d^12*f^2 + 80*C^3*b^3*c^6*d^10*f^2 - 80*C^3*b^3*c^8*
d^8*f^2 - 144*C^3*b^3*c^10*d^6*f^2 - 80*C^3*b^3*c^12*d^4*f^2 - 16*C^3*b^3*c
^14*d^2*f^2 - 16*A*B^2*b^3*d^16*f^2 - 48*A*C^2*b^3*d^16*f^2 + 48*A^2*C*b^3*
d^16*f^2 + 16*B^2*C*b^3*d^16*f^2 + 32*B^3*b^3*c*d^15*f^2 - 80*A*B^2*b^3*c^2
*d^14*f^2 - 144*A*B^2*b^3*c^4*d^12*f^2 - 80*A*B^2*b^3*c^6*d^10*f^2 + 80*A*B
^2*b^3*c^8*d^8*f^2 + 144*A*B^2*b^3*c^10*d^6*f^2 + 80*A*B^2*b^3*c^12*d^4*f^2
+ 16*A*B^2*b^3*c^14*d^2*f^2 + 192*A^2*B*b^3*c^3*d^13*f^2 + 480*A^2*B*b^3*c
^5*d^11*f^2 + 640*A^2*B*b^3*c^7*d^9*f^2 + 480*A^2*B*b^3*c^9*d^7*f^2 + 192*A
^2*B*b^3*c^11*d^5*f^2 + 32*A^2*B*b^3*c^13*d^3*f^2 - 240*A*C^2*b^3*c^2*d^14*
f^2 - 432*A*C^2*b^3*c^4*d^12*f^2 - 240*A*C^2*b^3*c^6*d^10*f^2 + 240*A*C^2*b
^3*c^8*d^8*f^2 + 432*A*C^2*b^3*c^10*d^6*f^2 + 240*A*C^2*b^3*c^12*d^4*f^2 +
48*A*C^2*b^3*c^14*d^2*f^2 + 240*A^2*C*b^3*c^2*d^14*f^2 + 432*A^2*C*b^3*c^4*
d^12*f^2 + 240*A^2*C*b^3*c^6*d^10*f^2 - 240*A^2*C*b^3*c^8*d^8*f^2 - 432*A^2
*C*b^3*c^10*d^6*f^2 - 240*A^2*C*b^3*c^12*d^4*f^2 - 48*A^2*C*b^3*c^14*d^2*f^
2 + 192*B*C^2*b^3*c^3*d^13*f^2 + 480*B*C^2*b^3*c^5*d^11*f^2 + 640*B*C^2*b^3
*c^7*d^9*f^2 + 480*B*C^2*b^3*c^9*d^7*f^2 + 192*B*C^2*b^3*c^11*d^5*f^2 + 32*
B*C^2*b^3*c^13*d^3*f^2 + 80*B^2*C*b^3*c^2*d^14*f^2 + 144*B^2*C*b^3*c^4*d^12
*f^2 + 80*B^2*C*b^3*c^6*d^10*f^2 - 80*B^2*C*b^3*c^8*d^8*f^2 - 144*B^2*C*b^3
*c^10*d^6*f^2 - 80*B^2*C*b^3*c^12*d^4*f^2 - 16*B^2*C*b^3*c^14*d^2*f^2 + 32*
A^2*B*b^3*c*d^15*f^2 + 32*B*C^2*b^3*c*d^15*f^2 - 384*A*B*C*b^3*c^3*d^13*f^2
- 960*A*B*C*b^3*c^5*d^11*f^2 - 1280*A*B*C*b^3*c^7*d^9*f^2 - 960*A*B*C*b^3*
c^9*d^7*f^2 - 384*A*B*C*b^3*c^11*d^5*f^2 - 64*A*B*C*b^3*c^13*d^3*f^2 - 64*A
*B*C*b^3*c*d^15*f^2))*((((8*A^2*b^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2
*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3
*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 4
0*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*
b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c
^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c
^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4
+ 80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4

+ 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4
*A^2*b^2*c^5*f^2 - 4*B^2*b^2*c^5*f^2 + 4*C^2*b^2*c^5*f^2 - 40*A^2*b^2*c^3*d
^2*f^2 + 40*B^2*b^2*c^3*d^2*f^2 - 40*C^2*b^2*c^3*d^2*f^2 + 8*A*B*b^2*d^5*f^
2 - 8*A*C*b^2*c^5*f^2 - 8*B*C*b^2*d^5*f^2 + 20*A^2*b^2*c*d^4*f^2 - 20*B^2*b
^2*c*d^4*f^2 + 20*C^2*b^2*c*d^4*f^2 + 40*A*B*b^2*c^4*d*f^2 - 40*A*C*b^2*c*d
^4*f^2 - 40*B*C*b^2*c^4*d*f^2 - 80*A*B*b^2*c^2*d^3*f^2 + 80*A*C*b^2*c^3*d^2
*f^2 + 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 1
0*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*2i - atan((((((8*A
^2*a^2*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2
*f^2 + 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2
 - 16*A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*
a^2*c*d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*
d^4*f^2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*
d^2*f^2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^
2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 +
B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a
^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^5*f^2 + 4*B^2*a^2*
c^5*f^2 - 4*C^2*a^2*c^5*f^2 + 40*A^2*a^2*c^3*d^2*f^2 - 40*B^2*a^2*c^3*d^2*f
^2 + 40*C^2*a^2*c^3*d^2*f^2 - 8*A*B*a^2*d^5*f^2 + 8*A*C*a^2*c^5*f^2 + 8*B*C
*a^2*d^5*f^2 - 20*A^2*a^2*c*d^4*f^2 + 20*B^2*a^2*c*d^4*f^2 - 20*C^2*a^2*c*d
^4*f^2 - 40*A*B*a^2*c^4*d*f^2 + 40*A*C*a^2*c*d^4*f^2 + 40*B*C*a^2*c^4*d*f^2
 + 80*A*B*a^2*c^2*d^3*f^2 - 80*A*C*a^2*c^3*d^2*f^2 - 80*B*C*a^2*c^2*d^3*f^2
)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^
4 + 5*c^8*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^2*c^5*f^
2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2
*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2
*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2
 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*
B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 160
*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 1
60*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4
*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^
2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^5*f^2 + 4*B^2*a^2*c^5*f^2 - 4*C
^2*a^2*c^5*f^2 + 40*A^2*a^2*c^3*d^2*f^2 - 40*B^2*a^2*c^3*d^2*f^2 + 40*C^2*a
^2*c^3*d^2*f^2 - 8*A*B*a^2*d^5*f^2 + 8*A*C*a^2*c^5*f^2 + 8*B*C*a^2*d^5*f^2
- 20*A^2*a^2*c*d^4*f^2 + 20*B^2*a^2*c*d^4*f^2 - 20*C^2*a^2*c*d^4*f^2 - 40*A
*B*a^2*c^4*d*f^2 + 40*A*C*a^2*c*d^4*f^2 + 40*B*C*a^2*c^4*d*f^2 + 80*A*B*a^2
*c^2*d^3*f^2 - 80*A*C*a^2*c^3*d^2*f^2 - 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f
^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2
f^4)))^(1/2)(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*
c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f
^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2
*f^5) - 32*A*a*d^21*f^4 + 32*C*a*d^21*f^4 - 160*A*a*c^2*d^19*f^4 - 128*A*a*
c^4*d^17*f^4 + 896*A*a*c^6*d^15*f^4 + 3136*A*a*c^8*d^13*f^4 + 4928*A*a*c^10
*d^11*f^4 + 4480*A*a*c^12*d^9*f^4 + 2432*A*a*c^14*d^7*f^4 + 736*A*a*c^16*d^

5*f^4 + 96*A*a*c^18*d^3*f^4 + 736*B*a*c^3*d^18*f^4 + 2432*B*a*c^5*d^16*f^4 + 4480*B*a*c^7*d^14*f^4 + 4928*B*a*c^9*d^12*f^4 + 3136*B*a*c^11*d^10*f^4 + 896*B*a*c^13*d^8*f^4 - 128*B*a*c^15*d^6*f^4 - 160*B*a*c^17*d^4*f^4 - 32*B*a*c^19*d^2*f^4 + 160*C*a*c^2*d^19*f^4 + 128*C*a*c^4*d^17*f^4 - 896*C*a*c^6*d^15*f^4 - 3136*C*a*c^8*d^13*f^4 - 4928*C*a*c^10*d^11*f^4 - 4480*C*a*c^12*d^9*f^4 - 2432*C*a*c^14*d^7*f^4 - 736*C*a*c^16*d^5*f^4 - 96*C*a*c^18*d^3*f^4 + 96*B*a*c*d^20*f^4) + (c + d*tan(e + f*x))^(1/2)*(16*A^2*a^2*d^18*f^3 - 16*B^2*a^2*d^18*f^3 + 16*C^2*a^2*d^18*f^3 - 320*A^2*a^2*c^4*d^14*f^3 - 1024*A^2*a^2*c^6*d^12*f^3 - 1440*A^2*a^2*c^8*d^10*f^3 - 1024*A^2*a^2*c^10*d^8*f^3 - 320*A^2*a^2*c^12*d^6*f^3 + 16*A^2*a^2*c^16*d^2*f^3 + 320*B^2*a^2*c^4*d^14*f^3 + 1024*B^2*a^2*c^6*d^12*f^3 + 1440*B^2*a^2*c^8*d^10*f^3 + 1024*B^2*a^2*c^10*d^8*f^3 + 320*B^2*a^2*c^12*d^6*f^3 - 16*B^2*a^2*c^16*d^2*f^3 - 320*C^2*a^2*c^4*d^14*f^3 - 1024*C^2*a^2*c^6*d^12*f^3 - 1440*C^2*a^2*c^8*d^10*f^3 - 1024*C^2*a^2*c^10*d^8*f^3 - 320*C^2*a^2*c^12*d^6*f^3 + 16*C^2*a^2*c^16*d^2*f^3 - 32*A*C*a^2*d^18*f^3 - 128*A*B*a^2*c*d^17*f^3 + 128*B*C*a^2*c*d^17*f^3 - 640*A*B*a^2*c^3*d^15*f^3 - 1152*A*B*a^2*c^5*d^13*f^3 - 640*A*B*a^2*c^7*d^11*f^3 + 640*A*B*a^2*c^9*d^9*f^3 + 1152*A*B*a^2*c^11*d^7*f^3 + 640*A*B*a^2*c^13*d^5*f^3 + 128*A*B*a^2*c^15*d^3*f^3 + 640*A*C*a^2*c^4*d^14*f^3 + 2048*A*C*a^2*c^6*d^12*f^3 + 2880*A*C*a^2*c^8*d^10*f^3 + 2048*A*C*a^2*c^10*d^8*f^3 + 640*A*C*a^2*c^12*d^6*f^3 - 32*A*C*a^2*c^16*d^2*f^3 + 640*B*C*a^2*c^3*d^15*f^3 + 1152*B*C*a^2*c^5*d^13*f^3 + 640*B*C*a^2*c^7*d^11*f^3 - 640*B*C*a^2*c^9*d^9*f^3 - 1152*B*C*a^2*c^11*d^7*f^3 - 640*B*C*a^2*c^13*d^5*f^3 - 128*B*C*a^2*c^15*d^3*f^3))*((((8*A^2*a^2*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^5*f^2 + 4*B^2*a^2*c^5*f^2 - 4*C^2*a^2*c^5*f^2 + 40*A^2*a^2*c^3*d^2*f^2 - 40*B^2*a^2*c^3*d^2*f^2 + 40*C^2*a^2*c^3*d^2*f^2 - 8*A*B*a^2*d^5*f^2 + 8*A*C*a^2*c^5*f^2 + 8*B*C*a^2*d^5*f^2 - 20*A^2*a^2*c*d^4*f^2 + 20*B^2*a^2*c*d^4*f^2 - 20*C^2*a^2*c*d^4*f^2 - 40*A*B*a^2*c^4*d*f^2 + 40*A*C*a^2*c*d^4*f^2 + 40*B*C*a^2*c^4*d*f^2 + 80*A*B*a^2*c^2*d^3*f^2 - 80*A*C*a^2*c^3*d^2*f^2 - 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*1i - (((((8*A^2*a^2*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4 *a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^5*f^2 + 4*B^2*a^2*c^5 *f^2 - 4*C^2*a^2*c^5*f^2 + 40*A^2*a^2*c^3*d^2*f^2 - 40*B^2*a^2*c^3*d^2*f^2 + 40*C^2*a^2*c^3*d^2*f^2 - 8*A*B*a^2*d^5*f^2 + 8*A*C*a^2*c^5*f^2 + 8*B*C*a^ 2*d^5*f^2 - 20*A^2*a^2*c*d^4*f^2 + 20*B^2*a^2*c*d^4*f^2 - 20*C^2*a^2*c*d^4* f^2 - 40*A*B*a^2*c^4*d*f^2 + 40*A*C*a^2*c*d^4*f^2 + 40*B*C*a^2*c^4*d*f^2 + 80*A*B*a^2*c^2*d^3*f^2 - 80*A*C*a^2*c^3*d^2*f^2 - 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(32*C*a*d^21*f^4 - 32*A*a*d^21*f^4 - (c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^2*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2 *c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d *f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^ 2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8* d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B ^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2* c^5*f^2 + 4*B^2*a^2*c^5*f^2 - 4*C^2*a^2*c^5*f^2 + 40*A^2*a^2*c^3*d^2*f^2 - 40*B^2*a^2*c^3*d^2*f^2 + 40*C^2*a^2*c^3*d^2*f^2 - 8*A*B*a^2*d^5*f^2 + 8*A*C *a^2*c^5*f^2 + 8*B*C*a^2*d^5*f^2 - 20*A^2*a^2*c*d^4*f^2 + 20*B^2*a^2*c*d^4* f^2 - 20*C^2*a^2*c*d^4*f^2 - 40*A*B*a^2*c^4*d*f^2 + 40*A*C*a^2*c*d^4*f^2 + 40*B*C*a^2*c^4*d*f^2 + 80*A*B*a^2*c^2*d^3*f^2 - 80*A*C*a^2*c^3*d^2*f^2 - 80 *B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6 *f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^2 0*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128* c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 160*A*a*c^2*d^19*f^4 - 128*A*a*c^4 *d^17*f^4 + 896*A*a*c^6*d^15*f^4 + 3136*A*a*c^8*d^13*f^4 + 4928*A*a*c^10*d^ 11*f^4 + 4480*A*a*c^12*d^9*f^4 + 2432*A*a*c^14*d^7*f^4 + 736*A*a*c^16*d^5*f ^4 + 96*A*a*c^18*d^3*f^4 + 736*B*a*c^3*d^18*f^4 + 2432*B*a*c^5*d^16*f^4 + 4 480*B*a*c^7*d^14*f^4 + 4928*B*a*c^9*d^12*f^4 + 3136*B*a*c^11*d^10*f^4 + 896 *B*a*c^13*d^8*f^4 - 128*B*a*c^15*d^6*f^4 - 160*B*a*c^17*d^4*f^4 - 32*B*a*c^ 19*d^2*f^4 + 160*C*a*c^2*d^19*f^4 + 128*C*a*c^4*d^17*f^4 - 896*C*a*c^6*d^15 *f^4 - 3136*C*a*c^8*d^13*f^4 - 4928*C*a*c^10*d^11*f^4 - 4480*C*a*c^12*d^9*f ^4 - 2432*C*a*c^14*d^7*f^4 - 736*C*a*c^16*d^5*f^4 - 96*C*a*c^18*d^3*f^4 + 9 6*B*a*c*d^20*f^4) - (c + d*tan(e + f*x))^(1/2)*(16*A^2*a^2*d^18*f^3 - 16*B^ 2*a^2*d^18*f^3 + 16*C^2*a^2*d^18*f^3 - 320*A^2*a^2*c^4*d^14*f^3 - 1024*A^2* a^2*c^6*d^12*f^3 - 1440*A^2*a^2*c^8*d^10*f^3 - 1024*A^2*a^2*c^10*d^8*f^3 - 320*A^2*a^2*c^12*d^6*f^3 + 16*A^2*a^2*c^16*d^2*f^3 + 320*B^2*a^2*c^4*d^14*f ^3 + 1024*B^2*a^2*c^6*d^12*f^3 + 1440*B^2*a^2*c^8*d^10*f^3 + 1024*B^2*a^2*c ^10*d^8*f^3 + 320*B^2*a^2*c^12*d^6*f^3 - 16*B^2*a^2*c^16*d^2*f^3 - 320*C^2* a^2*c^4*d^14*f^3 - 1024*C^2*a^2*c^6*d^12*f^3 - 1440*C^2*a^2*c^8*d^10*f^3 - 1024*C^2*a^2*c^10*d^8*f^3 - 320*C^2*a^2*c^12*d^6*f^3 + 16*C^2*a^2*c^16*d^2* f^3 - 32*A*C*a^2*d^18*f^3 - 128*A*B*a^2*c*d^17*f^3 + 128*B*C*a^2*c*d^17*f^3

- 640*A*B*a^2*c^3*d^15*f^3 - 1152*A*B*a^2*c^5*d^13*f^3 - 640*A*B*a^2*c^7*d^11*f^3 + 640*A*B*a^2*c^9*d^9*f^3 + 1152*A*B*a^2*c^11*d^7*f^3 + 640*A*B*a^2*c^13*d^5*f^3 + 128*A*B*a^2*c^15*d^3*f^3 + 640*A*C*a^2*c^4*d^14*f^3 + 2048*A*C*a^2*c^6*d^12*f^3 + 2880*A*C*a^2*c^8*d^10*f^3 + 2048*A*C*a^2*c^10*d^8*f^3 + 640*A*C*a^2*c^12*d^6*f^3 - 32*A*C*a^2*c^16*d^2*f^3 + 640*B*C*a^2*c^3*d^15*f^3 + 1152*B*C*a^2*c^5*d^13*f^3 + 640*B*C*a^2*c^7*d^11*f^3 - 640*B*C*a^2*c^9*d^9*f^3 - 1152*B*C*a^2*c^11*d^7*f^3 - 640*B*C*a^2*c^13*d^5*f^3 - 128*B*C*a^2*c^15*d^3*f^3))*((((8*A^2*a^2*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^5*f^2 + 4*B^2*a^2*c^5*f^2 - 4*C^2*a^2*c^5*f^2 + 40*A^2*a^2*c^3*d^2*f^2 - 40*B^2*a^2*c^3*d^2*f^2 + 40*C^2*a^2*c^3*d^2*f^2 - 8*A*B*a^2*d^5*f^2 + 8*A*C*a^2*c^5*f^2 + 8*B*C*a^2*d^5*f^2 - 20*A^2*a^2*c*d^4*f^2 + 20*B^2*a^2*c*d^4*f^2 - 20*C^2*a^2*c*d^4*f^2 - 40*A*B*a^2*c^4*d*f^2 + 40*A*C*a^2*c*d^4*f^2 + 40*B*C*a^2*c^4*d*f^2 + 80*A*B*a^2*c^2*d^3*f^2 - 80*A*C*a^2*c^3*d^2*f^2 - 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*1i)/((((((8*A^2*a^2*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^5*f^2 + 4*B^2*a^2*c^5*f^2 - 4*C^2*a^2*c^5*f^2 + 40*A^2*a^2*c^3*d^2*f^2 - 40*B^2*a^2*c^3*d^2*f^2 + 40*C^2*a^2*c^3*d^2*f^2 - 8*A*B*a^2*d^5*f^2 + 8*A*C*a^2*c^5*f^2 + 8*B*C*a^2*d^5*f^2 - 20*A^2*a^2*c*d^4*f^2 + 20*B^2*a^2*c*d^4*f^2 - 20*C^2*a^2*c*d^4*f^2 - 40*A*B*a^2*c^4*d*f^2 + 40*A*C*a^2*c*d^4*f^2 + 40*B*C*a^2*c^4*d*f^2 + 80*A*B*a^2*c^2*d^3*f^2 - 80*A*C*a^2*c^3*d^2*f^2 - 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^2*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4 *d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^5*f^2 + 4*B^2*a^2*c^5*f^2 - 4*C^2*a^2*c^5*f^2 + 40*A^2*a^2*c^3*d^2*f^2 - 40*B^2*a^2*c^3*d^2*f^2 + 40*C^2*a^2*c^3*d^2*f^2 - 8*A*B*a^2*d^5*f^2 + 8*A*C*a^2*c^5*f^2 + 8*B*C*a^2*d^5*f^2 - 20*A^2*a^2*c*d^4*f^2 + 20*B^2*a^2*c*d^4*f^2 - 20*C^2*a^2*c*d^4*f^2 - 40*A*B*a^2*c^4*d*f^2 + 40*A*C*a^2*c*d^4*f^2 + 40*B*C*a^2*c^4*d*f^2 + 80*A*B*a^2*c^2*d^3*f^2 - 80*A*C*a^2*c^3*d^2*f^2 - 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 32*A*a*d^21*f^4 + 32*C*a*d^21*f^4 - 160*A*a*c^2*d^19*f^4 - 128*A*a*c^4*d^17*f^4 + 896*A*a*c^6*d^15*f^4 + 3136*A*a*c^8*d^13*f^4 + 4928*A*a*c^10*d^11*f^4 + 4480*A*a*c^12*d^9*f^4 + 2432*A*a*c^14*d^7*f^4 + 736*A*a*c^16*d^5*f^4 + 96*A*a*c^18*d^3*f^4 + 736*B*a*c^3*d^18*f^4 + 2432*B*a*c^5*d^16*f^4 + 4480*B*a*c^7*d^14*f^4 + 4928*B*a*c^9*d^12*f^4 + 3136*B*a*c^11*d^10*f^4 + 896*B*a*c^13*d^8*f^4 - 128*B*a*c^15*d^6*f^4 - 160*B*a*c^17*d^4*f^4 - 32*B*a*c^19*d^2*f^4 + 160*C*a*c^2*d^19*f^4 + 128*C*a*c^4*d^17*f^4 - 896*C*a*c^6*d^15*f^4 - 3136*C*a*c^8*d^13*f^4 - 4928*C*a*c^10*d^11*f^4 - 4480*C*a*c^12*d^9*f^4 - 2432*C*a*c^14*d^7*f^4 - 736*C*a*c^16*d^5*f^4 - 96*C*a*c^18*d^3*f^4 + 96*B*a*c*d^20*f^4) + (c + d*tan(e + f*x))^(1/2)*(16*A^2*a^2*d^18*f^3 - 16*B^2*a^2*d^18*f^3 + 16*C^2*a^2*d^18*f^3 - 320*A^2*a^2*c^4*d^14*f^3 - 1024*A^2*a^2*c^6*d^12*f^3 - 1440*A^2*a^2*c^8*d^10*f^3 - 1024*A^2*a^2*c^10*d^8*f^3 - 320*A^2*a^2*c^12*d^6*f^3 + 16*A^2*a^2*c^16*d^2*f^3 + 320*B^2*a^2*c^4*d^14*f^3 + 1024*B^2*a^2*c^6*d^12*f^3 + 1440*B^2*a^2*c^8*d^10*f^3 + 1024*B^2*a^2*c^10*d^8*f^3 + 320*B^2*a^2*c^12*d^6*f^3 - 16*B^2*a^2*c^16*d^2*f^3 - 320*C^2*a^2*c^4*d^14*f^3 - 1024*C^2*a^2*c^6*d^12*f^3 - 1440*C^2*a^2*c^8*d^10*f^3 - 1024*C^2*a^2*c^10*d^8*f^3 - 320*C^2*a^2*c^12*d^6*f^3 + 16*C^2*a^2*c^16*d^2*f^3 - 32*A*C*a^2*d^18*f^3 - 128*A*B*a^2*c*d^17*f^3 + 128*B*C*a^2*c*d^17*f^3 - 640*A*B*a^2*c^3*d^15*f^3 - 1152*A*B*a^2*c^5*d^13*f^3 - 640*A*B*a^2*c^7*d^11*f^3 + 640*A*B*a^2*c^9*d^9*f^3 + 1152*A*B*a^2*c^11*d^7*f^3 + 640*A*B*a^2*c^13*d^5*f^3 + 128*A*B*a^2*c^15*d^3*f^3 + 640*A*C*a^2*c^4*d^14*f^3 + 2048*A*C*a^2*c^6*d^12*f^3 + 2880*A*C*a^2*c^8*d^10*f^3 + 2048*A*C*a^2*c^10*d^8*f^3 + 640*A*C*a^2*c^12*d^6*f^3 - 32*A*C*a^2*c^16*d^2*f^3 + 640*B*C*a^2*c^3*d^15*f^3 + 1152*B*C*a^2*c^5*d^13*f^3 + 640*B*C*a^2*c^7*d^11*f^3 - 640*B*C*a^2*c^9*d^9*f^3 - 1152*B*C*a^2*c^11*d^7*f^3 - 640*B*C*a^2*c^13*d^5*f^3 - 128*B*C*a^2*c^15*d^3*f^3))*((((8*A^2*a^2*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2

*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^5*f^2 + 4*B^2*a^2*c^5*f^2 - 4*C^2*a^2*c^5*f^2 + 40*A^2*a^2*c^3*d^2*f^2 - 40*B^2*a^2*c^3*d^2*f^2 + 40*C^2*a^2*c^3*d^2*f^2 - 8*A*B*a^2*d^5*f^2 + 8*A*C*a^2*c^5*f^2 + 8*B*C*a^2*d^5*f^2 - 20*A^2*a^2*c*d^4*f^2 + 20*B^2*a^2*c*d^4*f^2 - 20*C^2*a^2*c*d^4*f^2 - 40*A*B*a^2*c^4*d*f^2 + 40*A*C*a^2*c*d^4*f^2 + 40*B*C*a^2*c^4*d*f^2 + 80*A*B*a^2*c^2*d^3*f^2 - 80*A*C*a^2*c^3*d^2*f^2 - 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2) + (((((8*A^2*a^2*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^5*f^2 + 4*B^2*a^2*c^5*f^2 - 4*C^2*a^2*c^5*f^2 + 40*A^2*a^2*c^3*d^2*f^2 - 40*B^2*a^2*c^3*d^2*f^2 + 40*C^2*a^2*c^3*d^2*f^2 - 8*A*B*a^2*d^5*f^2 + 8*A*C*a^2*c^5*f^2 + 8*B*C*a^2*d^5*f^2 - 20*A^2*a^2*c*d^4*f^2 + 20*B^2*a^2*c*d^4*f^2 - 20*C^2*a^2*c*d^4*f^2 - 40*A*B*a^2*c^4*d*f^2 + 40*A*C*a^2*c*d^4*f^2 + 40*B*C*a^2*c^4*d*f^2 + 80*A*B*a^2*c^2*d^3*f^2 - 80*A*C*a^2*c^3*d^2*f^2 - 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(32*C*a*d^21*f^4 - 32*A*a*d^21*f^4 - (c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^2*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^5*f^2 + 4*B^2*a^2*c^5*f^2 - 4*C^2*a^2*c^5*f^2 + 40*A^2*a^2*c^3*d^2*f^2 - 40*B^2*a^2*c^3*d^2*f^2 + 40*C^2*a^2*c^3*d^2*f^2 - 8*A*B*a^2*d^5*f^2 + 8*A*C*a^2*c^5*f^2 + 8*B*C*a^2*d^5*f^2 - 20*A^2*a^2*c*d^4*f^2 + 20*B^2*a^2*c*d^4*f^2 - 20*C^2*a^2*c*d^4*f^2 - 40*A*B*a^2*c^4*d*f^2 + 40*A*C*a^2*c*d^4*f^2 + 40*B*C*a^2*c^4*d*f^2 + 80*A*B*a^2*c^2*d^3*f^2 - 80*A*C*a^2*c^3*d^2*f^2 - 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 160*A*a*c^2*d^19*f^4 - 128*A*a*c^4*d^17*f^4 + 896*A*a*c^6*d^15*f^4 + 3136*A*a*c^8*d^13*f^4 + 4928*A*a*c^10*d^11*f^4 + 4480*A*a*c^12*d^9*f^4 + 2432*A*a*c^14*d^7*f^4 + 736*A*a*c^16*d^5*f^4 + 96*A

*a*c^18*d^3*f^4 + 736*B*a*c^3*d^18*f^4 + 2432*B*a*c^5*d^16*f^4 + 4480*B*a*c
^7*d^14*f^4 + 4928*B*a*c^9*d^12*f^4 + 3136*B*a*c^11*d^10*f^4 + 896*B*a*c^13
*d^8*f^4 - 128*B*a*c^15*d^6*f^4 - 160*B*a*c^17*d^4*f^4 - 32*B*a*c^19*d^2*f^
4 + 160*C*a*c^2*d^19*f^4 + 128*C*a*c^4*d^17*f^4 - 896*C*a*c^6*d^15*f^4 - 31
36*C*a*c^8*d^13*f^4 - 4928*C*a*c^10*d^11*f^4 - 4480*C*a*c^12*d^9*f^4 - 2432
*C*a*c^14*d^7*f^4 - 736*C*a*c^16*d^5*f^4 - 96*C*a*c^18*d^3*f^4 + 96*B*a*c*d
^20*f^4) - (c + d*tan(e + f*x))^(1/2)*(16*A^2*a^2*d^18*f^3 - 16*B^2*a^2*d^1
8*f^3 + 16*C^2*a^2*d^18*f^3 - 320*A^2*a^2*c^4*d^14*f^3 - 1024*A^2*a^2*c^6*d
^12*f^3 - 1440*A^2*a^2*c^8*d^10*f^3 - 1024*A^2*a^2*c^10*d^8*f^3 - 320*A^2*a
^2*c^12*d^6*f^3 + 16*A^2*a^2*c^16*d^2*f^3 + 320*B^2*a^2*c^4*d^14*f^3 + 1024
*B^2*a^2*c^6*d^12*f^3 + 1440*B^2*a^2*c^8*d^10*f^3 + 1024*B^2*a^2*c^10*d^8*f
^3 + 320*B^2*a^2*c^12*d^6*f^3 - 16*B^2*a^2*c^16*d^2*f^3 - 320*C^2*a^2*c^4*d
^14*f^3 - 1024*C^2*a^2*c^6*d^12*f^3 - 1440*C^2*a^2*c^8*d^10*f^3 - 1024*C^2*
a^2*c^10*d^8*f^3 - 320*C^2*a^2*c^12*d^6*f^3 + 16*C^2*a^2*c^16*d^2*f^3 - 32*
A*C*a^2*d^18*f^3 - 128*A*B*a^2*c*d^17*f^3 + 128*B*C*a^2*c*d^17*f^3 - 640*A*
B*a^2*c^3*d^15*f^3 - 1152*A*B*a^2*c^5*d^13*f^3 - 640*A*B*a^2*c^7*d^11*f^3 +
 640*A*B*a^2*c^9*d^9*f^3 + 1152*A*B*a^2*c^11*d^7*f^3 + 640*A*B*a^2*c^13*d^5
*f^3 + 128*A*B*a^2*c^15*d^3*f^3 + 640*A*C*a^2*c^4*d^14*f^3 + 2048*A*C*a^2*c
^6*d^12*f^3 + 2880*A*C*a^2*c^8*d^10*f^3 + 2048*A*C*a^2*c^10*d^8*f^3 + 640*A
*C*a^2*c^12*d^6*f^3 - 32*A*C*a^2*c^16*d^2*f^3 + 640*B*C*a^2*c^3*d^15*f^3 +
1152*B*C*a^2*c^5*d^13*f^3 + 640*B*C*a^2*c^7*d^11*f^3 - 640*B*C*a^2*c^9*d^9*
f^3 - 1152*B*C*a^2*c^11*d^7*f^3 - 640*B*C*a^2*c^13*d^5*f^3 - 128*B*C*a^2*c^
15*d^3*f^3))*((((8*A^2*a^2*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2
- 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2
+ 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2
*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d
*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^
2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 +
 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*
d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B
^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*
c^5*f^2 + 4*B^2*a^2*c^5*f^2 - 4*C^2*a^2*c^5*f^2 + 40*A^2*a^2*c^3*d^2*f^2 -
40*B^2*a^2*c^3*d^2*f^2 + 40*C^2*a^2*c^3*d^2*f^2 - 8*A*B*a^2*d^5*f^2 + 8*A*C
*a^2*c^5*f^2 + 8*B*C*a^2*d^5*f^2 - 20*A^2*a^2*c*d^4*f^2 + 20*B^2*a^2*c*d^4*
f^2 - 20*C^2*a^2*c*d^4*f^2 - 40*A*B*a^2*c^4*d*f^2 + 40*A*C*a^2*c*d^4*f^2 +
40*B*C*a^2*c^4*d*f^2 + 80*A*B*a^2*c^2*d^3*f^2 - 80*A*C*a^2*c^3*d^2*f^2 - 80
*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6
*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2) - 16*B^3*a^3*d^16*f^2 - 192*
A^3*a^3*c^3*d^13*f^2 - 480*A^3*a^3*c^5*d^11*f^2 - 640*A^3*a^3*c^7*d^9*f^2 -
 480*A^3*a^3*c^9*d^7*f^2 - 192*A^3*a^3*c^11*d^5*f^2 - 32*A^3*a^3*c^13*d^3*f
^2 - 80*B^3*a^3*c^2*d^14*f^2 - 144*B^3*a^3*c^4*d^12*f^2 - 80*B^3*a^3*c^6*d^
10*f^2 + 80*B^3*a^3*c^8*d^8*f^2 + 144*B^3*a^3*c^10*d^6*f^2 + 80*B^3*a^3*c^1
2*d^4*f^2 + 16*B^3*a^3*c^14*d^2*f^2 + 192*C^3*a^3*c^3*d^13*f^2 + 480*C^3*a^
3*c^5*d^11*f^2 + 640*C^3*a^3*c^7*d^9*f^2 + 480*C^3*a^3*c^9*d^7*f^2 + 192*C^
3*a^3*c^11*d^5*f^2 + 32*C^3*a^3*c^13*d^3*f^2 - 16*A^2*B*a^3*d^16*f^2 - 16*B

*C^2*a^3*d^16*f^2 - 32*A^3*a^3*c*d^15*f^2 + 32*C^3*a^3*c*d^15*f^2 - 192*A*B
^2*a^3*c^3*d^13*f^2 - 480*A*B^2*a^3*c^5*d^11*f^2 - 640*A*B^2*a^3*c^7*d^9*f^
2 - 480*A*B^2*a^3*c^9*d^7*f^2 - 192*A*B^2*a^3*c^11*d^5*f^2 - 32*A*B^2*a^3*c
^13*d^3*f^2 - 80*A^2*B*a^3*c^2*d^14*f^2 - 144*A^2*B*a^3*c^4*d^12*f^2 - 80*A
^2*B*a^3*c^6*d^10*f^2 + 80*A^2*B*a^3*c^8*d^8*f^2 + 144*A^2*B*a^3*c^10*d^6*f
^2 + 80*A^2*B*a^3*c^12*d^4*f^2 + 16*A^2*B*a^3*c^14*d^2*f^2 - 576*A*C^2*a^3*
c^3*d^13*f^2 - 1440*A*C^2*a^3*c^5*d^11*f^2 - 1920*A*C^2*a^3*c^7*d^9*f^2 - 1
440*A*C^2*a^3*c^9*d^7*f^2 - 576*A*C^2*a^3*c^11*d^5*f^2 - 96*A*C^2*a^3*c^13*
d^3*f^2 + 576*A^2*C*a^3*c^3*d^13*f^2 + 1440*A^2*C*a^3*c^5*d^11*f^2 + 1920*A
^2*C*a^3*c^7*d^9*f^2 + 1440*A^2*C*a^3*c^9*d^7*f^2 + 576*A^2*C*a^3*c^11*d^5*
f^2 + 96*A^2*C*a^3*c^13*d^3*f^2 - 80*B*C^2*a^3*c^2*d^14*f^2 - 144*B*C^2*a^3
*c^4*d^12*f^2 - 80*B*C^2*a^3*c^6*d^10*f^2 + 80*B*C^2*a^3*c^8*d^8*f^2 + 144*
B*C^2*a^3*c^10*d^6*f^2 + 80*B*C^2*a^3*c^12*d^4*f^2 + 16*B*C^2*a^3*c^14*d^2*
f^2 + 192*B^2*C*a^3*c^3*d^13*f^2 + 480*B^2*C*a^3*c^5*d^11*f^2 + 640*B^2*C*a
^3*c^7*d^9*f^2 + 480*B^2*C*a^3*c^9*d^7*f^2 + 192*B^2*C*a^3*c^11*d^5*f^2 + 3
2*B^2*C*a^3*c^13*d^3*f^2 + 32*A*B*C*a^3*d^16*f^2 - 32*A*B^2*a^3*c*d^15*f^2
- 96*A*C^2*a^3*c*d^15*f^2 + 96*A^2*C*a^3*c*d^15*f^2 + 32*B^2*C*a^3*c*d^15*f
^2 + 160*A*B*C*a^3*c^2*d^14*f^2 + 288*A*B*C*a^3*c^4*d^12*f^2 + 160*A*B*C*a^
3*c^6*d^10*f^2 - 160*A*B*C*a^3*c^8*d^8*f^2 - 288*A*B*C*a^3*c^10*d^6*f^2 - 1
60*A*B*C*a^3*c^12*d^4*f^2 - 32*A*B*C*a^3*c^14*d^2*f^2))*((((8*A^2*a^2*c^5*f
^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^
2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^
2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^
2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80
*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 16
0*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 +
160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^
4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C
^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^5*f^2 + 4*B^2*a^2*c^5*f^2 - 4*
C^2*a^2*c^5*f^2 + 40*A^2*a^2*c^3*d^2*f^2 - 40*B^2*a^2*c^3*d^2*f^2 + 40*C^2*
a^2*c^3*d^2*f^2 - 8*A*B*a^2*d^5*f^2 + 8*A*C*a^2*c^5*f^2 + 8*B*C*a^2*d^5*f^2
- 20*A^2*a^2*c*d^4*f^2 + 20*B^2*a^2*c*d^4*f^2 - 20*C^2*a^2*c*d^4*f^2 - 40*
A*B*a^2*c^4*d*f^2 + 40*A*C*a^2*c*d^4*f^2 + 40*B*C*a^2*c^4*d*f^2 + 80*A*B*a^
2*c^2*d^3*f^2 - 80*A*C*a^2*c^3*d^2*f^2 - 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*
f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^
2*f^4)))^(1/2)*2i - atan((((-(((8*A^2*a^2*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C
^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a
^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f
^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 8
0*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B
*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 -
(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^
4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*
C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/
2) + 4*A^2*a^2*c^5*f^2 - 4*B^2*a^2*c^5*f^2 + 4*C^2*a^2*c^5*f^2 - 40*A^2*a^2

*c^3*d^2*f^2 + 40*B^2*a^2*c^3*d^2*f^2 - 40*C^2*a^2*c^3*d^2*f^2 + 8*A*B*a^2*
d^5*f^2 - 8*A*C*a^2*c^5*f^2 - 8*B*C*a^2*d^5*f^2 + 20*A^2*a^2*c*d^4*f^2 - 20
*B^2*a^2*c*d^4*f^2 + 20*C^2*a^2*c*d^4*f^2 + 40*A*B*a^2*c^4*d*f^2 - 40*A*C*a
^2*c*d^4*f^2 - 40*B*C*a^2*c^4*d*f^2 - 80*A*B*a^2*c^2*d^3*f^2 + 80*A*C*a^2*c
^3*d^2*f^2 + 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f
^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*((c + d*tan(e
 + f*x))^(1/2)*(-(((8*A^2*a^2*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f
^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f
^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*
a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^
4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3
*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^
4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c
^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^
2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a
^2*c^5*f^2 - 4*B^2*a^2*c^5*f^2 + 4*C^2*a^2*c^5*f^2 - 40*A^2*a^2*c^3*d^2*f^2
 + 40*B^2*a^2*c^3*d^2*f^2 - 40*C^2*a^2*c^3*d^2*f^2 + 8*A*B*a^2*d^5*f^2 - 8*
A*C*a^2*c^5*f^2 - 8*B*C*a^2*d^5*f^2 + 20*A^2*a^2*c*d^4*f^2 - 20*B^2*a^2*c*d
^4*f^2 + 20*C^2*a^2*c*d^4*f^2 + 40*A*B*a^2*c^4*d*f^2 - 40*A*C*a^2*c*d^4*f^2
 - 40*B*C*a^2*c^4*d*f^2 - 80*A*B*a^2*c^2*d^3*f^2 + 80*A*C*a^2*c^3*d^2*f^2 +
 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*
d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*
d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 161
28*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*
f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 32*A*a*d^21*f^4 + 32*C*a*d^21*f
^4 - 160*A*a*c^2*d^19*f^4 - 128*A*a*c^4*d^17*f^4 + 896*A*a*c^6*d^15*f^4 + 3
136*A*a*c^8*d^13*f^4 + 4928*A*a*c^10*d^11*f^4 + 4480*A*a*c^12*d^9*f^4 + 243
2*A*a*c^14*d^7*f^4 + 736*A*a*c^16*d^5*f^4 + 96*A*a*c^18*d^3*f^4 + 736*B*a*c
^3*d^18*f^4 + 2432*B*a*c^5*d^16*f^4 + 4480*B*a*c^7*d^14*f^4 + 4928*B*a*c^9*
d^12*f^4 + 3136*B*a*c^11*d^10*f^4 + 896*B*a*c^13*d^8*f^4 - 128*B*a*c^15*d^6
*f^4 - 160*B*a*c^17*d^4*f^4 - 32*B*a*c^19*d^2*f^4 + 160*C*a*c^2*d^19*f^4 +
128*C*a*c^4*d^17*f^4 - 896*C*a*c^6*d^15*f^4 - 3136*C*a*c^8*d^13*f^4 - 4928*
C*a*c^10*d^11*f^4 - 4480*C*a*c^12*d^9*f^4 - 2432*C*a*c^14*d^7*f^4 - 736*C*a
*c^16*d^5*f^4 - 96*C*a*c^18*d^3*f^4 + 96*B*a*c*d^20*f^4) + (c + d*tan(e + f
x))^(1/2)(16*A^2*a^2*d^18*f^3 - 16*B^2*a^2*d^18*f^3 + 16*C^2*a^2*d^18*f^3
 - 320*A^2*a^2*c^4*d^14*f^3 - 1024*A^2*a^2*c^6*d^12*f^3 - 1440*A^2*a^2*c^8*
d^10*f^3 - 1024*A^2*a^2*c^10*d^8*f^3 - 320*A^2*a^2*c^12*d^6*f^3 + 16*A^2*a^
2*c^16*d^2*f^3 + 320*B^2*a^2*c^4*d^14*f^3 + 1024*B^2*a^2*c^6*d^12*f^3 + 144
0*B^2*a^2*c^8*d^10*f^3 + 1024*B^2*a^2*c^10*d^8*f^3 + 320*B^2*a^2*c^12*d^6*f
^3 - 16*B^2*a^2*c^16*d^2*f^3 - 320*C^2*a^2*c^4*d^14*f^3 - 1024*C^2*a^2*c^6*
d^12*f^3 - 1440*C^2*a^2*c^8*d^10*f^3 - 1024*C^2*a^2*c^10*d^8*f^3 - 320*C^2*
a^2*c^12*d^6*f^3 + 16*C^2*a^2*c^16*d^2*f^3 - 32*A*C*a^2*d^18*f^3 - 128*A*B*
a^2*c*d^17*f^3 + 128*B*C*a^2*c*d^17*f^3 - 640*A*B*a^2*c^3*d^15*f^3 - 1152*A
*B*a^2*c^5*d^13*f^3 - 640*A*B*a^2*c^7*d^11*f^3 + 640*A*B*a^2*c^9*d^9*f^3 +
1152*A*B*a^2*c^11*d^7*f^3 + 640*A*B*a^2*c^13*d^5*f^3 + 128*A*B*a^2*c^15*d^3

*f^3 + 640*A*C*a^2*c^4*d^14*f^3 + 2048*A*C*a^2*c^6*d^12*f^3 + 2880*A*C*a^2*
c^8*d^10*f^3 + 2048*A*C*a^2*c^10*d^8*f^3 + 640*A*C*a^2*c^12*d^6*f^3 - 32*A*
C*a^2*c^16*d^2*f^3 + 640*B*C*a^2*c^3*d^15*f^3 + 1152*B*C*a^2*c^5*d^13*f^3 +
640*B*C*a^2*c^7*d^11*f^3 - 640*B*C*a^2*c^9*d^9*f^3 - 1152*B*C*a^2*c^11*d^7
*f^3 - 640*B*C*a^2*c^13*d^5*f^3 - 128*B*C*a^2*c^15*d^3*f^3))*(-(((8*A^2*a^2
*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 +
80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*
A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*
d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^
2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^
2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*
f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^
4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2
*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^5*f^2 - 4*B^2*a^2*c^5*f^
2 + 4*C^2*a^2*c^5*f^2 - 40*A^2*a^2*c^3*d^2*f^2 + 40*B^2*a^2*c^3*d^2*f^2 - 4
0*C^2*a^2*c^3*d^2*f^2 + 8*A*B*a^2*d^5*f^2 - 8*A*C*a^2*c^5*f^2 - 8*B*C*a^2*d
^5*f^2 + 20*A^2*a^2*c*d^4*f^2 - 20*B^2*a^2*c*d^4*f^2 + 20*C^2*a^2*c*d^4*f^2
+ 40*A*B*a^2*c^4*d*f^2 - 40*A*C*a^2*c*d^4*f^2 - 40*B*C*a^2*c^4*d*f^2 - 80*
A*B*a^2*c^2*d^3*f^2 + 80*A*C*a^2*c^3*d^2*f^2 + 80*B*C*a^2*c^2*d^3*f^2)/(16*
(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*
c^8*d^2*f^4)))^(1/2)*1i - ((-(((8*A^2*a^2*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C
^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a
^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f
^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 8
0*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B
*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 -
(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^
4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*
C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/
2) + 4*A^2*a^2*c^5*f^2 - 4*B^2*a^2*c^5*f^2 + 4*C^2*a^2*c^5*f^2 - 40*A^2*a^2
*c^3*d^2*f^2 + 40*B^2*a^2*c^3*d^2*f^2 - 40*C^2*a^2*c^3*d^2*f^2 + 8*A*B*a^2*
d^5*f^2 - 8*A*C*a^2*c^5*f^2 - 8*B*C*a^2*d^5*f^2 + 20*A^2*a^2*c*d^4*f^2 - 20
*B^2*a^2*c*d^4*f^2 + 20*C^2*a^2*c*d^4*f^2 + 40*A*B*a^2*c^4*d*f^2 - 40*A*C*a
^2*c*d^4*f^2 - 40*B*C*a^2*c^4*d*f^2 - 80*A*B*a^2*c^2*d^3*f^2 + 80*A*C*a^2*c
^3*d^2*f^2 + 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f
^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(32*C*a*d^21*
f^4 - 32*A*a*d^21*f^4 - (c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^2*c^5*f^2 -
8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^
2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^
5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 +
40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C
*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*
C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*
c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^
4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a

^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^5*f^2 - 4*B^2*a^2*c^5*f^2 + 4*C^2*
a^2*c^5*f^2 - 40*A^2*a^2*c^3*d^2*f^2 + 40*B^2*a^2*c^3*d^2*f^2 - 40*C^2*a^2*
c^3*d^2*f^2 + 8*A*B*a^2*d^5*f^2 - 8*A*C*a^2*c^5*f^2 - 8*B*C*a^2*d^5*f^2 + 2
0*A^2*a^2*c*d^4*f^2 - 20*B^2*a^2*c*d^4*f^2 + 20*C^2*a^2*c*d^4*f^2 + 40*A*B*
a^2*c^4*d*f^2 - 40*A*C*a^2*c*d^4*f^2 - 40*B*C*a^2*c^4*d*f^2 - 80*A*B*a^2*c^
2*d^3*f^2 + 80*A*C*a^2*c^3*d^2*f^2 + 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4
+ d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^
4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7
*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5
+ 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^
5) - 160*A*a*c^2*d^19*f^4 - 128*A*a*c^4*d^17*f^4 + 896*A*a*c^6*d^15*f^4 + 3
136*A*a*c^8*d^13*f^4 + 4928*A*a*c^10*d^11*f^4 + 4480*A*a*c^12*d^9*f^4 + 243
2*A*a*c^14*d^7*f^4 + 736*A*a*c^16*d^5*f^4 + 96*A*a*c^18*d^3*f^4 + 736*B*a*c
^3*d^18*f^4 + 2432*B*a*c^5*d^16*f^4 + 4480*B*a*c^7*d^14*f^4 + 4928*B*a*c^9*
d^12*f^4 + 3136*B*a*c^11*d^10*f^4 + 896*B*a*c^13*d^8*f^4 - 128*B*a*c^15*d^6
*f^4 - 160*B*a*c^17*d^4*f^4 - 32*B*a*c^19*d^2*f^4 + 160*C*a*c^2*d^19*f^4 +
128*C*a*c^4*d^17*f^4 - 896*C*a*c^6*d^15*f^4 - 3136*C*a*c^8*d^13*f^4 - 4928*
C*a*c^10*d^11*f^4 - 4480*C*a*c^12*d^9*f^4 - 2432*C*a*c^14*d^7*f^4 - 736*C*a
*c^16*d^5*f^4 - 96*C*a*c^18*d^3*f^4 + 96*B*a*c*d^20*f^4) - (c + d*tan(e + f
x))^(1/2)(16*A^2*a^2*d^18*f^3 - 16*B^2*a^2*d^18*f^3 + 16*C^2*a^2*d^18*f^3
 - 320*A^2*a^2*c^4*d^14*f^3 - 1024*A^2*a^2*c^6*d^12*f^3 - 1440*A^2*a^2*c^8*
d^10*f^3 - 1024*A^2*a^2*c^10*d^8*f^3 - 320*A^2*a^2*c^12*d^6*f^3 + 16*A^2*a^
2*c^16*d^2*f^3 + 320*B^2*a^2*c^4*d^14*f^3 + 1024*B^2*a^2*c^6*d^12*f^3 + 144
0*B^2*a^2*c^8*d^10*f^3 + 1024*B^2*a^2*c^10*d^8*f^3 + 320*B^2*a^2*c^12*d^6*f
^3 - 16*B^2*a^2*c^16*d^2*f^3 - 320*C^2*a^2*c^4*d^14*f^3 - 1024*C^2*a^2*c^6*
d^12*f^3 - 1440*C^2*a^2*c^8*d^10*f^3 - 1024*C^2*a^2*c^10*d^8*f^3 - 320*C^2*
a^2*c^12*d^6*f^3 + 16*C^2*a^2*c^16*d^2*f^3 - 32*A*C*a^2*d^18*f^3 - 128*A*B*
a^2*c*d^17*f^3 + 128*B*C*a^2*c*d^17*f^3 - 640*A*B*a^2*c^3*d^15*f^3 - 1152*A
*B*a^2*c^5*d^13*f^3 - 640*A*B*a^2*c^7*d^11*f^3 + 640*A*B*a^2*c^9*d^9*f^3 +
1152*A*B*a^2*c^11*d^7*f^3 + 640*A*B*a^2*c^13*d^5*f^3 + 128*A*B*a^2*c^15*d^3
*f^3 + 640*A*C*a^2*c^4*d^14*f^3 + 2048*A*C*a^2*c^6*d^12*f^3 + 2880*A*C*a^2*
c^8*d^10*f^3 + 2048*A*C*a^2*c^10*d^8*f^3 + 640*A*C*a^2*c^12*d^6*f^3 - 32*A*
C*a^2*c^16*d^2*f^3 + 640*B*C*a^2*c^3*d^15*f^3 + 1152*B*C*a^2*c^5*d^13*f^3 +
 640*B*C*a^2*c^7*d^11*f^3 - 640*B*C*a^2*c^9*d^9*f^3 - 1152*B*C*a^2*c^11*d^7
*f^3 - 640*B*C*a^2*c^13*d^5*f^3 - 128*B*C*a^2*c^15*d^3*f^3))*(-(((8*A^2*a^2
*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 +
 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*
A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*
d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^
2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^
2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*
f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^
4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2
*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^5*f^2 - 4*B^2*a^2*c^5*f^
2 + 4*C^2*a^2*c^5*f^2 - 40*A^2*a^2*c^3*d^2*f^2 + 40*B^2*a^2*c^3*d^2*f^2 - 4

0*C^2*a^2*c^3*d^2*f^2 + 8*A*B*a^2*d^5*f^2 - 8*A*C*a^2*c^5*f^2 - 8*B*C*a^2*d
^5*f^2 + 20*A^2*a^2*c*d^4*f^2 - 20*B^2*a^2*c*d^4*f^2 + 20*C^2*a^2*c*d^4*f^2
 + 40*A*B*a^2*c^4*d*f^2 - 40*A*C*a^2*c*d^4*f^2 - 40*B*C*a^2*c^4*d*f^2 - 80*
A*B*a^2*c^2*d^3*f^2 + 80*A*C*a^2*c^3*d^2*f^2 + 80*B*C*a^2*c^2*d^3*f^2)/(16*
(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*
c^8*d^2*f^4)))^(1/2)*1i)/(((-(((8*A^2*a^2*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C
^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a
^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f
^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 8
0*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B
*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 -
 (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^
4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*
C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/
2) + 4*A^2*a^2*c^5*f^2 - 4*B^2*a^2*c^5*f^2 + 4*C^2*a^2*c^5*f^2 - 40*A^2*a^2
*c^3*d^2*f^2 + 40*B^2*a^2*c^3*d^2*f^2 - 40*C^2*a^2*c^3*d^2*f^2 + 8*A*B*a^2*
d^5*f^2 - 8*A*C*a^2*c^5*f^2 - 8*B*C*a^2*d^5*f^2 + 20*A^2*a^2*c*d^4*f^2 - 20
*B^2*a^2*c*d^4*f^2 + 20*C^2*a^2*c*d^4*f^2 + 40*A*B*a^2*c^4*d*f^2 - 40*A*C*a
^2*c*d^4*f^2 - 40*B*C*a^2*c^4*d*f^2 - 80*A*B*a^2*c^2*d^3*f^2 + 80*A*C*a^2*c
^3*d^2*f^2 + 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f
^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*((c + d*tan(e
 + f*x))^(1/2)*(-(((8*A^2*a^2*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f
^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f
^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*
a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^
4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3
*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^
4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c
^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^
2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a
^2*c^5*f^2 - 4*B^2*a^2*c^5*f^2 + 4*C^2*a^2*c^5*f^2 - 40*A^2*a^2*c^3*d^2*f^2
 + 40*B^2*a^2*c^3*d^2*f^2 - 40*C^2*a^2*c^3*d^2*f^2 + 8*A*B*a^2*d^5*f^2 - 8*
A*C*a^2*c^5*f^2 - 8*B*C*a^2*d^5*f^2 + 20*A^2*a^2*c*d^4*f^2 - 20*B^2*a^2*c*d
^4*f^2 + 20*C^2*a^2*c*d^4*f^2 + 40*A*B*a^2*c^4*d*f^2 - 40*A*C*a^2*c*d^4*f^2
 - 40*B*C*a^2*c^4*d*f^2 - 80*A*B*a^2*c^2*d^3*f^2 + 80*A*C*a^2*c^3*d^2*f^2 +
 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*
d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*
d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 161
28*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*
f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 32*A*a*d^21*f^4 + 32*C*a*d^21*f
^4 - 160*A*a*c^2*d^19*f^4 - 128*A*a*c^4*d^17*f^4 + 896*A*a*c^6*d^15*f^4 + 3
136*A*a*c^8*d^13*f^4 + 4928*A*a*c^10*d^11*f^4 + 4480*A*a*c^12*d^9*f^4 + 243
2*A*a*c^14*d^7*f^4 + 736*A*a*c^16*d^5*f^4 + 96*A*a*c^18*d^3*f^4 + 736*B*a*c
^3*d^18*f^4 + 2432*B*a*c^5*d^16*f^4 + 4480*B*a*c^7*d^14*f^4 + 4928*B*a*c^9*
d^12*f^4 + 3136*B*a*c^11*d^10*f^4 + 896*B*a*c^13*d^8*f^4 - 128*B*a*c^15*d^6

*f^4 - 160*B*a*c^17*d^4*f^4 - 32*B*a*c^19*d^2*f^4 + 160*C*a*c^2*d^19*f^4 + 128*C*a*c^4*d^17*f^4 - 896*C*a*c^6*d^15*f^4 - 3136*C*a*c^8*d^13*f^4 - 4928*C*a*c^10*d^11*f^4 - 4480*C*a*c^12*d^9*f^4 - 2432*C*a*c^14*d^7*f^4 - 736*C*a*c^16*d^5*f^4 - 96*C*a*c^18*d^3*f^4 + 96*B*a*c*d^20*f^4) + (c + d*tan(e + f*x))^(1/2)*(16*A^2*a^2*d^18*f^3 - 16*B^2*a^2*d^18*f^3 + 16*C^2*a^2*d^18*f^3 - 320*A^2*a^2*c^4*d^14*f^3 - 1024*A^2*a^2*c^6*d^12*f^3 - 1440*A^2*a^2*c^8*d^10*f^3 - 1024*A^2*a^2*c^10*d^8*f^3 - 320*A^2*a^2*c^12*d^6*f^3 + 16*A^2*a^2*c^16*d^2*f^3 + 320*B^2*a^2*c^4*d^14*f^3 + 1024*B^2*a^2*c^6*d^12*f^3 + 1440*B^2*a^2*c^8*d^10*f^3 + 1024*B^2*a^2*c^10*d^8*f^3 + 320*B^2*a^2*c^12*d^6*f^3 - 16*B^2*a^2*c^16*d^2*f^3 - 320*C^2*a^2*c^4*d^14*f^3 - 1024*C^2*a^2*c^6*d^12*f^3 - 1440*C^2*a^2*c^8*d^10*f^3 - 1024*C^2*a^2*c^10*d^8*f^3 - 320*C^2*a^2*c^12*d^6*f^3 + 16*C^2*a^2*c^16*d^2*f^3 - 32*A*C*a^2*d^18*f^3 - 128*A*B*a^2*c*d^17*f^3 + 128*B*C*a^2*c*d^17*f^3 - 640*A*B*a^2*c^3*d^15*f^3 - 1152*A*B*a^2*c^5*d^13*f^3 - 640*A*B*a^2*c^7*d^11*f^3 + 640*A*B*a^2*c^9*d^9*f^3 + 1152*A*B*a^2*c^11*d^7*f^3 + 640*A*B*a^2*c^13*d^5*f^3 + 128*A*B*a^2*c^15*d^3*f^3 + 640*A*C*a^2*c^4*d^14*f^3 + 2048*A*C*a^2*c^6*d^12*f^3 + 2880*A*C*a^2*c^8*d^10*f^3 + 2048*A*C*a^2*c^10*d^8*f^3 + 640*A*C*a^2*c^12*d^6*f^3 - 32*A*C*a^2*c^16*d^2*f^3 + 640*B*C*a^2*c^3*d^15*f^3 + 1152*B*C*a^2*c^5*d^13*f^3 + 640*B*C*a^2*c^7*d^11*f^3 - 640*B*C*a^2*c^9*d^9*f^3 - 1152*B*C*a^2*c^11*d^7*f^3 - 640*B*C*a^2*c^13*d^5*f^3 - 128*B*C*a^2*c^15*d^3*f^3))*(-(((8*A^2*a^2*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^5*f^2 - 4*B^2*a^2*c^5*f^2 + 4*C^2*a^2*c^5*f^2 - 40*A^2*a^2*c^3*d^2*f^2 + 40*B^2*a^2*c^3*d^2*f^2 - 40*C^2*a^2*c^3*d^2*f^2 + 8*A*B*a^2*d^5*f^2 - 8*A*C*a^2*c^5*f^2 - 8*B*C*a^2*d^5*f^2 + 20*A^2*a^2*c*d^4*f^2 - 20*B^2*a^2*c*d^4*f^2 + 20*C^2*a^2*c*d^4*f^2 + 40*A*B*a^2*c^4*d*f^2 - 40*A*C*a^2*c*d^4*f^2 - 40*B*C*a^2*c^4*d*f^2 - 80*A*B*a^2*c^2*d^3*f^2 + 80*A*C*a^2*c^3*d^2*f^2 + 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2) + ((-(((8*A^2*a^2*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^5*f^2 - 4*B^2*a^2*c^5*f^2 + 4*C^2*a^2*c^5*f^2 - 40*A^2*a^2*c^

3*d^2*f^2 + 40*B^2*a^2*c^3*d^2*f^2 - 40*C^2*a^2*c^3*d^2*f^2 + 8*A*B*a^2*d^5
*f^2 - 8*A*C*a^2*c^5*f^2 - 8*B*C*a^2*d^5*f^2 + 20*A^2*a^2*c*d^4*f^2 - 20*B^
2*a^2*c*d^4*f^2 + 20*C^2*a^2*c*d^4*f^2 + 40*A*B*a^2*c^4*d*f^2 - 40*A*C*a^2*
c*d^4*f^2 - 40*B*C*a^2*c^4*d*f^2 - 80*A*B*a^2*c^2*d^3*f^2 + 80*A*C*a^2*c^3*
d^2*f^2 + 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4
+ 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(32*C*a*d^21*f^4
 - 32*A*a*d^21*f^4 - (c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^2*c^5*f^2 - 8*
B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^2*c
^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^5*f
^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 + 40*
C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C*a^
2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*C*a
^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4
*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 -
 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4
- 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^5*f^2 - 4*B^2*a^2*c^5*f^2 + 4*C^2*a^2
*c^5*f^2 - 40*A^2*a^2*c^3*d^2*f^2 + 40*B^2*a^2*c^3*d^2*f^2 - 40*C^2*a^2*c^3
*d^2*f^2 + 8*A*B*a^2*d^5*f^2 - 8*A*C*a^2*c^5*f^2 - 8*B*C*a^2*d^5*f^2 + 20*A
^2*a^2*c*d^4*f^2 - 20*B^2*a^2*c*d^4*f^2 + 20*C^2*a^2*c*d^4*f^2 + 40*A*B*a^2
*c^4*d*f^2 - 40*A*C*a^2*c*d^4*f^2 - 40*B*C*a^2*c^4*d*f^2 - 80*A*B*a^2*c^2*d
^3*f^2 + 80*A*C*a^2*c^3*d^2*f^2 + 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d
^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))
)^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^
16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7
680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5)
- 160*A*a*c^2*d^19*f^4 - 128*A*a*c^4*d^17*f^4 + 896*A*a*c^6*d^15*f^4 + 3136
*A*a*c^8*d^13*f^4 + 4928*A*a*c^10*d^11*f^4 + 4480*A*a*c^12*d^9*f^4 + 2432*A
*a*c^14*d^7*f^4 + 736*A*a*c^16*d^5*f^4 + 96*A*a*c^18*d^3*f^4 + 736*B*a*c^3*
d^18*f^4 + 2432*B*a*c^5*d^16*f^4 + 4480*B*a*c^7*d^14*f^4 + 4928*B*a*c^9*d^1
2*f^4 + 3136*B*a*c^11*d^10*f^4 + 896*B*a*c^13*d^8*f^4 - 128*B*a*c^15*d^6*f^
4 - 160*B*a*c^17*d^4*f^4 - 32*B*a*c^19*d^2*f^4 + 160*C*a*c^2*d^19*f^4 + 128
*C*a*c^4*d^17*f^4 - 896*C*a*c^6*d^15*f^4 - 3136*C*a*c^8*d^13*f^4 - 4928*C*a
*c^10*d^11*f^4 - 4480*C*a*c^12*d^9*f^4 - 2432*C*a*c^14*d^7*f^4 - 736*C*a*c^
16*d^5*f^4 - 96*C*a*c^18*d^3*f^4 + 96*B*a*c*d^20*f^4) - (c + d*tan(e + f*x)
)^(1/2)*(16*A^2*a^2*d^18*f^3 - 16*B^2*a^2*d^18*f^3 + 16*C^2*a^2*d^18*f^3 -
320*A^2*a^2*c^4*d^14*f^3 - 1024*A^2*a^2*c^6*d^12*f^3 - 1440*A^2*a^2*c^8*d^1
0*f^3 - 1024*A^2*a^2*c^10*d^8*f^3 - 320*A^2*a^2*c^12*d^6*f^3 + 16*A^2*a^2*c
^16*d^2*f^3 + 320*B^2*a^2*c^4*d^14*f^3 + 1024*B^2*a^2*c^6*d^12*f^3 + 1440*B
^2*a^2*c^8*d^10*f^3 + 1024*B^2*a^2*c^10*d^8*f^3 + 320*B^2*a^2*c^12*d^6*f^3
- 16*B^2*a^2*c^16*d^2*f^3 - 320*C^2*a^2*c^4*d^14*f^3 - 1024*C^2*a^2*c^6*d^1
2*f^3 - 1440*C^2*a^2*c^8*d^10*f^3 - 1024*C^2*a^2*c^10*d^8*f^3 - 320*C^2*a^2
*c^12*d^6*f^3 + 16*C^2*a^2*c^16*d^2*f^3 - 32*A*C*a^2*d^18*f^3 - 128*A*B*a^2
*c*d^17*f^3 + 128*B*C*a^2*c*d^17*f^3 - 640*A*B*a^2*c^3*d^15*f^3 - 1152*A*B*
a^2*c^5*d^13*f^3 - 640*A*B*a^2*c^7*d^11*f^3 + 640*A*B*a^2*c^9*d^9*f^3 + 115
2*A*B*a^2*c^11*d^7*f^3 + 640*A*B*a^2*c^13*d^5*f^3 + 128*A*B*a^2*c^15*d^3*f^

3 + 640*A*C*a^2*c^4*d^14*f^3 + 2048*A*C*a^2*c^6*d^12*f^3 + 2880*A*C*a^2*c^8
*d^10*f^3 + 2048*A*C*a^2*c^10*d^8*f^3 + 640*A*C*a^2*c^12*d^6*f^3 - 32*A*C*a
^2*c^16*d^2*f^3 + 640*B*C*a^2*c^3*d^15*f^3 + 1152*B*C*a^2*c^5*d^13*f^3 + 64
0*B*C*a^2*c^7*d^11*f^3 - 640*B*C*a^2*c^9*d^9*f^3 - 1152*B*C*a^2*c^11*d^7*f^
3 - 640*B*C*a^2*c^13*d^5*f^3 - 128*B*C*a^2*c^15*d^3*f^3))*(-(((8*A^2*a^2*c^
5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80
*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C
*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4
*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 -
80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 +
160*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4
+ 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 +
C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^
2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^5*f^2 - 4*B^2*a^2*c^5*f^2 +
4*C^2*a^2*c^5*f^2 - 40*A^2*a^2*c^3*d^2*f^2 + 40*B^2*a^2*c^3*d^2*f^2 - 40*C
^2*a^2*c^3*d^2*f^2 + 8*A*B*a^2*d^5*f^2 - 8*A*C*a^2*c^5*f^2 - 8*B*C*a^2*d^5*
f^2 + 20*A^2*a^2*c*d^4*f^2 - 20*B^2*a^2*c*d^4*f^2 + 20*C^2*a^2*c*d^4*f^2 +
40*A*B*a^2*c^4*d*f^2 - 40*A*C*a^2*c*d^4*f^2 - 40*B*C*a^2*c^4*d*f^2 - 80*A*B
*a^2*c^2*d^3*f^2 + 80*A*C*a^2*c^3*d^2*f^2 + 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^
10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8
*d^2*f^4)))^(1/2) - 16*B^3*a^3*d^16*f^2 - 192*A^3*a^3*c^3*d^13*f^2 - 480*A^
3*a^3*c^5*d^11*f^2 - 640*A^3*a^3*c^7*d^9*f^2 - 480*A^3*a^3*c^9*d^7*f^2 - 19
2*A^3*a^3*c^11*d^5*f^2 - 32*A^3*a^3*c^13*d^3*f^2 - 80*B^3*a^3*c^2*d^14*f^2
- 144*B^3*a^3*c^4*d^12*f^2 - 80*B^3*a^3*c^6*d^10*f^2 + 80*B^3*a^3*c^8*d^8*f
^2 + 144*B^3*a^3*c^10*d^6*f^2 + 80*B^3*a^3*c^12*d^4*f^2 + 16*B^3*a^3*c^14*d
^2*f^2 + 192*C^3*a^3*c^3*d^13*f^2 + 480*C^3*a^3*c^5*d^11*f^2 + 640*C^3*a^3*
c^7*d^9*f^2 + 480*C^3*a^3*c^9*d^7*f^2 + 192*C^3*a^3*c^11*d^5*f^2 + 32*C^3*a
^3*c^13*d^3*f^2 - 16*A^2*B*a^3*d^16*f^2 - 16*B*C^2*a^3*d^16*f^2 - 32*A^3*a^
3*c*d^15*f^2 + 32*C^3*a^3*c*d^15*f^2 - 192*A*B^2*a^3*c^3*d^13*f^2 - 480*A*B
^2*a^3*c^5*d^11*f^2 - 640*A*B^2*a^3*c^7*d^9*f^2 - 480*A*B^2*a^3*c^9*d^7*f^2
- 192*A*B^2*a^3*c^11*d^5*f^2 - 32*A*B^2*a^3*c^13*d^3*f^2 - 80*A^2*B*a^3*c^
2*d^14*f^2 - 144*A^2*B*a^3*c^4*d^12*f^2 - 80*A^2*B*a^3*c^6*d^10*f^2 + 80*A^
2*B*a^3*c^8*d^8*f^2 + 144*A^2*B*a^3*c^10*d^6*f^2 + 80*A^2*B*a^3*c^12*d^4*f^
2 + 16*A^2*B*a^3*c^14*d^2*f^2 - 576*A*C^2*a^3*c^3*d^13*f^2 - 1440*A*C^2*a^3
*c^5*d^11*f^2 - 1920*A*C^2*a^3*c^7*d^9*f^2 - 1440*A*C^2*a^3*c^9*d^7*f^2 - 5
76*A*C^2*a^3*c^11*d^5*f^2 - 96*A*C^2*a^3*c^13*d^3*f^2 + 576*A^2*C*a^3*c^3*d
^13*f^2 + 1440*A^2*C*a^3*c^5*d^11*f^2 + 1920*A^2*C*a^3*c^7*d^9*f^2 + 1440*A
^2*C*a^3*c^9*d^7*f^2 + 576*A^2*C*a^3*c^11*d^5*f^2 + 96*A^2*C*a^3*c^13*d^3*f
^2 - 80*B*C^2*a^3*c^2*d^14*f^2 - 144*B*C^2*a^3*c^4*d^12*f^2 - 80*B*C^2*a^3*
c^6*d^10*f^2 + 80*B*C^2*a^3*c^8*d^8*f^2 + 144*B*C^2*a^3*c^10*d^6*f^2 + 80*B
*C^2*a^3*c^12*d^4*f^2 + 16*B*C^2*a^3*c^14*d^2*f^2 + 192*B^2*C*a^3*c^3*d^13*
f^2 + 480*B^2*C*a^3*c^5*d^11*f^2 + 640*B^2*C*a^3*c^7*d^9*f^2 + 480*B^2*C*a^
3*c^9*d^7*f^2 + 192*B^2*C*a^3*c^11*d^5*f^2 + 32*B^2*C*a^3*c^13*d^3*f^2 + 32
*A*B*C*a^3*d^16*f^2 - 32*A*B^2*a^3*c*d^15*f^2 - 96*A*C^2*a^3*c*d^15*f^2 + 9
6*A^2*C*a^3*c*d^15*f^2 + 32*B^2*C*a^3*c*d^15*f^2 + 160*A*B*C*a^3*c^2*d^14*f

^2 + 288*A*B*C*a^3*c^4*d^12*f^2 + 160*A*B*C*a^3*c^6*d^10*f^2 - 160*A*B*C*a^3*c^8*d^8*f^2 - 288*A*B*C*a^3*c^10*d^6*f^2 - 160*A*B*C*a^3*c^12*d^4*f^2 - 32*A*B*C*a^3*c^14*d^2*f^2))*(-(((8*A^2*a^2*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^5*f^2 - 4*B^2*a^2*c^5*f^2 + 4*C^2*a^2*c^5*f^2 - 40*A^2*a^2*c^3*d^2*f^2 + 40*B^2*a^2*c^3*d^2*f^2 - 40*C^2*a^2*c^3*d^2*f^2 + 8*A*B*a^2*d^5*f^2 - 8*A*C*a^2*c^5*f^2 - 8*B*C*a^2*d^5*f^2 + 20*A^2*a^2*c*d^4*f^2 - 20*B^2*a^2*c*d^4*f^2 + 20*C^2*a^2*c*d^4*f^2 + 40*A*B*a^2*c^4*d*f^2 - 40*A*C*a^2*c*d^4*f^2 - 40*B*C*a^2*c^4*d*f^2 - 80*A*B*a^2*c^2*d^3*f^2 + 80*A*C*a^2*c^3*d^2*f^2 + 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*2i - ((2*(A*a*d^2 + C*a*c^2 - B*a*c*d))/(3*(c^2 + d^2)) - (2*d*(c + d*tan(e + f*x))*(B*a*c^2 - B*a*d^2 - 2*A*a*c*d + 2*C*a*c*d))/(c^2 + d^2)^2)/(d*f*(c + d*tan(e + f*x))^(3/2)) - atan(-(((c + d*tan(e + f*x))^(1/2)*(16*A^2*b^2*d^18*f^3 - 16*B^2*b^2*d^18*f^3 + 16*C^2*b^2*d^18*f^3 - 320*A^2*b^2*c^4*d^14*f^3 - 1024*A^2*b^2*c^6*d^12*f^3 - 1440*A^2*b^2*c^8*d^10*f^3 - 1024*A^2*b^2*c^10*d^8*f^3 - 320*A^2*b^2*c^12*d^6*f^3 + 16*A^2*b^2*c^16*d^2*f^3 + 320*B^2*b^2*c^4*d^14*f^3 + 1024*B^2*b^2*c^6*d^12*f^3 + 1440*B^2*b^2*c^8*d^10*f^3 + 1024*B^2*b^2*c^10*d^8*f^3 + 320*B^2*b^2*c^12*d^6*f^3 - 16*B^2*b^2*c^16*d^2*f^3 - 320*C^2*b^2*c^4*d^14*f^3 - 1024*C^2*b^2*c^6*d^12*f^3 - 1440*C^2*b^2*c^8*d^10*f^3 - 1024*C^2*b^2*c^10*d^8*f^3 - 320*C^2*b^2*c^12*d^6*f^3 + 16*C^2*b^2*c^16*d^2*f^3 - 32*A*C*b^2*d^18*f^3 - 128*A*B*b^2*c*d^17*f^3 + 128*B*C*b^2*c*d^17*f^3 - 640*A*B*b^2*c^3*d^15*f^3 - 1152*A*B*b^2*c^5*d^13*f^3 - 640*A*B*b^2*c^7*d^11*f^3 + 640*A*B*b^2*c^9*d^9*f^3 + 1152*A*B*b^2*c^11*d^7*f^3 + 640*A*B*b^2*c^13*d^5*f^3 + 128*A*B*b^2*c^15*d^3*f^3 + 640*A*C*b^2*c^4*d^14*f^3 + 2048*A*C*b^2*c^6*d^12*f^3 + 2880*A*C*b^2*c^8*d^10*f^3 + 2048*A*C*b^2*c^10*d^8*f^3 + 640*A*C*b^2*c^12*d^6*f^3 - 32*A*C*b^2*c^16*d^2*f^3 + 640*B*C*b^2*c^3*d^15*f^3 + 1152*B*C*b^2*c^5*d^13*f^3 + 640*B*C*b^2*c^7*d^11*f^3 - 640*B*C*b^2*c^9*d^9*f^3 - 1152*B*C*b^2*c^11*d^7*f^3 - 640*B*C*b^2*c^13*d^5*f^3 - 128*B*C*b^2*c^15*d^3*f^3) + (-(((8*A^2*b^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/

2) - 4*A^2*b^2*c^5*f^2 + 4*B^2*b^2*c^5*f^2 - 4*C^2*b^2*c^5*f^2 + 40*A^2*b^2
*c^3*d^2*f^2 - 40*B^2*b^2*c^3*d^2*f^2 + 40*C^2*b^2*c^3*d^2*f^2 - 8*A*B*b^2*
d^5*f^2 + 8*A*C*b^2*c^5*f^2 + 8*B*C*b^2*d^5*f^2 - 20*A^2*b^2*c*d^4*f^2 + 20
*B^2*b^2*c*d^4*f^2 - 20*C^2*b^2*c*d^4*f^2 - 40*A*B*b^2*c^4*d*f^2 + 40*A*C*b
^2*c*d^4*f^2 + 40*B*C*b^2*c^4*d*f^2 + 80*A*B*b^2*c^2*d^3*f^2 - 80*A*C*b^2*c
^3*d^2*f^2 - 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f
^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(128*A*b*c^15
*d^6*f^4 - 32*B*b*d^21*f^4 - 736*A*b*c^3*d^18*f^4 - 2432*A*b*c^5*d^16*f^4 -
4480*A*b*c^7*d^14*f^4 - 4928*A*b*c^9*d^12*f^4 - 3136*A*b*c^11*d^10*f^4 - 8
96*A*b*c^13*d^8*f^4 - (c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*b^2*c^5*f^2 - 8
*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b^2*
c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c^5*
f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f^2 + 40
*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 - 80*B*C*b
^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 + 160*B*C*
b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^
4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4
- 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4
- 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c^5*f^2 + 4*B^2*b^2*c^5*f^2 - 4*C^2*b^
2*c^5*f^2 + 40*A^2*b^2*c^3*d^2*f^2 - 40*B^2*b^2*c^3*d^2*f^2 + 40*C^2*b^2*c^
3*d^2*f^2 - 8*A*B*b^2*d^5*f^2 + 8*A*C*b^2*c^5*f^2 + 8*B*C*b^2*d^5*f^2 - 20*
A^2*b^2*c*d^4*f^2 + 20*B^2*b^2*c*d^4*f^2 - 20*C^2*b^2*c*d^4*f^2 - 40*A*B*b^
2*c^4*d*f^2 + 40*A*C*b^2*c*d^4*f^2 + 40*B*C*b^2*c^4*d*f^2 + 80*A*B*b^2*c^2*
d^3*f^2 - 80*A*C*b^2*c^3*d^2*f^2 - 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 +
d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)
))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d
^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 +
7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5)
+ 160*A*b*c^17*d^4*f^4 + 32*A*b*c^19*d^2*f^4 - 160*B*b*c^2*d^19*f^4 - 128*
B*b*c^4*d^17*f^4 + 896*B*b*c^6*d^15*f^4 + 3136*B*b*c^8*d^13*f^4 + 4928*B*b*
c^10*d^11*f^4 + 4480*B*b*c^12*d^9*f^4 + 2432*B*b*c^14*d^7*f^4 + 736*B*b*c^1
6*d^5*f^4 + 96*B*b*c^18*d^3*f^4 + 736*C*b*c^3*d^18*f^4 + 2432*C*b*c^5*d^16*
f^4 + 4480*C*b*c^7*d^14*f^4 + 4928*C*b*c^9*d^12*f^4 + 3136*C*b*c^11*d^10*f^
4 + 896*C*b*c^13*d^8*f^4 - 128*C*b*c^15*d^6*f^4 - 160*C*b*c^17*d^4*f^4 - 32
*C*b*c^19*d^2*f^4 - 96*A*b*c*d^20*f^4 + 96*C*b*c*d^20*f^4))*(-(((8*A^2*b^2*
c^5*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 +
80*B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A
*C*b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d
^4*f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2
- 80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2
+ 160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f
^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4
+ C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*
B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c^5*f^2 + 4*B^2*b^2*c^5*f^2
- 4*C^2*b^2*c^5*f^2 + 40*A^2*b^2*c^3*d^2*f^2 - 40*B^2*b^2*c^3*d^2*f^2 + 40

*C^2*b^2*c^3*d^2*f^2 - 8*A*B*b^2*d^5*f^2 + 8*A*C*b^2*c^5*f^2 + 8*B*C*b^2*d^
5*f^2 - 20*A^2*b^2*c*d^4*f^2 + 20*B^2*b^2*c*d^4*f^2 - 20*C^2*b^2*c*d^4*f^2
- 40*A*B*b^2*c^4*d*f^2 + 40*A*C*b^2*c*d^4*f^2 + 40*B*C*b^2*c^4*d*f^2 + 80*A
*B*b^2*c^2*d^3*f^2 - 80*A*C*b^2*c^3*d^2*f^2 - 80*B*C*b^2*c^2*d^3*f^2)/(16*(
c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c
^8*d^2*f^4)))^(1/2)*1i + ((c + d*tan(e + f*x))^(1/2)*(16*A^2*b^2*d^18*f^3 -
16*B^2*b^2*d^18*f^3 + 16*C^2*b^2*d^18*f^3 - 320*A^2*b^2*c^4*d^14*f^3 - 102
4*A^2*b^2*c^6*d^12*f^3 - 1440*A^2*b^2*c^8*d^10*f^3 - 1024*A^2*b^2*c^10*d^8*
f^3 - 320*A^2*b^2*c^12*d^6*f^3 + 16*A^2*b^2*c^16*d^2*f^3 + 320*B^2*b^2*c^4*
d^14*f^3 + 1024*B^2*b^2*c^6*d^12*f^3 + 1440*B^2*b^2*c^8*d^10*f^3 + 1024*B^2
*b^2*c^10*d^8*f^3 + 320*B^2*b^2*c^12*d^6*f^3 - 16*B^2*b^2*c^16*d^2*f^3 - 32
0*C^2*b^2*c^4*d^14*f^3 - 1024*C^2*b^2*c^6*d^12*f^3 - 1440*C^2*b^2*c^8*d^10*
f^3 - 1024*C^2*b^2*c^10*d^8*f^3 - 320*C^2*b^2*c^12*d^6*f^3 + 16*C^2*b^2*c^1
6*d^2*f^3 - 32*A*C*b^2*d^18*f^3 - 128*A*B*b^2*c*d^17*f^3 + 128*B*C*b^2*c*d^
17*f^3 - 640*A*B*b^2*c^3*d^15*f^3 - 1152*A*B*b^2*c^5*d^13*f^3 - 640*A*B*b^2
*c^7*d^11*f^3 + 640*A*B*b^2*c^9*d^9*f^3 + 1152*A*B*b^2*c^11*d^7*f^3 + 640*A
*B*b^2*c^13*d^5*f^3 + 128*A*B*b^2*c^15*d^3*f^3 + 640*A*C*b^2*c^4*d^14*f^3 +
2048*A*C*b^2*c^6*d^12*f^3 + 2880*A*C*b^2*c^8*d^10*f^3 + 2048*A*C*b^2*c^10*
d^8*f^3 + 640*A*C*b^2*c^12*d^6*f^3 - 32*A*C*b^2*c^16*d^2*f^3 + 640*B*C*b^2*
c^3*d^15*f^3 + 1152*B*C*b^2*c^5*d^13*f^3 + 640*B*C*b^2*c^7*d^11*f^3 - 640*B
*C*b^2*c^9*d^9*f^3 - 1152*B*C*b^2*c^11*d^7*f^3 - 640*B*C*b^2*c^13*d^5*f^3 -
128*B*C*b^2*c^15*d^3*f^3) - (-(((8*A^2*b^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 + 8
*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C^2
*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^5
*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2 +
80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*A
*B*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4
- (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*
d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^
3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(
1/2) - 4*A^2*b^2*c^5*f^2 + 4*B^2*b^2*c^5*f^2 - 4*C^2*b^2*c^5*f^2 + 40*A^2*b
^2*c^3*d^2*f^2 - 40*B^2*b^2*c^3*d^2*f^2 + 40*C^2*b^2*c^3*d^2*f^2 - 8*A*B*b^
2*d^5*f^2 + 8*A*C*b^2*c^5*f^2 + 8*B*C*b^2*d^5*f^2 - 20*A^2*b^2*c*d^4*f^2 +
20*B^2*b^2*c*d^4*f^2 - 20*C^2*b^2*c*d^4*f^2 - 40*A*B*b^2*c^4*d*f^2 + 40*A*C
*b^2*c*d^4*f^2 + 40*B*C*b^2*c^4*d*f^2 + 80*A*B*b^2*c^2*d^3*f^2 - 80*A*C*b^2
*c^3*d^2*f^2 - 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8
*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*((c + d*tan
(e + f*x))^(1/2)*(-(((8*A^2*b^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5
*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2
*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^
2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*
c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d
^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*
f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80
*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*

A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2
*b^2*c^5*f^2 + 4*B^2*b^2*c^5*f^2 - 4*C^2*b^2*c^5*f^2 + 40*A^2*b^2*c^3*d^2*f
^2 - 40*B^2*b^2*c^3*d^2*f^2 + 40*C^2*b^2*c^3*d^2*f^2 - 8*A*B*b^2*d^5*f^2 +
8*A*C*b^2*c^5*f^2 + 8*B*C*b^2*d^5*f^2 - 20*A^2*b^2*c*d^4*f^2 + 20*B^2*b^2*c
*d^4*f^2 - 20*C^2*b^2*c*d^4*f^2 - 40*A*B*b^2*c^4*d*f^2 + 40*A*C*b^2*c*d^4*f
^2 + 40*B*C*b^2*c^4*d*f^2 + 80*A*B*b^2*c^2*d^3*f^2 - 80*A*C*b^2*c^3*d^2*f^2
 - 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^
4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^
3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 1
6128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^
6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 32*B*b*d^21*f^4 - 736*A*b*c^3
*d^18*f^4 - 2432*A*b*c^5*d^16*f^4 - 4480*A*b*c^7*d^14*f^4 - 4928*A*b*c^9*d^
12*f^4 - 3136*A*b*c^11*d^10*f^4 - 896*A*b*c^13*d^8*f^4 + 128*A*b*c^15*d^6*f
^4 + 160*A*b*c^17*d^4*f^4 + 32*A*b*c^19*d^2*f^4 - 160*B*b*c^2*d^19*f^4 - 12
8*B*b*c^4*d^17*f^4 + 896*B*b*c^6*d^15*f^4 + 3136*B*b*c^8*d^13*f^4 + 4928*B*
b*c^10*d^11*f^4 + 4480*B*b*c^12*d^9*f^4 + 2432*B*b*c^14*d^7*f^4 + 736*B*b*c
^16*d^5*f^4 + 96*B*b*c^18*d^3*f^4 + 736*C*b*c^3*d^18*f^4 + 2432*C*b*c^5*d^1
6*f^4 + 4480*C*b*c^7*d^14*f^4 + 4928*C*b*c^9*d^12*f^4 + 3136*C*b*c^11*d^10*
f^4 + 896*C*b*c^13*d^8*f^4 - 128*C*b*c^15*d^6*f^4 - 160*C*b*c^17*d^4*f^4 -
32*C*b*c^19*d^2*f^4 - 96*A*b*c*d^20*f^4 + 96*C*b*c*d^20*f^4))*(-(((8*A^2*b^
2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2
+ 80*B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16
*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c
*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f
^2 - 80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f
^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8
*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b
^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 +
2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c^5*f^2 + 4*B^2*b^2*c^5*f
^2 - 4*C^2*b^2*c^5*f^2 + 40*A^2*b^2*c^3*d^2*f^2 - 40*B^2*b^2*c^3*d^2*f^2 +
40*C^2*b^2*c^3*d^2*f^2 - 8*A*B*b^2*d^5*f^2 + 8*A*C*b^2*c^5*f^2 + 8*B*C*b^2*
d^5*f^2 - 20*A^2*b^2*c*d^4*f^2 + 20*B^2*b^2*c*d^4*f^2 - 20*C^2*b^2*c*d^4*f^
2 - 40*A*B*b^2*c^4*d*f^2 + 40*A*C*b^2*c*d^4*f^2 + 40*B*C*b^2*c^4*d*f^2 + 80
*A*B*b^2*c^2*d^3*f^2 - 80*A*C*b^2*c^3*d^2*f^2 - 80*B*C*b^2*c^2*d^3*f^2)/(16
*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5
*c^8*d^2*f^4)))^(1/2)*1i)/(((c + d*tan(e + f*x))^(1/2)*(16*A^2*b^2*d^18*f^3
 - 16*B^2*b^2*d^18*f^3 + 16*C^2*b^2*d^18*f^3 - 320*A^2*b^2*c^4*d^14*f^3 - 1
024*A^2*b^2*c^6*d^12*f^3 - 1440*A^2*b^2*c^8*d^10*f^3 - 1024*A^2*b^2*c^10*d^
8*f^3 - 320*A^2*b^2*c^12*d^6*f^3 + 16*A^2*b^2*c^16*d^2*f^3 + 320*B^2*b^2*c^
4*d^14*f^3 + 1024*B^2*b^2*c^6*d^12*f^3 + 1440*B^2*b^2*c^8*d^10*f^3 + 1024*B
^2*b^2*c^10*d^8*f^3 + 320*B^2*b^2*c^12*d^6*f^3 - 16*B^2*b^2*c^16*d^2*f^3 -
320*C^2*b^2*c^4*d^14*f^3 - 1024*C^2*b^2*c^6*d^12*f^3 - 1440*C^2*b^2*c^8*d^1
0*f^3 - 1024*C^2*b^2*c^10*d^8*f^3 - 320*C^2*b^2*c^12*d^6*f^3 + 16*C^2*b^2*c
^16*d^2*f^3 - 32*A*C*b^2*d^18*f^3 - 128*A*B*b^2*c*d^17*f^3 + 128*B*C*b^2*c*
d^17*f^3 - 640*A*B*b^2*c^3*d^15*f^3 - 1152*A*B*b^2*c^5*d^13*f^3 - 640*A*B*b

^2*c^7*d^11*f^3 + 640*A*B*b^2*c^9*d^9*f^3 + 1152*A*B*b^2*c^11*d^7*f^3 + 640
*A*B*b^2*c^13*d^5*f^3 + 128*A*B*b^2*c^15*d^3*f^3 + 640*A*C*b^2*c^4*d^14*f^3
 + 2048*A*C*b^2*c^6*d^12*f^3 + 2880*A*C*b^2*c^8*d^10*f^3 + 2048*A*C*b^2*c^1
0*d^8*f^3 + 640*A*C*b^2*c^12*d^6*f^3 - 32*A*C*b^2*c^16*d^2*f^3 + 640*B*C*b^
2*c^3*d^15*f^3 + 1152*B*C*b^2*c^5*d^13*f^3 + 640*B*C*b^2*c^7*d^11*f^3 - 640
*B*C*b^2*c^9*d^9*f^3 - 1152*B*C*b^2*c^11*d^7*f^3 - 640*B*C*b^2*c^13*d^5*f^3
 - 128*B*C*b^2*c^15*d^3*f^3) - (-(((8*A^2*b^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 +
 8*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C
^2*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d
^5*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2
 + 80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160
*A*B*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2
/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^
6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*
A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))
^(1/2) - 4*A^2*b^2*c^5*f^2 + 4*B^2*b^2*c^5*f^2 - 4*C^2*b^2*c^5*f^2 + 40*A^2
*b^2*c^3*d^2*f^2 - 40*B^2*b^2*c^3*d^2*f^2 + 40*C^2*b^2*c^3*d^2*f^2 - 8*A*B*
b^2*d^5*f^2 + 8*A*C*b^2*c^5*f^2 + 8*B*C*b^2*d^5*f^2 - 20*A^2*b^2*c*d^4*f^2
+ 20*B^2*b^2*c*d^4*f^2 - 20*C^2*b^2*c*d^4*f^2 - 40*A*B*b^2*c^4*d*f^2 + 40*A
*C*b^2*c*d^4*f^2 + 40*B*C*b^2*c^4*d*f^2 + 80*A*B*b^2*c^2*d^3*f^2 - 80*A*C*b
^2*c^3*d^2*f^2 - 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d
^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*((c + d*t
an(e + f*x))^(1/2)*(-(((8*A^2*b^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c
^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d
^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*
A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^
2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2
*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^1
0*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 +
80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 +
2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A
^2*b^2*c^5*f^2 + 4*B^2*b^2*c^5*f^2 - 4*C^2*b^2*c^5*f^2 + 40*A^2*b^2*c^3*d^2
*f^2 - 40*B^2*b^2*c^3*d^2*f^2 + 40*C^2*b^2*c^3*d^2*f^2 - 8*A*B*b^2*d^5*f^2
+ 8*A*C*b^2*c^5*f^2 + 8*B*C*b^2*d^5*f^2 - 20*A^2*b^2*c*d^4*f^2 + 20*B^2*b^2
*c*d^4*f^2 - 20*C^2*b^2*c*d^4*f^2 - 40*A*B*b^2*c^4*d*f^2 + 40*A*C*b^2*c*d^4
*f^2 + 40*B*C*b^2*c^4*d*f^2 + 80*A*B*b^2*c^2*d^3*f^2 - 80*A*C*b^2*c^3*d^2*f
^2 - 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*
c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*
c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 +
 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*
d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 32*B*b*d^21*f^4 - 736*A*b*c
^3*d^18*f^4 - 2432*A*b*c^5*d^16*f^4 - 4480*A*b*c^7*d^14*f^4 - 4928*A*b*c^9*
d^12*f^4 - 3136*A*b*c^11*d^10*f^4 - 896*A*b*c^13*d^8*f^4 + 128*A*b*c^15*d^6
*f^4 + 160*A*b*c^17*d^4*f^4 + 32*A*b*c^19*d^2*f^4 - 160*B*b*c^2*d^19*f^4 -
128*B*b*c^4*d^17*f^4 + 896*B*b*c^6*d^15*f^4 + 3136*B*b*c^8*d^13*f^4 + 4928*

B*b*c^10*d^11*f^4 + 4480*B*b*c^12*d^9*f^4 + 2432*B*b*c^14*d^7*f^4 + 736*B*b
*c^16*d^5*f^4 + 96*B*b*c^18*d^3*f^4 + 736*C*b*c^3*d^18*f^4 + 2432*C*b*c^5*d
^16*f^4 + 4480*C*b*c^7*d^14*f^4 + 4928*C*b*c^9*d^12*f^4 + 3136*C*b*c^11*d^1
0*f^4 + 896*C*b*c^13*d^8*f^4 - 128*C*b*c^15*d^6*f^4 - 160*C*b*c^17*d^4*f^4
- 32*C*b*c^19*d^2*f^4 - 96*A*b*c*d^20*f^4 + 96*C*b*c*d^20*f^4))*(-(((8*A^2*
b^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^
2 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 -
16*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2
*c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4
*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2
*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d
^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4
*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4
+ 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c^5*f^2 + 4*B^2*b^2*c^5
*f^2 - 4*C^2*b^2*c^5*f^2 + 40*A^2*b^2*c^3*d^2*f^2 - 40*B^2*b^2*c^3*d^2*f^2
+ 40*C^2*b^2*c^3*d^2*f^2 - 8*A*B*b^2*d^5*f^2 + 8*A*C*b^2*c^5*f^2 + 8*B*C*b^
2*d^5*f^2 - 20*A^2*b^2*c*d^4*f^2 + 20*B^2*b^2*c*d^4*f^2 - 20*C^2*b^2*c*d^4*
f^2 - 40*A*B*b^2*c^4*d*f^2 + 40*A*C*b^2*c*d^4*f^2 + 40*B*C*b^2*c^4*d*f^2 +
80*A*B*b^2*c^2*d^3*f^2 - 80*A*C*b^2*c^3*d^2*f^2 - 80*B*C*b^2*c^2*d^3*f^2)/(
16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 +
5*c^8*d^2*f^4)))^(1/2) - ((c + d*tan(e + f*x))^(1/2)*(16*A^2*b^2*d^18*f^3
- 16*B^2*b^2*d^18*f^3 + 16*C^2*b^2*d^18*f^3 - 320*A^2*b^2*c^4*d^14*f^3 - 10
24*A^2*b^2*c^6*d^12*f^3 - 1440*A^2*b^2*c^8*d^10*f^3 - 1024*A^2*b^2*c^10*d^8
*f^3 - 320*A^2*b^2*c^12*d^6*f^3 + 16*A^2*b^2*c^16*d^2*f^3 + 320*B^2*b^2*c^4
*d^14*f^3 + 1024*B^2*b^2*c^6*d^12*f^3 + 1440*B^2*b^2*c^8*d^10*f^3 + 1024*B^
2*b^2*c^10*d^8*f^3 + 320*B^2*b^2*c^12*d^6*f^3 - 16*B^2*b^2*c^16*d^2*f^3 - 3
20*C^2*b^2*c^4*d^14*f^3 - 1024*C^2*b^2*c^6*d^12*f^3 - 1440*C^2*b^2*c^8*d^10
*f^3 - 1024*C^2*b^2*c^10*d^8*f^3 - 320*C^2*b^2*c^12*d^6*f^3 + 16*C^2*b^2*c^
16*d^2*f^3 - 32*A*C*b^2*d^18*f^3 - 128*A*B*b^2*c*d^17*f^3 + 128*B*C*b^2*c*d
^17*f^3 - 640*A*B*b^2*c^3*d^15*f^3 - 1152*A*B*b^2*c^5*d^13*f^3 - 640*A*B*b^
2*c^7*d^11*f^3 + 640*A*B*b^2*c^9*d^9*f^3 + 1152*A*B*b^2*c^11*d^7*f^3 + 640*
A*B*b^2*c^13*d^5*f^3 + 128*A*B*b^2*c^15*d^3*f^3 + 640*A*C*b^2*c^4*d^14*f^3
+ 2048*A*C*b^2*c^6*d^12*f^3 + 2880*A*C*b^2*c^8*d^10*f^3 + 2048*A*C*b^2*c^10
*d^8*f^3 + 640*A*C*b^2*c^12*d^6*f^3 - 32*A*C*b^2*c^16*d^2*f^3 + 640*B*C*b^2
*c^3*d^15*f^3 + 1152*B*C*b^2*c^5*d^13*f^3 + 640*B*C*b^2*c^7*d^11*f^3 - 640*
B*C*b^2*c^9*d^9*f^3 - 1152*B*C*b^2*c^11*d^7*f^3 - 640*B*C*b^2*c^13*d^5*f^3
- 128*B*C*b^2*c^15*d^3*f^3) + (-(((8*A^2*b^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 +
8*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C^
2*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^
5*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2
+ 80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*
A*B*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/
4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6
*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A
^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^

(1/2) - 4*A^2*b^2*c^5*f^2 + 4*B^2*b^2*c^5*f^2 - 4*C^2*b^2*c^5*f^2 + 40*A^2*
b^2*c^3*d^2*f^2 - 40*B^2*b^2*c^3*d^2*f^2 + 40*C^2*b^2*c^3*d^2*f^2 - 8*A*B*b
^2*d^5*f^2 + 8*A*C*b^2*c^5*f^2 + 8*B*C*b^2*d^5*f^2 - 20*A^2*b^2*c*d^4*f^2 +
 20*B^2*b^2*c*d^4*f^2 - 20*C^2*b^2*c*d^4*f^2 - 40*A*B*b^2*c^4*d*f^2 + 40*A*
C*b^2*c*d^4*f^2 + 40*B*C*b^2*c^4*d*f^2 + 80*A*B*b^2*c^2*d^3*f^2 - 80*A*C*b^
2*c^3*d^2*f^2 - 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^
8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(128*A*b*c
^15*d^6*f^4 - 32*B*b*d^21*f^4 - 736*A*b*c^3*d^18*f^4 - 2432*A*b*c^5*d^16*f^
4 - 4480*A*b*c^7*d^14*f^4 - 4928*A*b*c^9*d^12*f^4 - 3136*A*b*c^11*d^10*f^4
- 896*A*b*c^13*d^8*f^4 - (c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*b^2*c^5*f^2
- 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b
^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c
^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f^2 +
 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 - 80*B*
C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 + 160*B
*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160
*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b
^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*
b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c^5*f^2 + 4*B^2*b^2*c^5*f^2 - 4*C^2
*b^2*c^5*f^2 + 40*A^2*b^2*c^3*d^2*f^2 - 40*B^2*b^2*c^3*d^2*f^2 + 40*C^2*b^2
*c^3*d^2*f^2 - 8*A*B*b^2*d^5*f^2 + 8*A*C*b^2*c^5*f^2 + 8*B*C*b^2*d^5*f^2 -
20*A^2*b^2*c*d^4*f^2 + 20*B^2*b^2*c*d^4*f^2 - 20*C^2*b^2*c*d^4*f^2 - 40*A*B
*b^2*c^4*d*f^2 + 40*A*C*b^2*c*d^4*f^2 + 40*B*C*b^2*c^4*d*f^2 + 80*A*B*b^2*c
^2*d^3*f^2 - 80*A*C*b^2*c^3*d^2*f^2 - 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4
 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f
^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^
7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5
 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f
^5) + 160*A*b*c^17*d^4*f^4 + 32*A*b*c^19*d^2*f^4 - 160*B*b*c^2*d^19*f^4 - 1
28*B*b*c^4*d^17*f^4 + 896*B*b*c^6*d^15*f^4 + 3136*B*b*c^8*d^13*f^4 + 4928*B
*b*c^10*d^11*f^4 + 4480*B*b*c^12*d^9*f^4 + 2432*B*b*c^14*d^7*f^4 + 736*B*b*
c^16*d^5*f^4 + 96*B*b*c^18*d^3*f^4 + 736*C*b*c^3*d^18*f^4 + 2432*C*b*c^5*d^
16*f^4 + 4480*C*b*c^7*d^14*f^4 + 4928*C*b*c^9*d^12*f^4 + 3136*C*b*c^11*d^10
*f^4 + 896*C*b*c^13*d^8*f^4 - 128*C*b*c^15*d^6*f^4 - 160*C*b*c^17*d^4*f^4 -
 32*C*b*c^19*d^2*f^4 - 96*A*b*c*d^20*f^4 + 96*C*b*c*d^20*f^4))*(-(((8*A^2*b
^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2
 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 1
6*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*
c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*
f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2*
f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^
8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4*
b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 +
 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c^5*f^2 + 4*B^2*b^2*c^5*
f^2 - 4*C^2*b^2*c^5*f^2 + 40*A^2*b^2*c^3*d^2*f^2 - 40*B^2*b^2*c^3*d^2*f^2 +

40*C^2*b^2*c^3*d^2*f^2 - 8*A*B*b^2*d^5*f^2 + 8*A*C*b^2*c^5*f^2 + 8*B*C*b^2
*d^5*f^2 - 20*A^2*b^2*c*d^4*f^2 + 20*B^2*b^2*c*d^4*f^2 - 20*C^2*b^2*c*d^4*f
^2 - 40*A*B*b^2*c^4*d*f^2 + 40*A*C*b^2*c*d^4*f^2 + 40*B*C*b^2*c^4*d*f^2 + 8
0*A*B*b^2*c^2*d^3*f^2 - 80*A*C*b^2*c^3*d^2*f^2 - 80*B*C*b^2*c^2*d^3*f^2)/(1
6*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 +
5*c^8*d^2*f^4)))^(1/2) - 16*A^3*b^3*d^16*f^2 + 16*C^3*b^3*d^16*f^2 - 80*A^3
*b^3*c^2*d^14*f^2 - 144*A^3*b^3*c^4*d^12*f^2 - 80*A^3*b^3*c^6*d^10*f^2 + 80
*A^3*b^3*c^8*d^8*f^2 + 144*A^3*b^3*c^10*d^6*f^2 + 80*A^3*b^3*c^12*d^4*f^2 +
16*A^3*b^3*c^14*d^2*f^2 + 192*B^3*b^3*c^3*d^13*f^2 + 480*B^3*b^3*c^5*d^11*
f^2 + 640*B^3*b^3*c^7*d^9*f^2 + 480*B^3*b^3*c^9*d^7*f^2 + 192*B^3*b^3*c^11*
d^5*f^2 + 32*B^3*b^3*c^13*d^3*f^2 + 80*C^3*b^3*c^2*d^14*f^2 + 144*C^3*b^3*c
^4*d^12*f^2 + 80*C^3*b^3*c^6*d^10*f^2 - 80*C^3*b^3*c^8*d^8*f^2 - 144*C^3*b^
3*c^10*d^6*f^2 - 80*C^3*b^3*c^12*d^4*f^2 - 16*C^3*b^3*c^14*d^2*f^2 - 16*A*B
^2*b^3*d^16*f^2 - 48*A*C^2*b^3*d^16*f^2 + 48*A^2*C*b^3*d^16*f^2 + 16*B^2*C*
b^3*d^16*f^2 + 32*B^3*b^3*c*d^15*f^2 - 80*A*B^2*b^3*c^2*d^14*f^2 - 144*A*B^
2*b^3*c^4*d^12*f^2 - 80*A*B^2*b^3*c^6*d^10*f^2 + 80*A*B^2*b^3*c^8*d^8*f^2 +
144*A*B^2*b^3*c^10*d^6*f^2 + 80*A*B^2*b^3*c^12*d^4*f^2 + 16*A*B^2*b^3*c^14
*d^2*f^2 + 192*A^2*B*b^3*c^3*d^13*f^2 + 480*A^2*B*b^3*c^5*d^11*f^2 + 640*A^
2*B*b^3*c^7*d^9*f^2 + 480*A^2*B*b^3*c^9*d^7*f^2 + 192*A^2*B*b^3*c^11*d^5*f^
2 + 32*A^2*B*b^3*c^13*d^3*f^2 - 240*A*C^2*b^3*c^2*d^14*f^2 - 432*A*C^2*b^3*
c^4*d^12*f^2 - 240*A*C^2*b^3*c^6*d^10*f^2 + 240*A*C^2*b^3*c^8*d^8*f^2 + 432
*A*C^2*b^3*c^10*d^6*f^2 + 240*A*C^2*b^3*c^12*d^4*f^2 + 48*A*C^2*b^3*c^14*d^
2*f^2 + 240*A^2*C*b^3*c^2*d^14*f^2 + 432*A^2*C*b^3*c^4*d^12*f^2 + 240*A^2*C
*b^3*c^6*d^10*f^2 - 240*A^2*C*b^3*c^8*d^8*f^2 - 432*A^2*C*b^3*c^10*d^6*f^2
- 240*A^2*C*b^3*c^12*d^4*f^2 - 48*A^2*C*b^3*c^14*d^2*f^2 + 192*B*C^2*b^3*c^
3*d^13*f^2 + 480*B*C^2*b^3*c^5*d^11*f^2 + 640*B*C^2*b^3*c^7*d^9*f^2 + 480*B
*C^2*b^3*c^9*d^7*f^2 + 192*B*C^2*b^3*c^11*d^5*f^2 + 32*B*C^2*b^3*c^13*d^3*f
^2 + 80*B^2*C*b^3*c^2*d^14*f^2 + 144*B^2*C*b^3*c^4*d^12*f^2 + 80*B^2*C*b^3*
c^6*d^10*f^2 - 80*B^2*C*b^3*c^8*d^8*f^2 - 144*B^2*C*b^3*c^10*d^6*f^2 - 80*B
^2*C*b^3*c^12*d^4*f^2 - 16*B^2*C*b^3*c^14*d^2*f^2 + 32*A^2*B*b^3*c*d^15*f^2
+ 32*B*C^2*b^3*c*d^15*f^2 - 384*A*B*C*b^3*c^3*d^13*f^2 - 960*A*B*C*b^3*c^5
*d^11*f^2 - 1280*A*B*C*b^3*c^7*d^9*f^2 - 960*A*B*C*b^3*c^9*d^7*f^2 - 384*A*
B*C*b^3*c^11*d^5*f^2 - 64*A*B*C*b^3*c^13*d^3*f^2 - 64*A*B*C*b^3*c*d^15*f^2)
)*(-(((8*A^2*b^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80*A^2*b
^2*c^3*d^2*f^2 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16*A*B*b
^2*d^5*f^2 - 16*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d^4*f^2
- 40*B^2*b^2*c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2 - 80*
A*C*b^2*c*d^4*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 + 160*A*
C*b^2*c^3*d^2*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f
^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(
A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6
*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c^5*f^2 +
4*B^2*b^2*c^5*f^2 - 4*C^2*b^2*c^5*f^2 + 40*A^2*b^2*c^3*d^2*f^2 - 40*B^2*b^2
*c^3*d^2*f^2 + 40*C^2*b^2*c^3*d^2*f^2 - 8*A*B*b^2*d^5*f^2 + 8*A*C*b^2*c^5*f
^2 + 8*B*C*b^2*d^5*f^2 - 20*A^2*b^2*c*d^4*f^2 + 20*B^2*b^2*c*d^4*f^2 - 20*C

^2*b^2*c*d^4*f^2 - 40*A*B*b^2*c^4*d*f^2 + 40*A*C*b^2*c*d^4*f^2 + 40*B*C*b^2*c^4*d*f^2 + 80*A*B*b^2*c^2*d^3*f^2 - 80*A*C*b^2*c^3*d^2*f^2 - 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*2i

3.125 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^{5/2}}\,dx$

Optimal result 1826
Rubi [A] (verified) 1826
Mathematica [C] (verified) 1829
Maple [B] (verified) 1829
Fricas [B] (verification not implemented) 1830
Sympy [F] 1830
Maxima [F(-1)] 1830
Giac [F(-1)] 1830
Mupad [B] (verification not implemented) 1831

Optimal result

Integrand size = 35, antiderivative size = 209

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^{5/2}}\,dx = -\frac{(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(c-id)^{5/2}f}-\frac{(B-i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(c+id)^{5/2}f}-\frac{2(c^2C-Bcd+Ad^2)}{3d(c^2+d^2)f(c+d\tan(e+fx))^{3/2}}-\frac{2(2c(A-C)d-B(c^2-d^2))}{(c^2+d^2)^2f\sqrt{c+d\tan(e+fx)}}$$

```
[Out] -(I*A+B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(c-I*d)^(5/2)/f-
(B-I*(A-C))*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))/(c+I*d)^(5/2)/f-2
*(2*c*(A-C)*d-B*(c^2-d^2))/(c^2+d^2)^2/f/(c+d*tan(f*x+e))^(1/2)-2/3*(A*d^2-
B*c*d+C*c^2)/d/(c^2+d^2)/f/(c+d*tan(f*x+e))^(3/2)
```

Rubi [A] (verified)

Time = 0.54 (sec) , antiderivative size = 209, normalized size of antiderivative = 1.00, number of steps used = 9, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.171, Rules used = {3709, 3610, 3620, 3618, 65, 214}

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^{5/2}}\,dx = -\frac{(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f(c-id)^{5/2}}-\frac{(B-i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f(c+id)^{5/2}}-\frac{2(Ad^2-Bcd+c^2C)}{3df(c^2+d^2)(c+d\tan(e+fx))^{3/2}}-\frac{2(2cd(A-C)-B(c^2-d^2))}{f(c^2+d^2)^2\sqrt{c+d\tan(e+fx)}}$$

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/(c + d*Tan[e + f*x])^(5/2),x]

[Out] -(((I*A + B - I*C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]])/((c - I
*d)^(5/2)*f)) - ((B - I*(A - C))*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c +
I*d]])/((c + I*d)^(5/2)*f) - (2*(c^2*C - B*c*d + A*d^2))/(3*d*(c^2 + d^2)*f
*(c + d*Tan[e + f*x])^(3/2)) - (2*(2*c*(A - C)*d - B*(c^2 - d^2)))/((c^2 +
d^2)^2*f*Sqrt[c + d*Tan[e + f*x]])
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3610

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(b*c - a*d)*((a + b*Tan[e + f*x])^(m + 1)/
(f*(m + 1)*(a^2 + b^2))), x] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])
^(m + 1)*Simp[a*c + b*d - (b*c - a*d)*Tan[e + f*x], x], x], x] /; FreeQ[{a,
 b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && LtQ[m, -1]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3709

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 -
```

```
 a*b*B + a^2*C)*((a + b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1)*(a^2 + b^2))), x
] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[b*B + a*(A -
C) - (A*b - a*B - b*C)*Tan[e + f*x], x], x], x] /; FreeQ[{a, b, e, f, A, B,
 C}, x] && NeQ[A*b^2 - a*b*B + a^2*C, 0] && LtQ[m, -1] && NeQ[a^2 + b^2, 0]
```

Rubi steps

$$\text{integral} = -\frac{2(c^2C - Bcd + Ad^2)}{3d(c^2+d^2)f(c+d\tan(e+fx))^{3/2}} + \frac{\int \frac{Ac-cC+Bd+(Bc-(A-C)d)\tan(e+fx)}{(c+d\tan(e+fx))^{3/2}}\,dx}{c^2+d^2}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)}{3d(c^2+d^2)f(c+d\tan(e+fx))^{3/2}} - \frac{2(2c(A-C)d - B(c^2-d^2))}{(c^2+d^2)^2 f\sqrt{c+d\tan(e+fx)}} + \frac{\int \frac{-c^2C+2Bcd+Cd^2+A(c^2-d^2)-(2c(A-C)d-B(c^2-d^2))\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{(c^2+d^2)^2}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)}{3d(c^2+d^2)f(c+d\tan(e+fx))^{3/2}} - \frac{2(2c(A-C)d - B(c^2-d^2))}{(c^2+d^2)^2 f\sqrt{c+d\tan(e+fx)}} + \frac{(A-iB-C)\int \frac{1+i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(c-id)^2} + \frac{(A+iB-C)\int \frac{1-i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(c+id)^2}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)}{3d(c^2+d^2)f(c+d\tan(e+fx))^{3/2}} - \frac{2(2c(A-C)d - B(c^2-d^2))}{(c^2+d^2)^2 f\sqrt{c+d\tan(e+fx)}} + \frac{(iA+B-iC)\text{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c-idx}}\,dx, x, i\tan(e+fx)\right)}{2(c-id)^2 f} - \frac{(i(A+iB-C))\text{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c+idx}}\,dx, x, -i\tan(e+fx)\right)}{2(c+id)^2 f}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)}{3d(c^2+d^2)f(c+d\tan(e+fx))^{3/2}} - \frac{2(2c(A-C)d - B(c^2-d^2))}{(c^2+d^2)^2 f\sqrt{c+d\tan(e+fx)}} - \frac{(A+iB-C)\text{Subst}\left(\int \frac{1}{-1+\frac{ic}{d}-\frac{ix^2}{d}}\,dx, x, \sqrt{c+d\tan(e+fx)}\right)}{(c+id)^2 df} + \frac{(A-iB-C)\text{Subst}\left(\int \frac{1}{-1-\frac{ic}{d}+\frac{ix^2}{d}}\,dx, x, \sqrt{c+d\tan(e+fx)}\right)}{d(ic+d)^2 f}$$

$$= -\frac{(B+i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(c-id)^{5/2} f} - \frac{(B-i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(c+id)^{5/2} f} - \frac{2(c^2C - Bcd + Ad^2)}{3d(c^2+d^2)f(c+d\tan(e+fx))^{3/2}} - \frac{2(2c(A-C)d - B(c^2-d^2))}{(c^2+d^2)^2 f\sqrt{c+d\tan(e+fx)}}$$

Mathematica [C] (verified)

Result contains higher order function than in optimal. Order 5 vs. order 3 in optimal.

Time = 0.98 (sec) , antiderivative size = 223, normalized size of antiderivative = 1.07

$$\int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{(c + d\tan(e + fx))^{5/2}}\,dx =$$
$$-\frac{2C(c^2 + d^2) + (Bc + (-A + C)d)\left(i(c + id)\,\text{Hypergeometric2F1}\left(-\frac{3}{2}, 1, -\frac{1}{2}, \frac{c+d\tan(e+fx)}{c-id}\right) - (ic + d)\,\text{Hyp}\right.}{}$$

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/(c + d*Tan[e + f*x])^(5/2
),x]
```

```
[Out] -1/3*(2*C*(c^2 + d^2) + (B*c + (-A + C)*d)*(I*(c + I*d)*Hypergeometric2F1[-
3/2, 1, -1/2, (c + d*Tan[e + f*x])/(c - I*d)] - (I*c + d)*Hypergeometric2F1
[-3/2, 1, -1/2, (c + d*Tan[e + f*x])/(c + I*d)]) - 3*B*(I*(c + I*d)*Hyperge
ometric2F1[-1/2, 1, 1/2, (c + d*Tan[e + f*x])/(c - I*d)] - (I*c + d)*Hyperg
eometric2F1[-1/2, 1, 1/2, (c + d*Tan[e + f*x])/(c + I*d)])*(c + d*Tan[e + f
*x]))/(d*(c^2 + d^2)*f*(c + d*Tan[e + f*x])^(3/2))
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 6787 vs. 2(184) = 368.

Time = 0.18 (sec) , antiderivative size = 6788, normalized size of antiderivative = 32.48

method	result	size
parts	Expression too large to display	6788
derivativedivides	Expression too large to display	20647
default	Expression too large to display	20647

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(5/2),x,method=_RETURN
VERBOSE)
```

```
[Out] result too large to display
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 13143 vs. 2(177) = 354.

Time = 7.15 (sec) , antiderivative size = 13143, normalized size of antiderivative = 62.89

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^{5/2}}\,dx = \text{Too large to display}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(5/2),x, algorit
hm="fricas")
```

```
[Out] Too large to include
```

Sympy [F]

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^{5/2}}\,dx = \int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^{\frac{5}{2}}}\,dx$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x+e))**(5/2),x)
```

```
[Out] Integral((A + B*tan(e + f*x) + C*tan(e + f*x)**2)/(c + d*tan(e + f*x))**(5/
2), x)
```

Maxima [F(-1)]

Timed out.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^{5/2}}\,dx = \text{Timed out}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(5/2),x, algorit
hm="maxima")
```

```
[Out] Timed out
```

Giac [F(-1)]

Timed out.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^{5/2}}\,dx = \text{Timed out}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(5/2),x, algorit
hm="giac")
```

```
[Out] Timed out
```

Mupad [B] (verification not implemented)

Time = 35.92 (sec) , antiderivative size = 14163, normalized size of antiderivative = 67.77

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^{5/2}}\,dx = \text{Too large to display}$$

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/(c + d*tan(e + f*x))^(5/2),x)
```

```
[Out] (log(96*A^3*c^3*d^13*f^2 - ((((((320*A^4*c^2*d^8*f^4 - 16*A^4*d^10*f^4 - 17
60*A^4*c^4*d^6*f^4 + 1600*A^4*c^6*d^4*f^4 - 400*A^4*c^8*d^2*f^4)^(1/2) - 4*
A^2*c^5*f^2 + 40*A^2*c^3*d^2*f^2 - 20*A^2*c*d^4*f^2)/(c^10*f^4 + d^10*f^4 +
 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))^(1/2)*((
(((320*A^4*c^2*d^8*f^4 - 16*A^4*d^10*f^4 - 1760*A^4*c^4*d^6*f^4 + 1600*A^4*
c^6*d^4*f^4 - 400*A^4*c^8*d^2*f^4)^(1/2) - 4*A^2*c^5*f^2 + 40*A^2*c^3*d^2*f
^2 - 20*A^2*c*d^4*f^2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^
4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(64*c
*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 1344
0*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*
f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5))/4 - 32*A*d^2
1*f^4 - 160*A*c^2*d^19*f^4 - 128*A*c^4*d^17*f^4 + 896*A*c^6*d^15*f^4 + 3136
*A*c^8*d^13*f^4 + 4928*A*c^10*d^11*f^4 + 4480*A*c^12*d^9*f^4 + 2432*A*c^14*
d^7*f^4 + 736*A*c^16*d^5*f^4 + 96*A*c^18*d^3*f^4))/4 - (c + d*tan(e + f*x))
^(1/2)*(320*A^2*c^4*d^14*f^3 - 16*A^2*d^18*f^3 + 1024*A^2*c^6*d^12*f^3 + 14
40*A^2*c^8*d^10*f^3 + 1024*A^2*c^10*d^8*f^3 + 320*A^2*c^12*d^6*f^3 - 16*A^2
*c^16*d^2*f^3))*(((320*A^4*c^2*d^8*f^4 - 16*A^4*d^10*f^4 - 1760*A^4*c^4*d^6
*f^4 + 1600*A^4*c^6*d^4*f^4 - 400*A^4*c^8*d^2*f^4)^(1/2) - 4*A^2*c^5*f^2 +
40*A^2*c^3*d^2*f^2 - 20*A^2*c*d^4*f^2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4
 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))^(1/2))/4 + 240*A^3*c^5
*d^11*f^2 + 320*A^3*c^7*d^9*f^2 + 240*A^3*c^9*d^7*f^2 + 96*A^3*c^11*d^5*f^2
 + 16*A^3*c^13*d^3*f^2 + 16*A^3*c*d^15*f^2)*(((320*A^4*c^2*d^8*f^4 - 16*A^4
*d^10*f^4 - 1760*A^4*c^4*d^6*f^4 + 1600*A^4*c^6*d^4*f^4 - 400*A^4*c^8*d^2*f
^4)^(1/2) - 4*A^2*c^5*f^2 + 40*A^2*c^3*d^2*f^2 - 20*A^2*c*d^4*f^2)/(c^10*f^
4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*
f^4))^(1/2))/4 + (log(96*A^3*c^3*d^13*f^2 - ((((-((320*A^4*c^2*d^8*f^4 - 16
*A^4*d^10*f^4 - 1760*A^4*c^4*d^6*f^4 + 1600*A^4*c^6*d^4*f^4 - 400*A^4*c^8*d
^2*f^4)^(1/2) + 4*A^2*c^5*f^2 - 40*A^2*c^3*d^2*f^2 + 20*A^2*c*d^4*f^2)/(c^1
0*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*
d^2*f^4))^(1/2)*(((-((320*A^4*c^2*d^8*f^4 - 16*A^4*d^10*f^4 - 1760*A^4*c^4*
d^6*f^4 + 1600*A^4*c^6*d^4*f^4 - 400*A^4*c^8*d^2*f^4)^(1/2) + 4*A^2*c^5*f^2
 - 40*A^2*c^3*d^2*f^2 + 20*A^2*c*d^4*f^2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*
f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))^(1/2)*(c + d*tan(e
+ f*x))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*
c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f
^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2
```

*f^5))/4 - 32*A*d^21*f^4 - 160*A*c^2*d^19*f^4 - 128*A*c^4*d^17*f^4 + 896*A*
c^6*d^15*f^4 + 3136*A*c^8*d^13*f^4 + 4928*A*c^10*d^11*f^4 + 4480*A*c^12*d^9
*f^4 + 2432*A*c^14*d^7*f^4 + 736*A*c^16*d^5*f^4 + 96*A*c^18*d^3*f^4))/4 - (
c + d*tan(e + f*x))^(1/2)*(320*A^2*c^4*d^14*f^3 - 16*A^2*d^18*f^3 + 1024*A^
2*c^6*d^12*f^3 + 1440*A^2*c^8*d^10*f^3 + 1024*A^2*c^10*d^8*f^3 + 320*A^2*c^
12*d^6*f^3 - 16*A^2*c^16*d^2*f^3))*(-((320*A^4*c^2*d^8*f^4 - 16*A^4*d^10*f^
4 - 1760*A^4*c^4*d^6*f^4 + 1600*A^4*c^6*d^4*f^4 - 400*A^4*c^8*d^2*f^4)^(1/2
) + 4*A^2*c^5*f^2 - 40*A^2*c^3*d^2*f^2 + 20*A^2*c*d^4*f^2)/(c^10*f^4 + d^10
*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))^(1
/2))/4 + 240*A^3*c^5*d^11*f^2 + 320*A^3*c^7*d^9*f^2 + 240*A^3*c^9*d^7*f^2 +
 96*A^3*c^11*d^5*f^2 + 16*A^3*c^13*d^3*f^2 + 16*A^3*c*d^15*f^2)*(-((320*A^4
*c^2*d^8*f^4 - 16*A^4*d^10*f^4 - 1760*A^4*c^4*d^6*f^4 + 1600*A^4*c^6*d^4*f^
4 - 400*A^4*c^8*d^2*f^4)^(1/2) + 4*A^2*c^5*f^2 - 40*A^2*c^3*d^2*f^2 + 20*A^
2*c*d^4*f^2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6
*d^4*f^4 + 5*c^8*d^2*f^4))^(1/2))/4 - log(96*A^3*c^3*d^13*f^2 - ((((320*A^4
*c^2*d^8*f^4 - 16*A^4*d^10*f^4 - 1760*A^4*c^4*d^6*f^4 + 1600*A^4*c^6*d^4*f^
4 - 400*A^4*c^8*d^2*f^4)^(1/2) - 4*A^2*c^5*f^2 + 40*A^2*c^3*d^2*f^2 - 20*A^
2*c*d^4*f^2)/(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4
+ 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2)*(896*A*c^6*d^15*f^4 - (((320*A^4
*c^2*d^8*f^4 - 16*A^4*d^10*f^4 - 1760*A^4*c^4*d^6*f^4 + 1600*A^4*c^6*d^4*f^
4 - 400*A^4*c^8*d^2*f^4)^(1/2) - 4*A^2*c^5*f^2 + 40*A^2*c^3*d^2*f^2 - 20*A^
2*c*d^4*f^2)/(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4
+ 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(64*c
*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 1344
0*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*
f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 160*A*c^2*d
^19*f^4 - 128*A*c^4*d^17*f^4 - 32*A*d^21*f^4 + 3136*A*c^8*d^13*f^4 + 4928*A
*c^10*d^11*f^4 + 4480*A*c^12*d^9*f^4 + 2432*A*c^14*d^7*f^4 + 736*A*c^16*d^5
*f^4 + 96*A*c^18*d^3*f^4) + (c + d*tan(e + f*x))^(1/2)*(320*A^2*c^4*d^14*f^
3 - 16*A^2*d^18*f^3 + 1024*A^2*c^6*d^12*f^3 + 1440*A^2*c^8*d^10*f^3 + 1024*
A^2*c^10*d^8*f^3 + 320*A^2*c^12*d^6*f^3 - 16*A^2*c^16*d^2*f^3))*(((320*A^4*
c^2*d^8*f^4 - 16*A^4*d^10*f^4 - 1760*A^4*c^4*d^6*f^4 + 1600*A^4*c^6*d^4*f^4
 - 400*A^4*c^8*d^2*f^4)^(1/2) - 4*A^2*c^5*f^2 + 40*A^2*c^3*d^2*f^2 - 20*A^2
*c*d^4*f^2)/(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 +
 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 240*A^3*c^5*d^11*f^2 + 320*A^3*
c^7*d^9*f^2 + 240*A^3*c^9*d^7*f^2 + 96*A^3*c^11*d^5*f^2 + 16*A^3*c^13*d^3*f
^2 + 16*A^3*c*d^15*f^2)*(((320*A^4*c^2*d^8*f^4 - 16*A^4*d^10*f^4 - 1760*A^4
*c^4*d^6*f^4 + 1600*A^4*c^6*d^4*f^4 - 400*A^4*c^8*d^2*f^4)^(1/2) - 4*A^2*c^
5*f^2 + 40*A^2*c^3*d^2*f^2 - 20*A^2*c*d^4*f^2)/(16*c^10*f^4 + 16*d^10*f^4 +
 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2
) - log(96*A^3*c^3*d^13*f^2 - ((-((320*A^4*c^2*d^8*f^4 - 16*A^4*d^10*f^4 -
1760*A^4*c^4*d^6*f^4 + 1600*A^4*c^6*d^4*f^4 - 400*A^4*c^8*d^2*f^4)^(1/2) +
4*A^2*c^5*f^2 - 40*A^2*c^3*d^2*f^2 + 20*A^2*c*d^4*f^2)/(16*c^10*f^4 + 16*d^
10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^
4))^(1/2)*(896*A*c^6*d^15*f^4 - (-((320*A^4*c^2*d^8*f^4 - 16*A^4*d^10*f^4 -

1760*A^4*c^4*d^6*f^4 + 1600*A^4*c^6*d^4*f^4 - 400*A^4*c^8*d^2*f^4)^(1/2) +
4*A^2*c^5*f^2 - 40*A^2*c^3*d^2*f^2 + 20*A^2*c*d^4*f^2)/(16*c^10*f^4 + 16*d
^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f
^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2
880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12
*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^
19*d^4*f^5 + 64*c^21*d^2*f^5) - 160*A*c^2*d^19*f^4 - 128*A*c^4*d^17*f^4 - 3
2*A*d^21*f^4 + 3136*A*c^8*d^13*f^4 + 4928*A*c^10*d^11*f^4 + 4480*A*c^12*d^9
*f^4 + 2432*A*c^14*d^7*f^4 + 736*A*c^16*d^5*f^4 + 96*A*c^18*d^3*f^4) + (c +
d*tan(e + f*x))^(1/2)*(320*A^2*c^4*d^14*f^3 - 16*A^2*d^18*f^3 + 1024*A^2*c
^6*d^12*f^3 + 1440*A^2*c^8*d^10*f^3 + 1024*A^2*c^10*d^8*f^3 + 320*A^2*c^12*
d^6*f^3 - 16*A^2*c^16*d^2*f^3))*(-((320*A^4*c^2*d^8*f^4 - 16*A^4*d^10*f^4 -
1760*A^4*c^4*d^6*f^4 + 1600*A^4*c^6*d^4*f^4 - 400*A^4*c^8*d^2*f^4)^(1/2) +
4*A^2*c^5*f^2 - 40*A^2*c^3*d^2*f^2 + 20*A^2*c*d^4*f^2)/(16*c^10*f^4 + 16*d
^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f
^4))^(1/2) + 240*A^3*c^5*d^11*f^2 + 320*A^3*c^7*d^9*f^2 + 240*A^3*c^9*d^7*f
^2 + 96*A^3*c^11*d^5*f^2 + 16*A^3*c^13*d^3*f^2 + 16*A^3*c*d^15*f^2)*(-((320
*A^4*c^2*d^8*f^4 - 16*A^4*d^10*f^4 - 1760*A^4*c^4*d^6*f^4 + 1600*A^4*c^6*d^
4*f^4 - 400*A^4*c^8*d^2*f^4)^(1/2) + 4*A^2*c^5*f^2 - 40*A^2*c^3*d^2*f^2 + 2
0*A^2*c*d^4*f^2)/(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*
f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + (log(((((320*C^4*c^2*d^8*f
^4 - 16*C^4*d^10*f^4 - 1760*C^4*c^4*d^6*f^4 + 1600*C^4*c^6*d^4*f^4 - 400*C^
4*c^8*d^2*f^4)^(1/2) - 4*C^2*c^5*f^2 + 40*C^2*c^3*d^2*f^2 - 20*C^2*c*d^4*f^
2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 +
5*c^8*d^2*f^4))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(320*C^2*c^4*d^14*f^3 -
16*C^2*d^18*f^3 + 1024*C^2*c^6*d^12*f^3 + 1440*C^2*c^8*d^10*f^3 + 1024*C^2*
c^10*d^8*f^3 + 320*C^2*c^12*d^6*f^3 - 16*C^2*c^16*d^2*f^3) + ((((320*C^4*c^
2*d^8*f^4 - 16*C^4*d^10*f^4 - 1760*C^4*c^4*d^6*f^4 + 1600*C^4*c^6*d^4*f^4 -
400*C^4*c^8*d^2*f^4)^(1/2) - 4*C^2*c^5*f^2 + 40*C^2*c^3*d^2*f^2 - 20*C^2*c
*d^4*f^2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^
4*f^4 + 5*c^8*d^2*f^4))^(1/2)*(896*C*c^6*d^15*f^4 - ((((320*C^4*c^2*d^8*f^4
- 16*C^4*d^10*f^4 - 1760*C^4*c^4*d^6*f^4 + 1600*C^4*c^6*d^4*f^4 - 400*C^4*
c^8*d^2*f^4)^(1/2) - 4*C^2*c^5*f^2 + 40*C^2*c^3*d^2*f^2 - 20*C^2*c*d^4*f^2)
/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5
*c^8*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^
20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128
*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^
5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5))/4 - 160*C*c^2*d^19*f^4 - 128*C*c^4
*d^17*f^4 - 32*C*d^21*f^4 + 3136*C*c^8*d^13*f^4 + 4928*C*c^10*d^11*f^4 + 44
80*C*c^12*d^9*f^4 + 2432*C*c^14*d^7*f^4 + 736*C*c^16*d^5*f^4 + 96*C*c^18*d^
3*f^4))/4))/4 - 96*C^3*c^3*d^13*f^2 - 240*C^3*c^5*d^11*f^2 - 320*C^3*c^7*d^
9*f^2 - 240*C^3*c^9*d^7*f^2 - 96*C^3*c^11*d^5*f^2 - 16*C^3*c^13*d^3*f^2 - 1
6*C^3*c*d^15*f^2)*(((320*C^4*c^2*d^8*f^4 - 16*C^4*d^10*f^4 - 1760*C^4*c^4*d
^6*f^4 + 1600*C^4*c^6*d^4*f^4 - 400*C^4*c^8*d^2*f^4)^(1/2) - 4*C^2*c^5*f^2
+ 40*C^2*c^3*d^2*f^2 - 20*C^2*c*d^4*f^2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f

^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))^(1/2))/4 + (log(((-(
(320*C^4*c^2*d^8*f^4 - 16*C^4*d^10*f^4 - 1760*C^4*c^4*d^6*f^4 + 1600*C^4*c^
6*d^4*f^4 - 400*C^4*c^8*d^2*f^4)^(1/2) + 4*C^2*c^5*f^2 - 40*C^2*c^3*d^2*f^2
 + 20*C^2*c*d^4*f^2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4
+ 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(320*C
^2*c^4*d^14*f^3 - 16*C^2*d^18*f^3 + 1024*C^2*c^6*d^12*f^3 + 1440*C^2*c^8*d^
10*f^3 + 1024*C^2*c^10*d^8*f^3 + 320*C^2*c^12*d^6*f^3 - 16*C^2*c^16*d^2*f^3
) + ((-((320*C^4*c^2*d^8*f^4 - 16*C^4*d^10*f^4 - 1760*C^4*c^4*d^6*f^4 + 160
0*C^4*c^6*d^4*f^4 - 400*C^4*c^8*d^2*f^4)^(1/2) + 4*C^2*c^5*f^2 - 40*C^2*c^3
*d^2*f^2 + 20*C^2*c*d^4*f^2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*
d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))^(1/2)*(896*C*c^6*d^15*f^4 - ((-(
(320*C^4*c^2*d^8*f^4 - 16*C^4*d^10*f^4 - 1760*C^4*c^4*d^6*f^4 + 1600*C^4*c^
6*d^4*f^4 - 400*C^4*c^8*d^2*f^4)^(1/2) + 4*C^2*c^5*f^2 - 40*C^2*c^3*d^2*f^2
 + 20*C^2*c*d^4*f^2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4
+ 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(64*c*d
^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*
c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^
5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5))/4 - 160*C*c^2*
d^19*f^4 - 128*C*c^4*d^17*f^4 - 32*C*d^21*f^4 + 3136*C*c^8*d^13*f^4 + 4928*
C*c^10*d^11*f^4 + 4480*C*c^12*d^9*f^4 + 2432*C*c^14*d^7*f^4 + 736*C*c^16*d^
5*f^4 + 96*C*c^18*d^3*f^4))/4))/4 - 96*C^3*c^3*d^13*f^2 - 240*C^3*c^5*d^11*
f^2 - 320*C^3*c^7*d^9*f^2 - 240*C^3*c^9*d^7*f^2 - 96*C^3*c^11*d^5*f^2 - 16*
C^3*c^13*d^3*f^2 - 16*C^3*c*d^15*f^2)*(-((320*C^4*c^2*d^8*f^4 - 16*C^4*d^10
*f^4 - 1760*C^4*c^4*d^6*f^4 + 1600*C^4*c^6*d^4*f^4 - 400*C^4*c^8*d^2*f^4)^(
1/2) + 4*C^2*c^5*f^2 - 40*C^2*c^3*d^2*f^2 + 20*C^2*c*d^4*f^2)/(c^10*f^4 + d
^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))
^(1/2))/4 - log(- (((320*C^4*c^2*d^8*f^4 - 16*C^4*d^10*f^4 - 1760*C^4*c^4*d
^6*f^4 + 1600*C^4*c^6*d^4*f^4 - 400*C^4*c^8*d^2*f^4)^(1/2) - 4*C^2*c^5*f^2
+ 40*C^2*c^3*d^2*f^2 - 20*C^2*c*d^4*f^2)/(16*c^10*f^4 + 16*d^10*f^4 + 80*c^
2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2)*((c
+ d*tan(e + f*x))^(1/2)*(320*C^2*c^4*d^14*f^3 - 16*C^2*d^18*f^3 + 1024*C^2*
c^6*d^12*f^3 + 1440*C^2*c^8*d^10*f^3 + 1024*C^2*c^10*d^8*f^3 + 320*C^2*c^12
*d^6*f^3 - 16*C^2*c^16*d^2*f^3) - (((320*C^4*c^2*d^8*f^4 - 16*C^4*d^10*f^4
- 1760*C^4*c^4*d^6*f^4 + 1600*C^4*c^6*d^4*f^4 - 400*C^4*c^8*d^2*f^4)^(1/2)
- 4*C^2*c^5*f^2 + 40*C^2*c^3*d^2*f^2 - 20*C^2*c*d^4*f^2)/(16*c^10*f^4 + 16*
d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*
f^4))^(1/2)*((((320*C^4*c^2*d^8*f^4 - 16*C^4*d^10*f^4 - 1760*C^4*c^4*d^6*f^
4 + 1600*C^4*c^6*d^4*f^4 - 400*C^4*c^8*d^2*f^4)^(1/2) - 4*C^2*c^5*f^2 + 40*
C^2*c^3*d^2*f^2 - 20*C^2*c*d^4*f^2)/(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8
*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2)*(c + d*ta
n(e + f*x))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7
680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^
10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21
*d^2*f^5) - 32*C*d^21*f^4 - 160*C*c^2*d^19*f^4 - 128*C*c^4*d^17*f^4 + 896*C
*c^6*d^15*f^4 + 3136*C*c^8*d^13*f^4 + 4928*C*c^10*d^11*f^4 + 4480*C*c^12*d^

9*f^4 + 2432*C*c^14*d^7*f^4 + 736*C*c^16*d^5*f^4 + 96*C*c^18*d^3*f^4)) - 96
*C^3*c^3*d^13*f^2 - 240*C^3*c^5*d^11*f^2 - 320*C^3*c^7*d^9*f^2 - 240*C^3*c^
9*d^7*f^2 - 96*C^3*c^11*d^5*f^2 - 16*C^3*c^13*d^3*f^2 - 16*C^3*c*d^15*f^2)*
(((320*C^4*c^2*d^8*f^4 - 16*C^4*d^10*f^4 - 1760*C^4*c^4*d^6*f^4 + 1600*C^4*
c^6*d^4*f^4 - 400*C^4*c^8*d^2*f^4)^(1/2) - 4*C^2*c^5*f^2 + 40*C^2*c^3*d^2*f
^2 - 20*C^2*c*d^4*f^2)/(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^
4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - log(- (-((320*C^4*c^
2*d^8*f^4 - 16*C^4*d^10*f^4 - 1760*C^4*c^4*d^6*f^4 + 1600*C^4*c^6*d^4*f^4 -
400*C^4*c^8*d^2*f^4)^(1/2) + 4*C^2*c^5*f^2 - 40*C^2*c^3*d^2*f^2 + 20*C^2*c
*d^4*f^2)/(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 1
60*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(320*C^
2*c^4*d^14*f^3 - 16*C^2*d^18*f^3 + 1024*C^2*c^6*d^12*f^3 + 1440*C^2*c^8*d^1
0*f^3 + 1024*C^2*c^10*d^8*f^3 + 320*C^2*c^12*d^6*f^3 - 16*C^2*c^16*d^2*f^3)
- (-((320*C^4*c^2*d^8*f^4 - 16*C^4*d^10*f^4 - 1760*C^4*c^4*d^6*f^4 + 1600*
C^4*c^6*d^4*f^4 - 400*C^4*c^8*d^2*f^4)^(1/2) + 4*C^2*c^5*f^2 - 40*C^2*c^3*d
^2*f^2 + 20*C^2*c*d^4*f^2)/(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 16
0*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2)*((-((320*C^4*c^2*d
^8*f^4 - 16*C^4*d^10*f^4 - 1760*C^4*c^4*d^6*f^4 + 1600*C^4*c^6*d^4*f^4 - 40
0*C^4*c^8*d^2*f^4)^(1/2) + 4*C^2*c^5*f^2 - 40*C^2*c^3*d^2*f^2 + 20*C^2*c*d^
4*f^2)/(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*
c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(64*c*d^22*
f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*
d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 +
2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 32*C*d^21*f^4 - 1
60*C*c^2*d^19*f^4 - 128*C*c^4*d^17*f^4 + 896*C*c^6*d^15*f^4 + 3136*C*c^8*d^
13*f^4 + 4928*C*c^10*d^11*f^4 + 4480*C*c^12*d^9*f^4 + 2432*C*c^14*d^7*f^4 +
736*C*c^16*d^5*f^4 + 96*C*c^18*d^3*f^4)) - 96*C^3*c^3*d^13*f^2 - 240*C^3*c
^5*d^11*f^2 - 320*C^3*c^7*d^9*f^2 - 240*C^3*c^9*d^7*f^2 - 96*C^3*c^11*d^5*f
^2 - 16*C^3*c^13*d^3*f^2 - 16*C^3*c*d^15*f^2)*(-((320*C^4*c^2*d^8*f^4 - 16*
C^4*d^10*f^4 - 1760*C^4*c^4*d^6*f^4 + 1600*C^4*c^6*d^4*f^4 - 400*C^4*c^8*d^
2*f^4)^(1/2) + 4*C^2*c^5*f^2 - 40*C^2*c^3*d^2*f^2 + 20*C^2*c*d^4*f^2)/(16*c
^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4
+ 80*c^8*d^2*f^4))^(1/2) + (log(8*B^3*d^16*f^2 - ((((320*B^4*c^2*d^8*f^4 -
16*B^4*d^10*f^4 - 1760*B^4*c^4*d^6*f^4 + 1600*B^4*c^6*d^4*f^4 - 400*B^4*c^8
*d^2*f^4)^(1/2) + 4*B^2*c^5*f^2 - 40*B^2*c^3*d^2*f^2 + 20*B^2*c*d^4*f^2)/(c
^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^
8*d^2*f^4))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(320*B^2*c^4*d^14*f^3 - 16*B^
2*d^18*f^3 + 1024*B^2*c^6*d^12*f^3 + 1440*B^2*c^8*d^10*f^3 + 1024*B^2*c^10*
d^8*f^3 + 320*B^2*c^12*d^6*f^3 - 16*B^2*c^16*d^2*f^3) + ((((320*B^4*c^2*d^8
*f^4 - 16*B^4*d^10*f^4 - 1760*B^4*c^4*d^6*f^4 + 1600*B^4*c^6*d^4*f^4 - 400*
B^4*c^8*d^2*f^4)^(1/2) + 4*B^2*c^5*f^2 - 40*B^2*c^3*d^2*f^2 + 20*B^2*c*d^4*
f^2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4
+ 5*c^8*d^2*f^4))^(1/2)*(((((320*B^4*c^2*d^8*f^4 - 16*B^4*d^10*f^4 - 1760*
B^4*c^4*d^6*f^4 + 1600*B^4*c^6*d^4*f^4 - 400*B^4*c^8*d^2*f^4)^(1/2) + 4*B^2
*c^5*f^2 - 40*B^2*c^3*d^2*f^2 + 20*B^2*c*d^4*f^2)/(c^10*f^4 + d^10*f^4 + 5*

c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5))/4 + 96*B*c*d^20*f^4 + 736*B*c^3*d^18*f^4 + 2432*B*c^5*d^16*f^4 + 4480*B*c^7*d^14*f^4 + 4928*B*c^9*d^12*f^4 + 3136*B*c^11*d^10*f^4 + 896*B*c^13*d^8*f^4 - 128*B*c^15*d^6*f^4 - 160*B*c^17*d^4*f^4 - 32*B*c^19*d^2*f^4))/4))/4 + 40*B^3*c^2*d^14*f^2 + 72*B^3*c^4*d^12*f^2 + 40*B^3*c^6*d^10*f^2 - 40*B^3*c^8*d^8*f^2 - 72*B^3*c^10*d^6*f^2 - 40*B^3*c^12*d^4*f^2 - 8*B^3*c^14*d^2*f^2)*(((320*B^4*c^2*d^8*f^4 - 16*B^4*d^10*f^4 - 1760*B^4*c^4*d^6*f^4 + 1600*B^4*c^6*d^4*f^4 - 400*B^4*c^8*d^2*f^4)^(1/2) + 4*B^2*c^5*f^2 - 40*B^2*c^3*d^2*f^2 + 20*B^2*c*d^4*f^2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))^(1/2))/4 + (log(8*B^3*d^16*f^2 - ((-((320*B^4*c^2*d^8*f^4 - 16*B^4*d^10*f^4 - 1760*B^4*c^4*d^6*f^4 + 1600*B^4*c^6*d^4*f^4 - 400*B^4*c^8*d^2*f^4)^(1/2) - 4*B^2*c^5*f^2 + 40*B^2*c^3*d^2*f^2 - 20*B^2*c*d^4*f^2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(320*B^2*c^4*d^14*f^3 - 16*B^2*d^18*f^3 + 1024*B^2*c^6*d^12*f^3 + 1440*B^2*c^8*d^10*f^3 + 1024*B^2*c^10*d^8*f^3 + 320*B^2*c^12*d^6*f^3 - 16*B^2*c^16*d^2*f^3) + ((-((320*B^4*c^2*d^8*f^4 - 16*B^4*d^10*f^4 - 1760*B^4*c^4*d^6*f^4 + 1600*B^4*c^6*d^4*f^4 - 400*B^4*c^8*d^2*f^4)^(1/2) - 4*B^2*c^5*f^2 + 40*B^2*c^3*d^2*f^2 - 20*B^2*c*d^4*f^2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))^(1/2)*(((-((320*B^4*c^2*d^8*f^4 - 16*B^4*d^10*f^4 - 1760*B^4*c^4*d^6*f^4 + 1600*B^4*c^6*d^4*f^4 - 400*B^4*c^8*d^2*f^4)^(1/2) - 4*B^2*c^5*f^2 + 40*B^2*c^3*d^2*f^2 - 20*B^2*c*d^4*f^2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5))/4 + 96*B*c*d^20*f^4 + 736*B*c^3*d^18*f^4 + 2432*B*c^5*d^16*f^4 + 4480*B*c^7*d^14*f^4 + 4928*B*c^9*d^12*f^4 + 3136*B*c^11*d^10*f^4 + 896*B*c^13*d^8*f^4 - 128*B*c^15*d^6*f^4 - 160*B*c^17*d^4*f^4 - 32*B*c^19*d^2*f^4))/4))/4 + 40*B^3*c^2*d^14*f^2 + 72*B^3*c^4*d^12*f^2 + 40*B^3*c^6*d^10*f^2 - 40*B^3*c^8*d^8*f^2 - 72*B^3*c^10*d^6*f^2 - 40*B^3*c^12*d^4*f^2 - 8*B^3*c^14*d^2*f^2)*(-((320*B^4*c^2*d^8*f^4 - 16*B^4*d^10*f^4 - 1760*B^4*c^4*d^6*f^4 + 1600*B^4*c^6*d^4*f^4 - 400*B^4*c^8*d^2*f^4)^(1/2) - 4*B^2*c^5*f^2 + 40*B^2*c^3*d^2*f^2 - 20*B^2*c*d^4*f^2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))^(1/2))/4 - log((((320*B^4*c^2*d^8*f^4 - 16*B^4*d^10*f^4 - 1760*B^4*c^4*d^6*f^4 + 1600*B^4*c^6*d^4*f^4 - 400*B^4*c^8*d^2*f^4)^(1/2) + 4*B^2*c^5*f^2 - 40*B^2*c^3*d^2*f^2 + 20*B^2*c*d^4*f^2)/(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(320*B^2*c^4*d^14*f^3 - 16*B^2*d^18*f^3 + 1024*B^2*c^6*d^12*f^3 + 1440*B^2*c^8*d^10*f^3 + 1024*B^2*c^10*d^8*f^3 + 320*B^2*c^12*d^6*f^3 - 16*B^2*c^16*d^2*f^3) - (((320*B^4*c^2*d^8*f^4 - 16*B

^4*d^10*f^4 - 1760*B^4*c^4*d^6*f^4 + 1600*B^4*c^6*d^4*f^4 - 400*B^4*c^8*d^2
*f^4)^(1/2) + 4*B^2*c^5*f^2 - 40*B^2*c^3*d^2*f^2 + 20*B^2*c*d^4*f^2)/(16*c^
10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 +
 80*c^8*d^2*f^4))^(1/2)*(96*B*c*d^20*f^4 - (((320*B^4*c^2*d^8*f^4 - 16*B^4*
d^10*f^4 - 1760*B^4*c^4*d^6*f^4 + 1600*B^4*c^6*d^4*f^4 - 400*B^4*c^8*d^2*f^
4)^(1/2) + 4*B^2*c^5*f^2 - 40*B^2*c^3*d^2*f^2 + 20*B^2*c*d^4*f^2)/(16*c^10*
f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80
*c^8*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^
20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128
*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^
5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) + 736*B*c^3*d^18*f^4 + 2432*B*c^5*d
^16*f^4 + 4480*B*c^7*d^14*f^4 + 4928*B*c^9*d^12*f^4 + 3136*B*c^11*d^10*f^4
+ 896*B*c^13*d^8*f^4 - 128*B*c^15*d^6*f^4 - 160*B*c^17*d^4*f^4 - 32*B*c^19*
d^2*f^4)) + 8*B^3*d^16*f^2 + 40*B^3*c^2*d^14*f^2 + 72*B^3*c^4*d^12*f^2 + 40
*B^3*c^6*d^10*f^2 - 40*B^3*c^8*d^8*f^2 - 72*B^3*c^10*d^6*f^2 - 40*B^3*c^12*
d^4*f^2 - 8*B^3*c^14*d^2*f^2)*(((320*B^4*c^2*d^8*f^4 - 16*B^4*d^10*f^4 - 17
60*B^4*c^4*d^6*f^4 + 1600*B^4*c^6*d^4*f^4 - 400*B^4*c^8*d^2*f^4)^(1/2) + 4*
B^2*c^5*f^2 - 40*B^2*c^3*d^2*f^2 + 20*B^2*c*d^4*f^2)/(16*c^10*f^4 + 16*d^10
*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)
)^(1/2) - log((-((320*B^4*c^2*d^8*f^4 - 16*B^4*d^10*f^4 - 1760*B^4*c^4*d^6*
f^4 + 1600*B^4*c^6*d^4*f^4 - 400*B^4*c^8*d^2*f^4)^(1/2) - 4*B^2*c^5*f^2 + 4
0*B^2*c^3*d^2*f^2 - 20*B^2*c*d^4*f^2)/(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d
^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2)*((c + d
*tan(e + f*x))^(1/2)*(320*B^2*c^4*d^14*f^3 - 16*B^2*d^18*f^3 + 1024*B^2*c^6
*d^12*f^3 + 1440*B^2*c^8*d^10*f^3 + 1024*B^2*c^10*d^8*f^3 + 320*B^2*c^12*d^
6*f^3 - 16*B^2*c^16*d^2*f^3) - (-((320*B^4*c^2*d^8*f^4 - 16*B^4*d^10*f^4 -
1760*B^4*c^4*d^6*f^4 + 1600*B^4*c^6*d^4*f^4 - 400*B^4*c^8*d^2*f^4)^(1/2) -
4*B^2*c^5*f^2 + 40*B^2*c^3*d^2*f^2 - 20*B^2*c*d^4*f^2)/(16*c^10*f^4 + 16*d^
10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^
4))^(1/2)*(96*B*c*d^20*f^4 - (-((320*B^4*c^2*d^8*f^4 - 16*B^4*d^10*f^4 - 17
60*B^4*c^4*d^6*f^4 + 1600*B^4*c^6*d^4*f^4 - 400*B^4*c^8*d^2*f^4)^(1/2) - 4*
B^2*c^5*f^2 + 40*B^2*c^3*d^2*f^2 - 20*B^2*c*d^4*f^2)/(16*c^10*f^4 + 16*d^10
*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)
)^(1/2)*(c + d*tan(e + f*x))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880
*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^
5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*
d^4*f^5 + 64*c^21*d^2*f^5) + 736*B*c^3*d^18*f^4 + 2432*B*c^5*d^16*f^4 + 448
0*B*c^7*d^14*f^4 + 4928*B*c^9*d^12*f^4 + 3136*B*c^11*d^10*f^4 + 896*B*c^13*
d^8*f^4 - 128*B*c^15*d^6*f^4 - 160*B*c^17*d^4*f^4 - 32*B*c^19*d^2*f^4)) + 8
*B^3*d^16*f^2 + 40*B^3*c^2*d^14*f^2 + 72*B^3*c^4*d^12*f^2 + 40*B^3*c^6*d^10
*f^2 - 40*B^3*c^8*d^8*f^2 - 72*B^3*c^10*d^6*f^2 - 40*B^3*c^12*d^4*f^2 - 8*B
^3*c^14*d^2*f^2)*(-((320*B^4*c^2*d^8*f^4 - 16*B^4*d^10*f^4 - 1760*B^4*c^4*d
^6*f^4 + 1600*B^4*c^6*d^4*f^4 - 400*B^4*c^8*d^2*f^4)^(1/2) - 4*B^2*c^5*f^2
+ 40*B^2*c^3*d^2*f^2 - 20*B^2*c*d^4*f^2)/(16*c^10*f^4 + 16*d^10*f^4 + 80*c^
2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + ((

```
2*B*c)/(3*(c^2 + d^2)) + (2*B*(c^2 - d^2)*(c + d*tan(e + f*x)))/(c^2 + d^2)
^2)/(f*(c + d*tan(e + f*x))^(3/2)) - ((2*A*d)/(3*(c^2 + d^2)) + (4*A*c*d*(c
 + d*tan(e + f*x)))/(c^2 + d^2)^2)/(f*(c + d*tan(e + f*x))^(3/2)) - ((2*C*c
^2)/(3*(c^2 + d^2)) - (4*C*c*d^2*(c + d*tan(e + f*x)))/(c^2 + d^2)^2)/(d*f*
(c + d*tan(e + f*x))^(3/2))
```

3.126 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^{5/2}}\,dx$

Optimal result 1839
Rubi [A] (verified) 1840
Mathematica [B] (verified) 1844
Maple [B] (verified) 1845
Fricas [F(-1)] 1845
Sympy [F] 1846
Maxima [F(-2)] 1846
Giac [F] 1846
Mupad [F(-1)] 1847

Optimal result

Integrand size = 47, antiderivative size = 365

$$\begin{aligned}\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^{5/2}}\,dx &= \frac{(A-iB-C)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(ia+b)(c-id)^{5/2}f}\\ &+ \frac{(iA-B-iC)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(a+ib)(c+id)^{5/2}f}\\ &- \frac{2b^{3/2}(Ab^2-a(bB-aC))\,\text{arctanh}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{(a^2+b^2)\,(bc-ad)^{5/2}f}\\ &+ \frac{2(c^2C-Bcd+Ad^2)}{3(bc-ad)\,(c^2+d^2)\,f(c+d\tan(e+fx))^{3/2}}\\ &+ \frac{2(b(c^4C-2Bc^3d+c^2(3A-C)d^2+Ad^4)-ad^2(2c(A-C)d-B(c^2-d^2)))}{(bc-ad)^2\,(c^2+d^2)^2\,f\sqrt{c+d\tan(e+fx)}}\end{aligned}$$

```
[Out] (A-I*B-C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(I*a+b)/(c-I*d)^(5/
2)/f+(I*A-B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))/(a+I*b)/(c+I
*d)^(5/2)/f-2*b^(3/2)*(A*b^2-a*(B*b-C*a))*arctanh(b^(1/2)*(c+d*tan(f*x+e))^
(1/2)/(-a*d+b*c)^(1/2))/(a^2+b^2)/(-a*d+b*c)^(5/2)/f+2*(b*(c^4*C-2*B*c^3*d+
c^2*(3*A-C)*d^2+A*d^4)-a*d^2*(2*c*(A-C)*d-B*(c^2-d^2)))/(-a*d+b*c)^2/(c^2+d
^2)^2/f/(c+d*tan(f*x+e))^(1/2)+2/3*(A*d^2-B*c*d+C*c^2)/(-a*d+b*c)/(c^2+d^2)
/f/(c+d*tan(f*x+e))^(3/2)
```

Rubi [A] (verified)

Time = 2.75 (sec) , antiderivative size = 365, normalized size of antiderivative = 1.00, number of steps used = 13, number of rules used = 7, $\frac{\text{number of rules}}{\text{integrand size}} = 0.149$, Rules used = {3730, 3734, 3620, 3618, 65, 214, 3715}

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^{5/2}}\,dx = -\frac{2b^{3/2}(Ab^2-a(bB-aC))\,\text{arctanh}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{f\,(a^2+b^2)\,(bc-ad)^{5/2}} + \frac{(A-iB-C)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f(b+ia)(c-id)^{5/2}} + \frac{(iA-B-iC)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f(a+ib)(c+id)^{5/2}} + \frac{2(Ad^2-Bcd+c^2C)}{3f\,(c^2+d^2)\,(bc-ad)(c+d\tan(e+fx))^{3/2}} + \frac{2(b(c^2d^2(3A-C)+Ad^4-2Bc^3d+c^4C)-ad^2(2cd(A-C)-B(c^2-d^2)))}{f\,(c^2+d^2)^2\,(bc-ad)^2\sqrt{c+d\tan(e+fx)}}$$

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])*(c + d*Ta
n[e + f*x])^(5/2)),x]
```

```
[Out] ((A - I*B - C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]])/((I*a + b)*
(c - I*d)^(5/2)*f) + ((I*A - B - I*C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt
[c + I*d]])/((a + I*b)*(c + I*d)^(5/2)*f) - (2*b^(3/2)*(A*b^2 - a*(b*B - a*
C))*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*d]])/((a^2 + b^
2)*(b*c - a*d)^(5/2)*f) + (2*(c^2*C - B*c*d + A*d^2))/(3*(b*c - a*d)*(c^2 +
 d^2)*f*(c + d*Tan[e + f*x])^(3/2)) + (2*(b*(c^4*C - 2*B*c^3*d + c^2*(3*A -
 C)*d^2 + A*d^4) - a*d^2*(2*c*(A - C)*d - B*(c^2 - d^2))))/((b*c - a*d)^2*(
c^2 + d^2)^2*f*Sqrt[c + d*Tan[e + f*x]])
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3715

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_) + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :>
 Dist[A/f, Subst[Int[(a + b*x)^m*(c + d*x)^n, x], x, Tan[e + f*x]], x] /; F
reeQ[{a, b, c, d, e, f, A, C, m, n}, x] && EqQ[A, C]
```

Rule 3730

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 - a*(b*B - a*C))*(a + b*Tan[e +
 f*x])^(m + 1)*((c + d*Tan[e + f*x])^(n + 1)/(f*(m + 1)*(b*c - a*d)*(a^2 +
b^2))), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3734

```
Int[(((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2))/((a_.) + (b_.)*tan[(e_.)
+ (f_.)*(x_)]), x_Symbol] :> Dist[1/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n
*Simp[b*B + a*(A - C) + (a*B - b*(A - C))*Tan[e + f*x], x], x], x] + Dist[(
A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n*((1 + Tan[e
+ f*x]^2)/(a + b*Tan[e + f*x])), x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C
, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&
 !GtQ[n, 0] &&  !LeQ[n, -1]
```

Rubi steps

$$\text{integral} = \frac{2(c^2C - Bcd + Ad^2)}{3(bc-ad)\,(c^2+d^2)\,f(c+d\tan(e+fx))^{3/2}} + \frac{2\int \frac{-\frac{3}{2}(aAcd - ad(cC-Bd) - Ab(c^2+d^2)) + \frac{3}{2}(bc-ad)(Bc-(A-C)d)\tan(e+fx) + \frac{3}{2}b(c^2C-Bcd+Ad^2)\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}}\,dx}{3(bc-ad)\,(c^2+d^2)}$$

$$= \frac{2(c^2C - Bcd + Ad^2)}{3(bc-ad)\,(c^2+d^2)\,f(c+d\tan(e+fx))^{3/2}} + \frac{2(b(c^4C - 2Bc^3d + c^2(3A-C)d^2 + Ad^4) - ad^2(2c(A-C)d - B(c^2-d^2)))}{(bc-ad)^2\,(c^2+d^2)^2\,f\sqrt{c+d\tan(e+fx)}} + \frac{4\int \frac{-\frac{3}{4}(A(2abc^3d - a^2d^2(c^2-d^2) - b^2(c^2+d^2)^2) + ad(ad(c^2C-2Bcd-Cd^2) - b(2c^3C-3Bc^2d-Bd^3))) - \frac{3}{4}(bc-ad)^2(2c(A-C)d-B \cdots}{(a+b\tan(e+fx))\sqrt{c+d\tan(\cdots}}}{3(bc-ad)^2\,(c^2+d \cdots}$$

$$= \frac{2(c^2C - Bcd + Ad^2)}{3(bc-ad)\,(c^2+d^2)\,f(c+d\tan(e+fx))^{3/2}} + \frac{2(b(c^4C - 2Bc^3d + c^2(3A-C)d^2 + Ad^4) - ad^2(2c(A-C)d - B(c^2-d^2)))}{(bc-ad)^2\,(c^2+d^2)^2\,f\sqrt{c+d\tan(e+fx)}} + \frac{(b^2(Ab^2 - a(bB-aC)))\int \frac{1+\tan^2(e+fx)}{(a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}}\,dx}{(a^2+b^2)\,(bc-ad)^2} + \frac{4\int \frac{-\frac{3}{4}(bc-ad)^2(a(c^2C-2Bcd-Cd^2-A(c^2-d^2)) + b(2c(A-C)d - B(c^2-d^2))) - \frac{3}{4}(bc-ad)^2(2aAcd - 2acCd + Ab(c^2-d^2) - aB(c^2- \cdots}{\sqrt{c+d\tan(e+fx)}}}{3\,(a^2+b^2)\,(bc-ad)^2\,(c^2+d^2)^2}$$

$$= \frac{2(c^2C - Bcd + Ad^2)}{3(bc-ad)\,(c^2+d^2)\,f(c+d\tan(e+fx))^{3/2}} + \frac{2(b(c^4C - 2Bc^3d + c^2(3A-C)d^2 + Ad^4) - ad^2(2c(A-C)d - B(c^2-d^2)))}{(bc-ad)^2\,(c^2+d^2)^2\,f\sqrt{c+d\tan(e+fx)}} + \frac{(A-iB-C)\int \frac{1+i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(a-ib)(c-id)^2} + \frac{(A+iB-C)\int \frac{1-i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(a+ib)(c+id)^2} + \frac{(b^2(Ab^2 - a(bB-aC)))\,\text{Subst}\left(\int \frac{1}{(a+bx)\sqrt{c+dx}}\,dx, x, \tan(e+fx)\right)}{(a^2+b^2)\,(bc-ad)^2 f}$$

$$
\begin{aligned}
&= \frac{2(c^2C - Bcd + Ad^2)}{3(bc - ad)\,(c^2 + d^2)\,f(c + d\tan(e + fx))^{3/2}} \\
&\quad + \frac{2(b(c^4C - 2Bc^3d + c^2(3A - C)d^2 + Ad^4) - ad^2(2c(A - C)d - B(c^2 - d^2)))}{(bc - ad)^2\,(c^2 + d^2)^2\,f\sqrt{c + d\tan(e + fx)}} \\
&\quad + \frac{(iA + B - iC)\text{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c-idx}}\,dx, x, i\tan(e + fx)\right)}{2(a - ib)(c - id)^2f} \\
&\quad - \frac{(i(A + iB - C))\text{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c+idx}}\,dx, x, -i\tan(e + fx)\right)}{2(a + ib)(c + id)^2f} \\
&\quad + \frac{(2b^2(Ab^2 - a(bB - aC)))\,\text{Subst}\left(\int \frac{1}{a-\frac{bc}{d}+\frac{bx^2}{d}}\,dx, x, \sqrt{c + d\tan(e + fx)}\right)}{(a^2 + b^2)\,d(bc - ad)^2f} \\
&= -\frac{2b^{3/2}(Ab^2 - a(bB - aC))\,\text{arctanh}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{(a^2 + b^2)\,(bc - ad)^{5/2}f} \\
&\quad + \frac{2(c^2C - Bcd + Ad^2)}{3(bc - ad)\,(c^2 + d^2)\,f(c + d\tan(e + fx))^{3/2}} \\
&\quad + \frac{2(b(c^4C - 2Bc^3d + c^2(3A - C)d^2 + Ad^4) - ad^2(2c(A - C)d - B(c^2 - d^2)))}{(bc - ad)^2\,(c^2 + d^2)^2\,f\sqrt{c + d\tan(e + fx)}} \\
&\quad - \frac{(A + iB - C)\text{Subst}\left(\int \frac{1}{-1+\frac{ic}{d}-\frac{ix^2}{d}}\,dx, x, \sqrt{c + d\tan(e + fx)}\right)}{(a + ib)(c + id)^2df} \\
&\quad + \frac{(A - iB - C)\text{Subst}\left(\int \frac{1}{-1-\frac{ic}{d}+\frac{ix^2}{d}}\,dx, x, \sqrt{c + d\tan(e + fx)}\right)}{(a - ib)d(ic + d)^2f} \\
&= \frac{(A - iB - C)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(ia + b)(c - id)^{5/2}f} - \frac{(A + iB - C)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(ia - b)(c + id)^{5/2}f} \\
&\quad - \frac{2b^{3/2}(Ab^2 - a(bB - aC))\,\text{arctanh}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{(a^2 + b^2)\,(bc - ad)^{5/2}f} \\
&\quad + \frac{2(c^2C - Bcd + Ad^2)}{3(bc - ad)\,(c^2 + d^2)\,f(c + d\tan(e + fx))^{3/2}} \\
&\quad + \frac{2(b(c^4C - 2Bc^3d + c^2(3A - C)d^2 + Ad^4) - ad^2(2c(A - C)d - B(c^2 - d^2)))}{(bc - ad)^2\,(c^2 + d^2)^2\,f\sqrt{c + d\tan(e + fx)}}
\end{aligned}
$$

Mathematica [B] (verified)

Both result and optimal contain complex but leaf count is larger than twice the leaf count of optimal. 1948 vs. $2(365) = 730$.

Time = 6.39 (sec) , antiderivative size = 1948, normalized size of antiderivative = 5.34

$$\int \frac{A + B\tan(e+fx) + C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^{5/2}}\,dx = -\frac{2(Ad^2 - c(-cC+Bd))}{3(-bc+ad)\,(c^2+d^2)\,f(c+d\tan(e+fx))^{3/2}}$$
$$-\frac{2\left(-\frac{2\left(\frac{i\sqrt{c-id}\left(\frac{1}{2}b(-bc+ad)\left(-\frac{3}{2}c(bc-ad)(Bc-(A-C)d)-\frac{3}{2}bd\left(c^2C-Bcd+Ad^2\right)-\frac{3}{2}d\left(aAcd-ad(cC-Bd)-Ab\left(c^2+d^2\right)\right)\right)+a\left(-\frac{3}{2}\left(\frac{bd^2}{2}-\frac{1}{2}c(-bc+ad)\right)\left(aAc\cdots\right.\right.\right.}{\cdots}}{\cdots}\right.}{\cdots}$$

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])*(c
 + d*Tan[e + f*x])^(5/2)),x]
```

```
[Out] (-2*(A*d^2 - c*(-(c*C) + B*d)))/(3*(-(b*c) + a*d)*(c^2 + d^2)*f*(c + d*Tan[
e + f*x])^(3/2)) - (2*((-2*(((I*Sqrt[c - I*d]*((b*(-(b*c) + a*d)*((-3*c*(b*
c - a*d)*(B*c - (A - C)*d))/2 - (3*b*d*(c^2*C - B*c*d + A*d^2))/2 - (3*d*(a
*A*c*d - a*d*(c*C - B*d) - A*b*(c^2 + d^2)))/2))/2 + a*((-3*((b*d^2)/2 - (c
*(-(b*c) + a*d))/2)*(a*A*c*d - a*d*(c*C - B*d) - A*b*(c^2 + d^2)))/2 - (a*d
*((3*d*(b*c - a*d)*(B*c - (A - C)*d))/2 - (3*b*c*(c^2*C - B*c*d + A*d^2))/2
))/2 - (b*((-3*d^2*(a*A*c*d - a*d*(c*C - B*d) - A*b*(c^2 + d^2)))/2 - c*((3
*d*(b*c - a*d)*(B*c - (A - C)*d))/2 - (3*b*c*(c^2*C - B*c*d + A*d^2))/2)))/
2) - I*((a*(-(b*c) + a*d)*((-3*c*(b*c - a*d)*(B*c - (A - C)*d))/2 - (3*b*d*
(c^2*C - B*c*d + A*d^2))/2 - (3*d*(a*A*c*d - a*d*(c*C - B*d) - A*b*(c^2 + d
^2)))/2))/2 - b*((-3*((b*d^2)/2 - (c*(-(b*c) + a*d))/2)*(a*A*c*d - a*d*(c*C
 - B*d) - A*b*(c^2 + d^2)))/2 - (a*d*((3*d*(b*c - a*d)*(B*c - (A - C)*d))/2
 - (3*b*c*(c^2*C - B*c*d + A*d^2))/2))/2 - (b*((-3*d^2*(a*A*c*d - a*d*(c*C
- B*d) - A*b*(c^2 + d^2)))/2 - c*((3*d*(b*c - a*d)*(B*c - (A - C)*d))/2 - (
3*b*c*(c^2*C - B*c*d + A*d^2))/2)))/2)))*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/S
qrt[c - I*d]])/((-c + I*d)*f) - (I*Sqrt[c + I*d]*((b*(-(b*c) + a*d)*((-3*c*
(b*c - a*d)*(B*c - (A - C)*d))/2 - (3*b*d*(c^2*C - B*c*d + A*d^2))/2 - (3*d
*(a*A*c*d - a*d*(c*C - B*d) - A*b*(c^2 + d^2)))/2))/2 + a*((-3*((b*d^2)/2 -
 (c*(-(b*c) + a*d))/2)*(a*A*c*d - a*d*(c*C - B*d) - A*b*(c^2 + d^2)))/2 - (
a*d*((3*d*(b*c - a*d)*(B*c - (A - C)*d))/2 - (3*b*c*(c^2*C - B*c*d + A*d^2)
)/2))/2 - (b*((-3*d^2*(a*A*c*d - a*d*(c*C - B*d) - A*b*(c^2 + d^2)))/2 - c*
((3*d*(b*c - a*d)*(B*c - (A - C)*d))/2 - (3*b*c*(c^2*C - B*c*d + A*d^2))/2)
))/2) + I*((a*(-(b*c) + a*d)*((-3*c*(b*c - a*d)*(B*c - (A - C)*d))/2 - (3*b
```

```
*d*(c^2*C - B*c*d + A*d^2))/2 - (3*d*(a*A*c*d - a*d*(c*C - B*d) - A*b*(c^2
+ d^2)))/2))/2 - b*((-3*((b*d^2)/2 - (c*(-(b*c) + a*d))/2)*(a*A*c*d - a*d*(
c*C - B*d) - A*b*(c^2 + d^2)))/2 - (a*d*((3*d*(b*c - a*d)*(B*c - (A - C)*d)
)/2 - (3*b*c*(c^2*C - B*c*d + A*d^2))/2))/2 - (b*((-3*d^2*(a*A*c*d - a*d*(c
*C - B*d) - A*b*(c^2 + d^2)))/2 - c*((3*d*(b*c - a*d)*(B*c - (A - C)*d))/2
- (3*b*c*(c^2*C - B*c*d + A*d^2))/2)))/2)))*ArcTanh[Sqrt[c + d*Tan[e + f*x]
]/Sqrt[c + I*d]])/((-c - I*d)*f))/(a^2 + b^2) + (2*Sqrt[b*c - a*d]*(-1/2*(a
*b*(-(b*c) + a*d)*((-3*c*(b*c - a*d)*(B*c - (A - C)*d))/2 - (3*b*d*(c^2*C -
 B*c*d + A*d^2))/2 - (3*d*(a*A*c*d - a*d*(c*C - B*d) - A*b*(c^2 + d^2)))/2)
) + (a^2*b*((-3*d^2*(a*A*c*d - a*d*(c*C - B*d) - A*b*(c^2 + d^2)))/2 - c*((
3*d*(b*c - a*d)*(B*c - (A - C)*d))/2 - (3*b*c*(c^2*C - B*c*d + A*d^2))/2)))
/2 + b^2*((-3*((b*d^2)/2 - (c*(-(b*c) + a*d))/2)*(a*A*c*d - a*d*(c*C - B*d)
 - A*b*(c^2 + d^2)))/2 - (a*d*((3*d*(b*c - a*d)*(B*c - (A - C)*d))/2 - (3*b
*c*(c^2*C - B*c*d + A*d^2))/2))/2))*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x
]])/Sqrt[b*c - a*d]])/(Sqrt[b]*(a^2 + b^2)*(-(b*c) + a*d)*f)))/((-(b*c) + a
*d)*(c^2 + d^2)) - (2*((-3*d^2*(a*A*c*d - a*d*(c*C - B*d) - A*b*(c^2 + d^2)
))/2 - c*((3*d*(b*c - a*d)*(B*c - (A - C)*d))/2 - (3*b*c*(c^2*C - B*c*d + A
*d^2))/2)))/((-(b*c) + a*d)*(c^2 + d^2)*f*Sqrt[c + d*Tan[e + f*x]])))/(3*(-
(b*c) + a*d)*(c^2 + d^2))
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 45118 vs. 2(328) = 656.

Time = 0.23 (sec) , antiderivative size = 45119, normalized size of antiderivative = 123.61

method	result	size
derivativedivides	Expression too large to display	45119
default	Expression too large to display	45119

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))^(5/2)
,x,method=_RETURNVERBOSE)
```

```
[Out] result too large to display
```

Fricas [F(-1)]

Timed out.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^{5/2}}\,dx=\text{Timed out}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))
^(5/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^{5/2}}\,dx = \int \frac{A+B\tan{(e+fx)}+C\tan^{2}{(e+fx)}}{(a+b\tan{(e+fx)})\,(c+d\tan{(e+fx)})^{\frac{5}{2}}}\,dx$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e)
)**(5/2),x)
```

```
[Out] Integral((A + B*tan(e + f*x) + C*tan(e + f*x)**2)/((a + b*tan(e + f*x))*(c
+ d*tan(e + f*x))**(5/2)), x)
```

Maxima [F(-2)]

Exception generated.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^{5/2}}\,dx = \text{Exception raised: ValueError}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))
^(5/2),x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 detail
```

Giac [F]

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^{5/2}}\,dx = \int \frac{C\tan{(fx+e)}^{2}+B\tan{(fx+e)}+A}{(b\tan{(fx+e)}+a)(d\tan{(fx+e)}+c)^{\frac{5}{2}}}\,dx$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))
^(5/2),x, algorithm="giac")
```

```
[Out] sage0*x
```

Mupad [F(-1)]

Timed out.

$$\int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{(a + b\tan(e + fx))(c + d\tan(e + fx))^{5/2}}\,dx = \text{Hanged}$$

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))*(c + d*ta
n(e + f*x))^(5/2)),x)
```

```
[Out] \text{Hanged}
```

3.127 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}}\,dx$

Optimal result 1848
Rubi [A] (verified) 1849
Mathematica [B] (verified) 1853
Maple [B] (verified) 1854
Fricas [F(-1)] 1854
Sympy [F] 1854
Maxima [F(-2)] 1855
Giac [F(-1)] 1855
Mupad [F(-1)] 1855

Optimal result

Integrand size = 47, antiderivative size = 679

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}}\,dx =$$
$$-\frac{(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(a-ib)^2(c-id)^{5/2}f}-\frac{(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(a+ib)^2(c+id)^{5/2}f}$$
$$-\frac{b^{3/2}(7a^3bBd-5a^4Cd+b^4(2Bc-5Ad)+ab^3(4Ac-4cC+3Bd)-a^2b^2(2Bc+(9A+C)d))\,\text{arctanh}\left(\frac{\sqrt{b}\sqrt{\ }\cdots}{\cdots}\right)}{(a^2+b^2)^2\,(bc-ad)^{7/2}f}$$
$$-\frac{d(2b^2c(cC-Bd)-3abB(c^2+d^2)+a^2(5c^2C-2Bcd+3Cd^2)+A(2a^2d^2+b^2(3c^2+5d^2)))}{3\,(a^2+b^2)\,(bc-ad)^2\,(c^2+d^2)\,f(c+d\tan(e+fx))^{3/2}}$$
$$-\frac{Ab^2-a(bB-aC)}{(a^2+b^2)\,(bc-ad)f(a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}}$$
$$-\frac{d(2a^3d^2(Bc^2+2cCd-Bd^2)+2b^3c(2c^3C-3Bc^2d-Bd^3)-ab^2(Bc^4-4cCd^3+3Bd^4)+a^2b(5c^4C-6Bc\cdots}{(a^2+b^2)\,(bc-ad)^3\,(c^2+d^2)^2\cdots}$$

```
[Out] -(I*A+B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(a-I*b)^2/(c-I*d
)^(5/2)/f-(B-I*(A-C))*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))/(a+I*b)
^2/(c+I*d)^(5/2)/f-b^(3/2)*(7*a^3*b*B*d-5*a^4*C*d+b^4*(-5*A*d+2*B*c)+a*b^3*
(4*A*c+3*B*d-4*C*c)-a^2*b^2*(2*B*c+(9*A+C)*d))*arctanh(b^(1/2)*(c+d*tan(f*x
+e))^(1/2)/(-a*d+b*c)^(1/2))/(a^2+b^2)^2/(-a*d+b*c)^(7/2)/f-d*(2*a^3*d^2*(B
*c^2-B*d^2+2*C*c*d)+2*b^3*c*(-3*B*c^2*d-B*d^3+2*C*c^3)-a*b^2*(B*c^4+3*B*d^4
-4*C*c*d^3)+a^2*b*(-6*B*c^3*d-2*B*c*d^3+5*C*c^4+2*C*c^2*d^2+C*d^4)-A*(4*a^3
*c*d^3+4*a*b^2*c*d^3-4*a^2*b*d^2*(2*c^2+d^2)-b^3*(c^4+10*c^2*d^2+5*d^4)))/(
a^2+b^2)/(-a*d+b*c)^3/(c^2+d^2)^2/f/(c+d*tan(f*x+e))^(1/2)-1/3*d*(2*b^2*c*(
-B*d+C*c)-3*a*b*B*(c^2+d^2)+a^2*(-2*B*c*d+5*C*c^2+3*C*d^2)+A*(2*a^2*d^2+b^2
*(3*c^2+5*d^2)))/(a^2+b^2)/(-a*d+b*c)^2/(c^2+d^2)/f/(c+d*tan(f*x+e))^(3/2)+
```

```
(-A*b^2+a*(B*b-C*a))/(a^2+b^2)/(-a*d+b*c)/f/(a+b*tan(f*x+e))/(c+d*tan(f*x+e
))^(3/2)
```

Rubi [A] (verified)

Time = 5.83 (sec) , antiderivative size = 678, normalized size of antiderivative = 1.00, number of steps used = 14, number of rules used = 7, $\frac{\text{number of rules}}{\text{integrand size}} = 0.149$, Rules used = {3730, 3734, 3620, 3618, 65, 214, 3715}

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}}\,dx =$$
$$-\frac{d(2a^2Ad^2+a^2(-2Bcd+5c^2C+3Cd^2)-3abB(c^2+d^2)+Ab^2(3c^2+5d^2)+2b^2c(cC-Bd))}{3f\,(a^2+b^2)\,(c^2+d^2)\,(bc-ad)^2(c+d\tan(e+fx))^{3/2}}$$
$$-\frac{Ab^2-a(bB-aC)}{f\,(a^2+b^2)\,(bc-ad)(a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}}$$
$$-\frac{d(2a^3d^2(Bc^2-Bd^2+2cCd)+a^2b(-6Bc^3d-2Bcd^3+5c^4C+2c^2Cd^2+Cd^4)-A(4a^3cd^3-4a^2bd^2(2c^2}{f\,(a^2+b^2)\,(c^2+d^2)^2\,(bc-}$$
$$-\frac{b^{3/2}(-5a^4Cd+7a^3bBd-a^2b^2(d(9A+C)+2Bc)+ab^3(4Ac+3Bd-4cC)+b^4(2Bc-5Ad))\operatorname{arctanh}\Big(}{f\,(a^2+b^2)^2\,(bc-ad)^{7/2}}$$
$$-\frac{(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f(a-ib)^2(c-id)^{5/2}}-\frac{(B-i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f(a+ib)^2(c+id)^{5/2}}$$

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^2*(c + d*
Tan[e + f*x])^(5/2)),x]
```

```
[Out] -(((I*A + B - I*C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]])/((a - I
*b)^2*(c - I*d)^(5/2)*f)) - ((B - I*(A - C))*ArcTanh[Sqrt[c + d*Tan[e + f*x
]]/Sqrt[c + I*d]])/((a + I*b)^2*(c + I*d)^(5/2)*f) - (b^(3/2)*(7*a^3*b*B*d
 - 5*a^4*C*d + b^4*(2*B*c - 5*A*d) + a*b^3*(4*A*c - 4*c*C + 3*B*d) - a^2*b^2
*(2*B*c + (9*A + C)*d))*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c
 - a*d]])/((a^2 + b^2)^2*(b*c - a*d)^(7/2)*f) - (d*(2*a^2*A*d^2 + 2*b^2*c*(
c*C - B*d) - 3*a*b*B*(c^2 + d^2) + A*b^2*(3*c^2 + 5*d^2) + a^2*(5*c^2*C - 2
*B*c*d + 3*C*d^2)))/(3*(a^2 + b^2)*(b*c - a*d)^2*(c^2 + d^2)*f*(c + d*Tan[e
 + f*x])^(3/2)) - (A*b^2 - a*(b*B - a*C))/((a^2 + b^2)*(b*c - a*d)*f*(a + b
*Tan[e + f*x])*(c + d*Tan[e + f*x])^(3/2)) - (d*(2*a^3*d^2*(B*c^2 + 2*c*C*d
 - B*d^2) + 2*b^3*c*(2*c^3*C - 3*B*c^2*d - B*d^3) - a*b^2*(B*c^4 - 4*c*C*d^
3 + 3*B*d^4) + a^2*b*(5*c^4*C - 6*B*c^3*d + 2*c^2*C*d^2 - 2*B*c*d^3 + C*d^4
) - A*(4*a^3*c*d^3 + 4*a*b^2*c*d^3 - 4*a^2*b*d^2*(2*c^2 + d^2) - b^3*(c^4 +
 10*c^2*d^2 + 5*d^4))))/((a^2 + b^2)*(b*c - a*d)^3*(c^2 + d^2)^2*f*Sqrt[c +
 d*Tan[e + f*x]])
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3715

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_) + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :>
 Dist[A/f, Subst[Int[(a + b*x)^m*(c + d*x)^n, x], x, Tan[e + f*x]], x] /; F
reeQ[{a, b, c, d, e, f, A, C, m, n}, x] && EqQ[A, C]
```

Rule 3730

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 - a*(b*B - a*C))*(a + b*Tan[e +
 f*x])^(m + 1)*((c + d*Tan[e + f*x])^(n + 1)/(f*(m + 1)*(b*c - a*d)*(a^2 +
b^2))), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
```

```
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3734

```
Int[(((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2))/((a_.) + (b_.)*tan[(e_.)
+ (f_.)*(x_)]), x_Symbol] :> Dist[1/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n
*Simp[b*B + a*(A - C) + (a*B - b*(A - C))*Tan[e + f*x], x], x], x] + Dist[(
A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n*((1 + Tan[e
+ f*x]^2)/(a + b*Tan[e + f*x])), x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C
, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&
 !GtQ[n, 0] &&  !LeQ[n, -1]
```

Rubi steps

$$\text{integral} = -\frac{Ab^2-a(bB-aC)}{(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}} - \frac{\int \frac{\frac{1}{2}(5Ab^2d-2aA(bc-ad)-(bB-aC)(2bc+3ad))+(Ab-aB-bC)(bc-ad)\tan(e+fx)+\frac{5}{2}(Ab^2-a(bB-aC))d\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^{5/2}}dx}{(a^2+b^2)(bc-ad)}$$

$$= -\frac{d(2a^2Ad^2+2b^2c(cC-Bd)-3abB(c^2+d^2)+Ab^2(3c^2+5d^2)+a^2(5c^2C-2Bcd+3Cd^2))}{3(a^2+b^2)(bc-ad)^2(c^2+d^2)f(c+d\tan(e+fx))^{3/2}} - \frac{Ab^2-a(bB-aC)}{(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}} - \frac{2\int \frac{-\frac{3}{4}(2a^3d^2(Ac-cC+Bd)-a^2b(4A+C)d(c^2+d^2)+b^3(2Bc-5Ad)(c^2+d^2)-ab^2(2c^3C-Bc^2d+4cCd^2-3Bd^3-2A(c^3+2cd^2)))+ \cdots}{(\cdots}}{3(\cdots}$$

$$= -\frac{d(2a^2Ad^2+2b^2c(cC-Bd)-3abB(c^2+d^2)+Ab^2(3c^2+5d^2)+a^2(5c^2C-2Bcd+3Cd^2))}{3(a^2+b^2)(bc-ad)^2(c^2+d^2)f(c+d\tan(e+fx))^{3/2}} - \frac{Ab^2-a(bB-aC)}{(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}} - \frac{d(2a^3d^2(Bc^2+2cCd-Bd^2)+2b^3c(2c^3C-3Bc^2d-Bd^3)-ab^2(Bc^4-4cCd^3+3Bd^4)+a^2b(\cdots}{(a^2+b^2)(bc-ad\cdots} - \frac{4\int \frac{-\frac{3}{8}\left(b^4(2Bc-5Ad)(c^2+d^2)^2+2a^4d^3(c^2C-2Bcd-Cd^2-A(c^2-d^2))-2a^3bd^2(3c^3C-4Bc^2d+cCd^2-2Bd^3-Ac(3c^2+d^2))+a^2\cdots\right.}{\cdots}}{\cdots}$$

$$= -\frac{d(2a^2Ad^2+2b^2c(cC-Bd)-3abB(c^2+d^2)+Ab^2(3c^2+5d^2)+a^2(5c^2C-2Bcd+3Cd^2))}{3(a^2+b^2)(bc-ad)^2(c^2+d^2)f(c+d\tan(e+fx))^{3/2}} - \frac{Ab^2-a(bB-aC)}{(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}} - \frac{d(2a^3d^2(Bc^2+2cCd-Bd^2)+2b^3c(2c^3C-3Bc^2d-Bd^3)-ab^2(Bc^4-4cCd^3+3Bd^4)+a^2b(5}{(a^2+b^2)(bc-ad)^3} - \frac{4\int \frac{\frac{3}{4}(bc-ad)^3(a^2(c^2C-2Bcd-Cd^2-A(c^2-d^2))-b^2(c^2C-2Bcd-Cd^2-A(c^2-d^2))+2ab(2c(A-C)d-B(c^2-d^2)))-\frac{3}{4}(bc-ad)^3(2}{\sqrt{c+d\tan(e+fx)}}}{3(a^2+b^2)^2(bc-ad)^3(c} + \frac{(b^2(7a^3bBd-5a^4Cd+b^4(2Bc-5Ad)+ab^3(4Ac-4cC+3Bd)-a^2b^2(2Bc+(9A+C)d)))\int}{2(a^2+b^2)^2(bc-ad)^3}$$

$$= -\frac{d(2a^2Ad^2+2b^2c(cC-Bd)-3abB(c^2+d^2)+Ab^2(3c^2+5d^2)+a^2(5c^2C-2Bcd+3Cd^2))}{3(a^2+b^2)(bc-ad)^2(c^2+d^2)f(c+d\tan(e+fx))^{3/2}} - \frac{Ab^2-a(bB-aC)}{(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}} - \frac{d(2a^3d^2(Bc^2+2cCd-Bd^2)+2b^3c(2c^3C-3Bc^2d-Bd^3)-ab^2(Bc^4-4cCd^3+3Bd^4)+a^2b(5}{(a^2+b^2)(bc-ad)^3} + \frac{(A-iB-C)\int \frac{1+i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(a-ib)^2(c-id)^2} + \frac{(A+iB-C)\int \frac{1-i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(a+ib)^2(c+id)^2} + \frac{(b^2(7a^3bBd-5a^4Cd+b^4(2Bc-5Ad)+ab^3(4Ac-4cC+3Bd)-a^2b^2(2Bc+(9A+C)d)))\,\mathrm{Su}}{2(a^2+b^2)^2(bc-ad)^3f}$$

$$= -\frac{d(2a^2Ad^2+2b^2c(cC-Bd)-3abB(c^2+d^2)+Ab^2(3c^2+5d^2)+a^2(5c^2C-2Bcd+3Cd^2))}{3(a^2+b^2)(bc-ad)^2(c^2+d^2)f(c+d\tan(e+fx))^{3/2}} - \frac{Ab^2-a(bB-aC)}{(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}} - \frac{d(2a^3d^2(Bc^2+2cCd-Bd^2)+2b^3c(2c^3C-3Bc^2d-Bd^3)-ab^2(Bc^4-4cCd^3+3Bd^4)+a^2b(5}{(a^2+b^2)(bc-ad)^3} + \frac{(iA+B-iC)\mathrm{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c-idx}}\,dx, x, i\tan(e+fx)\right)}{2(a-ib)^2(c-id)^2f} - \frac{(i(A+iB-C))\mathrm{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c+idx}}\,dx, x, -i\tan(e+fx)\right)}{2(a+ib)^2(c+id)^2f} + \frac{(b^2(7a^3bBd-5a^4Cd+b^4(2Bc-5Ad)+ab^3(4Ac-4cC+3Bd)-a^2b^2(2Bc+(9A+C)d)))\,\mathrm{Su}}{(a^2+b^2)^2d(bc-ad)^3f}$$

$$= -\frac{b^{3/2}(7a^3bBd-5a^4Cd+b^4(2Bc-5Ad)+ab^3(4Ac-4cC+3Bd)-a^2b^2(2Bc+(9A+C)d))\,\mathrm{a}}{\left(a^2+b^2\right)^2(bc-ad)^{7/2}f}$$
$$-\frac{d(2a^2Ad^2+2b^2c(cC-Bd)-3abB(c^2+d^2)+Ab^2(3c^2+5d^2)+a^2(5c^2C-2Bcd+3Cd^2))}{3\left(a^2+b^2\right)(bc-ad)^2\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}}$$
$$-\frac{Ab^2-a(bB-aC)}{\left(a^2+b^2\right)(bc-ad)f(a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}}$$
$$-\frac{d(2a^3d^2(Bc^2+2cCd-Bd^2)+2b^3c(2c^3C-3Bc^2d-Bd^3)-ab^2(Bc^4-4cCd^3+3Bd^4)+a^2b(}{\left(a^2+b^2\right)(bc-ad}$$
$$+\frac{(i(iA+B-iC))\mathrm{Subst}\left(\int\frac{1}{-1-\frac{ic}{d}+\frac{ix^2}{d}}\,dx,x,\sqrt{c+d\tan(e+fx)}\right)}{(a-ib)^2(c-id)^2df}$$
$$-\frac{(A+iB-C)\mathrm{Subst}\left(\int\frac{1}{-1+\frac{ic}{d}-\frac{ix^2}{d}}\,dx,x,\sqrt{c+d\tan(e+fx)}\right)}{(a+ib)^2(c+id)^2df}$$

$$= -\frac{(iA+B-iC)\mathrm{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(a-ib)^2(c-id)^{5/2}f}-\frac{(B-i(A-C))\mathrm{arctanh}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{(a+ib)^2(c+id)^{5/2}f}$$
$$-\frac{b^{3/2}(7a^3bBd-5a^4Cd+b^4(2Bc-5Ad)+ab^3(4Ac-4cC+3Bd)-a^2b^2(2Bc+(9A+C)d))\,\mathrm{a}}{\left(a^2+b^2\right)^2(bc-ad)^{7/2}f}$$
$$-\frac{d(2a^2Ad^2+2b^2c(cC-Bd)-3abB(c^2+d^2)+Ab^2(3c^2+5d^2)+a^2(5c^2C-2Bcd+3Cd^2))}{3\left(a^2+b^2\right)(bc-ad)^2\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}}$$
$$-\frac{Ab^2-a(bB-aC)}{\left(a^2+b^2\right)(bc-ad)f(a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}}$$
$$-\frac{d(2a^3d^2(Bc^2+2cCd-Bd^2)+2b^3c(2c^3C-3Bc^2d-Bd^3)-ab^2(Bc^4-4cCd^3+3Bd^4)+a^2b(}{\left(a^2+b^2\right)(bc-ad}$$

Mathematica [B] (verified)

Both result and optimal contain complex but leaf count is larger than twice the leaf count of optimal. 6052 vs. 2(679) = 1358.

Time = 6.84 (sec) , antiderivative size = 6052, normalized size of antiderivative = 8.91

$$\int\frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}}\,dx=\text{Result too large to show}$$

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^2*(
c + d*Tan[e + f*x])^(5/2)),x]

[Out] Result too large to show
```

Maple [B] (verified)

Leaf count of result is larger than twice the leaf count of optimal. 67569 vs. 2(639) = 1278.

Time = 0.35 (sec) , antiderivative size = 67570, normalized size of antiderivative = 99.51

method	result	size
derivativedivides	Expression too large to display	67570
default	Expression too large to display	67570

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e))^(5/
2),x,method=_RETURNVERBOSE)
```

```
[Out] result too large to display
```

Fricas [F(-1)]

Timed out.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}}\,dx = \text{Timed out}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e
))^(5/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}}\,dx = \int \frac{A+B\tan{\left(e+fx\right)}+C\tan^2{\left(e+fx\right)}}{\left(a+b\tan{\left(e+fx\right)}\right)^2\left(c+d\tan{\left(e+fx\right)}\right)^{\frac{5}{2}}}\,dx$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e))**2/(c+d*tan(f*x
+e))**(5/2),x)
```

```
[Out] Integral((A + B*tan(e + f*x) + C*tan(e + f*x)**2)/((a + b*tan(e + f*x))**2*
(c + d*tan(e + f*x))**(5/2)), x)
```

Maxima [F(-2)]

Exception generated.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}}\,dx = \text{Exception raised: ValueError}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e
))^(5/2),x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 detail
```

Giac [F(-1)]

Timed out.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}}\,dx = \text{Timed out}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e
))^(5/2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}}\,dx = \text{Hanged}$$

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))^2*(c + d*
tan(e + f*x))^(5/2)),x)
```

```
[Out] \text{Hanged}
```

3.128 $\int (a+b\tan(e+fx))^{5/2}\sqrt{c+d\tan(e+fx)}(A+B\tan($

Optimal result 1856
Rubi [A] (verified) 1857
Mathematica [A] (verified) 1863
Maple [F(-1)] 1864
Fricas [F(-1)] 1864
Sympy [F] 1865
Maxima [F] 1865
Giac [F(-1)] 1865
Mupad [F(-1)] 1866

Optimal result

Integrand size = 49, antiderivative size = 679

$$\int (a+b\tan(e+fx))^{5/2}\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx =$$
$$-\frac{(a-ib)^{5/2}(iA+B-iC)\sqrt{c-id}\,\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f}$$
$$-\frac{(a+ib)^{5/2}(B-i(A-C))\sqrt{c+id}\,\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f}$$
$$-\frac{(5a^4Cd^4-20a^3bd^3(cC+2Bd)+30a^2b^2d^2(c^2C-4Bcd-8(A-C)d^2)-20ab^3d(c^3C-2Bc^2d+8c(A-C)}{64b^{3/}}$$
$$+\frac{(64b(a^2B-b^2B+2ab(A-C))\,d^3-(bc-ad)\,(16b(Ab+aB-bC)d^2+(bc-ad)(5bcC-8bBd-5aCd))}{64bd^3f}$$
$$+\frac{(16b(Ab+aB-bC)d^2+(bc-ad)(5bcC-8bBd-5aCd))\,\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{32d^3f}$$
$$-\frac{(5bcC-8bBd-5aCd)(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}{24d^2f}$$
$$+\frac{C(a+b\tan(e+fx))^{5/2}(c+d\tan(e+fx))^{3/2}}{4df}$$

```
[Out] -1/64*(5*a^4*C*d^4-20*a^3*b*d^3*(2*B*d+C*c)+30*a^2*b^2*d^2*(c^2*C-4*B*c*d-8
*(A-C)*d^2)-20*a*b^3*d*(c^3*C-2*B*c^2*d+8*c*(A-C)*d^2-16*B*d^3)+b^4*(5*c^4*
C-8*B*c^3*d+16*c^2*(A-C)*d^2+64*B*c*d^3+128*(A-C)*d^4))*arctanh(d^(1/2)*(a+
b*tan(f*x+e))^(1/2)/b^(1/2)/(c+d*tan(f*x+e))^(1/2))/b^(3/2)/d^(7/2)/f-(a-I*
b)^(5/2)*(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(
1/2)/(c+d*tan(f*x+e))^(1/2))*(c-I*d)^(1/2)/f-(a+I*b)^(5/2)*(B-I*(A-C))*arct
anh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/
```

```
2))*(c+I*d)^(1/2)/f+1/64*(64*b*(B*a^2-B*b^2+2*a*b*(A-C))*d^3-(-a*d+b*c)*(16
*b*(A*b+B*a-C*b)*d^2+(-a*d+b*c)*(-8*B*b*d-5*C*a*d+5*C*b*c)))*(a+b*tan(f*x+e
))^(1/2)*(c+d*tan(f*x+e))^(1/2)/b/d^3/f+1/32*(16*b*(A*b+B*a-C*b)*d^2+(-a*d+
b*c)*(-8*B*b*d-5*C*a*d+5*C*b*c))*(a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(3
/2)/d^3/f-1/24*(-8*B*b*d-5*C*a*d+5*C*b*c)*(a+b*tan(f*x+e))^(3/2)*(c+d*tan(f
*x+e))^(3/2)/d^2/f+1/4*C*(a+b*tan(f*x+e))^(5/2)*(c+d*tan(f*x+e))^(3/2)/d/f
```

Rubi [A] (verified)

Time = 11.77 (sec) , antiderivative size = 679, normalized size of antiderivative = 1.00, number of steps used = 16, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.163, Rules used = {3728, 3736, 6857, 65, 223, 212, 95, 214}

$$\int (a+b\tan(e+fx))^{5/2}\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))\,dx = \frac{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}(64bd^3(a^2B+2ab(A-C)-b^2B)-(bc-a\ldots}{64bd^3f}$$
$$-\frac{(5a^4Cd^4-20a^3bd^3(2Bd+cC)+30a^2b^2d^2(-8d^2(A-C)-4Bcd+c^2C)-20ab^3d(8cd^2(A-C)-2Bc^2\ldots}{64\ldots}$$
$$-\frac{(a-ib)^{5/2}\sqrt{c-id}(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f}$$
$$-\frac{(a+ib)^{5/2}\sqrt{c+id}(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f}$$
$$+\frac{\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}\left(16bd^2(aB+Ab-bC)+(bc-ad)(-5aCd-8bBd+5bcC)\right)}{32d^3f}$$
$$-\frac{(-5aCd-8bBd+5bcC)(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}{24d^2f}$$
$$+\frac{C(a+b\tan(e+fx))^{5/2}(c+d\tan(e+fx))^{3/2}}{4df}$$

```
[In] Int[(a + b*Tan[e + f*x])^(5/2)*Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x]
 + C*Tan[e + f*x]^2),x]
```

```
[Out] -(((a - I*b)^(5/2)*(I*A + B - I*C)*Sqrt[c - I*d]*ArcTanh[(Sqrt[c - I*d]*Sqr
t[a + b*Tan[e + f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/f) - ((a
+ I*b)^(5/2)*(B - I*(A - C))*Sqrt[c + I*d]*ArcTanh[(Sqrt[c + I*d]*Sqrt[a +
b*Tan[e + f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/f - ((5*a^4*C*d
^4 - 20*a^3*b*d^3*(c*C + 2*B*d) + 30*a^2*b^2*d^2*(c^2*C - 4*B*c*d - 8*(A -
C)*d^2) - 20*a*b^3*d*(c^3*C - 2*B*c^2*d + 8*c*(A - C)*d^2 - 16*B*d^3) + b^4
*(5*c^4*C - 8*B*c^3*d + 16*c^2*(A - C)*d^2 + 64*B*c*d^3 + 128*(A - C)*d^4))
*ArcTanh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c + d*Tan[e + f*x
]])])/(64*b^(3/2)*d^(7/2)*f) + ((64*b*(a^2*B - b^2*B + 2*a*b*(A - C))*d^3 -
```

```
 (b*c - a*d)*(16*b*(A*b + a*B - b*C)*d^2 + (b*c - a*d)*(5*b*c*C - 8*b*B*d -
 5*a*C*d)))*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(64*b*d^3*f)
 + ((16*b*(A*b + a*B - b*C)*d^2 + (b*c - a*d)*(5*b*c*C - 8*b*B*d - 5*a*C*d)
)*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(3/2))/(32*d^3*f) - ((5*b*c
*C - 8*b*B*d - 5*a*C*d)*(a + b*Tan[e + f*x])^(3/2)*(c + d*Tan[e + f*x])^(3/
2))/(24*d^2*f) + (C*(a + b*Tan[e + f*x])^(5/2)*(c + d*Tan[e + f*x])^(3/2))/
(4*d*f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 95

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 223

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
```

```
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3736

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[(a + b*ff*x)^m*(c + d*ff*x)^n*((A + B*ff*x + C*ff^2*x^2
)/(1 + ff^2*x^2)), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6857

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\text{integral} = \frac{C(a+b\tan(e+fx))^{5/2}(c+d\tan(e+fx))^{3/2}}{4df}$$
$$+\frac{\int (a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}\left(\frac{1}{2}(-5bcC+a(8A-3C)d)+4(Ab+aB-bC)d\tan(e+f\right.}{4d}$$
$$=-\frac{(5bcC-8bBd-5aCd)(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}{24d^2f}$$
$$+\frac{C(a+b\tan(e+fx))^{5/2}(c+d\tan(e+fx))^{3/2}}{4df}$$
$$+\frac{\int \sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}\left(\frac{3}{4}(a^2(16A-11C)d^2+b^2c(5cC-8Bd)-2abd(5cC\right.}{}$$
$$=\frac{(16b(Ab+aB-bC)d^2+(bc-ad)(5bcC-8bBd-5aCd))\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx}{32d^3f}$$
$$-\frac{(5bcC-8bBd-5aCd)(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}{24d^2f}$$
$$+\frac{C(a+b\tan(e+fx))^{5/2}(c+d\tan(e+fx))^{3/2}}{4df}$$
$$+\frac{\int \frac{\sqrt{c+d\tan(e+fx)}\left(\frac{3}{8}(a^3(64A-59C)d^3-a^2bd^2(15cC+104Bd)+ab^2d(15c^2C-32Bcd-48(A-C)d^2)-b^3c(5c^2C-8Bcd+16(A-C\right.}{}}{}$$

$$= \frac{\left(64b(a^2B - b^2B + 2ab(A-C))\,d^3 - (bc-ad)\left(16b(Ab + aB - bC)d^2 + (bc-ad)(5bcC - 8bBd - \right.\right.}{64bd^3f}$$

$$+ \frac{\left(16b(Ab + aB - bC)d^2 + (bc-ad)(5bcC - 8bBd - 5aCd)\right)\sqrt{a+b\tan(e+fx)}(c + d\tan(e+f}{32d^3f}$$

$$- \frac{(5bcC - 8bBd - 5aCd)(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}{24d^2f}$$

$$+ \frac{C(a+b\tan(e+fx))^{5/2}(c+d\tan(e+fx))^{3/2}}{4df}$$

$$+ \frac{\int \frac{-\frac{3}{16}\left(5a^4Cd^4 - 4a^3bd^3(32Ac - 27cC - 22Bd) + 6a^2b^2d^2\left(5c^2C + 44Bcd + 24(A-C)d^2\right) + b^4c\left(5c^3C - 8Bc^2d + 16c(A-C)d^2 - 64Bd^3\right) - 4}{}}{}$$

$$= \frac{\left(64b(a^2B - b^2B + 2ab(A-C))\,d^3 - (bc-ad)\left(16b(Ab + aB - bC)d^2 + (bc-ad)(5bcC - 8bBd - \right.\right.}{64bd^3f}$$

$$+ \frac{\left(16b(Ab + aB - bC)d^2 + (bc-ad)(5bcC - 8bBd - 5aCd)\right)\sqrt{a+b\tan(e+fx)}(c + d\tan(e+f}{32d^3f}$$

$$- \frac{(5bcC - 8bBd - 5aCd)(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}{24d^2f}$$

$$+ \frac{C(a+b\tan(e+fx))^{5/2}(c+d\tan(e+fx))^{3/2}}{4df}$$

$$+ \frac{\text{Subst}\left(\int \frac{-\frac{3}{16}\left(5a^4Cd^4 - 4a^3bd^3(32Ac - 27cC - 22Bd) + 6a^2b^2d^2\left(5c^2C + 44Bcd + 24(A-C)d^2\right) + b^4c\left(5c^3C - 8Bc^2d + 16c(A-C)d^2 - 6\right.\right.}{}\right.}{}$$

$$= \frac{\left(64b(a^2B - b^2B + 2ab(A-C))\,d^3 - (bc-ad)\left(16b(Ab + aB - bC)d^2 + (bc-ad)(5bcC - 8bBd - \right.\right.}{64bd^3f}$$

$$+ \frac{\left(16b(Ab + aB - bC)d^2 + (bc-ad)(5bcC - 8bBd - 5aCd)\right)\sqrt{a+b\tan(e+fx)}(c + d\tan(e+f}{32d^3f}$$

$$- \frac{(5bcC - 8bBd - 5aCd)(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}{24d^2f}$$

$$+ \frac{C(a+b\tan(e+fx))^{5/2}(c+d\tan(e+fx))^{3/2}}{4df}$$

$$+ \frac{\text{Subst}\left(\int \left(-\frac{3\left(5a^4Cd^4 - 20a^3bd^3(cC + 2Bd) + 30a^2b^2d^2\left(c^2C - 4Bcd - 8(A-C)d^2\right) - 20ab^3d\left(c^3C - 2Bc^2d + 8c(A-C)d^2 - 16Bd^3\right) + \right.\right.}{16\sqrt{a+bx}\sqrt{c+dx}}\right.\right.}{}$$

$$= \frac{(64b(a^2B - b^2B + 2ab(A-C))\, d^3 - (bc-ad)\,(16b(Ab+aB-bC)d^2 + (bc-ad)(5bcC - 8bBd}{64bd^3f}$$
$$+ \frac{(16b(Ab+aB-bC)d^2 + (bc-ad)(5bcC - 8bBd - 5aCd))\,\sqrt{a+b\tan(e+fx)}(c+d\tan(e+}{32d^3f}$$
$$- \frac{(5bcC - 8bBd - 5aCd)(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}{24d^2f}$$
$$+ \frac{C(a+b\tan(e+fx))^{5/2}(c+d\tan(e+fx))^{3/2}}{4df}$$
$$+ \frac{\text{Subst}\Big(\int \frac{bd^3(a^3(Ac-cC-Bd)-3ab^2(Ac-cC-Bd)-3a^2b(Bc+(A-C)d)+b^3(Bc+(A-C)d))+bd^3(3a^2b(Ac-cC-Bd)-b^3(Ac-}{\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}}{bd^3f}$$
$$- \frac{(5a^4Cd^4 - 20a^3bd^3(cC+2Bd) + 30a^2b^2d^2(c^2C - 4Bcd - 8(A-C)d^2) - 20ab^3d(c^3C - 2Bc^2d}{}$$

$$= \frac{(64b(a^2B - b^2B + 2ab(A-C))\, d^3 - (bc-ad)\,(16b(Ab+aB-bC)d^2 + (bc-ad)(5bcC - 8bBd}{64bd^3f}$$
$$+ \frac{(16b(Ab+aB-bC)d^2 + (bc-ad)(5bcC - 8bBd - 5aCd))\,\sqrt{a+b\tan(e+fx)}(c+d\tan(e+}{32d^3f}$$
$$- \frac{(5bcC - 8bBd - 5aCd)(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}{24d^2f}$$
$$+ \frac{C(a+b\tan(e+fx))^{5/2}(c+d\tan(e+fx))^{3/2}}{4df}$$
$$+ \frac{\text{Subst}\Big(\int \Big(\frac{-bd^3(3a^2b(Ac-cC-Bd)-b^3(Ac-cC-Bd)+a^3(Bc+(A-C)d)-3ab^2(Bc+(A-C)d))+ibd^3(a^3(Ac-cC-Bd)-3ab^2(}{2(i-x)\sqrt{a+bx}\sqrt{c+dx}}}{}$$
$$- \frac{(5a^4Cd^4 - 20a^3bd^3(cC+2Bd) + 30a^2b^2d^2(c^2C - 4Bcd - 8(A-C)d^2) - 20ab^3d(c^3C - 2Bc^2d}{}$$

$$= \frac{(64b(a^2B - b^2B + 2ab(A-C))\,d^3 - (bc-ad)\,(16b(Ab+aB-bC)d^2 + (bc-ad)(5bcC - 8bBd -}{64bd^3f}$$

$$+ \frac{(16b(Ab+aB-bC)d^2 + (bc-ad)(5bcC - 8bBd - 5aCd))\,\sqrt{a+b\tan(e+fx)}(c+d\tan(e+f}{32d^3f}$$

$$- \frac{(5bcC - 8bBd - 5aCd)(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}{24d^2f}$$

$$+ \frac{C(a+b\tan(e+fx))^{5/2}(c+d\tan(e+fx))^{3/2}}{4df}$$

$$+ \frac{((a-ib)^3(A-iB-C)(ic+d))\,\mathrm{Subst}\left(\int \frac{1}{(i+x)\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e+fx)\right)}{2f}$$

$$- \frac{(5a^4Cd^4 - 20a^3bd^3(cC+2Bd) + 30a^2b^2d^2(c^2C - 4Bcd - 8(A-C)d^2) - 20ab^3d(c^3C - 2Bc^2d -}{}$$

$$+ \frac{(-bd^3(3a^2b(Ac - cC - Bd) - b^3(Ac - cC - Bd) + a^3(Bc + (A-C)d) - 3ab^2(Bc + (A-C)d)}{}$$

$$=$$

$$- \frac{(5a^4Cd^4 - 20a^3bd^3(cC+2Bd) + 30a^2b^2d^2(c^2C - 4Bcd - 8(A-C)d^2) - 20ab^3d(c^3C - 2Bc^2d -}{}$$

$$+ \frac{(64b(a^2B - b^2B + 2ab(A-C))\,d^3 - (bc-ad)\,(16b(Ab+aB-bC)d^2 + (bc-ad)(5bcC - 8bBd}{64bd^3f}$$

$$+ \frac{(16b(Ab+aB-bC)d^2 + (bc-ad)(5bcC - 8bBd - 5aCd))\,\sqrt{a+b\tan(e+fx)}(c+d\tan(e+f}{32d^3f}$$

$$- \frac{(5bcC - 8bBd - 5aCd)(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}{24d^2f}$$

$$+ \frac{C(a+b\tan(e+fx))^{5/2}(c+d\tan(e+fx))^{3/2}}{4df}$$

$$+ \frac{((a-ib)^3(A-iB-C)(ic+d))\,\mathrm{Subst}\left(\int \frac{1}{-a+ib-(-c+id)x^2}\,dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{f}$$

$$+ \frac{(-bd^3(3a^2b(Ac - cC - Bd) - b^3(Ac - cC - Bd) + a^3(Bc + (A-C)d) - 3ab^2(Bc + (A-C)d)}{}$$

$$= -\frac{(a-ib)^{5/2}(iA+B-iC)\sqrt{c-id}\,\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f}$$
$$-\frac{(a+ib)^{5/2}(B-i(A-C))\sqrt{c+id}\,\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f}$$
$$-\frac{(5a^4Cd^4-20a^3bd^3(cC+2Bd)+30a^2b^2d^2(c^2C-4Bcd-8(A-C)d^2)-20ab^3d(c^3C-2Bc^2d}{}$$
$$+\frac{(64b(a^2B-b^2B+2ab(A-C))\,d^3-(bc-ad)\,(16b(Ab+aB-bC)d^2+(bc-ad)(5bcC-8bB}{64bd^3f}$$
$$+\frac{(16b(Ab+aB-bC)d^2+(bc-ad)(5bcC-8bBd-5aCd))\,\sqrt{a+b\tan(e+fx)}(c+d\tan(e+}{32d^3f}$$
$$-\frac{(5bcC-8bBd-5aCd)(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}{24d^2f}$$
$$+\frac{C(a+b\tan(e+fx))^{5/2}(c+d\tan(e+fx))^{3/2}}{4df}$$

Mathematica [A] (verified)

Time = 10.25 (sec) , antiderivative size = 1202, normalized size of antiderivative = 1.77

$$\int(a+b\tan(e+fx))^{5/2}\sqrt{c+d\tan(e+fx)}\big(A+B\tan(e+fx)$$
$$+C\tan^2(e+fx)\big)\ dx=\frac{C(a+b\tan(e+fx))^{5/2}(c+d\tan(e+fx))^{3/2}}{4df}$$

$$+\frac{\frac{(-5bcC+8bBd+5aCd)(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}{6df}+\frac{3\left(16b(Ab+aB-bC)d^2+(bc-ad)(5bcC-8bBd-5aCd)\right)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx}{8df}}{}$$

```
[In] Integrate[(a + b*Tan[e + f*x])^(5/2)*Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e
+ f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] (C*(a + b*Tan[e + f*x])^(5/2)*(c + d*Tan[e + f*x])^(3/2))/(4*d*f) + (((-5*b
*c*C + 8*b*B*d + 5*a*C*d)*(a + b*Tan[e + f*x])^(3/2)*(c + d*Tan[e + f*x])^(
3/2))/(6*d*f) + ((3*(16*b*(A*b + a*B - b*C)*d^2 + (b*c - a*d)*(5*b*c*C - 8*
b*B*d - 5*a*C*d))*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(3/2))/(8*d
*f) + (((24*b*(a^2*B - b^2*B + 2*a*b*(A - C))*d^3 - (3*(b*c - a*d)*(16*b*(A
*b + a*B - b*C)*d^2 + (b*c - a*d)*(5*b*c*C - 8*b*B*d - 5*a*C*d)))/8)*Sqrt[a
 + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(b*f) + ((24*b*d^3*(b*(3*a^2*b
*(A*c - c*C - B*d) - b^3*(A*c - c*C - B*d) + a^3*(B*c + (A - C)*d) - 3*a*b^
2*(B*c + (A - C)*d)) + Sqrt[-b^2]*(a^3*(A*c - c*C - B*d) - 3*a*b^2*(A*c - c
```

```
*C - B*d) - 3*a^2*b*(B*c + (A - C)*d) + b^3*(B*c + (A - C)*d)))*ArcTanh[(Sq
rt[-c + (Sqrt[-b^2]*d)/b]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[-a + Sqrt[-b^2]]*
Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[-c + (Sqrt[-b^2]*d)
/b]) - (24*b*d^3*(b*(3*a^2*b*(A*c - c*C - B*d) - b^3*(A*c - c*C - B*d) + a^
3*(B*c + (A - C)*d) - 3*a*b^2*(B*c + (A - C)*d)) - Sqrt[-b^2]*(a^3*(A*c - c
*C - B*d) - 3*a*b^2*(A*c - c*C - B*d) - 3*a^2*b*(B*c + (A - C)*d) + b^3*(B*
c + (A - C)*d)))*ArcTanh[(Sqrt[c + (Sqrt[-b^2]*d)/b]*Sqrt[a + b*Tan[e + f*x
]])/(Sqrt[a + Sqrt[-b^2]]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[a + Sqrt[-b^2]]
*Sqrt[c + (Sqrt[-b^2]*d)/b]) - (3*Sqrt[b]*Sqrt[c - (a*d)/b]*Sqrt[(c/(c - (a
*d)/b) - (a*d)/(b*(c - (a*d)/b)))^(-1)]*Sqrt[c/(c - (a*d)/b) - (a*d)/(b*(c
- (a*d)/b))]*(5*a^4*C*d^4 - 20*a^3*b*d^3*(c*C + 2*B*d) + 30*a^2*b^2*d^2*(c^
2*C - 4*B*c*d - 8*(A - C)*d^2) - 20*a*b^3*d*(c^3*C - 2*B*c^2*d + 8*c*(A - C
)*d^2 - 16*B*d^3) + b^4*(5*c^4*C - 8*B*c^3*d + 16*c^2*(A - C)*d^2 + 64*B*c*
d^3 + 128*(A - C)*d^4))*ArcSinh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]
*Sqrt[c - (a*d)/b]*Sqrt[c/(c - (a*d)/b) - (a*d)/(b*(c - (a*d)/b))])]*Sqrt[(
c + d*Tan[e + f*x])/(c - (a*d)/b)])/(8*Sqrt[d]*Sqrt[c + d*Tan[e + f*x]]))/(
b^2*f))/(2*d))/(3*d))/(4*d)
```

Maple [F(-1)]

Timed out.

$$\int \sqrt{c+d\tan(fx+e)}\,(a+b\tan(fx+e))^{\frac{5}{2}}\left(A+B\tan(fx+e)+C\tan(fx+e)^2\right)dx$$

```
[In] int((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x)
```

```
[Out] int((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x)
```

Fricas [F(-1)]

Timed out.

$$\int (a+b\tan(e+fx))^{5/2}\sqrt{c+d\tan(e+fx)}\big(A+B\tan(e+fx)+C\tan^2(e+fx)\big)\,dx=\text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*t
an(f*x+e)^2),x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int (a+b\tan(e+fx))^{5/2}\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\int (a+b\tan{(e+fx)})^{\frac{5}{2}}\sqrt{c+d\tan{(e+fx)}}\left(A+B\tan{(e+fx)}+C\tan^2{(e+fx)}\right)\,dx$$

```
[In] integrate((c+d*tan(f*x+e))**(1/2)*(a+b*tan(f*x+e))**(5/2)*(A+B*tan(f*x+e)+C
*tan(f*x+e)**2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**(5/2)*sqrt(c + d*tan(e + f*x))*(A + B*tan(e
+ f*x) + C*tan(e + f*x)**2), x)
```

Maxima [F]

$$\int (a+b\tan(e+fx))^{5/2}\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\int \left(C\tan{(fx+e)}^2+B\tan{(fx+e)}+A\right)(b\tan{(fx+e)}+a)^{\frac{5}{2}}\sqrt{d\tan{(fx+e)}+c}$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*t
an(f*x+e)^2),x, algorithm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)^(5/2
)*sqrt(d*tan(f*x + e) + c), x)
```

Giac [F(-1)]

Timed out.

$$\int (a+b\tan(e+fx))^{5/2}\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*t
an(f*x+e)^2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int (a+b\tan(e+fx))^{5/2}\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))\,dx=\int (a+b\tan(e+f\,x))^{5/2}\,\sqrt{c+d\tan(e+f\,x)}\,\left(C\tan(e+f\,x)^2+B\tan(e+f\,x)+A\right)\,d$$

```
[In] int((a + b*tan(e + f*x))^(5/2)*(c + d*tan(e + f*x))^(1/2)*(A + B*tan(e + f*
x) + C*tan(e + f*x)^2),x)
[Out] int((a + b*tan(e + f*x))^(5/2)*(c + d*tan(e + f*x))^(1/2)*(A + B*tan(e + f*
x) + C*tan(e + f*x)^2), x)
```

3.129 $\int (a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}(A+B\tan$

Optimal result 1867
Rubi [A] (verified) 1868
Mathematica [A] (verified) 1873
Maple [F(-1)] 1874
Fricas [F(-1)] 1874
Sympy [F] 1874
Maxima [F] 1875
Giac [F(-1)] 1875
Mupad [F(-1)] 1875

Optimal result

Integrand size = 49, antiderivative size = 505

$$\int (a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx =$$

$$-\frac{(a-ib)^{3/2}(iA+B-iC)\sqrt{c-id}\,\mathrm{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f}$$

$$+\frac{(a+ib)^{3/2}(iA-B-iC)\sqrt{c+id}\,\mathrm{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f}$$

$$-\frac{(a^3Cd^3-3a^2bd^2(cC+2Bd)+3ab^2d(c^2C-4Bcd-8(A-C)d^2)-b^3(c^3C-2Bc^2d+8c(A-C)d^2-16}{8b^{3/2}d^{5/2}f}$$

$$+\frac{(8b(Ab+aB-bC)d^2+(bc-ad)(bcC-2bBd-aCd))\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{8bd^2f}$$

$$-\frac{(bcC-2bBd-aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{4d^2f}$$

$$+\frac{C(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}{3df}$$

```
[Out] -1/8*(a^3*C*d^3-3*a^2*b*d^2*(2*B*d+C*c)+3*a*b^2*d*(c^2*C-4*B*c*d-8*(A-C)*d^
2)-b^3*(c^3*C-2*B*c^2*d+8*c*(A-C)*d^2-16*B*d^3))*arctanh(d^(1/2)*(a+b*tan(f
*x+e))^(1/2)/b^(1/2)/(c+d*tan(f*x+e))^(1/2))/b^(3/2)/d^(5/2)/f-(a-I*b)^(3/2
)*(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(1/2)/(c
+d*tan(f*x+e))^(1/2))*(c-I*d)^(1/2)/f+(a+I*b)^(3/2)*(I*A-B-I*C)*arctanh((c+
I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))*(c+
I*d)^(1/2)/f+1/8*(8*b*(A*b+B*a-C*b)*d^2+(-a*d+b*c)*(-2*B*b*d-C*a*d+C*b*c))*
(a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)/b/d^2/f-1/4*(-2*B*b*d-C*a*d+C
*b*c)*(a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(3/2)/d^2/f+1/3*C*(a+b*tan(f*
x+e))^(3/2)*(c+d*tan(f*x+e))^(3/2)/d/f
```

Rubi [A] (verified)

Time = 8.63 (sec) , antiderivative size = 505, normalized size of antiderivative = 1.00, number of steps used = 15, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}} = 0.163$, Rules used = {3728, 3736, 6857, 65, 223, 212, 95, 214}

$$\int (a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx =$$

$$-\frac{(a^3Cd^3-3a^2bd^2(2Bd+cC)+3ab^2d(-8d^2(A-C)-4Bcd+c^2C)-(b^3(8cd^2(A-C)-2Bc^2d-16Bd^3+}{8b^{3/2}d^{5/2}f}$$

$$-\frac{(a-ib)^{3/2}\sqrt{c-id}(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f}$$

$$+\frac{(a+ib)^{3/2}\sqrt{c+id}(iA-B-iC)\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f}$$

$$+\frac{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}(8bd^2(aB+Ab-bC)+(bc-ad)(-aCd-2bBd+bcC))}{8bd^2f}$$

$$-\frac{(-aCd-2bBd+bcC)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{4d^2f}$$

$$+\frac{C(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}{3df}$$

```
[In] Int[(a + b*Tan[e + f*x])^(3/2)*Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x]
 + C*Tan[e + f*x]^2),x]
```

```
[Out] -(((a - I*b)^(3/2)*(I*A + B - I*C)*Sqrt[c - I*d]*ArcTanh[(Sqrt[c - I*d]*Sqr
t[a + b*Tan[e + f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/f) + ((a
+ I*b)^(3/2)*(I*A - B - I*C)*Sqrt[c + I*d]*ArcTanh[(Sqrt[c + I*d]*Sqrt[a +
b*Tan[e + f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/f - ((a^3*C*d^3
 - 3*a^2*b*d^2*(c*C + 2*B*d) + 3*a*b^2*d*(c^2*C - 4*B*c*d - 8*(A - C)*d^2)
- b^3*(c^3*C - 2*B*c^2*d + 8*c*(A - C)*d^2 - 16*B*d^3))*ArcTanh[(Sqrt[d]*Sq
rt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])])/(8*b^(3/2)*d^(
5/2)*f) + ((8*b*(A*b + a*B - b*C)*d^2 + (b*c - a*d)*(b*c*C - 2*b*B*d - a*C*
d))*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(8*b*d^2*f) - ((b*c*
C - 2*b*B*d - a*C*d)*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(3/2))/(
4*d^2*f) + (C*(a + b*Tan[e + f*x])^(3/2)*(c + d*Tan[e + f*x])^(3/2))/(3*d*f
)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
```

```
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 95

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 223

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3736

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[(a + b*ff*x)^m*(c + d*ff*x)^n*((A + B*ff*x + C*ff^2*x^2
)/(1 + ff^2*x^2)), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
```

```
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6857

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\text{integral} = \frac{C(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}{3df} + \frac{\int \sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}\left(-\frac{3}{2}(bcC-a(2A-C)d)+3(Ab+aB-bC)d\tan(e+fx)-\right.}{3d}$$

$$= -\frac{(bcC-2bBd-aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{4d^2f} + \frac{C(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}{3df} + \frac{\int \frac{\sqrt{c+d\tan(e+fx)}\left(\frac{3}{4}\left(a^2(8A-7C)d^2+b^2c(cC-2Bd)-2abd(cC+3Bd)\right)+6\left(a^2B-b^2B+2ab(A-C)\right)d^2\tan(e+fx)+\frac{3}{4}\left(8b(Ab+aB-b\right.\right.}{\sqrt{a+b\tan(e+fx)}}}{6d^2}$$

$$= \frac{\left(8b(Ab+aB-bC)d^2+(bc-ad)(bcC-2bBd-aCd)\right)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{8bd^2f} - \frac{(bcC-2bBd-aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{4d^2f} + \frac{C(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}{3df} + \frac{\int \frac{-\frac{3}{8}\left(a^3Cd^3-a^2bd^2(16Ac-13cC-10Bd)-b^3c\left(c^2C-2Bcd-8(A-C)d^2\right)+ab^2d\left(3c^2C+20Bcd+8(A-C)d^2\right)\right)+6bd^2\left(2ab(Ac-cC-B\right.}{\sqrt{}}}{}$$

$$= \frac{\left(8b(Ab+aB-bC)d^2+(bc-ad)(bcC-2bBd-aCd)\right)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{8bd^2f} - \frac{(bcC-2bBd-aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{4d^2f} + \frac{C(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}{3df} + \frac{\text{Subst}\left(\int \frac{-\frac{3}{8}\left(a^3Cd^3-a^2bd^2(16Ac-13cC-10Bd)-b^3c\left(c^2C-2Bcd-8(A-C)d^2\right)+ab^2d\left(3c^2C+20Bcd+8(A-C)d^2\right)\right)+6bd^2\left(2ab(A\right.}{}\right.}{}$$

$$= \frac{(8b(Ab+aB-bC)d^2+(bc-ad)(bcC-2bBd-aCd))\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{8bd^2f} - \frac{(bcC-2bBd-aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{4d^2f} + \frac{C(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}{3df} + \frac{\text{Subst}\left(\int\left(-\frac{3(a^3Cd^3-3a^2bd^2(cC+2Bd)+3ab^2d(c^2C-4Bcd-8(A-C)d^2)-b^3(c^3C-2Bc^2d+8c(A-C)d^2-16Bd^3))}{8\sqrt{a+bx}\sqrt{c+dx}}+\frac{6(bd^2}{}\right.\right.}{}$$

$$= \frac{(8b(Ab+aB-bC)d^2+(bc-ad)(bcC-2bBd-aCd))\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{8bd^2f} - \frac{(bcC-2bBd-aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{4d^2f} + \frac{C(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}{3df} + \frac{\text{Subst}\left(\int\frac{bd^2(a^2(Ac-cC-Bd)-b^2(Ac-cC-Bd)-2ab(Bc+(A-C)d))+bd^2(2ab(Ac-cC-Bd)+a^2(Bc+(A-C)d)-b^2(Bc+(A-}{\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}\right.}{bd^2f} - \frac{(a^3Cd^3-3a^2bd^2(cC+2Bd)+3ab^2d(c^2C-4Bcd-8(A-C)d^2)-b^3(c^3C-2Bc^2d+8c(A-}{16bd^2f}$$

$$= \frac{(8b(Ab+aB-bC)d^2+(bc-ad)(bcC-2bBd-aCd))\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{8bd^2f} - \frac{(bcC-2bBd-aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{4d^2f} + \frac{C(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}{3df} + \frac{\text{Subst}\left(\int\left(\frac{ibd^2(a^2(Ac-cC-Bd)-b^2(Ac-cC-Bd)-2ab(Bc+(A-C)d))-bd^2(2ab(Ac-cC-Bd)+a^2(Bc+(A-C)d)-b^2(Bc+(}{2(i-x)\sqrt{a+bx}\sqrt{c+dx}}\right.\right.}{} - \frac{(a^3Cd^3-3a^2bd^2(cC+2Bd)+3ab^2d(c^2C-4Bcd-8(A-C)d^2)-b^3(c^3C-2Bc^2d+8c(A-}{8b^2d^2f}$$

$$= \frac{(8b(Ab+aB-bC)d^2+(bc-ad)(bcC-2bBd-aCd))\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{8bd^2f}$$
$$-\frac{(bcC-2bBd-aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{4d^2f}$$
$$+\frac{C(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}{3df}$$
$$+\frac{((a-ib)^2(A-iB-C)(ic+d))\operatorname{Subst}\left(\int\frac{1}{(i+x)\sqrt{a+bx}\sqrt{c+dx}}\,dx,x,\tan(e+fx)\right)}{2f}$$
$$-\frac{(a^3Cd^3-3a^2bd^2(cC+2Bd)+3ab^2d(c^2C-4Bcd-8(A-C)d^2)-b^3(c^3C-2Bc^2d+8c(A-C}{8b^2d^2f}$$
$$+\frac{(ibd^2(a^2(Ac-cC-Bd)-b^2(Ac-cC-Bd)-2ab(Bc+(A-C)d))-bd^2(2ab(Ac-cC-Bd}{2bd^2}$$
$$=$$
$$-\frac{(a^3Cd^3-3a^2bd^2(cC+2Bd)+3ab^2d(c^2C-4Bcd-8(A-C)d^2)-b^3(c^3C-2Bc^2d+8c(A-C}{8b^{3/2}d^{5/2}f}$$
$$+\frac{(8b(Ab+aB-bC)d^2+(bc-ad)(bcC-2bBd-aCd))\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{8bd^2f}$$
$$-\frac{(bcC-2bBd-aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{4d^2f}$$
$$+\frac{C(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}{3df}$$
$$+\frac{((a-ib)^2(A-iB-C)(ic+d))\operatorname{Subst}\left(\int\frac{1}{-a+ib-(-c+id)x^2}\,dx,x,\frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{f}$$
$$+\frac{(ibd^2(a^2(Ac-cC-Bd)-b^2(Ac-cC-Bd)-2ab(Bc+(A-C)d))-bd^2(2ab(Ac-cC-Bd}{bd^2f}$$

$$= -\frac{(a-ib)^{3/2}(iA+B-iC)\sqrt{c-id}\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f}$$
$$-\frac{(a+ib)^{3/2}(B-i(A-C))\sqrt{c+id}\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f}$$
$$-\frac{(a^3Cd^3-3a^2bd^2(cC+2Bd)+3ab^2d(c^2C-4Bcd-8(A-C)d^2)-b^3(c^3C-2Bc^2d+8c(A-}{8b^{3/2}d^{5/2}f}$$
$$+\frac{(8b(Ab+aB-bC)d^2+(bc-ad)(bcC-2bBd-aCd))\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+f}{8bd^2f}$$
$$-\frac{(bcC-2bBd-aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{4d^2f}$$
$$+\frac{C(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}{3df}$$

Mathematica [A] (verified)

Time = 9.10 (sec) , antiderivative size = 835, normalized size of antiderivative = 1.65

$$\int(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))\,dx=\frac{C(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}{3df}$$
$$+\frac{-\frac{3(bcC-2bBd-aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{4df}+\frac{\frac{3\left(8b(Ab+aB-bC)d^2+(bc-ad)(bcC-2bBd-aCd)\right)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{4bf}+-}{}$$

```
[In] Integrate[(a + b*Tan[e + f*x])^(3/2)*Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e
+ f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] (C*(a + b*Tan[e + f*x])^(3/2)*(c + d*Tan[e + f*x])^(3/2))/(3*d*f) + ((-3*(b
*c*C - 2*b*B*d - a*C*d)*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(3/2)
)/(4*d*f) + ((3*(8*b*(A*b + a*B - b*C)*d^2 + (b*c - a*d)*(b*c*C - 2*b*B*d -
 a*C*d))*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(4*b*f) + ((6*b
*d^2*(Sqrt[-b^2]*(a^2*(A*c - c*C - B*d) - b^2*(A*c - c*C - B*d) - 2*a*b*(B*
c + (A - C)*d)) + b*(2*a*b*(A*c - c*C - B*d) + a^2*(B*c + (A - C)*d) - b^2*
(B*c + (A - C)*d)))*ArcTanh[(Sqrt[-c + (Sqrt[-b^2]*d)/b]*Sqrt[a + b*Tan[e +
 f*x]])/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[-a + Sqrt[
-b^2]]*Sqrt[-c + (Sqrt[-b^2]*d)/b]) + (6*b*d^2*(Sqrt[-b^2]*(a^2*(A*c - c*C
- B*d) - b^2*(A*c - c*C - B*d) - 2*a*b*(B*c + (A - C)*d)) - b*(2*a*b*(A*c -
 c*C - B*d) + a^2*(B*c + (A - C)*d) - b^2*(B*c + (A - C)*d)))*ArcTanh[(Sqrt
[c + (Sqrt[-b^2]*d)/b]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + Sqrt[-b^2]]*Sqrt
```

```
[c + d*Tan[e + f*x]])])/(Sqrt[a + Sqrt[-b^2]]*Sqrt[c + (Sqrt[-b^2]*d)/b]) -
 (3*Sqrt[b]*Sqrt[c - (a*d)/b]*(a^3*C*d^3 - 3*a^2*b*d^2*(c*C + 2*B*d) + 3*a*
b^2*d*(c^2*C - 4*B*c*d - 8*(A - C)*d^2) - b^3*(c^3*C - 2*B*c^2*d + 8*c*(A -
 C)*d^2 - 16*B*d^3))*ArcSinh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sq
rt[c - (a*d)/b])]*Sqrt[(b*c + b*d*Tan[e + f*x])/(b*c - a*d)])/(4*Sqrt[d]*Sq
rt[c + d*Tan[e + f*x]]))/(b^2*f))/(2*d))/(3*d)
```

Maple [F(-1)]

Timed out.

$$\int \sqrt{c+d\tan\left(fx+e\right)}\left(a+b\tan\left(fx+e\right)\right)^{\frac{3}{2}}\left(A+B\tan\left(fx+e\right)+C\tan\left(fx+e\right)^{2}\right)dx$$

```
[In] int((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x)
```

```
[Out] int((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x)
```

Fricas [F(-1)]

Timed out.

$$\int (a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*t
an(f*x+e)^2),x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int (a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\int \left(a+b\tan\left(e+fx\right)\right)^{\frac{3}{2}}\sqrt{c+d\tan\left(e+fx\right)}\left(A+B\tan\left(e+fx\right)+C\tan^{2}\left(e+fx\right)\right)\,dx$$

```
[In] integrate((c+d*tan(f*x+e))**(1/2)*(a+b*tan(f*x+e))**(3/2)*(A+B*tan(f*x+e)+C
*tan(f*x+e)**2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**(3/2)*sqrt(c + d*tan(e + f*x))*(A + B*tan(e
+ f*x) + C*tan(e + f*x)**2), x)
```

Maxima [F]

$$\int (a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))\,dx=\int (C\tan(fx+e)^2+B\tan(fx+e)+A)(b\tan(fx+e)+a)^{\frac{3}{2}}\sqrt{d\tan(fx+e)+c}$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*t
an(f*x+e)^2),x, algorithm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)^(3/2
)*sqrt(d*tan(f*x + e) + c), x)
```

Giac [F(-1)]

Timed out.

$$\int (a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))\,dx=\text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*t
an(f*x+e)^2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int (a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))\,dx=\int (a+b\tan(e+f\,x))^{3/2}\,\sqrt{c+d\tan(e+f\,x)}\,(C\tan(e+f\,x)^2+B\tan(e+f\,x)+A)$$

```
[In] int((a + b*tan(e + f*x))^(3/2)*(c + d*tan(e + f*x))^(1/2)*(A + B*tan(e + f*
x) + C*tan(e + f*x)^2),x)
```

```
[Out] int((a + b*tan(e + f*x))^(3/2)*(c + d*tan(e + f*x))^(1/2)*(A + B*tan(e + f*
x) + C*tan(e + f*x)^2), x)
```

3.130 $\int \sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}(A+B\tan$

Optimal result 1876
Rubi [A] (verified) 1877
Mathematica [A] (verified) 1881
Maple [F(-1)] 1882
Fricas [B] (verification not implemented) 1882
Sympy [F] 1882
Maxima [F] 1883
Giac [F] 1883
Mupad [F(-1)] 1883

Optimal result

Integrand size = 49, antiderivative size = 381

$$\begin{aligned}
&\int \sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx\\
&=-\frac{\sqrt{a-ib}(iA+B-iC)\sqrt{c-id}\,\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f}\\
&\quad-\frac{\sqrt{a+ib}(B-i(A-C))\sqrt{c+id}\,\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f}\\
&\quad-\frac{(a^2Cd^2-2abd(cC+2Bd)+b^2(c^2C-4Bcd-8(A-C)d^2))\,\text{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{4b^{3/2}d^{3/2}f}\\
&\quad-\frac{(bcC-4bBd-aCd)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{4bdf}\\
&\quad+\frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{2df}
\end{aligned}$$

```
[Out] -1/4*(a^2*C*d^2-2*a*b*d*(2*B*d+C*c)+b^2*(c^2*C-4*B*c*d-8*(A-C)*d^2))*arctan
h(d^(1/2)*(a+b*tan(f*x+e))^(1/2)/b^(1/2)/(c+d*tan(f*x+e))^(1/2))/b^(3/2)/d^
(3/2)/f-(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(1
/2)/(c+d*tan(f*x+e))^(1/2))*(a-I*b)^(1/2)*(c-I*d)^(1/2)/f-(B-I*(A-C))*arcta
nh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2
))*(a+I*b)^(1/2)*(c+I*d)^(1/2)/f-1/4*(-4*B*b*d-C*a*d+C*b*c)*(a+b*tan(f*x+e)
)^(1/2)*(c+d*tan(f*x+e))^(1/2)/b/d/f+1/2*C*(a+b*tan(f*x+e))^(1/2)*(c+d*tan(
f*x+e))^(3/2)/d/f
```

Rubi [A] (verified)

Time = 5.53 (sec) , antiderivative size = 383, normalized size of antiderivative = 1.01, number of steps used = 14, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.163, Rules used = {3728, 3736, 6857, 65, 223, 212, 95, 214}

$$\int \sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = -\frac{(a^2Cd^2-2abd(2Bd+cC)+b^2(-8d^2(A-C)-4Bcd+c^2C))\operatorname{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{4b^{3/2}d^{3/2}f} - \frac{\sqrt{a-ib}\sqrt{c-id}(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f} + \frac{\sqrt{a+ib}\sqrt{c+id}(iA-B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f} - \frac{(-aCd-4bBd+bcC)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{4bdf} + \frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{2df}$$

```
[In] Int[Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] +
 C*Tan[e + f*x]^2),x]
```

```
[Out] -((Sqrt[a - I*b]*(I*A + B - I*C)*Sqrt[c - I*d]*ArcTanh[(Sqrt[c - I*d]*Sqrt[
a + b*Tan[e + f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/f) + (Sqrt[
a + I*b]*(I*A - B - I*C)*Sqrt[c + I*d]*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Ta
n[e + f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/f - ((a^2*C*d^2 - 2
*a*b*d*(c*C + 2*B*d) + b^2*(c^2*C - 4*B*c*d - 8*(A - C)*d^2))*ArcTanh[(Sqrt
[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])])/(4*b^(3/
2)*d^(3/2)*f) - ((b*c*C - 4*b*B*d - a*C*d)*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c
+ d*Tan[e + f*x]])/(4*b*d*f) + (C*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f
*x])^(3/2))/(2*d*f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 95

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
```

```
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 223

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3736

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[(a + b*ff*x)^m*(c + d*ff*x)^n*((A + B*ff*x + C*ff^2*x^2
)/(1 + ff^2*x^2)), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6857

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
```

```
[n, 0]
```

Rubi steps

integral

$$= \frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{2df} + \frac{\int \frac{\sqrt{c+d\tan(e+fx)}\left(\frac{1}{2}(-bcC+a(4A-3C)d)+2(Ab+aB-bC)d\tan(e+fx)-\frac{1}{2}(bcC-4bBd-aCd)\tan^2(e+fx)\right)}{\sqrt{a+b\tan(e+fx)}}\,dx}{2d}$$

$$= -\frac{(bcC-4bBd-aCd)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{4bdf} + \frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{2df} + \frac{\int \frac{\frac{1}{4}(-a^2Cd^2+2abd(4Ac-3cC-2Bd)-b^2c(cC+4Bd))+2bd(Abc+aBc-bcC+aAd-bBd-aCd)\tan(e+fx)+\frac{1}{4}(8b(Ab+aB-bC)d^2-(bc-ad)(}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}}{2bd}$$

$$= -\frac{(bcC-4bBd-aCd)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{4bdf} + \frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{2df} + \frac{\text{Subst}\left(\int \frac{\frac{1}{4}(-a^2Cd^2+2abd(4Ac-3cC-2Bd)-b^2c(cC+4Bd))+2bd(Abc+aBc-bcC+aAd-bBd-aCd)x+\frac{1}{4}(8b(Ab+aB-bC)d^2-(bc-ad)(}{\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}\right.}{2bdf}$$

$$= -\frac{(bcC-4bBd-aCd)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{4bdf} + \frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{2df} + \frac{\text{Subst}\left(\int \left(\frac{-a^2Cd^2+2abd(cC+2Bd)-b^2(c^2C-4Bcd-8(A-C)d^2)}{4\sqrt{a+bx}\sqrt{c+dx}} + \frac{2(-bd(bBc+b(A-C)d-a(Ac-cC-Bd))+bd(Abc+aBc-bcC+aAd}{\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}\right.\right.}{2bdf}$$

$$= -\frac{(bcC-4bBd-aCd)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{4bdf} + \frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{2df} + \frac{\text{Subst}\left(\int \frac{-bd(bBc+b(A-C)d-a(Ac-cC-Bd))+bd(Abc+aBc-bcC+aAd-bBd-aCd)x}{\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}\,dx, x, \tan(e+fx)\right)}{bdf} - \frac{(a^2Cd^2-2abd(cC+2Bd)+b^2(c^2C-4Bcd-8(A-C)d^2))\,\text{Subst}\left(\int \frac{1}{\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e+fx)\right)}{8bdf}$$

$$= -\frac{(bcC-4bBd-aCd)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{4bdf} + \frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{2df} + \frac{\text{Subst}\left(\int\left(\frac{-bd(Abc+aBc-bcC+aAd-bBd-aCd)-ibd(bBc+b(A-C)d-a(Ac-cC-Bd))}{2(i-x)\sqrt{a+bx}\sqrt{c+dx}}+\frac{bd(Abc+aBc-bcC+aAd-bBd-aCd)-ibd(b}{2(i+x)\sqrt{a+bx}\sqrt{c}}\right.\right.}{bdf} - \frac{(a^2Cd^2-2abd(cC+2Bd)+b^2(c^2C-4Bcd-8(A-C)d^2))\,\text{Subst}\left(\int\frac{1}{\sqrt{c-\frac{ad}{b}+\frac{dx^2}{b}}}\,dx,x,\sqrt{a+b\tan(e+\right.}{4b^2df}$$

$$= -\frac{(bcC-4bBd-aCd)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{4bdf} + \frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{2df} + \frac{((ia+b)(A-iB-C)(c-id))\text{Subst}\left(\int\frac{1}{(i+x)\sqrt{a+bx}\sqrt{c+dx}}\,dx,x,\tan(e+fx)\right)}{2f} - \frac{(a^2Cd^2-2abd(cC+2Bd)+b^2(c^2C-4Bcd-8(A-C)d^2))\,\text{Subst}\left(\int\frac{1}{1-\frac{dx^2}{b}}\,dx,x,\frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{4b^2df} - \frac{(bd(Abc+aBc-bcC+aAd-bBd-aCd)+ibd(bBc+b(A-C)d-a(Ac-cC-Bd)))\text{Subst}\left(\int\frac{}{(i-x}\right.}{2bdf}$$

$$= -\frac{(a^2Cd^2-2abd(cC+2Bd)+b^2(c^2C-4Bcd-8(A-C)d^2))\,\text{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{4b^{3/2}d^{3/2}f} - \frac{(bcC-4bBd-aCd)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{4bdf} + \frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{2df} + \frac{((ia+b)(A-iB-C)(c-id))\text{Subst}\left(\int\frac{1}{-a+ib-(-c+id)x^2}\,dx,x,\frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{f} - \frac{(bd(Abc+aBc-bcC+aAd-bBd-aCd)+ibd(bBc+b(A-C)d-a(Ac-cC-Bd)))\text{Subst}\left(\int\frac{}{a+ib}\right.}{bdf}$$

$$
\begin{aligned}
&= -\frac{\sqrt{a-ib}(iA+B-iC)\sqrt{c-id}\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f} \\
&+ \frac{\sqrt{a+ib}(iA-B-iC)\sqrt{c+id}\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f} \\
&- \frac{(a^2Cd^2-2abd(cC+2Bd)+b^2(c^2C-4Bcd-8(A-C)d^2))\operatorname{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{4b^{3/2}d^{3/2}f} \\
&- \frac{(bcC-4bBd-aCd)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{4bdf} \\
&+ \frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{2df}
\end{aligned}
$$

Mathematica [A] (verified)

Time = 7.94 (sec) , antiderivative size = 619, normalized size of antiderivative = 1.62

$$
\begin{aligned}
&\int \sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx \\
&= \frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{2df} \\
&+ \frac{\frac{(-bcC+4bBd+aCd)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{2bf} + \frac{\frac{2bd\left(b(Abc+aBc-bcC+aAd-bBd-aCd)-\sqrt{-b^2}(bBc+b(A-C)d-a(Ac-cC-Bd))\right)\operatorname{arct}}{\cdots}}{\sqrt{-a+\sqrt{-b^2}}\sqrt{-c+\frac{\sqrt{-b^2}d}{b}}}}{\cdots}
\end{aligned}
$$

```
[In] Integrate[Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e +
f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] (C*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(3/2))/(2*d*f) + (((-(b*c*
C) + 4*b*B*d + a*C*d)*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(2
*b*f) + ((2*b*d*(b*(A*b*c + a*B*c - b*c*C + a*A*d - b*B*d - a*C*d) - Sqrt[-
b^2]*(b*B*c + b*(A - C)*d - a*(A*c - c*C - B*d)))*ArcTanh[(Sqrt[-c + (Sqrt[
-b^2]*d)/b]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[c + d*Tan
[e + f*x]])])/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[-c + (Sqrt[-b^2]*d)/b]) - (2*b*d*
(b*(A*b*c + a*B*c - b*c*C + a*A*d - b*B*d - a*C*d) + Sqrt[-b^2]*(b*B*c + b*
(A - C)*d - a*(A*c - c*C - B*d)))*ArcTanh[(Sqrt[c + (Sqrt[-b^2]*d)/b]*Sqrt[
a + b*Tan[e + f*x]])/(Sqrt[a + Sqrt[-b^2]]*Sqrt[c + d*Tan[e + f*x]])])/(Sqr
t[a + Sqrt[-b^2]]*Sqrt[c + (Sqrt[-b^2]*d)/b]) - (Sqrt[b]*Sqrt[c - (a*d)/b]*
(a^2*C*d^2 - 2*a*b*d*(c*C + 2*B*d) + b^2*(c^2*C - 4*B*c*d - 8*(A - C)*d^2))
*ArcSinh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c - (a*d)/b])]*Sq
rt[(b*c + b*d*Tan[e + f*x])/(b*c - a*d)])/(2*Sqrt[d]*Sqrt[c + d*Tan[e + f*x
]]))/(b^2*f))/(2*d)
```

Maple [F(-1)]

Timed out.

$$\int \sqrt{a+b\tan\left(fx+e\right)}\,\sqrt{c+d\tan\left(fx+e\right)}\,\left(A+B\tan\left(fx+e\right)+C\tan\left(fx+e\right)^{2}\right)dx$$

```
[In] int((a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x)
```

```
[Out] int((a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 34031 vs. $2(308) = 616$.

Time = 161.87 (sec) , antiderivative size = 68078, normalized size of antiderivative = 178.68

$$\int \sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*t
an(f*x+e)^2),x, algorithm="fricas")
```

```
[Out] Too large to include
```

Sympy [F]

$$\int \sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\int \sqrt{a+b\tan\left(e+fx\right)}\sqrt{c+d\tan\left(e+fx\right)}\left(A+B\tan\left(e+fx\right)+C\tan^{2}\left(e+fx\right)\right)\,dx$$

```
[In] integrate((a+b*tan(f*x+e))**(1/2)*(c+d*tan(f*x+e))**(1/2)*(A+B*tan(f*x+e)+C
*tan(f*x+e)**2),x)
```

```
[Out] Integral(sqrt(a + b*tan(e + f*x))*sqrt(c + d*tan(e + f*x))*(A + B*tan(e + f
*x) + C*tan(e + f*x)**2), x)
```

Maxima [F]

$$\int \sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\int \left(C\tan\left(fx+e\right)^2+B\tan\left(fx+e\right)+A\right)\sqrt{b\tan\left(fx+e\right)+a}\sqrt{d\tan\left(fx+e\right)+c}\,dx$$

```
[In] integrate((a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*t
an(f*x+e)^2),x, algorithm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*sqrt(b*tan(f*x + e) + a)*
sqrt(d*tan(f*x + e) + c), x)
```

Giac [F]

$$\int \sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\int \left(C\tan\left(fx+e\right)^2+B\tan\left(fx+e\right)+A\right)\sqrt{b\tan\left(fx+e\right)+a}\sqrt{d\tan\left(fx+e\right)+c}\,dx$$

```
[In] integrate((a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*t
an(f*x+e)^2),x, algorithm="giac")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*sqrt(b*tan(f*x + e) + a)*
sqrt(d*tan(f*x + e) + c), x)
```

Mupad [F(-1)]

Timed out.

$$\int \sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\text{Hanged}$$

```
[In] int((a + b*tan(e + f*x))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(A + B*tan(e + f*
x) + C*tan(e + f*x)^2),x)
```

```
[Out] \text{Hanged}
```

3.131 $\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{a+b\tan(e+fx)}}\,dx$

Optimal result 1884
Rubi [A] (verified) 1885
Mathematica [A] (verified) 1888
Maple [F(-1)] 1889
Fricas [B] (verification not implemented) 1889
Sympy [F] 1889
Maxima [F] 1890
Giac [F(-1)] 1890
Mupad [F(-1)] 1890

Optimal result

Integrand size = 49, antiderivative size = 287

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{a+b\tan(e+fx)}}\,dx = -\frac{(iA+B-iC)\sqrt{c-id}\,\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{a-ib}f} - \frac{(B-i(A-C))\sqrt{c+id}\,\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{a+ib}f} + \frac{(bcC+2bBd-aCd)\,\text{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{b^{3/2}\sqrt{d}f} + \frac{C\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{bf}$$

```
[Out] -(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(1/2)/(c+
d*tan(f*x+e))^(1/2))*(c-I*d)^(1/2)/f/(a-I*b)^(1/2)-(B-I*(A-C))*arctanh((c+I
*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))*(c+I
*d)^(1/2)/f/(a+I*b)^(1/2)+(2*B*b*d-C*a*d+C*b*c)*arctanh(d^(1/2)*(a+b*tan(f*
x+e))^(1/2)/b^(1/2)/(c+d*tan(f*x+e))^(1/2))/b^(3/2)/f/d^(1/2)+C*(a+b*tan(f*
x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)/b/f
```

Rubi [A] (verified)

Time = 3.15 (sec) , antiderivative size = 287, normalized size of antiderivative = 1.00, number of steps used = 13, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.163, Rules used = {3728, 3736, 6857, 65, 223, 212, 95, 214}

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{a+b\tan(e+fx)}}\,dx$$
$$= -\frac{\sqrt{c-id}(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f\sqrt{a-ib}}$$
$$- \frac{\sqrt{c+id}(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f\sqrt{a+ib}}$$
$$+ \frac{(-aCd+2bBd+bcC)\text{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{b^{3/2}\sqrt{d}f}$$
$$+ \frac{C\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{bf}$$

```
[In] Int[(Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/Sqrt
[a + b*Tan[e + f*x]],x]
```

```
[Out] -(((I*A + B - I*C)*Sqrt[c - I*d]*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[a - I*b]*f)) - ((B
- I*(A - C))*Sqrt[c + I*d]*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]])
/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[a + I*b]*f) + ((b*c*C + 2
*b*B*d - a*C*d)*ArcTanh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c
+ d*Tan[e + f*x]])])/(b^(3/2)*Sqrt[d]*f) + (C*Sqrt[a + b*Tan[e + f*x]]*Sqrt
[c + d*Tan[e + f*x]])/(b*f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 95

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 223

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3736

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[(a + b*ff*x)^m*(c + d*ff*x)^n*((A + B*ff*x + C*ff^2*x^2
)/(1 + ff^2*x^2)), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6857

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\text{integral} = \frac{C\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{bf} + \frac{\int \frac{\frac{1}{2}(2Abc-C(bc+ad))+b(Bc+(A-C)d)\tan(e+fx)+\frac{1}{2}(bcC+2bBd-aCd)\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx}{b}$$

$$= \frac{C\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{bf} + \frac{\text{Subst}\left(\int \frac{\frac{1}{2}(2Abc-C(bc+ad))+b(Bc+(A-C)d)x+\frac{1}{2}(bcC+2bBd-aCd)x^2}{\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}\,dx, x, \tan(e+fx)\right)}{bf}$$

$$= \frac{C\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{bf} + \frac{\text{Subst}\left(\int \left(\frac{bcC+2bBd-aCd}{2\sqrt{a+bx}\sqrt{c+dx}} + \frac{b(Ac-cC-Bd)+b(Bc+(A-C)d)x}{\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}\right)\,dx, x, \tan(e+fx)\right)}{bf}$$

$$= \frac{C\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{bf} + \frac{\text{Subst}\left(\int \frac{b(Ac-cC-Bd)+b(Bc+(A-C)d)x}{\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}\,dx, x, \tan(e+fx)\right)}{bf} + \frac{(bcC+2bBd-aCd)\text{Subst}\left(\int \frac{1}{\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e+fx)\right)}{2bf}$$

$$= \frac{C\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{bf} + \frac{\text{Subst}\left(\int \left(\frac{ib(Ac-cC-Bd)-b(Bc+(A-C)d)}{2(i-x)\sqrt{a+bx}\sqrt{c+dx}} + \frac{ib(Ac-cC-Bd)+b(Bc+(A-C)d)}{2(i+x)\sqrt{a+bx}\sqrt{c+dx}}\right)\,dx, x, \tan(e+fx)\right)}{bf} + \frac{(bcC+2bBd-aCd)\text{Subst}\left(\int \frac{1}{\sqrt{c-\frac{ad}{b}+\frac{dx^2}{b}}}\,dx, x, \sqrt{a+b\tan(e+fx)}\right)}{b^2f}$$

$$= \frac{C\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{bf} + \frac{((A-iB-C)(ic+d))\text{Subst}\left(\int \frac{1}{(i+x)\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e+fx)\right)}{2f} + \frac{(bcC+2bBd-aCd)\text{Subst}\left(\int \frac{1}{1-\frac{dx^2}{b}}\,dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{b^2f} + \frac{(ib(Ac-cC-Bd)-b(Bc+(A-C)d))\text{Subst}\left(\int \frac{1}{(i-x)\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e+fx)\right)}{2bf}$$

$$= \frac{(bcC+2bBd-aCd)\text{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{b^{3/2}\sqrt{d}f} + \frac{C\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{bf} + \frac{((A-iB-C)(ic+d))\text{Subst}\left(\int \frac{1}{-a+ib-(-c+id)x^2}\,dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{f} + \frac{(ib(Ac-cC-Bd)-b(Bc+(A-C)d))\text{Subst}\left(\int \frac{1}{a+ib-(c+id)x^2}\,dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{bf}$$

$$= -\frac{(iA+B-iC)\sqrt{c-id}\,\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{a-ib}f} - \frac{(B-i(A-C))\sqrt{c+id}\,\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{a+ib}f} + \frac{(bcC+2bBd-aCd)\text{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{b^{3/2}\sqrt{d}f} + \frac{C\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{bf}$$

Mathematica [A] (verified)

Time = 4.70 (sec) , antiderivative size = 441, normalized size of antiderivative = 1.54

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{a+b\tan(e+fx)}}\,dx$$

$$= \frac{\frac{b\left(bBc+b(A-C)d+\sqrt{-b^2}(Ac-cC-Bd)\right)\text{arctanh}\left(\frac{\sqrt{-c+\frac{\sqrt{-b^2}d}{b}}\sqrt{a+b\tan(e+fx)}}{\sqrt{-a+\sqrt{-b^2}}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{-a+\sqrt{-b^2}}\sqrt{-c+\frac{\sqrt{-b^2}d}{b}}} + \frac{b\left(\sqrt{-b^2}(Ac-cC-Bd)-b(Bc+(A-C)d)\right)\text{arctanh}\left(\sqrt{\ldots}\right.}{\sqrt{a+\sqrt{-b^2}}\sqrt{c+\frac{\sqrt{-b^2}d}{b}}}}{}$$

```
[In] Integrate[(Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)
)/Sqrt[a + b*Tan[e + f*x]],x]
```

```
[Out] ((b*(b*B*c + b*(A - C)*d + Sqrt[-b^2]*(A*c - c*C - B*d))*ArcTanh[(Sqrt[-c +
 (Sqrt[-b^2]*d)/b]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[c
+ d*Tan[e + f*x]])])/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[-c + (Sqrt[-b^2]*d)/b]) +
(b*(Sqrt[-b^2]*(A*c - c*C - B*d) - b*(B*c + (A - C)*d))*ArcTanh[(Sqrt[c + (
Sqrt[-b^2]*d)/b]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + Sqrt[-b^2]]*Sqrt[c + d
*Tan[e + f*x]])])/(Sqrt[a + Sqrt[-b^2]]*Sqrt[c + (Sqrt[-b^2]*d)/b]) + b*C*S
qrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]] + (Sqrt[b]*Sqrt[c - (a*d)/
b]*(b*c*C + 2*b*B*d - a*C*d)*ArcSinh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sq
```

```
rt[b]*Sqrt[c - (a*d)/b])]*Sqrt[(b*(c + d*Tan[e + f*x]))/(b*c - a*d)])/(Sqrt
[d]*Sqrt[c + d*Tan[e + f*x]]))/(b^2*f)
```

Maple [F(-1)]

Timed out.

$$\int \frac{\sqrt{c+d\tan(fx+e)}\,(A+B\tan(fx+e)+C\tan(fx+e)^2)}{\sqrt{a+b\tan(fx+e)}}dx$$

```
[In] int((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(1/2),x)
```

```
[Out] int((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(1/2),x)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 39018 vs. $2(225) = 450$.

Time = 105.64 (sec) , antiderivative size = 78051, normalized size of antiderivative = 271.95

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{a+b\tan(e+fx)}}\,dx = \text{Too large to display}$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(1/2),x, algorithm="fricas")
```

```
[Out] Too large to include
```

Sympy [F]

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{a+b\tan(e+fx)}}\,dx = \int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{a+b\tan(e+fx)}}\,dx$$

```
[In] integrate((c+d*tan(f*x+e))**(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e))**(1/2),x)
```

```
[Out] Integral(sqrt(c + d*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/
sqrt(a + b*tan(e + f*x)), x)
```

Maxima [F]

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{a+b\tan(e+fx)}}\,dx$$
$$=\int \frac{(C\tan(fx+e)^2+B\tan(fx+e)+A)\sqrt{d\tan(fx+e)+c}}{\sqrt{b\tan(fx+e)+a}}\,dx$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(1/2),x, algorithm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*sqrt(d*tan(f*x + e) + c)/
sqrt(b*tan(f*x + e) + a), x)
```

Giac [F(-1)]

Timed out.

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{a+b\tan(e+fx)}}\,dx=\text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(1/2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{a+b\tan(e+fx)}}\,dx=\text{Hanged}$$

```
[In] int(((c + d*tan(e + f*x))^(1/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x))^(1/2),x)
```

```
[Out] \text{Hanged}
```

3.132 $\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{3/2}}\,dx$

Optimal result . . . 1891
Rubi [A] (verified) . . . 1892
Mathematica [A] (verified) . . . 1895
Maple [F(-1)] . . . 1896
Fricas [B] (verification not implemented) . . . 1896
Sympy [F] . . . 1896
Maxima [F(-1)] . . . 1897
Giac [F(-1)] . . . 1897
Mupad [F(-1)] . . . 1897

Optimal result

Integrand size = 49, antiderivative size = 300

$$\begin{aligned}
\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{3/2}}\,dx = &-\frac{(iA+B-iC)\sqrt{c-id}\,\mathrm{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a-ib)^{3/2}f}\\
&-\frac{(B-i(A-C))\sqrt{c+id}\,\mathrm{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a+ib)^{3/2}f}\\
&+\frac{2C\sqrt{d}\,\mathrm{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{b^{3/2}f}-\frac{2(Ab^2-a(bB-aC))\sqrt{c+d\tan(e+fx)}}{b(a^2+b^2)f\sqrt{a+b\tan(e+fx)}}
\end{aligned}$$

```
[Out] -(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(1/2)/(c+
d*tan(f*x+e))^(1/2))*(c-I*d)^(1/2)/(a-I*b)^(3/2)/f-(B-I*(A-C))*arctanh((c+I
*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))*(c+I
*d)^(1/2)/(a+I*b)^(3/2)/f+2*C*arctanh(d^(1/2)*(a+b*tan(f*x+e))^(1/2)/b^(1/2
)/(c+d*tan(f*x+e))^(1/2))*d^(1/2)/b^(3/2)/f-2*(A*b^2-a*(B*b-C*a))*(c+d*tan(
f*x+e))^(1/2)/b/(a^2+b^2)/f/(a+b*tan(f*x+e))^(1/2)
```

Rubi [A] (verified)

Time = 4.50 (sec) , antiderivative size = 300, normalized size of antiderivative = 1.00, number of steps used = 13, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}} = 0.163$, Rules used = {3726, 3736, 6857, 65, 223, 212, 95, 214}

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{3/2}}\,dx = -\frac{2(Ab^2-a(bB-aC))\sqrt{c+d\tan(e+fx)}}{bf(a^2+b^2)\sqrt{a+b\tan(e+fx)}} - \frac{\sqrt{c-id}(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(a-ib)^{3/2}} - \frac{\sqrt{c+id}(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(a+ib)^{3/2}} + \frac{2C\sqrt{d}\text{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{b^{3/2}f}$$

```
[In] Int[(Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a +
 b*Tan[e + f*x])^(3/2),x]
```

```
[Out] -(((I*A + B - I*C)*Sqrt[c - I*d]*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a - I*b)^(3/2)*f)) - ((
B - I*(A - C))*Sqrt[c + I*d]*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]
])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a + I*b)^(3/2)*f) + (2*C*Sq
rt[d]*ArcTanh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c + d*Tan[e
+ f*x]])])/(b^(3/2)*f) - (2*(A*b^2 - a*(b*B - a*C))*Sqrt[c + d*Tan[e + f*x]
])/(b*(a^2 + b^2)*f*Sqrt[a + b*Tan[e + f*x]])
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 95

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 223

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3726

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*d^2 + c*(c*C - B*d))*(a + b*Tan[e +
 f*x])^m*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3736

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[(a + b*ff*x)^m*(c + d*ff*x)^n*((A + B*ff*x + C*ff^2*x^2
)/(1 + ff^2*x^2)), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6857

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

integral

$$= -\frac{2(Ab^2 - a(bB - aC))\sqrt{c + d\tan(e + fx)}}{b(a^2 + b^2) f\sqrt{a + b\tan(e + fx)}} + \frac{2\int \frac{\frac{1}{2}((bB-aC)(bc-ad)+Ab(ac+bd))-\frac{1}{2}b((A-C)(bc-ad)-B(ac+bd))\tan(e+fx)+\frac{1}{2}(a^2+b^2)Cd\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx}{b(a^2 + b^2)}$$

$$= -\frac{2(Ab^2 - a(bB - aC))\sqrt{c + d\tan(e + fx)}}{b(a^2 + b^2) f\sqrt{a + b\tan(e + fx)}} + \frac{2\text{Subst}\left(\int \frac{\frac{1}{2}((bB-aC)(bc-ad)+Ab(ac+bd))-\frac{1}{2}b((A-C)(bc-ad)-B(ac+bd))x+\frac{1}{2}(a^2+b^2)Cdx^2}{\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}\,dx, x, \tan(e + fx)\right)}{b(a^2 + b^2) f}$$

$$= -\frac{2(Ab^2 - a(bB - aC))\sqrt{c + d\tan(e + fx)}}{b(a^2 + b^2) f\sqrt{a + b\tan(e + fx)}} + \frac{2\text{Subst}\left(\int \left(\frac{(a^2+b^2)Cd}{2\sqrt{a+bx}\sqrt{c+dx}} + \frac{b(bBc+b(A-C)d+a(Ac-cC-Bd))-b(Abc-aBc-bcC-aAd-bBd+aCd)x}{2\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}\right)dx, x, \tan(e + fx)\right)}{b(a^2 + b^2) f}$$

$$= -\frac{2(Ab^2 - a(bB - aC))\sqrt{c + d\tan(e + fx)}}{b(a^2 + b^2) f\sqrt{a + b\tan(e + fx)}} + \frac{\text{Subst}\left(\int \frac{b(bBc+b(A-C)d+a(Ac-cC-Bd))-b(Abc-aBc-bcC-aAd-bBd+aCd)x}{\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}\,dx, x, \tan(e + fx)\right)}{b(a^2 + b^2) f} + \frac{(Cd)\text{Subst}\left(\int \frac{1}{\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e + fx)\right)}{bf}$$

$$= -\frac{2(Ab^2 - a(bB - aC))\sqrt{c + d\tan(e + fx)}}{b(a^2 + b^2) f\sqrt{a + b\tan(e + fx)}} + \frac{\text{Subst}\left(\int \left(\frac{b(Abc-aBc-bcC-aAd-bBd+aCd)+ib(bBc+b(A-C)d+a(Ac-cC-Bd))}{2(i-x)\sqrt{a+bx}\sqrt{c+dx}} + \frac{-b(Abc-aBc-bcC-aAd-bBd+aCd)+ib(bBc+b}{2(i+x)\sqrt{a+bx}\sqrt{c+dx}}\right.\right.}{b(a^2 + b^2) f} + \frac{(2Cd)\text{Subst}\left(\int \frac{1}{\sqrt{c-\frac{ad}{b}+\frac{dx^2}{b}}}\,dx, x, \sqrt{a + b\tan(e + fx)}\right)}{b^2 f}$$

$$= -\frac{2(Ab^2 - a(bB - aC))\sqrt{c + d\tan(e + fx)}}{b(a^2 + b^2)f\sqrt{a + b\tan(e + fx)}} + \frac{((ia + b)(A + iB - C)(c + id))\text{Subst}\left(\int \frac{1}{(i-x)\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e + fx)\right)}{2(a^2 + b^2)f} + \frac{(2Cd)\text{Subst}\left(\int \frac{1}{1-\frac{dx^2}{b}}\,dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{b^2f} + \frac{((A - iB - C)(ic + d))\text{Subst}\left(\int \frac{1}{(i+x)\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e + fx)\right)}{2(a - ib)f}$$

$$= \frac{2C\sqrt{d}\text{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{b^{3/2}f} - \frac{2(Ab^2 - a(bB - aC))\sqrt{c + d\tan(e + fx)}}{b(a^2 + b^2)f\sqrt{a + b\tan(e + fx)}} + \frac{((ia + b)(A + iB - C)(c + id))\text{Subst}\left(\int \frac{1}{a+ib-(c+id)x^2}\,dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{(a^2 + b^2)f} + \frac{((A - iB - C)(ic + d))\text{Subst}\left(\int \frac{1}{-a+ib-(-c+id)x^2}\,dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{(a - ib)f}$$

$$= -\frac{(iA + B - iC)\sqrt{c - id}\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a - ib)^{3/2}f} - \frac{(B - i(A - C))\sqrt{c + id}\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a + ib)^{3/2}f} + \frac{2C\sqrt{d}\text{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{b^{3/2}f} - \frac{2(Ab^2 - a(bB - aC))\sqrt{c + d\tan(e + fx)}}{b(a^2 + b^2)f\sqrt{a + b\tan(e + fx)}}$$

Mathematica [A] (verified)

Time = 5.86 (sec) , antiderivative size = 412, normalized size of antiderivative = 1.37

$$\int \frac{\sqrt{c + d\tan(e + fx)}(A + B\tan(e + fx) + C\tan^2(e + fx))}{(a + b\tan(e + fx))^{3/2}}\,dx = \frac{\frac{(iA+B-iC)\sqrt{-c+id}\text{arctanh}\left(\frac{\sqrt{-c+id}\sqrt{a+b\tan(e+}}{\sqrt{-a+ib}\sqrt{c+d\tan(e+}}\right.}{(-a+ib)^{3/2}}}{}$$

```
[In] Integrate[(Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)
)/(a + b*Tan[e + f*x])^(3/2),x]
```

```
[Out] (((I*A + B - I*C)*Sqrt[-c + I*d]*ArcTanh[(Sqrt[-c + I*d]*Sqrt[a + b*Tan[e +
 f*x]])/(Sqrt[-a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(-a + I*b)^(3/2) + (I*(
A + I*B - C)*Sqrt[c + I*d]*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]])
/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(a + I*b)^(3/2) + ((B + I*(A -
```

```
C))*Sqrt[c + d*Tan[e + f*x]])/((a - I*b)*Sqrt[a + b*Tan[e + f*x]]) + (((-I)
*A + B + I*C)*Sqrt[c + d*Tan[e + f*x]])/((a + I*b)*Sqrt[a + b*Tan[e + f*x]]
) + (2*C*(-((b*(c + d*Tan[e + f*x]))/Sqrt[a + b*Tan[e + f*x]]) + Sqrt[d]*Sq
rt[b*c - a*d]*ArcSinh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/Sqrt[b*c - a*d]]*S
qrt[(b*(c + d*Tan[e + f*x]))/(b*c - a*d)]))/(b^2*Sqrt[c + d*Tan[e + f*x]]))
/f
```

Maple [F(-1)]

Timed out.

$$\int \frac{\sqrt{c+d\tan\left(fx+e\right)}\,\left(A+B\tan\left(fx+e\right)+C\tan\left(fx+e\right)^{2}\right)}{\left(a+b\tan\left(fx+e\right)\right)^{\frac{3}{2}}}dx$$

```
[In] int((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(3/2),x)
```

```
[Out] int((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(3/2),x)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 69754 vs. $2(237) = 474$.

Time = 198.89 (sec) , antiderivative size = 139535, normalized size of antiderivative = 465.12

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{3/2}}\,dx = \text{Too large to display}$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(3/2),x, algorithm="fricas")
```

```
[Out] Too large to include
```

Sympy [F]

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{3/2}}\,dx = \int \frac{\sqrt{c+d\tan\left(e+fx\right)}(A+B\tan\left(e+f\right.}{\left(a+b\tan\left(e+fx\right.\right.}$$

```
[In] integrate((c+d*tan(f*x+e))**(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e))**(3/2),x)
```

```
[Out] Integral(sqrt(c + d*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/
(a + b*tan(e + f*x))**(3/2), x)
```

Maxima [F(-1)]

Timed out.

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(3/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

Giac [F(-1)]

Timed out.

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(3/2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{3/2}}\,dx = \text{Hanged}$$

```
[In] int(((c + d*tan(e + f*x))^(1/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x))^(3/2),x)
```

```
[Out] \text{Hanged}
```

3.133 $\int \frac{\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{5/2}}\,dx$

Optimal result 1898
Rubi [A] (verified) 1899
Mathematica [A] (verified) 1902
Maple [F(-1)] 1903
Fricas [F(-1)] 1903
Sympy [F] 1903
Maxima [F(-2)] 1904
Giac [F(-1)] 1904
Mupad [F(-1)] 1904

Optimal result

Integrand size = 49, antiderivative size = 370

$$\int \frac{\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{5/2}}\,dx = -\frac{(iA+B-iC)\sqrt{c-id}\,\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a-ib)^{5/2}f} - \frac{(B-i(A-C))\sqrt{c+id}\,\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a+ib)^{5/2}f} - \frac{2(Ab^2-a(bB-aC))\sqrt{c+d\tan(e+fx)}}{3b\,(a^2+b^2)\,f(a+b\tan(e+fx))^{3/2}} - \frac{2(2a^3bBd+a^4Cd+b^4(3Bc+Ad)+2ab^3(3Ac-3cC-2Bd)-a^2b^2(3Bc+5Ad-7Cd))\sqrt{c+d\tan(e+}}{3b\,(a^2+b^2)^2\,(bc-ad)f\sqrt{a+b\tan(e+fx)}}$$

```
[Out] -(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(1/2)/(c+
d*tan(f*x+e))^(1/2))*(c-I*d)^(1/2)/(a-I*b)^(5/2)/f-(B-I*(A-C))*arctanh((c+I
*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))*(c+I
*d)^(1/2)/(a+I*b)^(5/2)/f-2/3*(2*a^3*b*B*d+a^4*C*d+b^4*(A*d+3*B*c)+2*a*b^3*
(3*A*c-2*B*d-3*C*c)-a^2*b^2*(5*A*d+3*B*c-7*C*d))*(c+d*tan(f*x+e))^(1/2)/b/(
a^2+b^2)^2/(-a*d+b*c)/f/(a+b*tan(f*x+e))^(1/2)-2/3*(A*b^2-a*(B*b-C*a))*(c+d
*tan(f*x+e))^(1/2)/b/(a^2+b^2)/f/(a+b*tan(f*x+e))^(3/2)
```

Rubi [A] (verified)

Time = 2.26 (sec) , antiderivative size = 370, normalized size of antiderivative = 1.00, number of steps used = 9, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.122, Rules used = {3726, 3730, 3697, 3696, 95, 214}

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{5/2}}\,dx =$$
$$-\frac{2(Ab^2-a(bB-aC))\sqrt{c+d\tan(e+fx)}}{3bf\,(a^2+b^2)\,(a+b\tan(e+fx))^{3/2}}$$
$$-\frac{2\sqrt{c+d\tan(e+fx)}(a^4Cd+2a^3bBd-a^2b^2(5Ad+3Bc-7Cd)+2ab^3(3Ac-2Bd-3cC)+b^4(Ad+3}{3bf\,(a^2+b^2)^2\,(bc-ad)\sqrt{a+b\tan(e+fx)}}$$
$$-\frac{\sqrt{c-id}(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(a-ib)^{5/2}}$$
$$-\frac{\sqrt{c+id}(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(a+ib)^{5/2}}$$

```
[In] Int[(Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a +
 b*Tan[e + f*x])^(5/2),x]
```

```
[Out] -(((I*A + B - I*C)*Sqrt[c - I*d]*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])]])/((a - I*b)^(5/2)*f)) - ((
B - I*(A - C))*Sqrt[c + I*d]*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]
])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])]])/((a + I*b)^(5/2)*f) - (2*(A*b
^2 - a*(b*B - a*C))*Sqrt[c + d*Tan[e + f*x]])/(3*b*(a^2 + b^2)*f*(a + b*Tan
[e + f*x])^(3/2)) - (2*(2*a^3*b*B*d + a^4*C*d + b^4*(3*B*c + A*d) + 2*a*b^3
*(3*A*c - 3*c*C - 2*B*d) - a^2*b^2*(3*B*c + 5*A*d - 7*C*d))*Sqrt[c + d*Tan[
e + f*x]])/(3*b*(a^2 + b^2)^2*(b*c - a*d)*f*Sqrt[a + b*Tan[e + f*x]])
```

Rule 95

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3696

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[A^2/f, Subst[Int[(a + b*x)^m*((c + d*x)^n/(A - B*x)), x], x, Tan[e + f*x
]], x] /; FreeQ[{a, b, c, d, e, f, A, B, m, n}, x] && NeQ[b*c - a*d, 0] &&
NeQ[a^2 + b^2, 0] && EqQ[A^2 + B^2, 0]
```

Rule 3697

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[(A + I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^n*(1 - I*Ta
n[e + f*x]), x], x] + Dist[(A - I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*T
an[e + f*x])^n*(1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A,
B, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[A^2 + B^2, 0]
```

Rule 3726

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*d^2 + c*(c*C - B*d))*(a + b*Tan[e +
 f*x])^m*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3730

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 - a*(b*B - a*C))*(a + b*Tan[e +
 f*x])^(m + 1)*((c + d*Tan[e + f*x])^(n + 1)/(f*(m + 1)*(b*c - a*d)*(a^2 +
b^2))), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rubi steps

$$\text{integral} = -\frac{2(Ab^2-a(bB-aC))\sqrt{c+d\tan(e+fx)}}{3b(a^2+b^2)f(a+b\tan(e+fx))^{3/2}} + \frac{2\int \frac{\frac{1}{2}((bB-aC)(3bc-ad)+Ab(3ac+bd))-\frac{3}{2}b((A-C)(bc-ad)-B(ac+bd))\tan(e+fx)-\frac{1}{2}(2Ab^2-2abB-a^2C-3b^2C)d\tan^2(e+fx)}{(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}\,dx}{3b(a^2+b^2)}$$

$$= -\frac{2(Ab^2-a(bB-aC))\sqrt{c+d\tan(e+fx)}}{3b(a^2+b^2)f(a+b\tan(e+fx))^{3/2}} - \frac{2(2a^3bBd+a^4Cd+b^4(3Bc+Ad)+2ab^3(3Ac-3cC-2Bd)-a^2b^2(3Bc+5Ad-7Cd))\sqrt{c}}{3b(a^2+b^2)^2(bc-ad)f\sqrt{a+b\tan(e+fx)}} - \frac{4\int \frac{-\frac{3}{4}b(bc-ad)(a^2(Ac-cC-Bd)-b^2(Ac-cC-Bd)+2ab(Bc+(A-C)d))+\frac{3}{4}b(bc-ad)(2ab(Ac-cC-Bd)-a^2(Bc+(A-C)d)+b^2(}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}}{3b(a^2+b^2)^2(bc-ad)}$$

$$= -\frac{2(Ab^2-a(bB-aC))\sqrt{c+d\tan(e+fx)}}{3b(a^2+b^2)f(a+b\tan(e+fx))^{3/2}} - \frac{2(2a^3bBd+a^4Cd+b^4(3Bc+Ad)+2ab^3(3Ac-3cC-2Bd)-a^2b^2(3Bc+5Ad-7Cd))\sqrt{c}}{3b(a^2+b^2)^2(bc-ad)f\sqrt{a+b\tan(e+fx)}} + \frac{((A-iB-C)(c-id))\int \frac{1+i\tan(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx}{2(a-ib)^2} + \frac{((A+iB-C)(c+id))\int \frac{1-i\tan(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx}{2(a+ib)^2}$$

$$= -\frac{2(Ab^2-a(bB-aC))\sqrt{c+d\tan(e+fx)}}{3b(a^2+b^2)f(a+b\tan(e+fx))^{3/2}} - \frac{2(2a^3bBd+a^4Cd+b^4(3Bc+Ad)+2ab^3(3Ac-3cC-2Bd)-a^2b^2(3Bc+5Ad-7Cd))\sqrt{c}}{3b(a^2+b^2)^2(bc-ad)f\sqrt{a+b\tan(e+fx)}} + \frac{((A-iB-C)(c-id))\text{Subst}\left(\int \frac{1}{(1-ix)\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e+fx)\right)}{2(a-ib)^2f} + \frac{((A+iB-C)(c+id))\text{Subst}\left(\int \frac{1}{(1+ix)\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e+fx)\right)}{2(a+ib)^2f}$$

$$= -\frac{2(Ab^2 - a(bB - aC))\sqrt{c + d\tan(e + fx)}}{3b(a^2 + b^2) f(a + b\tan(e + fx))^{3/2}}$$

$$-\frac{2(2a^3bBd + a^4Cd + b^4(3Bc + Ad) + 2ab^3(3Ac - 3cC - 2Bd) - a^2b^2(3Bc + 5Ad - 7Cd))\sqrt{c+\dots}}{3b(a^2 + b^2)^2(bc - ad)f\sqrt{a + b\tan(e + fx)}}$$

$$+\frac{((A - iB - C)(c - id))\text{Subst}\left(\int \frac{1}{ia+b-(ic+d)x^2}\,dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{(a - ib)^2 f}$$

$$+\frac{((A + iB - C)(c + id))\text{Subst}\left(\int \frac{1}{-ia+b-(-ic+d)x^2}\,dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{(a + ib)^2 f}$$

$$= -\frac{(iA + B - iC)\sqrt{c - id}\,\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a - ib)^{5/2} f}$$

$$-\frac{(B - i(A - C))\sqrt{c + id}\,\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a + ib)^{5/2} f}$$

$$-\frac{2(Ab^2 - a(bB - aC))\sqrt{c + d\tan(e + fx)}}{3b(a^2 + b^2) f(a + b\tan(e + fx))^{3/2}}$$

$$-\frac{2(2a^3bBd + a^4Cd + b^4(3Bc + Ad) + 2ab^3(3Ac - 3cC - 2Bd) - a^2b^2(3Bc + 5Ad - 7Cd))\sqrt{c+\dots}}{3b(a^2 + b^2)^2(bc - ad)f\sqrt{a + b\tan(e + fx)}}$$

Mathematica [A] (verified)

Time = 7.10 (sec) , antiderivative size = 600, normalized size of antiderivative = 1.62

$$\int \frac{\sqrt{c + d\tan(e + fx)}(A + B\tan(e + fx) + C\tan^2(e + fx))}{(a + b\tan(e + fx))^{5/2}}\,dx =$$

$$-\frac{C\sqrt{c + d\tan(e + fx)}}{bf(a + b\tan(e + fx))^{3/2}}$$

$$-\frac{-\frac{2\left(\frac{1}{2}b^2(-2Abc+3bcC-aCd)-a\left(-b^2(Bc+(A-C)d)-\frac{1}{2}a(bcC-2bBd-aCd)\right)\right)\sqrt{c+d\tan(e+fx)}}{3(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))^{3/2}} - \frac{2\left(-\frac{3b(bc-ad)\left(\frac{(a+ib)^2(iA+B-iC)\sqrt{-c+id}\,\text{ar}\dots}{\dots}\right.}{\dots}\right.}{\dots}}{\dots}$$

```
[In] Integrate[(Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)
)/(a + b*Tan[e + f*x])^(5/2),x]
```

```
[Out] -((C*Sqrt[c + d*Tan[e + f*x]])/(b*f*(a + b*Tan[e + f*x])^(3/2))) - ((-2*((b
^2*(-2*A*b*c + 3*b*c*C - a*C*d))/2 - a*(-(b^2*(B*c + (A - C)*d)) - (a*(b*c*
C - 2*b*B*d - a*C*d))/2))*Sqrt[c + d*Tan[e + f*x]])/(3*(a^2 + b^2)*(b*c - a
*d)*f*(a + b*Tan[e + f*x])^(3/2)) - (2*((-3*b*(b*c - a*d)*(((a + I*b)^2*(I*
```

```
A + B - I*C)*Sqrt[-c + I*d]*ArcTanh[(Sqrt[-c + I*d]*Sqrt[a + b*Tan[e + f*x]
])/(Sqrt[-a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/Sqrt[-a + I*b] + ((a - I*b)^
2*(B - I*(A - C))*Sqrt[c + I*d]*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f
*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/Sqrt[a + I*b]))/(2*(a^2 +
b^2)*f) - (2*((b^2*(b*c - a*d)*(a^2*C*d + b^2*(3*B*c + A*d) + a*b*(3*A*c -
3*c*C - B*d)))/2 - a*((a*(2*A*b^2 - 2*a*b*B - a^2*C - 3*b^2*C)*d*(b*c - a*d
))/2 - (3*b^2*(b*c - a*d)*(A*b*c - a*B*c - b*c*C - a*A*d - b*B*d + a*C*d))/
2))*Sqrt[c + d*Tan[e + f*x]])/((a^2 + b^2)*(b*c - a*d)*f*Sqrt[a + b*Tan[e +
 f*x]])))/(3*(a^2 + b^2)*(b*c - a*d)))/b
```

Maple [F(-1)]

Timed out.

$$\int \frac{\sqrt{c + d\tan(fx+e)}\,(A + B\tan(fx+e) + C\tan(fx+e)^2)}{(a + b\tan(fx+e))^{\frac{5}{2}}}\,dx$$

```
[In] int((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(5/2),x)
```

```
[Out] int((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(5/2),x)
```

Fricas [F(-1)]

Timed out.

$$\int \frac{\sqrt{c + d\tan(e+fx)}(A + B\tan(e+fx) + C\tan^2(e+fx))}{(a + b\tan(e+fx))^{5/2}}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(5/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int \frac{\sqrt{c + d\tan(e+fx)}(A + B\tan(e+fx) + C\tan^2(e+fx))}{(a + b\tan(e+fx))^{5/2}}\,dx = \int \frac{\sqrt{c + d\tan(e+fx)}(A + B\tan(e+ \cdots}{(a + b\tan(e+f \cdots}$$

```
[In] integrate((c+d*tan(f*x+e))**(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e))**(5/2),x)
```

```
[Out] Integral(sqrt(c + d*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/
(a + b*tan(e + f*x))**(5/2), x)
```

Maxima [F(-2)]

Exception generated.

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{5/2}}\,dx = \text{Exception raised: ValueError}$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(5/2),x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(((2*b*d+2*a*c)^2>0)', see `assume?`
 for mo
```

Giac [F(-1)]

Timed out.

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{5/2}}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(5/2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{5/2}}\,dx = \text{Hanged}$$

```
[In] int(((c + d*tan(e + f*x))^(1/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x))^(5/2),x)
```

```
[Out] \text{Hanged}
```

3.134 $\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{7/2}} dx$

Optimal result 1905
Rubi [A] (verified) 1906
Mathematica [A] (verified) 1910
Maple [F(-1)] 1911
Fricas [F(-1)] 1911
Sympy [F] 1911
Maxima [F(-1)] 1912
Giac [F(-1)] 1912
Mupad [F(-1)] 1912

Optimal result

Integrand size = 49, antiderivative size = 597

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{7/2}} dx =$$
$$-\frac{(iA+B-iC)\sqrt{c-id}\,\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a-ib)^{7/2}f}$$
$$-\frac{(B-i(A-C))\sqrt{c+id}\,\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a+ib)^{7/2}f}$$
$$-\frac{2(Ab^2-a(bB-aC))\sqrt{c+d\tan(e+fx)}}{5b(a^2+b^2)f(a+b\tan(e+fx))^{5/2}}$$
$$-\frac{2(4a^3bBd+a^4Cd+b^4(5Bc+Ad)+2ab^3(5Ac-5cC-3Bd)-a^2b^2(5Bc+9Ad-11Cd))\sqrt{c+d\tan(e}{15b(a^2+b^2)^2(bc-ad)f(a+b\tan(e+fx))^{3/2}}$$
$$+\frac{2(8a^5bBd^2+2a^6Cd^2-a^4b^2d(25Bc+33Ad-39Cd)-a^2b^4(45Ac^2-45c^2C-90Bcd-29Ad^2+23Cd^2)}{15b(a^2}$$

```
[Out] -(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(1/2)/(c+
d*tan(f*x+e))^(1/2))*(c-I*d)^(1/2)/(a-I*b)^(7/2)/f-(B-I*(A-C))*arctanh((c+I
*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))*(c+I
*d)^(1/2)/(a+I*b)^(7/2)/f+2/15*(8*a^5*b*B*d^2+2*a^6*C*d^2-a^4*b^2*d*(33*A*d
+25*B*c-39*C*d)-a^2*b^4*(45*A*c^2-29*A*d^2-90*B*c*d-45*C*c^2+23*C*d^2)+a^3*
b^3*(80*c*(A-C)*d+B*(15*c^2-49*d^2))-a*b^5*(40*c*(A-C)*d+B*(45*c^2-3*d^2))-
b^6*(5*c*(B*d+3*C*c)-A*(15*c^2+2*d^2)))*(c+d*tan(f*x+e))^(1/2)/b/(a^2+b^2)^
3/(-a*d+b*c)^2/f/(a+b*tan(f*x+e))^(1/2)-2/5*(A*b^2-a*(B*b-C*a))*(c+d*tan(f*
x+e))^(1/2)/b/(a^2+b^2)/f/(a+b*tan(f*x+e))^(5/2)-2/15*(4*a^3*b*B*d+a^4*C*d+
b^4*(A*d+5*B*c)+2*a*b^3*(5*A*c-3*B*d-5*C*c)-a^2*b^2*(9*A*d+5*B*c-11*C*d))*(
c+d*tan(f*x+e))^(1/2)/b/(a^2+b^2)^2/(-a*d+b*c)/f/(a+b*tan(f*x+e))^(3/2)
```

Rubi [A] (verified)

Time = 4.04 (sec) , antiderivative size = 597, normalized size of antiderivative = 1.00, number of steps used = 10, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.122, Rules used = {3726, 3730, 3697, 3696, 95, 214}

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{7/2}}\,dx =$$

$$-\frac{2(Ab^2-a(bB-aC))\sqrt{c+d\tan(e+fx)}}{5bf(a^2+b^2)(a+b\tan(e+fx))^{5/2}}$$

$$-\frac{2\sqrt{c+d\tan(e+fx)}(a^4Cd+4a^3bBd-a^2b^2(9Ad+5Bc-11Cd)+2ab^3(5Ac-3Bd-5cC)+b^4(Ad+5\cdots}{15bf(a^2+b^2)^2(bc-ad)(a+b\tan(e+fx))^{3/2}}$$

$$+\frac{2\sqrt{c+d\tan(e+fx)}(2a^6Cd^2+8a^5bBd^2-a^4b^2d(33Ad+25Bc-39Cd)+a^3b^3(80cd(A-C)+B(15c^2-\cdots}{15bf\,(a^2\cdots}$$

$$-\frac{\sqrt{c-id}(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(a-ib)^{7/2}}$$

$$-\frac{\sqrt{c+id}(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(a+ib)^{7/2}}$$

```
[In] Int[(Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a +
 b*Tan[e + f*x])^(7/2),x]
```

```
[Out] -(((I*A + B - I*C)*Sqrt[c - I*d]*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])]])/((a - I*b)^(7/2)*f)) - ((
B - I*(A - C))*Sqrt[c + I*d]*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]
])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])]])/((a + I*b)^(7/2)*f) - (2*(A*b
^2 - a*(b*B - a*C))*Sqrt[c + d*Tan[e + f*x]])/(5*b*(a^2 + b^2)*f*(a + b*Tan
[e + f*x])^(5/2)) - (2*(4*a^3*b*B*d + a^4*C*d + b^4*(5*B*c + A*d) + 2*a*b^3
*(5*A*c - 5*c*C - 3*B*d) - a^2*b^2*(5*B*c + 9*A*d - 11*C*d))*Sqrt[c + d*Tan
[e + f*x]])/(15*b*(a^2 + b^2)^2*(b*c - a*d)*f*(a + b*Tan[e + f*x])^(3/2)) +
 (2*(8*a^5*b*B*d^2 + 2*a^6*C*d^2 - a^4*b^2*d*(25*B*c + 33*A*d - 39*C*d) - a
^2*b^4*(45*A*c^2 - 45*c^2*C - 90*B*c*d - 29*A*d^2 + 23*C*d^2) + a^3*b^3*(80
*c*(A - C)*d + B*(15*c^2 - 49*d^2)) - a*b^5*(40*c*(A - C)*d + B*(45*c^2 - 3
*d^2)) - b^6*(5*c*(3*c*C + B*d) - A*(15*c^2 + 2*d^2)))*Sqrt[c + d*Tan[e + f
*x]])/(15*b*(a^2 + b^2)^3*(b*c - a*d)^2*f*Sqrt[a + b*Tan[e + f*x]])
```

<u>Rule 95</u>

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3696

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[A^2/f, Subst[Int[(a + b*x)^m*((c + d*x)^n/(A - B*x)), x], x, Tan[e + f*x
]], x] /; FreeQ[{a, b, c, d, e, f, A, B, m, n}, x] && NeQ[b*c - a*d, 0] &&
NeQ[a^2 + b^2, 0] && EqQ[A^2 + B^2, 0]
```

Rule 3697

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[(A + I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^n*(1 - I*Ta
n[e + f*x]), x], x] + Dist[(A - I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*T
an[e + f*x])^n*(1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A,
B, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[A^2 + B^2, 0]
```

Rule 3726

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*d^2 + c*(c*C - B*d))*(a + b*Tan[e +
 f*x])^m*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3730

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 - a*(b*B - a*C))*(a + b*Tan[e +
 f*x])^(m + 1)*((c + d*Tan[e + f*x])^(n + 1)/(f*(m + 1)*(b*c - a*d)*(a^2 +
b^2))), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
```

```
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] && !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rubi steps

$$\text{integral} = -\frac{2(Ab^2-a(bB-aC))\sqrt{c+d\tan(e+fx)}}{5b(a^2+b^2)f(a+b\tan(e+fx))^{5/2}} + \frac{2\int \frac{\frac{1}{2}((bB-aC)(5bc-ad)+Ab(5ac+bd))-\frac{5}{2}b((A-C)(bc-ad)-B(ac+bd))\tan(e+fx)-\frac{1}{2}(4Ab^2-4abB-a^2C-5b^2C)d\tan^2(e+fx)}{(a+b\tan(e+fx))^{5/2}\sqrt{c+d\tan(e+fx)}}dx}{5b(a^2+b^2)}$$

$$= -\frac{2(Ab^2-a(bB-aC))\sqrt{c+d\tan(e+fx)}}{5b(a^2+b^2)f(a+b\tan(e+fx))^{5/2}} - \frac{2(4a^3bBd+a^4Cd+b^4(5Bc+Ad)+2ab^3(5Ac-5cC-3Bd)-a^2b^2(5Bc+9Ad-11Cd))\sqrt{c-}}{15b(a^2+b^2)^2(bc-ad)f(a+b\tan(e+fx))^{3/2}} - \frac{4\int \frac{\frac{1}{4}(2(b^2d-\frac{3}{2}a(bc-ad))((bB-aC)(5bc-ad)+Ab(5ac+bd))+(3bc-ad)(4a^2bBd+a^3Cd+Ab^2(5bc-9ad)-5b^3(cC+Bd)-5ab^2(Bc-}{}}{}$$

$$= -\frac{2(Ab^2-a(bB-aC))\sqrt{c+d\tan(e+fx)}}{5b(a^2+b^2)f(a+b\tan(e+fx))^{5/2}} - \frac{2(4a^3bBd+a^4Cd+b^4(5Bc+Ad)+2ab^3(5Ac-5cC-3Bd)-a^2b^2(5Bc+9Ad-11Cd))\sqrt{c-}}{15b(a^2+b^2)^2(bc-ad)f(a+b\tan(e+fx))^{3/2}} + \frac{2(8a^5bBd^2+2a^6Cd^2-a^4b^2d(25Bc+33Ad-39Cd)-a^2b^4(45Ac^2-45c^2C-90Bcd-29Ad^2-}{} + \frac{8\int \frac{\frac{15}{8}b(bc-ad)^2(a^3(Ac-cC-Bd)-3ab^2(Ac-cC-Bd)+3a^2b(Bc+(A-C)d)-b^3(Bc+(A-C)d))-\frac{15}{8}b(bc-ad)^2(3a^2b(Ac-cC-Bd)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}}{15b(a^2+b^2)^3(bc-ad)^2}$$

$$= -\frac{2(Ab^2-a(bB-aC))\sqrt{c+d\tan(e+fx)}}{5b(a^2+b^2)f(a+b\tan(e+fx))^{5/2}} - \frac{2(4a^3bBd+a^4Cd+b^4(5Bc+Ad)+2ab^3(5Ac-5cC-3Bd)-a^2b^2(5Bc+9Ad-11Cd))\sqrt{c-}}{15b(a^2+b^2)^2(bc-ad)f(a+b\tan(e+fx))^{3/2}} + \frac{2(8a^5bBd^2+2a^6Cd^2-a^4b^2d(25Bc+33Ad-39Cd)-a^2b^4(45Ac^2-45c^2C-90Bcd-29Ad^2-}{} + \frac{((A-iB-C)(c-id))\int \frac{1+i\tan(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}dx}{2(a-ib)^3} + \frac{((A+iB-C)(c+id))\int \frac{1-i\tan(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}dx}{2(a+ib)^3}$$

$$= -\frac{2(Ab^2 - a(bB - aC))\sqrt{c + d\tan(e + fx)}}{5b(a^2 + b^2) f(a + b\tan(e + fx))^{5/2}}$$
$$- \frac{2(4a^3bBd + a^4Cd + b^4(5Bc + Ad) + 2ab^3(5Ac - 5cC - 3Bd) - a^2b^2(5Bc + 9Ad - 11Cd))\sqrt{c}}{15b(a^2 + b^2)^2 (bc - ad)f(a + b\tan(e + fx))^{3/2}}$$
$$+ \frac{2(8a^5bBd^2 + 2a^6Cd^2 - a^4b^2d(25Bc + 33Ad - 39Cd) - a^2b^4(45Ac^2 - 45c^2C - 90Bcd - 29Ad^2}{}$$
$$+ \frac{((A - iB - C)(c - id))\text{Subst}\left(\int \frac{1}{(1-ix)\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e + fx)\right)}{2(a - ib)^3 f}$$
$$+ \frac{((A + iB - C)(c + id))\text{Subst}\left(\int \frac{1}{(1+ix)\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e + fx)\right)}{2(a + ib)^3 f}$$

$$= -\frac{2(Ab^2 - a(bB - aC))\sqrt{c + d\tan(e + fx)}}{5b(a^2 + b^2) f(a + b\tan(e + fx))^{5/2}}$$
$$- \frac{2(4a^3bBd + a^4Cd + b^4(5Bc + Ad) + 2ab^3(5Ac - 5cC - 3Bd) - a^2b^2(5Bc + 9Ad - 11Cd))\sqrt{c}}{15b(a^2 + b^2)^2 (bc - ad)f(a + b\tan(e + fx))^{3/2}}$$
$$+ \frac{2(8a^5bBd^2 + 2a^6Cd^2 - a^4b^2d(25Bc + 33Ad - 39Cd) - a^2b^4(45Ac^2 - 45c^2C - 90Bcd - 29Ad^2}{}$$
$$+ \frac{((A - iB - C)(c - id))\text{Subst}\left(\int \frac{1}{ia+b-(ic+d)x^2}\,dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{(a - ib)^3 f}$$
$$+ \frac{((A + iB - C)(c + id))\text{Subst}\left(\int \frac{1}{-ia+b-(-ic+d)x^2}\,dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{(a + ib)^3 f}$$

$$= -\frac{(iA + B - iC)\sqrt{c - id}\,\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a - ib)^{7/2} f}$$
$$- \frac{(B - i(A - C))\sqrt{c + id}\,\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a + ib)^{7/2} f}$$
$$- \frac{2(Ab^2 - a(bB - aC))\sqrt{c + d\tan(e + fx)}}{5b(a^2 + b^2) f(a + b\tan(e + fx))^{5/2}}$$
$$- \frac{2(4a^3bBd + a^4Cd + b^4(5Bc + Ad) + 2ab^3(5Ac - 5cC - 3Bd) - a^2b^2(5Bc + 9Ad - 11Cd))\sqrt{c}}{15b(a^2 + b^2)^2 (bc - ad)f(a + b\tan(e + fx))^{3/2}}$$
$$+ \frac{2(8a^5bBd^2 + 2a^6Cd^2 - a^4b^2d(25Bc + 33Ad - 39Cd) - a^2b^4(45Ac^2 - 45c^2C - 90Bcd - 29Ad^2}{}$$

Mathematica [A] (verified)

Time = 7.51 (sec) , antiderivative size = 1109, normalized size of antiderivative = 1.86

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{7/2}}\,dx =$$
$$-\frac{C\sqrt{c+d\tan(e+fx)}}{2bf(a+b\tan(e+fx))^{5/2}}$$
$$-\frac{-\frac{2\left(\frac{1}{2}b^2(-4Abc+5bcC-aCd)-a\left(-2b^2(Bc+(A-C)d)-\frac{1}{2}a(bcC-4bBd-aCd)\right)\right)\sqrt{c+d\tan(e+fx)}}{5(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))^{5/2}} - \frac{2\left(-\frac{2\left(b^2(bc-ad)\left(a^2Cd+b^2(5Bc+Ad)+ab(5Ac\right.\right.}{}\right.}{}$$

```
[In] Integrate[(Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)
)/(a + b*Tan[e + f*x])^(7/2),x]
```

```
[Out] -1/2*(C*Sqrt[c + d*Tan[e + f*x]])/(b*f*(a + b*Tan[e + f*x])^(5/2)) - ((-2*(
(b^2*(-4*A*b*c + 5*b*c*C - a*C*d))/2 - a*(-2*b^2*(B*c + (A - C)*d) - (a*(b*
c*C - 4*b*B*d - a*C*d))/2))*Sqrt[c + d*Tan[e + f*x]])/(5*(a^2 + b^2)*(b*c -
 a*d)*f*(a + b*Tan[e + f*x])^(5/2)) - (2*((-2*(b^2*(b*c - a*d)*(a^2*C*d + b
^2*(5*B*c + A*d) + a*b*(5*A*c - 5*c*C - B*d)) - a*(a*(4*A*b^2 - 4*a*b*B - a
^2*C - 5*b^2*C)*d*(b*c - a*d) - 5*b^2*(b*c - a*d)*(A*b*c - a*B*c - b*c*C -
a*A*d - b*B*d + a*C*d)))*Sqrt[c + d*Tan[e + f*x]])/(3*(a^2 + b^2)*(b*c - a*
d)*f*(a + b*Tan[e + f*x])^(3/2)) - (2*((-15*b*(b*c - a*d)^2*(((I*a - b)^3*(
A - I*B - C)*Sqrt[-c + I*d]*ArcTanh[(Sqrt[-c + I*d]*Sqrt[a + b*Tan[e + f*x]
])/(Sqrt[-a + I*b]*Sqrt[c + d*Tan[e + f*x]])]])/Sqrt[-a + I*b] - ((I*a + b)^
3*(A + I*B - C)*Sqrt[c + I*d]*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x
]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])]])/Sqrt[a + I*b]))/(2*(a^2 + b^
2)*f) - (2*(b^2*((b*c - a*d)*(b^2*d - (3*a*(b*c - a*d))/2)*(a^2*C*d + b^2*(
5*B*c + A*d) + a*b*(5*A*c - 5*c*C - B*d)) + ((-3*b*c)/2 + (a*d)/2)*(a*(4*A*
b^2 - 4*a*b*B - a^2*C - 5*b^2*C)*d*(b*c - a*d) - 5*b^2*(b*c - a*d)*(A*b*c -
 a*B*c - b*c*C - a*A*d - b*B*d + a*C*d))) - a*((3*b*(b*c - a*d)*(b*(4*A*b^2
 - 4*a*b*B - a^2*C - 5*b^2*C)*d*(b*c - a*d) + 5*a*b*(b*c - a*d)*(A*b*c - a*
B*c - b*c*C - a*A*d - b*B*d + a*C*d) + b*(b*c - a*d)*(a^2*C*d + b^2*(5*B*c
+ A*d) + a*b*(5*A*c - 5*c*C - B*d))))/2 - a*d*(b^2*(b*c - a*d)*(a^2*C*d + b
^2*(5*B*c + A*d) + a*b*(5*A*c - 5*c*C - B*d)) - a*(a*(4*A*b^2 - 4*a*b*B - a
^2*C - 5*b^2*C)*d*(b*c - a*d) - 5*b^2*(b*c - a*d)*(A*b*c - a*B*c - b*c*C -
a*A*d - b*B*d + a*C*d)))))*Sqrt[c + d*Tan[e + f*x]])/((a^2 + b^2)*(b*c - a*
```

```
d)*f*Sqrt[a + b*Tan[e + f*x]])))/(3*(a^2 + b^2)*(b*c - a*d))))/(5*(a^2 + b^
2)*(b*c - a*d)))/(2*b)
```

Maple [F(-1)]

Timed out.

$$\int \frac{\sqrt{c+d\tan\left(fx+e\right)}\,\left(A+B\tan\left(fx+e\right)+C\tan\left(fx+e\right)^{2}\right)}{\left(a+b\tan\left(fx+e\right)\right)^{\frac{7}{2}}}dx$$

```
[In] int((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(7/2),x)
```

```
[Out] int((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(7/2),x)
```

Fricas [F(-1)]

Timed out.

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{7/2}}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(7/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{7/2}}\,dx = \int \frac{\sqrt{c+d\tan\left(e+fx\right)}(A+B\tan\left(e+f\right.$$

$$\left(a+b\tan\left(e+f\right.\right.$$

```
[In] integrate((c+d*tan(f*x+e))**(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e))**(7/2),x)
```

```
[Out] Integral(sqrt(c + d*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/
(a + b*tan(e + f*x))**(7/2), x)
```

Maxima [F(-1)]

Timed out.

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{7/2}}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(7/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

Giac [F(-1)]

Timed out.

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{7/2}}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(7/2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{\sqrt{c+d\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{7/2}}\,dx = \text{Hanged}$$

```
[In] int(((c + d*tan(e + f*x))^(1/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x))^(7/2),x)
```

```
[Out] \text{Hanged}
```

3.135 $\int (a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}\left(A+B\tan\right.$

Optimal result 1913
Rubi [A] (verified) 1914
Mathematica [A] (verified) 1920
Maple [F(-1)] 1921
Fricas [F(-1)] 1921
Sympy [F] 1922
Maxima [F] 1922
Giac [F(-1)] 1922
Mupad [F(-1)] 1923

Optimal result

Integrand size = 49, antiderivative size = 682

$$\int (a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=$$

$$-\frac{(a-ib)^{3/2}(B+i(A-C))(c-id)^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f}$$

$$-\frac{(a+ib)^{3/2}(B-i(A-C))(c+id)^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f}$$

$$+\frac{(3a^4Cd^4-4a^3bd^3(3cC+2Bd)+6a^2b^2d^2(3c^2C+12Bcd+8(A-C)d^2)-12ab^3d(c^3C-6Bc^2d-24c(A}{64}$$

$$+\frac{(64b(a^2B-b^2B+2ab(A-C))\,d^3+(bc-ad)\,(48b(Ab+aB-bC)d^2+(bc-ad)(3bcC-8bBd-3aCd)}{64b^2d^2f}$$

$$+\frac{(48b(Ab+aB-bC)d^2+(bc-ad)(3bcC-8bBd-3aCd))\,\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{96bd^2f}$$

$$-\frac{(3bcC-8bBd-3aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{24d^2f}$$

$$+\frac{C(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{5/2}}{4df}$$

```
[Out] -(a-I*b)^(3/2)*(B+I*(A-C))*(c-I*d)^(3/2)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x
+e))^(1/2)/(a-I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/f-(a+I*b)^(3/2)*(B-I*(A-C)
)*(c+I*d)^(3/2)*arctanh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/
(c+d*tan(f*x+e))^(1/2))/f+1/64*(3*a^4*C*d^4-4*a^3*b*d^3*(2*B*d+3*C*c)+6*a^2
*b^2*d^2*(3*c^2*C+12*B*c*d+8*(A-C)*d^2)-12*a*b^3*d*(c^3*C-6*B*c^2*d-24*c*(A
-C)*d^2+16*B*d^3)+b^4*(3*c^4*C-8*B*c^3*d+48*c^2*(A-C)*d^2-192*B*c*d^3-128*(
A-C)*d^4))*arctanh(d^(1/2)*(a+b*tan(f*x+e))^(1/2)/b^(1/2)/(c+d*tan(f*x+e))^
```

```
(1/2))/b^(5/2)/d^(5/2)/f+1/64*(64*b*(B*a^2-B*b^2+2*a*b*(A-C))*d^3+(-a*d+b*c
)*(48*b*(A*b+B*a-C*b)*d^2+(-a*d+b*c)*(-8*B*b*d-3*C*a*d+3*C*b*c)))*(a+b*tan(
f*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)/b^2/d^2/f+1/96*(48*b*(A*b+B*a-C*b)*d^2
+(-a*d+b*c)*(-8*B*b*d-3*C*a*d+3*C*b*c))*(a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x
+e))^(3/2)/b/d^2/f-1/24*(-8*B*b*d-3*C*a*d+3*C*b*c)*(a+b*tan(f*x+e))^(1/2)*(
c+d*tan(f*x+e))^(5/2)/d^2/f+1/4*C*(a+b*tan(f*x+e))^(3/2)*(c+d*tan(f*x+e))^(
5/2)/d/f
```

Rubi [A] (verified)

Time = 14.55 (sec) , antiderivative size = 682, normalized size of antiderivative = 1.00, number of steps used = 16, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}} = 0.163$, Rules used = {3728, 3736, 6857, 65, 223, 212, 95, 214}

$$\int (a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\frac{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}(64bd^3(a$$

$$+\frac{(3a^4Cd^4-4a^3bd^3(2Bd+3cC)+6a^2b^2d^2(8d^2(A-C)+12Bcd+3c^2C)-12ab^3d(-24cd^2(A-C)-6Bc^2d}{64b}$$

$$-\frac{(a-ib)^{3/2}(c-id)^{3/2}(B+i(A-C))\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f}$$

$$-\frac{(a+ib)^{3/2}(c+id)^{3/2}(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f}$$

$$+\frac{\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}\left(48bd^2(aB+Ab-bC)+(bc-ad)(-3aCd-8bBd+3bcC)\right)}{96bd^2f}$$

$$-\frac{(-3aCd-8bBd+3bcC)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{24d^2f}$$

$$+\frac{C(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{5/2}}{4df}$$

```
[In] Int[(a + b*Tan[e + f*x])^(3/2)*(c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*
x] + C*Tan[e + f*x]^2),x]
```

```
[Out] -(((a - I*b)^(3/2)*(B + I*(A - C))*(c - I*d)^(3/2)*ArcTanh[(Sqrt[c - I*d]*S
qrt[a + b*Tan[e + f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/f) - ((
a + I*b)^(3/2)*(B - I*(A - C))*(c + I*d)^(3/2)*ArcTanh[(Sqrt[c + I*d]*Sqrt[
a + b*Tan[e + f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/f + ((3*a^4
*C*d^4 - 4*a^3*b*d^3*(3*c*C + 2*B*d) + 6*a^2*b^2*d^2*(3*c^2*C + 12*B*c*d +
8*(A - C)*d^2) - 12*a*b^3*d*(c^3*C - 6*B*c^2*d - 24*c*(A - C)*d^2 + 16*B*d^
3) + b^4*(3*c^4*C - 8*B*c^3*d + 48*c^2*(A - C)*d^2 - 192*B*c*d^3 - 128*(A -
 C)*d^4))*ArcTanh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c + d*Ta
```

```
n[e + f*x]])])/(64*b^(5/2)*d^(5/2)*f) + ((64*b*(a^2*B - b^2*B + 2*a*b*(A -
C))*d^3 + (b*c - a*d)*(48*b*(A*b + a*B - b*C)*d^2 + (b*c - a*d)*(3*b*c*C -
8*b*B*d - 3*a*C*d)))*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(64
*b^2*d^2*f) + ((48*b*(A*b + a*B - b*C)*d^2 + (b*c - a*d)*(3*b*c*C - 8*b*B*d
 - 3*a*C*d))*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(3/2))/(96*b*d^2
*f) - ((3*b*c*C - 8*b*B*d - 3*a*C*d)*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e
+ f*x])^(5/2))/(24*d^2*f) + (C*(a + b*Tan[e + f*x])^(3/2)*(c + d*Tan[e + f*
x])^(5/2))/(4*d*f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 95

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 223

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
```

```
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3736

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[(a + b*ff*x)^m*(c + d*ff*x)^n*((A + B*ff*x + C*ff^2*x^2
)/(1 + ff^2*x^2)), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6857

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\text{integral} = \frac{C(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{5/2}}{4df} + \frac{\int \sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}\left(\frac{1}{2}(-3bcC+a(8A-5C)d)+4(Ab+aB-bC)d\tan(e+f\right.}{4d}$$

$$= -\frac{(3bcC-8bBd-3aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{24d^2f} + \frac{C(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{5/2}}{4df} + \frac{\int \frac{(c+d\tan(e+fx))^{3/2}\left(\frac{1}{4}\left(3a^2(16A-15C)d^2+b^2c(3cC-8Bd)-2abd(3cC+20Bd)\right)+12\left(a^2B-b^2B+2ab(A-C)\right)d^2\tan(e+fx)+\frac{1}{4}\left(48\right.\right.}{\sqrt{a+b\tan(e+fx)}}}{12d^2}$$

$$= \frac{(48b(Ab+aB-bC)d^2+(bc-ad)(3bcC-8bBd-3aCd))\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx}{96bd^2f}$$

$$-\frac{(3bcC-8bBd-3aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{24d^2f}$$

$$+\frac{C(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{5/2}}{4df}$$

$$+\frac{\int\frac{\sqrt{c+d\tan(e+fx)}\left(-\frac{3}{8}\left(3a^3Cd^3-a^2bd^2(64Ac-55cC-56Bd)-b^3c(3c^2C-8Bcd-16(A-C)d^2)+ab^2d(9c^2C+64Bcd+48(A-C)d^2\right.\right.}{}}{}$$

$$= \frac{(64b(a^2B-b^2B+2ab(A-C))\,d^3+(bc-ad)\,(48b(Ab+aB-bC)d^2+(bc-ad)(3bcC-8bBd}{64b^2d^2f}$$

$$+\frac{(48b(Ab+aB-bC)d^2+(bc-ad)(3bcC-8bBd-3aCd))\sqrt{a+b\tan(e+fx)}(c+d\tan(e+}{96bd^2f}$$

$$-\frac{(3bcC-8bBd-3aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{24d^2f}$$

$$+\frac{C(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{5/2}}{4df}$$

$$+\frac{\int\frac{\frac{3}{16}\left(3a^4Cd^4-4a^3bd^3(3cC+2Bd)-4ab^3d(3c^3C+46Bc^2d+56c(A-C)d^2-16Bd^3)+b^4c(3c^3C-8Bc^2d-80c(A-C)d^2+64Bd^3)-2a\right.}{}}{}$$

$$= \frac{(64b(a^2B-b^2B+2ab(A-C))\,d^3+(bc-ad)\,(48b(Ab+aB-bC)d^2+(bc-ad)(3bcC-8bBd}{64b^2d^2f}$$

$$+\frac{(48b(Ab+aB-bC)d^2+(bc-ad)(3bcC-8bBd-3aCd))\sqrt{a+b\tan(e+fx)}(c+d\tan(e+}{96bd^2f}$$

$$-\frac{(3bcC-8bBd-3aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{24d^2f}$$

$$+\frac{C(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{5/2}}{4df}$$

$$+\frac{\text{Subst}\left(\int\frac{\frac{3}{16}\left(3a^4Cd^4-4a^3bd^3(3cC+2Bd)-4ab^3d(3c^3C+46Bc^2d+56c(A-C)d^2-16Bd^3)+b^4c(3c^3C-8Bc^2d-80c(A-C)d^2+6\right.}{}\right.}{}$$

$$= \frac{(64b(a^2B - b^2B + 2ab(A-C))\,d^3 + (bc-ad)\,(48b(Ab + aB - bC)d^2 + (bc-ad)(3bcC - 8bBd -}{64b^2d^2f}$$
$$+ \frac{(48b(Ab + aB - bC)d^2 + (bc-ad)(3bcC - 8bBd - 3aCd))\,\sqrt{a + b\tan(e+fx)}(c + d\tan(e+f}{96bd^2f}$$
$$- \frac{(3bcC - 8bBd - 3aCd)\sqrt{a + b\tan(e+fx)}(c + d\tan(e+fx))^{5/2}}{24d^2f}$$
$$+ \frac{C(a + b\tan(e+fx))^{3/2}(c + d\tan(e+fx))^{5/2}}{4df}$$
$$+ \frac{\text{Subst}\Big(\int \Big(\frac{3(3a^4Cd^4 - 4a^3bd^3(3cC+2Bd) + 6a^2b^2d^2(3c^2C + 12Bcd + 8(A-C)d^2) - 12ab^3d(c^3C - 6Bc^2d - 24c(A-C)d^2 + 16Bd^3) + b}{16\sqrt{a+bx}\sqrt{c+dx}}}{}$$

$$= \frac{(64b(a^2B - b^2B + 2ab(A-C))\,d^3 + (bc-ad)\,(48b(Ab + aB - bC)d^2 + (bc-ad)(3bcC - 8bBd -}{64b^2d^2f}$$
$$+ \frac{(48b(Ab + aB - bC)d^2 + (bc-ad)(3bcC - 8bBd - 3aCd))\,\sqrt{a + b\tan(e+fx)}(c + d\tan(e+f}{96bd^2f}$$
$$- \frac{(3bcC - 8bBd - 3aCd)\sqrt{a + b\tan(e+fx)}(c + d\tan(e+fx))^{5/2}}{24d^2f}$$
$$+ \frac{C(a + b\tan(e+fx))^{3/2}(c + d\tan(e+fx))^{5/2}}{4df}$$
$$+ \frac{\text{Subst}\Big(\int \frac{-b^2d^2(a^2(c^2C + 2Bcd - Cd^2 - A(c^2-d^2)) - b^2(c^2C + 2Bcd - Cd^2 - A(c^2-d^2)) + 2ab(2c(A-C)d + B(c^2-d^2))) - b^2d^2(2ab(}{\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}}{b^2d^2f}$$
$$+ \frac{(3a^4Cd^4 - 4a^3bd^3(3cC + 2Bd) + 6a^2b^2d^2(3c^2C + 12Bcd + 8(A-C)d^2) - 12ab^3d(c^3C - 6Bc^2d}{}$$

$$= \frac{(64b(a^2B - b^2B + 2ab(A-C))\,d^3 + (bc-ad)\,(48b(Ab + aB - bC)d^2 + (bc-ad)(3bcC - 8bBd -}{64b^2d^2f}$$
$$+ \frac{(48b(Ab + aB - bC)d^2 + (bc-ad)(3bcC - 8bBd - 3aCd))\,\sqrt{a + b\tan(e+fx)}(c + d\tan(e+f}{96bd^2f}$$
$$- \frac{(3bcC - 8bBd - 3aCd)\sqrt{a + b\tan(e+fx)}(c + d\tan(e+fx))^{5/2}}{24d^2f}$$
$$+ \frac{C(a + b\tan(e+fx))^{3/2}(c + d\tan(e+fx))^{5/2}}{4df}$$
$$+ \frac{\text{Subst}\Big(\int \Big(\frac{-ib^2d^2(a^2(c^2C + 2Bcd - Cd^2 - A(c^2-d^2)) - b^2(c^2C + 2Bcd - Cd^2 - A(c^2-d^2)) + 2ab(2c(A-C)d + B(c^2-d^2))) + b^2d^2(2a}{2(i-x)\sqrt{a+bx}\sqrt{c+dx}}}{}$$
$$+ \frac{(3a^4Cd^4 - 4a^3bd^3(3cC + 2Bd) + 6a^2b^2d^2(3c^2C + 12Bcd + 8(A-C)d^2) - 12ab^3d(c^3C - 6Bc^2d}{}$$

$$= \frac{(64b(a^2B - b^2B + 2ab(A - C))\, d^3 + (bc - ad)\,(48b(Ab + aB - bC)d^2 + (bc - ad)(3bcC - 8bBd}{64b^2d^2f}$$

$$+ \frac{(48b(Ab + aB - bC)d^2 + (bc - ad)(3bcC - 8bBd - 3aCd))\,\sqrt{a + b\tan(e + fx)}(c + d\tan(e +}{96bd^2f}$$

$$- \frac{(3bcC - 8bBd - 3aCd)\sqrt{a + b\tan(e + fx)}(c + d\tan(e + fx))^{5/2}}{24d^2f}$$

$$+ \frac{C(a + b\tan(e + fx))^{3/2}(c + d\tan(e + fx))^{5/2}}{4df}$$

$$+ \frac{((a - ib)^2(B + i(A - C))(c - id)^2)\,\mathrm{Subst}\left(\int \frac{1}{(i+x)\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e + fx)\right)}{2f}$$

$$+ \frac{(3a^4Cd^4 - 4a^3bd^3(3cC + 2Bd) + 6a^2b^2d^2(3c^2C + 12Bcd + 8(A - C)d^2) - 12ab^3d(c^3C - 6Bc^2}{}$$

$$+ \frac{(-ib^2d^2(a^2(c^2C + 2Bcd - Cd^2 - A(c^2 - d^2)) - b^2(c^2C + 2Bcd - Cd^2 - A(c^2 - d^2)) + 2ab(2c(}{}$$

$$= \frac{(3a^4Cd^4 - 4a^3bd^3(3cC + 2Bd) + 6a^2b^2d^2(3c^2C + 12Bcd + 8(A - C)d^2) - 12ab^3d(c^3C - 6Bc^2d}{}$$

$$+ \frac{(64b(a^2B - b^2B + 2ab(A - C))\, d^3 + (bc - ad)\,(48b(Ab + aB - bC)d^2 + (bc - ad)(3bcC - 8bB}{64b^2d^2f}$$

$$+ \frac{(48b(Ab + aB - bC)d^2 + (bc - ad)(3bcC - 8bBd - 3aCd))\,\sqrt{a + b\tan(e + fx)}(c + d\tan(e +}{96bd^2f}$$

$$- \frac{(3bcC - 8bBd - 3aCd)\sqrt{a + b\tan(e + fx)}(c + d\tan(e + fx))^{5/2}}{24d^2f}$$

$$+ \frac{C(a + b\tan(e + fx))^{3/2}(c + d\tan(e + fx))^{5/2}}{4df}$$

$$+ \frac{((a - ib)^2(B + i(A - C))(c - id)^2)\,\mathrm{Subst}\left(\int \frac{1}{-a+ib-(-c+id)x^2}\,dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{f}$$

$$+ \frac{(-ib^2d^2(a^2(c^2C + 2Bcd - Cd^2 - A(c^2 - d^2)) - b^2(c^2C + 2Bcd - Cd^2 - A(c^2 - d^2)) + 2ab(2c(}{}$$

$$= -\frac{(a-ib)^{3/2}(B+i(A-C))(c-id)^{3/2}\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f}$$
$$-\frac{(a+ib)^{3/2}(B-i(A-C))(c+id)^{3/2}\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f}$$
$$+\frac{(3a^4Cd^4-4a^3bd^3(3cC+2Bd)+6a^2b^2d^2(3c^2C+12Bcd+8(A-C)d^2)-12ab^3d(c^3C-6Bc^2d}{}$$
$$+\frac{(64b(a^2B-b^2B+2ab(A-C))\,d^3+(bc-ad)\,(48b(Ab+aB-bC)d^2+(bc-ad)(3bcC-8bBd}{64b^2d^2f}$$
$$+\frac{(48b(Ab+aB-bC)d^2+(bc-ad)(3bcC-8bBd-3aCd))\,\sqrt{a+b\tan(e+fx)}(c+d\tan(e+f}{96bd^2f}$$
$$-\frac{(3bcC-8bBd-3aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{24d^2f}$$
$$+\frac{C(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{5/2}}{4df}$$

Mathematica [A] (verified)

Time = 9.56 (sec) , antiderivative size = 1304, normalized size of antiderivative = 1.91

$$\int(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\frac{C(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{5/2}}{4df}$$

$$+\frac{\frac{(-3bcC+8bBd+3aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{6df}+\frac{\left(48b(Ab+aB-bC)d^2+(bc-ad)(3bcC-8bBd-3aCd)\right)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{8bf}}{}$$

```
[In] Integrate[(a + b*Tan[e + f*x])^(3/2)*(c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[
e + f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] (C*(a + b*Tan[e + f*x])^(3/2)*(c + d*Tan[e + f*x])^(5/2))/(4*d*f) + (((-3*b
*c*C + 8*b*B*d + 3*a*C*d)*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(5/
2))/(6*d*f) + (((48*b*(A*b + a*B - b*C)*d^2 + (b*c - a*d)*(3*b*c*C - 8*b*B*
d - 3*a*C*d))*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(3/2))/(8*b*f)
+ (((24*b*(a^2*B - b^2*B + 2*a*b*(A - C))*d^3 - (3*(-(b*c) + a*d)*(48*b*(A*
b + a*B - b*C)*d^2 + (b*c - a*d)*(3*b*c*C - 8*b*B*d - 3*a*C*d)))/8)*Sqrt[a
+ b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(b*f) + ((24*(-(b^4*Sqrt[-b^2]*
d^2*(a^2*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - b^2*(c^2*C + 2*B*c*d -
```

```
C*d^2 - A*(c^2 - d^2)) + 2*a*b*(2*c*(A - C)*d + B*(c^2 - d^2)))) - b^5*d^2
*(2*a*b*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - a^2*(2*c*(A - C)*d + B*
(c^2 - d^2)) + b^2*(2*c*(A - C)*d + B*(c^2 - d^2))))*ArcTanh[(Sqrt[-c + (Sq
rt[-b^2]*d)/b]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[c + d*
Tan[e + f*x]])])/(b^2*Sqrt[-a + Sqrt[-b^2]]*Sqrt[-c + (Sqrt[-b^2]*d)/b]) -
(24*b^2*d^2*(Sqrt[-b^2]*(a^2*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - b^
2*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2)) + 2*a*b*(2*c*(A - C)*d + B*(c^2
 - d^2))) - b*(2*a*b*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - a^2*(2*c*(
A - C)*d + B*(c^2 - d^2)) + b^2*(2*c*(A - C)*d + B*(c^2 - d^2))))*ArcTanh[(
Sqrt[c + (Sqrt[-b^2]*d)/b]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + Sqrt[-b^2]]*
Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[a + Sqrt[-b^2]]*Sqrt[c + (Sqrt[-b^2]*d)/b
]) + (3*Sqrt[b]*Sqrt[c - (a*d)/b]*Sqrt[(c/(c - (a*d)/b) - (a*d)/(b*(c - (a*
d)/b)))^(-1)]*Sqrt[c/(c - (a*d)/b) - (a*d)/(b*(c - (a*d)/b))]*(3*a^4*C*d^4
- 4*a^3*b*d^3*(3*c*C + 2*B*d) + 6*a^2*b^2*d^2*(3*c^2*C + 12*B*c*d + 8*(A -
C)*d^2) - 12*a*b^3*d*(c^3*C - 6*B*c^2*d - 24*c*(A - C)*d^2 + 16*B*d^3) + b^
4*(3*c^4*C - 8*B*c^3*d + 48*c^2*(A - C)*d^2 - 192*B*c*d^3 - 128*(A - C)*d^4
))*ArcSinh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c - (a*d)/b]*Sq
rt[c/(c - (a*d)/b) - (a*d)/(b*(c - (a*d)/b))])]*Sqrt[(c + d*Tan[e + f*x])/(
c - (a*d)/b)])/(8*Sqrt[d]*Sqrt[c + d*Tan[e + f*x]]))/(b^2*f))/(2*b))/(3*d))
/(4*d)
```

Maple [F(-1)]

Timed out.

$$\int (a+b\tan(fx+e))^{\frac{3}{2}}(c+d\tan(fx+e))^{\frac{3}{2}}\left(A+B\tan(fx+e)+C\tan(fx+e)^2\right)dx$$

```
[In] int((a+b*tan(f*x+e))^(3/2)*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x)
```

```
[Out] int((a+b*tan(f*x+e))^(3/2)*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x)
```

Fricas [F(-1)]

Timed out.

$$\int (a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^(3/2)*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*t
an(f*x+e)^2),x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int (a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\int (a+b\tan(e+fx))^{\frac{3}{2}}(c+d\tan(e+fx))^{\frac{3}{2}}(A+$$

```
[In] integrate((a+b*tan(f*x+e))**(3/2)*(c+d*tan(f*x+e))**(3/2)*(A+B*tan(f*x+e)+C
*tan(f*x+e)**2),x)
[Out] Integral((a + b*tan(e + f*x))**(3/2)*(c + d*tan(e + f*x))**(3/2)*(A + B*tan
(e + f*x) + C*tan(e + f*x)**2), x)
```

Maxima [F]

$$\int (a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\int \left(C\tan(fx+e)^2+B\tan(fx+e)+A\right)(b\tan$$

```
[In] integrate((a+b*tan(f*x+e))^(3/2)*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*t
an(f*x+e)^2),x, algorithm="maxima")
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)^(3/2
)*(d*tan(f*x + e) + c)^(3/2), x)
```

Giac [F(-1)]

Timed out.

$$\int (a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^(3/2)*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*t
an(f*x+e)^2),x, algorithm="giac")
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int (a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\int (a+b\tan(e+f\,x))^{3/2}\,(c+d\tan(e+f\,x))^{3/}$$

```
[In] int((a + b*tan(e + f*x))^(3/2)*(c + d*tan(e + f*x))^(3/2)*(A + B*tan(e + f*
x) + C*tan(e + f*x)^2),x)
[Out] int((a + b*tan(e + f*x))^(3/2)*(c + d*tan(e + f*x))^(3/2)*(A + B*tan(e + f*
x) + C*tan(e + f*x)^2), x)
```

3.136 $\int \sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}\left(A+B\tan\right.$

Optimal result 1924
Rubi [A] (verified) 1925
Mathematica [A] (verified) 1930
Maple [F(-1)] 1931
Fricas [F(-1)] 1931
Sympy [F] 1931
Maxima [F] 1932
Giac [F(-1)] 1932
Mupad [F(-1)] 1932

Optimal result

Integrand size = 49, antiderivative size = 508

$$\int \sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx =$$
$$-\frac{\sqrt{a-ib}(iA+B-iC)(c-id)^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f}$$
$$-\frac{\sqrt{a+ib}(B-i(A-C))(c+id)^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f}$$
$$+\frac{(a^3Cd^3-a^2bd^2(3cC+2Bd)+ab^2d(3c^2C+12Bcd+8(A-C)d^2)-b^3(c^3C-6Bc^2d-24c(A-C)d^2+16}{8b^{5/2}d^{3/2}f}$$
$$+\frac{(8b(Ab+aB-bC)d^2-(bc-ad)(bcC-6bBd-aCd))\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{8b^2df}$$
$$-\frac{(bcC-6bBd-aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{12bdf}$$
$$+\frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{3df}$$

```
[Out] 1/8*(a^3*C*d^3-a^2*b*d^2*(2*B*d+3*C*c)+a*b^2*d*(3*c^2*C+12*B*c*d+8*(A-C)*d^
2)-b^3*(c^3*C-6*B*c^2*d-24*c*(A-C)*d^2+16*B*d^3))*arctanh(d^(1/2)*(a+b*tan(
f*x+e))^(1/2)/b^(1/2)/(c+d*tan(f*x+e))^(1/2))/b^(5/2)/d^(3/2)/f-(I*A+B-I*C)
*(c-I*d)^(3/2)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(1/2)/(
c+d*tan(f*x+e))^(1/2))*(a-I*b)^(1/2)/f-(B-I*(A-C))*(c+I*d)^(3/2)*arctanh((c
+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))*(a
+I*b)^(1/2)/f+1/8*(8*b*(A*b+B*a-C*b)*d^2-(-a*d+b*c)*(-6*B*b*d-C*a*d+C*b*c))
*(a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)/b^2/d/f-1/12*(-6*B*b*d-C*a*d
+C*b*c)*(a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(3/2)/b/d/f+1/3*C*(a+b*tan(
f*x+e))^(1/2)*(c+d*tan(f*x+e))^(5/2)/d/f
```

Rubi [A] (verified)

Time = 8.48 (sec) , antiderivative size = 508, normalized size of antiderivative = 1.00, number of steps used = 15, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.163, Rules used = {3728, 3736, 6857, 65, 223, 212, 95, 214}

$$\int \sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\frac{(a^3Cd^3-a^2bd^2(2Bd+3cC)+ab^2d(8d^2(A-C)+12Bcd+3c^2C)-(b^3(-24cd^2(A}{8b^{5/2}d^{3/2}f}$$
$$-\frac{\sqrt{a-ib}(c-id)^{3/2}(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f}$$
$$-\frac{\sqrt{a+ib}(c+id)^{3/2}(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f}$$
$$+\frac{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}(8bd^2(aB+Ab-bC)-(bc-ad)(-aCd-6bBd+bcC))}{8b^2df}$$
$$-\frac{(-aCd-6bBd+bcC)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{12bdf}$$
$$+\frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{3df}$$

```
[In] Int[Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x]
 + C*Tan[e + f*x]^2),x]
```

```
[Out] -((Sqrt[a - I*b]*(I*A + B - I*C)*(c - I*d)^(3/2)*ArcTanh[(Sqrt[c - I*d]*Sqr
t[a + b*Tan[e + f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/f) - (Sqr
t[a + I*b]*(B - I*(A - C))*(c + I*d)^(3/2)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a +
b*Tan[e + f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/f + ((a^3*C*d^3
 - a^2*b*d^2*(3*c*C + 2*B*d) + a*b^2*d*(3*c^2*C + 12*B*c*d + 8*(A - C)*d^2)
 - b^3*(c^3*C - 6*B*c^2*d - 24*c*(A - C)*d^2 + 16*B*d^3))*ArcTanh[(Sqrt[d]*
Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])])/(8*b^(5/2)*d
^(3/2)*f) + ((8*b*(A*b + a*B - b*C)*d^2 - (b*c - a*d)*(b*c*C - 6*b*B*d - a*
C*d))*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(8*b^2*d*f) - ((b*
c*C - 6*b*B*d - a*C*d)*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(3/2))
/(12*b*d*f) + (C*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(5/2))/(3*d*
f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
```

```
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 95

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 223

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3736

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[(a + b*ff*x)^m*(c + d*ff*x)^n*((A + B*ff*x + C*ff^2*x^2
)/(1 + ff^2*x^2)), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
```

```
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6857

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

integral

$$= \frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{3df} + \frac{\int \frac{(c+d\tan(e+fx))^{3/2}\left(\frac{1}{2}(-bcC+a(6A-5C)d)+3(Ab+aB-bC)d\tan(e+fx)-\frac{1}{2}(bcC-6bBd-aCd)\tan^2(e+fx)\right)}{\sqrt{a+b\tan(e+fx)}}\,dx}{3d}$$

$$= -\frac{(bcC-6bBd-aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{12bdf} + \frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{3df} + \frac{\int \frac{\sqrt{c+d\tan(e+fx)}\left(-\frac{3}{4}(a^2Cd^2-2abd(4Ac-3cC-3Bd)+b^2c(cC+2Bd))+6bd(Abc+aBc-bcC+aAd-bBd-aCd)\tan(e+fx)+\frac{3}{4}(8b(Ab+aB\cdots\right.}{\sqrt{a+b\tan(e+fx)}}}{6bd}$$

$$= \frac{(8b(Ab+aB-bC)d^2-(bc-ad)(bcC-6bBd-aCd))\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{8b^2df} - \frac{(bcC-6bBd-aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{12bdf} + \frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{3df} + \frac{\int \frac{3}{8}(a^3Cd^3-a^2bd^2(3cC+2Bd)-b^3c(c^2C+10Bcd+8(A-C)d^2)-ab^2d(13c^2C+20Bcd-8Cd^2-8A(2c^2-d^2)))+6b^2d(2aAcd-2acCd+Ab(c^2-\cdots}{\cdots}$$

$$= \frac{(8b(Ab+aB-bC)d^2-(bc-ad)(bcC-6bBd-aCd))\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{8b^2df} - \frac{(bcC-6bBd-aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{12bdf} + \frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{3df} + \frac{\text{Subst}\left(\int \frac{3}{8}(a^3Cd^3-a^2bd^2(3cC+2Bd)-b^3c(c^2C+10Bcd+8(A-C)d^2)-ab^2d(13c^2C+20Bcd-8Cd^2-8A(2c^2-d^2)))+6b^2d(2aAcd-2acC\cdots\right.}{\cdots}$$

$$= \frac{(8b(Ab+aB-bC)d^2-(bc-ad)(bcC-6bBd-aCd))\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{8b^2df}$$
$$-\frac{(bcC-6bBd-aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{12bdf}$$
$$+\frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{3df}$$
$$+\frac{\text{Subst}\left(\int\left(\frac{3\left(a^3Cd^3-a^2bd^2(3cC+2Bd)+ab^2d\left(3c^2C+12Bcd+8(A-C)d^2\right)-b^3\left(c^3C-6Bc^2d-24c(A-C)d^2+16Bd^3\right)\right)}{8\sqrt{a+bx}\sqrt{c+dx}}+\frac{6\left(-b^2d\left(a\left(c^2C+2\right.\right.\right.}{}\right.\right.}{}$$

$$= \frac{(8b(Ab+aB-bC)d^2-(bc-ad)(bcC-6bBd-aCd))\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{8b^2df}$$
$$-\frac{(bcC-6bBd-aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{12bdf}$$
$$+\frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{3df}$$
$$+\frac{\text{Subst}\left(\int\frac{-b^2d\left(a\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)+b\left(2c(A-C)d+B\left(c^2-d^2\right)\right)\right)+b^2d\left(2aAcd-2acCd+Ab\left(c^2-d^2\right)+aB\left(c^2-d^2\right)-b\left(c^2C+2Bcd\right.\right.}{\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}\right.}{b^2df}$$
$$+\frac{(a^3Cd^3-a^2bd^2(3cC+2Bd)+ab^2d(3c^2C+12Bcd+8(A-C)d^2)-b^3(c^3C-6Bc^2d-24c(A-C)d^2+}{16b^2df}$$

$$= \frac{(8b(Ab+aB-bC)d^2-(bc-ad)(bcC-6bBd-aCd))\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{8b^2df}$$
$$-\frac{(bcC-6bBd-aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{12bdf}$$
$$+\frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{3df}$$
$$+\frac{\text{Subst}\left(\int\left(\frac{-b^2d\left(2aAcd-2acCd+Ab\left(c^2-d^2\right)+aB\left(c^2-d^2\right)-b\left(c^2C+2Bcd-Cd^2\right)\right)-ib^2d\left(a\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)+b\left(2c(A-C)d+B\right.\right.}{2(i-x)\sqrt{a+bx}\sqrt{c+dx}}\right.\right.}{}$$
$$+\frac{(a^3Cd^3-a^2bd^2(3cC+2Bd)+ab^2d(3c^2C+12Bcd+8(A-C)d^2)-b^3(c^3C-6Bc^2d-24c(A-C)d^2+}{8b^3df}$$

$$= \frac{(8b(Ab+aB-bC)d^2-(bc-ad)(bcC-6bBd-aCd))\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{8b^2df}$$
$$-\frac{(bcC-6bBd-aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{12bdf}$$
$$+\frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{3df}$$
$$+\frac{((ia+b)(A-iB-C)(c-id)^2)\operatorname{Subst}\left(\int\frac{1}{(i+x)\sqrt{a+bx}\sqrt{c+dx}}\,dx,x,\tan(e+fx)\right)}{2f}$$
$$+\frac{((ia-b)(A+iB-C)(c+id)^2)\operatorname{Subst}\left(\int\frac{1}{(i-x)\sqrt{a+bx}\sqrt{c+dx}}\,dx,x,\tan(e+fx)\right)}{2f}$$
$$+\frac{(a^3Cd^3-a^2bd^2(3cC+2Bd)+ab^2d(3c^2C+12Bcd+8(A-C)d^2)-b^3(c^3C-6Bc^2d-24c(A-C)d^2}{8b^3df}$$

$$= \frac{(a^3Cd^3-a^2bd^2(3cC+2Bd)+ab^2d(3c^2C+12Bcd+8(A-C)d^2)-b^3(c^3C-6Bc^2d-24c(A-C)d^2+}{8b^{5/2}d^{3/2}f}$$
$$+\frac{(8b(Ab+aB-bC)d^2-(bc-ad)(bcC-6bBd-aCd))\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{8b^2df}$$
$$-\frac{(bcC-6bBd-aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{12bdf}$$
$$+\frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{3df}$$
$$+\frac{((ia+b)(A-iB-C)(c-id)^2)\operatorname{Subst}\left(\int\frac{1}{-a+ib-(-c+id)x^2}\,dx,x,\frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{f}$$
$$+\frac{((ia-b)(A+iB-C)(c+id)^2)\operatorname{Subst}\left(\int\frac{1}{a+ib-(c+id)x^2}\,dx,x,\frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{f}$$

$$= -\frac{\sqrt{a-ib}(iA+B-iC)(c-id)^{3/2}\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f}$$
$$- \frac{\sqrt{a+ib}(B-i(A-C))(c+id)^{3/2}\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f}$$
$$+ \frac{(a^3Cd^3 - a^2bd^2(3cC+2Bd) + ab^2d(3c^2C+12Bcd+8(A-C)d^2) - b^3(c^3C-6Bc^2d-24c(A-C)d^2 +}{8b^{5/2}d^{3/2}f}$$
$$+ \frac{(8b(Ab+aB-bC)d^2-(bc-ad)(bcC-6bBd-aCd))\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{8b^2df}$$
$$- \frac{(bcC-6bBd-aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{12bdf}$$
$$+ \frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{3df}$$

Mathematica [A] (verified)

Time = 9.23 (sec) , antiderivative size = 867, normalized size of antiderivative = 1.71

$$\int \sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{3df}$$
$$+ \frac{\frac{(-bcC+6bBd+aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{4bf} + \frac{\frac{3\left(8b(Ab+aB-bC)d^2-(bc-ad)(bcC-6bBd-aCd)\right)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{4bf} + \frac{6b^2d(}{}}{}}{}$$

```
[In] Integrate[Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e
+ f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] (C*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(5/2))/(3*d*f) + (((-(b*c*
C) + 6*b*B*d + a*C*d)*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(3/2))/
(4*b*f) + ((3*(8*b*(A*b + a*B - b*C)*d^2 - (b*c - a*d)*(b*c*C - 6*b*B*d - a
*C*d))*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(4*b*f) + ((6*b^2
*d*(b*(2*a*A*c*d - 2*a*c*C*d + A*b*(c^2 - d^2) + a*B*(c^2 - d^2) - b*(c^2*C
 + 2*B*c*d - C*d^2)) - Sqrt[-b^2]*(a*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^
2)) + b*(2*c*(A - C)*d + B*(c^2 - d^2))))*ArcTanh[(Sqrt[-c + (Sqrt[-b^2]*d)
/b]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[c + d*Tan[e + f*x
]])])/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[-c + (Sqrt[-b^2]*d)/b]) - (6*b^2*d*(b*(2*
a*A*c*d - 2*a*c*C*d + A*b*(c^2 - d^2) + a*B*(c^2 - d^2) - b*(c^2*C + 2*B*c*
d - C*d^2)) + Sqrt[-b^2]*(a*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2)) + b*(
2*c*(A - C)*d + B*(c^2 - d^2))))*ArcTanh[(Sqrt[c + (Sqrt[-b^2]*d)/b]*Sqrt[a
```

```
 + b*Tan[e + f*x]])/(Sqrt[a + Sqrt[-b^2]]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt
[a + Sqrt[-b^2]]*Sqrt[c + (Sqrt[-b^2]*d)/b]) + (3*Sqrt[b]*Sqrt[c - (a*d)/b]
*(a^3*C*d^3 - a^2*b*d^2*(3*c*C + 2*B*d) + a*b^2*d*(3*c^2*C + 12*B*c*d + 8*(
A - C)*d^2) - b^3*(c^3*C - 6*B*c^2*d - 24*c*(A - C)*d^2 + 16*B*d^3))*ArcSin
h[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c - (a*d)/b])]*Sqrt[(b*c
 + b*d*Tan[e + f*x])/(b*c - a*d)])/(4*Sqrt[d]*Sqrt[c + d*Tan[e + f*x]]))/(b
^2*f))/(2*b))/(3*d)
```

Maple [F(-1)]

Timed out.

$$\int \sqrt{a+b\tan(fx+e)}\,(c+d\tan(fx+e))^{\frac{3}{2}}\left(A+B\tan(fx+e)+C\tan(fx+e)^{2}\right)dx$$

```
[In] int((a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x)
[Out] int((a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x)
```

Fricas [F(-1)]

Timed out.

$$\int \sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*t
an(f*x+e)^2),x, algorithm="fricas")
[Out] Timed out
```

Sympy [F]

$$\int \sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\int \sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{\frac{3}{2}}\left(A+B\tan(e+fx)+C\tan^{2}(e+fx)\right)\,dx$$

```
[In] integrate((a+b*tan(f*x+e))**(1/2)*(c+d*tan(f*x+e))**(3/2)*(A+B*tan(f*x+e)+C
*tan(f*x+e)**2),x)
[Out] Integral(sqrt(a + b*tan(e + f*x))*(c + d*tan(e + f*x))**(3/2)*(A + B*tan(e
 + f*x) + C*tan(e + f*x)**2), x)
```

Maxima [F]

$$\int \sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \int \left(C\tan\left(fx+e\right)^2+B\tan\left(fx+e\right)+A\right)\sqrt{b\tan\left(fx+e\right)+a}\left(d\tan\left(fx+e\right)+c\right)^{\frac{3}{2}}\,d$$

```
[In] integrate((a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*t
an(f*x+e)^2),x, algorithm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*sqrt(b*tan(f*x + e) + a)*
(d*tan(f*x + e) + c)^(3/2), x)
```

Giac [F(-1)]

Timed out.

$$\int \sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*t
an(f*x+e)^2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \int \sqrt{a+b\tan\left(e+f\,x\right)}\,\left(c+d\tan(e+f\,x)\right)^{3/2}\,\left(C\tan(e+f\,x)^2+B\tan(e+f\,x)+A\right)\,d$$

```
[In] int((a + b*tan(e + f*x))^(1/2)*(c + d*tan(e + f*x))^(3/2)*(A + B*tan(e + f*
x) + C*tan(e + f*x)^2),x)
```

```
[Out] int((a + b*tan(e + f*x))^(1/2)*(c + d*tan(e + f*x))^(3/2)*(A + B*tan(e + f*
x) + C*tan(e + f*x)^2), x)
```

3.137 $\int \frac{(c+d\tan(e+fx))^{3/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{a+b\tan(e+fx)}}\,dx$

Optimal result . . . 1933
Rubi [A] (verified) . . . 1934
Mathematica [A] (verified) . . . 1938
Maple [F(-1)] . . . 1939
Fricas [B] (verification not implemented) . . . 1939
Sympy [F] . . . 1939
Maxima [F] . . . 1940
Giac [F(-1)] . . . 1940
Mupad [F(-1)] . . . 1940

Optimal result

Integrand size = 49, antiderivative size = 384

$$\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{a+b\tan(e+fx)}}\,dx = -\frac{(iA+B-iC)(c-id)^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{a-ib}f} + \frac{(iA-B-iC)(c+id)^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{a+ib}f} + \frac{(3a^2Cd^2-2abd(3cC+2Bd)+b^2(3c^2C+12Bcd+8(A-C)d^2))\operatorname{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{4b^{5/2}\sqrt{d}f} + \frac{(3bcC+4bBd-3aCd)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{4b^2f} + \frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{2bf}$$

```
[Out] -(I*A+B-I*C)*(c-I*d)^(3/2)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-
I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/f/(a-I*b)^(1/2)+(I*A-B-I*C)*(c+I*d)^(3/2
)*arctanh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e
))^(1/2))/f/(a+I*b)^(1/2)+1/4*(3*a^2*C*d^2-2*a*b*d*(2*B*d+3*C*c)+b^2*(3*c^2
*C+12*B*c*d+8*(A-C)*d^2))*arctanh(d^(1/2)*(a+b*tan(f*x+e))^(1/2)/b^(1/2)/(c
+d*tan(f*x+e))^(1/2))/b^(5/2)/f/d^(1/2)+1/4*(4*B*b*d-3*C*a*d+3*C*b*c)*(a+b*
tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)/b^2/f+1/2*C*(a+b*tan(f*x+e))^(1/2)
*(c+d*tan(f*x+e))^(3/2)/b/f
```

Rubi [A] (verified)

Time = 5.01 (sec) , antiderivative size = 384, normalized size of antiderivative = 1.00, number of steps used = 14, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.163, Rules used = {3728, 3736, 6857, 65, 223, 212, 95, 214}

$$\int \frac{(c+d\tan(e+fx))^{3/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{a+b\tan(e+fx)}}\,dx = \frac{(3a^2Cd^2-2abd(2Bd+3cC)+b^2(8d^2}{}$$

$$-\frac{(c-id)^{3/2}(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f\sqrt{a-ib}}$$

$$+\frac{(c+id)^{3/2}(iA-B-iC)\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f\sqrt{a+ib}}$$

$$+\frac{(-3aCd+4bBd+3bcC)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{4b^2f}$$

$$+\frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{2bf}$$

```
[In] Int[((c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/Sq
rt[a + b*Tan[e + f*x]],x]
```

```
[Out] -(((I*A + B - I*C)*(c - I*d)^(3/2)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e
+ f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[a - I*b]*f)) + ((
I*A - B - I*C)*(c + I*d)^(3/2)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*
x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[a + I*b]*f) + ((3*a^2
*C*d^2 - 2*a*b*d*(3*c*C + 2*B*d) + b^2*(3*c^2*C + 12*B*c*d + 8*(A - C)*d^2)
)*ArcTanh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c + d*Tan[e + f*
x]])])/(4*b^(5/2)*Sqrt[d]*f) + ((3*b*c*C + 4*b*B*d - 3*a*C*d)*Sqrt[a + b*Ta
n[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(4*b^2*f) + (C*Sqrt[a + b*Tan[e + f*x
]]*(c + d*Tan[e + f*x])^(3/2))/(2*b*f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 95

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
```

```
&& LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 223

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3736

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[(a + b*ff*x)^m*(c + d*ff*x)^n*((A + B*ff*x + C*ff^2*x^2
)/(1 + ff^2*x^2)), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6857

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

integral

$$=\frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{2bf}+\frac{\int\frac{\sqrt{c+d\tan(e+fx)}\left(\frac{1}{2}(4Abc-C(bc+3ad))+2b(Bc+(A-C)d)\tan(e+fx)+\frac{1}{2}(3bcC+4bBd-3aCd)\tan^2(e+fx)\right)}{\sqrt{a+b\tan(e+fx)}}\,dx}{2b}$$

$$=\frac{(3bcC+4bBd-3aCd)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{4b^2f}+\frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{2bf}+\frac{\int\frac{\frac{1}{4}\left(8Ab^2c^2+3a^2Cd^2-2abd(3cC+2Bd)-b^2c(5cC+4Bd)\right)+2b^2\left(2c(A-C)d+B\left(c^2-d^2\right)\right)\tan(e+fx)+\frac{1}{4}\left(8b^2d(Bc+(A-C)d)+(bc-ad)(3bcC+4\right.}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}}{2b^2}$$

$$=\frac{(3bcC+4bBd-3aCd)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{4b^2f}+\frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{2bf}+\frac{\text{Subst}\left(\int\frac{\frac{1}{4}\left(8Ab^2c^2+3a^2Cd^2-2abd(3cC+2Bd)-b^2c(5cC+4Bd)\right)+2b^2\left(2c(A-C)d+B\left(c^2-d^2\right)\right)x+\frac{1}{4}\left(8b^2d(Bc+(A-C)d)+(bc-ad)(3bcC+4\right.}{\sqrt{a+bx}\sqrt{c+dx}\left(1+x^2\right)}\right.}{2b^2f}$$

$$=\frac{(3bcC+4bBd-3aCd)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{4b^2f}+\frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{2bf}+\frac{\text{Subst}\left(\int\left(\frac{3a^2Cd^2-2abd(3cC+2Bd)+b^2\left(3c^2C+12Bcd+8(A-C)d^2\right)}{4\sqrt{a+bx}\sqrt{c+dx}}+\frac{2\left(-b^2\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)+b^2\left(2c(A-C)d+B\left(c^2-d^2\right)\right.\right.}{\sqrt{a+bx}\sqrt{c+dx}\left(1+x^2\right)}\right.\right.}{2b^2f}$$

$$=\frac{(3bcC+4bBd-3aCd)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{4b^2f}+\frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{2bf}+\frac{\text{Subst}\left(\int\frac{-b^2\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)+b^2\left(2c(A-C)d+B\left(c^2-d^2\right)\right)x}{\sqrt{a+bx}\sqrt{c+dx}\left(1+x^2\right)}\,dx,x,\tan(e+fx)\right)}{b^2f}+\frac{\left(3a^2Cd^2-2abd(3cC+2Bd)+b^2\left(3c^2C+12Bcd+8(A-C)d^2\right)\right)\text{Subst}\left(\int\frac{1}{\sqrt{a+bx}\sqrt{c+dx}}\,dx,x,\tan(e+fx\right.}{8b^2f}$$

$$= \frac{(3bcC+4bBd-3aCd)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{4b^2f}$$
$$+ \frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{2bf}$$
$$+ \frac{\text{Subst}\left(\int\left(\frac{-ib^2(c^2C+2Bcd-Cd^2-A(c^2-d^2))-b^2(2c(A-C)d+B(c^2-d^2))}{2(i-x)\sqrt{a+bx}\sqrt{c+dx}}+\frac{-ib^2(c^2C+2Bcd-Cd^2-A(c^2-d^2))+b^2(2c(A-C)d+B(\cdots}{2(i+x)\sqrt{a+bx}\sqrt{c+dx}}\right.\right.}{b^2f}$$
$$+ \frac{(3a^2Cd^2-2abd(3cC+2Bd)+b^2(3c^2C+12Bcd+8(A-C)d^2))\,\text{Subst}\left(\int\frac{1}{\sqrt{c-\frac{ad}{b}+\frac{dx^2}{b}}}\,dx,x,\sqrt{a+b\,\text{ta}\cdots}\right.}{4b^3f}$$

$$= \frac{(3bcC+4bBd-3aCd)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{4b^2f}$$
$$+ \frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{2bf}$$
$$+ \frac{((iA+B-iC)(c-id)^2)\,\text{Subst}\left(\int\frac{1}{(i+x)\sqrt{a+bx}\sqrt{c+dx}}\,dx,x,\tan(e+fx)\right)}{2f}$$
$$+ \frac{(3a^2Cd^2-2abd(3cC+2Bd)+b^2(3c^2C+12Bcd+8(A-C)d^2))\,\text{Subst}\left(\int\frac{1}{1-\frac{dx^2}{b}}\,dx,x,\frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{4b^3f}$$
$$- \frac{(ib^2(c^2C+2Bcd-Cd^2-A(c^2-d^2))+b^2(2c(A-C)d+B(c^2-d^2)))\,\text{Subst}\left(\int\frac{1}{(i-x)\sqrt{a+bx}\sqrt{c+dx}}\,dx,x,\cdots\right.}{2b^2f}$$

$$= \frac{(3a^2Cd^2-2abd(3cC+2Bd)+b^2(3c^2C+12Bcd+8(A-C)d^2))\,\text{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{4b^{5/2}\sqrt{d}f}$$
$$+ \frac{(3bcC+4bBd-3aCd)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{4b^2f}$$
$$+ \frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{2bf}$$
$$+ \frac{((iA+B-iC)(c-id)^2)\,\text{Subst}\left(\int\frac{1}{-a+ib-(-c+id)x^2}\,dx,x,\frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{f}$$
$$- \frac{(ib^2(c^2C+2Bcd-Cd^2-A(c^2-d^2))+b^2(2c(A-C)d+B(c^2-d^2)))\,\text{Subst}\left(\int\frac{1}{a+ib-(c+id)x^2}\,dx,x,\frac{\sqrt{a\cdots}}{\sqrt{c\cdots}}\right.}{b^2f}$$

$$=-\frac{(iA+B-iC)(c-id)^{3/2}\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{a-ib}f}+\frac{(iA-B-iC)(c+id)^{3/2}\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{a+ib}f}+\frac{(3a^2Cd^2-2abd(3cC+2Bd)+b^2(3c^2C+12Bcd+8(A-C)d^2))\,\text{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{4b^{5/2}\sqrt{d}f}+\frac{(3bcC+4bBd-3aCd)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{4b^2f}+\frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{2bf}$$

Mathematica [A] (verified)

Time = 7.86 (sec) , antiderivative size = 613, normalized size of antiderivative = 1.60

$$\int\frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{a+b\tan(e+fx)}}\,dx=\frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+f\ldots}{2bf}+\frac{\frac{(3bcC+4bBd-3aCd)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{2bf}+\frac{-\frac{2b^2\left(\sqrt{-b^2}\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-b\left(2c(A-C)d+B\left(c^2-d^2\right)\right)\right)\text{arctanh}\left(\frac{\sqrt{-c+\ldots}}{\sqrt{-a\ldots}}\right.}{\sqrt{-a+\sqrt{-b^2}}\sqrt{-c+\frac{\sqrt{-b^2}d}{b}}}}{\ldots}}{\ldots}$$

```
[In] Integrate[((c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/Sqrt[a + b*Tan[e + f*x]],x]
```

```
[Out] (C*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(3/2))/(2*b*f) + (((3*b*c*
C + 4*b*B*d - 3*a*C*d)*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(
2*b*f) + ((-2*b^2*(Sqrt[-b^2]*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - b
*(2*c*(A - C)*d + B*(c^2 - d^2)))*ArcTanh[(Sqrt[-c + (Sqrt[-b^2]*d)/b]*Sqrt
[a + b*Tan[e + f*x]])/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[c + d*Tan[e + f*x]])])/(S
qrt[-a + Sqrt[-b^2]]*Sqrt[-c + (Sqrt[-b^2]*d)/b]) - (2*b^2*(Sqrt[-b^2]*(c^2
*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2)) + b*(2*c*(A - C)*d + B*(c^2 - d^2)))*
ArcTanh[(Sqrt[c + (Sqrt[-b^2]*d)/b]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + Sqr
t[-b^2]]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[a + Sqrt[-b^2]]*Sqrt[c + (Sqrt[-
b^2]*d)/b]) + (Sqrt[b]*Sqrt[c - (a*d)/b]*(3*a^2*C*d^2 - 2*a*b*d*(3*c*C + 2*
B*d) + b^2*(3*c^2*C + 12*B*c*d + 8*(A - C)*d^2))*ArcSinh[(Sqrt[d]*Sqrt[a +
b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c - (a*d)/b])]*Sqrt[(b*c + b*d*Tan[e + f*x])
/(b*c - a*d)])/(2*Sqrt[d]*Sqrt[c + d*Tan[e + f*x]]))/(b^2*f))/(2*b)
```

Maple [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(fx+e))^{\frac{3}{2}}\left(A+B\tan(fx+e)+C\tan(fx+e)^{2}\right)}{\sqrt{a+b\tan(fx+e)}}dx$$

```
[In] int((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(1/2),x)
```

```
[Out] int((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(1/2),x)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 57709 vs. $2(311) = 622$.

Time = 196.39 (sec) , antiderivative size = 115434, normalized size of antiderivative = 300.61

$$\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{a+b\tan(e+fx)}}\,dx = \text{Too large to display}$$

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(1/2),x, algorithm="fricas")
```

```
[Out] Too large to include
```

Sympy [F]

$$\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{a+b\tan(e+fx)}}\,dx = \int \frac{(c+d\tan(e+fx))^{\frac{3}{2}}\left(A+B\tan(\right.}{\sqrt{a+b\tan(e}}$$

```
[In] integrate((c+d*tan(f*x+e))**(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e))**(1/2),x)
```

```
[Out] Integral((c + d*tan(e + f*x))**(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)**
2)/sqrt(a + b*tan(e + f*x)), x)
```

Maxima [F]

$$\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{a+b\tan(e+fx)}}\,dx = \int \frac{\left(C\tan\left(fx+e\right)^2+B\tan\left(fx+e\right)\right.}{\sqrt{b\tan\left(fx+e\right.}}$$

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(1/2),x, algorithm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(d*tan(f*x + e) + c)^(3/2
)/sqrt(b*tan(f*x + e) + a), x)
```

Giac [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{a+b\tan(e+fx)}}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(1/2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{a+b\tan(e+fx)}}\,dx = \text{Hanged}$$

```
[In] int(((c + d*tan(e + f*x))^(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x))^(1/2),x)
```

```
[Out] \text{Hanged}
```

3.138 $\int \frac{(c+d\tan(e+fx))^{3/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{3/2}}\,dx$

Optimal result 1941
Rubi [A] (verified) 1942
Mathematica [B] (verified) 1946
Maple [F(-1)] 1948
Fricas [B] (verification not implemented) 1948
Sympy [F] 1948
Maxima [F(-1)] 1949
Giac [F(-1)] 1949
Mupad [F(-1)] 1949

Optimal result

Integrand size = 49, antiderivative size = 382

$$\int \frac{(c+d\tan(e+fx))^{3/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{3/2}}\,dx = -\frac{(iA+B-iC)(c-id)^{3/2}\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a-ib)^{3/2}f} - \frac{(B-i(A-C))(c+id)^{3/2}\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a+ib)^{3/2}f} + \frac{\sqrt{d}(3bcC+2bBd-3aCd)\text{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{b^{5/2}f} + \frac{(2Ab^2-2abB+3a^2C+b^2C)\,d\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{b^2\,(a^2+b^2)\,f} - \frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{3/2}}{b\,(a^2+b^2)\,f\sqrt{a+b\tan(e+fx)}}$$

```
[Out] -(I*A+B-I*C)*(c-I*d)^(3/2)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-
I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/(a-I*b)^(3/2)/f-(B-I*(A-C))*(c+I*d)^(3/2
)*arctanh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e
))^(1/2))/(a+I*b)^(3/2)/f+(2*B*b*d-3*C*a*d+3*C*b*c)*arctanh(d^(1/2)*(a+b*ta
n(f*x+e))^(1/2)/b^(1/2)/(c+d*tan(f*x+e))^(1/2))*d^(1/2)/b^(5/2)/f+(2*A*b^2-
2*B*a*b+3*C*a^2+C*b^2)*d*(a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)/b^2/
(a^2+b^2)/f-2*(A*b^2-a*(B*b-C*a))*(c+d*tan(f*x+e))^(3/2)/b/(a^2+b^2)/f/(a+b
*tan(f*x+e))^(1/2)
```

Rubi [A] (verified)

Time = 6.64 (sec) , antiderivative size = 382, normalized size of antiderivative = 1.00, number of steps used = 14, number of rules used = 9, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.184, Rules used = {3726, 3728, 3736, 6857, 65, 223, 212, 95, 214}

$$\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{3/2}}\,dx = -\frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{3/2}}{bf\,(a^2+b^2)\,\sqrt{a+b\tan(e+fx)}} + \frac{d(3a^2C-2abB+2Ab^2+b^2C)\,\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{b^2f\,(a^2+b^2)} - \frac{(c-id)^{3/2}(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(a-ib)^{3/2}} - \frac{(c+id)^{3/2}(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(a+ib)^{3/2}} + \frac{\sqrt{d}(-3aCd+2bBd+3bcC)\text{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{b^{5/2}f}$$

```
[In] Int[((c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a
 + b*Tan[e + f*x])^(3/2),x]
```

```
[Out] -(((I*A + B - I*C)*(c - I*d)^(3/2)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e
+ f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a - I*b)^(3/2)*f)) -
((B - I*(A - C))*(c + I*d)^(3/2)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a + I*b)^(3/2)*f) + (Sq
rt[d]*(3*b*c*C + 2*b*B*d - 3*a*C*d)*ArcTanh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x
]])/(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])])/(b^(5/2)*f) + ((2*A*b^2 - 2*a*b*B
+ 3*a^2*C + b^2*C)*d*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(b^
2*(a^2 + b^2)*f) - (2*(A*b^2 - a*(b*B - a*C))*(c + d*Tan[e + f*x])^(3/2))/(
b*(a^2 + b^2)*f*Sqrt[a + b*Tan[e + f*x]])
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 95

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
```

```
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 223

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3726

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*d^2 + c*(c*C - B*d))*(a + b*Tan[e +
 f*x])^m*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3736

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[(a + b*ff*x)^m*(c + d*ff*x)^n*((A + B*ff*x + C*ff^2*x^2
)/(1 + ff^2*x^2)), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6857

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\text{integral} = -\frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{3/2}}{b\,(a^2+b^2)\,f\sqrt{a+b\tan(e+fx)}} + \frac{2\int \frac{\sqrt{c+d\tan(e+fx)}\left(\frac{1}{2}((bB-aC)(bc-3ad)+Ab(ac+3bd))-\frac{1}{2}b((A-C)(bc-ad)-B(ac+bd))\tan(e+fx)+\frac{1}{2}\left(2Ab^2-2abB+3a^2C+b^2C\right)d\tan^2(\cdots}{\sqrt{a+b\tan(e+fx)}}}{b\,(a^2+b^2)}$$

$$= \frac{\left(2Ab^2-2abB+3a^2C+b^2C\right)d\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{b^2\,(a^2+b^2)\,f} - \frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{3/2}}{b\,(a^2+b^2)\,f\sqrt{a+b\tan(e+fx)}} + \frac{2\int \frac{\frac{1}{4}\left(-\left(\left(2Ab^2-2abB+3a^2C+b^2C\right)d(bc+ad)\right)+2bc((bB-aC)(bc-3ad)+Ab(ac+3bd))\right)+\frac{1}{2}b^2\left(2aAcd-2acCd-Ab\left(c^2-d^2\right)+aB\left(c^2\cdots\right.\right.}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}}{b^2\,(a^2+b^2)}$$

$$= \frac{\left(2Ab^2-2abB+3a^2C+b^2C\right)d\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{b^2\,(a^2+b^2)\,f} - \frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{3/2}}{b\,(a^2+b^2)\,f\sqrt{a+b\tan(e+fx)}} + \frac{2\text{Subst}\left(\int \frac{\frac{1}{4}\left(-\left(\left(2Ab^2-2abB+3a^2C+b^2C\right)d(bc+ad)\right)+2bc((bB-aC)(bc-3ad)+Ab(ac+3bd))\right)+\frac{1}{2}b^2\left(2aAcd-2acCd-Ab\left(c^2-d^2\cdots\right.\right.}{\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}\right.}{b^2\,(a^2+b^2)\,f}$$

$$= \frac{(2Ab^2 - 2abB + 3a^2C + b^2C)\, d\sqrt{a + b\tan(e + fx)}\sqrt{c + d\tan(e + fx)}}{b^2\,(a^2 + b^2)\, f} - \frac{2(Ab^2 - a(bB - aC))\,(c + d\tan(e + fx))^{3/2}}{b\,(a^2 + b^2)\, f\sqrt{a + b\tan(e + fx)}} + \frac{2\text{Subst}\left(\int\left(\frac{(a^2+b^2)d(3bcC+2bBd-3aCd)}{4\sqrt{a+bx}\sqrt{c+dx}} + \frac{-b^2(a(c^2C+2Bcd-Cd^2-A(c^2-d^2))-b(2c(A-C)d+B(c^2-d^2)))+b^2(2aAcd}{2\sqrt{a+bx}\sqrt{c+dx}(1+x}\right.\right.}{b^2\,(a^2 + b^2)\, f}$$

$$= \frac{(2Ab^2 - 2abB + 3a^2C + b^2C)\, d\sqrt{a + b\tan(e + fx)}\sqrt{c + d\tan(e + fx)}}{b^2\,(a^2 + b^2)\, f} - \frac{2(Ab^2 - a(bB - aC))\,(c + d\tan(e + fx))^{3/2}}{b\,(a^2 + b^2)\, f\sqrt{a + b\tan(e + fx)}} + \frac{\text{Subst}\left(\int\frac{-b^2(a(c^2C+2Bcd-Cd^2-A(c^2-d^2))-b(2c(A-C)d+B(c^2-d^2)))+b^2(2aAcd-2acCd-Ab(c^2-d^2)+aB(c^2-d^2)+b(c}{\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}\right.}{b^2\,(a^2 + b^2)\, f} + \frac{(d(3bcC + 2bBd - 3aCd))\text{Subst}\left(\int\frac{1}{\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e + fx)\right)}{2b^2 f}$$

$$= \frac{(2Ab^2 - 2abB + 3a^2C + b^2C)\, d\sqrt{a + b\tan(e + fx)}\sqrt{c + d\tan(e + fx)}}{b^2\,(a^2 + b^2)\, f} - \frac{2(Ab^2 - a(bB - aC))\,(c + d\tan(e + fx))^{3/2}}{b\,(a^2 + b^2)\, f\sqrt{a + b\tan(e + fx)}} + \frac{\text{Subst}\left(\int\left(\frac{-b^2(2aAcd-2acCd-Ab(c^2-d^2)+aB(c^2-d^2)+b(c^2C+2Bcd-Cd^2))-ib^2(a(c^2C+2Bcd-Cd^2-A(c^2-d^2))-b(2c(}{2(i-x)\sqrt{a+bx}\sqrt{c+dx}}\right.\right.}{} + \frac{(d(3bcC + 2bBd - 3aCd))\text{Subst}\left(\int\frac{1}{\sqrt{c-\frac{ad}{b}+\frac{dx^2}{b}}}\,dx, x, \sqrt{a + b\tan(e + fx)}\right)}{b^3 f}$$

$$= \frac{(2Ab^2 - 2abB + 3a^2C + b^2C)\, d\sqrt{a + b\tan(e + fx)}\sqrt{c + d\tan(e + fx)}}{b^2\,(a^2 + b^2)\, f} - \frac{2(Ab^2 - a(bB - aC))\,(c + d\tan(e + fx))^{3/2}}{b\,(a^2 + b^2)\, f\sqrt{a + b\tan(e + fx)}} + \frac{((iA + B - iC)(c - id)^2)\,\text{Subst}\left(\int\frac{1}{(i+x)\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e + fx)\right)}{2(a - ib)f} + \frac{(d(3bcC + 2bBd - 3aCd))\text{Subst}\left(\int\frac{1}{1-\frac{dx^2}{b}}\,dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{b^3 f} - \frac{(b^2(2aAcd - 2acCd - Ab(c^2 - d^2) + aB(c^2 - d^2) + b(c^2C + 2Bcd - Cd^2)) + ib^2(a(c^2C + 2Bc}{2b^2(}$$

$$= \frac{\sqrt{d}(3bcC+2bBd-3aCd)\text{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{b^{5/2}f} + \frac{(2Ab^2-2abB+3a^2C+b^2C)\,d\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{b^2(a^2+b^2)\,f} - \frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{3/2}}{b(a^2+b^2)\,f\sqrt{a+b\tan(e+fx)}} + \frac{((iA+B-iC)(c-id)^2)\,\text{Subst}\left(\int\frac{1}{-a+ib-(-c+id)x^2}\,dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{(a-ib)f} - \frac{(b^2(2aAcd-2acCd-Ab(c^2-d^2)+aB(c^2-d^2)+b(c^2C+2Bcd-Cd^2))+ib^2(a(c^2C+2Bcd}{b^2(a^2}$$

$$= -\frac{(iA+B-iC)(c-id)^{3/2}\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a-ib)^{3/2}f} + \frac{(iA-B-iC)(c+id)^{3/2}\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a+ib)^{3/2}f} + \frac{\sqrt{d}(3bcC+2bBd-3aCd)\text{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{b^{5/2}f} + \frac{(2Ab^2-2abB+3a^2C+b^2C)\,d\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{b^2(a^2+b^2)\,f} - \frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{3/2}}{b(a^2+b^2)\,f\sqrt{a+b\tan(e+fx)}}$$

Mathematica [B] (verified)

Both result and optimal contain complex but leaf count is larger than twice the leaf count of optimal. 1664 vs. $2(382) = 764$.

Time = 7.33 (sec) , antiderivative size = 1664, normalized size of antiderivative = 4.36

$$\int\frac{(c+d\tan(e+fx))^{3/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{3/2}}\,dx = \frac{C(c+d\tan(e+fx))^{3/2}}{bf\sqrt{a+b\tan(e+fx)}}$$

$$+ \frac{-\frac{2b(iA+B-iC)(-c+id)^{3/2}\text{arctanh}\left(\frac{\sqrt{-c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{-a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(-a+ib)^{3/2}f} + \frac{2b(iA-B-iC)(c+id)^{3/2}\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a+ib)^{3/2}f} - \frac{2b(A+iB-}{(a+}}{}$$

```
[In] Integrate[((c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/(a + b*Tan[e + f*x])^(3/2),x]
```

```
[Out] (C*(c + d*Tan[e + f*x])^(3/2))/(b*f*Sqrt[a + b*Tan[e + f*x]]) + ((-2*b*(I*A
 + B - I*C)*(-c + I*d)^(3/2)*ArcTanh[(Sqrt[-c + I*d]*Sqrt[a + b*Tan[e + f*x
]])/(Sqrt[-a + I*b]*Sqrt[c + d*Tan[e + f*x]])]])/((-a + I*b)^(3/2)*f) + (2*b
*(I*A - B - I*C)*(c + I*d)^(3/2)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])]])/((a + I*b)^(3/2)*f) - (2*
b*(A + I*B - C)*(I*c - d)*Sqrt[c + d*Tan[e + f*x]])/((a + I*b)*f*Sqrt[a + b
*Tan[e + f*x]]) + (2*b*(A - I*B - C)*(I*c + d)*Sqrt[c + d*Tan[e + f*x]])/((
a - I*b)*f*Sqrt[a + b*Tan[e + f*x]]) + (6*c*C*Sqrt[c + d*Tan[e + f*x]]*(1 +
 (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*
c - a*d))))^(3/2)*(1 - (Sqrt[b]*Sqrt[d]*ArcSinh[(Sqrt[b]*Sqrt[d]*Sqrt[a + b
*Tan[e + f*x]])/(Sqrt[b*c - a*d]*Sqrt[(b^2*c)/(b*c - a*d) - (a*b*d)/(b*c -
a*d)])]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b*c - a*d]*Sqrt[(b^2*c)/(b*c - a*d)
 - (a*b*d)/(b*c - a*d)]*Sqrt[1 + (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((
b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)))])))/(Sqrt[b/((b^2*c)/(b*c - a*d)
 - (a*b*d)/(b*c - a*d))]*f*Sqrt[a + b*Tan[e + f*x]]*Sqrt[(b*(c + d*Tan[e +
f*x]))/(b*c - a*d)]*(-1 - (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/
(b*c - a*d) - (a*b*d)/(b*c - a*d))))) + (4*B*d*Sqrt[c + d*Tan[e + f*x]]*(1
+ (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b
*c - a*d))))^(3/2)*(1 - (Sqrt[b]*Sqrt[d]*ArcSinh[(Sqrt[b]*Sqrt[d]*Sqrt[a +
b*Tan[e + f*x]])/(Sqrt[b*c - a*d]*Sqrt[(b^2*c)/(b*c - a*d) - (a*b*d)/(b*c -
 a*d)])]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b*c - a*d]*Sqrt[(b^2*c)/(b*c - a*d
) - (a*b*d)/(b*c - a*d)]*Sqrt[1 + (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*(
(b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)))])))/(Sqrt[b/((b^2*c)/(b*c - a*d
) - (a*b*d)/(b*c - a*d))]*f*Sqrt[a + b*Tan[e + f*x]]*Sqrt[(b*(c + d*Tan[e +
 f*x]))/(b*c - a*d)]*(-1 - (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)
/(b*c - a*d) - (a*b*d)/(b*c - a*d))))) - (6*a*C*d*Sqrt[c + d*Tan[e + f*x]]*
(1 + (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)
/(b*c - a*d))))^(3/2)*(1 - (Sqrt[b]*Sqrt[d]*ArcSinh[(Sqrt[b]*Sqrt[d]*Sqrt[a
 + b*Tan[e + f*x]])/(Sqrt[b*c - a*d]*Sqrt[(b^2*c)/(b*c - a*d) - (a*b*d)/(b*
c - a*d)])]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b*c - a*d]*Sqrt[(b^2*c)/(b*c -
a*d) - (a*b*d)/(b*c - a*d)]*Sqrt[1 + (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d
)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)))])))/(b*Sqrt[b/((b^2*c)/(b*c
- a*d) - (a*b*d)/(b*c - a*d))]*f*Sqrt[a + b*Tan[e + f*x]]*Sqrt[(b*(c + d*Ta
n[e + f*x]))/(b*c - a*d)]*(-1 - (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b
^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d))))))/(2*b)
```

Maple [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(fx+e))^{\frac{3}{2}}\left(A+B\tan(fx+e)+C\tan(fx+e)^{2}\right)}{(a+b\tan(fx+e))^{\frac{3}{2}}}dx$$

```
[In] int((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(3/2),x)
```

```
[Out] int((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(3/2),x)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 103394 vs. $2(315) = 630$.

Time = 289.27 (sec) , antiderivative size = 206814, normalized size of antiderivative = 541.40

$$\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{3/2}}\,dx = \text{Too large to display}$$

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(3/2),x, algorithm="fricas")
```

```
[Out] Too large to include
```

Sympy [F]

$$\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{3/2}}\,dx = \int \frac{(c+d\tan(e+fx))^{\frac{3}{2}}\left(A+B\tan(e\right.}{(a+b\tan(e+}$$

```
[In] integrate((c+d*tan(f*x+e))**(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e))**(3/2),x)
```

```
[Out] Integral((c + d*tan(e + f*x))**(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)**
2)/(a + b*tan(e + f*x))**(3/2), x)
```

Maxima [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(3/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

Giac [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(3/2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{3/2}}\,dx = \text{Hanged}$$

```
[In] int(((c + d*tan(e + f*x))^(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x))^(3/2),x)
```

```
[Out] \text{Hanged}
```

3.139 $\int \frac{(c+d\tan(e+fx))^{3/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{5/2}}\,dx$

Optimal result 1950
Rubi [A] (verified) 1951
Mathematica [C] (verified) 1955
Maple [F(-1)] 1956
Fricas [F(-1)] 1957
Sympy [F] 1957
Maxima [F(-1)] 1957
Giac [F(-1)] 1957
Mupad [F(-1)] 1958

Optimal result

Integrand size = 49, antiderivative size = 402

$$\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{5/2}}\,dx = -\frac{(iA+B-iC)(c-id)^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a-ib)^{5/2}f} - \frac{(B-i(A-C))(c+id)^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a+ib)^{5/2}f} + \frac{2Cd^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{b^{5/2}f} - \frac{2(a^4Cd+b^4(Bc+Ad)+2ab^3(Ac-cC-Bd)-a^2b^2(Bc+(A-3C)d))\sqrt{c+d\tan(e+fx)}}{b^2(a^2+b^2)^2f\sqrt{a+b\tan(e+fx)}} - \frac{2(Ab^2-a(bB-aC))(c+d\tan(e+fx))^{3/2}}{3b(a^2+b^2)f(a+b\tan(e+fx))^{3/2}}$$

```
[Out] -(I*A+B-I*C)*(c-I*d)^(3/2)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-
I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/(a-I*b)^(5/2)/f-(B-I*(A-C))*(c+I*d)^(3/2
)*arctanh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e
))^(1/2))/(a+I*b)^(5/2)/f+2*C*d^(3/2)*arctanh(d^(1/2)*(a+b*tan(f*x+e))^(1/2
)/b^(1/2)/(c+d*tan(f*x+e))^(1/2))/b^(5/2)/f-2*(a^4*C*d+b^4*(A*d+B*c)+2*a*b^
3*(A*c-B*d-C*c)-a^2*b^2*(B*c+(A-3*C)*d))*(c+d*tan(f*x+e))^(1/2)/b^2/(a^2+b^
2)^2/f/(a+b*tan(f*x+e))^(1/2)-2/3*(A*b^2-a*(B*b-C*a))*(c+d*tan(f*x+e))^(3/2
)/b/(a^2+b^2)/f/(a+b*tan(f*x+e))^(3/2)
```

Rubi [A] (verified)

Time = 8.31 (sec) , antiderivative size = 402, normalized size of antiderivative = 1.00, number of steps used = 14, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.163, Rules used = {3726, 3736, 6857, 65, 223, 212, 95, 214}

$$\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{5/2}}\,dx = -\frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{3/2}}{3bf\,(a^2+b^2)\,(a+b\tan(e+fx))^{3/2}} - \frac{2\sqrt{c+d\tan(e+fx)}(a^4Cd-a^2b^2(d(A-3C)+Bc)+2ab^3(Ac-Bd-cC)+b^4(Ad+Bc))}{b^2f\,(a^2+b^2)^2\,\sqrt{a+b\tan(e+fx)}} - \frac{(c-id)^{3/2}(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(a-ib)^{5/2}} - \frac{(c+id)^{3/2}(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(a+ib)^{5/2}} + \frac{2Cd^{3/2}\text{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{b^{5/2}f}$$

```
[In] Int[((c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a
 + b*Tan[e + f*x])^(5/2),x]
```

```
[Out] -(((I*A + B - I*C)*(c - I*d)^(3/2)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e
+ f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a - I*b)^(5/2)*f)) -
((B - I*(A - C))*(c + I*d)^(3/2)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a + I*b)^(5/2)*f) + (2*
C*d^(3/2)*ArcTanh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c + d*Ta
n[e + f*x]])])/(b^(5/2)*f) - (2*(a^4*C*d + b^4*(B*c + A*d) + 2*a*b^3*(A*c -
 c*C - B*d) - a^2*b^2*(B*c + (A - 3*C)*d))*Sqrt[c + d*Tan[e + f*x]])/(b^2*(
a^2 + b^2)^2*f*Sqrt[a + b*Tan[e + f*x]]) - (2*(A*b^2 - a*(b*B - a*C))*(c +
d*Tan[e + f*x])^(3/2))/(3*b*(a^2 + b^2)*f*(a + b*Tan[e + f*x])^(3/2))
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 95

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
```

```
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 223

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3726

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*d^2 + c*(c*C - B*d))*(a + b*Tan[e +
 f*x])^m*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3736

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[(a + b*ff*x)^m*(c + d*ff*x)^n*((A + B*ff*x + C*ff^2*x^2
)/(1 + ff^2*x^2)), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6857

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
```

`[n, 0]`

Rubi steps

$$\text{integral} = -\frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{3/2}}{3b\,(a^2+b^2)\,f(a+b\tan(e+fx))^{3/2}} + \frac{2\int \frac{\sqrt{c+d\tan(e+fx)}\left(\frac{3}{2}((bB-aC)(bc-ad)+Ab(ac+bd))-\frac{3}{2}b((A-C)(bc-ad)-B(ac+bd))\tan(e+fx)+\frac{3}{2}(a^2+b^2)Cd\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{3/2}}\,dx}{3b\,(a^2+b^2)}$$

$$= -\frac{2(a^4Cd+b^4(Bc+Ad)+2ab^3(Ac-cC-Bd)-a^2b^2(Bc+(A-3C)d))\sqrt{c+d\tan(e+fx)}}{b^2\,(a^2+b^2)^2\,f\sqrt{a+b\tan(e+fx)}} - \frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{3/2}}{3b\,(a^2+b^2)\,f(a+b\tan(e+fx))^{3/2}} + \frac{4\int \frac{\frac{3}{4}\left(b(ac+bd)\left(a^2Cd+b^2(Bc+Ad)+ab(Ac-cC-Bd)\right)-(bc-ad)\left(a^3Cd+Ab^2(bc-ad)-b^3(cC+Bd)-ab^2(Bc-2Cd)\right)\right)-\frac{3}{4}b^2((ac+}{\sqrt{a+b\tan(e+fx)}\sqrt{c}}}{3b^2\,(a^2+}$$

$$= -\frac{2(a^4Cd+b^4(Bc+Ad)+2ab^3(Ac-cC-Bd)-a^2b^2(Bc+(A-3C)d))\sqrt{c+d\tan(e+fx)}}{b^2\,(a^2+b^2)^2\,f\sqrt{a+b\tan(e+fx)}} - \frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{3/2}}{3b\,(a^2+b^2)\,f(a+b\tan(e+fx))^{3/2}} + \frac{4\text{Subst}\left(\int \frac{\frac{3}{4}\left(b(ac+bd)\left(a^2Cd+b^2(Bc+Ad)+ab(Ac-cC-Bd)\right)-(bc-ad)\left(a^3Cd+Ab^2(bc-ad)-b^3(cC+Bd)-ab^2(Bc-2Cd)\right)\right)-}{\sqrt{a+bx}\sqrt{c+dx}}\right.}{3b^2\,(a}$$

$$= -\frac{2(a^4Cd+b^4(Bc+Ad)+2ab^3(Ac-cC-Bd)-a^2b^2(Bc+(A-3C)d))\sqrt{c+d\tan(e+fx)}}{b^2\,(a^2+b^2)^2\,f\sqrt{a+b\tan(e+fx)}} - \frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{3/2}}{3b\,(a^2+b^2)\,f(a+b\tan(e+fx))^{3/2}} + \frac{4\text{Subst}\left(\int \left(\frac{3(a^2+b^2)^2Cd^2}{4\sqrt{a+bx}\sqrt{c+dx}} + \frac{3\left(-b^2\left(a^2\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-b^2\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-2ab\left(2c(A-C)d\right.\right.\right.}{}\right.\right.}{3b}$$

$$= -\frac{2(a^4Cd+b^4(Bc+Ad)+2ab^3(Ac-cC-Bd)-a^2b^2(Bc+(A-3C)d))\sqrt{c+d\tan(e+fx)}}{b^2(a^2+b^2)^2f\sqrt{a+b\tan(e+fx)}} - \frac{2(Ab^2-a(bB-aC))(c+d\tan(e+fx))^{3/2}}{3b(a^2+b^2)f(a+b\tan(e+fx))^{3/2}} + \frac{\text{Subst}\left(\int \frac{-b^2(a^2(c^2C+2Bcd-Cd^2-A(c^2-d^2))-b^2(c^2C+2Bcd-Cd^2-A(c^2-d^2))-2ab(2c(A-C)d+B(c^2-d^2)))+b^2(2ab(c^2C\cdots}{\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}\cdots\right)}{b^2(a^2+b^2)^2f\cdots} + \frac{(Cd^2)\,\text{Subst}\left(\int \frac{1}{\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e+fx)\right)}{b^2f}$$

$$= -\frac{2(a^4Cd+b^4(Bc+Ad)+2ab^3(Ac-cC-Bd)-a^2b^2(Bc+(A-3C)d))\sqrt{c+d\tan(e+fx)}}{b^2(a^2+b^2)^2f\sqrt{a+b\tan(e+fx)}} - \frac{2(Ab^2-a(bB-aC))(c+d\tan(e+fx))^{3/2}}{3b(a^2+b^2)f(a+b\tan(e+fx))^{3/2}} + \frac{\text{Subst}\left(\int \left(\frac{-ib^2(a^2(c^2C+2Bcd-Cd^2-A(c^2-d^2))-b^2(c^2C+2Bcd-Cd^2-A(c^2-d^2))-2ab(2c(A-C)d+B(c^2-d^2)))-b^2(2ab(c^2\cdots}{2(i-x)\sqrt{a+bx}\sqrt{c+dx}}\cdots\right)\right)}{\cdots} + \frac{(2Cd^2)\,\text{Subst}\left(\int \frac{1}{\sqrt{c-\frac{ad}{b}+\frac{dx^2}{b}}}\,dx, x, \sqrt{a+b\tan(e+fx)}\right)}{b^3f}$$

$$= -\frac{2(a^4Cd+b^4(Bc+Ad)+2ab^3(Ac-cC-Bd)-a^2b^2(Bc+(A-3C)d))\sqrt{c+d\tan(e+fx)}}{b^2(a^2+b^2)^2f\sqrt{a+b\tan(e+fx)}} - \frac{2(Ab^2-a(bB-aC))(c+d\tan(e+fx))^{3/2}}{3b(a^2+b^2)f(a+b\tan(e+fx))^{3/2}} + \frac{((iA+B-iC)(c-id)^2)\,\text{Subst}\left(\int \frac{1}{(i+x)\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e+fx)\right)}{2(a-ib)^2f} + \frac{(2Cd^2)\,\text{Subst}\left(\int \frac{1}{1-\frac{dx^2}{b}}\,dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{b^3f} - \frac{(ib^2(a^2(c^2C+2Bcd-Cd^2-A(c^2-d^2))-b^2(c^2C+2Bcd-Cd^2-A(c^2-d^2))-2ab(2c(A-\cdots}{\cdots}$$

$$= \frac{2Cd^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{b^{5/2}f} - \frac{2(a^4Cd+b^4(Bc+Ad)+2ab^3(Ac-cC-Bd)-a^2b^2(Bc+(A-3C)d))\sqrt{c+d\tan(e+fx)}}{b^2(a^2+b^2)^2f\sqrt{a+b\tan(e+fx)}} - \frac{2(Ab^2-a(bB-aC))(c+d\tan(e+fx))^{3/2}}{3b(a^2+b^2)f(a+b\tan(e+fx))^{3/2}} + \frac{((iA+B-iC)(c-id)^2)\operatorname{Subst}\left(\int\frac{1}{-a+ib-(-c+id)x^2}dx,x,\frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{(a-ib)^2f} - \frac{(ib^2(a^2(c^2C+2Bcd-Cd^2-A(c^2-d^2))-b^2(c^2C+2Bcd-Cd^2-A(c^2-d^2))-2ab(2c(A-}{}$$

$$= -\frac{(iA+B-iC)(c-id)^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a-ib)^{5/2}f} + \frac{(iA-B-iC)(c+id)^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a+ib)^{5/2}f} + \frac{2Cd^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{b^{5/2}f} - \frac{2(a^4Cd+b^4(Bc+Ad)+2ab^3(Ac-cC-Bd)-a^2b^2(Bc+(A-3C)d))\sqrt{c+d\tan(e+fx)}}{b^2(a^2+b^2)^2f\sqrt{a+b\tan(e+fx)}} - \frac{2(Ab^2-a(bB-aC))(c+d\tan(e+fx))^{3/2}}{3b(a^2+b^2)f(a+b\tan(e+fx))^{3/2}}$$

Mathematica [C] (verified)

Result contains higher order function than in optimal. Order 5 vs. order 3 in optimal.

Time = 6.76 (sec) , antiderivative size = 519, normalized size of antiderivative = 1.29

$$\int \frac{(c+d\tan(e+fx))^{3/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{5/2}}\,dx = \frac{(B+i(A-C))(c+d\tan(e+fx))^{3/2}}{3(a-ib)f(a+b\tan(e+fx))^{3/2}} - \frac{(iA-B-iC)(c+d\tan(e+fx))^{3/2}}{3(a+ib)f(a+b\tan(e+fx))^{3/2}} - \frac{2C(bc-ad)\,\text{Hypergeometric2F1}\left(-\frac{3}{2},-\frac{3}{2},-\frac{1}{2},-\frac{d(a+b\tan(e+fx))}{bc-ad}\right)\sqrt{c+d\tan(e+fx)}}{3b^2f(a+b\tan(e+fx))^{3/2}\sqrt{\frac{b(c+d\tan(e+fx))}{bc-ad}}} + \frac{(A-iB-C)(ic+d)\left(\frac{\sqrt{-c+id}\,\text{arctanh}\left(\frac{\sqrt{-c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{-a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(-a+ib)^{3/2}}+\frac{\sqrt{c+d\tan(e+fx)}}{(a-ib)\sqrt{a+b\tan(e+fx)}}\right)}{(a-ib)f} + \frac{(A+iB-C)(ic-d)\left(\frac{\sqrt{c+id}\,\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a+ib)^{3/2}}-\frac{\sqrt{c+d\tan(e+fx)}}{(a+ib)\sqrt{a+b\tan(e+fx)}}\right)}{(a+ib)f}$$

```
[In] Integrate[((c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/(a + b*Tan[e + f*x])^(5/2),x]
```

```
[Out] ((B + I*(A - C))*(c + d*Tan[e + f*x])^(3/2))/(3*(a - I*b)*f*(a + b*Tan[e +
f*x])^(3/2)) - ((I*A - B - I*C)*(c + d*Tan[e + f*x])^(3/2))/(3*(a + I*b)*f*
(a + b*Tan[e + f*x])^(3/2)) - (2*C*(b*c - a*d)*Hypergeometric2F1[-3/2, -3/2
, -1/2, -((d*(a + b*Tan[e + f*x]))/(b*c - a*d))]*Sqrt[c + d*Tan[e + f*x]])/
(3*b^2*f*(a + b*Tan[e + f*x])^(3/2)*Sqrt[(b*(c + d*Tan[e + f*x]))/(b*c - a*
d)]) + ((A - I*B - C)*(I*c + d)*((Sqrt[-c + I*d]*ArcTanh[(Sqrt[-c + I*d]*Sq
rt[a + b*Tan[e + f*x]])/(Sqrt[-a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(-a + I
*b)^(3/2) + Sqrt[c + d*Tan[e + f*x]]/((a - I*b)*Sqrt[a + b*Tan[e + f*x]])))
/((a - I*b)*f) + ((A + I*B - C)*(I*c - d)*((Sqrt[c + I*d]*ArcTanh[(Sqrt[c +
 I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/
(a + I*b)^(3/2) - Sqrt[c + d*Tan[e + f*x]]/((a + I*b)*Sqrt[a + b*Tan[e + f*
x]])))/((a + I*b)*f)
```

Maple [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(fx+e))^{\frac{3}{2}}\left(A+B\tan(fx+e)+C\tan(fx+e)^2\right)}{(a+b\tan(fx+e))^{\frac{5}{2}}}dx$$

```
[In] int((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(5/2),x)
```

```
[Out] int((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(5/2),x)
```

Fricas [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{5/2}}\,dx=\text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(5/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{5/2}}\,dx=\int \frac{(c+d\tan(e+fx))^{\frac{3}{2}}\left(A+B\tan(\right.}{(a+b\tan(e+}$$

```
[In] integrate((c+d*tan(f*x+e))**(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e))**(5/2),x)
```

```
[Out] Integral((c + d*tan(e + f*x))**(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)**
2)/(a + b*tan(e + f*x))**(5/2), x)
```

Maxima [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{5/2}}\,dx=\text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(5/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

Giac [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{5/2}}\,dx=\text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(5/2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{3/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{5/2}}\,dx = \text{Hanged}$$

```
[In] int(((c + d*tan(e + f*x))^(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x))^(5/2),x)
```

```
[Out] \text{Hanged}
```

3.140 $\int \frac{(c+d\tan(e+fx))^{3/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{7/2}}\,dx$

Optimal result 1959
Rubi [A] (verified) 1960
Mathematica [B] (verified) 1964
Maple [F(-1)] 1965
Fricas [F(-1)] 1966
Sympy [F] 1966
Maxima [F(-2)] 1966
Giac [F(-1)] 1967
Mupad [F(-1)] 1967

Optimal result

Integrand size = 49, antiderivative size = 586

$$\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{7/2}}\,dx =$$

$$-\frac{(iA+B-iC)(c-id)^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a-ib)^{7/2}f}$$

$$-\frac{(B-i(A-C))(c+id)^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a+ib)^{7/2}f}$$

$$-\frac{2(2a^3bBd+3a^4Cd+b^4(5Bc+3Ad)+2ab^3(5Ac-5cC-4Bd)-a^2b^2(5Bc+7Ad-13Cd))\sqrt{c+d\tan}}{15b^2(a^2+b^2)^2f(a+b\tan(e+fx))^{3/2}}$$

$$-\frac{2(2a^5bBd^2+3a^6Cd^2+a^4b^2d(10Bc+(8A+C)d)+a^2b^4(45Ac^2-45c^2C-90Bcd-49Ad^2+58Cd^2)-}{15b^2(a^2}$$

$$-\frac{2(Ab^2-a(bB-aC))(c+d\tan(e+fx))^{3/2}}{5b(a^2+b^2)f(a+b\tan(e+fx))^{5/2}}$$

```
[Out] -(I*A+B-I*C)*(c-I*d)^(3/2)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-
I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/(a-I*b)^(7/2)/f-(B-I*(A-C))*(c+I*d)^(3/2
)*arctanh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e
))^(1/2))/(a+I*b)^(7/2)/f-2/15*(2*a^5*b*B*d^2+3*a^6*C*d^2+a^4*b^2*d*(10*B*c
+(8*A+C)*d)+a^2*b^4*(45*A*c^2-49*A*d^2-90*B*c*d-45*C*c^2+58*C*d^2)-a^3*b^3*
(50*c*(A-C)*d+B*(15*c^2-39*d^2))+a*b^5*(70*c*(A-C)*d+B*(45*c^2-23*d^2))+b^6
*(5*c*(4*B*d+3*C*c)-3*A*(5*c^2-d^2)))*(c+d*tan(f*x+e))^(1/2)/b^2/(a^2+b^2)^
3/(-a*d+b*c)/f/(a+b*tan(f*x+e))^(1/2)-2/15*(2*a^3*b*B*d+3*a^4*C*d+b^4*(3*A*
d+5*B*c)+2*a*b^3*(5*A*c-4*B*d-5*C*c)-a^2*b^2*(7*A*d+5*B*c-13*C*d))*(c+d*tan
(f*x+e))^(1/2)/b^2/(a^2+b^2)^2/f/(a+b*tan(f*x+e))^(3/2)-2/5*(A*b^2-a*(B*b-C
*a))*(c+d*tan(f*x+e))^(3/2)/b/(a^2+b^2)/f/(a+b*tan(f*x+e))^(5/2)
```

Rubi [A] (verified)

Time = 4.27 (sec) , antiderivative size = 586, normalized size of antiderivative = 1.00, number of steps used = 10, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.122, Rules used = {3726, 3730, 3697, 3696, 95, 214}

$$\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{7/2}}\,dx =$$
$$-\frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{3/2}}{5bf\,(a^2+b^2)\,(a+b\tan(e+fx))^{5/2}}$$
$$-\frac{2\sqrt{c+d\tan(e+fx)}(3a^4Cd+2a^3bBd-a^2b^2(7Ad+5Bc-13Cd)+2ab^3(5Ac-4Bd-5cC)+b^4(3Ad+}{15b^2f\,(a^2+b^2)^2\,(a+b\tan(e+fx))^{3/2}}$$
$$-\frac{2\sqrt{c+d\tan(e+fx)}(3a^6Cd^2+2a^5bBd^2+a^4b^2d(d(8A+C)+10Bc)-a^3b^3(50cd(A-C)+B(15c^2-39}{15b^2f\,(a^2}$$
$$-\frac{(c-id)^{3/2}(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(a-ib)^{7/2}}$$
$$-\frac{(c+id)^{3/2}(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(a+ib)^{7/2}}$$

```
[In] Int[((c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a
 + b*Tan[e + f*x])^(7/2),x]
```

```
[Out] -(((I*A + B - I*C)*(c - I*d)^(3/2)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e
+ f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a - I*b)^(7/2)*f)) -
((B - I*(A - C))*(c + I*d)^(3/2)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a + I*b)^(7/2)*f) - (2*
(2*a^3*b*B*d + 3*a^4*C*d + b^4*(5*B*c + 3*A*d) + 2*a*b^3*(5*A*c - 5*c*C - 4
*B*d) - a^2*b^2*(5*B*c + 7*A*d - 13*C*d))*Sqrt[c + d*Tan[e + f*x]])/(15*b^2
*(a^2 + b^2)^2*f*(a + b*Tan[e + f*x])^(3/2)) - (2*(2*a^5*b*B*d^2 + 3*a^6*C*
d^2 + a^4*b^2*d*(10*B*c + (8*A + C)*d) + a^2*b^4*(45*A*c^2 - 45*c^2*C - 90*
B*c*d - 49*A*d^2 + 58*C*d^2) - a^3*b^3*(50*c*(A - C)*d + B*(15*c^2 - 39*d^2
)) + a*b^5*(70*c*(A - C)*d + B*(45*c^2 - 23*d^2)) + b^6*(5*c*(3*c*C + 4*B*d
) - 3*A*(5*c^2 - d^2)))*Sqrt[c + d*Tan[e + f*x]])/(15*b^2*(a^2 + b^2)^3*(b*
c - a*d)*f*Sqrt[a + b*Tan[e + f*x]]) - (2*(A*b^2 - a*(b*B - a*C))*(c + d*Ta
n[e + f*x])^(3/2))/(5*b*(a^2 + b^2)*f*(a + b*Tan[e + f*x])^(5/2))
```

Rule 95

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3696

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[A^2/f, Subst[Int[(a + b*x)^m*((c + d*x)^n/(A - B*x)), x], x, Tan[e + f*x
]], x] /; FreeQ[{a, b, c, d, e, f, A, B, m, n}, x] && NeQ[b*c - a*d, 0] &&
NeQ[a^2 + b^2, 0] && EqQ[A^2 + B^2, 0]
```

Rule 3697

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[(A + I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^n*(1 - I*Ta
n[e + f*x]), x], x] + Dist[(A - I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*T
an[e + f*x])^n*(1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A,
B, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[A^2 + B^2, 0]
```

Rule 3726

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*d^2 + c*(c*C - B*d))*(a + b*Tan[e +
 f*x])^m*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3730

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 - a*(b*B - a*C))*(a + b*Tan[e +
 f*x])^(m + 1)*((c + d*Tan[e + f*x])^(n + 1)/(f*(m + 1)*(b*c - a*d)*(a^2 +
b^2))), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
```

```
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rubi steps

$$\text{integral} = -\frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{3/2}}{5b\,(a^2+b^2)\,f(a+b\tan(e+fx))^{5/2}} + \frac{2\int \frac{\sqrt{c+d\tan(e+fx)}\left(\frac{1}{2}((bB-aC)(5bc-3ad)+Ab(5ac+3bd))-\frac{5}{2}b((A-C)(bc-ad)-B(ac+bd))\tan(e+fx)-\frac{1}{2}(2Ab^2-2abB-3a^2C-5b^2C)d\tan\ldots}{(a+b\tan(e+fx))^{5/2}}}{5b\,(a^2+b^2)}$$

$$= -\frac{2(2a^3bBd+3a^4Cd+b^4(5Bc+3Ad)+2ab^3(5Ac-5cC-4Bd)-a^2b^2(5Bc+7Ad-13Cd))\,\sqrt{\ldots}}{15b^2\,(a^2+b^2)^2\,f(a+b\tan(e+fx))^{3/2}} - \frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{3/2}}{5b\,(a^2+b^2)\,f(a+b\tan(e+fx))^{5/2}} + \frac{4\int \frac{1}{4}(b(3ac+bd)((bB-aC)(5bc-3ad)+Ab(5ac+3bd))-(3bc-ad)(2a^2bBd+3a^3Cd+Ab^2(5bc-7ad)-5b^3(cC+Bd)-5ab^2(Bc-2Cd))\ldots}{\ldots}$$

$$= -\frac{2(2a^3bBd+3a^4Cd+b^4(5Bc+3Ad)+2ab^3(5Ac-5cC-4Bd)-a^2b^2(5Bc+7Ad-13Cd))\,\sqrt{\ldots}}{15b^2\,(a^2+b^2)^2\,f(a+b\tan(e+fx))^{3/2}} - \frac{2(2a^5bBd^2+3a^6Cd^2+a^4b^2d(10Bc+(8A+C)d)+a^2b^4(45Ac^2-45c^2C-90Bcd-49Ad^2+5\ldots}{\ldots} - \frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{3/2}}{5b\,(a^2+b^2)\,f(a+b\tan(e+fx))^{5/2}} - \frac{8\int \frac{15}{8}b^2(bc-ad)(a^3(c^2C+2Bcd-Cd^2-A(c^2-d^2))-3ab^2(c^2C+2Bcd-Cd^2-A(c^2-d^2))-3a^2b(2c(A-C)d+B(c^2-d^2))+b^3(2c(A\ldots}{\ldots}$$

$$= -\frac{2(2a^3bBd+3a^4Cd+b^4(5Bc+3Ad)+2ab^3(5Ac-5cC-4Bd)-a^2b^2(5Bc+7Ad-13Cd))\,\sqrt{\ldots}}{15b^2\,(a^2+b^2)^2\,f(a+b\tan(e+fx))^{3/2}} - \frac{2(2a^5bBd^2+3a^6Cd^2+a^4b^2d(10Bc+(8A+C)d)+a^2b^4(45Ac^2-45c^2C-90Bcd-49Ad^2+5\ldots}{\ldots} - \frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{3/2}}{5b\,(a^2+b^2)\,f(a+b\tan(e+fx))^{5/2}} + \frac{((A-iB-C)(c-id)^2)\int \frac{1+i\tan(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx}{2(a-ib)^3} + \frac{((A+iB-C)(c+id)^2)\int \frac{1-i\tan(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx}{2(a+ib)^3}$$

$$= -\frac{2(2a^3bBd+3a^4Cd+b^4(5Bc+3Ad)+2ab^3(5Ac-5cC-4Bd)-a^2b^2(5Bc+7Ad-13Cd))}{15b^2(a^2+b^2)^2 f(a+b\tan(e+fx))^{3/2}}$$

$$-\frac{2(2a^5bBd^2+3a^6Cd^2+a^4b^2d(10Bc+(8A+C)d)+a^2b^4(45Ac^2-45c^2C-90Bcd-49Ad^2+}{}$$

$$-\frac{2(Ab^2-a(bB-aC))(c+d\tan(e+fx))^{3/2}}{5b(a^2+b^2)f(a+b\tan(e+fx))^{5/2}}$$

$$+\frac{((A-iB-C)(c-id)^2)\operatorname{Subst}\left(\int\frac{1}{(1-ix)\sqrt{a+bx}\sqrt{c+dx}}\,dx,x,\tan(e+fx)\right)}{2(a-ib)^3f}$$

$$+\frac{((A+iB-C)(c+id)^2)\operatorname{Subst}\left(\int\frac{1}{(1+ix)\sqrt{a+bx}\sqrt{c+dx}}\,dx,x,\tan(e+fx)\right)}{2(a+ib)^3f}$$

$$= -\frac{2(2a^3bBd+3a^4Cd+b^4(5Bc+3Ad)+2ab^3(5Ac-5cC-4Bd)-a^2b^2(5Bc+7Ad-13Cd))}{15b^2(a^2+b^2)^2 f(a+b\tan(e+fx))^{3/2}}$$

$$-\frac{2(2a^5bBd^2+3a^6Cd^2+a^4b^2d(10Bc+(8A+C)d)+a^2b^4(45Ac^2-45c^2C-90Bcd-49Ad^2+}{}$$

$$-\frac{2(Ab^2-a(bB-aC))(c+d\tan(e+fx))^{3/2}}{5b(a^2+b^2)f(a+b\tan(e+fx))^{5/2}}$$

$$+\frac{((A-iB-C)(c-id)^2)\operatorname{Subst}\left(\int\frac{1}{ia+b-(ic+d)x^2}\,dx,x,\frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{(a-ib)^3f}$$

$$+\frac{((A+iB-C)(c+id)^2)\operatorname{Subst}\left(\int\frac{1}{-ia+b-(-ic+d)x^2}\,dx,x,\frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{(a+ib)^3f}$$

$$= -\frac{(iA+B-iC)(c-id)^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a-ib)^{7/2}f}$$

$$-\frac{(B-i(A-C))(c+id)^{3/2}\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a+ib)^{7/2}f}$$

$$-\frac{2(2a^3bBd+3a^4Cd+b^4(5Bc+3Ad)+2ab^3(5Ac-5cC-4Bd)-a^2b^2(5Bc+7Ad-13Cd))}{15b^2(a^2+b^2)^2 f(a+b\tan(e+fx))^{3/2}}$$

$$-\frac{2(2a^5bBd^2+3a^6Cd^2+a^4b^2d(10Bc+(8A+C)d)+a^2b^4(45Ac^2-45c^2C-90Bcd-49Ad^2+}{}$$

$$-\frac{2(Ab^2-a(bB-aC))(c+d\tan(e+fx))^{3/2}}{5b(a^2+b^2)f(a+b\tan(e+fx))^{5/2}}$$

Mathematica [B] (verified)

Both result and optimal contain complex but leaf count is larger than twice the leaf count of optimal. 3134 vs. 2(586) = 1172.

Time = 9.50 (sec) , antiderivative size = 3134, normalized size of antiderivative = 5.35

$$\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{7/2}}\,dx = \text{Result too large to show}$$

```
[In] Integrate[((c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/(a + b*Tan[e + f*x])^(7/2),x]
```

```
[Out] -((C*(c + d*Tan[e + f*x])^(3/2))/(b*f*(a + b*Tan[e + f*x])^(5/2))) - (-1/4*
((3*b*c*C - 2*b*B*d - 3*a*C*d)*Sqrt[c + d*Tan[e + f*x]])/(b*f*(a + b*Tan[e
+ f*x])^(5/2)) - ((-2*((b^2*(8*A*b^2*c^2 + 3*a^2*C*d^2 - 2*a*b*d*(3*c*C - B
*d) - 5*b^2*c*(c*C + 2*B*d)))/4 - a*(-1/4*(a*(8*b^2*d*(B*c + (A - C)*d) + (
b*c - a*d)*(3*b*c*C - 2*b*B*d - 3*a*C*d))) + 2*b^3*(2*c*(A - C)*d + B*(c^2
- d^2))))*Sqrt[c + d*Tan[e + f*x]])/(5*(a^2 + b^2)*(b*c - a*d)*f*(a + b*Tan
[e + f*x])^(5/2)) - (2*((-2*(b^2*(((2*b^2*d - (5*a*(b*c - a*d))/2)*(8*A*b^2
*c^2 + 3*a^2*C*d^2 - 2*a*b*d*(3*c*C - B*d) - 5*b^2*c*(c*C + 2*B*d)))/4 + ((
-5*b*c)/2 + (a*d)/2)*(-1/4*(a*(8*b^2*d*(B*c + (A - C)*d) + (b*c - a*d)*(3*b
*c*C - 2*b*B*d - 3*a*C*d))) + 2*b^3*(2*c*(A - C)*d + B*(c^2 - d^2)))) - a*(
(5*b*(b*c - a*d)*((b*(8*A*b^2*c^2 + 3*a^2*C*d^2 - 2*a*b*d*(3*c*C - B*d) - 5
*b^2*c*(c*C + 2*B*d)))/4 - (b*(8*b^2*d*(B*c + (A - C)*d) + (b*c - a*d)*(3*b
*c*C - 2*b*B*d - 3*a*C*d)))/4 - 2*a*b^2*(2*c*(A - C)*d + B*(c^2 - d^2))))/2
 - 2*a*d*((b^2*(8*A*b^2*c^2 + 3*a^2*C*d^2 - 2*a*b*d*(3*c*C - B*d) - 5*b^2*c
*(c*C + 2*B*d)))/4 - a*(-1/4*(a*(8*b^2*d*(B*c + (A - C)*d) + (b*c - a*d)*(3
*b*c*C - 2*b*B*d - 3*a*C*d))) + 2*b^3*(2*c*(A - C)*d + B*(c^2 - d^2))))))*S
qrt[c + d*Tan[e + f*x]])/(3*(a^2 + b^2)*(b*c - a*d)*f*(a + b*Tan[e + f*x])^
(3/2)) - (2*((-15*b^2*(b*c - a*d)^2*(((3*a^2*A*b*c^2 - A*b^3*c^2 - a^3*B*c^
2 + 3*a*b^2*B*c^2 - 3*a^2*b*c^2*C + b^3*c^2*C - 2*a^3*A*c*d + 6*a*A*b^2*c*d
 - 6*a^2*b*B*c*d + 2*b^3*B*c*d + 2*a^3*c*C*d - 6*a*b^2*c*C*d - 3*a^2*A*b*d^
2 + A*b^3*d^2 + a^3*B*d^2 - 3*a*b^2*B*d^2 + 3*a^2*b*C*d^2 - b^3*C*d^2 + I*(
-(a^3*A*c^2) + 3*a*A*b^2*c^2 - 3*a^2*b*B*c^2 + b^3*B*c^2 + a^3*c^2*C - 3*a*
b^2*c^2*C - 6*a^2*A*b*c*d + 2*A*b^3*c*d + 2*a^3*B*c*d - 6*a*b^2*B*c*d + 6*a
^2*b*c*C*d - 2*b^3*c*C*d + a^3*A*d^2 - 3*a*A*b^2*d^2 + 3*a^2*b*B*d^2 - b^3*
B*d^2 - a^3*C*d^2 + 3*a*b^2*C*d^2))*ArcTanh[(Sqrt[-c + I*d]*Sqrt[a + b*Tan[
e + f*x]])/(Sqrt[-a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[-a + I*b]*Sqrt
[-c + I*d]) + ((-3*a^2*A*b*c^2 + A*b^3*c^2 + a^3*B*c^2 - 3*a*b^2*B*c^2 + 3*
a^2*b*c^2*C - b^3*c^2*C + 2*a^3*A*c*d - 6*a*A*b^2*c*d + 6*a^2*b*B*c*d - 2*b
^3*B*c*d - 2*a^3*c*C*d + 6*a*b^2*c*C*d + 3*a^2*A*b*d^2 - A*b^3*d^2 - a^3*B*
d^2 + 3*a*b^2*B*d^2 - 3*a^2*b*C*d^2 + b^3*C*d^2 + I*(-(a^3*A*c^2) + 3*a*A*b
^2*c^2 - 3*a^2*b*B*c^2 + b^3*B*c^2 + a^3*c^2*C - 3*a*b^2*c^2*C - 6*a^2*A*b*
c*d + 2*A*b^3*c*d + 2*a^3*B*c*d - 6*a*b^2*B*c*d + 6*a^2*b*c*C*d - 2*b^3*c*C
*d + a^3*A*d^2 - 3*a*A*b^2*d^2 + 3*a^2*b*B*d^2 - b^3*B*d^2 - a^3*C*d^2 + 3*
```

```
a*b^2*C*d^2))*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + I*
b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[a + I*b]*Sqrt[c + I*d])))/(2*(a^2 + b^
2)*f) - (2*(b^2*((b^2*d - (3*a*(b*c - a*d))/2)*(((2*b^2*d - (5*a*(b*c - a*d
))/2)*(8*A*b^2*c^2 + 3*a^2*C*d^2 - 2*a*b*d*(3*c*C - B*d) - 5*b^2*c*(c*C + 2
*B*d)))/4 + ((-5*b*c)/2 + (a*d)/2)*(-1/4*(a*(8*b^2*d*(B*c + (A - C)*d) + (b
*c - a*d)*(3*b*c*C - 2*b*B*d - 3*a*C*d))) + 2*b^3*(2*c*(A - C)*d + B*(c^2 -
 d^2)))) + ((-3*b*c)/2 + (a*d)/2)*((5*b*(b*c - a*d)*((b*(8*A*b^2*c^2 + 3*a^
2*C*d^2 - 2*a*b*d*(3*c*C - B*d) - 5*b^2*c*(c*C + 2*B*d)))/4 - (b*(8*b^2*d*(
B*c + (A - C)*d) + (b*c - a*d)*(3*b*c*C - 2*b*B*d - 3*a*C*d)))/4 - 2*a*b^2*
(2*c*(A - C)*d + B*(c^2 - d^2))))/2 - 2*a*d*((b^2*(8*A*b^2*c^2 + 3*a^2*C*d^
2 - 2*a*b*d*(3*c*C - B*d) - 5*b^2*c*(c*C + 2*B*d)))/4 - a*(-1/4*(a*(8*b^2*d
*(B*c + (A - C)*d) + (b*c - a*d)*(3*b*c*C - 2*b*B*d - 3*a*C*d))) + 2*b^3*(2
*c*(A - C)*d + B*(c^2 - d^2)))))) - a*((3*b*(b*c - a*d)*((-5*a*(b*c - a*d)*
((b*(8*A*b^2*c^2 + 3*a^2*C*d^2 - 2*a*b*d*(3*c*C - B*d) - 5*b^2*c*(c*C + 2*B
*d)))/4 - (b*(8*b^2*d*(B*c + (A - C)*d) + (b*c - a*d)*(3*b*c*C - 2*b*B*d -
3*a*C*d)))/4 - 2*a*b^2*(2*c*(A - C)*d + B*(c^2 - d^2))))/2 - 2*b*d*((b^2*(8
*A*b^2*c^2 + 3*a^2*C*d^2 - 2*a*b*d*(3*c*C - B*d) - 5*b^2*c*(c*C + 2*B*d)))/
4 - a*(-1/4*(a*(8*b^2*d*(B*c + (A - C)*d) + (b*c - a*d)*(3*b*c*C - 2*b*B*d
- 3*a*C*d))) + 2*b^3*(2*c*(A - C)*d + B*(c^2 - d^2)))) + b*(((2*b^2*d - (5*
a*(b*c - a*d))/2)*(8*A*b^2*c^2 + 3*a^2*C*d^2 - 2*a*b*d*(3*c*C - B*d) - 5*b^
2*c*(c*C + 2*B*d)))/4 + ((-5*b*c)/2 + (a*d)/2)*(-1/4*(a*(8*b^2*d*(B*c + (A
- C)*d) + (b*c - a*d)*(3*b*c*C - 2*b*B*d - 3*a*C*d))) + 2*b^3*(2*c*(A - C)*
d + B*(c^2 - d^2))))))/2 - a*d*(b^2*(((2*b^2*d - (5*a*(b*c - a*d))/2)*(8*A*
b^2*c^2 + 3*a^2*C*d^2 - 2*a*b*d*(3*c*C - B*d) - 5*b^2*c*(c*C + 2*B*d)))/4 +
 ((-5*b*c)/2 + (a*d)/2)*(-1/4*(a*(8*b^2*d*(B*c + (A - C)*d) + (b*c - a*d)*(
3*b*c*C - 2*b*B*d - 3*a*C*d))) + 2*b^3*(2*c*(A - C)*d + B*(c^2 - d^2)))) -
a*((5*b*(b*c - a*d)*((b*(8*A*b^2*c^2 + 3*a^2*C*d^2 - 2*a*b*d*(3*c*C - B*d)
- 5*b^2*c*(c*C + 2*B*d)))/4 - (b*(8*b^2*d*(B*c + (A - C)*d) + (b*c - a*d)*(
3*b*c*C - 2*b*B*d - 3*a*C*d)))/4 - 2*a*b^2*(2*c*(A - C)*d + B*(c^2 - d^2)))
)/2 - 2*a*d*((b^2*(8*A*b^2*c^2 + 3*a^2*C*d^2 - 2*a*b*d*(3*c*C - B*d) - 5*b^
2*c*(c*C + 2*B*d)))/4 - a*(-1/4*(a*(8*b^2*d*(B*c + (A - C)*d) + (b*c - a*d)
*(3*b*c*C - 2*b*B*d - 3*a*C*d))) + 2*b^3*(2*c*(A - C)*d + B*(c^2 - d^2)))))
)))*Sqrt[c + d*Tan[e + f*x]])/((a^2 + b^2)*(b*c - a*d)*f*Sqrt[a + b*Tan[e +
 f*x]])))/(3*(a^2 + b^2)*(b*c - a*d))))/(5*(a^2 + b^2)*(b*c - a*d)))/(2*b))
/b
```

Maple [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(fx+e))^{\frac{3}{2}}\left(A+B\tan(fx+e)+C\tan(fx+e)^2\right)}{(a+b\tan(fx+e))^{\frac{7}{2}}}dx$$

```
[In] int((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(7/2),x)
```

```
[Out] int((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(7/2),x)
```

Fricas [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{3/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{7/2}}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(7/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int \frac{(c+d\tan(e+fx))^{3/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{7/2}}\,dx = \int \frac{(c+d\tan(e+fx))^{\frac{3}{2}}\,(A+B\tan(e$$

```
[In] integrate((c+d*tan(f*x+e))**(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e))**(7/2),x)
```

```
[Out] Integral((c + d*tan(e + f*x))**(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)**
2)/(a + b*tan(e + f*x))**(7/2), x)
```

Maxima [F(-2)]

Exception generated.

$$\int \frac{(c+d\tan(e+fx))^{3/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{7/2}}\,dx = \text{Exception raised: ValueError}$$

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(7/2),x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(((2*b*d+2*a*c)^2>0)', see `assume?`
 for mo
```

Giac [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{7/2}}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(7/2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{7/2}}\,dx = \text{Hanged}$$

```
[In] int(((c + d*tan(e + f*x))^(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x))^(7/2),x)
```

```
[Out] \text{Hanged}
```

3.141 $\int \sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}\,(A+B\tan$

Optimal result 1968
Rubi [A] (verified) 1969
Mathematica [A] (verified) 1975
Maple [F(-1)] 1976
Fricas [F(-1)] 1976
Sympy [F] 1977
Maxima [F] 1977
Giac [F(-1)] 1977
Mupad [F(-1)] 1978

Optimal result

Integrand size = 49, antiderivative size = 697

$$\int \sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx =$$
$$-\frac{\sqrt{a-ib}(iA+B-iC)(c-id)^{5/2}\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f}$$
$$+\frac{\sqrt{a+ib}(iA-B-iC)(c+id)^{5/2}\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f}$$
$$-\frac{(5a^4Cd^4-4a^3bd^3(5cC+2Bd)+2a^2b^2d^2(15c^2C+20Bcd+8(A-C)d^2)-4ab^3d(5c^3C+30Bc^2d+40c(A}{6}$$
$$+\frac{(64b^2d^2(Abc+aBc-bcC+aAd-bBd-aCd)+(bc-ad)\left(48b(Ab+aB-bC)d^2-5(bc-ad)(bcC-8b\right.}{64b^3df}$$
$$+\frac{(48b(Ab+aB-bC)d^2-5(bc-ad)(bcC-8bBd-aCd))\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{96b^2df}$$
$$-\frac{(bcC-8bBd-aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{24bdf}$$
$$+\frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{7/2}}{4df}$$

```
[Out] -1/64*(5*a^4*C*d^4-4*a^3*b*d^3*(2*B*d+5*C*c)+2*a^2*b^2*d^2*(15*c^2*C+20*B*c
*d+8*(A-C)*d^2)-4*a*b^3*d*(5*c^3*C+30*B*c^2*d+40*c*(A-C)*d^2-16*B*d^3)+b^4*
(5*c^4*C-40*B*c^3*d-240*c^2*(A-C)*d^2+320*B*c*d^3+128*(A-C)*d^4))*arctanh(d
^(1/2)*(a+b*tan(f*x+e))^(1/2)/b^(1/2)/(c+d*tan(f*x+e))^(1/2))/b^(7/2)/d^(3/
2)/f-(I*A+B-I*C)*(c-I*d)^(5/2)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)
/(a-I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))*(a-I*b)^(1/2)/f+(I*A-B-I*C)*(c+I*d)^
(5/2)*arctanh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f
```

```
*x+e))^(1/2))*(a+I*b)^(1/2)/f+1/64*(64*b^2*d^2*(A*a*d+A*b*c+B*a*c-B*b*d-C*a
*d-C*b*c)+(-a*d+b*c)*(48*b*(A*b+B*a-C*b)*d^2-5*(-a*d+b*c)*(-8*B*b*d-C*a*d+C
*b*c)))*(a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)/b^3/d/f+1/96*(48*b*(A
*b+B*a-C*b)*d^2-5*(-a*d+b*c)*(-8*B*b*d-C*a*d+C*b*c))*(a+b*tan(f*x+e))^(1/2)
*(c+d*tan(f*x+e))^(3/2)/b^2/d/f-1/24*(-8*B*b*d-C*a*d+C*b*c)*(a+b*tan(f*x+e)
)^(1/2)*(c+d*tan(f*x+e))^(5/2)/b/d/f+1/4*C*(a+b*tan(f*x+e))^(1/2)*(c+d*tan(
f*x+e))^(7/2)/d/f
```

Rubi [A] (verified)

Time = 12.03 (sec) , antiderivative size = 697, normalized size of antiderivative = 1.00, number of steps used = 16, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}} = 0.163$, Rules used = {3728, 3736, 6857, 65, 223, 212, 95, 214}

$$\int \sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx =$$

$$-\frac{(5a^4Cd^4-4a^3bd^3(2Bd+5cC)+2a^2b^2d^2(8d^2(A-C)+20Bcd+15c^2C)-4ab^3d(40cd^2(A-C)+30Bc\cdots}{\cdots}$$

$$-\frac{\sqrt{a-ib}(c-id)^{5/2}(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f}$$

$$+\frac{\sqrt{a+ib}(c+id)^{5/2}(iA-B-iC)\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f}$$

$$+\frac{\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}\left(48bd^2(aB+Ab-bC)-5(bc-ad)(-aCd-8bBd+bcC)\right)}{96b^2df}$$

$$+\frac{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}(64b^2d^2(aAd+aBc-aCd+Abc-bBd-bcC)+(bc-ad)\,(48b\cdots}{64b^3df}$$

$$-\frac{(-aCd-8bBd+bcC)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{24bdf}$$

$$+\frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{7/2}}{4df}$$

```
[In] Int[Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x]
 + C*Tan[e + f*x]^2),x]
```

```
[Out] -((Sqrt[a - I*b]*(I*A + B - I*C)*(c - I*d)^(5/2)*ArcTanh[(Sqrt[c - I*d]*Sqr
t[a + b*Tan[e + f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/f) + (Sqr
t[a + I*b]*(I*A - B - I*C)*(c + I*d)^(5/2)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a +
b*Tan[e + f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/f - ((5*a^4*C*d
^4 - 4*a^3*b*d^3*(5*c*C + 2*B*d) + 2*a^2*b^2*d^2*(15*c^2*C + 20*B*c*d + 8*(
A - C)*d^2) - 4*a*b^3*d*(5*c^3*C + 30*B*c^2*d + 40*c*(A - C)*d^2 - 16*B*d^3
) + b^4*(5*c^4*C - 40*B*c^3*d - 240*c^2*(A - C)*d^2 + 320*B*c*d^3 + 128*(A
- C)*d^4))*ArcTanh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c + d*T
```

```
an[e + f*x]])])/(64*b^(7/2)*d^(3/2)*f) + ((64*b^2*d^2*(A*b*c + a*B*c - b*c*
C + a*A*d - b*B*d - a*C*d) + (b*c - a*d)*(48*b*(A*b + a*B - b*C)*d^2 - 5*(b
*c - a*d)*(b*c*C - 8*b*B*d - a*C*d)))*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*T
an[e + f*x]])/(64*b^3*d*f) + ((48*b*(A*b + a*B - b*C)*d^2 - 5*(b*c - a*d)*(
b*c*C - 8*b*B*d - a*C*d))*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(3/
2))/(96*b^2*d*f) - ((b*c*C - 8*b*B*d - a*C*d)*Sqrt[a + b*Tan[e + f*x]]*(c +
 d*Tan[e + f*x])^(5/2))/(24*b*d*f) + (C*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan
[e + f*x])^(7/2))/(4*d*f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 95

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 223

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
```

```
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3736

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[(a + b*ff*x)^m*(c + d*ff*x)^n*((A + B*ff*x + C*ff^2*x^2
)/(1 + ff^2*x^2)), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6857

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

integral

$$= \frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{7/2}}{4df} + \frac{\int \frac{(c+d\tan(e+fx))^{5/2}\left(\frac{1}{2}(-bcC+a(8A-7C)d)+4(Ab+aB-bC)d\tan(e+fx)-\frac{1}{2}(bcC-8bBd-aCd)\tan^2(e+fx)\right)}{\sqrt{a+b\tan(e+fx)}}\,dx}{4d}$$

$$= -\frac{(bcC-8bBd-aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{24bdf} + \frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{7/2}}{4df} + \frac{\int \frac{(c+d\tan(e+fx))^{3/2}\left(\frac{1}{4}(-6bc(bcC-a(8A-7C)d)+(bc+5ad)(bcC-8bBd-aCd))+12bd(Abc+aBc-bcC+aAd-bBd-aCd)\tan(e+fx)+\frac{1}{4}(4\right.}{\sqrt{a+b\tan(e+fx)}}}{12bd}$$

$$= \frac{(48b(Ab+aB-bC)d^2-5(bc-ad)(bcC-8bBd-aCd))\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{96b^2df} - \frac{(bcC-8bBd-aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{24bdf} + \frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{7/2}}{4df} + \frac{\int \frac{\sqrt{c+d\tan(e+fx)}\left(\frac{3}{8}(5a^3Cd^3-a^2bd^2(15cC+8Bd)-b^3c(5c^2C+24Bcd+16(A-C)d^2)+ab^2d(64Ac^2-49c^2C-96Bcd-48Ad^2+48Cd^2))+24b^2c}{}}{}$$

$$= \frac{(64b^2d^2(Abc+aBc-bcC+aAd-bBd-aCd)+(bc-ad)(48b(Ab+aB-bC)d^2-5(bc-ad)(bcC-8}{64b^3df} + \frac{(48b(Ab+aB-bC)d^2-5(bc-ad)(bcC-8bBd-aCd))\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{96b^2df} - \frac{(bcC-8bBd-aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{24bdf} + \frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{7/2}}{4df} + \frac{\int \frac{-\frac{3}{16}(5a^4Cd^4-4a^3bd^3(5cC+2Bd)+2a^2b^2d^2(15c^2C+20Bcd+8(A-C)d^2)+b^4c(5c^3C+88Bc^2d+144c(A-C)d^2-64Bd^3)+4ab^3d(27c^3C+66B}{}}{}$$

$$= \frac{(64b^2d^2(Abc+aBc-bcC+aAd-bBd-aCd)+(bc-ad)(48b(Ab+aB-bC)d^2-5(bc-ad)(bcC-8}{64b^3df} + \frac{(48b(Ab+aB-bC)d^2-5(bc-ad)(bcC-8bBd-aCd))\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{96b^2df} - \frac{(bcC-8bBd-aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{24bdf} + \frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{7/2}}{4df} + \frac{\text{Subst}\left(\int \frac{-\frac{3}{16}(5a^4Cd^4-4a^3bd^3(5cC+2Bd)+2a^2b^2d^2(15c^2C+20Bcd+8(A-C)d^2)+b^4c(5c^3C+88Bc^2d+144c(A-C)d^2-64Bd^3)+4ab^3d(27}{}\right.}{}$$

$$= \frac{(64b^2d^2(Abc+aBc-bcC+aAd-bBd-aCd)+(bc-ad)\,(48b(Ab+aB-bC)d^2-5(bc-ad)(bcC-}{64b^3df}$$

$$+\frac{(48b(Ab+aB-bC)d^2-5(bc-ad)(bcC-8bBd-aCd))\,\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{96b^2df}$$

$$-\frac{(bcC-8bBd-aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{24bdf}$$

$$+\frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{7/2}}{4df}$$

$$+\frac{\text{Subst}\Big(\int\Big(-\frac{3\left(5a^4Cd^4-4a^3bd^3(5cC+2Bd)+2a^2b^2d^2\left(15c^2C+20Bcd+8(A-C)d^2\right)-4ab^3d\left(5c^3C+30Bc^2d+40c(A-C)d^2-16Bd^3\right)+b^4\left(5\right.\right.}{16\sqrt{a+bx}\sqrt{c+dx}}}{}$$

$$= \frac{(64b^2d^2(Abc+aBc-bcC+aAd-bBd-aCd)+(bc-ad)\,(48b(Ab+aB-bC)d^2-5(bc-ad)(bcC-}{64b^3df}$$

$$+\frac{(48b(Ab+aB-bC)d^2-5(bc-ad)(bcC-8bBd-aCd))\,\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{96b^2df}$$

$$-\frac{(bcC-8bBd-aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{24bdf}$$

$$+\frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{7/2}}{4df}$$

$$+\frac{\text{Subst}\Big(\int\frac{-b^3d\left(a\left(c^3C+3Bc^2d-3cCd^2-Bd^3-A\left(c^3-3cd^2\right)\right)+b\left((A-C)d\left(3c^2-d^2\right)+B\left(c^3-3cd^2\right)\right)\right)+b^3d\left(A\left(bc^3+3ac^2d-3bcd^2-ad^3\right)-b\left(c^3\right.\right.}{\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}}{b^3df}$$

$$-\frac{(5a^4Cd^4-4a^3bd^3(5cC+2Bd)+2a^2b^2d^2(15c^2C+20Bcd+8(A-C)d^2)-4ab^3d(5c^3C+30Bc^2d+4}{}$$

$$= \frac{(64b^2d^2(Abc+aBc-bcC+aAd-bBd-aCd)+(bc-ad)\,(48b(Ab+aB-bC)d^2-5(bc-ad)(bcC-}{64b^3df}$$

$$+\frac{(48b(Ab+aB-bC)d^2-5(bc-ad)(bcC-8bBd-aCd))\,\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{96b^2df}$$

$$-\frac{(bcC-8bBd-aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{24bdf}$$

$$+\frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{7/2}}{4df}$$

$$+\frac{\text{Subst}\Big(\int\Big(\frac{-b^3d\left(A\left(bc^3+3ac^2d-3bcd^2-ad^3\right)-b\left(c^3C+3Bc^2d-3cCd^2-Bd^3\right)+a\left(Bc^3-3c^2Cd-3Bcd^2+Cd^3\right)\right)-ib^3d\left(a\left(c^3C+3Bc^2d-3cC\right.\right.}{2(i-x)\sqrt{a+bx}\sqrt{c+dx}}}{}$$

$$-\frac{(5a^4Cd^4-4a^3bd^3(5cC+2Bd)+2a^2b^2d^2(15c^2C+20Bcd+8(A-C)d^2)-4ab^3d(5c^3C+30Bc^2d+4}{}$$

$$= \frac{(64b^2d^2(Abc + aBc - bcC + aAd - bBd - aCd) + (bc - ad)\,(48b(Ab + aB - bC)d^2 - 5(bc - ad)(bcC - 8}{64b^3df}$$

$$+ \frac{(48b(Ab + aB - bC)d^2 - 5(bc - ad)(bcC - 8bBd - aCd))\,\sqrt{a + b\tan(e + fx)}(c + d\tan(e + fx))^{3/2}}{96b^2df}$$

$$- \frac{(bcC - 8bBd - aCd)\sqrt{a + b\tan(e + fx)}(c + d\tan(e + fx))^{5/2}}{24bdf}$$

$$+ \frac{C\sqrt{a + b\tan(e + fx)}(c + d\tan(e + fx))^{7/2}}{4df}$$

$$+ \frac{((ia + b)(A - iB - C)(c - id)^3)\,\mathrm{Subst}\left(\int \frac{1}{(i+x)\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e + fx)\right)}{2f}$$

$$+ \frac{((ia - b)(A + iB - C)(c + id)^3)\,\mathrm{Subst}\left(\int \frac{1}{(i-x)\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e + fx)\right)}{2f}$$

$$- \frac{(5a^4Cd^4 - 4a^3bd^3(5cC + 2Bd) + 2a^2b^2d^2(15c^2C + 20Bcd + 8(A - C)d^2) - 4ab^3d(5c^3C + 30Bc^2d + 40c}{}$$

$$=$$

$$- \frac{(5a^4Cd^4 - 4a^3bd^3(5cC + 2Bd) + 2a^2b^2d^2(15c^2C + 20Bcd + 8(A - C)d^2) - 4ab^3d(5c^3C + 30Bc^2d + 40c}{}$$

$$+ \frac{(64b^2d^2(Abc + aBc - bcC + aAd - bBd - aCd) + (bc - ad)\,(48b(Ab + aB - bC)d^2 - 5(bc - ad)(bcC -}{64b^3df}$$

$$+ \frac{(48b(Ab + aB - bC)d^2 - 5(bc - ad)(bcC - 8bBd - aCd))\,\sqrt{a + b\tan(e + fx)}(c + d\tan(e + fx))^{3/2}}{96b^2df}$$

$$- \frac{(bcC - 8bBd - aCd)\sqrt{a + b\tan(e + fx)}(c + d\tan(e + fx))^{5/2}}{24bdf}$$

$$+ \frac{C\sqrt{a + b\tan(e + fx)}(c + d\tan(e + fx))^{7/2}}{4df}$$

$$+ \frac{((ia + b)(A - iB - C)(c - id)^3)\,\mathrm{Subst}\left(\int \frac{1}{-a+ib-(-c+id)x^2}\,dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{f}$$

$$+ \frac{((ia - b)(A + iB - C)(c + id)^3)\,\mathrm{Subst}\left(\int \frac{1}{a+ib-(c+id)x^2}\,dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{f}$$

$$= -\frac{\sqrt{a-ib}(iA+B-iC)(c-id)^{5/2}\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f}$$

$$-\frac{\sqrt{a+ib}(B-i(A-C))(c+id)^{5/2}\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f}$$

$$-\frac{(5a^4Cd^4-4a^3bd^3(5cC+2Bd)+2a^2b^2d^2(15c^2C+20Bcd+8(A-C)d^2)-4ab^3d(5c^3C+30Bc^2d+4}{}$$

$$+\frac{(64b^2d^2(Abc+aBc-bcC+aAd-bBd-aCd)+(bc-ad)\left(48b(Ab+aB-bC)d^2-5(bc-ad)(bcC\right.}{64b^3df}$$

$$+\frac{(48b(Ab+aB-bC)d^2-5(bc-ad)(bcC-8bBd-aCd))\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{96b^2df}$$

$$-\frac{(bcC-8bBd-aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{24bdf}$$

$$+\frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{7/2}}{4df}$$

Mathematica [A] (verified)

Time = 9.88 (sec) , antiderivative size = 1261, normalized size of antiderivative = 1.81

$$\int \sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{7/2}}{4df}$$

$$+\frac{\frac{(-bcC+8bBd+aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{6bf}+\frac{\left(48b(Ab+aB-bC)d^2-5(bc-ad)(bcC-8bBd-aCd)\right)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{8bf}+}{}$$

```
[In] Integrate[Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e
+ f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] (C*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(7/2))/(4*d*f) + (((-(b*c*
C) + 8*b*B*d + a*C*d)*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(5/2))/
(6*b*f) + (((48*b*(A*b + a*B - b*C)*d^2 - 5*(b*c - a*d)*(b*c*C - 8*b*B*d -
a*C*d))*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(3/2))/(8*b*f) + (((2
4*b^2*d^2*(A*b*c + a*B*c - b*c*C + a*A*d - b*B*d - a*C*d) - (3*(-(b*c) + a*
d)*(48*b*(A*b + a*B - b*C)*d^2 - 5*(b*c - a*d)*(b*c*C - 8*b*B*d - a*C*d)))/
8)*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(b*f) + ((-24*b^3*d*(
Sqrt[-b^2]*(b*(A - C)*d*(3*c^2 - d^2) + b*B*(c^3 - 3*c*d^2) - a*(A*c^3 - c^
```

```
3*C - 3*B*c^2*d - 3*A*c*d^2 + 3*c*C*d^2 + B*d^3)) - b*(A*(b*c^3 + 3*a*c^2*d
 - 3*b*c*d^2 - a*d^3) - b*(c^3*C + 3*B*c^2*d - 3*c*C*d^2 - B*d^3) + a*(B*c^
3 - 3*c^2*C*d - 3*B*c*d^2 + C*d^3)))*ArcTanh[(Sqrt[-c + (Sqrt[-b^2]*d)/b]*S
qrt[a + b*Tan[e + f*x]])/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[c + d*Tan[e + f*x]])])
/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[-c + (Sqrt[-b^2]*d)/b]) - (24*b^3*d*(Sqrt[-b^2
]*(b*(A - C)*d*(3*c^2 - d^2) + b*B*(c^3 - 3*c*d^2) - a*(A*c^3 - c^3*C - 3*B
*c^2*d - 3*A*c*d^2 + 3*c*C*d^2 + B*d^3)) + b*(A*(b*c^3 + 3*a*c^2*d - 3*b*c*
d^2 - a*d^3) - b*(c^3*C + 3*B*c^2*d - 3*c*C*d^2 - B*d^3) + a*(B*c^3 - 3*c^2
*C*d - 3*B*c*d^2 + C*d^3)))*ArcTanh[(Sqrt[c + (Sqrt[-b^2]*d)/b]*Sqrt[a + b*
Tan[e + f*x]])/(Sqrt[a + Sqrt[-b^2]]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[a +
Sqrt[-b^2]]*Sqrt[c + (Sqrt[-b^2]*d)/b]) - (3*Sqrt[b]*Sqrt[c - (a*d)/b]*Sqrt
[(c/(c - (a*d)/b) - (a*d)/(b*(c - (a*d)/b)))^(-1)]*Sqrt[c/(c - (a*d)/b) - (
a*d)/(b*(c - (a*d)/b))]*(5*a^4*C*d^4 - 4*a^3*b*d^3*(5*c*C + 2*B*d) + 2*a^2*
b^2*d^2*(15*c^2*C + 20*B*c*d + 8*(A - C)*d^2) - 4*a*b^3*d*(5*c^3*C + 30*B*c
^2*d + 40*c*(A - C)*d^2 - 16*B*d^3) + b^4*(5*c^4*C - 40*B*c^3*d - 240*c^2*(
A - C)*d^2 + 320*B*c*d^3 + 128*(A - C)*d^4))*ArcSinh[(Sqrt[d]*Sqrt[a + b*Ta
n[e + f*x]])/(Sqrt[b]*Sqrt[c - (a*d)/b]*Sqrt[c/(c - (a*d)/b) - (a*d)/(b*(c
- (a*d)/b))])]*Sqrt[(c + d*Tan[e + f*x])/(c - (a*d)/b)])/(8*Sqrt[d]*Sqrt[c
+ d*Tan[e + f*x]]))/(b^2*f))/(2*b))/(3*b))/(4*d)
```

Maple [F(-1)]

Timed out.

$$\int \sqrt{a+b\tan\left(fx+e\right)}\left(c+d\tan\left(fx+e\right)\right)^{\frac{5}{2}}\left(A+B\tan\left(fx+e\right)+C\tan\left(fx+e\right)^{2}\right)dx$$

```
[In] int((a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x)
```

```
[Out] int((a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x)
```

Fricas [F(-1)]

Timed out.

$$\int \sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*t
an(f*x+e)^2),x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int \sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\int \sqrt{a+b\tan{(e+fx)}}(c+d\tan{(e+fx)})^{\frac{5}{2}}\left(A+B\tan{(e+fx)}+C\tan^2{(e+fx)}\right)\,dx$$

```
[In] integrate((a+b*tan(f*x+e))**(1/2)*(c+d*tan(f*x+e))**(5/2)*(A+B*tan(f*x+e)+C
*tan(f*x+e)**2),x)
```

```
[Out] Integral(sqrt(a + b*tan(e + f*x))*(c + d*tan(e + f*x))**(5/2)*(A + B*tan(e
+ f*x) + C*tan(e + f*x)**2), x)
```

Maxima [F]

$$\int \sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\int \left(C\tan{(fx+e)}^2+B\tan{(fx+e)}+A\right)\sqrt{b\tan{(fx+e)}+a}(d\tan{(fx+e)}+c)^{\frac{5}{2}}$$

```
[In] integrate((a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*t
an(f*x+e)^2),x, algorithm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*sqrt(b*tan(f*x + e) + a)*
(d*tan(f*x + e) + c)^(5/2), x)
```

Giac [F(-1)]

Timed out.

$$\int \sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*t
an(f*x+e)^2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\int \sqrt{a+b\tan\left(e+f\,x\right)}\,{\left(c+d\tan\left(e+f\,x\right)\right)}^{5/2}\,\left(C\,{\tan\left(e+f\,x\right)}^2+B\tan\left(e+f\,x\right)+A\right)\,d$$

```
[In] int((a + b*tan(e + f*x))^(1/2)*(c + d*tan(e + f*x))^(5/2)*(A + B*tan(e + f*
x) + C*tan(e + f*x)^2),x)
```

```
[Out] int((a + b*tan(e + f*x))^(1/2)*(c + d*tan(e + f*x))^(5/2)*(A + B*tan(e + f*
x) + C*tan(e + f*x)^2), x)
```

3.142 $\int \frac{(c+d\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{a+b\tan(e+fx)}}\,dx$

Optimal result 1979
Rubi [A] (verified) 1980
Mathematica [A] (verified) 1985
Maple [F(-1)] 1986
Fricas [F(-1)] 1986
Sympy [F] 1987
Maxima [F(-1)] 1987
Giac [F(-1)] 1987
Mupad [F(-1)] 1987

Optimal result

Integrand size = 49, antiderivative size = 505

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{a+b\tan(e+fx)}}\,dx =$$

$$-\frac{(iA+B-iC)(c-id)^{5/2}\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{a-ib}f}$$

$$-\frac{(B-i(A-C))(c+id)^{5/2}\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{a+ib}f}$$

$$-\frac{(5a^3Cd^3-3a^2bd^2(5cC+2Bd)+ab^2d(15c^2C+20Bcd+8(A-C)d^2)-b^3(5c^3C+30Bc^2d+40c(A-C\right.\cdots}{8b^{7/2}\sqrt{d}f}$$

$$+\frac{(8b^2d(Bc+(A-C)d)+(bc-ad)(5bcC+6bBd-5aCd))\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{8b^3f}$$

$$+\frac{(5bcC+6bBd-5aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{12b^2f}$$

$$+\frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{3bf}$$

```
[Out] -(I*A+B-I*C)*(c-I*d)^(5/2)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-
I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/f/(a-I*b)^(1/2)-(B-I*(A-C))*(c+I*d)^(5/2
)*arctanh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e
))^(1/2))/f/(a+I*b)^(1/2)-1/8*(5*a^3*C*d^3-3*a^2*b*d^2*(2*B*d+5*C*c)+a*b^2*
d*(15*c^2*C+20*B*c*d+8*(A-C)*d^2)-b^3*(5*c^3*C+30*B*c^2*d+40*c*(A-C)*d^2-16
*B*d^3))*arctanh(d^(1/2)*(a+b*tan(f*x+e))^(1/2)/b^(1/2)/(c+d*tan(f*x+e))^(1
/2))/b^(7/2)/f/d^(1/2)+1/8*(8*b^2*d*(B*c+(A-C)*d)+(-a*d+b*c)*(6*B*b*d-5*C*a
*d+5*C*b*c))*(a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)/b^3/f+1/12*(6*B*
```

```
b*d-5*C*a*d+5*C*b*c)*(a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(3/2)/b^2/f+1/
3*C*(a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(5/2)/b/f
```

Rubi [A] (verified)

Time = 7.13 (sec) , antiderivative size = 505, normalized size of antiderivative = 1.00, number of steps used = 15, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}} = 0.163$, Rules used = {3728, 3736, 6857, 65, 223, 212, 95, 214}

$$\int \frac{(c+d\tan(e+fx))^{5/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{a+b\tan(e+fx)}}\,dx =$$

$$-\frac{(5a^3Cd^3-3a^2bd^2(2Bd+5cC)+ab^2d(8d^2(A-C)+20Bcd+15c^2C)-(b^3(40cd^2(A-C)+30Bc^2d-16}{8b^{7/2}\sqrt{d}f}$$

$$-\frac{(c-id)^{5/2}(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f\sqrt{a-ib}}$$

$$-\frac{(c+id)^{5/2}(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f\sqrt{a+ib}}$$

$$+\frac{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}((bc-ad)(-5aCd+6bBd+5bcC)+8b^2d(d(A-C)+Bc))}{8b^3f}$$

$$+\frac{(-5aCd+6bBd+5bcC)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{12b^2f}$$

$$+\frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{3bf}$$

```
[In] Int[((c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/Sq
rt[a + b*Tan[e + f*x]],x]
```

```
[Out] -(((I*A + B - I*C)*(c - I*d)^(5/2)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e
+ f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])]])/(Sqrt[a - I*b]*f)) - ((
B - I*(A - C))*(c + I*d)^(5/2)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*
x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])]])/(Sqrt[a + I*b]*f) - ((5*a^3
*C*d^3 - 3*a^2*b*d^2*(5*c*C + 2*B*d) + a*b^2*d*(15*c^2*C + 20*B*c*d + 8*(A
- C)*d^2) - b^3*(5*c^3*C + 30*B*c^2*d + 40*c*(A - C)*d^2 - 16*B*d^3))*ArcTa
nh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])]])/
(8*b^(7/2)*Sqrt[d]*f) + ((8*b^2*d*(B*c + (A - C)*d) + (b*c - a*d)*(5*b*c*C
+ 6*b*B*d - 5*a*C*d))*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(8
*b^3*f) + ((5*b*c*C + 6*b*B*d - 5*a*C*d)*Sqrt[a + b*Tan[e + f*x]]*(c + d*Ta
n[e + f*x])^(3/2))/(12*b^2*f) + (C*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e +
f*x])^(5/2))/(3*b*f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 95

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 223

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3736

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
```

```
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[(a + b*ff*x)^m*(c + d*ff*x)^n*((A + B*ff*x + C*ff^2*x^2
)/(1 + ff^2*x^2)), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6857

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

integral

$$= \frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{3bf} + \frac{\int \frac{(c+d\tan(e+fx))^{3/2}\left(\frac{1}{2}(6Abc-C(bc+5ad))+3b(Bc+(A-C)d)\tan(e+fx)+\frac{1}{2}(5bcC+6bBd-5aCd)\tan^2(e+fx)\right)}{\sqrt{a+b\tan(e+fx)}}\,dx}{3b}$$

$$= \frac{(5bcC+6bBd-5aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{12b^2f} + \frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{3bf} + \frac{\int \frac{\sqrt{c+d\tan(e+fx)}\left(\frac{1}{4}(-((bc+3ad)(5bcC+6bBd-5aCd))+4bc(6Abc-C(bc+5ad)))+6b^2\left(2c(A-C)d+B\left(c^2-d^2\right)\right)\tan(e+fx)+\frac{3}{4}\left(8b^2d(Bc+(A\right.\right.}{\sqrt{a+b\tan(e+fx)}}}{6b^2}$$

$$= \frac{\left(8b^2d(Bc+(A-C)d)+(bc-ad)(5bcC+6bBd-5aCd)\right)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{8b^3f} + \frac{(5bcC+6bBd-5aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{12b^2f} + \frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{3bf} + \frac{\int \frac{-\frac{3}{8}\left(5a^3Cd^3-3a^2bd^2(5cC+2Bd)+b^3c\left(11c^2C+18Bcd-8Cd^2\right)+ab^2d\left(15c^2C+20Bcd-8Cd^2\right)-8Ab^2\left(2bc^3-bcd^2-ad^3\right)\right)+6b^3\left((A-C)d\left(3c^2-\right.\right.}{\sqrt{a+b\tan(e+fx)}}}{6}$$

$$= \frac{(8b^2d(Bc+(A-C)d)+(bc-ad)(5bcC+6bBd-5aCd))\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{8b^3f}$$
$$+\frac{(5bcC+6bBd-5aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{12b^2f}$$
$$+\frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{3bf}$$
$$+\frac{\text{Subst}\Big(\int \frac{-\frac{3}{8}(5a^3Cd^3-3a^2bd^2(5cC+2Bd)+b^3c(11c^2C+18Bcd-8Cd^2)+ab^2d(15c^2C+20Bcd-8Cd^2)-8Ab^2(2bc^3-bcd^2-ad^3))+6b^3((A-}{\sqrt{a+bx}\sqrt{}}}{}$$

$$= \frac{(8b^2d(Bc+(A-C)d)+(bc-ad)(5bcC+6bBd-5aCd))\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{8b^3f}$$
$$+\frac{(5bcC+6bBd-5aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{12b^2f}$$
$$+\frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{3bf}$$
$$+\frac{\text{Subst}\Big(\int\Big(-\frac{3(5a^3Cd^3-3a^2bd^2(5cC+2Bd)+ab^2d(15c^2C+20Bcd+8(A-C)d^2)-b^3(5c^3C+30Bc^2d+40c(A-C)d^2-16Bd^3))}{8\sqrt{a+bx}\sqrt{c+dx}}+\frac{6(-b^3(}{}}{6b^3f}$$

$$= \frac{(8b^2d(Bc+(A-C)d)+(bc-ad)(5bcC+6bBd-5aCd))\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{8b^3f}$$
$$+\frac{(5bcC+6bBd-5aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{12b^2f}$$
$$+\frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{3bf}$$
$$+\frac{\text{Subst}\Big(\int \frac{-b^3(c^3C+3Bc^2d-3cCd^2-Bd^3-A(c^3-3cd^2))+b^3((A-C)d(3c^2-d^2)+B(c^3-3cd^2))x}{\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}dx,x,\tan(e+fx)\Big)}{b^3f}$$
$$-\frac{(5a^3Cd^3-3a^2bd^2(5cC+2Bd)+ab^2d(15c^2C+20Bcd+8(A-C)d^2)-b^3(5c^3C+30Bc^2d+40c(A-}{16b^3f}$$

$$= \frac{(8b^2d(Bc+(A-C)d)+(bc-ad)(5bcC+6bBd-5aCd))\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{8b^3f}$$
$$+ \frac{(5bcC+6bBd-5aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{12b^2f}$$
$$+ \frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{3bf}$$
$$+ \frac{\text{Subst}\Big(\int\Big(\frac{-ib^3(c^3C+3Bc^2d-3cCd^2-Bd^3-A(c^3-3cd^2))-b^3((A-C)d(3c^2-d^2)+B(c^3-3cd^2))}{2(i-x)\sqrt{a+bx}\sqrt{c+dx}}+\frac{-ib^3(c^3C+3Bc^2d-3cCd^2-Bd^3-A(}{2(i+}}{b^3f}$$
$$- \frac{(5a^3Cd^3-3a^2bd^2(5cC+2Bd)+ab^2d(15c^2C+20Bcd+8(A-C)d^2)-b^3(5c^3C+30Bc^2d+40c(A-C}{8b^4f}$$

$$= \frac{(8b^2d(Bc+(A-C)d)+(bc-ad)(5bcC+6bBd-5aCd))\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{8b^3f}$$
$$+ \frac{(5bcC+6bBd-5aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{12b^2f}$$
$$+ \frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{3bf}$$
$$+ \frac{((iA+B-iC)(c-id)^3)\,\text{Subst}\Big(\int\frac{1}{(i+x)\sqrt{a+bx}\sqrt{c+dx}}\,dx,x,\tan(e+fx)\Big)}{2f}$$
$$- \frac{(5a^3Cd^3-3a^2bd^2(5cC+2Bd)+ab^2d(15c^2C+20Bcd+8(A-C)d^2)-b^3(5c^3C+30Bc^2d+40c(A-C}{8b^4f}$$
$$- \frac{(ib^3(c^3C+3Bc^2d-3cCd^2-Bd^3-A(c^3-3cd^2))+b^3((A-C)d(3c^2-d^2)+B(c^3-3cd^2)))\,\text{Subst}\Big(\int}{2b^3f}$$

$$= -\frac{(5a^3Cd^3 - 3a^2bd^2(5cC + 2Bd) + ab^2d(15c^2C + 20Bcd + 8(A-C)d^2) - b^3(5c^3C + 30Bc^2d + 40c(A-}{8b^{7/2}\sqrt{d}f}$$
$$+ \frac{(8b^2d(Bc + (A-C)d) + (bc-ad)(5bcC + 6bBd - 5aCd))\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{8b^3f}$$
$$+ \frac{(5bcC + 6bBd - 5aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{12b^2f}$$
$$+ \frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{3bf}$$
$$+ \frac{((iA+B-iC)(c-id)^3)\,\text{Subst}\left(\int \frac{1}{-a+ib-(-c+id)x^2}\,dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{f}$$
$$- \frac{(ib^3(c^3C + 3Bc^2d - 3cCd^2 - Bd^3 - A(c^3 - 3cd^2)) + b^3((A-C)d(3c^2-d^2) + B(c^3 - 3cd^2)))\,\text{Subst}\Big(\int}{b^3f}$$

$$= -\frac{(iA+B-iC)(c-id)^{5/2}\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{a-ib}f}$$
$$+ \frac{(iA-B-iC)(c+id)^{5/2}\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{a+ib}f}$$
$$- \frac{(5a^3Cd^3 - 3a^2bd^2(5cC + 2Bd) + ab^2d(15c^2C + 20Bcd + 8(A-C)d^2) - b^3(5c^3C + 30Bc^2d + 40c(A-}{8b^{7/2}\sqrt{d}f}$$
$$+ \frac{(8b^2d(Bc + (A-C)d) + (bc-ad)(5bcC + 6bBd - 5aCd))\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{8b^3f}$$
$$+ \frac{(5bcC + 6bBd - 5aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{12b^2f}$$
$$+ \frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}{3bf}$$

Mathematica [A] (verified)

Time = 9.04 (sec) , antiderivative size = 780, normalized size of antiderivative = 1.54

$$\int \frac{(c+d\tan(e+fx))^{5/2}(A + B\tan(e+fx) + C\tan^2(e+fx))}{\sqrt{a+b\tan(e+fx)}}\,dx = \frac{C\sqrt{a+b\tan(e+fx)}(c+d\tan(e+}{3bf}$$

$$+ \frac{\frac{(5bcC+6bBd-5aCd)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{4bf} + \frac{\frac{3\left(8b^2d(Bc+(A-C)d)+(bc-ad)(5bcC+6bBd-5aCd)\right)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{4bf} + \cdots}{}}{}$$

```
[In] Integrate[((c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/Sqrt[a + b*Tan[e + f*x]],x]
```

```
[Out] (C*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(5/2))/(3*b*f) + (((5*b*c*
C + 6*b*B*d - 5*a*C*d)*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(3/2))
/(4*b*f) + ((3*(8*b^2*d*(B*c + (A - C)*d) + (b*c - a*d)*(5*b*c*C + 6*b*B*d
- 5*a*C*d))*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(4*b*f) + ((
6*b^3*(b*(A - C)*d*(3*c^2 - d^2) + b*B*(c^3 - 3*c*d^2) + Sqrt[-b^2]*(A*c^3
- c^3*C - 3*B*c^2*d - 3*A*c*d^2 + 3*c*C*d^2 + B*d^3))*ArcTanh[(Sqrt[-c + (S
qrt[-b^2]*d)/b]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[c + d
*Tan[e + f*x]])]))/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[-c + (Sqrt[-b^2]*d)/b]) - (6*
b^3*(b*(A - C)*d*(3*c^2 - d^2) + b*B*(c^3 - 3*c*d^2) - Sqrt[-b^2]*(A*c^3 -
c^3*C - 3*B*c^2*d - 3*A*c*d^2 + 3*c*C*d^2 + B*d^3))*ArcTanh[(Sqrt[c + (Sqrt
[-b^2]*d)/b]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + Sqrt[-b^2]]*Sqrt[c + d*Tan
[e + f*x]])])/(Sqrt[a + Sqrt[-b^2]]*Sqrt[c + (Sqrt[-b^2]*d)/b]) - (3*Sqrt[b
]*Sqrt[c - (a*d)/b]*(5*a^3*C*d^3 - 3*a^2*b*d^2*(5*c*C + 2*B*d) + a*b^2*d*(1
5*c^2*C + 20*B*c*d + 8*(A - C)*d^2) - b^3*(5*c^3*C + 30*B*c^2*d + 40*c*(A -
 C)*d^2 - 16*B*d^3))*ArcSinh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sq
rt[c - (a*d)/b])]*Sqrt[(b*c + b*d*Tan[e + f*x])/(b*c - a*d)])/(4*Sqrt[d]*Sq
rt[c + d*Tan[e + f*x]]))/(b^2*f))/(2*b))/(3*b)
```

Maple [F(-1)]

Timed out.

hanged

```
[In] int((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(1/2),x)
```

```
[Out] int((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(1/2),x)
```

Fricas [F(-1)]

Timed out.

$$\int \frac{(c + d\tan(e + fx))^{5/2} \left(A + B\tan(e + fx) + C\tan^2(e + fx)\right)}{\sqrt{a + b\tan(e + fx)}}\, dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(1/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int \frac{(c+d\tan(e+fx))^{5/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{a+b\tan(e+fx)}}\,dx = \int \frac{(c+d\tan(e+fx))^{\frac{5}{2}}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{a+b\tan(e+fx)}}\,dx$$

```
[In] integrate((c+d*tan(f*x+e))**(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e))**(1/2),x)
[Out] Integral((c + d*tan(e + f*x))**(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)**
2)/sqrt(a + b*tan(e + f*x)), x)
```

Maxima [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{5/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{a+b\tan(e+fx)}}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(1/2),x, algorithm="maxima")
[Out] Timed out
```

Giac [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{5/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{a+b\tan(e+fx)}}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(1/2),x, algorithm="giac")
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{5/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{a+b\tan(e+fx)}}\,dx = \text{Hanged}$$

```
[In] int(((c + d*tan(e + f*x))^(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x))^(1/2),x)
[Out] \text{Hanged}
```

3.143 $\int \frac{(c+d\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{3/2}}\,dx$

Optimal result 1988
Rubi [A] (verified) 1989
Mathematica [B] (verified) 1994
Maple [F(-1)] 1996
Fricas [F(-1)] 1996
Sympy [F] 1996
Maxima [F(-1)] 1997
Giac [F(-1)] 1997
Mupad [F(-1)] 1997

Optimal result

Integrand size = 49, antiderivative size = 535

$$\int \frac{(c+d\tan(e+fx))^{5/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{3/2}}\,dx =$$
$$-\frac{(iA+B-iC)(c-id)^{5/2}\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a-ib)^{3/2}f}$$
$$-\frac{(B-i(A-C))(c+id)^{5/2}\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a+ib)^{3/2}f}$$
$$+\frac{\sqrt{d}(15a^2Cd^2-6abd(5cC+2Bd)+b^2(15c^2C+20Bcd+8(A-C)d^2))\,\text{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{4b^{7/2}f}$$
$$-\frac{d(15a^3Cd-8Ab^2(bc-ad)-3a^2b(5cC+4Bd)-b^3(7cC+4Bd)+ab^2(8Bc+7Cd))\,\sqrt{a+b\tan(e+fx)}}{4b^3\,(a^2+b^2)\,f}$$
$$+\frac{(4Ab^2-4abB+5a^2C+b^2C)\,d\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{2b^2\,(a^2+b^2)\,f}$$
$$-\frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{5/2}}{b\,(a^2+b^2)\,f\sqrt{a+b\tan(e+fx)}}$$

```
[Out] -(I*A+B-I*C)*(c-I*d)^(5/2)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-
I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/(a-I*b)^(3/2)/f-(B-I*(A-C))*(c+I*d)^(5/2
)*arctanh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e
))^(1/2))/(a+I*b)^(3/2)/f+1/4*(15*a^2*C*d^2-6*a*b*d*(2*B*d+5*C*c)+b^2*(15*c
^2*C+20*B*c*d+8*(A-C)*d^2))*arctanh(d^(1/2)*(a+b*tan(f*x+e))^(1/2)/b^(1/2)/
(c+d*tan(f*x+e))^(1/2))*d^(1/2)/b^(7/2)/f-1/4*d*(15*a^3*C*d-8*A*b^2*(-a*d+b
*c)-3*a^2*b*(4*B*d+5*C*c)-b^3*(4*B*d+7*C*c)+a*b^2*(8*B*c+7*C*d))*(a+b*tan(f
*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)/b^3/(a^2+b^2)/f+1/2*(4*A*b^2-4*B*a*b+5*
```

```
C*a^2+C*b^2)*d*(a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(3/2)/b^2/(a^2+b^2)/
f-2*(A*b^2-a*(B*b-C*a))*(c+d*tan(f*x+e))^(5/2)/b/(a^2+b^2)/f/(a+b*tan(f*x+e
))^(1/2)
```

Rubi [A] (verified)

Time = 9.89 (sec) , antiderivative size = 535, normalized size of antiderivative = 1.00, number of steps used = 15, number of rules used = 9, $\frac{\text{number of rules}}{\text{integrand size}} = 0.184$, Rules used = {3726, 3728, 3736, 6857, 65, 223, 212, 95, 214}

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{3/2}}\,dx = \frac{\sqrt{d}(15a^2Cd^2-6abd(2Bd+5cC)+b}{}$$

$$-\frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{5/2}}{bf\,(a^2+b^2)\,\sqrt{a+b\tan(e+fx)}}$$

$$+\frac{d(5a^2C-4abB+4Ab^2+b^2C)\,\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{2b^2f\,(a^2+b^2)}$$

$$-\frac{d\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}(15a^3Cd-3a^2b(4Bd+5cC)-8Ab^2(bc-ad)+ab^2(8Bc+7Cd}{4b^3f\,(a^2+b^2)}$$

$$-\frac{(c-id)^{5/2}(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(a-ib)^{3/2}}$$

$$-\frac{(c+id)^{5/2}(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(a+ib)^{3/2}}$$

```
[In] Int[((c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a
 + b*Tan[e + f*x])^(3/2),x]
```

```
[Out] -(((I*A + B - I*C)*(c - I*d)^(5/2)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e
+ f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a - I*b)^(3/2)*f)) -
((B - I*(A - C))*(c + I*d)^(5/2)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a + I*b)^(3/2)*f) + (Sq
rt[d]*(15*a^2*C*d^2 - 6*a*b*d*(5*c*C + 2*B*d) + b^2*(15*c^2*C + 20*B*c*d +
8*(A - C)*d^2))*ArcTanh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c
+ d*Tan[e + f*x]])])/(4*b^(7/2)*f) - (d*(15*a^3*C*d - 8*A*b^2*(b*c - a*d) -
 3*a^2*b*(5*c*C + 4*B*d) - b^3*(7*c*C + 4*B*d) + a*b^2*(8*B*c + 7*C*d))*Sqr
t[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(4*b^3*(a^2 + b^2)*f) + ((4
*A*b^2 - 4*a*b*B + 5*a^2*C + b^2*C)*d*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e
 + f*x])^(3/2))/(2*b^2*(a^2 + b^2)*f) - (2*(A*b^2 - a*(b*B - a*C))*(c + d*T
an[e + f*x])^(5/2))/(b*(a^2 + b^2)*f*Sqrt[a + b*Tan[e + f*x]])
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
```

```
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 95

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 223

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3726

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*d^2 + c*(c*C - B*d))*(a + b*Tan[e +
 f*x])^m*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
```

```
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3736

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[(a + b*ff*x)^m*(c + d*ff*x)^n*((A + B*ff*x + C*ff^2*x^2
)/(1 + ff^2*x^2)), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6857

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\text{integral} = -\frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{5/2}}{b\,(a^2+b^2)\,f\sqrt{a+b\tan(e+fx)}} + \frac{2\int \frac{(c+d\tan(e+fx))^{3/2}\left(\frac{1}{2}((bB-aC)(bc-5ad)+Ab(ac+5bd))-\frac{1}{2}b((A-C)(bc-ad)-B(ac+bd))\tan(e+fx)+\frac{1}{2}\left(4Ab^2-4abB+5a^2C+b^2C\right)d\,\mathrm{t}\right.}{\sqrt{a+b\tan(e+fx)}}}{b\,(a^2+b^2)}$$

$$= \frac{(4Ab^2-4abB+5a^2C+b^2C)\,d\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{2b^2\,(a^2+b^2)\,f} - \frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{5/2}}{b\,(a^2+b^2)\,f\sqrt{a+b\tan(e+fx)}} + \frac{\int \frac{\sqrt{c+d\tan(e+fx)}\left(\frac{1}{4}\left(-\left(\left(4Ab^2-4abB+5a^2C+b^2C\right)d(bc+3ad)\right)+4bc((bB-aC)(bc-5ad)+Ab(ac+5bd))\right)+b^2\left(2aAcd-2acCd-A\right.\right.}{}}{}$$

$$= -\frac{d(15a^3Cd - 8Ab^2(bc-ad) - 3a^2b(5cC+4Bd) - b^3(7cC+4Bd) + ab^2(8Bc+7Cd))\sqrt{a+b\tan}}{4b^3(a^2+b^2)f}$$
$$+ \frac{(4Ab^2 - 4abB + 5a^2C + b^2C)\,d\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{2b^2(a^2+b^2)f}$$
$$- \frac{2(Ab^2 - a(bB-aC))(c+d\tan(e+fx))^{5/2}}{b(a^2+b^2)f\sqrt{a+b\tan(e+fx)}}$$
$$+ \frac{\int \frac{\frac{1}{8}(15a^4Cd^3 - 6a^3bd^2(5cC+2Bd) + b^4c(8Bc^2+24Acd-9cCd-4Bd^2) + a^2b^2d(15c^2C+20Bcd+(8A+7C)d^2) - 2ab^3(4c^3C+12Bc^2d+}{}}{}$$

$$= -\frac{d(15a^3Cd - 8Ab^2(bc-ad) - 3a^2b(5cC+4Bd) - b^3(7cC+4Bd) + ab^2(8Bc+7Cd))\sqrt{a+b\tan}}{4b^3(a^2+b^2)f}$$
$$+ \frac{(4Ab^2 - 4abB + 5a^2C + b^2C)\,d\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{2b^2(a^2+b^2)f}$$
$$- \frac{2(Ab^2 - a(bB-aC))(c+d\tan(e+fx))^{5/2}}{b(a^2+b^2)f\sqrt{a+b\tan(e+fx)}}$$
$$+ \frac{\text{Subst}\left(\int \frac{\frac{1}{8}(15a^4Cd^3 - 6a^3bd^2(5cC+2Bd) + b^4c(8Bc^2+24Acd-9cCd-4Bd^2) + a^2b^2d(15c^2C+20Bcd+(8A+7C)d^2) - 2ab^3(4c^3C+}{}\right.}{}$$

$$= -\frac{d(15a^3Cd - 8Ab^2(bc-ad) - 3a^2b(5cC+4Bd) - b^3(7cC+4Bd) + ab^2(8Bc+7Cd))\sqrt{a+b\tan}}{4b^3(a^2+b^2)f}$$
$$+ \frac{(4Ab^2 - 4abB + 5a^2C + b^2C)\,d\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{2b^2(a^2+b^2)f}$$
$$- \frac{2(Ab^2 - a(bB-aC))(c+d\tan(e+fx))^{5/2}}{b(a^2+b^2)f\sqrt{a+b\tan(e+fx)}}$$
$$+ \frac{\text{Subst}\left(\int \left(\frac{(a^2+b^2)d(15a^2Cd^2 - 6abd(5cC+2Bd) + b^2(15c^2C+20Bcd+8(A-C)d^2))}{8\sqrt{a+bx}\sqrt{c+dx}} + \frac{-b^3(a(c^3C+3Bc^2d-3cCd^2-Bd^3-A(c^3}{}\right.\right.}{}$$

$$= -\frac{d(15a^3Cd-8Ab^2(bc-ad)-3a^2b(5cC+4Bd)-b^3(7cC+4Bd)+ab^2(8Bc+7Cd))\sqrt{a+b\,\mathrm{t}}}{4b^3(a^2+b^2)f} + \frac{(4Ab^2-4abB+5a^2C+b^2C)\,d\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{2b^2(a^2+b^2)f} - \frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{5/2}}{b(a^2+b^2)f\sqrt{a+b\tan(e+fx)}} + \frac{\mathrm{Subst}\Big(\int\frac{-b^3(a(c^3C+3Bc^2d-3cCd^2-Bd^3-A(c^3-3cd^2))-b((A-C)d(3c^2-d^2)+B(c^3-3cd^2)))+b^3(b(c^3C+3Bc^2d-3cCd^2-}{\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}}{b^3(a^2+b^2)f} + \frac{(d(15a^2Cd^2-6abd(5cC+2Bd)+b^2(15c^2C+20Bcd+8(A-C)d^2)))\,\mathrm{Subst}\Big(\int\frac{1}{\sqrt{a+bx}\sqrt{c+dx}}\,dx}{8b^3f}$$

$$= -\frac{d(15a^3Cd-8Ab^2(bc-ad)-3a^2b(5cC+4Bd)-b^3(7cC+4Bd)+ab^2(8Bc+7Cd))\sqrt{a+b\,\mathrm{t}}}{4b^3(a^2+b^2)f} + \frac{(4Ab^2-4abB+5a^2C+b^2C)\,d\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{2b^2(a^2+b^2)f} - \frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{5/2}}{b(a^2+b^2)f\sqrt{a+b\tan(e+fx)}} + \frac{\mathrm{Subst}\Big(\int\Big(\frac{-b^3(b(c^3C+3Bc^2d-3cCd^2-Bd^3)+a(Bc^3-3c^2Cd-3Bcd^2+Cd^3)+A(ad(3c^2-d^2)-b(c^3-3cd^2)))-ib^3(a(c^3C+3B}{2(i-x)\sqrt{a+bx}\sqrt{c+dx}}}{} + \frac{(d(15a^2Cd^2-6abd(5cC+2Bd)+b^2(15c^2C+20Bcd+8(A-C)d^2)))\,\mathrm{Subst}\Big(\int\frac{1}{\sqrt{c-\frac{ad}{b}+\frac{dx^2}{b}}}\,dx}{4b^4f}$$

$$= -\frac{d(15a^3Cd-8Ab^2(bc-ad)-3a^2b(5cC+4Bd)-b^3(7cC+4Bd)+ab^2(8Bc+7Cd))\sqrt{a+b\,\mathrm{t}}}{4b^3(a^2+b^2)f} + \frac{(4Ab^2-4abB+5a^2C+b^2C)\,d\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}{2b^2(a^2+b^2)f} - \frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{5/2}}{b(a^2+b^2)f\sqrt{a+b\tan(e+fx)}} + \frac{((iA+B-iC)(c-id)^3)\,\mathrm{Subst}\Big(\int\frac{1}{(i+x)\sqrt{a+bx}\sqrt{c+dx}}\,dx,x,\tan(e+fx)\Big)}{2(a-ib)f} + \frac{((ia+b)(A+iB-C)(c+id)^3)\,\mathrm{Subst}\Big(\int\frac{1}{(i-x)\sqrt{a+bx}\sqrt{c+dx}}\,dx,x,\tan(e+fx)\Big)}{2(a^2+b^2)f} + \frac{(d(15a^2Cd^2-6abd(5cC+2Bd)+b^2(15c^2C+20Bcd+8(A-C)d^2)))\,\mathrm{Subst}\Big(\int\frac{1}{1-\frac{dx^2}{b}}\,dx,x,\frac{\surd}{\surd}}{4b^4f}$$

$$
\begin{aligned}
&= \frac{\sqrt{d}(15a^2Cd^2 - 6abd(5cC + 2Bd) + b^2(15c^2C + 20Bcd + 8(A - C)d^2)) \operatorname{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{4b^{7/2}f} \\
&\quad - \frac{d(15a^3Cd - 8Ab^2(bc - ad) - 3a^2b(5cC + 4Bd) - b^3(7cC + 4Bd) + ab^2(8Bc + 7Cd))\sqrt{a + b\,\mathrm{ta}}}{4b^3(a^2 + b^2)f} \\
&\quad + \frac{(4Ab^2 - 4abB + 5a^2C + b^2C)\,d\sqrt{a + b\tan(e + fx)}(c + d\tan(e + fx))^{3/2}}{2b^2(a^2 + b^2)f} \\
&\quad - \frac{2(Ab^2 - a(bB - aC))(c + d\tan(e + fx))^{5/2}}{b(a^2 + b^2)f\sqrt{a + b\tan(e + fx)}} \\
&\quad + \frac{((iA + B - iC)(c - id)^3)\operatorname{Subst}\left(\int \frac{1}{-a+ib-(-c+id)x^2}\,dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{(a - ib)f} \\
&\quad + \frac{((ia + b)(A + iB - C)(c + id)^3)\operatorname{Subst}\left(\int \frac{1}{a+ib-(c+id)x^2}\,dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{(a^2 + b^2)f}
\end{aligned}
$$

$$
\begin{aligned}
&= -\frac{(iA + B - iC)(c - id)^{5/2}\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a - ib)^{3/2}f} \\
&\quad - \frac{(B - i(A - C))(c + id)^{5/2}\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a + ib)^{3/2}f} \\
&\quad + \frac{\sqrt{d}(15a^2Cd^2 - 6abd(5cC + 2Bd) + b^2(15c^2C + 20Bcd + 8(A - C)d^2)) \operatorname{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{4b^{7/2}f} \\
&\quad - \frac{d(15a^3Cd - 8Ab^2(bc - ad) - 3a^2b(5cC + 4Bd) - b^3(7cC + 4Bd) + ab^2(8Bc + 7Cd))\sqrt{a + b\,\mathrm{ta}}}{4b^3(a^2 + b^2)f} \\
&\quad + \frac{(4Ab^2 - 4abB + 5a^2C + b^2C)\,d\sqrt{a + b\tan(e + fx)}(c + d\tan(e + fx))^{3/2}}{2b^2(a^2 + b^2)f} \\
&\quad - \frac{2(Ab^2 - a(bB - aC))(c + d\tan(e + fx))^{5/2}}{b(a^2 + b^2)f\sqrt{a + b\tan(e + fx)}}
\end{aligned}
$$

Mathematica [B] (verified)

Both result and optimal contain complex but leaf count is larger than twice the leaf count of optimal. 1774 vs. 2(535) = 1070.

Time = 8.66 (sec) , antiderivative size = 1774, normalized size of antiderivative = 3.32

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{3/2}}\,dx=\frac{C(c+d\tan(e+fx))^{5/2}}{2bf\sqrt{a+b\tan(e+fx)}}$$

$$+\frac{\frac{(5bcC+4bBd-5aCd)(c+d\tan(e+fx))^{3/2}}{2bf\sqrt{a+b\tan(e+fx)}}+\frac{\frac{8b^2(iA+B-iC)(-c+id)^{5/2}\operatorname{arctanh}\left(\frac{\sqrt{-c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{-a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(-a+ib)^{3/2}f}-\frac{8b^2(B-i(A-C))(c+id)^{5/2}\operatorname{arctanh}}{(a+ib)^{3/2}f}}{}}{}$$

[In] Integrate[((c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^ 2))/(a + b*Tan[e + f*x])^(3/2),x]

[Out] (C*(c + d*Tan[e + f*x])^(5/2))/(2*b*f*Sqrt[a + b*Tan[e + f*x]]) + (((5*b*c* C + 4*b*B*d - 5*a*C*d)*(c + d*Tan[e + f*x])^(3/2))/(2*b*f*Sqrt[a + b*Tan[e + f*x]]) + ((8*b^2*(I*A + B - I*C)*(-c + I*d)^(5/2)*ArcTanh[(Sqrt[-c + I*d] *Sqrt[a + b*Tan[e + f*x]])/(Sqrt[-a + I*b]*Sqrt[c + d*Tan[e + f*x]])]/((-a + I*b)^(3/2)*f) - (8*b^2*(B - I*(A - C))*(c + I*d)^(5/2)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/ ((a + I*b)^(3/2)*f) + (8*b^2*(I*A + B - I*C)*(c - I*d)^2*Sqrt[c + d*Tan[e + f*x]])/((a - I*b)*f*Sqrt[a + b*Tan[e + f*x]]) + (8*b^2*(A + I*B - C)*(c + I*d)^2*Sqrt[c + d*Tan[e + f*x]])/((I*a - b)*f*Sqrt[a + b*Tan[e + f*x]]) + (30*a^2*C*d^2*Sqrt[c + d*Tan[e + f*x]]*(1 + (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d))))^(3/2)*(1 - (Sqrt[b]*S qrt[d]*ArcSinh[(Sqrt[b]*Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b*c - a*d]* Sqrt[(b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)])]*Sqrt[a + b*Tan[e + f*x]]) /(Sqrt[b*c - a*d]*Sqrt[(b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)]*Sqrt[1 + (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)))])))/(b*Sqrt[b/((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d))]*f*Sqrt [a + b*Tan[e + f*x]]*Sqrt[(b*(c + d*Tan[e + f*x]))/(b*c - a*d)]*(-1 - (b*d* (a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a* d))))) - (12*a*d*(5*c*C + 2*B*d)*Sqrt[c + d*Tan[e + f*x]]*(1 + (b*d*(a + b* Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d))))^(3/2)*(1 - (Sqrt[b]*Sqrt[d]*ArcSinh[(Sqrt[b]*Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b*c - a*d]*Sqrt[(b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)])]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b*c - a*d]*Sqrt[(b^2*c)/(b*c - a*d) - (a*b*d)/(b *c - a*d)]*Sqrt[1 + (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)))])))/(Sqrt[b/((b^2*c)/(b*c - a*d) - (a*b*d)/(b *c - a*d))]*f*Sqrt[a + b*Tan[e + f*x]]*Sqrt[(b*(c + d*Tan[e + f*x]))/(b*c - a*d)]*(-1 - (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d))))) + (2*b*(15*c^2*C + 20*B*c*d + 8*(A - C)*d^2)*Sqrt[c + d*Tan[e + f*x]]*(1 + (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/(

```
b*c - a*d) - (a*b*d)/(b*c - a*d)))^(3/2)*(1 - (Sqrt[b]*Sqrt[d]*ArcSinh[(Sq
rt[b]*Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b*c - a*d]*Sqrt[(b^2*c)/(b*c
- a*d) - (a*b*d)/(b*c - a*d)])]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b*c - a*d]*
Sqrt[(b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)]*Sqrt[1 + (b*d*(a + b*Tan[e
+ f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)))])))/(Sqr
t[b/((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d))]*f*Sqrt[a + b*Tan[e + f*x]]
*Sqrt[(b*(c + d*Tan[e + f*x]))/(b*c - a*d)]*(-1 - (b*d*(a + b*Tan[e + f*x])
)/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d))))))/(4*b))/(2*b)
```

Maple [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(fx+e))^{\frac{5}{2}}\left(A+B\tan(fx+e)+C\tan(fx+e)^{2}\right)}{(a+b\tan(fx+e))^{\frac{3}{2}}}dx$$

```
[In] int((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(3/2),x)
```

```
[Out] int((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(3/2),x)
```

Fricas [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{3/2}}\,dx=\text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(3/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{3/2}}\,dx=\int \frac{(c+d\tan(e+fx))^{\frac{5}{2}}\left(A+B\tan(e\right.}{(a+b\tan(e+}$$

```
[In] integrate((c+d*tan(f*x+e))**(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e))**(3/2),x)
```

```
[Out] Integral((c + d*tan(e + f*x))**(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)**
2)/(a + b*tan(e + f*x))**(3/2), x)
```

Maxima [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(3/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

Giac [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(3/2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{3/2}}\,dx = \text{Hanged}$$

```
[In] int(((c + d*tan(e + f*x))^(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x))^(3/2),x)
```

```
[Out] \text{Hanged}
```

3.144 $\int \frac{(c+d\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{5/2}}\,dx$

Optimal result 1998
Rubi [A] (verified) 1999
Mathematica [C] (verified) 2004
Maple [F(-1)] 2005
Fricas [F(-1)] 2005
Sympy [F] 2006
Maxima [F(-1)] 2006
Giac [F(-1)] 2006
Mupad [F(-1)] 2006

Optimal result

Integrand size = 49, antiderivative size = 545

$$\int \frac{(c+d\tan(e+fx))^{5/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{5/2}}\,dx =$$
$$-\frac{(iA+B-iC)(c-id)^{5/2}\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a-ib)^{5/2}f}$$
$$-\frac{(B-i(A-C))(c+id)^{5/2}\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a+ib)^{5/2}f}$$
$$+\frac{d^{3/2}(5bcC+2bBd-5aCd)\text{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{b^{7/2}f}$$
$$-\frac{d(2a^3bBd-5a^4Cd-2ab^3(2Ac-2cC-3Bd)+2a^2b^2(Bc-5Cd)-b^4(2Bc+(4A+C)d))\sqrt{a+b\tan(e}}{b^3(a^2+b^2)^2f}$$
$$+\frac{2(2a^3bBd-5a^4Cd-b^4(3Bc+5Ad)-2ab^3(3Ac-3cC-4Bd)+a^2b^2(3Bc+(A-11C)d))(c+d\tan(e+}{3b^2(a^2+b^2)^2f\sqrt{a+b\tan(e+fx)}}$$
$$-\frac{2(Ab^2-a(bB-aC))(c+d\tan(e+fx))^{5/2}}{3b(a^2+b^2)f(a+b\tan(e+fx))^{3/2}}$$

```
[Out] -(I*A+B-I*C)*(c-I*d)^(5/2)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-
I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/(a-I*b)^(5/2)/f-(B-I*(A-C))*(c+I*d)^(5/2
)*arctanh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e
))^(1/2))/(a+I*b)^(5/2)/f+d^(3/2)*(2*B*b*d-5*C*a*d+5*C*b*c)*arctanh(d^(1/2)
*(a+b*tan(f*x+e))^(1/2)/b^(1/2)/(c+d*tan(f*x+e))^(1/2))/b^(7/2)/f-d*(2*a^3*
b*B*d-5*a^4*C*d-2*a*b^3*(2*A*c-3*B*d-2*C*c)+2*a^2*b^2*(B*c-5*C*d)-b^4*(2*B*
c+(4*A+C)*d))*(a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)/b^3/(a^2+b^2)^2
/f+2/3*(2*a^3*b*B*d-5*a^4*C*d-b^4*(5*A*d+3*B*c)-2*a*b^3*(3*A*c-4*B*d-3*C*c)
```

```
+a^2*b^2*(3*B*c+(A-11*C)*d))*(c+d*tan(f*x+e))^(3/2)/b^2/(a^2+b^2)^2/f/(a+b*
tan(f*x+e))^(1/2)-2/3*(A*b^2-a*(B*b-C*a))*(c+d*tan(f*x+e))^(5/2)/b/(a^2+b^2
)/f/(a+b*tan(f*x+e))^(3/2)
```

Rubi [A] (verified)

Time = 12.75 (sec) , antiderivative size = 545, normalized size of antiderivative = 1.00, number of steps used = 15, number of rules used = 9, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.184, Rules used = {3726, 3728, 3736, 6857, 65, 223, 212, 95, 214}

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{5/2}}\,dx =$$

$$-\frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{5/2}}{3bf\,(a^2+b^2)\,(a+b\tan(e+fx))^{3/2}}$$

$$+\frac{2(c+d\tan(e+fx))^{3/2}\,(-5a^4Cd+2a^3bBd+a^2b^2(d(A-11C)+3Bc)-2ab^3(3Ac-4Bd-3cC)-b^4(5}{3b^2f\,(a^2+b^2)^2\,\sqrt{a+b\tan(e+fx)}}$$

$$-\frac{d\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}(-5a^4Cd+2a^3bBd+2a^2b^2(Bc-5Cd)-2ab^3(2Ac-3Bd-2c}{b^3f\,(a^2+b^2)^2}$$

$$-\frac{(c-id)^{5/2}(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(a-ib)^{5/2}}$$

$$-\frac{(c+id)^{5/2}(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(a+ib)^{5/2}}$$

$$+\frac{d^{3/2}(-5aCd+2bBd+5bcC)\text{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{b^{7/2}f}$$

```
[In] Int[((c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a
 + b*Tan[e + f*x])^(5/2),x]
```

```
[Out] -(((I*A + B - I*C)*(c - I*d)^(5/2)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e
+ f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a - I*b)^(5/2)*f)) -
((B - I*(A - C))*(c + I*d)^(5/2)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a + I*b)^(5/2)*f) + (d^
(3/2)*(5*b*c*C + 2*b*B*d - 5*a*C*d)*ArcTanh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x
]])/(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])])/(b^(7/2)*f) - (d*(2*a^3*b*B*d - 5*
a^4*C*d - 2*a*b^3*(2*A*c - 2*c*C - 3*B*d) + 2*a^2*b^2*(B*c - 5*C*d) - b^4*(
2*B*c + (4*A + C)*d))*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(b
^3*(a^2 + b^2)^2*f) + (2*(2*a^3*b*B*d - 5*a^4*C*d - b^4*(3*B*c + 5*A*d) - 2
*a*b^3*(3*A*c - 3*c*C - 4*B*d) + a^2*b^2*(3*B*c + (A - 11*C)*d))*(c + d*Tan
[e + f*x])^(3/2))/(3*b^2*(a^2 + b^2)^2*f*Sqrt[a + b*Tan[e + f*x]]) - (2*(A*
b^2 - a*(b*B - a*C))*(c + d*Tan[e + f*x])^(5/2))/(3*b*(a^2 + b^2)*f*(a + b*
Tan[e + f*x])^(3/2))
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 95

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 223

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3726

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*d^2 + c*(c*C - B*d))*(a + b*Tan[e +
 f*x])^m*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3736

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[(a + b*ff*x)^m*(c + d*ff*x)^n*((A + B*ff*x + C*ff^2*x^2
)/(1 + ff^2*x^2)), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6857

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\text{integral} = -\frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{5/2}}{3b\,(a^2+b^2)\,f(a+b\tan(e+fx))^{3/2}} + \frac{2\int \frac{(c+d\tan(e+fx))^{3/2}\left(\frac{1}{2}((bB-aC)(3bc-5ad)+Ab(3ac+5bd))-\frac{3}{2}b((A-C)(bc-ad)-B(ac+bd))\tan(e+fx)+\frac{1}{2}\left(2Ab^2-2abB+5a^2C+3b^2C\right.\right.}{(a+b\tan(e+fx))^{3/2}}}{3b\,(a^2+b^2)}$$

$$= \frac{2(2a^3bBd-5a^4Cd-b^4(3Bc+5Ad)-2ab^3(3Ac-3cC-4Bd)+a^2b^2(3Bc+(A-11C)d))\,(c+}{3b^2\,(a^2+b^2)^2\,f\sqrt{a+b\tan(e+fx)}} - \frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{5/2}}{3b\,(a^2+b^2)\,f(a+b\tan(e+fx))^{3/2}} + \frac{4\int \frac{\sqrt{c+d\tan(e+fx)}\left(\frac{1}{4}(b(ac+3bd)((bB-aC)(3bc-5ad)+Ab(3ac+5bd))+(bc-3ad)\left(2a^2bBd-5a^3Cd-Ab^2(3bc-ad)+3b^3(cC+Bd\right.\right.}{}}{}$$

$$= -\frac{d(2a^3bBd-5a^4Cd-2ab^3(2Ac-2cC-3Bd)+2a^2b^2(Bc-5Cd)-b^4(2Bc+(4A+C)d))\sqrt{a}}{b^3\left(a^2+b^2\right)^2 f}$$

$$+\frac{2(2a^3bBd-5a^4Cd-b^4(3Bc+5Ad)-2ab^3(3Ac-3cC-4Bd)+a^2b^2(3Bc+(A-11C)d))(c}{3b^2\left(a^2+b^2\right)^2 f\sqrt{a+b\tan(e+fx)}}$$

$$-\frac{2(Ab^2-a(bB-aC))\left(c+d\tan(e+fx)\right)^{5/2}}{3b\left(a^2+b^2\right)f(a+b\tan(e+fx))^{3/2}}$$

$$+\frac{4\int\frac{-\frac{3}{8}\left(5a^5Cd^3+10a^3b^2Cd^3-a^4bd^2(5cC+2Bd)-2a^2b^3\left(Ac^3-c^3C-3Bc^2d-3Acd^2+8cCd^2+3Bd^3\right)-b^5c\left(2c^2C+6Bcd-Cd^2-2A(c\right.\right.}{}}{}$$

$$= -\frac{d(2a^3bBd-5a^4Cd-2ab^3(2Ac-2cC-3Bd)+2a^2b^2(Bc-5Cd)-b^4(2Bc+(4A+C)d))\sqrt{a}}{b^3\left(a^2+b^2\right)^2 f}$$

$$+\frac{2(2a^3bBd-5a^4Cd-b^4(3Bc+5Ad)-2ab^3(3Ac-3cC-4Bd)+a^2b^2(3Bc+(A-11C)d))(c}{3b^2\left(a^2+b^2\right)^2 f\sqrt{a+b\tan(e+fx)}}$$

$$-\frac{2(Ab^2-a(bB-aC))\left(c+d\tan(e+fx)\right)^{5/2}}{3b\left(a^2+b^2\right)f(a+b\tan(e+fx))^{3/2}}$$

$$+\frac{4\text{Subst}\left(\int\frac{-\frac{3}{8}\left(5a^5Cd^3+10a^3b^2Cd^3-a^4bd^2(5cC+2Bd)-2a^2b^3\left(Ac^3-c^3C-3Bc^2d-3Acd^2+8cCd^2+3Bd^3\right)-b^5c\left(2c^2C+6Bcd-C\right.\right.}{}\right.}{}$$

$$= -\frac{d(2a^3bBd-5a^4Cd-2ab^3(2Ac-2cC-3Bd)+2a^2b^2(Bc-5Cd)-b^4(2Bc+(4A+C)d))\sqrt{a}}{b^3\left(a^2+b^2\right)^2 f}$$

$$+\frac{2(2a^3bBd-5a^4Cd-b^4(3Bc+5Ad)-2ab^3(3Ac-3cC-4Bd)+a^2b^2(3Bc+(A-11C)d))(c}{3b^2\left(a^2+b^2\right)^2 f\sqrt{a+b\tan(e+fx)}}$$

$$-\frac{2(Ab^2-a(bB-aC))\left(c+d\tan(e+fx)\right)^{5/2}}{3b\left(a^2+b^2\right)f(a+b\tan(e+fx))^{3/2}}$$

$$+\frac{4\text{Subst}\left(\int\left(\frac{3\left(a^2+b^2\right)^2d^2(5bcC+2bBd-5aCd)}{8\sqrt{a+bx}\sqrt{c+dx}}+\frac{3\left(-b^3\left(b^2\left(Ac^3-c^3C-3Bc^2d-3Acd^2+3cCd^2+Bd^3\right)+a^2\left(c^3C+3Bc^2d-3cCd\right.\right.\right.}{}\right.\right.}{}$$

$$= -\frac{d(2a^3bBd-5a^4Cd-2ab^3(2Ac-2cC-3Bd)+2a^2b^2(Bc-5Cd)-b^4(2Bc+(4A+C)d))\sqrt{}}{b^3\left(a^2+b^2\right)^2 f} + \frac{2(2a^3bBd-5a^4Cd-b^4(3Bc+5Ad)-2ab^3(3Ac-3cC-4Bd)+a^2b^2(3Bc+(A-11C)d))\,(}{3b^2\left(a^2+b^2\right)^2 f\sqrt{a+b\tan(e+fx)}} - \frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{5/2}}{3b\left(a^2+b^2\right)f(a+b\tan(e+fx))^{3/2}} + \frac{\text{Subst}\Big(\int\frac{-b^3(b^2(Ac^3-c^3C-3Bc^2d-3Acd^2+3cCd^2+Bd^3)+a^2(c^3C+3Bc^2d-3cCd^2-Bd^3-A(c^3-3cd^2))-2ab((A-C)d(3c^2-}{}}{} + \frac{(d^2(5bcC+2bBd-5aCd))\,\text{Subst}\Big(\int\frac{1}{\sqrt{a+bx}\sqrt{c+dx}}\,dx,x,\tan(e+fx)\Big)}{2b^3f}$$

$$= -\frac{d(2a^3bBd-5a^4Cd-2ab^3(2Ac-2cC-3Bd)+2a^2b^2(Bc-5Cd)-b^4(2Bc+(4A+C)d))\sqrt{}}{b^3\left(a^2+b^2\right)^2 f} + \frac{2(2a^3bBd-5a^4Cd-b^4(3Bc+5Ad)-2ab^3(3Ac-3cC-4Bd)+a^2b^2(3Bc+(A-11C)d))\,(}{3b^2\left(a^2+b^2\right)^2 f\sqrt{a+b\tan(e+fx)}} - \frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{5/2}}{3b\left(a^2+b^2\right)f(a+b\tan(e+fx))^{3/2}} + \frac{\text{Subst}\Big(\int\Big(\frac{-ib^3(b^2(Ac^3-c^3C-3Bc^2d-3Acd^2+3cCd^2+Bd^3)+a^2(c^3C+3Bc^2d-3cCd^2-Bd^3-A(c^3-3cd^2))-2ab((A-C)d(3c}{}}{} + \frac{(d^2(5bcC+2bBd-5aCd))\,\text{Subst}\Big(\int\frac{1}{\sqrt{c-\frac{ad}{b}+\frac{dx^2}{b}}}\,dx,x,\sqrt{a+b\tan(e+fx)}\Big)}{b^4f}$$

$$= -\frac{d(2a^3bBd-5a^4Cd-2ab^3(2Ac-2cC-3Bd)+2a^2b^2(Bc-5Cd)-b^4(2Bc+(4A+C)d))\sqrt{}}{b^3\left(a^2+b^2\right)^2 f} + \frac{2(2a^3bBd-5a^4Cd-b^4(3Bc+5Ad)-2ab^3(3Ac-3cC-4Bd)+a^2b^2(3Bc+(A-11C)d))\,(}{3b^2\left(a^2+b^2\right)^2 f\sqrt{a+b\tan(e+fx)}} - \frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{5/2}}{3b\left(a^2+b^2\right)f(a+b\tan(e+fx))^{3/2}} + \frac{((iA+B-iC)(c-id)^3)\,\text{Subst}\Big(\int\frac{1}{(i+x)\sqrt{a+bx}\sqrt{c+dx}}\,dx,x,\tan(e+fx)\Big)}{2(a-ib)^2f} + \frac{(d^2(5bcC+2bBd-5aCd))\,\text{Subst}\Big(\int\frac{1}{1-\frac{dx^2}{b}}\,dx,x,\frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\Big)}{b^4f} + \frac{(-ib^3(b^2(Ac^3-c^3C-3Bc^2d-3Acd^2+3cCd^2+Bd^3)+a^2(c^3C+3Bc^2d-3cCd^2-Bd^3-A}{}$$

$$= \frac{d^{3/2}(5bcC+2bBd-5aCd)\operatorname{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{b^{7/2}f}$$
$$-\frac{d(2a^3bBd-5a^4Cd-2ab^3(2Ac-2cC-3Bd)+2a^2b^2(Bc-5Cd)-b^4(2Bc+(4A+C)d))\sqrt{a}}{b^3(a^2+b^2)^2f}$$
$$+\frac{2(2a^3bBd-5a^4Cd-b^4(3Bc+5Ad)-2ab^3(3Ac-3cC-4Bd)+a^2b^2(3Bc+(A-11C)d))(c}{3b^2(a^2+b^2)^2f\sqrt{a+b\tan(e+fx)}}$$
$$-\frac{2(Ab^2-a(bB-aC))(c+d\tan(e+fx))^{5/2}}{3b(a^2+b^2)f(a+b\tan(e+fx))^{3/2}}$$
$$+\frac{((iA+B-iC)(c-id)^3)\operatorname{Subst}\left(\int\frac{1}{-a+ib-(-c+id)x^2}\,dx,x,\frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{(a-ib)^2f}$$
$$+\frac{(-ib^3(b^2(Ac^3-c^3C-3Bc^2d-3Acd^2+3cCd^2+Bd^3)+a^2(c^3C+3Bc^2d-3cCd^2-Bd^3-A(}{}$$

$$= -\frac{(iA+B-iC)(c-id)^{5/2}\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a-ib)^{5/2}f}$$
$$-\frac{(B-i(A-C))(c+id)^{5/2}\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a+ib)^{5/2}f}$$
$$+\frac{d^{3/2}(5bcC+2bBd-5aCd)\operatorname{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{b^{7/2}f}$$
$$-\frac{d(2a^3bBd-5a^4Cd-2ab^3(2Ac-2cC-3Bd)+2a^2b^2(Bc-5Cd)-b^4(2Bc+(4A+C)d))\sqrt{a}}{b^3(a^2+b^2)^2f}$$
$$+\frac{2(2a^3bBd-5a^4Cd-b^4(3Bc+5Ad)-2ab^3(3Ac-3cC-4Bd)+a^2b^2(3Bc+(A-11C)d))(c}{3b^2(a^2+b^2)^2f\sqrt{a+b\tan(e+fx)}}$$
$$-\frac{2(Ab^2-a(bB-aC))(c+d\tan(e+fx))^{5/2}}{3b(a^2+b^2)f(a+b\tan(e+fx))^{3/2}}$$

Mathematica [C] (verified)

Result contains higher order function than in optimal. Order 5 vs. order 3 in optimal.

Time = 7.63 (sec) , antiderivative size = 802, normalized size of antiderivative = 1.47

$$\int\frac{(c+d\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{5/2}}\,dx = \frac{C(c+d\tan(e+fx))^{5/2}}{bf(a+b\tan(e+fx))^{3/2}}$$
$$+\frac{-\frac{2b(A-iB-C)(c-id)(c+d\tan(e+fx))^{3/2}}{3(ia+b)f(a+b\tan(e+fx))^{3/2}}+\frac{2b(A+iB-C)(c+id)(c+d\tan(e+fx))^{3/2}}{3(ia-b)f(a+b\tan(e+fx))^{3/2}}-\frac{10cC(bc-ad)\,\text{Hypergeometric2F1}\left(-\frac{3}{2},-\frac{3}{2},-\frac{1}{2},-\frac{d(a}{}\right.}{3bf(a+b\tan(e+fx))^{3/2}\sqrt{\frac{b(c+d}{}}}}{}$$

```
[In] Integrate[((c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/(a + b*Tan[e + f*x])^(5/2),x]
```

```
[Out] (C*(c + d*Tan[e + f*x])^(5/2))/(b*f*(a + b*Tan[e + f*x])^(3/2)) + ((-2*b*(A
 - I*B - C)*(c - I*d)*(c + d*Tan[e + f*x])^(3/2))/(3*(I*a + b)*f*(a + b*Tan
[e + f*x])^(3/2)) + (2*b*(A + I*B - C)*(c + I*d)*(c + d*Tan[e + f*x])^(3/2)
)/(3*(I*a - b)*f*(a + b*Tan[e + f*x])^(3/2)) - (10*c*C*(b*c - a*d)*Hypergeo
metric2F1[-3/2, -3/2, -1/2, -((d*(a + b*Tan[e + f*x]))/(b*c - a*d))]*Sqrt[c
 + d*Tan[e + f*x]])/(3*b*f*(a + b*Tan[e + f*x])^(3/2)*Sqrt[(b*(c + d*Tan[e
+ f*x]))/(b*c - a*d)]) - (4*B*d*(b*c - a*d)*Hypergeometric2F1[-3/2, -3/2, -
1/2, -((d*(a + b*Tan[e + f*x]))/(b*c - a*d))]*Sqrt[c + d*Tan[e + f*x]])/(3*
b*f*(a + b*Tan[e + f*x])^(3/2)*Sqrt[(b*(c + d*Tan[e + f*x]))/(b*c - a*d)])
+ (10*a*C*d*(b*c - a*d)*Hypergeometric2F1[-3/2, -3/2, -1/2, -((d*(a + b*Tan
[e + f*x]))/(b*c - a*d))]*Sqrt[c + d*Tan[e + f*x]])/(3*b^2*f*(a + b*Tan[e +
 f*x])^(3/2)*Sqrt[(b*(c + d*Tan[e + f*x]))/(b*c - a*d)]) + (2*b*(I*A + B -
I*C)*(c - I*d)^2*((Sqrt[-c + I*d]*ArcTanh[(Sqrt[-c + I*d]*Sqrt[a + b*Tan[e
+ f*x]])/(Sqrt[-a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(-a + I*b)^(3/2) + Sqr
t[c + d*Tan[e + f*x]]/((a - I*b)*Sqrt[a + b*Tan[e + f*x]])))/((a - I*b)*f)
- (2*b*(A + I*B - C)*(c + I*d)^2*((Sqrt[c + I*d]*ArcTanh[(Sqrt[c + I*d]*Sqr
t[a + b*Tan[e + f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(a + I*b)
^(3/2) - Sqrt[c + d*Tan[e + f*x]]/((a + I*b)*Sqrt[a + b*Tan[e + f*x]])))/((
I*a - b)*f))/(2*b)
```

Maple [F(-1)]

Timed out.

$$\int \frac{(c + d\tan(fx + e))^{\frac{5}{2}} \left(A + B\tan(fx + e) + C\tan(fx + e)^2\right)}{(a + b\tan(fx + e))^{\frac{5}{2}}}dx$$

```
[In] int((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(5/2),x)
```

```
[Out] int((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(5/2),x)
```

Fricas [F(-1)]

Timed out.

$$\int \frac{(c + d\tan(e + fx))^{5/2} (A + B\tan(e + fx) + C\tan^2(e + fx))}{(a + b\tan(e + fx))^{5/2}} dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(5/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int \frac{(c+d\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{5/2}}\,dx = \int \frac{(c+d\tan{(e+fx)})^{\frac{5}{2}}(A+B\tan{(e}}{(a+b\tan{(e+}}$$

```
[In] integrate((c+d*tan(f*x+e))**(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e))**(5/2),x)
```

```
[Out] Integral((c + d*tan(e + f*x))**(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)**
2)/(a + b*tan(e + f*x))**(5/2), x)
```

Maxima [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{5/2}}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(5/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

Giac [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{5/2}}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(5/2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{5/2}}\,dx = \text{Hanged}$$

```
[In] int(((c + d*tan(e + f*x))^(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x))^(5/2),x)
```

```
[Out] \text{Hanged}
```

3.145 $\int \frac{(c+d\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{7/2}}\,dx$

Optimal result 2007
Rubi [A] (verified) 2008
Mathematica [C] (verified) 2014
Maple [F(-1)] 2015
Fricas [F(-1)] 2015
Sympy [F] 2015
Maxima [F(-1)] 2015
Giac [F(-1)] 2016
Mupad [F(-1)] 2016

Optimal result

Integrand size = 49, antiderivative size = 590

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{7/2}}\,dx =$$
$$-\frac{(iA+B-iC)(c-id)^{5/2}\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a-ib)^{7/2}f}$$
$$-\frac{(B-i(A-C))(c+id)^{5/2}\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a+ib)^{7/2}f}$$
$$+\frac{2Cd^{5/2}\text{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{b^{7/2}f}$$
$$-\frac{2(a^6Cd^2+3a^4b^2Cd^2-3a^2b^4(c^2C+2Bcd-2Cd^2-A(c^2-d^2))+b^6(c(cC+2Bd)-A(c^2-d^2))-a^3b^3}{b^3(a^2+b^2)^3f\sqrt{a+b\tan(e}}$$
$$-\frac{2(a^4Cd+b^4(Bc+Ad)+2ab^3(Ac-cC-Bd)-a^2b^2(Bc+(A-3C)d))(c+d\tan(e+fx))^{3/2}}{3b^2(a^2+b^2)^2f(a+b\tan(e+fx))^{3/2}}$$
$$-\frac{2(Ab^2-a(bB-aC))(c+d\tan(e+fx))^{5/2}}{5b(a^2+b^2)f(a+b\tan(e+fx))^{5/2}}$$

```
[Out] -(I*A+B-I*C)*(c-I*d)^(5/2)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-
I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/(a-I*b)^(7/2)/f-(B-I*(A-C))*(c+I*d)^(5/2
)*arctanh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e
))^(1/2))/(a+I*b)^(7/2)/f+2*C*d^(5/2)*arctanh(d^(1/2)*(a+b*tan(f*x+e))^(1/2
)/b^(1/2)/(c+d*tan(f*x+e))^(1/2))/b^(7/2)/f-2*(a^6*C*d^2+3*a^4*b^2*C*d^2-3*
a^2*b^4*(c^2*C+2*B*c*d-2*C*d^2-A*(c^2-d^2))+b^6*(c*(2*B*d+C*c)-A*(c^2-d^2))
-a^3*b^3*(2*c*(A-C)*d+B*(c^2-d^2))+3*a*b^5*(2*c*(A-C)*d+B*(c^2-d^2)))*(c+d*
tan(f*x+e))^(1/2)/b^3/(a^2+b^2)^3/f/(a+b*tan(f*x+e))^(1/2)-2/3*(a^4*C*d+b^4
```

```
*(A*d+B*c)+2*a*b^3*(A*c-B*d-C*c)-a^2*b^2*(B*c+(A-3*C)*d))*(c+d*tan(f*x+e))^
(3/2)/b^2/(a^2+b^2)^2/f/(a+b*tan(f*x+e))^(3/2)-2/5*(A*b^2-a*(B*b-C*a))*(c+d
*tan(f*x+e))^(5/2)/b/(a^2+b^2)/f/(a+b*tan(f*x+e))^(5/2)
```

Rubi [A] (verified)

Time = 14.55 (sec) , antiderivative size = 590, normalized size of antiderivative = 1.00, number of steps used = 15, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}} = 0.163$, Rules used = {3726, 3736, 6857, 65, 223, 212, 95, 214}

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{7/2}}\,dx =$$
$$-\frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{5/2}}{5bf\,(a^2+b^2)\,(a+b\tan(e+fx))^{5/2}}$$
$$-\frac{2(c+d\tan(e+fx))^{3/2}\,(a^4Cd-a^2b^2(d(A-3C)+Bc)+2ab^3(Ac-Bd-cC)+b^4(Ad+Bc))}{3b^2f\,(a^2+b^2)^2\,(a+b\tan(e+fx))^{3/2}}$$
$$-\frac{2\sqrt{c+d\tan(e+fx)}(a^6Cd^2+3a^4b^2Cd^2-a^3b^3(2cd(A-C)+B(c^2-d^2))-3a^2b^4(-A(c^2-d^2)+2Bcd+}{b^3f\,(a^2+b^2)^3\,\sqrt{a+b\tan(e-}}$$
$$-\frac{(c-id)^{5/2}(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(a-ib)^{7/2}}$$
$$-\frac{(c+id)^{5/2}(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(a+ib)^{7/2}}$$
$$+\frac{2Cd^{5/2}\text{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{b^{7/2}f}$$

```
[In] Int[((c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a
 + b*Tan[e + f*x])^(7/2),x]
```

```
[Out] -(((I*A + B - I*C)*(c - I*d)^(5/2)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e
+ f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a - I*b)^(7/2)*f)) -
((B - I*(A - C))*(c + I*d)^(5/2)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a + I*b)^(7/2)*f) + (2*
C*d^(5/2)*ArcTanh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c + d*Ta
n[e + f*x]])])/(b^(7/2)*f) - (2*(a^6*C*d^2 + 3*a^4*b^2*C*d^2 - 3*a^2*b^4*(c
^2*C + 2*B*c*d - 2*C*d^2 - A*(c^2 - d^2)) + b^6*(c*(c*C + 2*B*d) - A*(c^2 -
 d^2)) - a^3*b^3*(2*c*(A - C)*d + B*(c^2 - d^2)) + 3*a*b^5*(2*c*(A - C)*d +
 B*(c^2 - d^2)))*Sqrt[c + d*Tan[e + f*x]])/(b^3*(a^2 + b^2)^3*f*Sqrt[a + b*
Tan[e + f*x]]) - (2*(a^4*C*d + b^4*(B*c + A*d) + 2*a*b^3*(A*c - c*C - B*d)
- a^2*b^2*(B*c + (A - 3*C)*d))*(c + d*Tan[e + f*x])^(3/2))/(3*b^2*(a^2 + b^
2)^2*f*(a + b*Tan[e + f*x])^(3/2)) - (2*(A*b^2 - a*(b*B - a*C))*(c + d*Tan[
e + f*x])^(5/2))/(5*b*(a^2 + b^2)*f*(a + b*Tan[e + f*x])^(5/2))
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 95

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 223

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3726

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*d^2 + c*(c*C - B*d))*(a + b*Tan[e +
 f*x])^m*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3736

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[(a + b*ff*x)^m*(c + d*ff*x)^n*((A + B*ff*x + C*ff^2*x^2
)/(1 + ff^2*x^2)), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6857

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\text{integral} = -\frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{5/2}}{5b\,(a^2+b^2)\,f(a+b\tan(e+fx))^{5/2}} + \frac{2\int \frac{(c+d\tan(e+fx))^{3/2}\left(\frac{5}{2}((bB-aC)(bc-ad)+Ab(ac+bd))-\frac{5}{2}b((A-C)(bc-ad)-B(ac+bd))\tan(e+fx)+\frac{5}{2}(a^2+b^2)Cd\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{5/2}}\,dx}{5b\,(a^2+b^2)}$$

$$= -\frac{2(a^4Cd+b^4(Bc+Ad)+2ab^3(Ac-cC-Bd)-a^2b^2(Bc+(A-3C)d))\,(c+d\tan(e+fx))^{3/2}}{3b^2\,(a^2+b^2)^2\,f(a+b\tan(e+fx))^{3/2}} - \frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{5/2}}{5b\,(a^2+b^2)\,f(a+b\tan(e+fx))^{5/2}} + \frac{4\int \frac{\sqrt{c+d\tan(e+fx)}\left(\frac{15}{4}(b(ac+bd)((bB-aC)(bc-ad)+Ab(ac+bd))-(bc-ad)(a^3Cd+Ab^2(bc-ad)-b^3(cC+Bd)-ab^2(Bc-2Cd)))\right.}{(a+b\tan}}{15b^2\,(}$$

$$= -\frac{2(a^6Cd^2+3a^4b^2Cd^2-3a^2b^4(c^2C+2Bcd-2Cd^2-A(c^2-d^2))+b^6(c(cC+2Bd)-A(c^2-d^2}{b^3\,(a^2+b^2)^3\,f\sqrt{a-}} - \frac{2(a^4Cd+b^4(Bc+Ad)+2ab^3(Ac-cC-Bd)-a^2b^2(Bc+(A-3C)d))\,(c+d\tan(e+fx))^{3/2}}{3b^2\,(a^2+b^2)^2\,f(a+b\tan(e+fx))^{3/2}} - \frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{5/2}}{5b\,(a^2+b^2)\,f(a+b\tan(e+fx))^{5/2}} + \frac{8\int \frac{15}{8}(a^6Cd^3+3a^4b^2Cd^3-b^6(Bc^3+3Ac^2d-3c^2Cd-3Bcd^2-Ad^3)-3ab^5(Ac^3-c^3C-3Bc^2d-3Acd^2+3cCd^2+Bd^3)+3a^2b^4(Bc^3+3}{}$$

$$= -\frac{2(a^6Cd^2+3a^4b^2Cd^2-3a^2b^4(c^2C+2Bcd-2Cd^2-A(c^2-d^2))+b^6(c(cC+2Bd)-A(c^2-a}{b^3(a^2+b^2)^3 f\sqrt{a}$$

$$-\frac{2(a^4Cd+b^4(Bc+Ad)+2ab^3(Ac-cC-Bd)-a^2b^2(Bc+(A-3C)d))(c+d\tan(e+fx))^{3/}}{3b^2(a^2+b^2)^2 f(a+b\tan(e+fx))^{3/2}}$$

$$-\frac{2(Ab^2-a(bB-aC))(c+d\tan(e+fx))^{5/2}}{5b(a^2+b^2)f(a+b\tan(e+fx))^{5/2}}$$

$$+\frac{8\text{Subst}\left(\int \frac{\frac{15}{8}(a^6Cd^3+3a^4b^2Cd^3-b^6(Bc^3+3Ac^2d-3c^2Cd-3Bcd^2-Ad^3)-3ab^5(Ac^3-c^3C-3Bc^2d-3Acd^2+3cCd^2+Bd^3)+3c}{}}{}$$

$$= -\frac{2(a^6Cd^2+3a^4b^2Cd^2-3a^2b^4(c^2C+2Bcd-2Cd^2-A(c^2-d^2))+b^6(c(cC+2Bd)-A(c^2-a}{b^3(a^2+b^2)^3 f\sqrt{a}$$

$$-\frac{2(a^4Cd+b^4(Bc+Ad)+2ab^3(Ac-cC-Bd)-a^2b^2(Bc+(A-3C)d))(c+d\tan(e+fx))^{3/}}{3b^2(a^2+b^2)^2 f(a+b\tan(e+fx))^{3/2}}$$

$$-\frac{2(Ab^2-a(bB-aC))(c+d\tan(e+fx))^{5/2}}{5b(a^2+b^2)f(a+b\tan(e+fx))^{5/2}}$$

$$+\frac{8\text{Subst}\left(\int \left(\frac{15(a^2+b^2)^3Cd^3}{8\sqrt{a+bx}\sqrt{c+dx}}+\frac{15(-b^3(3ab^2(Ac^3-c^3C-3Bc^2d-3Acd^2+3cCd^2+Bd^3)+a^3(c^3C+3Bc^2d-3cCd^2-Bd^3-A(c}{}\right.\right.}{}$$

$$= -\frac{2(a^6Cd^2+3a^4b^2Cd^2-3a^2b^4(c^2C+2Bcd-2Cd^2-A(c^2-d^2))+b^6(c(cC+2Bd)-A(c^2-a}{b^3(a^2+b^2)^3 f\sqrt{a}$$

$$-\frac{2(a^4Cd+b^4(Bc+Ad)+2ab^3(Ac-cC-Bd)-a^2b^2(Bc+(A-3C)d))(c+d\tan(e+fx))^{3/}}{3b^2(a^2+b^2)^2 f(a+b\tan(e+fx))^{3/2}}$$

$$-\frac{2(Ab^2-a(bB-aC))(c+d\tan(e+fx))^{5/2}}{5b(a^2+b^2)f(a+b\tan(e+fx))^{5/2}}$$

$$+\frac{\text{Subst}\left(\int \frac{-b^3(3ab^2(Ac^3-c^3C-3Bc^2d-3Acd^2+3cCd^2+Bd^3)+a^3(c^3C+3Bc^2d-3cCd^2-Bd^3-A(c^3-3cd^2))-3a^2b((A-C)d(3}{}\right.}{}$$

$$+\frac{(Cd^3)\,\text{Subst}\left(\int \frac{1}{\sqrt{a+bx}\sqrt{c+dx}}\,dx,x,\tan(e+fx)\right)}{b^3 f}$$

$$= -\frac{2(a^6Cd^2+3a^4b^2Cd^2-3a^2b^4(c^2C+2Bcd-2Cd^2-A(c^2-d^2))+b^6(c(cC+2Bd)-A(c^2-d^2}{b^3\left(a^2+b^2\right)^3f\sqrt{a-}}$$

$$-\frac{2(a^4Cd+b^4(Bc+Ad)+2ab^3(Ac-cC-Bd)-a^2b^2(Bc+(A-3C)d))\left(c+d\tan(e+fx)\right)^{3/2}}{3b^2\left(a^2+b^2\right)^2f(a+b\tan(e+fx))^{3/2}}$$

$$-\frac{2(Ab^2-a(bB-aC))\left(c+d\tan(e+fx)\right)^{5/2}}{5b\left(a^2+b^2\right)f(a+b\tan(e+fx))^{5/2}}$$

$$+\frac{\text{Subst}\left(\int\left(\frac{b^3(3a^2b(Ac^3-c^3C-3Bc^2d-3Acd^2+3cCd^2+Bd^3)+b^3(c^3C+3Bc^2d-3cCd^2-Bd^3-A(c^3-3cd^2))-a^3((A-C)d(3c^2-}{}\right.\right.}{}$$

$$+\frac{(2Cd^3)\,\text{Subst}\left(\int\frac{1}{\sqrt{c-\frac{ad}{b}+\frac{dx^2}{b}}}\,dx,x,\sqrt{a+b\tan(e+fx)}\right)}{b^4f}$$

$$= -\frac{2(a^6Cd^2+3a^4b^2Cd^2-3a^2b^4(c^2C+2Bcd-2Cd^2-A(c^2-d^2))+b^6(c(cC+2Bd)-A(c^2-d^2}{b^3\left(a^2+b^2\right)^3f\sqrt{a-}}$$

$$-\frac{2(a^4Cd+b^4(Bc+Ad)+2ab^3(Ac-cC-Bd)-a^2b^2(Bc+(A-3C)d))\left(c+d\tan(e+fx)\right)^{3/2}}{3b^2\left(a^2+b^2\right)^2f(a+b\tan(e+fx))^{3/2}}$$

$$-\frac{2(Ab^2-a(bB-aC))\left(c+d\tan(e+fx)\right)^{5/2}}{5b\left(a^2+b^2\right)f(a+b\tan(e+fx))^{5/2}}$$

$$+\frac{((A-iB-C)(c-id)^3)\,\text{Subst}\left(\int\frac{1}{(i+x)\sqrt{a+bx}\sqrt{c+dx}}\,dx,x,\tan(e+fx)\right)}{2(ia+b)^3f}$$

$$+\frac{((A+iB-C)(c+id)^3)\,\text{Subst}\left(\int\frac{1}{(i-x)\sqrt{a+bx}\sqrt{c+dx}}\,dx,x,\tan(e+fx)\right)}{2(ia-b)^3f}$$

$$+\frac{(2Cd^3)\,\text{Subst}\left(\int\frac{1}{1-\frac{dx^2}{b}}\,dx,x,\frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{b^4f}$$

$$= \frac{2Cd^{5/2}\operatorname{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{b^{7/2}f} - \frac{2(a^6Cd^2+3a^4b^2Cd^2-3a^2b^4(c^2C+2Bcd-2Cd^2-A(c^2-d^2))+b^6(c(cC+2Bd)-A(c^2-d}{b^3(a^2+b^2)^3f\sqrt{a}} - \frac{2(a^4Cd+b^4(Bc+Ad)+2ab^3(Ac-cC-Bd)-a^2b^2(Bc+(A-3C)d))(c+d\tan(e+fx))^{3/}}{3b^2(a^2+b^2)^2f(a+b\tan(e+fx))^{3/2}} - \frac{2(Ab^2-a(bB-aC))(c+d\tan(e+fx))^{5/2}}{5b(a^2+b^2)f(a+b\tan(e+fx))^{5/2}} + \frac{((A-iB-C)(c-id)^3)\operatorname{Subst}\left(\int\frac{1}{-a+ib-(-c+id)x^2}dx,x,\frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{(ia+b)^3f} + \frac{((A+iB-C)(c+id)^3)\operatorname{Subst}\left(\int\frac{1}{a+ib-(c+id)x^2}dx,x,\frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{(ia-b)^3f}$$

$$= -\frac{(iA+B-iC)(c-id)^{5/2}\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a-ib)^{7/2}f} - \frac{(B-i(A-C))(c+id)^{5/2}\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a+ib)^{7/2}f} + \frac{2Cd^{5/2}\operatorname{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{b^{7/2}f} - \frac{2(a^6Cd^2+3a^4b^2Cd^2-3a^2b^4(c^2C+2Bcd-2Cd^2-A(c^2-d^2))+b^6(c(cC+2Bd)-A(c^2-d}{b^3(a^2+b^2)^3f\sqrt{a}} - \frac{2(a^4Cd+b^4(Bc+Ad)+2ab^3(Ac-cC-Bd)-a^2b^2(Bc+(A-3C)d))(c+d\tan(e+fx))^{3/}}{3b^2(a^2+b^2)^2f(a+b\tan(e+fx))^{3/2}} - \frac{2(Ab^2-a(bB-aC))(c+d\tan(e+fx))^{5/2}}{5b(a^2+b^2)f(a+b\tan(e+fx))^{5/2}}$$

Mathematica [C] (verified)

Result contains higher order function than in optimal. Order 5 vs. order 3 in optimal.

Time = 7.14 (sec) , antiderivative size = 641, normalized size of antiderivative = 1.09

$$\int \frac{(c+d\tan(e+fx))^{5/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{7/2}}\,dx = \frac{(B+i(A-C))(c+d\tan(e+fx))^{5/2}}{5(a-ib)f(a+b\tan(e+fx))^{5/2}} - \frac{(iA-B-iC)(c+d\tan(e+fx))^{5/2}}{5(a+ib)f(a+b\tan(e+fx))^{5/2}} - \frac{2C(bc-ad)^2\,\text{Hypergeometric2F1}\left(-\frac{5}{2},-\frac{5}{2},-\frac{3}{2},-\frac{d(a+b\tan(e+fx))}{bc-ad}\right)\sqrt{c+d\tan(e+fx)}}{5b^3f(a+b\tan(e+fx))^{5/2}\sqrt{\frac{b(c+d\tan(e+fx))}{bc-ad}}} + \frac{(A-iB-C)(ic+d)\left(\frac{(c+d\tan(e+fx))^{3/2}}{(a-ib)(a+b\tan(e+fx))^{3/2}}+\frac{3(c-id)\left(\frac{\sqrt{-c+id}\,\text{arctanh}\left(\frac{\sqrt{-c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{-a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(-a+ib)^{3/2}}+\frac{\sqrt{c+d\tan(e+fx)}}{(a-ib)\sqrt{a+b\tan(e+fx)}}\right)}{a-ib}\right)}{3(a-ib)f} - \frac{(A+iB-C)(ic-d)\left(\frac{(c+d\tan(e+fx))^{3/2}}{(a+ib)(a+b\tan(e+fx))^{3/2}}-\frac{3(c+id)\left(\frac{\sqrt{c+id}\,\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a+ib)^{3/2}}-\frac{\sqrt{c+d\tan(e+fx)}}{(a+ib)\sqrt{a+b\tan(e+fx)}}\right)}{a+ib}\right)}{3(a+ib)f}$$

```
[In] Integrate[((c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/(a + b*Tan[e + f*x])^(7/2),x]
```

```
[Out] ((B + I*(A - C))*(c + d*Tan[e + f*x])^(5/2))/(5*(a - I*b)*f*(a + b*Tan[e +
f*x])^(5/2)) - ((I*A - B - I*C)*(c + d*Tan[e + f*x])^(5/2))/(5*(a + I*b)*f*
(a + b*Tan[e + f*x])^(5/2)) - (2*C*(b*c - a*d)^2*Hypergeometric2F1[-5/2, -5
/2, -3/2, -((d*(a + b*Tan[e + f*x]))/(b*c - a*d))]*Sqrt[c + d*Tan[e + f*x]]
)/(5*b^3*f*(a + b*Tan[e + f*x])^(5/2)*Sqrt[(b*(c + d*Tan[e + f*x]))/(b*c -
a*d)]) + ((A - I*B - C)*(I*c + d)*((c + d*Tan[e + f*x])^(3/2)/((a - I*b)*(a
 + b*Tan[e + f*x])^(3/2)) + (3*(c - I*d)*((Sqrt[-c + I*d]*ArcTanh[(Sqrt[-c
+ I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[-a + I*b]*Sqrt[c + d*Tan[e + f*x]])]
)/(-a + I*b)^(3/2) + Sqrt[c + d*Tan[e + f*x]]/((a - I*b)*Sqrt[a + b*Tan[e +
 f*x]])))/(a - I*b)))/(3*(a - I*b)*f) - ((A + I*B - C)*(I*c - d)*((c + d*Ta
n[e + f*x])^(3/2)/((a + I*b)*(a + b*Tan[e + f*x])^(3/2)) - (3*(c + I*d)*((S
qrt[c + I*d]*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + I*b
]*Sqrt[c + d*Tan[e + f*x]])])/(a + I*b)^(3/2) - Sqrt[c + d*Tan[e + f*x]]/((
a + I*b)*Sqrt[a + b*Tan[e + f*x]])))/(a + I*b)))/(3*(a + I*b)*f)
```

Maple [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(fx+e))^{\frac{5}{2}}\left(A+B\tan(fx+e)+C\tan(fx+e)^{2}\right)}{(a+b\tan(fx+e))^{\frac{7}{2}}}dx$$

```
[In] int((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(7/2),x)
```

```
[Out] int((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(7/2),x)
```

Fricas [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{7/2}}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(7/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{7/2}}\,dx = \int \frac{(c+d\tan(e+fx))^{\frac{5}{2}}\left(A+B\tan(\right.}{(a+b\tan(e+}$$

```
[In] integrate((c+d*tan(f*x+e))**(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e))**(7/2),x)
```

```
[Out] Integral((c + d*tan(e + f*x))**(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)**
2)/(a + b*tan(e + f*x))**(7/2), x)
```

Maxima [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{7/2}}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(7/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

Giac [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{7/2}}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(7/2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{7/2}}\,dx = \text{Hanged}$$

```
[In] int(((c + d*tan(e + f*x))^(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x))^(7/2),x)
```

```
[Out] \text{Hanged}
```

3.146 $\int \frac{(c+d\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{9/2}}\,dx$

Optimal result 2017
Rubi [A] (verified) 2018
Mathematica [B] (warning: unable to verify) 2023
Maple [F(-1)] 2023
Fricas [F(-1)] 2024
Sympy [F(-1)] 2024
Maxima [F(-2)] 2024
Giac [F(-1)] 2025
Mupad [F(-1)] 2025

Optimal result

Integrand size = 49, antiderivative size = 946

$$\int \frac{(c+d\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{9/2}}\,dx =$$
$$-\frac{(iA+B-iC)(c-id)^{5/2}\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a-ib)^{9/2}f}$$
$$-\frac{(B-i(A-C))(c+id)^{5/2}\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a+ib)^{9/2}f}$$
$$-\frac{2(6a^5bBd^2+15a^6Cd^2+a^4b^2d(14Bc+8Ad+37Cd)+3a^2b^4(35Ac^2-35c^2C-70Bcd-39Ad^2+54Cd^2}{}$$
$$-\frac{2(6a^7bBd^3+15a^8Cd^3+2a^6b^2d^2(7Bc+4Ad+26Cd)-2ab^7(210Ac^3-210c^3C-525Bc^2d-406Acd^2+}{}$$
$$-\frac{2(2a^3bBd+5a^4Cd+b^4(7Bc+5Ad)+2ab^3(7Ac-7cC-6Bd)-a^2b^2(7Bc+9Ad-19Cd))\,(c+d\tan(}{35b^2(a^2+b^2)^2f(a+b\tan(e+fx))^{5/2}}$$
$$-\frac{2(Ab^2-a(bB-aC))(c+d\tan(e+fx))^{5/2}}{7b(a^2+b^2)f(a+b\tan(e+fx))^{7/2}}$$

```
[Out] -(I*A+B-I*C)*(c-I*d)^(5/2)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-
I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/(a-I*b)^(9/2)/f-(B-I*(A-C))*(c+I*d)^(5/2
)*arctanh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e
))^(1/2))/(a+I*b)^(9/2)/f-2/105*(6*a^7*b*B*d^3+15*a^8*C*d^3+2*a^6*b^2*d^2*(
4*A*d+7*B*c+26*C*d)-2*a*b^7*(210*A*c^3-406*A*c*d^2-525*B*c^2*d+88*B*d^3-210
*C*c^3+406*C*c*d^2)-a^4*b^4*(525*A*c^2*d-311*A*d^3+105*B*c^3-749*B*c*d^2-52
5*C*c^2*d+221*C*d^3)+2*a^2*b^6*(875*A*c^2*d-261*A*d^3+315*B*c^3-812*B*c*d^2
-875*C*c^2*d+291*C*d^3)+2*a^5*b^3*d*(56*c*(A-C)*d+B*(35*c^2-12*d^2))-b^8*(5
```

```
*d*(49*A*c^2-3*A*d^2-49*C*c^2)+7*B*(15*c^3-23*c*d^2))-2*a^3*b^5*(210*c^3*C+
700*B*c^2*d-798*C*c*d^2-317*B*d^3-42*A*(5*c^3-19*c*d^2)))*(c+d*tan(f*x+e))^
(1/2)/b^3/(a^2+b^2)^4/(-a*d+b*c)/f/(a+b*tan(f*x+e))^(1/2)-2/105*(6*a^5*b*B*
d^2+15*a^6*C*d^2+a^4*b^2*d*(8*A*d+14*B*c+37*C*d)+3*a^2*b^4*(35*A*c^2-39*A*d
^2-70*B*c*d-35*C*c^2+54*C*d^2)-a^3*b^3*(98*c*(A-C)*d+B*(35*c^2-75*d^2))+a*b
^5*(182*c*(A-C)*d+B*(105*c^2-71*d^2))+b^6*(7*c*(8*B*d+5*C*c)-5*A*(7*c^2-3*d
^2)))*(c+d*tan(f*x+e))^(1/2)/b^3/(a^2+b^2)^3/f/(a+b*tan(f*x+e))^(3/2)-2/35*
(2*a^3*b*B*d+5*a^4*C*d+b^4*(5*A*d+7*B*c)+2*a*b^3*(7*A*c-6*B*d-7*C*c)-a^2*b^
2*(9*A*d+7*B*c-19*C*d))*(c+d*tan(f*x+e))^(3/2)/b^2/(a^2+b^2)^2/f/(a+b*tan(f
*x+e))^(5/2)-2/7*(A*b^2-a*(B*b-C*a))*(c+d*tan(f*x+e))^(5/2)/b/(a^2+b^2)/f/(
a+b*tan(f*x+e))^(7/2)
```

Rubi [A] (verified)

Time = 7.16 (sec) , antiderivative size = 946, normalized size of antiderivative = 1.00, number of steps used = 11, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}} = 0.122$, Rules used = {3726, 3730, 3697, 3696, 95, 214}

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{9/2}}\,dx =$$

$$-\frac{(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)(c-id)^{5/2}}{(a-ib)^{9/2}f}$$

$$-\frac{2(Ab^2-a(bB-aC))(c+d\tan(e+fx))^{5/2}}{7b(a^2+b^2)f(a+b\tan(e+fx))^{7/2}}$$

$$-\frac{2(5Cda^4+2bBda^3-b^2(7Bc+9Ad-19Cd)a^2+2b^3(7Ac-7Cc-6Bd)a+b^4(7Bc+5Ad))(c+d\tan(e}{35b^2(a^2+b^2)^2f(a+b\tan(e+fx))^{5/2}}$$

$$-\frac{(B-i(A-C))(c+id)^{5/2}\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a+ib)^{9/2}f}$$

$$-\frac{2(15Cd^3a^8+6bBd^3a^7+2b^2d^2(7Bc+4Ad+26Cd)a^6+2b^3d(56c(A-C)d+B(35c^2-12d^2))a^5-b^4(105}{}$$

$$-\frac{2(15Cd^2a^6+6bBd^2a^5+b^2d(14Bc+8Ad+37Cd)a^4-b^3(98c(A-C)d+B(35c^2-75d^2))a^3+3b^4(35Ac^2}{}$$

1(

```
[In] Int[((c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a
 + b*Tan[e + f*x])^(9/2),x]

[Out] -(((I*A + B - I*C)*(c - I*d)^(5/2)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e
+ f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a - I*b)^(9/2)*f)) -
((B - I*(A - C))*(c + I*d)^(5/2)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a + I*b)^(9/2)*f) - (2*
(6*a^5*b*B*d^2 + 15*a^6*C*d^2 + a^4*b^2*d*(14*B*c + 8*A*d + 37*C*d) + 3*a^2
*b^4*(35*A*c^2 - 35*c^2*C - 70*B*c*d - 39*A*d^2 + 54*C*d^2) - a^3*b^3*(98*c
```

```
*(A - C)*d + B*(35*c^2 - 75*d^2)) + a*b^5*(182*c*(A - C)*d + B*(105*c^2 - 7
1*d^2)) + b^6*(7*c*(5*c*C + 8*B*d) - 5*A*(7*c^2 - 3*d^2)))*Sqrt[c + d*Tan[e
 + f*x]])/(105*b^3*(a^2 + b^2)^3*f*(a + b*Tan[e + f*x])^(3/2)) - (2*(6*a^7*
b*B*d^3 + 15*a^8*C*d^3 + 2*a^6*b^2*d^2*(7*B*c + 4*A*d + 26*C*d) - 2*a*b^7*(
210*A*c^3 - 210*c^3*C - 525*B*c^2*d - 406*A*c*d^2 + 406*c*C*d^2 + 88*B*d^3)
 - a^4*b^4*(105*B*c^3 + 525*A*c^2*d - 525*c^2*C*d - 749*B*c*d^2 - 311*A*d^3
 + 221*C*d^3) + 2*a^2*b^6*(315*B*c^3 + 875*A*c^2*d - 875*c^2*C*d - 812*B*c*
d^2 - 261*A*d^3 + 291*C*d^3) + 2*a^5*b^3*d*(56*c*(A - C)*d + B*(35*c^2 - 12
*d^2)) - b^8*(5*d*(49*A*c^2 - 49*c^2*C - 3*A*d^2) + 7*B*(15*c^3 - 23*c*d^2)
) - 2*a^3*b^5*(210*c^3*C + 700*B*c^2*d - 798*c*C*d^2 - 317*B*d^3 - 42*A*(5*
c^3 - 19*c*d^2)))*Sqrt[c + d*Tan[e + f*x]])/(105*b^3*(a^2 + b^2)^4*(b*c - a
*d)*f*Sqrt[a + b*Tan[e + f*x]]) - (2*(2*a^3*b*B*d + 5*a^4*C*d + b^4*(7*B*c
+ 5*A*d) + 2*a*b^3*(7*A*c - 7*c*C - 6*B*d) - a^2*b^2*(7*B*c + 9*A*d - 19*C*
d))*(c + d*Tan[e + f*x])^(3/2))/(35*b^2*(a^2 + b^2)^2*f*(a + b*Tan[e + f*x]
)^(5/2)) - (2*(A*b^2 - a*(b*B - a*C))*(c + d*Tan[e + f*x])^(5/2))/(7*b*(a^2
 + b^2)*f*(a + b*Tan[e + f*x])^(7/2))
```

Rule 95

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3696

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[A^2/f, Subst[Int[(a + b*x)^m*((c + d*x)^n/(A - B*x)), x], x, Tan[e + f*x
]], x] /; FreeQ[{a, b, c, d, e, f, A, B, m, n}, x] && NeQ[b*c - a*d, 0] &&
NeQ[a^2 + b^2, 0] && EqQ[A^2 + B^2, 0]
```

Rule 3697

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[(A + I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^n*(1 - I*Ta
n[e + f*x]), x], x] + Dist[(A - I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*T
an[e + f*x])^n*(1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A,
B, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[A^2 + B^2, 0]
```

Rule 3726

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*d^2 + c*(c*C - B*d))*(a + b*Tan[e +
 f*x])^m*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3730

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 - a*(b*B - a*C))*(a + b*Tan[e +
 f*x])^(m + 1)*((c + d*Tan[e + f*x])^(n + 1)/(f*(m + 1)*(b*c - a*d)*(a^2 +
b^2))), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rubi steps

$$\text{integral} = -\frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{5/2}}{7b\,(a^2+b^2)\,f(a+b\tan(e+fx))^{7/2}} + \frac{2\int \frac{(c+d\tan(e+fx))^{3/2}\left(\frac{1}{2}((bB-aC)(7bc-5ad)+Ab(7ac+5bd))-\frac{7}{2}b((A-C)(bc-ad)-B(ac+bd))\tan(e+fx)-\frac{1}{2}\left(2Ab^2-2abB-5a^2C-7b^2C\right)d\right.}{(a+b\tan(e+fx))^{7/2}}}{7b\,(a^2+b^2)}$$

$$= -\frac{2(2a^3bBd+5a^4Cd+b^4(7Bc+5Ad)+2ab^3(7Ac-7cC-6Bd)-a^2b^2(7Bc+9Ad-19Cd))\,(c}{35b^2\,(a^2+b^2)^2\,f(a+b\tan(e+fx))^{5/2}} - \frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{5/2}}{7b\,(a^2+b^2)\,f(a+b\tan(e+fx))^{7/2}} + \frac{4\int \sqrt{c+d\tan(e+fx)}\left(\frac{1}{4}\left(b(5ac+3bd)((bB-aC)(7bc-5ad)+Ab(7ac+5bd))-(5bc-3ad)\left(2a^2bBd+5a^3Cd+Ab^2(7bc-9ad)-7b^3(cC+\right.\right.\right.}{}$$

$$=$$

$$-\frac{2(6a^5bBd^2+15a^6Cd^2+a^4b^2d(14Bc+8Ad+37Cd)+3a^2b^4(35Ac^2-35c^2C-70Bcd-39Ad}{}$$

$$-\frac{2(2a^3bBd+5a^4Cd+b^4(7Bc+5Ad)+2ab^3(7Ac-7cC-6Bd)-a^2b^2(7Bc+9Ad-19Cd))(}{35b^2\left(a^2+b^2\right)^2f(a+b\tan(e+fx))^{5/2}}$$

$$-\frac{2(Ab^2-a(bB-aC))\left(c+d\tan(e+fx)\right)^{5/2}}{7b\left(a^2+b^2\right)f(a+b\tan(e+fx))^{7/2}}$$

$$+\frac{8\int\frac{\frac{1}{8}\left(6a^5bBd^3+15a^6Cd^3+a^4b^2d^2(14Bc+8Ad+37Cd)-ab^5\left(315Ac^3-315c^3C-735Bc^2d-497Acd^2+497cCd^2+71Bd^3\right)-a^3b^3\left(10\right.\right.}{}}{}$$

$$=$$

$$-\frac{2(6a^5bBd^2+15a^6Cd^2+a^4b^2d(14Bc+8Ad+37Cd)+3a^2b^4(35Ac^2-35c^2C-70Bcd-39Ad}{}$$

$$-\frac{2(6a^7bBd^3+15a^8Cd^3+2a^6b^2d^2(7Bc+4Ad+26Cd)-2ab^7(210Ac^3-210c^3C-525Bc^2d-4}{}$$

$$-\frac{2(2a^3bBd+5a^4Cd+b^4(7Bc+5Ad)+2ab^3(7Ac-7cC-6Bd)-a^2b^2(7Bc+9Ad-19Cd))(}{35b^2\left(a^2+b^2\right)^2f(a+b\tan(e+fx))^{5/2}}$$

$$-\frac{2(Ab^2-a(bB-aC))\left(c+d\tan(e+fx)\right)^{5/2}}{7b\left(a^2+b^2\right)f(a+b\tan(e+fx))^{7/2}}$$

$$-\frac{16\int\frac{\frac{105}{16}b^3(bc-ad)\left(6a^2b^2\left(Ac^3-c^3C-3Bc^2d-3Acd^2+3cCd^2+Bd^3\right)+a^4\left(c^3C+3Bc^2d-3cCd^2-Bd^3-A\left(c^3-3cd^2\right)\right)+b^4\left(c^3C+3B\right.\right.}{}}{}$$

$$=$$

$$-\frac{2(6a^5bBd^2+15a^6Cd^2+a^4b^2d(14Bc+8Ad+37Cd)+3a^2b^4(35Ac^2-35c^2C-70Bcd-39Ad}{}$$

$$-\frac{2(6a^7bBd^3+15a^8Cd^3+2a^6b^2d^2(7Bc+4Ad+26Cd)-2ab^7(210Ac^3-210c^3C-525Bc^2d-4}{}$$

$$-\frac{2(2a^3bBd+5a^4Cd+b^4(7Bc+5Ad)+2ab^3(7Ac-7cC-6Bd)-a^2b^2(7Bc+9Ad-19Cd))(}{35b^2\left(a^2+b^2\right)^2f(a+b\tan(e+fx))^{5/2}}$$

$$-\frac{2(Ab^2-a(bB-aC))\left(c+d\tan(e+fx)\right)^{5/2}}{7b\left(a^2+b^2\right)f(a+b\tan(e+fx))^{7/2}}$$

$$+\frac{((A-iB-C)(c-id)^3)\int\frac{1+i\tan(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx}{2(a-ib)^4}$$

$$+\frac{((A+iB-C)(c+id)^3)\int\frac{1-i\tan(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx}{2(a+ib)^4}$$

$$=$$
$$-\frac{2(6a^5bBd^2+15a^6Cd^2+a^4b^2d(14Bc+8Ad+37Cd)+3a^2b^4(35Ac^2-35c^2C-70Bcd-39Ad^2}{}$$
$$-\frac{2(6a^7bBd^3+15a^8Cd^3+2a^6b^2d^2(7Bc+4Ad+26Cd)-2ab^7(210Ac^3-210c^3C-525Bc^2d-40}{}$$
$$-\frac{2(2a^3bBd+5a^4Cd+b^4(7Bc+5Ad)+2ab^3(7Ac-7cC-6Bd)-a^2b^2(7Bc+9Ad-19Cd))\,(c}{35b^2\left(a^2+b^2\right)^2f(a+b\tan(e+fx))^{5/2}}$$
$$-\frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{5/2}}{7b\left(a^2+b^2\right)f(a+b\tan(e+fx))^{7/2}}$$
$$+\frac{((A-iB-C)(c-id)^3)\operatorname{Subst}\left(\int\frac{1}{(1-ix)\sqrt{a+bx}\sqrt{c+dx}}\,dx,x,\tan(e+fx)\right)}{2(a-ib)^4f}$$
$$+\frac{((A+iB-C)(c+id)^3)\operatorname{Subst}\left(\int\frac{1}{(1+ix)\sqrt{a+bx}\sqrt{c+dx}}\,dx,x,\tan(e+fx)\right)}{2(a+ib)^4f}$$
$$=$$
$$-\frac{2(6a^5bBd^2+15a^6Cd^2+a^4b^2d(14Bc+8Ad+37Cd)+3a^2b^4(35Ac^2-35c^2C-70Bcd-39Ad^2}{}$$
$$-\frac{2(6a^7bBd^3+15a^8Cd^3+2a^6b^2d^2(7Bc+4Ad+26Cd)-2ab^7(210Ac^3-210c^3C-525Bc^2d-40}{}$$
$$-\frac{2(2a^3bBd+5a^4Cd+b^4(7Bc+5Ad)+2ab^3(7Ac-7cC-6Bd)-a^2b^2(7Bc+9Ad-19Cd))\,(c}{35b^2\left(a^2+b^2\right)^2f(a+b\tan(e+fx))^{5/2}}$$
$$-\frac{2(Ab^2-a(bB-aC))\,(c+d\tan(e+fx))^{5/2}}{7b\left(a^2+b^2\right)f(a+b\tan(e+fx))^{7/2}}$$
$$+\frac{((A-iB-C)(c-id)^3)\operatorname{Subst}\left(\int\frac{1}{ia+b-(ic+d)x^2}\,dx,x,\frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{(a-ib)^4f}$$
$$+\frac{((A+iB-C)(c+id)^3)\operatorname{Subst}\left(\int\frac{1}{-ia+b-(-ic+d)x^2}\,dx,x,\frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{(a+ib)^4f}$$

$$= -\frac{(iA+B-iC)(c-id)^{5/2}\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a-ib)^{9/2}f} - \frac{(B-i(A-C))(c+id)^{5/2}\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a+ib)^{9/2}f}$$

$$- \frac{2(6a^5bBd^2+15a^6Cd^2+a^4b^2d(14Bc+8Ad+37Cd)+3a^2b^4(35Ac^2-35c^2C-70Bcd-39Ad}{}$$

$$- \frac{2(6a^7bBd^3+15a^8Cd^3+2a^6b^2d^2(7Bc+4Ad+26Cd)-2ab^7(210Ac^3-210c^3C-525Bc^2d-}{}$$

$$- \frac{2(2a^3bBd+5a^4Cd+b^4(7Bc+5Ad)+2ab^3(7Ac-7cC-6Bd)-a^2b^2(7Bc+9Ad-19Cd))(}{35b^2(a^2+b^2)^2 f(a+b\tan(e+fx))^{5/2}}$$

$$- \frac{2(Ab^2-a(bB-aC))(c+d\tan(e+fx))^{5/2}}{7b(a^2+b^2)f(a+b\tan(e+fx))^{7/2}}$$

Mathematica [B] (warning: unable to verify)

Both result and optimal contain complex but leaf count is larger than twice the leaf count of optimal. 10121 vs. 2(946) = 1892.

Time = 56.13 (sec) , antiderivative size = 10121, normalized size of antiderivative = 10.70

$$\int \frac{(c+d\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(a+b\tan(e+fx))^{9/2}}\,dx = \text{Result too large to show}$$

```
[In] Integrate[((c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/(a + b*Tan[e + f*x])^(9/2),x]
```

```
[Out] Result too large to show
```

Maple [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(fx+e))^{\frac{5}{2}}(A+B\tan(fx+e)+C\tan(fx+e)^2)}{(a+b\tan(fx+e))^{\frac{9}{2}}}dx$$

```
[In] int((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(9/2),x)
```

```
[Out] int((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(9/2),x)
```

Fricas [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{9/2}}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(9/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{9/2}}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))**(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e))**(9/2),x)
```

```
[Out] Timed out
```

Maxima [F(-2)]

Exception generated.

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{9/2}}\,dx = \text{Exception raised: ValueError}$$

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(9/2),x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(((2*b*d+2*a*c)^2>0)', see `assume?`
 for mo
```

Giac [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{9/2}}\,dx = \text{Timed out}$$

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(9/2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{9/2}}\,dx = \text{Hanged}$$

```
[In] int(((c + d*tan(e + f*x))^(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x))^(9/2),x)
```

```
[Out] \text{Hanged}
```

3.147 $\int \frac{(a+b\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{c+d\tan(e+fx)}}\,dx$

Optimal result 2026
Rubi [A] (verified) 2027
Mathematica [A] (verified) 2032
Maple [F(-1)] 2033
Fricas [F(-1)] 2033
Sympy [F] 2033
Maxima [F(-1)] 2034
Giac [F(-1)] 2034
Mupad [F(-1)] 2034

Optimal result

Integrand size = 49, antiderivative size = 505

$$\int \frac{(a+b\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx =$$

$$-\frac{(a-ib)^{5/2}(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{c-id}f}$$

$$-\frac{(a+ib)^{5/2}(B-i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{c+id}f}$$

$$+\frac{(5a^3Cd^3-15a^2bd^2(cC-2Bd)+5ab^2d(3c^2C-4Bcd+8(A-C)d^2)-b^3(5c^3C-6Bc^2d+8c(A-C)d^2+}{8\sqrt{b}d^{7/2}f}$$

$$+\frac{(8b(Ab+aB-bC)d^2+(bc-ad)(5bcC-6bBd-5aCd))\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{8d^3f}$$

$$-\frac{(5bcC-6bBd-5aCd)(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}{12d^2f}$$

$$+\frac{C(a+b\tan(e+fx))^{5/2}\sqrt{c+d\tan(e+fx)}}{3df}$$

```
[Out] 1/8*(5*a^3*C*d^3-15*a^2*b*d^2*(-2*B*d+C*c)+5*a*b^2*d*(3*c^2*C-4*B*c*d+8*(A-
C)*d^2)-b^3*(5*c^3*C-6*B*c^2*d+8*c*(A-C)*d^2+16*B*d^3))*arctanh(d^(1/2)*(a+
b*tan(f*x+e))^(1/2)/b^(1/2)/(c+d*tan(f*x+e))^(1/2))/d^(7/2)/f/b^(1/2)-(a-I*
b)^(5/2)*(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(
1/2)/(c+d*tan(f*x+e))^(1/2))/f/(c-I*d)^(1/2)-(a+I*b)^(5/2)*(B-I*(A-C))*arct
anh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/
2))/f/(c+I*d)^(1/2)+1/8*(8*b*(A*b+B*a-C*b)*d^2+(-a*d+b*c)*(-6*B*b*d-5*C*a*d
+5*C*b*c))*(a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)/d^3/f-1/12*(-6*B*b
```

```
*d-5*C*a*d+5*C*b*c)*(c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^(3/2)/d^2/f+1/3
*C*(c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^(5/2)/d/f
```

Rubi [A] (verified)

Time = 6.88 (sec) , antiderivative size = 505, normalized size of antiderivative = 1.00, number of steps used = 15, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.163, Rules used = {3728, 3736, 6857, 65, 223, 212, 95, 214}

$$\int \frac{(a+b\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{c+d\tan(e+fx)}}\,dx = \frac{(5a^3Cd^3-15a^2bd^2(cC-2Bd)+5ab$$

$$-\frac{(a-ib)^{5/2}(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f\sqrt{c-id}}$$
$$-\frac{(a+ib)^{5/2}(B-i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f\sqrt{c+id}}$$
$$+\frac{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}(8bd^2(aB+Ab-bC)+(bc-ad)(-5aCd-6bBd+5bcC))}{8d^3f}$$
$$-\frac{(-5aCd-6bBd+5bcC)(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}{12d^2f}$$
$$+\frac{C(a+b\tan(e+fx))^{5/2}\sqrt{c+d\tan(e+fx)}}{3df}$$

```
[In] Int[((a + b*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/Sq
rt[c + d*Tan[e + f*x]],x]
```

```
[Out] -(((a - I*b)^(5/2)*(I*A + B - I*C)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e
+ f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[c - I*d]*f)) - ((
a + I*b)^(5/2)*(B - I*(A - C))*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*
x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[c + I*d]*f) + ((5*a^3
*C*d^3 - 15*a^2*b*d^2*(c*C - 2*B*d) + 5*a*b^2*d*(3*c^2*C - 4*B*c*d + 8*(A -
 C)*d^2) - b^3*(5*c^3*C - 6*B*c^2*d + 8*c*(A - C)*d^2 + 16*B*d^3))*ArcTanh[
(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])])/(8*
Sqrt[b]*d^(7/2)*f) + ((8*b*(A*b + a*B - b*C)*d^2 + (b*c - a*d)*(5*b*c*C - 6
*b*B*d - 5*a*C*d))*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(8*d^
3*f) - ((5*b*c*C - 6*b*B*d - 5*a*C*d)*(a + b*Tan[e + f*x])^(3/2)*Sqrt[c + d
*Tan[e + f*x]])/(12*d^2*f) + (C*(a + b*Tan[e + f*x])^(5/2)*Sqrt[c + d*Tan[e
 + f*x]])/(3*d*f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
```

```
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 95

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 223

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3736

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[(a + b*ff*x)^m*(c + d*ff*x)^n*((A + B*ff*x + C*ff^2*x^2
```

```
)/(1 + ff^2*x^2)), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6857

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

integral

$$= \frac{C(a+b\tan(e+fx))^{5/2}\sqrt{c+d\tan(e+fx)}}{3df} + \frac{\int \frac{(a+b\tan(e+fx))^{3/2}\left(\frac{1}{2}(-5bcC+a(6A-C)d)+3(Ab+aB-bC)d\tan(e+fx)-\frac{1}{2}(5bcC-6bBd-5aCd)\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx}{3d}$$

$$= -\frac{(5bcC-6bBd-5aCd)(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}{12d^2f} + \frac{C(a+b\tan(e+fx))^{5/2}\sqrt{c+d\tan(e+fx)}}{3df} + \frac{\int \frac{\sqrt{a+b\tan(e+fx)}\left(\frac{1}{4}(-4ad(5bcC-a(6A-C)d)+(3bc+ad)(5bcC-6bBd-5aCd))+6\left(a^2B-b^2B+2ab(A-C)\right)d^2\tan(e+fx)+\frac{3}{4}(8b(Ab+aB-\right.}{\sqrt{c+d\tan(e+fx)}}}{6d^2}$$

$$= \frac{\left(8b(Ab+aB-bC)d^2+(bc-ad)(5bcC-6bBd-5aCd)\right)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{8d^3f} - \frac{(5bcC-6bBd-5aCd)(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}{12d^2f} + \frac{C(a+b\tan(e+fx))^{5/2}\sqrt{c+d\tan(e+fx)}}{3df} + \frac{\int \frac{\frac{3}{8}\left(a^3(16A-11C)d^3-3a^2bd^2(5cC+6Bd)+ab^2d\left(15c^2C-20Bcd-8(A-C)d^2\right)-b^3c\left(5c^2C-6Bcd+8(A-C)d^2\right)\right)+6\left(a^3B-3ab^2B+3a^2b(A-C\right.}{\sqrt{a+b\tan(e+fx)}}}{6}$$

$$= \frac{\left(8b(Ab+aB-bC)d^2+(bc-ad)(5bcC-6bBd-5aCd)\right)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{8d^3f} - \frac{(5bcC-6bBd-5aCd)(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}{12d^2f} + \frac{C(a+b\tan(e+fx))^{5/2}\sqrt{c+d\tan(e+fx)}}{3df} + \frac{\text{Subst}\left(\int \frac{\frac{3}{8}\left(a^3(16A-11C)d^3-3a^2bd^2(5cC+6Bd)+ab^2d\left(15c^2C-20Bcd-8(A-C)d^2\right)-b^3c\left(5c^2C-6Bcd+8(A-C)d^2\right)\right)+6\left(a^3B-3ab^2B+3\right.}{\sqrt{a+bx}\sqrt{c-}}\right.}{}$$

$$= \frac{(8b(Ab+aB-bC)d^2+(bc-ad)(5bcC-6bBd-5aCd))\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{8d^3f} - \frac{(5bcC-6bBd-5aCd)(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}{12d^2f} + \frac{C(a+b\tan(e+fx))^{5/2}\sqrt{c+d\tan(e+fx)}}{3df} + \frac{\text{Subst}\left(\int\left(\frac{3(5a^3Cd^3-15a^2bd^2(cC-2Bd)+5ab^2d(3c^2C-4Bcd+8(A-C)d^2)-b^3(5c^3C-6Bc^2d+8c(A-C)d^2+16Bd^3))}{8\sqrt{a+bx}\sqrt{c+dx}} + \frac{6(-((3a^2bB-b}{}\right.\right.}{6d^3f}$$

$$= \frac{(8b(Ab+aB-bC)d^2+(bc-ad)(5bcC-6bBd-5aCd))\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{8d^3f} - \frac{(5bcC-6bBd-5aCd)(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}{12d^2f} + \frac{C(a+b\tan(e+fx))^{5/2}\sqrt{c+d\tan(e+fx)}}{3df} + \frac{\text{Subst}\left(\int\frac{-((3a^2bB-b^3B-a^3(A-C)+3ab^2(A-C))d^3)+(a^3B-3ab^2B+3a^2b(A-C)-b^3(A-C))d^3x}{\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}dx,x,\tan(e+fx)\right)}{d^3f} + \frac{(5a^3Cd^3-15a^2bd^2(cC-2Bd)+5ab^2d(3c^2C-4Bcd+8(A-C)d^2)-b^3(5c^3C-6Bc^2d+8c(A-C)d}{16d^3f}$$

$$= \frac{(8b(Ab+aB-bC)d^2+(bc-ad)(5bcC-6bBd-5aCd))\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{8d^3f} - \frac{(5bcC-6bBd-5aCd)(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}{12d^2f} + \frac{C(a+b\tan(e+fx))^{5/2}\sqrt{c+d\tan(e+fx)}}{3df} + \frac{\text{Subst}\left(\int\left(\frac{-i(3a^2bB-b^3B-a^3(A-C)+3ab^2(A-C))d^3-(a^3B-3ab^2B+3a^2b(A-C)-b^3(A-C))d^3}{2(i-x)\sqrt{a+bx}\sqrt{c+dx}} + \frac{-i(3a^2bB-b^3B-a^3(A-C)+3ab^2(}{2(}\right.\right.}{d^3f} + \frac{(5a^3Cd^3-15a^2bd^2(cC-2Bd)+5ab^2d(3c^2C-4Bcd+8(A-C)d^2)-b^3(5c^3C-6Bc^2d+8c(A-C)d}{8bd^3f}$$

$$= \frac{(8b(Ab+aB-bC)d^2+(bc-ad)(5bcC-6bBd-5aCd))\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{8d^3f}$$
$$-\frac{(5bcC-6bBd-5aCd)(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}{12d^2f}$$
$$+\frac{C(a+b\tan(e+fx))^{5/2}\sqrt{c+d\tan(e+fx)}}{3df}$$
$$+\frac{((a-ib)^3(iA+B-iC))\operatorname{Subst}\left(\int\frac{1}{(i+x)\sqrt{a+bx}\sqrt{c+dx}}\,dx,x,\tan(e+fx)\right)}{2f}$$
$$-\frac{(i(3a^2bB-b^3B-a^3(A-C)+3ab^2(A-C))\,d^3+(a^3B-3ab^2B+3a^2b(A-C)-b^3(A-C))\,d^3)\operatorname{Sub}}{2d^3f}$$
$$+\frac{(5a^3Cd^3-15a^2bd^2(cC-2Bd)+5ab^2d(3c^2C-4Bcd+8(A-C)d^2)-b^3(5c^3C-6Bc^2d+8c(A-C)}{8bd^3f}$$

$$= \frac{(5a^3Cd^3-15a^2bd^2(cC-2Bd)+5ab^2d(3c^2C-4Bcd+8(A-C)d^2)-b^3(5c^3C-6Bc^2d+8c(A-C)d^2}{8\sqrt{b}d^{7/2}f}$$
$$+\frac{(8b(Ab+aB-bC)d^2+(bc-ad)(5bcC-6bBd-5aCd))\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{8d^3f}$$
$$-\frac{(5bcC-6bBd-5aCd)(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}{12d^2f}$$
$$+\frac{C(a+b\tan(e+fx))^{5/2}\sqrt{c+d\tan(e+fx)}}{3df}$$
$$+\frac{((a-ib)^3(iA+B-iC))\operatorname{Subst}\left(\int\frac{1}{-a+ib-(-c+id)x^2}\,dx,x,\frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{f}$$
$$-\frac{(i(3a^2bB-b^3B-a^3(A-C)+3ab^2(A-C))\,d^3+(a^3B-3ab^2B+3a^2b(A-C)-b^3(A-C))\,d^3)\operatorname{Sub}}{d^3f}$$

$$= -\frac{(a-ib)^{5/2}(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{c-id}f}$$
$$+\frac{(a+ib)^{5/2}(iA-B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{c+id}f}$$
$$+\frac{(5a^3Cd^3-15a^2bd^2(cC-2Bd)+5ab^2d(3c^2C-4Bcd+8(A-C)d^2)-b^3(5c^3C-6Bc^2d+8c(A-C)d}{8\sqrt{b}d^{7/2}f}$$
$$+\frac{(8b(Ab+aB-bC)d^2+(bc-ad)(5bcC-6bBd-5aCd))\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{8d^3f}$$
$$-\frac{(5bcC-6bBd-5aCd)(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}{12d^2f}$$
$$+\frac{C(a+b\tan(e+fx))^{5/2}\sqrt{c+d\tan(e+fx)}}{3df}$$

Mathematica [A] (verified)

Time = 8.84 (sec) , antiderivative size = 785, normalized size of antiderivative = 1.55

$$\int\frac{(a+b\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx=\frac{C(a+b\tan(e+fx))^{5/2}\sqrt{c+d\tan(e+fx)}}{3df}$$

$$+\frac{\frac{(-5bcC+6bBd+5aCd)(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}{4df}+\frac{3\left(8b(Ab+aB-bC)d^2+(bc-ad)(5bcC-6bBd-5aCd)\right)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{4df}+\cdots}{\cdots}$$

```
[In] Integrate[((a + b*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/Sqrt[c + d*Tan[e + f*x]],x]
```

```
[Out] (C*(a + b*Tan[e + f*x])^(5/2)*Sqrt[c + d*Tan[e + f*x]])/(3*d*f) + (((-5*b*c
*C + 6*b*B*d + 5*a*C*d)*(a + b*Tan[e + f*x])^(3/2)*Sqrt[c + d*Tan[e + f*x]]
)/(4*d*f) + ((3*(8*b*(A*b + a*B - b*C)*d^2 + (b*c - a*d)*(5*b*c*C - 6*b*B*d
 - 5*a*C*d))*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(4*d*f) + (
(-6*(Sqrt[-b^2]*(3*a^2*b*B - b^3*B - a^3*(A - C) + 3*a*b^2*(A - C)) - b*(a^
3*B - 3*a*b^2*B + 3*a^2*b*(A - C) - b^3*(A - C)))*d^3*ArcTanh[(Sqrt[-c + (S
qrt[-b^2]*d)/b]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[c + d
*Tan[e + f*x]])]])/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[-c + (Sqrt[-b^2]*d)/b]) - (6*
(Sqrt[-b^2]*(3*a^2*b*B - b^3*B - a^3*(A - C) + 3*a*b^2*(A - C)) + b*(a^3*B
- 3*a*b^2*B + 3*a^2*b*(A - C) - b^3*(A - C)))*d^3*ArcTanh[(Sqrt[c + (Sqrt[-
b^2]*d)/b]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + Sqrt[-b^2]]*Sqrt[c + d*Tan[e
 + f*x]])]])/(Sqrt[a + Sqrt[-b^2]]*Sqrt[c + (Sqrt[-b^2]*d)/b]) + (3*Sqrt[b]*
Sqrt[c - (a*d)/b]*(5*a^3*C*d^3 - 15*a^2*b*d^2*(c*C - 2*B*d) + 5*a*b^2*d*(3*
```

```
c^2*C - 4*B*c*d + 8*(A - C)*d^2) - b^3*(5*c^3*C - 6*B*c^2*d + 8*c*(A - C)*d
^2 + 16*B*d^3))*ArcSinh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c
- (a*d)/b])]*Sqrt[(b*c + b*d*Tan[e + f*x])/(b*c - a*d)])/(4*Sqrt[d]*Sqrt[c
+ d*Tan[e + f*x]]))/(b*d*f))/(2*d))/(3*d)
```

Maple [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(fx+e))^{\frac{5}{2}}\left(A+B\tan(fx+e)+C\tan(fx+e)^2\right)}{\sqrt{c+d\tan(fx+e)}}dx$$

```
[In] int((a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(1/2),x)
```

```
[Out] int((a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(1/2),x)
```

Fricas [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(1/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int \frac{(a+b\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx = \int \frac{(a+b\tan(e+fx))^{\frac{5}{2}}\left(A+B\tan(\right.}{\sqrt{c+d\tan(e}}$$

```
[In] integrate((a+b*tan(f*x+e))**(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan
(f*x+e))**(1/2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)**
2)/sqrt(c + d*tan(e + f*x)), x)
```

Maxima [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(1/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

Giac [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(1/2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx = \int \frac{(a+b\tan(e+f\,x))^{5/2}\,(C\tan(e+f}{\sqrt{c+d\tan(e}$$

```
[In] int(((a + b*tan(e + f*x))^(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c
 + d*tan(e + f*x))^(1/2),x)
```

```
[Out] int(((a + b*tan(e + f*x))^(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c
 + d*tan(e + f*x))^(1/2), x)
```

3.148 $\int \frac{(a+b\tan(e+fx))^{3/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{c+d\tan(e+fx)}}\,dx$

Optimal result . . . 2035
Rubi [A] (verified) . . . 2036
Mathematica [A] (verified) . . . 2040
Maple [F(-1)] . . . 2041
Fricas [B] (verification not implemented) . . . 2041
Sympy [F] . . . 2041
Maxima [F] . . . 2042
Giac [F(-1)] . . . 2042
Mupad [F(-1)] . . . 2042

Optimal result

Integrand size = 49, antiderivative size = 383

$$\begin{aligned}
&\int \frac{(a+b\tan(e+fx))^{3/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{c+d\tan(e+fx)}}\,dx = \\
&-\frac{(a-ib)^{3/2}(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{c-id}f} \\
&+\frac{(a+ib)^{3/2}(iA-B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{c+id}f} \\
&+\frac{(3a^2Cd^2-6abd(cC-2Bd)+b^2(3c^2C-4Bcd+8(A-C)d^2))\operatorname{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{4\sqrt{b}d^{5/2}f} \\
&-\frac{(3bcC-4bBd-3aCd)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{4d^2f} \\
&+\frac{C(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}{2df}
\end{aligned}$$

```
[Out] 1/4*(3*a^2*C*d^2-6*a*b*d*(-2*B*d+C*c)+b^2*(3*c^2*C-4*B*c*d+8*(A-C)*d^2))*ar
ctanh(d^(1/2)*(a+b*tan(f*x+e))^(1/2)/b^(1/2)/(c+d*tan(f*x+e))^(1/2))/d^(5/2
)/f/b^(1/2)-(a-I*b)^(3/2)*(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e)
)^(1/2)/(a-I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/f/(c-I*d)^(1/2)+(a+I*b)^(3/2)
*(I*A-B-I*C)*arctanh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+
d*tan(f*x+e))^(1/2))/f/(c+I*d)^(1/2)-1/4*(-4*B*b*d-3*C*a*d+3*C*b*c)*(a+b*ta
n(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)/d^2/f+1/2*C*(c+d*tan(f*x+e))^(1/2)*(
a+b*tan(f*x+e))^(3/2)/d/f
```

Rubi [A] (verified)

Time = 4.83 (sec) , antiderivative size = 383, normalized size of antiderivative = 1.00, number of steps used = 14, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.163, Rules used = {3728, 3736, 6857, 65, 223, 212, 95, 214}

$$\int \frac{(a+b\tan(e+fx))^{3/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{c+d\tan(e+fx)}}\,dx = \frac{(3a^2Cd^2-6abd(cC-2Bd)+b^2(8d^2(\ldots}{\ldots}$$
$$-\frac{(a-ib)^{3/2}(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f\sqrt{c-id}}$$
$$+\frac{(a+ib)^{3/2}(iA-B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f\sqrt{c+id}}$$
$$-\frac{(-3aCd-4bBd+3bcC)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{4d^2f}$$
$$+\frac{C(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}{2df}$$

```
[In] Int[((a + b*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/Sq
rt[c + d*Tan[e + f*x]],x]
```

```
[Out] -(((a - I*b)^(3/2)*(I*A + B - I*C)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e
+ f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[c - I*d]*f)) + ((
a + I*b)^(3/2)*(I*A - B - I*C)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*
x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[c + I*d]*f) + ((3*a^2
*C*d^2 - 6*a*b*d*(c*C - 2*B*d) + b^2*(3*c^2*C - 4*B*c*d + 8*(A - C)*d^2))*A
rcTanh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]]
)])/(4*Sqrt[b]*d^(5/2)*f) - ((3*b*c*C - 4*b*B*d - 3*a*C*d)*Sqrt[a + b*Tan[e
 + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(4*d^2*f) + (C*(a + b*Tan[e + f*x])^(3/2
)*Sqrt[c + d*Tan[e + f*x]])/(2*d*f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 95

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
```

```
&& LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 223

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3736

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[(a + b*ff*x)^m*(c + d*ff*x)^n*((A + B*ff*x + C*ff^2*x^2
)/(1 + ff^2*x^2)), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6857

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

integral

$$= \frac{C(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}{2df} + \frac{\int \frac{\sqrt{a+b\tan(e+fx)}\left(\frac{1}{2}(-3bcC+a(4A-C)d)+2(Ab+aB-bC)d\tan(e+fx)-\frac{1}{2}(3bcC-4bBd-3aCd)\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2d}$$

$$= -\frac{(3bcC-4bBd-3aCd)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{4d^2f} + \frac{C(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}{2df} + \frac{\int \frac{\frac{1}{4}\left(a^2(8A-5C)d^2+b^2c(3cC-4Bd)-2abd(3cC+2Bd)\right)+2\left(a^2B-b^2B+2ab(A-C)\right)d^2\tan(e+fx)+\frac{1}{4}\left(8b(Ab+aB-bC)d^2+(bc-ad)(3bcC-4bB\right.}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}}{2d^2}$$

$$= -\frac{(3bcC-4bBd-3aCd)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{4d^2f} + \frac{C(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}{2df} + \frac{\text{Subst}\left(\int \frac{\frac{1}{4}\left(a^2(8A-5C)d^2+b^2c(3cC-4Bd)-2abd(3cC+2Bd)\right)+2\left(a^2B-b^2B+2ab(A-C)\right)d^2x+\frac{1}{4}\left(8b(Ab+aB-bC)d^2+(bc-ad)(3bcC-4bB\right.}{\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}\right.}{2d^2f}$$

$$= -\frac{(3bcC-4bBd-3aCd)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{4d^2f} + \frac{C(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}{2df} + \frac{\text{Subst}\left(\int \left(\frac{3a^2Cd^2-6abd(cC-2Bd)+b^2(3c^2C-4Bcd+8(A-C)d^2)}{4\sqrt{a+bx}\sqrt{c+dx}} + \frac{2\left(-\left((2abB-a^2(A-C)+b^2(A-C))d^2\right)+(a^2B-b^2B+2ab(A-C))d^2\right.}{\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}\right.\right.}{2d^2f}$$

$$= -\frac{(3bcC-4bBd-3aCd)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{4d^2f} + \frac{C(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}{2df} + \frac{\text{Subst}\left(\int \frac{-\left((2abB-a^2(A-C)+b^2(A-C))d^2\right)+(a^2B-b^2B+2ab(A-C))d^2x}{\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}\,dx,x,\tan(e+fx)\right)}{d^2f} + \frac{\left(3a^2Cd^2-6abd(cC-2Bd)+b^2(3c^2C-4Bcd+8(A-C)d^2)\right)\text{Subst}\left(\int \frac{1}{\sqrt{a+bx}\sqrt{c+dx}}\,dx,x,\tan(e+fx)\right)}{8d^2f}$$

$$= -\frac{(3bcC - 4bBd - 3aCd)\sqrt{a + b\tan(e + fx)}\sqrt{c + d\tan(e + fx)}}{4d^2 f}$$
$$+ \frac{C(a + b\tan(e + fx))^{3/2}\sqrt{c + d\tan(e + fx)}}{2df}$$
$$+ \frac{\text{Subst}\left(\int \left(\frac{-((a^2B - b^2B + 2ab(A - C))d^2) - i(2abB - a^2(A - C) + b^2(A - C))d^2}{2(i - x)\sqrt{a + bx}\sqrt{c + dx}} + \frac{(a^2B - b^2B + 2ab(A - C))d^2 - i(2abB - a^2(A - C) + b^2(A -}{2(i + x)\sqrt{a + bx}\sqrt{c + dx}}\right.\right.}{d^2 f}$$
$$+ \frac{(3a^2Cd^2 - 6abd(cC - 2Bd) + b^2(3c^2C - 4Bcd + 8(A - C)d^2))\,\text{Subst}\left(\int \frac{1}{\sqrt{c - \frac{ad}{b} + \frac{dx^2}{b}}}\,dx, x, \sqrt{a + b\tan(\right.}{4bd^2 f}$$
$$= -\frac{(3bcC - 4bBd - 3aCd)\sqrt{a + b\tan(e + fx)}\sqrt{c + d\tan(e + fx)}}{4d^2 f}$$
$$+ \frac{C(a + b\tan(e + fx))^{3/2}\sqrt{c + d\tan(e + fx)}}{2df}$$
$$+ \frac{((a - ib)^2(iA + B - iC))\,\text{Subst}\left(\int \frac{1}{(i + x)\sqrt{a + bx}\sqrt{c + dx}}\,dx, x, \tan(e + fx)\right)}{2f}$$
$$- \frac{((a^2B - b^2B + 2ab(A - C))\,d^2 + i(2abB - a^2(A - C) + b^2(A - C))\,d^2)\,\text{Subst}\left(\int \frac{1}{(i - x)\sqrt{a + bx}\sqrt{c + dx}}\,dx, x\right.}{2d^2 f}$$
$$+ \frac{(3a^2Cd^2 - 6abd(cC - 2Bd) + b^2(3c^2C - 4Bcd + 8(A - C)d^2))\,\text{Subst}\left(\int \frac{1}{1 - \frac{dx^2}{b}}\,dx, x, \frac{\sqrt{a + b\tan(e + fx)}}{\sqrt{c + d\tan(e + fx)}}\right)}{4bd^2 f}$$
$$= \frac{(3a^2Cd^2 - 6abd(cC - 2Bd) + b^2(3c^2C - 4Bcd + 8(A - C)d^2))\,\text{arctanh}\left(\frac{\sqrt{d}\sqrt{a + b\tan(e + fx)}}{\sqrt{b}\sqrt{c + d\tan(e + fx)}}\right)}{4\sqrt{b}d^{5/2} f}$$
$$- \frac{(3bcC - 4bBd - 3aCd)\sqrt{a + b\tan(e + fx)}\sqrt{c + d\tan(e + fx)}}{4d^2 f}$$
$$+ \frac{C(a + b\tan(e + fx))^{3/2}\sqrt{c + d\tan(e + fx)}}{2df}$$
$$+ \frac{((a - ib)^2(iA + B - iC))\,\text{Subst}\left(\int \frac{1}{-a + ib - (-c + id)x^2}\,dx, x, \frac{\sqrt{a + b\tan(e + fx)}}{\sqrt{c + d\tan(e + fx)}}\right)}{f}$$
$$- \frac{((a^2B - b^2B + 2ab(A - C))\,d^2 + i(2abB - a^2(A - C) + b^2(A - C))\,d^2)\,\text{Subst}\left(\int \frac{1}{a + ib - (c + id)x^2}\,dx, x, \frac{\sqrt{c}}{\sqrt{c}}\right.}{d^2 f}$$

$$= -\frac{(a-ib)^{3/2}(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{c-id}f}$$
$$+\frac{(a+ib)^{3/2}(iA-B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{c+id}f}$$
$$+\frac{(3a^2Cd^2-6abd(cC-2Bd)+b^2(3c^2C-4Bcd+8(A-C)d^2))\operatorname{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{4\sqrt{b}d^{5/2}f}$$
$$-\frac{(3bcC-4bBd-3aCd)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{4d^2f}$$
$$+\frac{C(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}{2df}$$

Mathematica [A] (verified)

Time = 7.76 (sec) , antiderivative size = 607, normalized size of antiderivative = 1.58

$$\int \frac{(a+b\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx = \frac{C(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e-}{2df}$$
$$+\frac{\frac{(-3bcC+4bBd+3aCd)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{2df}+\frac{2\left(b\left(a^2B-b^2B+2ab(A-C)\right)-\sqrt{-b^2}\left(2abB-a^2(A-C)+b^2(A-C)\right)\right)d^2\operatorname{arctanh}\left(\frac{\sqrt{-c+\sqrt{}}}{\sqrt{-a+\sqrt{}}}\right.}{\sqrt{-a+\sqrt{-b^2}}\sqrt{-c+\frac{\sqrt{-b^2}d}{b}}}}{}$$

```
[In] Integrate[((a + b*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/Sqrt[c + d*Tan[e + f*x]],x]
```

```
[Out] (C*(a + b*Tan[e + f*x])^(3/2)*Sqrt[c + d*Tan[e + f*x]])/(2*d*f) + (((-3*b*c
*C + 4*b*B*d + 3*a*C*d)*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/
(2*d*f) + ((2*(b*(a^2*B - b^2*B + 2*a*b*(A - C)) - Sqrt[-b^2]*(2*a*b*B - a^
2*(A - C) + b^2*(A - C)))*d^2*ArcTanh[(Sqrt[-c + (Sqrt[-b^2]*d)/b]*Sqrt[a +
 b*Tan[e + f*x]])/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[
-a + Sqrt[-b^2]]*Sqrt[-c + (Sqrt[-b^2]*d)/b]) - (2*(b*(a^2*B - b^2*B + 2*a*
b*(A - C)) + Sqrt[-b^2]*(2*a*b*B - a^2*(A - C) + b^2*(A - C)))*d^2*ArcTanh[
(Sqrt[c + (Sqrt[-b^2]*d)/b]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + Sqrt[-b^2]]
*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[a + Sqrt[-b^2]]*Sqrt[c + (Sqrt[-b^2]*d)/
b]) + (Sqrt[b]*Sqrt[c - (a*d)/b]*(3*a^2*C*d^2 - 6*a*b*d*(c*C - 2*B*d) + b^2
*(3*c^2*C - 4*B*c*d + 8*(A - C)*d^2))*ArcSinh[(Sqrt[d]*Sqrt[a + b*Tan[e + f
*x]])/(Sqrt[b]*Sqrt[c - (a*d)/b])]*Sqrt[(b*c + b*d*Tan[e + f*x])/(b*c - a*d
)])/(2*Sqrt[d]*Sqrt[c + d*Tan[e + f*x]]))/(b*d*f))/(2*d)
```

Maple [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(fx+e))^{\frac{3}{2}}\left(A+B\tan(fx+e)+C\tan(fx+e)^2\right)}{\sqrt{c+d\tan(fx+e)}}dx$$

```
[In] int((a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(1/2),x)
```

```
[Out] int((a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(1/2),x)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 57789 vs. $2(309) = 618$.

Time = 271.99 (sec) , antiderivative size = 115594, normalized size of antiderivative = 301.81

$$\int \frac{(a+b\tan(e+fx))^{3/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{c+d\tan(e+fx)}}\,dx = \text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(1/2),x, algorithm="fricas")
```

```
[Out] Too large to include
```

Sympy [F]

$$\int \frac{(a+b\tan(e+fx))^{3/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{c+d\tan(e+fx)}}\,dx = \int \frac{(a+b\tan(e+fx))^{\frac{3}{2}}(A+B\tan(}{\sqrt{c+d\tan(e}}$$

```
[In] integrate((a+b*tan(f*x+e))**(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan
(f*x+e))**(1/2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)**
2)/sqrt(c + d*tan(e + f*x)), x)
```

Maxima [F]

$$\int \frac{(a+b\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx = \int \frac{\left(C\tan\left(fx+e\right)^2+B\tan\left(fx+e\right)\cdots}{\sqrt{d\tan\left(fx+e\cdots}}$$

```
[In] integrate((a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(1/2),x, algorithm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)^(3/2
)/sqrt(d*tan(f*x + e) + c), x)
```

Giac [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(1/2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx = \int \frac{\left(a+b\tan(e+f\,x)\right)^{3/2}\left(C\tan(e+\cdots}{\sqrt{c+d\tan\left(e\cdots}}$$

```
[In] int(((a + b*tan(e + f*x))^(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c
 + d*tan(e + f*x))^(1/2),x)
```

```
[Out] int(((a + b*tan(e + f*x))^(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c
 + d*tan(e + f*x))^(1/2), x)
```

3.149 $\int \frac{\sqrt{a+b\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{c+d\tan(e+fx)}}\,dx$

Optimal result . . . 2043
Rubi [A] (verified) . . . 2044
Mathematica [A] (verified) . . . 2047
Maple [F(-1)] . . . 2048
Fricas [B] (verification not implemented) . . . 2048
Sympy [F] . . . 2048
Maxima [F] . . . 2049
Giac [F(-1)] . . . 2049
Mupad [F(-1)] . . . 2049

Optimal result

Integrand size = 49, antiderivative size = 290

$$\begin{aligned}
&\int \frac{\sqrt{a+b\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{c+d\tan(e+fx)}}\,dx \\
&= -\frac{\sqrt{a-ib}(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{c-id}f} \\
&\quad+\frac{\sqrt{a+ib}(iA-B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{c+id}f} \\
&\quad-\frac{(bcC-2bBd-aCd)\operatorname{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{b}d^{3/2}f} \\
&\quad+\frac{C\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{df}
\end{aligned}$$

```
[Out] -(-2*B*b*d-C*a*d+C*b*c)*arctanh(d^(1/2)*(a+b*tan(f*x+e))^(1/2)/b^(1/2)/(c+d
*tan(f*x+e))^(1/2))/d^(3/2)/f/b^(1/2)-(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+
b*tan(f*x+e))^(1/2)/(a-I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))*(a-I*b)^(1/2)/f/(
c-I*d)^(1/2)+(I*A-B-I*C)*arctanh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*
b)^(1/2)/(c+d*tan(f*x+e))^(1/2))*(a+I*b)^(1/2)/f/(c+I*d)^(1/2)+C*(a+b*tan(f
*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)/d/f
```

Rubi [A] (verified)

Time = 3.14 (sec) , antiderivative size = 290, normalized size of antiderivative = 1.00, number of steps used = 13, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}} = 0.163$, Rules used = {3728, 3736, 6857, 65, 223, 212, 95, 214}

$$\int \frac{\sqrt{a+b\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{c+d\tan(e+fx)}}\,dx$$
$$= -\frac{\sqrt{a-ib}(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f\sqrt{c-id}}$$
$$+\frac{\sqrt{a+ib}(iA-B-iC)\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f\sqrt{c+id}}$$
$$-\frac{(-aCd-2bBd+bcC)\text{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{b}d^{3/2}f}$$
$$+\frac{C\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{df}$$

```
[In] Int[(Sqrt[a + b*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/Sqrt
[c + d*Tan[e + f*x]],x]
```

```
[Out] -((Sqrt[a - I*b]*(I*A + B - I*C)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[c - I*d]*f)) + (Sqr
t[a + I*b]*(I*A - B - I*C)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]])
/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[c + I*d]*f) - ((b*c*C - 2
*b*B*d - a*C*d)*ArcTanh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c
+ d*Tan[e + f*x]])])/(Sqrt[b]*d^(3/2)*f) + (C*Sqrt[a + b*Tan[e + f*x]]*Sqrt
[c + d*Tan[e + f*x]])/(d*f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 95

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 223

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3736

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[(a + b*ff*x)^m*(c + d*ff*x)^n*((A + B*ff*x + C*ff^2*x^2
)/(1 + ff^2*x^2)), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6857

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\text{integral} = \frac{C\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{df} + \frac{\int \frac{\frac{1}{2}(-bcC+2aAd-aCd)+(Ab+aB-bC)d\tan(e+fx)-\frac{1}{2}(bcC-2bBd-aCd)\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx}{d}$$

$$= \frac{C\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{df} + \frac{\text{Subst}\left(\int \frac{\frac{1}{2}(-bcC+2aAd-aCd)+(Ab+aB-bC)dx+\frac{1}{2}(-bcC+2bBd+aCd)x^2}{\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}\,dx, x, \tan(e+fx)\right)}{df}$$

$$= \frac{C\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{df} + \frac{\text{Subst}\left(\int \left(\frac{-bcC+2bBd+aCd}{2\sqrt{a+bx}\sqrt{c+dx}} + \frac{-((bB-a(A-C))d)+(Ab+aB-bC)dx}{\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}\right)dx, x, \tan(e+fx)\right)}{df}$$

$$= \frac{C\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{df} + \frac{\text{Subst}\left(\int \frac{-((bB-a(A-C))d)+(Ab+aB-bC)dx}{\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}\,dx, x, \tan(e+fx)\right)}{df} - \frac{(bcC-2bBd-aCd)\text{Subst}\left(\int \frac{1}{\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e+fx)\right)}{2df}$$

$$= \frac{C\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{df} + \frac{\text{Subst}\left(\int \left(\frac{-i(bB-a(A-C))d-(Ab+aB-bC)d}{2(i-x)\sqrt{a+bx}\sqrt{c+dx}} + \frac{-i(bB-a(A-C))d+(Ab+aB-bC)d}{2(i+x)\sqrt{a+bx}\sqrt{c+dx}}\right)dx, x, \tan(e+fx)\right)}{df} - \frac{(bcC-2bBd-aCd)\text{Subst}\left(\int \frac{1}{\sqrt{c-\frac{ad}{b}+\frac{dx^2}{b}}}\,dx, x, \sqrt{a+b\tan(e+fx)}\right)}{bdf}$$

$$= \frac{C\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{df} + \frac{((ia+b)(A-iB-C))\text{Subst}\left(\int \frac{1}{(i+x)\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e+fx)\right)}{2f} + \frac{((ia-b)(A+iB-C))\text{Subst}\left(\int \frac{1}{(i-x)\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e+fx)\right)}{2f} - \frac{(bcC-2bBd-aCd)\text{Subst}\left(\int \frac{1}{1-\frac{dx^2}{b}}\,dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{bdf}$$

$$= -\frac{(bcC-2bBd-aCd)\operatorname{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{b}d^{3/2}f} + \frac{C\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{df} + \frac{((ia+b)(A-iB-C))\text{Subst}\left(\int\frac{1}{-a+ib-(-c+id)x^2}\,dx,x,\frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{f} + \frac{((ia-b)(A+iB-C))\text{Subst}\left(\int\frac{1}{a+ib-(c+id)x^2}\,dx,x,\frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{f}$$

$$= -\frac{\sqrt{a-ib}(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{c-id}f} - \frac{\sqrt{a+ib}(B-i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{c+id}f} - \frac{(bcC-2bBd-aCd)\operatorname{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{b}d^{3/2}f} + \frac{C\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{df}$$

Mathematica [A] (verified)

Time = 7.04 (sec) , antiderivative size = 456, normalized size of antiderivative = 1.57

$$\int\frac{\sqrt{a+b\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{c+d\tan(e+fx)}}\,dx = \frac{C\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{df} + \frac{-\frac{\left(\sqrt{-b^2}(bB-a(A-C))-b(Ab+aB-bC)\right)d\operatorname{arctanh}\left(\frac{\sqrt{-c+\frac{\sqrt{-b^2}d}{b}}\sqrt{a+b\tan(e+fx)}}{\sqrt{-a+\sqrt{-b^2}}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{-a+\sqrt{-b^2}}\sqrt{-c+\frac{\sqrt{-b^2}d}{b}}} - \frac{\left(\sqrt{-b^2}(bB-a(A-C))+b(Ab+aB-bC)\right)d\operatorname{arct}}{\sqrt{a+\sqrt{-b^2}}\sqrt{c+}}}{bdf}$$

```
[In] Integrate[(Sqrt[a + b*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)
)/Sqrt[c + d*Tan[e + f*x]],x]
```

```
[Out] (C*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(d*f) + (-(((Sqrt[-b^
2]*(b*B - a*(A - C)) - b*(A*b + a*B - b*C))*d*ArcTanh[(Sqrt[-c + (Sqrt[-b^2
]*d)/b]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[c + d*Tan[e +
 f*x]])])/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[-c + (Sqrt[-b^2]*d)/b])) - ((Sqrt[-b^
2]*(b*B - a*(A - C)) + b*(A*b + a*B - b*C))*d*ArcTanh[(Sqrt[c + (Sqrt[-b^2]
```

```
*d)/b]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + Sqrt[-b^2]]*Sqrt[c + d*Tan[e + f
*x]])])/(Sqrt[a + Sqrt[-b^2]]*Sqrt[c + (Sqrt[-b^2]*d)/b]) - (Sqrt[b]*Sqrt[c
 - (a*d)/b]*(b*c*C - 2*b*B*d - a*C*d)*ArcSinh[(Sqrt[d]*Sqrt[a + b*Tan[e + f
*x]])/(Sqrt[b]*Sqrt[c - (a*d)/b])]*Sqrt[(b*c + b*d*Tan[e + f*x])/(b*c - a*d
)])/(Sqrt[d]*Sqrt[c + d*Tan[e + f*x]]))/(b*d*f)
```

Maple [F(-1)]

Timed out.

$$\int \frac{\sqrt{a+b\tan\left(fx+e\right)}\left(A+B\tan\left(fx+e\right)+C\tan\left(fx+e\right)^{2}\right)}{\sqrt{c+d\tan\left(fx+e\right)}}dx$$

```
[In] int((a+b*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(1/2),x)
```

```
[Out] int((a+b*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(1/2),x)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 38950 vs. 2(226) = 452.

Time = 129.20 (sec) , antiderivative size = 77916, normalized size of antiderivative = 268.68

$$\int \frac{\sqrt{a+b\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{c+d\tan(e+fx)}}\,dx=\text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(1/2),x, algorithm="fricas")
```

```
[Out] Too large to include
```

Sympy [F]

$$\int \frac{\sqrt{a+b\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{c+d\tan(e+fx)}}\,dx = \int \frac{\sqrt{a+b\tan\left(e+fx\right)}\left(A+B\tan\left(e+fx\right)+C\tan^{2}\left(e+fx\right)\right)}{\sqrt{c+d\tan\left(e+fx\right)}}\,dx$$

```
[In] integrate((a+b*tan(f*x+e))**(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan
(f*x+e))**(1/2),x)
```

```
[Out] Integral(sqrt(a + b*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/
sqrt(c + d*tan(e + f*x)), x)
```

Maxima [F]

$$\int \frac{\sqrt{a+b\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{c+d\tan(e+fx)}}\,dx$$
$$=\int \frac{\left(C\tan\left(fx+e\right)^2+B\tan\left(fx+e\right)+A\right)\sqrt{b\tan\left(fx+e\right)+a}}{\sqrt{d\tan\left(fx+e\right)+c}}\,dx$$

```
[In] integrate((a+b*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(1/2),x, algorithm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*sqrt(b*tan(f*x + e) + a)/
sqrt(d*tan(f*x + e) + c), x)
```

Giac [F(-1)]

Timed out.

$$\int \frac{\sqrt{a+b\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{c+d\tan(e+fx)}}\,dx=\text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(1/2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{\sqrt{a+b\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{\sqrt{c+d\tan(e+fx)}}\,dx=\text{Hanged}$$

```
[In] int(((a + b*tan(e + f*x))^(1/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c
 + d*tan(e + f*x))^(1/2),x)
```

```
[Out] \text{Hanged}
```

3.150 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx$

Optimal result 2050
Rubi [A] (verified) 2050
Mathematica [A] (verified) 2053
Maple [F(-1)] 2054
Fricas [B] (verification not implemented) 2054
Sympy [F] 2054
Maxima [F] 2055
Giac [F(-1)] 2055
Mupad [F(-1)] 2055

Optimal result

Integrand size = 49, antiderivative size = 239

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx = -\frac{(B+i(A-C))\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{a-ib}\sqrt{c-id}f} + \frac{(iA-B-iC)\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{a+ib}\sqrt{c+id}f} + \frac{2C\text{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{b}\sqrt{d}f}$$

```
[Out] -(B+I*(A-C))*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(1/2)/(c+
d*tan(f*x+e))^(1/2))/f/(a-I*b)^(1/2)/(c-I*d)^(1/2)+(I*A-B-I*C)*arctanh((c+I
*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/f/(a
+I*b)^(1/2)/(c+I*d)^(1/2)+2*C*arctanh(d^(1/2)*(a+b*tan(f*x+e))^(1/2)/b^(1/2
)/(c+d*tan(f*x+e))^(1/2))/f/b^(1/2)/d^(1/2)
```

Rubi [A] (verified)

Time = 1.76 (sec) , antiderivative size = 239, normalized size of antiderivative = 1.00, number of steps used = 12, number of rules used = 7, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.143, Rules used

$= \{3736, 6857, 65, 223, 212, 95, 214\}$

$$\int \frac{A + B\tan(e+fx) + C\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx$$
$$= -\frac{(B+i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f\sqrt{a-ib}\sqrt{c-id}} + \frac{(iA-B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f\sqrt{a+ib}\sqrt{c+id}} + \frac{2C\operatorname{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{b}\sqrt{d}f}$$

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/(Sqrt[a + b*Tan[e + f*x]]*Sqrt[
c + d*Tan[e + f*x]]),x]
```

```
[Out] -(((B + I*(A - C))*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a
 - I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[a - I*b]*Sqrt[c - I*d]*f)) + ((I*
A - B - I*C)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + I*b
]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[a + I*b]*Sqrt[c + I*d]*f) + (2*C*ArcTan
h[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])])/(
Sqrt[b]*Sqrt[d]*f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 95

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 223

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3736

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[(a + b*ff*x)^m*(c + d*ff*x)^n*((A + B*ff*x + C*ff^2*x^2
)/(1 + ff^2*x^2)), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6857

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\text{integral} = \frac{\text{Subst}\left(\int \frac{A+Bx+Cx^2}{\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}\,dx, x, \tan(e+fx)\right)}{f}$$

$$= \frac{\text{Subst}\left(\int \left(\frac{C}{\sqrt{a+bx}\sqrt{c+dx}} + \frac{A-C+Bx}{\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}\right)\,dx, x, \tan(e+fx)\right)}{f}$$

$$= \frac{\text{Subst}\left(\int \frac{A-C+Bx}{\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}\,dx, x, \tan(e+fx)\right)}{f} + \frac{C\text{Subst}\left(\int \frac{1}{\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e+fx)\right)}{f}$$

$$= \frac{\text{Subst}\left(\int \left(\frac{-B+i(A-C)}{2(i-x)\sqrt{a+bx}\sqrt{c+dx}} + \frac{B+i(A-C)}{2(i+x)\sqrt{a+bx}\sqrt{c+dx}}\right)\,dx, x, \tan(e+fx)\right)}{f}$$

$$+ \frac{(2C)\text{Subst}\left(\int \frac{1}{\sqrt{c-\frac{ad}{b}+\frac{dx^2}{b}}}\,dx, x, \sqrt{a+b\tan(e+fx)}\right)}{bf}$$

$$= \frac{(-B+i(A-C))\text{Subst}\left(\int \frac{1}{(i-x)\sqrt{a+bx}\sqrt{c+dx}}\,dx,x,\tan(e+fx)\right)}{2f} + \frac{(B+i(A-C))\text{Subst}\left(\int \frac{1}{(i+x)\sqrt{a+bx}\sqrt{c+dx}}\,dx,x,\tan(e+fx)\right)}{2f} + \frac{(2C)\text{Subst}\left(\int \frac{1}{1-\frac{dx^2}{b}}\,dx,x,\frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{bf}$$

$$= \frac{2C\operatorname{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{b}\sqrt{d}f} + \frac{(-B+i(A-C))\text{Subst}\left(\int \frac{1}{a+ib-(c+id)x^2}\,dx,x,\frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{f} + \frac{(B+i(A-C))\text{Subst}\left(\int \frac{1}{-a+ib-(-c+id)x^2}\,dx,x,\frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{f}$$

$$= -\frac{(B+i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{a-ib}\sqrt{c-id}f} - \frac{(B-i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{a+ib}\sqrt{c+id}f} + \frac{2C\operatorname{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{b}\sqrt{d}f}$$

Mathematica [A] (verified)

Time = 2.42 (sec) , antiderivative size = 362, normalized size of antiderivative = 1.51

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx = \frac{\frac{\left(bB+\sqrt{-b^2}(A-C)\right)\operatorname{arctanh}\left(\frac{\sqrt{-c+\frac{\sqrt{-b^2}d}{b}}\sqrt{a+b\tan(e+fx)}}{\sqrt{-a+\sqrt{-b^2}}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{-a+\sqrt{-b^2}}\sqrt{-c+\frac{\sqrt{-b^2}d}{b}}} - \frac{\left(bB+\sqrt{-b^2}(-A+C)\right)\operatorname{arctanh}\left(\frac{\sqrt{c+\frac{\sqrt{-b^2}d}{b}}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+\sqrt{-b^2}}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{a+\sqrt{-b^2}}\sqrt{c+\frac{\sqrt{-b^2}d}{b}}} + \frac{2\sqrt{b}C}{}}{bf}$$

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/(Sqrt[a + b*Tan[e + f*x]]
*Sqrt[c + d*Tan[e + f*x]]),x]
```

```
[Out] (((b*B + Sqrt[-b^2]*(A - C))*ArcTanh[(Sqrt[-c + (Sqrt[-b^2]*d)/b]*Sqrt[a +
b*Tan[e + f*x]])/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[c + d*Tan[e + f*x]])]])/(Sqrt[-
a + Sqrt[-b^2]]*Sqrt[-c + (Sqrt[-b^2]*d)/b]) - ((b*B + Sqrt[-b^2]*(-A + C))
*ArcTanh[(Sqrt[c + (Sqrt[-b^2]*d)/b]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + Sq
rt[-b^2]]*Sqrt[c + d*Tan[e + f*x]])]])/(Sqrt[a + Sqrt[-b^2]]*Sqrt[c + (Sqrt[
-b^2]*d)/b]) + (2*Sqrt[b]*C*Sqrt[c - (a*d)/b]*ArcSinh[(Sqrt[d]*Sqrt[a + b*T
an[e + f*x]])/(Sqrt[b]*Sqrt[c - (a*d)/b])]*Sqrt[(b*(c + d*Tan[e + f*x]))/(b
*c - a*d)])/(Sqrt[d]*Sqrt[c + d*Tan[e + f*x]]))/(b*f)
```

Maple [F(-1)]

Timed out.

$$\int \frac{A+B\tan\left(fx+e\right)+C\tan\left(fx+e\right)^2}{\sqrt{a+b\tan\left(fx+e\right)}\,\sqrt{c+d\tan\left(fx+e\right)}}dx$$

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(1/2)/(c+d*tan(f*x+e))
^(1/2),x)
```

```
[Out] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(1/2)/(c+d*tan(f*x+e))
^(1/2),x)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 48154 vs. $2(181) = 362$.

Time = 141.31 (sec) , antiderivative size = 96324, normalized size of antiderivative = 403.03

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx = \text{Too large to display}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(1/2)/(c+d*tan(f
*x+e))^(1/2),x, algorithm="fricas")
```

```
[Out] Too large to include
```

Sympy [F]

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx = \int \frac{A+B\tan\left(e+fx\right)+C\tan^2\left(e+fx\right)}{\sqrt{a+b\tan\left(e+fx\right)}\sqrt{c+d\tan\left(e+fx\right)}}\,dx$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e))**(1/2)/(c+d*tan
(f*x+e))**(1/2),x)
```

```
[Out] Integral((A + B*tan(e + f*x) + C*tan(e + f*x)**2)/(sqrt(a + b*tan(e + f*x))
*sqrt(c + d*tan(e + f*x))), x)
```

Maxima [F]

$$\int \frac{A + B\tan(e+fx) + C\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx = \int \frac{C\tan\left(fx+e\right)^2 + B\tan\left(fx+e\right) + A}{\sqrt{b\tan\left(fx+e\right)+a}\sqrt{d\tan\left(fx+e\right)+c}}\,dx$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(1/2)/(c+d*tan(f
*x+e))^(1/2),x, algorithm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)/(sqrt(b*tan(f*x + e) + a)
*sqrt(d*tan(f*x + e) + c)), x)
```

Giac [F(-1)]

Timed out.

$$\int \frac{A + B\tan(e+fx) + C\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx = \text{Timed out}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(1/2)/(c+d*tan(f
*x+e))^(1/2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{A + B\tan(e+fx) + C\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx = \text{Hanged}$$

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))^(1/2)*(c
+ d*tan(e + f*x))^(1/2)),x)
```

```
[Out] \text{Hanged}
```

3.151 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}\,dx$

Optimal result 2056
Rubi [A] (verified) 2056
Mathematica [A] (verified) 2059
Maple [F(-1)] 2059
Fricas [B] (verification not implemented) 2060
Sympy [F] 2060
Maxima [F] 2060
Giac [F(-1)] 2061
Mupad [F(-1)] 2061

Optimal result

Integrand size = 49, antiderivative size = 251

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}\,dx = -\frac{(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a-ib)^{3/2}\sqrt{c-id}f} - \frac{(B-i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a+ib)^{3/2}\sqrt{c+id}f} - \frac{2(Ab^2-a(bB-aC))\sqrt{c+d\tan(e+fx)}}{(a^2+b^2)(bc-ad)f\sqrt{a+b\tan(e+fx)}}$$

```
[Out] -(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(1/2)/(c+
d*tan(f*x+e))^(1/2))/(a-I*b)^(3/2)/f/(c-I*d)^(1/2)-(B-I*(A-C))*arctanh((c+I
*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/(a+I
*b)^(3/2)/f/(c+I*d)^(1/2)-2*(A*b^2-a*(B*b-C*a))*(c+d*tan(f*x+e))^(1/2)/(a^2
+b^2)/(-a*d+b*c)/f/(a+b*tan(f*x+e))^(1/2)
```

Rubi [A] (verified)

Time = 1.15 (sec) , antiderivative size = 251, normalized size of antiderivative = 1.00, number of steps used = 8, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.102, Rules used

$= \{3730, 3697, 3696, 95, 214\}$

$$\int \frac{A + B\tan(e+fx) + C\tan^2(e+fx)}{(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}\,dx = -\frac{2(Ab^2 - a(bB - aC))\sqrt{c+d\tan(e+fx)}}{f(a^2+b^2)(bc-ad)\sqrt{a+b\tan(e+fx)}} - \frac{(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(a-ib)^{3/2}\sqrt{c-id}} - \frac{(B-i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(a+ib)^{3/2}\sqrt{c+id}}$$

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^(3/2)*Sqr
t[c + d*Tan[e + f*x]]),x]
```

```
[Out] -(((I*A + B - I*C)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a
 - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a - I*b)^(3/2)*Sqrt[c - I*d]*f)) - ((
B - I*(A - C))*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + I
*b]*Sqrt[c + d*Tan[e + f*x]])])/((a + I*b)^(3/2)*Sqrt[c + I*d]*f) - (2*(A*b
^2 - a*(b*B - a*C))*Sqrt[c + d*Tan[e + f*x]])/((a^2 + b^2)*(b*c - a*d)*f*Sq
rt[a + b*Tan[e + f*x]])
```

Rule 95

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3696

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[A^2/f, Subst[Int[(a + b*x)^m*((c + d*x)^n/(A - B*x)), x], x, Tan[e + f*x
]], x] /; FreeQ[{a, b, c, d, e, f, A, B, m, n}, x] && NeQ[b*c - a*d, 0] &&
NeQ[a^2 + b^2, 0] && EqQ[A^2 + B^2, 0]
```

Rule 3697

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[(A + I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^n*(1 - I*Ta
n[e + f*x]), x], x] + Dist[(A - I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*T
an[e + f*x])^n*(1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A,
B, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[A^2 + B^2, 0]
```

Rule 3730

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 - a*(b*B - a*C))*(a + b*Tan[e +
 f*x])^(m + 1)*((c + d*Tan[e + f*x])^(n + 1)/(f*(m + 1)*(b*c - a*d)*(a^2 +
b^2))), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rubi steps

$$\begin{aligned}
\text{integral} &= -\frac{2(Ab^2-a(bB-aC))\sqrt{c+d\tan(e+fx)}}{(a^2+b^2)(bc-ad)f\sqrt{a+b\tan(e+fx)}} \\
&\quad -\frac{2\int\frac{-\frac{1}{2}(bB+a(A-C))(bc-ad)+\frac{1}{2}(Ab-aB-bC)(bc-ad)\tan(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx}{(a^2+b^2)(bc-ad)} \\
&= -\frac{2(Ab^2-a(bB-aC))\sqrt{c+d\tan(e+fx)}}{(a^2+b^2)(bc-ad)f\sqrt{a+b\tan(e+fx)}} \\
&\quad +\frac{(A-iB-C)\int\frac{1+i\tan(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx}{2(a-ib)} \\
&\quad +\frac{(A+iB-C)\int\frac{1-i\tan(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx}{2(a+ib)} \\
&= -\frac{2(Ab^2-a(bB-aC))\sqrt{c+d\tan(e+fx)}}{(a^2+b^2)(bc-ad)f\sqrt{a+b\tan(e+fx)}} \\
&\quad +\frac{(A-iB-C)\text{Subst}\left(\int\frac{1}{(1-ix)\sqrt{a+bx}\sqrt{c+dx}}\,dx,x,\tan(e+fx)\right)}{2(a-ib)f} \\
&\quad +\frac{(A+iB-C)\text{Subst}\left(\int\frac{1}{(1+ix)\sqrt{a+bx}\sqrt{c+dx}}\,dx,x,\tan(e+fx)\right)}{2(a+ib)f}
\end{aligned}$$

$$= -\frac{2(Ab^2 - a(bB - aC))\sqrt{c + d\tan(e + fx)}}{(a^2 + b^2)(bc - ad)f\sqrt{a + b\tan(e + fx)}} + \frac{(A - iB - C)\text{Subst}\left(\int \frac{1}{ia+b-(ic+d)x^2}\,dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{(a - ib)f} + \frac{(A + iB - C)\text{Subst}\left(\int \frac{1}{-ia+b-(-ic+d)x^2}\,dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{(a + ib)f}$$

$$= -\frac{(iA + B - iC)\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a - ib)^{3/2}\sqrt{c - id}f} - \frac{(B - i(A - C))\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a + ib)^{3/2}\sqrt{c + id}f} - \frac{2(Ab^2 - a(bB - aC))\sqrt{c + d\tan(e + fx)}}{(a^2 + b^2)(bc - ad)f\sqrt{a + b\tan(e + fx)}}$$

Mathematica [A] (verified)

Time = 2.87 (sec) , antiderivative size = 264, normalized size of antiderivative = 1.05

$$\int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{(a + b\tan(e + fx))^{3/2}\sqrt{c + d\tan(e + fx)}}\,dx = \frac{\frac{(a+ib)(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{-c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{-a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{-a+ib}\sqrt{-c+id}} + \frac{(ia+b)(A+iB-}{}}{(a^2}$$

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^(3/
2)*Sqrt[c + d*Tan[e + f*x]]),x]
```

```
[Out] (((a + I*b)*(I*A + B - I*C)*ArcTanh[(Sqrt[-c + I*d]*Sqrt[a + b*Tan[e + f*x]
])/(Sqrt[-a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[-a + I*b]*Sqrt[-c + I*
d]) + ((I*a + b)*(A + I*B - C)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*
x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[a + I*b]*Sqrt[c + I*d
]) + (2*(A*b^2 + a*(-(b*B) + a*C))*Sqrt[c + d*Tan[e + f*x]])/((-(b*c) + a*d
)*Sqrt[a + b*Tan[e + f*x]]))/((a^2 + b^2)*f)
```

Maple [F(-1)]

Timed out.

$$\int \frac{A + B\tan(fx + e) + C\tan(fx + e)^2}{\sqrt{c + d\tan(fx + e)}\,(a + b\tan(fx + e))^{\frac{3}{2}}}\,dx$$

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f*x+e))
^(3/2),x)
```

```
[Out] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f*x+e))
^(3/2),x)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 83974 vs. 2(200) = 400.

Time = 274.81 (sec) , antiderivative size = 83974, normalized size of antiderivative = 334.56

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}\,dx = \text{Too large to display}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f
*x+e))^(3/2),x, algorithm="fricas")
```

```
[Out] Too large to include
```

Sympy [F]

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}\,dx = \int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{\frac{3}{2}}\sqrt{c+d\tan(e+fx)}}\,dx$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x+e))**(1/2)/(a+b*tan
(f*x+e))**(3/2),x)
```

```
[Out] Integral((A + B*tan(e + f*x) + C*tan(e + f*x)**2)/((a + b*tan(e + f*x))**(3
/2)*sqrt(c + d*tan(e + f*x))), x)
```

Maxima [F]

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}\,dx = \int \frac{C\tan(fx+e)^2+B\tan(fx+e)+A}{(b\tan(fx+e)+a)^{\frac{3}{2}}\sqrt{d\tan(fx+e)+c}}\,dx$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f
*x+e))^(3/2),x, algorithm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)/((b*tan(f*x + e) + a)^(3/
2)*sqrt(d*tan(f*x + e) + c)), x)
```

Giac [F(-1)]

Timed out.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}\,dx = \text{Timed out}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f
*x+e))^(3/2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}\,dx = \text{Hanged}$$

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))^(3/2)*(c
+ d*tan(e + f*x))^(1/2)),x)
```

```
[Out] \text{Hanged}
```

3.152 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{5/2}\sqrt{c+d\tan(e+fx)}}\,dx$

Optimal result 2062
Rubi [A] (verified) 2063
Mathematica [A] (verified) 2066
Maple [F(-1)] 2066
Fricas [F(-1)] 2066
Sympy [F] 2067
Maxima [F(-1)] 2067
Giac [F(-1)] 2067
Mupad [F(-1)] 2067

Optimal result

Integrand size = 49, antiderivative size = 375

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{5/2}\sqrt{c+d\tan(e+fx)}}\,dx = -\frac{(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a-ib)^{5/2}\sqrt{c-id}f} - \frac{(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a+ib)^{5/2}\sqrt{c+id}f} - \frac{2(Ab^2-a(bB-aC))\sqrt{c+d\tan(e+fx)}}{3(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))^{3/2}} - \frac{2(5a^3bBd-2a^4Cd+b^4(3Bc-2Ad)+ab^3(6Ac-6cC-Bd)-a^2b^2(3Bc+8Ad-4Cd))\sqrt{c+d\tan(e+}}{3(a^2+b^2)^2(bc-ad)^2f\sqrt{a+b\tan(e+fx)}}$$

```
[Out] -(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(1/2)/(c+
d*tan(f*x+e))^(1/2))/(a-I*b)^(5/2)/f/(c-I*d)^(1/2)-(B-I*(A-C))*arctanh((c+I
*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/(a+I
*b)^(5/2)/f/(c+I*d)^(1/2)-2/3*(5*a^3*b*B*d-2*a^4*C*d+b^4*(-2*A*d+3*B*c)+a*b
^3*(6*A*c-B*d-6*C*c)-a^2*b^2*(8*A*d+3*B*c-4*C*d))*(c+d*tan(f*x+e))^(1/2)/(a
^2+b^2)^2/(-a*d+b*c)^2/f/(a+b*tan(f*x+e))^(1/2)-2/3*(A*b^2-a*(B*b-C*a))*(c+
d*tan(f*x+e))^(1/2)/(a^2+b^2)/(-a*d+b*c)/f/(a+b*tan(f*x+e))^(3/2)
```

Rubi [A] (verified)

Time = 2.08 (sec) , antiderivative size = 375, normalized size of antiderivative = 1.00, number of steps used = 9, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.102, Rules used = {3730, 3697, 3696, 95, 214}

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{5/2}\sqrt{c+d\tan(e+fx)}}\,dx=$$
$$-\frac{2(Ab^2-a(bB-aC))\sqrt{c+d\tan(e+fx)}}{3f\,(a^2+b^2)\,(bc-ad)(a+b\tan(e+fx))^{3/2}}$$
$$-\frac{2\sqrt{c+d\tan(e+fx)}(-2a^4Cd+5a^3bBd-a^2b^2(8Ad+3Bc-4Cd)+ab^3(6Ac-Bd-6cC)+b^4(3Bc-}{3f\,(a^2+b^2)^2\,(bc-ad)^2\sqrt{a+b\tan(e+fx)}}$$
$$-\frac{(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(a-ib)^{5/2}\sqrt{c-id}}$$
$$-\frac{(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(a+ib)^{5/2}\sqrt{c+id}}$$

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^(5/2)*Sqr
t[c + d*Tan[e + f*x]]),x]
```

```
[Out] -(((I*A + B - I*C)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a
 - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a - I*b)^(5/2)*Sqrt[c - I*d]*f)) - ((
B - I*(A - C))*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + I
*b]*Sqrt[c + d*Tan[e + f*x]])])/((a + I*b)^(5/2)*Sqrt[c + I*d]*f) - (2*(A*b
^2 - a*(b*B - a*C))*Sqrt[c + d*Tan[e + f*x]])/(3*(a^2 + b^2)*(b*c - a*d)*f*
(a + b*Tan[e + f*x])^(3/2)) - (2*(5*a^3*b*B*d - 2*a^4*C*d + b^4*(3*B*c - 2*
A*d) + a*b^3*(6*A*c - 6*c*C - B*d) - a^2*b^2*(3*B*c + 8*A*d - 4*C*d))*Sqrt[
c + d*Tan[e + f*x]])/(3*(a^2 + b^2)^2*(b*c - a*d)^2*f*Sqrt[a + b*Tan[e + f*
x]])
```

Rule 95

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3696

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[A^2/f, Subst[Int[(a + b*x)^m*((c + d*x)^n/(A - B*x)), x], x, Tan[e + f*x
]], x] /; FreeQ[{a, b, c, d, e, f, A, B, m, n}, x] && NeQ[b*c - a*d, 0] &&
NeQ[a^2 + b^2, 0] && EqQ[A^2 + B^2, 0]
```

Rule 3697

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[(A + I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^n*(1 - I*Ta
n[e + f*x]), x], x] + Dist[(A - I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*T
an[e + f*x])^n*(1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A,
B, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[A^2 + B^2, 0]
```

Rule 3730

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 - a*(b*B - a*C))*(a + b*Tan[e +
 f*x])^(m + 1)*((c + d*Tan[e + f*x])^(n + 1)/(f*(m + 1)*(b*c - a*d)*(a^2 +
b^2))), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rubi steps

$$\text{integral} = -\frac{2(Ab^2-a(bB-aC))\sqrt{c+d\tan(e+fx)}}{3(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))^{3/2}} - \frac{2\int\frac{\frac{1}{2}(2Ab^2d-3aA(bc-ad)-(bB-aC)(3bc-ad))+\frac{3}{2}(Ab-aB-bC)(bc-ad)\tan(e+fx)+(Ab^2-a(bB-aC))d\tan^2(e+fx)}{(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}\,dx}{3(a^2+b^2)(bc-ad)}$$

$$= -\frac{2(Ab^2-a(bB-aC))\sqrt{c+d\tan(e+fx)}}{3(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))^{3/2}} - \frac{2(5a^3bBd-2a^4Cd+b^4(3Bc-2Ad)+ab^3(6Ac-6cC-Bd)-a^2b^2(3Bc+8Ad-4Cd))\sqrt{c+}}{3(a^2+b^2)^2(bc-ad)^2f\sqrt{a+b\tan(e+fx)}} + \frac{4\int\frac{\frac{3}{4}(2abB+a^2(A-C)-b^2(A-C))(bc-ad)^2+\frac{3}{4}(a^2B-b^2B-2ab(A-C))(bc-ad)^2\tan(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx}{3(a^2+b^2)^2(bc-ad)^2}$$

$$= -\frac{2(Ab^2 - a(bB - aC))\sqrt{c + d\tan(e + fx)}}{3(a^2 + b^2)(bc - ad)f(a + b\tan(e + fx))^{3/2}} - \frac{2(5a^3bBd - 2a^4Cd + b^4(3Bc - 2Ad) + ab^3(6Ac - 6cC - Bd) - a^2b^2(3Bc + 8Ad - 4Cd))\sqrt{c}}{3(a^2 + b^2)^2(bc - ad)^2f\sqrt{a + b\tan(e + fx)}} + \frac{(A - iB - C)\int \frac{1 + i\tan(e + fx)}{\sqrt{a + b\tan(e + fx)}\sqrt{c + d\tan(e + fx)}}\,dx}{2(a - ib)^2} + \frac{(A + iB - C)\int \frac{1 - i\tan(e + fx)}{\sqrt{a + b\tan(e + fx)}\sqrt{c + d\tan(e + fx)}}\,dx}{2(a + ib)^2}$$

$$= -\frac{2(Ab^2 - a(bB - aC))\sqrt{c + d\tan(e + fx)}}{3(a^2 + b^2)(bc - ad)f(a + b\tan(e + fx))^{3/2}} - \frac{2(5a^3bBd - 2a^4Cd + b^4(3Bc - 2Ad) + ab^3(6Ac - 6cC - Bd) - a^2b^2(3Bc + 8Ad - 4Cd))\sqrt{c}}{3(a^2 + b^2)^2(bc - ad)^2f\sqrt{a + b\tan(e + fx)}} + \frac{(A - iB - C)\text{Subst}\left(\int \frac{1}{(1 - ix)\sqrt{a + bx}\sqrt{c + dx}}\,dx, x, \tan(e + fx)\right)}{2(a - ib)^2f} + \frac{(A + iB - C)\text{Subst}\left(\int \frac{1}{(1 + ix)\sqrt{a + bx}\sqrt{c + dx}}\,dx, x, \tan(e + fx)\right)}{2(a + ib)^2f}$$

$$= -\frac{2(Ab^2 - a(bB - aC))\sqrt{c + d\tan(e + fx)}}{3(a^2 + b^2)(bc - ad)f(a + b\tan(e + fx))^{3/2}} - \frac{2(5a^3bBd - 2a^4Cd + b^4(3Bc - 2Ad) + ab^3(6Ac - 6cC - Bd) - a^2b^2(3Bc + 8Ad - 4Cd))\sqrt{c}}{3(a^2 + b^2)^2(bc - ad)^2f\sqrt{a + b\tan(e + fx)}} + \frac{(A - iB - C)\text{Subst}\left(\int \frac{1}{ia + b - (ic + d)x^2}\,dx, x, \frac{\sqrt{a + b\tan(e + fx)}}{\sqrt{c + d\tan(e + fx)}}\right)}{(a - ib)^2f} + \frac{(A + iB - C)\text{Subst}\left(\int \frac{1}{-ia + b - (-ic + d)x^2}\,dx, x, \frac{\sqrt{a + b\tan(e + fx)}}{\sqrt{c + d\tan(e + fx)}}\right)}{(a + ib)^2f}$$

$$= -\frac{(iA + B - iC)\text{arctanh}\left(\frac{\sqrt{c - id}\sqrt{a + b\tan(e + fx)}}{\sqrt{a - ib}\sqrt{c + d\tan(e + fx)}}\right)}{(a - ib)^{5/2}\sqrt{c - id}f} - \frac{(B - i(A - C))\text{arctanh}\left(\frac{\sqrt{c + id}\sqrt{a + b\tan(e + fx)}}{\sqrt{a + ib}\sqrt{c + d\tan(e + fx)}}\right)}{(a + ib)^{5/2}\sqrt{c + id}f} - \frac{2(Ab^2 - a(bB - aC))\sqrt{c + d\tan(e + fx)}}{3(a^2 + b^2)(bc - ad)f(a + b\tan(e + fx))^{3/2}} - \frac{2(5a^3bBd - 2a^4Cd + b^4(3Bc - 2Ad) + ab^3(6Ac - 6cC - Bd) - a^2b^2(3Bc + 8Ad - 4Cd))\sqrt{c}}{3(a^2 + b^2)^2(bc - ad)^2f\sqrt{a + b\tan(e + fx)}}$$

Mathematica [A] (verified)

Time = 6.66 (sec) , antiderivative size = 388, normalized size of antiderivative = 1.03

$$\int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{(a + b\tan(e + fx))^{5/2}\sqrt{c + d\tan(e + fx)}}\,dx = \frac{3(a+ib)^2(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{-c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{-a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{-a+ib}\sqrt{-c+id}} + \frac{3i(a-ib)^2(A+i}{}$$

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^(5/
2)*Sqrt[c + d*Tan[e + f*x]]),x]
```

```
[Out] ((3*(a + I*b)^2*(I*A + B - I*C)*ArcTanh[(Sqrt[-c + I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[-a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[-a + I*b]*Sqrt[-c
+ I*d]) + ((3*I)*(a - I*b)^2*(A + I*B - C)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a +
b*Tan[e + f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[a + I*b]*
Sqrt[c + I*d]) + (2*(a^2 + b^2)*(A*b^2 + a*(-(b*B) + a*C))*Sqrt[c + d*Tan[e
 + f*x]])/((-(b*c) + a*d)*(a + b*Tan[e + f*x])^(3/2)) + (2*(-5*a^3*b*B*d +
2*a^4*C*d + b^4*(-3*B*c + 2*A*d) + a*b^3*(-6*A*c + 6*c*C + B*d) + a^2*b^2*(
3*B*c + 8*A*d - 4*C*d))*Sqrt[c + d*Tan[e + f*x]])/((b*c - a*d)^2*Sqrt[a + b
*Tan[e + f*x]]))/(3*(a^2 + b^2)^2*f)
```

Maple [F(-1)]

Timed out.

$$\int \frac{A + B\tan(fx + e) + C\tan(fx + e)^2}{\sqrt{c + d\tan(fx + e)}\,(a + b\tan(fx + e))^{\frac{5}{2}}}\,dx$$

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f*x+e))
^(5/2),x)
```

```
[Out] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f*x+e))
^(5/2),x)
```

Fricas [F(-1)]

Timed out.

$$\int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{(a + b\tan(e + fx))^{5/2}\sqrt{c + d\tan(e + fx)}}\,dx = \text{Timed out}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f
*x+e))^(5/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{5/2}\sqrt{c+d\tan(e+fx)}}\,dx=\int \frac{A+B\tan{(e+fx)}+C\tan^2{(e+fx)}}{(a+b\tan{(e+fx)})^{\frac{5}{2}}\sqrt{c+d\tan{(e+fx)}}}\,dx$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x+e))**(1/2)/(a+b*tan
(f*x+e))**(5/2),x)
[Out] Integral((A + B*tan(e + f*x) + C*tan(e + f*x)**2)/((a + b*tan(e + f*x))**(5
/2)*sqrt(c + d*tan(e + f*x))), x)
```

Maxima [F(-1)]

Timed out.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{5/2}\sqrt{c+d\tan(e+fx)}}\,dx=\text{Timed out}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f
*x+e))^(5/2),x, algorithm="maxima")
[Out] Timed out
```

Giac [F(-1)]

Timed out.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{5/2}\sqrt{c+d\tan(e+fx)}}\,dx=\text{Timed out}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f
*x+e))^(5/2),x, algorithm="giac")
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{5/2}\sqrt{c+d\tan(e+fx)}}\,dx=\text{Hanged}$$

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))^(5/2)*(c
+ d*tan(e + f*x))^(1/2)),x)
[Out] \text{Hanged}
```

3.153 $\int \frac{(a+b\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx$

Optimal result . . . 2068
Rubi [A] (verified) . . . 2069
Mathematica [B] (verified) . . . 2074
Maple [F(-1)] . . . 2075
Fricas [F(-1)] . . . 2076
Sympy [F] . . . 2076
Maxima [F(-1)] . . . 2076
Giac [F(-1)] . . . 2076
Mupad [F(-1)] . . . 2077

Optimal result

Integrand size = 49, antiderivative size = 528

$$\int \frac{(a+b\tan(e+fx))^{5/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx =$$
$$-\frac{(a-ib)^{5/2}(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(c-id)^{3/2}f}$$
$$-\frac{(a+ib)^{5/2}(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(c+id)^{3/2}f}$$
$$+\frac{\sqrt{b}(15a^2Cd^2-10abd(3cC-2Bd)+b^2(15c^2C-12Bcd+8(A-C)d^2))\,\text{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{4d^{7/2}f}$$
$$-\frac{2(c^2C-Bcd+Ad^2)\,(a+b\tan(e+fx))^{5/2}}{d\,(c^2+d^2)\,f\sqrt{c+d\tan(e+fx)}}$$
$$-\frac{b(3(bc-ad)\,(5c^2C-4Bcd+(4A+C)d^2)-4d^2((A-C)(bc-ad)+B(ac+bd)))\,\sqrt{a+b\tan(e+fx)}\sqrt{c}}{4d^3\,(c^2+d^2)\,f}$$
$$+\frac{b(5c^2C-4Bcd+(4A+C)d^2)\,(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}{2d^2\,(c^2+d^2)\,f}$$

```
[Out] -(a-I*b)^(5/2)*(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-
I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/(c-I*d)^(3/2)/f-(a+I*b)^(5/2)*(B-I*(A-C)
)*arctanh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e
))^(1/2))/(c+I*d)^(3/2)/f+1/4*(15*a^2*C*d^2-10*a*b*d*(-2*B*d+3*C*c)+b^2*(15
*c^2*C-12*B*c*d+8*(A-C)*d^2))*arctanh(d^(1/2)*(a+b*tan(f*x+e))^(1/2)/b^(1/2
)/(c+d*tan(f*x+e))^(1/2))*b^(1/2)/d^(7/2)/f-1/4*b*(3*(-a*d+b*c)*(5*c^2*C-4*
B*c*d+(4*A+C)*d^2)-4*d^2*((A-C)*(-a*d+b*c)+B*(a*c+b*d)))*(a+b*tan(f*x+e))^(
1/2)*(c+d*tan(f*x+e))^(1/2)/d^3/(c^2+d^2)/f+1/2*b*(5*c^2*C-4*B*c*d+(4*A+C)*
```

```
d^2)*(c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^(3/2)/d^2/(c^2+d^2)/f-2*(A*d^2
-B*c*d+C*c^2)*(a+b*tan(f*x+e))^(5/2)/d/(c^2+d^2)/f/(c+d*tan(f*x+e))^(1/2)
```

Rubi [A] (verified)

Time = 9.81 (sec) , antiderivative size = 528, normalized size of antiderivative = 1.00, number of steps used = 15, number of rules used = 9, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.184, Rules used = {3726, 3728, 3736, 6857, 65, 223, 212, 95, 214}

$$\int \frac{(a+b\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{3/2}}\,dx = \frac{\sqrt{b}(15a^2Cd^2-10abd(3cC-2Bd)+}{}$$

$$-\frac{(a-ib)^{5/2}(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(c-id)^{3/2}}$$

$$-\frac{(a+ib)^{5/2}(B-i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(c+id)^{3/2}}$$

$$+\frac{b(d^2(4A+C)-4Bcd+5c^2C)\,(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}{2d^2f\,(c^2+d^2)}$$

$$-\frac{2(Ad^2-Bcd+c^2C)\,(a+b\tan(e+fx))^{5/2}}{df\,(c^2+d^2)\sqrt{c+d\tan(e+fx)}}$$

$$-\frac{b\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}(3(bc-ad)\,(d^2(4A+C)-4Bcd+5c^2C)-4d^2((A-C)(bc-a}{4d^3f\,(c^2+d^2)}$$

```
[In] Int[((a + b*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c
 + d*Tan[e + f*x])^(3/2),x]
```

```
[Out] -(((a - I*b)^(5/2)*(I*A + B - I*C)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e
+ f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((c - I*d)^(3/2)*f)) -
((a + I*b)^(5/2)*(B - I*(A - C))*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/((c + I*d)^(3/2)*f) + (Sq
rt[b]*(15*a^2*C*d^2 - 10*a*b*d*(3*c*C - 2*B*d) + b^2*(15*c^2*C - 12*B*c*d +
 8*(A - C)*d^2))*ArcTanh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c
 + d*Tan[e + f*x]])])/(4*d^(7/2)*f) - (2*(c^2*C - B*c*d + A*d^2)*(a + b*Tan
[e + f*x])^(5/2))/(d*(c^2 + d^2)*f*Sqrt[c + d*Tan[e + f*x]]) - (b*(3*(b*c -
 a*d)*(5*c^2*C - 4*B*c*d + (4*A + C)*d^2) - 4*d^2*((A - C)*(b*c - a*d) + B*
(a*c + b*d)))*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(4*d^3*(c^
2 + d^2)*f) + (b*(5*c^2*C - 4*B*c*d + (4*A + C)*d^2)*(a + b*Tan[e + f*x])^(
3/2)*Sqrt[c + d*Tan[e + f*x]])/(2*d^2*(c^2 + d^2)*f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
```

```
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 95

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 223

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3726

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*d^2 + c*(c*C - B*d))*(a + b*Tan[e +
 f*x])^m*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
```

```
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3736

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[(a + b*ff*x)^m*(c + d*ff*x)^n*((A + B*ff*x + C*ff^2*x^2
)/(1 + ff^2*x^2)), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6857

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\text{integral} = -\frac{2(c^2C-Bcd+Ad^2)\,(a+b\tan(e+fx))^{5/2}}{d\,(c^2+d^2)\,f\sqrt{c+d\tan(e+fx)}} + \frac{2\int \frac{(a+b\tan(e+fx))^{3/2}\left(\frac{1}{2}(Ad(ac+5bd)+(5bc-ad)(cC-Bd))+\frac{1}{2}d((A-C)(bc-ad)+B(ac+bd))\tan(e+fx)+\frac{1}{2}b\left(5c^2C-4Bcd+(4A+C)d^2\right)\tan\ldots}{\sqrt{c+d\tan(e+fx)}}}{d\,(c^2+d^2)}$$

$$= -\frac{2(c^2C-Bcd+Ad^2)\,(a+b\tan(e+fx))^{5/2}}{d\,(c^2+d^2)\,f\sqrt{c+d\tan(e+fx)}} + \frac{b(5c^2C-4Bcd+(4A+C)d^2)\,(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}{2d^2\,(c^2+d^2)\,f} + \frac{\int \frac{\sqrt{a+b\tan(e+fx)}\left(\frac{1}{4}\left(-b(3bc+ad)\left(5c^2C-4Bcd+(4A+C)d^2\right)+4ad(Ad(ac+5bd)+(5bc-ad)(cC-Bd))\right)+d^2\left(2ab(Ac-cC+Bd)+\ldots\right.\right.}{\sqrt{c\ldots}}}{d\ldots}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)(a + b\tan(e + fx))^{5/2}}{d(c^2 + d^2) f\sqrt{c + d\tan(e + fx)}} - \frac{b(3(bc - ad)(5c^2C - 4Bcd + (4A + C)d^2) - 4d^2((A - C)(bc - ad) + B(ac + bd)))\sqrt{a + b\tan(}}{4d^3(c^2 + d^2) f} + \frac{b(5c^2C - 4Bcd + (4A + C)d^2)(a + b\tan(e + fx))^{3/2}\sqrt{c + d\tan(e + fx)}}{2d^2(c^2 + d^2) f} + \frac{\int \frac{1}{8}(8a^3d^3(Ac - cC + Bd) + 3a^2bd^2(5c^2C - 8Bcd + (8A - 3C)d^2) + b^3c(15c^3C - 12Bc^2d + c(8A + 7C)d^2 - 4Bd^3) - 2ab^2d(15c^3C - 10Bc^2d}{}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)(a + b\tan(e + fx))^{5/2}}{d(c^2 + d^2) f\sqrt{c + d\tan(e + fx)}} - \frac{b(3(bc - ad)(5c^2C - 4Bcd + (4A + C)d^2) - 4d^2((A - C)(bc - ad) + B(ac + bd)))\sqrt{a + b\tan(}}{4d^3(c^2 + d^2) f} + \frac{b(5c^2C - 4Bcd + (4A + C)d^2)(a + b\tan(e + fx))^{3/2}\sqrt{c + d\tan(e + fx)}}{2d^2(c^2 + d^2) f} + \frac{\text{Subst}\left(\int \frac{1}{8}(8a^3d^3(Ac - cC + Bd) + 3a^2bd^2(5c^2C - 8Bcd + (8A - 3C)d^2) + b^3c(15c^3C - 12Bc^2d + c(8A + 7C)d^2 - 4Bd^3) - 2ab^2d(15c^3C\right.}{}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)(a + b\tan(e + fx))^{5/2}}{d(c^2 + d^2) f\sqrt{c + d\tan(e + fx)}} - \frac{b(3(bc - ad)(5c^2C - 4Bcd + (4A + C)d^2) - 4d^2((A - C)(bc - ad) + B(ac + bd)))\sqrt{a + b\tan(}}{4d^3(c^2 + d^2) f} + \frac{b(5c^2C - 4Bcd + (4A + C)d^2)(a + b\tan(e + fx))^{3/2}\sqrt{c + d\tan(e + fx)}}{2d^2(c^2 + d^2) f} + \frac{\text{Subst}\left(\int \left(\frac{b(c^2 + d^2)(15a^2Cd^2 - 10abd(3cC - 2Bd) + b^2(15c^2C - 12Bcd + 8(A - C)d^2))}{8\sqrt{a + bx}\sqrt{c + dx}} + \frac{d^3(a^3(Ac - cC + Bd) - 3ab^2(Ac - cC + Ba}{}\right.\right.}{}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)(a + b\tan(e + fx))^{5/2}}{d(c^2 + d^2) f\sqrt{c + d\tan(e + fx)}} - \frac{b(3(bc - ad)(5c^2C - 4Bcd + (4A + C)d^2) - 4d^2((A - C)(bc - ad) + B(ac + bd)))\sqrt{a + b\tan(}}{4d^3(c^2 + d^2) f} + \frac{b(5c^2C - 4Bcd + (4A + C)d^2)(a + b\tan(e + fx))^{3/2}\sqrt{c + d\tan(e + fx)}}{2d^2(c^2 + d^2) f} + \frac{\text{Subst}\left(\int \frac{d^3(a^3(Ac - cC + Bd) - 3ab^2(Ac - cC + Bd) - 3a^2b(Bc - (A - C)d) + b^3(Bc - (A - C)d)) + d^3(3a^2b(Ac - cC + Bd) - b^3(Ac - cC +}{\sqrt{a + bx}\sqrt{c + dx}(1 + x^2)}\right.}{d^3(c^2 + d^2) f} + \frac{(b(15a^2Cd^2 - 10abd(3cC - 2Bd) + b^2(15c^2C - 12Bcd + 8(A - C)d^2)))\,\text{Subst}\left(\int \frac{1}{\sqrt{a + bx}\sqrt{c + dx}}\,dx,\right.}{8d^3 f}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)(a + b\tan(e + fx))^{5/2}}{d(c^2 + d^2) f\sqrt{c + d\tan(e + fx)}} - \frac{b(3(bc - ad)(5c^2C - 4Bcd + (4A + C)d^2) - 4d^2((A - C)(bc - ad) + B(ac + bd)))\sqrt{a + b\tan}}{4d^3(c^2 + d^2) f} + \frac{b(5c^2C - 4Bcd + (4A + C)d^2)(a + b\tan(e + fx))^{3/2}\sqrt{c + d\tan(e + fx)}}{2d^2(c^2 + d^2) f} + \frac{\text{Subst}\left(\int\left(\frac{-d^3(3a^2b(Ac-cC+Bd)-b^3(Ac-cC+Bd)+a^3(Bc-(A-C)d)-3ab^2(Bc-(A-C)d))+id^3(a^3(Ac-cC+Bd)-3ab^2(A}{2(i-x)\sqrt{a+bx}\sqrt{c+dx}}\right.\right.}{} + \frac{(15a^2Cd^2 - 10abd(3cC - 2Bd) + b^2(15c^2C - 12Bcd + 8(A - C)d^2))\,\text{Subst}\left(\int\frac{1}{\sqrt{c-\frac{ad}{b}+\frac{dx^2}{b}}}\,dx, x\right.}{4d^3 f}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)(a + b\tan(e + fx))^{5/2}}{d(c^2 + d^2) f\sqrt{c + d\tan(e + fx)}} - \frac{b(3(bc - ad)(5c^2C - 4Bcd + (4A + C)d^2) - 4d^2((A - C)(bc - ad) + B(ac + bd)))\sqrt{a + b\tan}}{4d^3(c^2 + d^2) f} + \frac{b(5c^2C - 4Bcd + (4A + C)d^2)(a + b\tan(e + fx))^{3/2}\sqrt{c + d\tan(e + fx)}}{2d^2(c^2 + d^2) f} + \frac{((a + ib)^3(A + iB - C)(ic + d))\,\text{Subst}\left(\int\frac{1}{(i-x)\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e + fx)\right)}{2(c^2 + d^2) f} + \frac{(15a^2Cd^2 - 10abd(3cC - 2Bd) + b^2(15c^2C - 12Bcd + 8(A - C)d^2))\,\text{Subst}\left(\int\frac{1}{1-\frac{dx^2}{b}}\,dx, x, \frac{\sqrt{a+}}{\sqrt{c+}}\right.}{4d^3 f} + \frac{(d^3(3a^2b(Ac - cC + Bd) - b^3(Ac - cC + Bd) + a^3(Bc - (A - C)d) - 3ab^2(Bc - (A - C)d))}{}$$

$$= \frac{\sqrt{b}(15a^2Cd^2 - 10abd(3cC - 2Bd) + b^2(15c^2C - 12Bcd + 8(A - C)d^2))\,\text{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right.}{4d^{7/2} f} - \frac{2(c^2C - Bcd + Ad^2)(a + b\tan(e + fx))^{5/2}}{d(c^2 + d^2) f\sqrt{c + d\tan(e + fx)}} - \frac{b(3(bc - ad)(5c^2C - 4Bcd + (4A + C)d^2) - 4d^2((A - C)(bc - ad) + B(ac + bd)))\sqrt{a + b\tan}}{4d^3(c^2 + d^2) f} + \frac{b(5c^2C - 4Bcd + (4A + C)d^2)(a + b\tan(e + fx))^{3/2}\sqrt{c + d\tan(e + fx)}}{2d^2(c^2 + d^2) f} + \frac{((a - ib)^3(iA + B - iC))\,\text{Subst}\left(\int\frac{1}{-a+ib-(-c+id)x^2}\,dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{(c - id) f} + \frac{((a + ib)^3(A + iB - C)(ic + d))\,\text{Subst}\left(\int\frac{1}{a+ib-(c+id)x^2}\,dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{(c^2 + d^2) f}$$

$$= -\frac{(a-ib)^{5/2}(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(c-id)^{3/2}f}$$
$$- \frac{(a+ib)^{5/2}(B-i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(c+id)^{3/2}f}$$
$$+ \frac{\sqrt{b}(15a^2Cd^2-10abd(3cC-2Bd)+b^2(15c^2C-12Bcd+8(A-C)d^2))\operatorname{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{4d^{7/2}f}$$
$$- \frac{2(c^2C-Bcd+Ad^2)(a+b\tan(e+fx))^{5/2}}{d(c^2+d^2)f\sqrt{c+d\tan(e+fx)}}$$
$$- \frac{b(3(bc-ad)(5c^2C-4Bcd+(4A+C)d^2)-4d^2((A-C)(bc-ad)+B(ac+bd)))\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{4d^3(c^2+d^2)f}$$
$$+ \frac{b(5c^2C-4Bcd+(4A+C)d^2)(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}{2d^2(c^2+d^2)f}$$

Mathematica [B] (verified)

Both result and optimal contain complex but leaf count is larger than twice the leaf count of optimal. 2245 vs. 2(528) = 1056.

Time = 9.61 (sec) , antiderivative size = 2245, normalized size of antiderivative = 4.25

$$\int \frac{(a+b\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx = \text{Result too large to show}$$

```
[In] Integrate[((a + b*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/(c + d*Tan[e + f*x])^(3/2),x]
```

```
[Out] (C*(a + b*Tan[e + f*x])^(5/2))/(2*d*f*Sqrt[c + d*Tan[e + f*x]]) + (((-5*b*c
*C + 4*b*B*d + 5*a*C*d)*(a + b*Tan[e + f*x])^(3/2))/(2*d*f*Sqrt[c + d*Tan[e
 + f*x]]) + ((8*(-a + I*b)^(5/2)*(I*A + B - I*C)*d^2*ArcTanh[(Sqrt[-c + I*d
]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[-a + I*b]*Sqrt[c + d*Tan[e + f*x]])]])/((-
c + I*d)^(3/2)*f) - (8*(a + I*b)^(5/2)*(B - I*(A - C))*d^2*ArcTanh[(Sqrt[c
+ I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])
/((c + I*d)^(3/2)*f) + (8*(a - I*b)^2*(I*A + B - I*C)*d^2*Sqrt[a + b*Tan[e
+ f*x]])/((c - I*d)*f*Sqrt[c + d*Tan[e + f*x]]) + (8*(a + I*b)^2*(B - I*(A
- C))*d^2*Sqrt[a + b*Tan[e + f*x]])/((c + I*d)*f*Sqrt[c + d*Tan[e + f*x]])
+ (30*a^2*C*(b*c - a*d)*(b/((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)))^(3/
2)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d))^2*Sqrt[(b*(c + d*Tan[e + f*x
]))/(b*c - a*d)]*(-1 - (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/(b*
c - a*d) - (a*b*d)/(b*c - a*d))))*(-((b*d*(a + b*Tan[e + f*x]))/((b*c - a*d
)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d))*(-1 - (b*d*(a + b*Tan[e + f*x
]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)))))) - (Sqrt[b]
*Sqrt[d]*ArcSinh[(Sqrt[b]*Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b*c - a*d
```

```
]*Sqrt[(b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)])]*Sqrt[a + b*Tan[e + f*x]
])/(Sqrt[b*c - a*d]*Sqrt[(b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)]*Sqrt[1
+ (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b
*c - a*d)))])))/(b^2*f*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]]*Sq
rt[1 + (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b*
d)/(b*c - a*d)))]) - (20*a*(b*c - a*d)*(3*c*C - 2*B*d)*(b/((b^2*c)/(b*c - a
*d) - (a*b*d)/(b*c - a*d)))^(3/2)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d
))^2*Sqrt[(b*(c + d*Tan[e + f*x]))/(b*c - a*d)]*(-1 - (b*d*(a + b*Tan[e + f
*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d))))*(-((b*d*(a
 + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)
)*(-1 - (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b
*d)/(b*c - a*d)))))) - (Sqrt[b]*Sqrt[d]*ArcSinh[(Sqrt[b]*Sqrt[d]*Sqrt[a + b
*Tan[e + f*x]])/(Sqrt[b*c - a*d]*Sqrt[(b^2*c)/(b*c - a*d) - (a*b*d)/(b*c -
a*d)])]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b*c - a*d]*Sqrt[(b^2*c)/(b*c - a*d)
 - (a*b*d)/(b*c - a*d)]*Sqrt[1 + (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((
b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)))])))/(b*d*f*Sqrt[a + b*Tan[e + f*
x]]*Sqrt[c + d*Tan[e + f*x]]*Sqrt[1 + (b*d*(a + b*Tan[e + f*x]))/((b*c - a*
d)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)))]) + (2*(b*c - a*d)*(15*c^2*
C - 12*B*c*d + 8*(A - C)*d^2)*(b/((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)
))^(3/2)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d))^2*Sqrt[(b*(c + d*Tan[e
 + f*x]))/(b*c - a*d)]*(-1 - (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*
c)/(b*c - a*d) - (a*b*d)/(b*c - a*d))))*(-((b*d*(a + b*Tan[e + f*x]))/((b*c
 - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d))*(-1 - (b*d*(a + b*Tan[e
 + f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)))))) - (S
qrt[b]*Sqrt[d]*ArcSinh[(Sqrt[b]*Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b*c
 - a*d]*Sqrt[(b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)])]*Sqrt[a + b*Tan[e
+ f*x]])/(Sqrt[b*c - a*d]*Sqrt[(b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)]*S
qrt[1 + (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b
*d)/(b*c - a*d)))])))/(d^2*f*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*
x]]*Sqrt[1 + (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) -
 (a*b*d)/(b*c - a*d)))]))/(4*d))/(2*d)
```

Maple [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(fx+e))^{\frac{5}{2}}\left(A+B\tan(fx+e)+C\tan(fx+e)^2\right)}{(c+d\tan(fx+e))^{\frac{3}{2}}}dx$$

```
[In] int((a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(3/2),x)
```

```
[Out] int((a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(3/2),x)
```

Fricas [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(3/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int \frac{(a+b\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx = \int \frac{(a+b\tan(e+fx))^{\frac{5}{2}}(A+B\tan(e$$

```
[In] integrate((a+b*tan(f*x+e))**(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan
(f*x+e))**(3/2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)**
2)/(c + d*tan(e + f*x))**(3/2), x)
```

Maxima [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(3/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

Giac [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(3/2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^{5/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx = \int \frac{(a+b\tan(e+f\,x))^{5/2}\,\left(C\tan(e+\right.}{(c+d\tan\left(e+\right.}$$

```
[In] int(((a + b*tan(e + f*x))^(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c
 + d*tan(e + f*x))^(3/2),x)
```

```
[Out] int(((a + b*tan(e + f*x))^(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c
 + d*tan(e + f*x))^(3/2), x)
```

3.154 $\int \frac{(a+b\tan(e+fx))^{3/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx$

Optimal result 2078
Rubi [A] (verified) 2079
Mathematica [B] (verified) 2083
Maple [F(-1)] 2085
Fricas [F(-1)] 2085
Sympy [F] 2085
Maxima [F(-1)] 2085
Giac [F(-1)] 2086
Mupad [F(-1)] 2086

Optimal result

Integrand size = 49, antiderivative size = 380

$$\int \frac{(a+b\tan(e+fx))^{3/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx = -\frac{(a-ib)^{3/2}(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(c-id)^{3/2}f} - \frac{(a+ib)^{3/2}(B-i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(c+id)^{3/2}f} - \frac{\sqrt{b}(3bcC-2bBd-3aCd)\operatorname{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{d^{5/2}f} - \frac{2(c^2C-Bcd+Ad^2)(a+b\tan(e+fx))^{3/2}}{d(c^2+d^2)f\sqrt{c+d\tan(e+fx)}} + \frac{b(3c^2C-2Bcd+(2A+C)d^2)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{d^2(c^2+d^2)f}$$

```
[Out] -(a-I*b)^(3/2)*(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-
I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/(c-I*d)^(3/2)/f-(a+I*b)^(3/2)*(B-I*(A-C)
)*arctanh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e
))^(1/2))/(c+I*d)^(3/2)/f-(-2*B*b*d-3*C*a*d+3*C*b*c)*arctanh(d^(1/2)*(a+b*t
an(f*x+e))^(1/2)/b^(1/2)/(c+d*tan(f*x+e))^(1/2))*b^(1/2)/d^(5/2)/f+b*(3*c^2
*C-2*B*c*d+(2*A+C)*d^2)*(a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)/d^2/(
c^2+d^2)/f-2*(A*d^2-B*c*d+C*c^2)*(a+b*tan(f*x+e))^(3/2)/d/(c^2+d^2)/f/(c+d*
tan(f*x+e))^(1/2)
```

Rubi [A] (verified)

Time = 6.51 (sec) , antiderivative size = 380, normalized size of antiderivative = 1.00, number of steps used = 14, number of rules used = 9, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.184, Rules used = {3726, 3728, 3736, 6857, 65, 223, 212, 95, 214}

$$\int \frac{(a+b\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{3/2}}\,dx = -\frac{(a-ib)^{3/2}(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(c-id)^{3/2}} - \frac{(a+ib)^{3/2}(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(c+id)^{3/2}} + \frac{b(d^2(2A+C)-2Bcd+3c^2C)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{d^2f\,(c^2+d^2)} - \frac{2(Ad^2-Bcd+c^2C)\,(a+b\tan(e+fx))^{3/2}}{df\,(c^2+d^2)\sqrt{c+d\tan(e+fx)}} - \frac{\sqrt{b}(-3aCd-2bBd+3bcC)\text{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{d^{5/2}f}$$

```
[In] Int[((a + b*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c
 + d*Tan[e + f*x])^(3/2),x]
```

```
[Out] -(((a - I*b)^(3/2)*(I*A + B - I*C)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e
+ f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((c - I*d)^(3/2)*f)) -
((a + I*b)^(3/2)*(B - I*(A - C))*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/((c + I*d)^(3/2)*f) - (Sq
rt[b]*(3*b*c*C - 2*b*B*d - 3*a*C*d)*ArcTanh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x
]])/(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])])/(d^(5/2)*f) - (2*(c^2*C - B*c*d +
A*d^2)*(a + b*Tan[e + f*x])^(3/2))/(d*(c^2 + d^2)*f*Sqrt[c + d*Tan[e + f*x]
]) + (b*(3*c^2*C - 2*B*c*d + (2*A + C)*d^2)*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c
 + d*Tan[e + f*x]])/(d^2*(c^2 + d^2)*f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 95

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
```

```
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 223

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3726

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*d^2 + c*(c*C - B*d))*(a + b*Tan[e +
 f*x])^m*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3736

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[(a + b*ff*x)^m*(c + d*ff*x)^n*((A + B*ff*x + C*ff^2*x^2
)/(1 + ff^2*x^2)), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6857

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\text{integral} = -\frac{2(c^2C - Bcd + Ad^2)\,(a + b\tan(e + fx))^{3/2}}{d\,(c^2 + d^2)\,f\sqrt{c + d\tan(e + fx)}} + \frac{2\int \frac{\sqrt{a+b\tan(e+fx)}\left(\frac{1}{2}(Ad(ac+3bd)+(3bc-ad)(cC-Bd))+\frac{1}{2}d((A-C)(bc-ad)+B(ac+bd))\tan(e+fx)+\frac{1}{2}b\left(3c^2C-2Bcd+(2A+C)d^2\right)\tan^{2}\right.}{\sqrt{c+d\tan(e+fx)}}}{d\,(c^2 + d^2)}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)\,(a + b\tan(e + fx))^{3/2}}{d\,(c^2 + d^2)\,f\sqrt{c + d\tan(e + fx)}} + \frac{b(3c^2C - 2Bcd + (2A + C)d^2)\,\sqrt{a + b\tan(e + fx)}\sqrt{c + d\tan(e + fx)}}{d^2\,(c^2 + d^2)\,f} + \frac{2\int \frac{\frac{1}{4}\left(-b(bc+ad)\left(3c^2C-2Bcd+(2A+C)d^2\right)+2ad(Ad(ac+3bd)+(3bc-ad)(cC-Bd))\right)+\frac{1}{2}d^2\left(2ab(Ac-cC+Bd)+a^2(Bc-(A-C)d\right.}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}}{d^2\,(c^2 + d^2)}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)\,(a + b\tan(e + fx))^{3/2}}{d\,(c^2 + d^2)\,f\sqrt{c + d\tan(e + fx)}} + \frac{b(3c^2C - 2Bcd + (2A + C)d^2)\,\sqrt{a + b\tan(e + fx)}\sqrt{c + d\tan(e + fx)}}{d^2\,(c^2 + d^2)\,f} + \frac{2\text{Subst}\left(\int \frac{\frac{1}{4}\left(-b(bc+ad)\left(3c^2C-2Bcd+(2A+C)d^2\right)+2ad(Ad(ac+3bd)+(3bc-ad)(cC-Bd))\right)+\frac{1}{2}d^2\left(2ab(Ac-cC+Bd)+a^2(Bc-\right.}{\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}\right.}{d^2\,(c^2 + d^2)\,f}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)(a + b\tan(e + fx))^{3/2}}{d(c^2 + d^2) f\sqrt{c + d\tan(e + fx)}} + \frac{b(3c^2C - 2Bcd + (2A + C)d^2)\sqrt{a + b\tan(e + fx)}\sqrt{c + d\tan(e + fx)}}{d^2(c^2 + d^2) f} + \frac{2\text{Subst}\left(\int \left(-\frac{b(3bcC-2bBd-3aCd)(c^2+d^2)}{4\sqrt{a+bx}\sqrt{c+dx}} + \frac{d^2(a^2(Ac-cC+Bd)-b^2(Ac-cC+Bd)-2ab(Bc-(A-C)d))+d^2(2ab(Ac-cC+}{2\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}\right.\right.}{d^2(c^2 + d^2) f}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)(a + b\tan(e + fx))^{3/2}}{d(c^2 + d^2) f\sqrt{c + d\tan(e + fx)}} + \frac{b(3c^2C - 2Bcd + (2A + C)d^2)\sqrt{a + b\tan(e + fx)}\sqrt{c + d\tan(e + fx)}}{d^2(c^2 + d^2) f} - \frac{(b(3bcC - 2bBd - 3aCd))\text{Subst}\left(\int \frac{1}{\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e + fx)\right)}{2d^2 f} + \frac{\text{Subst}\left(\int \frac{d^2(a^2(Ac-cC+Bd)-b^2(Ac-cC+Bd)-2ab(Bc-(A-C)d))+d^2(2ab(Ac-cC+Bd)+a^2(Bc-(A-C)d)-b^2(Bc-(A-C)d}{\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}\right.}{d^2(c^2 + d^2) f}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)(a + b\tan(e + fx))^{3/2}}{d(c^2 + d^2) f\sqrt{c + d\tan(e + fx)}} + \frac{b(3c^2C - 2Bcd + (2A + C)d^2)\sqrt{a + b\tan(e + fx)}\sqrt{c + d\tan(e + fx)}}{d^2(c^2 + d^2) f} - \frac{(3bcC - 2bBd - 3aCd)\text{Subst}\left(\int \frac{1}{\sqrt{c - \frac{ad}{b} + \frac{dx^2}{b}}}\,dx, x, \sqrt{a + b\tan(e + fx)}\right)}{d^2 f} + \frac{\text{Subst}\left(\int \left(\frac{id^2(a^2(Ac-cC+Bd)-b^2(Ac-cC+Bd)-2ab(Bc-(A-C)d))-d^2(2ab(Ac-cC+Bd)+a^2(Bc-(A-C)d)-b^2(Bc-(A-C}{2(i-x)\sqrt{a+bx}\sqrt{c+dx}}\right.\right.}{}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)(a + b\tan(e + fx))^{3/2}}{d(c^2 + d^2) f\sqrt{c + d\tan(e + fx)}} + \frac{b(3c^2C - 2Bcd + (2A + C)d^2)\sqrt{a + b\tan(e + fx)}\sqrt{c + d\tan(e + fx)}}{d^2(c^2 + d^2) f} + \frac{((a - ib)^2(iA + B - iC))\,\text{Subst}\left(\int \frac{1}{(i+x)\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e + fx)\right)}{2(c - id) f} - \frac{(3bcC - 2bBd - 3aCd)\text{Subst}\left(\int \frac{1}{1 - \frac{dx^2}{b}}\,dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{d^2 f} + \frac{((a + ib)^2(A + iB - C)(ic + d))\,\text{Subst}\left(\int \frac{1}{(i-x)\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e + fx)\right)}{2(c^2 + d^2) f}$$

$$= -\frac{\sqrt{b}(3bcC - 2bBd - 3aCd)\operatorname{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{d^{5/2}f} - \frac{2(c^2C - Bcd + Ad^2)(a+b\tan(e+fx))^{3/2}}{d(c^2+d^2)f\sqrt{c+d\tan(e+fx)}} + \frac{b(3c^2C - 2Bcd + (2A+C)d^2)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{d^2(c^2+d^2)f} + \frac{((a-ib)^2(iA+B-iC))\operatorname{Subst}\left(\int \frac{1}{-a+ib-(-c+id)x^2}dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{(c-id)f} + \frac{((a+ib)^2(A+iB-C)(ic+d))\operatorname{Subst}\left(\int \frac{1}{a+ib-(c+id)x^2}dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{(c^2+d^2)f}$$

$$= -\frac{(a-ib)^{3/2}(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(c-id)^{3/2}f} - \frac{(a+ib)^{3/2}(B-i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(c+id)^{3/2}f} - \frac{\sqrt{b}(3bcC - 2bBd - 3aCd)\operatorname{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{d^{5/2}f} - \frac{2(c^2C - Bcd + Ad^2)(a+b\tan(e+fx))^{3/2}}{d(c^2+d^2)f\sqrt{c+d\tan(e+fx)}} + \frac{b(3c^2C - 2Bcd + (2A+C)d^2)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{d^2(c^2+d^2)f}$$

Mathematica [B] (verified)

Both result and optimal contain complex but leaf count is larger than twice the leaf count of optimal. 2141 vs. 2(380) = 760.

Time = 7.68 (sec) , antiderivative size = 2141, normalized size of antiderivative = 5.63

$$\int \frac{(a+b\tan(e+fx))^{3/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}dx = \text{Result too large to show}$$

```
[In] Integrate[((a + b*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/(c + d*Tan[e + f*x])^(3/2),x]
```

```
[Out] (C*(a + b*Tan[e + f*x])^(3/2))/(d*f*Sqrt[c + d*Tan[e + f*x]]) + ((-2*(-a +
I*b)^(3/2)*(B + I*(A - C))*d*ArcTanh[(Sqrt[-c + I*d]*Sqrt[a + b*Tan[e + f*x
]])/(Sqrt[-a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/((-c + I*d)^(3/2)*f) - (2*(
a + I*b)^(3/2)*(B - I*(A - C))*d*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/((c + I*d)^(3/2)*f) + (2*
```

```
(I*a + b)*(A - I*B - C)*d*Sqrt[a + b*Tan[e + f*x]])/((c - I*d)*f*Sqrt[c + d
*Tan[e + f*x]]) - (2*(I*a - b)*(A + I*B - C)*d*Sqrt[a + b*Tan[e + f*x]])/((
c + I*d)*f*Sqrt[c + d*Tan[e + f*x]]) - (6*c*C*(b*c - a*d)*(b/((b^2*c)/(b*c
 - a*d) - (a*b*d)/(b*c - a*d)))^(3/2)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c -
a*d))^2*Sqrt[(b*(c + d*Tan[e + f*x]))/(b*c - a*d)]*(-1 - (b*d*(a + b*Tan[e
+ f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d))))*(-((b*d
*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a
*d))*(-1 - (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (
a*b*d)/(b*c - a*d)))))) - (Sqrt[b]*Sqrt[d]*ArcSinh[(Sqrt[b]*Sqrt[d]*Sqrt[a
+ b*Tan[e + f*x]])/(Sqrt[b*c - a*d]*Sqrt[(b^2*c)/(b*c - a*d) - (a*b*d)/(b*c
 - a*d)])]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b*c - a*d]*Sqrt[(b^2*c)/(b*c - a
*d) - (a*b*d)/(b*c - a*d)]*Sqrt[1 + (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)
*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)))]))))/(b*d^2*f*Sqrt[a + b*Tan[e
 + f*x]]*Sqrt[c + d*Tan[e + f*x]]*Sqrt[1 + (b*d*(a + b*Tan[e + f*x]))/((b*c
 - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)))]) + (4*B*(b*c - a*d)*(
b/((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)))^(3/2)*((b^2*c)/(b*c - a*d) -
 (a*b*d)/(b*c - a*d))^2*Sqrt[(b*(c + d*Tan[e + f*x]))/(b*c - a*d)]*(-1 - (b
*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c -
 a*d))))*(-((b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) -
(a*b*d)/(b*c - a*d))*(-1 - (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)
/(b*c - a*d) - (a*b*d)/(b*c - a*d)))))) - (Sqrt[b]*Sqrt[d]*ArcSinh[(Sqrt[b]
*Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b*c - a*d]*Sqrt[(b^2*c)/(b*c - a*d
) - (a*b*d)/(b*c - a*d)])]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b*c - a*d]*Sqrt[
(b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)]*Sqrt[1 + (b*d*(a + b*Tan[e + f*x
]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)))]))))/(b*d*f*Sq
rt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]]*Sqrt[1 + (b*d*(a + b*Tan[e
+ f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)))]) + (6*a
*C*(b*c - a*d)*(b/((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)))^(3/2)*((b^2*
c)/(b*c - a*d) - (a*b*d)/(b*c - a*d))^2*Sqrt[(b*(c + d*Tan[e + f*x]))/(b*c
- a*d)]*(-1 - (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d)
- (a*b*d)/(b*c - a*d))))*(-((b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c
)/(b*c - a*d) - (a*b*d)/(b*c - a*d))*(-1 - (b*d*(a + b*Tan[e + f*x]))/((b*c
 - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)))))) - (Sqrt[b]*Sqrt[d]*
ArcSinh[(Sqrt[b]*Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b*c - a*d]*Sqrt[(b
^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)])]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[
b*c - a*d]*Sqrt[(b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)]*Sqrt[1 + (b*d*(a
 + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)
))]))))/(b^2*d*f*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]]*Sqrt[1 +
(b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c
 - a*d)))]))/(2*d)
```

Maple [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(fx+e))^{\frac{3}{2}}\left(A+B\tan(fx+e)+C\tan(fx+e)^{2}\right)}{(c+d\tan(fx+e))^{\frac{3}{2}}}\,dx$$

```
[In] int((a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(3/2),x)
```

```
[Out] int((a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(3/2),x)
```

Fricas [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(3/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int \frac{(a+b\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{3/2}}\,dx = \int \frac{(a+b\tan(e+fx))^{\frac{3}{2}}\left(A+B\tan(\right.}{(c+d\tan(e+}$$

```
[In] integrate((a+b*tan(f*x+e))**(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan
(f*x+e))**(3/2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)**
2)/(c + d*tan(e + f*x))**(3/2), x)
```

Maxima [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(3/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

Giac [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(3/2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{3/2}}\,dx = \int \frac{(a+b\tan(e+f\,x))^{3/2}\,\left(C\tan(e+f\right.}{(c+d\tan\left(e+\right.}$$

```
[In] int(((a + b*tan(e + f*x))^(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c
 + d*tan(e + f*x))^(3/2),x)
```

```
[Out] int(((a + b*tan(e + f*x))^(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c
 + d*tan(e + f*x))^(3/2), x)
```

3.155 $\int \frac{\sqrt{a+b\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx$

Optimal result . . . 2087
Rubi [A] (verified) . . . 2088
Mathematica [A] (verified) . . . 2091
Maple [F(-1)] . . . 2092
Fricas [B] (verification not implemented) . . . 2092
Sympy [F] . . . 2092
Maxima [F(-1)] . . . 2093
Giac [F(-1)] . . . 2093
Mupad [F(-1)] . . . 2093

Optimal result

Integrand size = 49, antiderivative size = 299

$$\int \frac{\sqrt{a+b\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx = -\frac{\sqrt{a-ib}(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(c-id)^{3/2}f} - \frac{\sqrt{a+ib}(B-i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(c+id)^{3/2}f} + \frac{2\sqrt{b}C\operatorname{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{d^{3/2}f} - \frac{2(c^2C-Bcd+Ad^2)\sqrt{a+b\tan(e+fx)}}{d(c^2+d^2)f\sqrt{c+d\tan(e+fx)}}$$

```
[Out] -(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(1/2)/(c+
d*tan(f*x+e))^(1/2))*(a-I*b)^(1/2)/(c-I*d)^(3/2)/f-(B-I*(A-C))*arctanh((c+I
*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))*(a+I
*b)^(1/2)/(c+I*d)^(3/2)/f+2*C*arctanh(d^(1/2)*(a+b*tan(f*x+e))^(1/2)/b^(1/2
)/(c+d*tan(f*x+e))^(1/2))*b^(1/2)/d^(3/2)/f-2*(A*d^2-B*c*d+C*c^2)*(a+b*tan(
f*x+e))^(1/2)/d/(c^2+d^2)/f/(c+d*tan(f*x+e))^(1/2)
```

Rubi [A] (verified)

Time = 3.84 (sec) , antiderivative size = 299, normalized size of antiderivative = 1.00, number of steps used = 13, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.163, Rules used = {3726, 3736, 6857, 65, 223, 212, 95, 214}

$$\int \frac{\sqrt{a+b\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx = -\frac{\sqrt{a-ib}(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(c-id)^{3/2}} - \frac{\sqrt{a+ib}(B-i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(c+id)^{3/2}} - \frac{2(Ad^2-Bcd+c^2C)\sqrt{a+b\tan(e+fx)}}{df(c^2+d^2)\sqrt{c+d\tan(e+fx)}} + \frac{2\sqrt{b}C\operatorname{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{d^{3/2}f}$$

```
[In] Int[(Sqrt[a + b*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c +
 d*Tan[e + f*x])^(3/2),x]
```

```
[Out] -((Sqrt[a - I*b]*(I*A + B - I*C)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((c - I*d)^(3/2)*f)) - (S
qrt[a + I*b]*(B - I*(A - C))*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]
])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/((c + I*d)^(3/2)*f) + (2*Sqrt
[b]*C*ArcTanh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c + d*Tan[e
+ f*x]])])/(d^(3/2)*f) - (2*(c^2*C - B*c*d + A*d^2)*Sqrt[a + b*Tan[e + f*x]
])/(d*(c^2 + d^2)*f*Sqrt[c + d*Tan[e + f*x]])
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 95

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 223

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3726

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*d^2 + c*(c*C - B*d))*(a + b*Tan[e +
 f*x])^m*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3736

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[(a + b*ff*x)^m*(c + d*ff*x)^n*((A + B*ff*x + C*ff^2*x^2
)/(1 + ff^2*x^2)), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6857

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

integral

$$=-\frac{2(c^2C-Bcd+Ad^2)\sqrt{a+b\tan(e+fx)}}{d(c^2+d^2)f\sqrt{c+d\tan(e+fx)}}+\frac{2\int\frac{\frac{1}{2}(Ad(ac+bd)+(bc-ad)(cC-Bd))+\frac{1}{2}d((A-C)(bc-ad)+B(ac+bd))\tan(e+fx)+\frac{1}{2}bC(c^2+d^2)\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx}{d(c^2+d^2)}$$

$$=-\frac{2(c^2C-Bcd+Ad^2)\sqrt{a+b\tan(e+fx)}}{d(c^2+d^2)f\sqrt{c+d\tan(e+fx)}}+\frac{2\text{Subst}\left(\int\frac{\frac{1}{2}(Ad(ac+bd)+(bc-ad)(cC-Bd))+\frac{1}{2}d((A-C)(bc-ad)+B(ac+bd))x+\frac{1}{2}bC(c^2+d^2)x^2}{\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}\,dx,x,\tan(e+fx)\right)}{d(c^2+d^2)f}$$

$$=-\frac{2(c^2C-Bcd+Ad^2)\sqrt{a+b\tan(e+fx)}}{d(c^2+d^2)f\sqrt{c+d\tan(e+fx)}}+\frac{2\text{Subst}\left(\int\left(\frac{bC(c^2+d^2)}{2\sqrt{a+bx}\sqrt{c+dx}}+\frac{d(a(Ac-cC+Bd)-b(Bc-(A-C)d))+d(Abc+aBc-bcC-aAd+bBd+aCd)x}{2\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}\right)dx,x,\tan(e+fx)\right)}{d(c^2+d^2)f}$$

$$=-\frac{2(c^2C-Bcd+Ad^2)\sqrt{a+b\tan(e+fx)}}{d(c^2+d^2)f\sqrt{c+d\tan(e+fx)}}+\frac{(bC)\text{Subst}\left(\int\frac{1}{\sqrt{a+bx}\sqrt{c+dx}}\,dx,x,\tan(e+fx)\right)}{df}+\frac{\text{Subst}\left(\int\frac{d(a(Ac-cC+Bd)-b(Bc-(A-C)d))+d(Abc+aBc-bcC-aAd+bBd+aCd)x}{\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}\,dx,x,\tan(e+fx)\right)}{d(c^2+d^2)f}$$

$$=-\frac{2(c^2C-Bcd+Ad^2)\sqrt{a+b\tan(e+fx)}}{d(c^2+d^2)f\sqrt{c+d\tan(e+fx)}}+\frac{(2C)\text{Subst}\left(\int\frac{1}{\sqrt{c-\frac{ad}{b}+\frac{dx^2}{b}}}\,dx,x,\sqrt{a+b\tan(e+fx)}\right)}{df}+\frac{\text{Subst}\left(\int\left(\frac{-d(Abc+aBc-bcC-aAd+bBd+aCd)+id(a(Ac-cC+Bd)-b(Bc-(A-C)d))}{2(i-x)\sqrt{a+bx}\sqrt{c+dx}}+\frac{d(Abc+aBc-bcC-aAd+bBd+aCd)+id(a(Ac}{2(i+x)\sqrt{a+bx}\sqrt{c+}}\right.\right.}{d(c^2+d^2)f}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)\sqrt{a + b\tan(e + fx)}}{d(c^2 + d^2)f\sqrt{c + d\tan(e + fx)}} + \frac{((ia + b)(A - iB - C))\text{Subst}\left(\int \frac{1}{(i+x)\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e + fx)\right)}{2(c - id)f} + \frac{((ia - b)(A + iB - C))\text{Subst}\left(\int \frac{1}{(i-x)\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e + fx)\right)}{2(c + id)f} + \frac{(2C)\text{Subst}\left(\int \frac{1}{1-\frac{dx^2}{b}}\,dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{df}$$

$$= \frac{2\sqrt{b}C\text{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{d^{3/2}f} - \frac{2(c^2C - Bcd + Ad^2)\sqrt{a + b\tan(e + fx)}}{d(c^2 + d^2)f\sqrt{c + d\tan(e + fx)}} + \frac{((ia + b)(A - iB - C))\text{Subst}\left(\int \frac{1}{-a+ib-(-c+id)x^2}\,dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{(c - id)f} + \frac{((ia - b)(A + iB - C))\text{Subst}\left(\int \frac{1}{a+ib-(c+id)x^2}\,dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{(c + id)f}$$

$$= -\frac{\sqrt{a - ib}(iA + B - iC)\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(c - id)^{3/2}f} - \frac{\sqrt{a + ib}(B - i(A - C))\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(c + id)^{3/2}f} + \frac{2\sqrt{b}C\text{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{d^{3/2}f} - \frac{2(c^2C - Bcd + Ad^2)\sqrt{a + b\tan(e + fx)}}{d(c^2 + d^2)f\sqrt{c + d\tan(e + fx)}}$$

Mathematica [A] (verified)

Time = 6.11 (sec) , antiderivative size = 403, normalized size of antiderivative = 1.35

$$\int \frac{\sqrt{a + b\tan(e + fx)}(A + B\tan(e + fx) + C\tan^2(e + fx))}{(c + d\tan(e + fx))^{3/2}}\,dx = \frac{\sqrt{-a+ib}(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{-c+id}\sqrt{a+b\tan(e+}}{\sqrt{-a+ib}\sqrt{c+d\tan(e+}}\right.}{(-c+id)^{3/2}}$$

```
[In] Integrate[(Sqrt[a + b*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)
)/(c + d*Tan[e + f*x])^(3/2),x]
```

```
[Out] ((Sqrt[-a + I*b]*(I*A + B - I*C)*ArcTanh[(Sqrt[-c + I*d]*Sqrt[a + b*Tan[e +
 f*x]])/(Sqrt[-a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(-c + I*d)^(3/2) + (I*S
qrt[a + I*b]*(A + I*B - C)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]])
/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(c + I*d)^(3/2) + ((B + I*(A -
```

```
C))*Sqrt[a + b*Tan[e + f*x]])/((c - I*d)*Sqrt[c + d*Tan[e + f*x]]) + (((-I)
*A + B + I*C)*Sqrt[a + b*Tan[e + f*x]])/((c + I*d)*Sqrt[c + d*Tan[e + f*x]]
) + (2*C*(-(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]]) + Sqrt[b*c - a*d]*ArcSinh[(Sq
rt[d]*Sqrt[a + b*Tan[e + f*x]])/Sqrt[b*c - a*d]]*Sqrt[(b*(c + d*Tan[e + f*x
]))/(b*c - a*d)]))/(d^(3/2)*Sqrt[c + d*Tan[e + f*x]]))/f
```

Maple [F(-1)]

Timed out.

$$\int \frac{\sqrt{a+b\tan\left(fx+e\right)}\,\left(A+B\tan\left(fx+e\right)+C\tan\left(fx+e\right)^{2}\right)}{\left(c+d\tan\left(fx+e\right)\right)^{\frac{3}{2}}}dx$$

```
[In] int((a+b*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(3/2),x)
```

```
[Out] int((a+b*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(3/2),x)
```

Fricas [B] (verification not implemented)

Leaf count of result is larger than twice the leaf count of optimal. 69574 vs. 2(237) = 474.

Time = 246.04 (sec) , antiderivative size = 139175, normalized size of antiderivative = 465.47

$$\int \frac{\sqrt{a+b\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx = \text{Too large to display}$$

```
[In] integrate((a+b*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(3/2),x, algorithm="fricas")
```

```
[Out] Too large to include
```

Sympy [F]

$$\int \frac{\sqrt{a+b\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx = \int \frac{\sqrt{a+b\tan\left(e+fx\right)}\left(A+B\tan\left(e+f\right.\right.}{\left(c+d\tan\left(e+fx\right)\right.}$$

```
[In] integrate((a+b*tan(f*x+e))**(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan
(f*x+e))**(3/2),x)
```

```
[Out] Integral(sqrt(a + b*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/
(c + d*tan(e + f*x))**(3/2), x)
```

Maxima [F(-1)]

Timed out.

$$\int \frac{\sqrt{a+b\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(3/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

Giac [F(-1)]

Timed out.

$$\int \frac{\sqrt{a+b\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(3/2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{\sqrt{a+b\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{3/2}}\,dx = \int \frac{\sqrt{a+b\tan\left(e+f\,x\right)}\,\left(C\tan(e+f\,x\right.}{\left(c+d\tan\left(e+f\right.\right.}$$

```
[In] int(((a + b*tan(e + f*x))^(1/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c
 + d*tan(e + f*x))^(3/2),x)
```

```
[Out] int(((a + b*tan(e + f*x))^(1/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c
 + d*tan(e + f*x))^(3/2), x)
```

3.156 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}\,dx$

Optimal result 2094
Rubi [A] (verified) 2094
Mathematica [A] (verified) 2097
Maple [F(-1)] 2098
Fricas [F(-1)] 2098
Sympy [F] 2098
Maxima [F] 2098
Giac [F(-1)] 2099
Mupad [F(-1)] 2099

Optimal result

Integrand size = 49, antiderivative size = 251

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}\,dx = -\frac{(B+i(A-C))\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{a-ib}(c-id)^{3/2}f} + \frac{(iA-B-iC)\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{a+ib}(c+id)^{3/2}f} + \frac{2(c^2C-Bcd+Ad^2)\sqrt{a+b\tan(e+fx)}}{(bc-ad)(c^2+d^2)f\sqrt{c+d\tan(e+fx)}}$$

```
[Out] -(B+I*(A-C))*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(1/2)/(c+
d*tan(f*x+e))^(1/2))/(c-I*d)^(3/2)/f/(a-I*b)^(1/2)+(I*A-B-I*C)*arctanh((c+I
*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/(c+I
*d)^(3/2)/f/(a+I*b)^(1/2)+2*(A*d^2-B*c*d+C*c^2)*(a+b*tan(f*x+e))^(1/2)/(-a*
d+b*c)/(c^2+d^2)/f/(c+d*tan(f*x+e))^(1/2)
```

Rubi [A] (verified)

Time = 1.15 (sec) , antiderivative size = 251, normalized size of antiderivative = 1.00, number of steps used = 8, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.102, Rules used

$= \{3730, 3697, 3696, 95, 214\}$

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}\,dx = -\frac{(B+i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f\sqrt{a-ib}(c-id)^{3/2}} + \frac{(iA-B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f\sqrt{a+ib}(c+id)^{3/2}} + \frac{2(Ad^2-Bcd+c^2C)\sqrt{a+b\tan(e+fx)}}{f(c^2+d^2)(bc-ad)\sqrt{c+d\tan(e+fx)}}$$

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/(Sqrt[a + b*Tan[e + f*x]]*(c +
d*Tan[e + f*x])^(3/2)),x]
```

```
[Out] -(((B + I*(A - C))*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a
 - I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[a - I*b]*(c - I*d)^(3/2)*f)) + ((
I*A - B - I*C)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + I
*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[a + I*b]*(c + I*d)^(3/2)*f) + (2*(c^2
*C - B*c*d + A*d^2)*Sqrt[a + b*Tan[e + f*x]])/((b*c - a*d)*(c^2 + d^2)*f*Sq
rt[c + d*Tan[e + f*x]])
```

Rule 95

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3696

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[A^2/f, Subst[Int[(a + b*x)^m*((c + d*x)^n/(A - B*x)), x], x, Tan[e + f*x
]], x] /; FreeQ[{a, b, c, d, e, f, A, B, m, n}, x] && NeQ[b*c - a*d, 0] &&
NeQ[a^2 + b^2, 0] && EqQ[A^2 + B^2, 0]
```

Rule 3697

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[(A + I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^n*(1 - I*Ta
n[e + f*x]), x], x] + Dist[(A - I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*T
an[e + f*x])^n*(1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A,
B, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[A^2 + B^2, 0]
```

Rule 3730

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 - a*(b*B - a*C))*(a + b*Tan[e +
 f*x])^(m + 1)*((c + d*Tan[e + f*x])^(n + 1)/(f*(m + 1)*(b*c - a*d)*(a^2 +
b^2))), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rubi steps

$$\text{integral} = \frac{2(c^2C - Bcd + Ad^2)\sqrt{a+b\tan(e+fx)}}{(bc-ad)(c^2+d^2)f\sqrt{c+d\tan(e+fx)}} + \frac{2\int \frac{\frac{1}{2}(bc-ad)(Ac-cC+Bd)+\frac{1}{2}(bc-ad)(Bc-(A-C)d)\tan(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx}{(bc-ad)(c^2+d^2)}$$

$$= \frac{2(c^2C - Bcd + Ad^2)\sqrt{a+b\tan(e+fx)}}{(bc-ad)(c^2+d^2)f\sqrt{c+d\tan(e+fx)}} + \frac{(A-iB-C)\int \frac{1+i\tan(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx}{2(c-id)} + \frac{(A+iB-C)\int \frac{1-i\tan(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx}{2(c+id)}$$

$$= \frac{2(c^2C - Bcd + Ad^2)\sqrt{a+b\tan(e+fx)}}{(bc-ad)(c^2+d^2)f\sqrt{c+d\tan(e+fx)}} + \frac{(A-iB-C)\text{Subst}\left(\int \frac{1}{(1-ix)\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e+fx)\right)}{2(c-id)f} + \frac{(A+iB-C)\text{Subst}\left(\int \frac{1}{(1+ix)\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e+fx)\right)}{2(c+id)f}$$

$$= \frac{2(c^2C - Bcd + Ad^2)\sqrt{a + b\tan(e + fx)}}{(bc - ad)(c^2 + d^2) f\sqrt{c + d\tan(e + fx)}} + \frac{(A - iB - C)\text{Subst}\left(\int \frac{1}{ia + b - (ic + d)x^2} dx, x, \frac{\sqrt{a + b\tan(e + fx)}}{\sqrt{c + d\tan(e + fx)}}\right)}{(c - id)f} + \frac{(A + iB - C)\text{Subst}\left(\int \frac{1}{-ia + b - (-ic + d)x^2} dx, x, \frac{\sqrt{a + b\tan(e + fx)}}{\sqrt{c + d\tan(e + fx)}}\right)}{(c + id)f}$$

$$= -\frac{(iA + B - iC)\text{arctanh}\left(\frac{\sqrt{c - id}\sqrt{a + b\tan(e + fx)}}{\sqrt{a - ib}\sqrt{c + d\tan(e + fx)}}\right)}{\sqrt{a - ib}(c - id)^{3/2} f} - \frac{(B - i(A - C))\text{arctanh}\left(\frac{\sqrt{c + id}\sqrt{a + b\tan(e + fx)}}{\sqrt{a + ib}\sqrt{c + d\tan(e + fx)}}\right)}{\sqrt{a + ib}(c + id)^{3/2} f} + \frac{2(c^2C - Bcd + Ad^2)\sqrt{a + b\tan(e + fx)}}{(bc - ad)(c^2 + d^2) f\sqrt{c + d\tan(e + fx)}}$$

Mathematica [A] (verified)

Time = 3.53 (sec) , antiderivative size = 275, normalized size of antiderivative = 1.10

$$\int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{\sqrt{a + b\tan(e + fx)}(c + d\tan(e + fx))^{3/2}} dx =$$

$$-\frac{(bc - ad)\left(\frac{(iA + B - iC)(c + id)\text{arctanh}\left(\frac{\sqrt{-c + id}\sqrt{a + b\tan(e + fx)}}{\sqrt{-a + ib}\sqrt{c + d\tan(e + fx)}}\right)}{\sqrt{-a + ib}\sqrt{-c + id}} + \frac{(A + iB - C)(ic + d)\text{arctanh}\left(\frac{\sqrt{c + id}\sqrt{a + b\tan(e + fx)}}{\sqrt{a + ib}\sqrt{c + d\tan(e + fx)}}\right)}{\sqrt{a + ib}\sqrt{c + id}}\right) + 2(c^2C}{(-bc + ad)(c^2 + d^2) f}$$

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/(Sqrt[a + b*Tan[e + f*x]]
*(c + d*Tan[e + f*x])^(3/2)),x]
```

```
[Out] -(((b*c - a*d)*(((I*A + B - I*C)*(c + I*d)*ArcTanh[(Sqrt[-c + I*d]*Sqrt[a +
 b*Tan[e + f*x]])/(Sqrt[-a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[-a + I*
b]*Sqrt[-c + I*d]) + ((A + I*B - C)*(I*c + d)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a
 + b*Tan[e + f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[a + I*
b]*Sqrt[c + I*d])) + (2*(c^2*C - B*c*d + A*d^2)*Sqrt[a + b*Tan[e + f*x]])/S
qrt[c + d*Tan[e + f*x]])/((-(b*c) + a*d)*(c^2 + d^2)*f))
```

Maple [F(-1)]

Timed out.

$$\int \frac{A+B\tan(fx+e)+C\tan(fx+e)^2}{\sqrt{a+b\tan(fx+e)}\,(c+d\tan(fx+e))^{\frac{3}{2}}}dx$$

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(1/2)/(c+d*tan(f*x+e))
^(3/2),x)
```

```
[Out] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(1/2)/(c+d*tan(f*x+e))
^(3/2),x)
```

Fricas [F(-1)]

Timed out.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(1/2)/(c+d*tan(f
*x+e))^(3/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}\,dx = \int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}\,(c+d\tan(e+fx))^{\frac{3}{2}}}\,dx$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e))**(1/2)/(c+d*tan
(f*x+e))**(3/2),x)
```

```
[Out] Integral((A + B*tan(e + f*x) + C*tan(e + f*x)**2)/(sqrt(a + b*tan(e + f*x))
*(c + d*tan(e + f*x))**(3/2)), x)
```

Maxima [F]

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}\,dx = \int \frac{C\tan(fx+e)^2+B\tan(fx+e)+A}{\sqrt{b\tan(fx+e)+a}(d\tan(fx+e)+c)^{\frac{3}{2}}}\,dx$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(1/2)/(c+d*tan(f
*x+e))^(3/2),x, algorithm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)/(sqrt(b*tan(f*x + e) + a)
*(d*tan(f*x + e) + c)^(3/2)), x)
```

Giac [F(-1)]

Timed out.

$$\int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{\sqrt{a + b\tan(e + fx)}(c + d\tan(e + fx))^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(1/2)/(c+d*tan(f
*x+e))^(3/2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{\sqrt{a + b\tan(e + fx)}(c + d\tan(e + fx))^{3/2}}\,dx = \text{Hanged}$$

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))^(1/2)*(c
+ d*tan(e + f*x))^(3/2)),x)
```

```
[Out] \text{Hanged}
```

3.157 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}\,dx$

Optimal result 2100
Rubi [A] (verified) 2101
Mathematica [A] (verified) 2104
Maple [F(-1)] 2104
Fricas [F(-1)] 2105
Sympy [F] 2105
Maxima [F(-1)] 2105
Giac [F(-1)] 2105
Mupad [F(-1)] 2106

Optimal result

Integrand size = 49, antiderivative size = 383

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}\,dx = -\frac{(iA+B-iC)\,\mathrm{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a-ib)^{3/2}(c-id)^{3/2}f} - \frac{(B-i(A-C))\,\mathrm{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a+ib)^{3/2}(c+id)^{3/2}f} - \frac{2(Ab^2-a(bB-aC))}{(a^2+b^2)(bc-ad)f\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}} - \frac{2d(b^2c(cC-Bd)-abB(c^2+d^2)+a^2(2c^2C-Bcd+Cd^2)+A(a^2d^2+b^2(c^2+2d^2)))\sqrt{a+b\tan(e+fx)}}{(a^2+b^2)(bc-ad)^2(c^2+d^2)f\sqrt{c+d\tan(e+fx)}}$$

```
[Out] -(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(1/2)/(c+
d*tan(f*x+e))^(1/2))/(a-I*b)^(3/2)/(c-I*d)^(3/2)/f-(B-I*(A-C))*arctanh((c+I
*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/(a+I
*b)^(3/2)/(c+I*d)^(3/2)/f-2*(A*b^2-a*(B*b-C*a))/(a^2+b^2)/(-a*d+b*c)/f/(a+b
*tan(f*x+e))^(1/2)/(c+d*tan(f*x+e))^(1/2)-2*d*(b^2*c*(-B*d+C*c)-a*b*B*(c^2+
d^2)+a^2*(-B*c*d+2*C*c^2+C*d^2)+A*(a^2*d^2+b^2*(c^2+2*d^2)))*(a+b*tan(f*x+e
))^(1/2)/(a^2+b^2)/(-a*d+b*c)^2/(c^2+d^2)/f/(c+d*tan(f*x+e))^(1/2)
```

Rubi [A] (verified)

Time = 2.28 (sec) , antiderivative size = 382, normalized size of antiderivative = 1.00, number of steps used = 9, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.102, Rules used = {3730, 3697, 3696, 95, 214}

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}\,dx =$$
$$-\frac{2d\sqrt{a+b\tan(e+fx)}(a^2Ad^2+a^2(-Bcd+2c^2C+Cd^2)-abB(c^2+d^2)+Ab^2(c^2+2d^2)+b^2c(cC-Bc}{f(a^2+b^2)(c^2+d^2)(bc-ad)^2\sqrt{c+d\tan(e+fx)}}$$
$$-\frac{2(Ab^2-a(bB-aC))}{f(a^2+b^2)(bc-ad)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}$$
$$-\frac{(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(a-ib)^{3/2}(c-id)^{3/2}}$$
$$-\frac{(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(a+ib)^{3/2}(c+id)^{3/2}}$$

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^(3/2)*(c
+ d*Tan[e + f*x])^(3/2)),x]
```

```
[Out] -(((I*A + B - I*C)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a
 - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a - I*b)^(3/2)*(c - I*d)^(3/2)*f)) -
((B - I*(A - C))*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a +
 I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a + I*b)^(3/2)*(c + I*d)^(3/2)*f) - (2*
(A*b^2 - a*(b*B - a*C)))/((a^2 + b^2)*(b*c - a*d)*f*Sqrt[a + b*Tan[e + f*x]
]*Sqrt[c + d*Tan[e + f*x]]) - (2*d*(a^2*A*d^2 + b^2*c*(c*C - B*d) - a*b*B*(
c^2 + d^2) + A*b^2*(c^2 + 2*d^2) + a^2*(2*c^2*C - B*c*d + C*d^2))*Sqrt[a +
b*Tan[e + f*x]])/((a^2 + b^2)*(b*c - a*d)^2*(c^2 + d^2)*f*Sqrt[c + d*Tan[e
+ f*x]])
```

Rule 95

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3696

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[A^2/f, Subst[Int[(a + b*x)^m*((c + d*x)^n/(A - B*x)), x], x, Tan[e + f*x
]], x] /; FreeQ[{a, b, c, d, e, f, A, B, m, n}, x] && NeQ[b*c - a*d, 0] &&
NeQ[a^2 + b^2, 0] && EqQ[A^2 + B^2, 0]
```

Rule 3697

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[(A + I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^n*(1 - I*Ta
n[e + f*x]), x], x] + Dist[(A - I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*T
an[e + f*x])^n*(1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A,
B, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[A^2 + B^2, 0]
```

Rule 3730

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 - a*(b*B - a*C))*(a + b*Tan[e +
 f*x])^(m + 1)*((c + d*Tan[e + f*x])^(n + 1)/(f*(m + 1)*(b*c - a*d)*(a^2 +
b^2))), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rubi steps

$$\text{integral} = -\frac{2(Ab^2-a(bB-aC))}{(a^2+b^2)(bc-ad)f\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}} - \frac{2\int \frac{\frac{1}{2}(2Ab^2d-aA(bc-ad)-(bB-aC)(bc+ad))+\frac{1}{2}(Ab-aB-bC)(bc-ad)\tan(e+fx)+(Ab^2-a(bB-aC))d\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}\,dx}{(a^2+b^2)(bc-ad)}$$

$$= -\frac{2(Ab^2-a(bB-aC))}{(a^2+b^2)(bc-ad)f\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}} - \frac{2d(a^2Ad^2+b^2c(cC-Bd)-abB(c^2+d^2)+Ab^2(c^2+2d^2)+a^2(2c^2C-Bcd+Cd^2))\sqrt{a+b\,\text{ta}}}{(a^2+b^2)(bc-ad)^2(c^2+d^2)f\sqrt{c+d\tan(e+fx)}} - \frac{4\int \frac{-\frac{1}{4}(bc-ad)^2(bBc-b(A-C)d+a(Ac-cC+Bd))-\frac{1}{4}(bc-ad)^2(bcC-bBd-A(bc+ad)+a(Bc+Cd))\tan(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx}{(a^2+b^2)(bc-ad)^2(c^2+d^2)}$$

$$= -\frac{2(Ab^2 - a(bB - aC))}{(a^2+b^2)(bc-ad)f\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}} - \frac{2d(a^2Ad^2 + b^2c(cC - Bd) - abB(c^2+d^2) + Ab^2(c^2+2d^2) + a^2(2c^2C - Bcd + Cd^2))\sqrt{a+b\tan(e+fx)}}{(a^2+b^2)(bc-ad)^2(c^2+d^2)f\sqrt{c+d\tan(e+fx)}} + \frac{(A-iB-C)\int\frac{1+i\tan(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx}{2(a-ib)(c-id)} + \frac{(A+iB-C)\int\frac{1-i\tan(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx}{2(a+ib)(c+id)}$$

$$= -\frac{2(Ab^2 - a(bB - aC))}{(a^2+b^2)(bc-ad)f\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}} - \frac{2d(a^2Ad^2 + b^2c(cC - Bd) - abB(c^2+d^2) + Ab^2(c^2+2d^2) + a^2(2c^2C - Bcd + Cd^2))\sqrt{a+b\tan(e+fx)}}{(a^2+b^2)(bc-ad)^2(c^2+d^2)f\sqrt{c+d\tan(e+fx)}} + \frac{(A-iB-C)\text{Subst}\left(\int\frac{1}{(1-ix)\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e+fx)\right)}{2(a-ib)(c-id)f} + \frac{(A+iB-C)\text{Subst}\left(\int\frac{1}{(1+ix)\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e+fx)\right)}{2(a+ib)(c+id)f}$$

$$= -\frac{2(Ab^2 - a(bB - aC))}{(a^2+b^2)(bc-ad)f\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}} - \frac{2d(a^2Ad^2 + b^2c(cC - Bd) - abB(c^2+d^2) + Ab^2(c^2+2d^2) + a^2(2c^2C - Bcd + Cd^2))\sqrt{a+b\tan(e+fx)}}{(a^2+b^2)(bc-ad)^2(c^2+d^2)f\sqrt{c+d\tan(e+fx)}} + \frac{(A-iB-C)\text{Subst}\left(\int\frac{1}{ia+b-(ic+d)x^2}\,dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{(a-ib)(c-id)f} + \frac{(A+iB-C)\text{Subst}\left(\int\frac{1}{-ia+b-(-ic+d)x^2}\,dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{(a+ib)(c+id)f}$$

$$= -\frac{(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a-ib)^{3/2}(c-id)^{3/2}f} - \frac{(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a+ib)^{3/2}(c+id)^{3/2}f} - \frac{2(Ab^2 - a(bB - aC))}{(a^2+b^2)(bc-ad)f\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}} - \frac{2d(a^2Ad^2 + b^2c(cC - Bd) - abB(c^2+d^2) + Ab^2(c^2+2d^2) + a^2(2c^2C - Bcd + Cd^2))\sqrt{a+b\tan(e+fx)}}{(a^2+b^2)(bc-ad)^2(c^2+d^2)f\sqrt{c+d\tan(e+fx)}}$$

Mathematica [A] (verified)

Time = 6.87 (sec) , antiderivative size = 484, normalized size of antiderivative = 1.26

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}\,dx = -\frac{2(Ab^2-a(bB-aC))}{(a^2+b^2)(bc-ad)f\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}} - \frac{2\left(\frac{(bc-ad)^2\left(\frac{(a+ib)(iA+B-iC)(c+id)\operatorname{arctanh}\left(\frac{\sqrt{-c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{-a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{-a+ib}\sqrt{-c+id}}+\frac{(ia+b)(A+iB-C)(c-id)\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{a+ib}\sqrt{c+id}}\right)}{2(-bc+ad)(c^2+d^2)f} - \frac{2(-\cdots}{\cdots}\right)}{(a^2+b^2)(bc-ad\cdots}$$

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^(3/
2)*(c + d*Tan[e + f*x])^(3/2)),x]
```

```
[Out] (-2*(A*b^2 - a*(b*B - a*C)))/((a^2 + b^2)*(b*c - a*d)*f*Sqrt[a + b*Tan[e +
f*x]]*Sqrt[c + d*Tan[e + f*x]]) - (2*(((b*c - a*d)^2*(((a + I*b)*(I*A + B -
 I*C)*(c + I*d)*ArcTanh[(Sqrt[-c + I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[-a
+ I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[-a + I*b]*Sqrt[-c + I*d]) + ((I*a
+ b)*(A + I*B - C)*(c - I*d)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]
])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[a + I*b]*Sqrt[c + I*d])
))/(2*(-(b*c) + a*d)*(c^2 + d^2)*f) - (2*(-(c*(-(c*(A*b^2 - a*(b*B - a*C))*
d) + ((A*b - a*B - b*C)*d*(b*c - a*d))/2)) + (d^2*(2*A*b^2*d - a*A*(b*c - a
*d) - (b*B - a*C)*(b*c + a*d)))/2)*Sqrt[a + b*Tan[e + f*x]])/((-(b*c) + a*d
)*(c^2 + d^2)*f*Sqrt[c + d*Tan[e + f*x]])))/((a^2 + b^2)*(b*c - a*d))
```

Maple [F(-1)]

Timed out.

$$\int \frac{A+B\tan(fx+e)+C\tan(fx+e)^2}{(a+b\tan(fx+e))^{\frac{3}{2}}(c+d\tan(fx+e))^{\frac{3}{2}}}\,dx$$

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(3/2)/(c+d*tan(f*x+e))
^(3/2),x)
```

```
[Out] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(3/2)/(c+d*tan(f*x+e))
^(3/2),x)
```

Fricas [F(-1)]

Timed out.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(3/2)/(c+d*tan(f
*x+e))^(3/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}\,dx = \int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{\frac{3}{2}}(c+d\tan(e+fx))^{\frac{3}{2}}}\,dx$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e))**(3/2)/(c+d*tan
(f*x+e))**(3/2),x)
```

```
[Out] Integral((A + B*tan(e + f*x) + C*tan(e + f*x)**2)/((a + b*tan(e + f*x))**(3
/2)*(c + d*tan(e + f*x))**(3/2)), x)
```

Maxima [F(-1)]

Timed out.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(3/2)/(c+d*tan(f
*x+e))^(3/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

Giac [F(-1)]

Timed out.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(3/2)/(c+d*tan(f
*x+e))^(3/2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{A + B\tan(e + fx) + C\tan^2(e + fx)}{(a + b\tan(e + fx))^{3/2}(c + d\tan(e + fx))^{3/2}}\,dx = \text{Hanged}$$

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))^(3/2)*(c
+ d*tan(e + f*x))^(3/2)),x)
```

```
[Out] \text{Hanged}
```

3.158 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{5/2}(c+d\tan(e+fx))^{3/2}}\,dx$

Optimal result 2107
Rubi [A] (verified) 2108
Mathematica [A] (verified) 2111
Maple [F(-1)] 2112
Fricas [F(-1)] 2112
Sympy [F] 2113
Maxima [F(-1)] 2113
Giac [F(-1)] 2113
Mupad [F(-1)] 2113

Optimal result

Integrand size = 49, antiderivative size = 598

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{5/2}(c+d\tan(e+fx))^{3/2}}\,dx =$$

$$-\frac{(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a-ib)^{5/2}(c-id)^{3/2}f}$$
$$-\frac{(B-i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a+ib)^{5/2}(c+id)^{3/2}f}$$
$$-\frac{2(Ab^2-a(bB-aC))}{3(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}$$
$$-\frac{2(7a^3bBd-4a^4Cd+b^4(3Bc-4Ad)+ab^3(6Ac-6cC+Bd)-a^2b^2(3Bc+2(5A-C)d))}{3(a^2+b^2)^2(bc-ad)^2f\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}$$
$$-\frac{2d(8a^3bBd(c^2+d^2)+2ab^3(3Ac-3cC+Bd)(c^2+d^2)-a^4d(8c^2C-3Bcd+(3A+5C)d^2)-a^2b^2(3Bc^3}{3(a^2+b^2)^2(bc-ad)^3}$$

```
[Out] -(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(1/2)/(c+
d*tan(f*x+e))^(1/2))/(a-I*b)^(5/2)/(c-I*d)^(3/2)/f-(B-I*(A-C))*arctanh((c+I
*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/(a+I
*b)^(5/2)/(c+I*d)^(3/2)/f-2/3*(7*a^3*b*B*d-4*a^4*C*d+b^4*(-4*A*d+3*B*c)+a*b
^3*(6*A*c+B*d-6*C*c)-a^2*b^2*(3*B*c+2*(5*A-C)*d))/(a^2+b^2)^2/(-a*d+b*c)^2/
f/(a+b*tan(f*x+e))^(1/2)/(c+d*tan(f*x+e))^(1/2)-2/3*d*(8*a^3*b*B*d*(c^2+d^2
)+2*a*b^3*(3*A*c+B*d-3*C*c)*(c^2+d^2)-a^4*d*(8*c^2*C-3*B*c*d+(3*A+5*C)*d^2)
-a^2*b^2*(11*A*c^2*d+17*A*d^3+3*B*c^3-3*B*c*d^2+5*C*c^2*d-C*d^3)-b^4*(d*(5*
A*c^2+8*A*d^2+3*C*c^2)-3*B*(c^3+2*c*d^2)))*(a+b*tan(f*x+e))^(1/2)/(a^2+b^2)
^2/(-a*d+b*c)^3/(c^2+d^2)/f/(c+d*tan(f*x+e))^(1/2)-2/3*(A*b^2-a*(B*b-C*a))/
(a^2+b^2)/(-a*d+b*c)/f/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f*x+e))^(3/2)
```

Rubi [A] (verified)

Time = 4.04 (sec) , antiderivative size = 598, normalized size of antiderivative = 1.00, number of steps used = 10, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.102, Rules used = {3730, 3697, 3696, 95, 214}

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{5/2}(c+d\tan(e+fx))^{3/2}}\,dx =$$

$$-\frac{2(Ab^2-a(bB-aC))}{3f(a^2+b^2)(bc-ad)(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}$$

$$-\frac{2d\sqrt{a+b\tan(e+fx)}(a^4(-d)\left(d^2(3A+5C)-3Bcd+8c^2C\right)+8a^3bBd(c^2+d^2)-a^2b^2(11Ac^2d+17Ad^3 \cdots}{3f\left(a^2+b^2\right)^2\left(c^2+d^2 \cdots\right.}$$

$$-\frac{2(-4a^4Cd+7a^3bBd-a^2b^2(2d(5A-C)+3Bc)+ab^3(6Ac+Bd-6cC)+b^4(3Bc-4Ad))}{3f\left(a^2+b^2\right)^2(bc-ad)^2\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}$$

$$-\frac{(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(a-ib)^{5/2}(c-id)^{3/2}}$$

$$-\frac{(B-i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(a+ib)^{5/2}(c+id)^{3/2}}$$

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^(5/2)*(c
+ d*Tan[e + f*x])^(3/2)),x]
```

```
[Out] -(((I*A + B - I*C)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a
 - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a - I*b)^(5/2)*(c - I*d)^(3/2)*f)) -
((B - I*(A - C))*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a +
 I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a + I*b)^(5/2)*(c + I*d)^(3/2)*f) - (2*
(A*b^2 - a*(b*B - a*C)))/(3*(a^2 + b^2)*(b*c - a*d)*f*(a + b*Tan[e + f*x])^
(3/2)*Sqrt[c + d*Tan[e + f*x]]) - (2*(7*a^3*b*B*d - 4*a^4*C*d + b^4*(3*B*c
- 4*A*d) + a*b^3*(6*A*c - 6*c*C + B*d) - a^2*b^2*(3*B*c + 2*(5*A - C)*d)))/
(3*(a^2 + b^2)^2*(b*c - a*d)^2*f*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e
+ f*x]]) - (2*d*(8*a^3*b*B*d*(c^2 + d^2) + 2*a*b^3*(3*A*c - 3*c*C + B*d)*(c
^2 + d^2) - a^4*d*(8*c^2*C - 3*B*c*d + (3*A + 5*C)*d^2) - a^2*b^2*(3*B*c^3
+ 11*A*c^2*d + 5*c^2*C*d - 3*B*c*d^2 + 17*A*d^3 - C*d^3) - b^4*(d*(5*A*c^2
+ 3*c^2*C + 8*A*d^2) - 3*B*(c^3 + 2*c*d^2)))*Sqrt[a + b*Tan[e + f*x]])/(3*(
a^2 + b^2)^2*(b*c - a*d)^3*(c^2 + d^2)*f*Sqrt[c + d*Tan[e + f*x]])
```

Rule 95

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3696

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[A^2/f, Subst[Int[(a + b*x)^m*((c + d*x)^n/(A - B*x)), x], x, Tan[e + f*x
]], x] /; FreeQ[{a, b, c, d, e, f, A, B, m, n}, x] && NeQ[b*c - a*d, 0] &&
NeQ[a^2 + b^2, 0] && EqQ[A^2 + B^2, 0]
```

Rule 3697

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[(A + I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^n*(1 - I*Ta
n[e + f*x]), x], x] + Dist[(A - I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*T
an[e + f*x])^n*(1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A,
B, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[A^2 + B^2, 0]
```

Rule 3730

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 - a*(b*B - a*C))*(a + b*Tan[e +
 f*x])^(m + 1)*((c + d*Tan[e + f*x])^(n + 1)/(f*(m + 1)*(b*c - a*d)*(a^2 +
b^2))), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rubi steps

$$\text{integral} = -\frac{2(Ab^2-a(bB-aC))}{3\,(a^2+b^2)\,(bc-ad)f(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}} - \frac{2\int \frac{\frac{1}{2}\left(4Ab^2d-3aA(bc-ad)-(bB-aC)(3bc+ad)\right)+\frac{3}{2}(Ab-aB-bC)(bc-ad)\tan(e+fx)+2\left(Ab^2-a(bB-aC)\right)d\tan^2(e+fx)}{(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}\,dx}{3\,(a^2+b^2)\,(bc-ad)}$$

$$= -\frac{2(Ab^2 - a(bB - aC))}{3(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}$$
$$-\frac{2(7a^3bBd - 4a^4Cd + b^4(3Bc-4Ad) + ab^3(6Ac-6cC+Bd) - a^2b^2(3Bc+2(5A-C)d))}{3(a^2+b^2)^2(bc-ad)^2f\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}$$
$$+\frac{4\int \frac{\frac{1}{4}\left(-\left((abc-a^2d-2b^2d)\left(a^2(3A+C)d-b^2(3Bc-4Ad)-ab(3Ac-3cC+Bd)\right)\right)+(bc+ad)\left(3b^3cC-7a^2bBd+4a^3Cd-Ab^2(3bc-7ad)+\right.\right.}{}}{}$$

$$= -\frac{2(Ab^2 - a(bB - aC))}{3(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}$$
$$-\frac{2(7a^3bBd - 4a^4Cd + b^4(3Bc-4Ad) + ab^3(6Ac-6cC+Bd) - a^2b^2(3Bc+2(5A-C)d))}{3(a^2+b^2)^2(bc-ad)^2f\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}$$
$$-\frac{2d(8a^3bBd(c^2+d^2) + 2ab^3(3Ac-3cC+Bd)(c^2+d^2) - a^4d(8c^2C-3Bcd+(3A+5C)d^2) - a}{3(a^2+b^2)^2(}$$
$$+\frac{8\int \frac{\frac{3}{8}(bc-ad)^3\left(a^2(Ac-cC+Bd)-b^2(Ac-cC+Bd)+2ab(Bc-(A-C)d)\right)-\frac{3}{8}(bc-ad)^3\left(2ab(Ac-cC+Bd)-a^2(Bc-(A-C)d)+b^2(Bc-\right.}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}}{3(a^2+b^2)^2(bc-ad)^3(c^2+d^2)}$$

$$= -\frac{2(Ab^2 - a(bB - aC))}{3(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}$$
$$-\frac{2(7a^3bBd - 4a^4Cd + b^4(3Bc-4Ad) + ab^3(6Ac-6cC+Bd) - a^2b^2(3Bc+2(5A-C)d))}{3(a^2+b^2)^2(bc-ad)^2f\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}$$
$$-\frac{2d(8a^3bBd(c^2+d^2) + 2ab^3(3Ac-3cC+Bd)(c^2+d^2) - a^4d(8c^2C-3Bcd+(3A+5C)d^2) - a}{3(a^2+b^2)^2(}$$
$$+\frac{(A-iB-C)\int \frac{1+i\tan(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx}{2(a-ib)^2(c-id)}$$
$$+\frac{(A+iB-C)\int \frac{1-i\tan(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx}{2(a+ib)^2(c+id)}$$

$$= -\frac{2(Ab^2 - a(bB - aC))}{3(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}$$
$$-\frac{2(7a^3bBd - 4a^4Cd + b^4(3Bc-4Ad) + ab^3(6Ac-6cC+Bd) - a^2b^2(3Bc+2(5A-C)d))}{3(a^2+b^2)^2(bc-ad)^2f\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}$$
$$-\frac{2d(8a^3bBd(c^2+d^2) + 2ab^3(3Ac-3cC+Bd)(c^2+d^2) - a^4d(8c^2C-3Bcd+(3A+5C)d^2) - a}{3(a^2+b^2)^2(}$$
$$+\frac{(A-iB-C)\text{Subst}\left(\int \frac{1}{(1-ix)\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e+fx)\right)}{2(a-ib)^2(c-id)f}$$
$$+\frac{(A+iB-C)\text{Subst}\left(\int \frac{1}{(1+ix)\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e+fx)\right)}{2(a+ib)^2(c+id)f}$$

$$= -\frac{2(Ab^2 - a(bB - aC))}{3\left(a^2+b^2\right)(bc-ad)f(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}$$
$$-\frac{2(7a^3bBd - 4a^4Cd + b^4(3Bc-4Ad) + ab^3(6Ac-6cC+Bd) - a^2b^2(3Bc+2(5A-C)d))}{3\left(a^2+b^2\right)^2(bc-ad)^2f\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}$$
$$-\frac{2d(8a^3bBd(c^2+d^2) + 2ab^3(3Ac-3cC+Bd)\left(c^2+d^2\right) - a^4d(8c^2C - 3Bcd + (3A+5C)d^2) -}{3\left(a^2+b^2\right)^2}$$
$$+\frac{(A-iB-C)\text{Subst}\left(\int \frac{1}{ia+b-(ic+d)x^2}\,dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{(a-ib)^2(c-id)f}$$
$$+\frac{(A+iB-C)\text{Subst}\left(\int \frac{1}{-ia+b-(-ic+d)x^2}\,dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{(a+ib)^2(c+id)f}$$

$$= -\frac{(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a-ib)^{5/2}(c-id)^{3/2}f}$$
$$-\frac{(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a+ib)^{5/2}(c+id)^{3/2}f}$$
$$-\frac{2(Ab^2 - a(bB - aC))}{3\left(a^2+b^2\right)(bc-ad)f(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}$$
$$-\frac{2(7a^3bBd - 4a^4Cd + b^4(3Bc-4Ad) + ab^3(6Ac-6cC+Bd) - a^2b^2(3Bc+2(5A-C)d))}{3\left(a^2+b^2\right)^2(bc-ad)^2f\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}$$
$$-\frac{2d(8a^3bBd(c^2+d^2) + 2ab^3(3Ac-3cC+Bd)\left(c^2+d^2\right) - a^4d(8c^2C - 3Bcd + (3A+5C)d^2) -}{3\left(a^2+b^2\right)^2}$$

Mathematica [A] (verified)

Time = 7.17 (sec) , antiderivative size = 902, normalized size of antiderivative = 1.51

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{5/2}(c+d\tan(e+fx))^{3/2}}\,dx =$$
$$-\frac{2(Ab^2 - a(bB - aC))}{3\left(a^2+b^2\right)(bc-ad)f(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}$$
$$-\frac{2\left(-\frac{2\left(-a\left(-2a\left(Ab^2-a(bB-aC)\right)d+\frac{3}{2}b(Ab-aB-bC)(bc-ad)\right)+\frac{1}{2}b^2\left(4Ab^2d-3aA(bc-ad)-(bB-aC)(3bc+ad)\right)\right)}{(a^2+b^2)(bc-ad)f\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}} - \frac{2\left(-\frac{3(bc-ad)^3\left(\frac{(a+ib)^2}{\cdots}\right.}{\cdots}\right.}{\cdots}\right.}{\cdots}$$

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^(5/
2)*(c + d*Tan[e + f*x])^(3/2)),x]
```

```
[Out] (-2*(A*b^2 - a*(b*B - a*C)))/(3*(a^2 + b^2)*(b*c - a*d)*f*(a + b*Tan[e + f*
x])^(3/2)*Sqrt[c + d*Tan[e + f*x]]) - (2*((-2*(-(a*(-2*a*(A*b^2 - a*(b*B -
a*C))*d + (3*b*(A*b - a*B - b*C)*(b*c - a*d))/2)) + (b^2*(4*A*b^2*d - 3*a*A
*(b*c - a*d) - (b*B - a*C)*(3*b*c + a*d)))/2))/((a^2 + b^2)*(b*c - a*d)*f*S
qrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]]) - (2*((-3*(b*c - a*d)^3*(
((a + I*b)^2*(I*A + B - I*C)*(c + I*d)*ArcTanh[(Sqrt[-c + I*d]*Sqrt[a + b*T
an[e + f*x]])/(Sqrt[-a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[-a + I*b]*S
qrt[-c + I*d]) + ((a - I*b)^2*(A + I*B - C)*(I*c + d)*ArcTanh[(Sqrt[c + I*d
]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqr
t[a + I*b]*Sqrt[c + I*d])))/(4*(-(b*c) + a*d)*(c^2 + d^2)*f) - (2*(d^2*((-1
/2*(b*c) - (a*d)/2)*(-2*a*(A*b^2 - a*(b*B - a*C))*d + (3*b*(A*b - a*B - b*C
)*(b*c - a*d))/2) + ((b^2*d - (a*(b*c - a*d))/2)*(4*A*b^2*d - 3*a*A*(b*c -
a*d) - (b*B - a*C)*(3*b*c + a*d)))/2) - c*((d*(b*c - a*d)*(-2*b*(A*b^2 - a*
(b*B - a*C))*d - (3*a*(A*b - a*B - b*C)*(b*c - a*d))/2 + (b*(4*A*b^2*d - 3*
a*A*(b*c - a*d) - (b*B - a*C)*(3*b*c + a*d)))/2))/2 - c*d*(-(a*(-2*a*(A*b^2
 - a*(b*B - a*C))*d + (3*b*(A*b - a*B - b*C)*(b*c - a*d))/2)) + (b^2*(4*A*b
^2*d - 3*a*A*(b*c - a*d) - (b*B - a*C)*(3*b*c + a*d)))/2)))*Sqrt[a + b*Tan[
e + f*x]])/((-(b*c) + a*d)*(c^2 + d^2)*f*Sqrt[c + d*Tan[e + f*x]])))/((a^2
+ b^2)*(b*c - a*d))))/(3*(a^2 + b^2)*(b*c - a*d))
```

Maple [F(-1)]

Timed out.

$$\int \frac{A+B\tan(fx+e)+C\tan(fx+e)^2}{(a+b\tan(fx+e))^{\frac{5}{2}}(c+d\tan(fx+e))^{\frac{3}{2}}}dx$$

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(5/2)/(c+d*tan(f*x+e))
^(3/2),x)
```

```
[Out] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(5/2)/(c+d*tan(f*x+e))
^(3/2),x)
```

Fricas [F(-1)]

Timed out.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{5/2}(c+d\tan(e+fx))^{3/2}}\,dx=\text{Timed out}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(5/2)/(c+d*tan(f
*x+e))^(3/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{5/2}(c+d\tan(e+fx))^{3/2}}\,dx = \int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{\frac{5}{2}}\,(c+d\tan(e+fx))^{\frac{3}{2}}}\,dx$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e))**(5/2)/(c+d*tan
(f*x+e))**(3/2),x)
[Out] Integral((A + B*tan(e + f*x) + C*tan(e + f*x)**2)/((a + b*tan(e + f*x))**(5
/2)*(c + d*tan(e + f*x))**(3/2)), x)
```

Maxima [F(-1)]

Timed out.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{5/2}(c+d\tan(e+fx))^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(5/2)/(c+d*tan(f
*x+e))^(3/2),x, algorithm="maxima")
[Out] Timed out
```

Giac [F(-1)]

Timed out.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{5/2}(c+d\tan(e+fx))^{3/2}}\,dx = \text{Timed out}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(5/2)/(c+d*tan(f
*x+e))^(3/2),x, algorithm="giac")
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{5/2}(c+d\tan(e+fx))^{3/2}}\,dx = \text{Hanged}$$

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))^(5/2)*(c
+ d*tan(e + f*x))^(3/2)),x)
[Out] \text{Hanged}
```

3.159 $\int \frac{(a+b\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx$

Optimal result 2114
Rubi [A] (verified) 2115
Mathematica [B] (verified) 2120
Maple [F(-1)] 2122
Fricas [F(-1)] 2122
Sympy [F] 2122
Maxima [F(-1)] 2122
Giac [F(-1)] 2123
Mupad [F(-1)] 2123

Optimal result

Integrand size = 49, antiderivative size = 549

$$\int \frac{(a+b\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{5/2}}\,dx =$$
$$-\frac{(a-ib)^{5/2}(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(c-id)^{5/2}f}$$
$$-\frac{(a+ib)^{5/2}(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(c+id)^{5/2}f}$$
$$-\frac{b^{3/2}(5bcC-2bBd-5aCd)\text{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{d^{7/2}f}$$
$$-\frac{2(c^2C-Bcd+Ad^2)\,(a+b\tan(e+fx))^{5/2}}{3d\,(c^2+d^2)\,f(c+d\tan(e+fx))^{3/2}}$$
$$-\frac{2(b(5c^4C-2Bc^3d-c^2(A-11C)d^2-8Bcd^3+5Ad^4)+3ad^2(2c(A-C)d-B(c^2-d^2)))\,(a+b\tan(e+f}{3d^2\,(c^2+d^2)^2\,f\sqrt{c+d\tan(e+fx)}}$$
$$+\frac{b(b(5c^4C-2Bc^3d+10c^2Cd^2-6Bcd^3+(4A+C)d^4)+2ad^2(2c(A-C)d-B(c^2-d^2)))\,\sqrt{a+b\tan(e+}{d^3\,(c^2+d^2)^2\,f}$$

```
[Out] -(a-I*b)^(5/2)*(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-
I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/(c-I*d)^(5/2)/f-(a+I*b)^(5/2)*(B-I*(A-C)
)*arctanh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e
))^(1/2))/(c+I*d)^(5/2)/f-b^(3/2)*(-2*B*b*d-5*C*a*d+5*C*b*c)*arctanh(d^(1/2
)*(a+b*tan(f*x+e))^(1/2)/b^(1/2)/(c+d*tan(f*x+e))^(1/2))/d^(7/2)/f+b*(b*(5*
c^4*C-2*B*c^3*d+10*C*c^2*d^2-6*B*c*d^3+(4*A+C)*d^4)+2*a*d^2*(2*c*(A-C)*d-B*
(c^2-d^2)))*(a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)/d^3/(c^2+d^2)^2/f
-2/3*(b*(5*c^4*C-2*B*c^3*d-c^2*(A-11*C)*d^2-8*B*c*d^3+5*A*d^4)+3*a*d^2*(2*c
```

```
*(A-C)*d-B*(c^2-d^2)))*(a+b*tan(f*x+e))^(3/2)/d^2/(c^2+d^2)^2/f/(c+d*tan(f*
x+e))^(1/2)-2/3*(A*d^2-B*c*d+C*c^2)*(a+b*tan(f*x+e))^(5/2)/d/(c^2+d^2)/f/(c
+d*tan(f*x+e))^(3/2)
```

Rubi [A] (verified)

Time = 12.40 (sec) , antiderivative size = 549, normalized size of antiderivative = 1.00, number of steps used = 15, number of rules used = 9, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.184, Rules used = {3726, 3728, 3736, 6857, 65, 223, 212, 95, 214}

$$\int \frac{(a+b\tan(e+fx))^{5/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx =$$
$$-\frac{(a-ib)^{5/2}(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(c-id)^{5/2}}$$
$$-\frac{(a+ib)^{5/2}(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(c+id)^{5/2}}$$
$$-\frac{2(Ad^2-Bcd+c^2C)\,(a+b\tan(e+fx))^{5/2}}{3df\,(c^2+d^2)\,(c+d\tan(e+fx))^{3/2}}$$
$$+\frac{b\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}(2ad^2(2cd(A-C)-B(c^2-d^2))+b(d^4(4A+C)-2Bc^3d-6B}{d^3f\,(c^2+d^2)^2}$$
$$-\frac{2(a+b\tan(e+fx))^{3/2}\,(3ad^2(2cd(A-C)-B(c^2-d^2))+b(-c^2d^2(A-11C)+5Ad^4-2Bc^3d-8Bcd^3}{3d^2f\,(c^2+d^2)^2\,\sqrt{c+d\tan(e+fx)}}$$
$$-\frac{b^{3/2}(-5aCd-2bBd+5bcC)\text{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{d^{7/2}f}$$

```
[In] Int[((a + b*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c
 + d*Tan[e + f*x])^(5/2),x]
```

```
[Out] -(((a - I*b)^(5/2)*(I*A + B - I*C)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e
+ f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((c - I*d)^(5/2)*f)) -
((a + I*b)^(5/2)*(B - I*(A - C))*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/((c + I*d)^(5/2)*f) - (b^
(3/2)*(5*b*c*C - 2*b*B*d - 5*a*C*d)*ArcTanh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x
]])/(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])])/(d^(7/2)*f) - (2*(c^2*C - B*c*d +
A*d^2)*(a + b*Tan[e + f*x])^(5/2))/(3*d*(c^2 + d^2)*f*(c + d*Tan[e + f*x])^
(3/2)) - (2*(b*(5*c^4*C - 2*B*c^3*d - c^2*(A - 11*C)*d^2 - 8*B*c*d^3 + 5*A*
d^4) + 3*a*d^2*(2*c*(A - C)*d - B*(c^2 - d^2)))*(a + b*Tan[e + f*x])^(3/2))
/(3*d^2*(c^2 + d^2)^2*f*Sqrt[c + d*Tan[e + f*x]]) + (b*(b*(5*c^4*C - 2*B*c^
3*d + 10*c^2*C*d^2 - 6*B*c*d^3 + (4*A + C)*d^4) + 2*a*d^2*(2*c*(A - C)*d -
B*(c^2 - d^2)))*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(d^3*(c^
2 + d^2)^2*f)
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 95

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 223

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3726

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*d^2 + c*(c*C - B*d))*(a + b*Tan[e +
 f*x])^m*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3736

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[(a + b*ff*x)^m*(c + d*ff*x)^n*((A + B*ff*x + C*ff^2*x^2
)/(1 + ff^2*x^2)), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6857

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\text{integral} = -\frac{2(c^2C-Bcd+Ad^2)\left(a+b\tan(e+fx)\right)^{5/2}}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}} + \frac{2\int \frac{(a+b\tan(e+fx))^{3/2}\left(\frac{1}{2}(Ad(3ac+5bd)+(5bc-3ad)(cC-Bd))+\frac{3}{2}d((A-C)(bc-ad)+B(ac+bd))\tan(e+fx)+\frac{1}{2}b\left(5c^2C-2Bcd+(2A+3C)d\right.\right.}{(c+d\tan(e+fx))^{3/2}}}{3d\left(c^2+d^2\right)}$$

$$= -\frac{2(c^2C-Bcd+Ad^2)\left(a+b\tan(e+fx)\right)^{5/2}}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}} - \frac{2(b(5c^4C-2Bc^3d-c^2(A-11C)d^2-8Bcd^3+5Ad^4)+3ad^2(2c(A-C)d-B(c^2-d^2)))\,(a+}{3d^2\left(c^2+d^2\right)^2f\sqrt{c+d\tan(e+fx)}} + \frac{4\int \frac{\sqrt{a+b\tan(e+fx)}\left(\frac{1}{4}\left(d(ac+3bd)\left(3ad(Ac-cC+Bd)+5b\left(c^2C-Bcd+Ad^2\right)\right)-(3bc-ad)\left(3ad^2(Bc-(A-C)d)-b\left(5c^3C-2Bc^2d-\right.\right.\right.\right.}{}}{}$$

$$= -\frac{2(c^2C-Bcd+Ad^2)(a+b\tan(e+fx))^{5/2}}{3d(c^2+d^2)f(c+d\tan(e+fx))^{3/2}}$$
$$-\frac{2(b(5c^4C-2Bc^3d-c^2(A-11C)d^2-8Bcd^3+5Ad^4)+3ad^2(2c(A-C)d-B(c^2-d^2)))(a+b}{3d^2(c^2+d^2)^2 f\sqrt{c+d\tan(e+fx)}}$$
$$+\frac{b(b(5c^4C-2Bc^3d+10c^2Cd^2-6Bcd^3+(4A+C)d^4)+2ad^2(2c(A-C)d-B(c^2-d^2)))\sqrt{a+}}{d^3(c^2+d^2)^2 f}$$
$$+\frac{4\int\frac{\frac{3}{8}(ab^2d(5c^4C-2c^2(3A-8C)d^2-12Bcd^3+(6A-C)d^4)-b^3c(5c^4C-2Bc^3d+10c^2Cd^2-6Bcd^3+(4A+C)d^4)-2a^3d^3(c^2C-2Bcd-}{}}{}$$

$$= -\frac{2(c^2C-Bcd+Ad^2)(a+b\tan(e+fx))^{5/2}}{3d(c^2+d^2)f(c+d\tan(e+fx))^{3/2}}$$
$$-\frac{2(b(5c^4C-2Bc^3d-c^2(A-11C)d^2-8Bcd^3+5Ad^4)+3ad^2(2c(A-C)d-B(c^2-d^2)))(a+b}{3d^2(c^2+d^2)^2 f\sqrt{c+d\tan(e+fx)}}$$
$$+\frac{b(b(5c^4C-2Bc^3d+10c^2Cd^2-6Bcd^3+(4A+C)d^4)+2ad^2(2c(A-C)d-B(c^2-d^2)))\sqrt{a+}}{d^3(c^2+d^2)^2 f}$$
$$+\frac{4\text{Subst}\left(\int\frac{\frac{3}{8}(ab^2d(5c^4C-2c^2(3A-8C)d^2-12Bcd^3+(6A-C)d^4)-b^3c(5c^4C-2Bc^3d+10c^2Cd^2-6Bcd^3+(4A+C)d^4)-2a^3d^3(c^2C}{}\right.}{}$$

$$= -\frac{2(c^2C-Bcd+Ad^2)(a+b\tan(e+fx))^{5/2}}{3d(c^2+d^2)f(c+d\tan(e+fx))^{3/2}}$$
$$-\frac{2(b(5c^4C-2Bc^3d-c^2(A-11C)d^2-8Bcd^3+5Ad^4)+3ad^2(2c(A-C)d-B(c^2-d^2)))(a+b}{3d^2(c^2+d^2)^2 f\sqrt{c+d\tan(e+fx)}}$$
$$+\frac{b(b(5c^4C-2Bc^3d+10c^2Cd^2-6Bcd^3+(4A+C)d^4)+2ad^2(2c(A-C)d-B(c^2-d^2)))\sqrt{a+}}{d^3(c^2+d^2)^2 f}$$
$$+\frac{4\text{Subst}\left(\int\left(-\frac{3b^2(5bcC-2bBd-5aCd)(c^2+d^2)^2}{8\sqrt{a+bx}\sqrt{c+dx}}+\frac{3(-d^3(a^3(c^2C-2Bcd-Cd^2-A(c^2-d^2))-3ab^2(c^2C-2Bcd-Cd^2-A(c^2-}{}\right.\right.}{}$$

$$= -\frac{2(c^2C-Bcd+Ad^2)(a+b\tan(e+fx))^{5/2}}{3d(c^2+d^2)f(c+d\tan(e+fx))^{3/2}}$$
$$-\frac{2(b(5c^4C-2Bc^3d-c^2(A-11C)d^2-8Bcd^3+5Ad^4)+3ad^2(2c(A-C)d-B(c^2-d^2)))(a+b}{3d^2(c^2+d^2)^2 f\sqrt{c+d\tan(e+fx)}}$$
$$+\frac{b(b(5c^4C-2Bc^3d+10c^2Cd^2-6Bcd^3+(4A+C)d^4)+2ad^2(2c(A-C)d-B(c^2-d^2)))\sqrt{a+}}{d^3(c^2+d^2)^2 f}$$
$$-\frac{(b^2(5bcC-2bBd-5aCd))\,\text{Subst}\left(\int\frac{1}{\sqrt{a+bx}\sqrt{c+dx}}\,dx,x,\tan(e+fx)\right)}{2d^3f}$$
$$+\frac{\text{Subst}\left(\int\frac{-d^3(a^3(c^2C-2Bcd-Cd^2-A(c^2-d^2))-3ab^2(c^2C-2Bcd-Cd^2-A(c^2-d^2))-3a^2b(2c(A-C)d-B(c^2-d^2))+b^3(2c(A-}{}\right.}{}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)\left(a + b\tan(e + fx)\right)^{5/2}}{3d\left(c^2 + d^2\right)f(c + d\tan(e + fx))^{3/2}} - \frac{2(b(5c^4C - 2Bc^3d - c^2(A - 11C)d^2 - 8Bcd^3 + 5Ad^4) + 3ad^2(2c(A - C)d - B(c^2 - d^2)))\,(a +}{3d^2\left(c^2 + d^2\right)^2 f\sqrt{c + d\tan(e + fx)}} + \frac{b(b(5c^4C - 2Bc^3d + 10c^2Cd^2 - 6Bcd^3 + (4A + C)d^4) + 2ad^2(2c(A - C)d - B(c^2 - d^2)))\sqrt{a}}{d^3\left(c^2 + d^2\right)^2 f} - \frac{(b(5bcC - 2bBd - 5aCd))\text{Subst}\left(\int \frac{1}{\sqrt{c - \frac{ad}{b} + \frac{dx^2}{b}}}\,dx, x, \sqrt{a + b\tan(e + fx)}\right)}{d^3 f} + \frac{\text{Subst}\left(\int \left(\frac{d^3\left(3a^2b\left(c^2C - 2Bcd - Cd^2 - A\left(c^2 - d^2\right)\right) - b^3\left(c^2C - 2Bcd - Cd^2 - A\left(c^2 - d^2\right)\right) + a^3\left(2c(A - C)d - B\left(c^2 - d^2\right)\right) - 3ab^2(2c(A}{}\right.\right.}{}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)\left(a + b\tan(e + fx)\right)^{5/2}}{3d\left(c^2 + d^2\right)f(c + d\tan(e + fx))^{3/2}} - \frac{2(b(5c^4C - 2Bc^3d - c^2(A - 11C)d^2 - 8Bcd^3 + 5Ad^4) + 3ad^2(2c(A - C)d - B(c^2 - d^2)))\,(a +}{3d^2\left(c^2 + d^2\right)^2 f\sqrt{c + d\tan(e + fx)}} + \frac{b(b(5c^4C - 2Bc^3d + 10c^2Cd^2 - 6Bcd^3 + (4A + C)d^4) + 2ad^2(2c(A - C)d - B(c^2 - d^2)))\sqrt{a}}{d^3\left(c^2 + d^2\right)^2 f} - \frac{((ia + b)^3(A - iB - C))\,\text{Subst}\left(\int \frac{1}{(i + x)\sqrt{a + bx}\sqrt{c + dx}}\,dx, x, \tan(e + fx)\right)}{2(c - id)^2 f} - \frac{(b(5bcC - 2bBd - 5aCd))\text{Subst}\left(\int \frac{1}{1 - \frac{dx^2}{b}}\,dx, x, \frac{\sqrt{a + b\tan(e + fx)}}{\sqrt{c + d\tan(e + fx)}}\right)}{d^3 f} + \frac{(d^3(3a^2b(c^2C - 2Bcd - Cd^2 - A(c^2 - d^2)) - b^3(c^2C - 2Bcd - Cd^2 - A(c^2 - d^2)) + a^3(2c(A -}{}$$

$$= -\frac{b^{3/2}(5bcC - 2bBd - 5aCd)\text{arctanh}\left(\frac{\sqrt{d}\sqrt{a + b\tan(e + fx)}}{\sqrt{b}\sqrt{c + d\tan(e + fx)}}\right)}{d^{7/2} f} - \frac{2(c^2C - Bcd + Ad^2)\left(a + b\tan(e + fx)\right)^{5/2}}{3d\left(c^2 + d^2\right)f(c + d\tan(e + fx))^{3/2}} - \frac{2(b(5c^4C - 2Bc^3d - c^2(A - 11C)d^2 - 8Bcd^3 + 5Ad^4) + 3ad^2(2c(A - C)d - B(c^2 - d^2)))\,(a +}{3d^2\left(c^2 + d^2\right)^2 f\sqrt{c + d\tan(e + fx)}} + \frac{b(b(5c^4C - 2Bc^3d + 10c^2Cd^2 - 6Bcd^3 + (4A + C)d^4) + 2ad^2(2c(A - C)d - B(c^2 - d^2)))\sqrt{a}}{d^3\left(c^2 + d^2\right)^2 f} - \frac{((ia + b)^3(A - iB - C))\,\text{Subst}\left(\int \frac{1}{-a + ib - (-c + id)x^2}\,dx, x, \frac{\sqrt{a + b\tan(e + fx)}}{\sqrt{c + d\tan(e + fx)}}\right)}{(c - id)^2 f} + \frac{(d^3(3a^2b(c^2C - 2Bcd - Cd^2 - A(c^2 - d^2)) - b^3(c^2C - 2Bcd - Cd^2 - A(c^2 - d^2)) + a^3(2c(A -}{}$$

$$= -\frac{(a-ib)^{5/2}(B+i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(c-id)^{5/2}f}$$
$$-\frac{(a+ib)^{5/2}(B-i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(c+id)^{5/2}f}$$
$$-\frac{b^{3/2}(5bcC-2bBd-5aCd)\operatorname{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{d^{7/2}f}$$
$$-\frac{2(c^2C-Bcd+Ad^2)(a+b\tan(e+fx))^{5/2}}{3d(c^2+d^2)f(c+d\tan(e+fx))^{3/2}}$$
$$-\frac{2(b(5c^4C-2Bc^3d-c^2(A-11C)d^2-8Bcd^3+5Ad^4)+3ad^2(2c(A-C)d-B(c^2-d^2)))(a+b}{3d^2(c^2+d^2)^2f\sqrt{c+d\tan(e+fx)}}$$
$$+\frac{b(b(5c^4C-2Bc^3d+10c^2Cd^2-6Bcd^3+(4A+C)d^4)+2ad^2(2c(A-C)d-B(c^2-d^2)))\sqrt{a+}}{d^3(c^2+d^2)^2f}$$

Mathematica [B] (verified)

Both result and optimal contain complex but leaf count is larger than twice the leaf count of optimal. 2650 vs. 2(549) = 1098.

Time = 9.79 (sec) , antiderivative size = 2650, normalized size of antiderivative = 4.83

$$\int \frac{(a+b\tan(e+fx))^{5/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx = \text{Result too large to show}$$

```
[In] Integrate[((a + b*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/(c + d*Tan[e + f*x])^(5/2),x]
```

```
[Out] (C*(a + b*Tan[e + f*x])^(5/2))/(d*f*(c + d*Tan[e + f*x])^(3/2)) + ((2*(I*a
+ b)*(A - I*B - C)*d*(a + b*Tan[e + f*x])^(3/2))/(3*(c - I*d)*f*(c + d*Tan[
e + f*x])^(3/2)) - (2*(I*a - b)*(A + I*B - C)*d*(a + b*Tan[e + f*x])^(3/2))
/(3*(c + I*d)*f*(c + d*Tan[e + f*x])^(3/2)) + (2*(a - I*b)^2*(I*A + B - I*C
)*d*((Sqrt[-a + I*b]*ArcTanh[(Sqrt[-c + I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqr
t[-a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(-c + I*d)^(3/2) + Sqrt[a + b*Tan[e
 + f*x]]/((c - I*d)*Sqrt[c + d*Tan[e + f*x]])))/((c - I*d)*f) + (2*(a + I*b
)^2*(I*A - B - I*C)*d*((Sqrt[a + I*b]*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan
[e + f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(c + I*d)^(3/2) - Sq
rt[a + b*Tan[e + f*x]]/((c + I*d)*Sqrt[c + d*Tan[e + f*x]])))/((c + I*d)*f)
 + (10*c*C*(b*c - a*d)*(b/((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)))^(5/2
)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d))^3*Sqrt[(b*(c + d*Tan[e + f*x]
))/(b*c - a*d)]*(-1 - (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/(b*c
 - a*d) - (a*b*d)/(b*c - a*d))))^2*((b^2*d^2*(a + b*Tan[e + f*x])^2)/(3*(b*
c - a*d)^2*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d))^2*(-1 - (b*d*(a + b*
Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d))))^2
```

```
) - (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/
(b*c - a*d))*(-1 - (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/(b*c -
a*d) - (a*b*d)/(b*c - a*d))))) - (Sqrt[b]*Sqrt[d]*ArcSinh[(Sqrt[b]*Sqrt[d]*
Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b*c - a*d]*Sqrt[(b^2*c)/(b*c - a*d) - (a*b*
d)/(b*c - a*d)])]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b*c - a*d]*Sqrt[(b^2*c)/(
b*c - a*d) - (a*b*d)/(b*c - a*d)]*Sqrt[1 + (b*d*(a + b*Tan[e + f*x]))/((b*c
 - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)))])))/(b*d^3*f*Sqrt[a +
b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]]*(1 + (b*d*(a + b*Tan[e + f*x]))/((
b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d))))^(3/2)) - (4*B*(b*c
 - a*d)*(b/((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)))^(5/2)*((b^2*c)/(b*c
 - a*d) - (a*b*d)/(b*c - a*d))^3*Sqrt[(b*(c + d*Tan[e + f*x]))/(b*c - a*d)]
*(-1 - (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b*
d)/(b*c - a*d))))^2*((b^2*d^2*(a + b*Tan[e + f*x])^2)/(3*(b*c - a*d)^2*((b^
2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d))^2*(-1 - (b*d*(a + b*Tan[e + f*x]))/
((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d))))^2) - (b*d*(a + b
*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d))*(-
1 - (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/
(b*c - a*d))))) - (Sqrt[b]*Sqrt[d]*ArcSinh[(Sqrt[b]*Sqrt[d]*Sqrt[a + b*Tan[
e + f*x]])/(Sqrt[b*c - a*d]*Sqrt[(b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)]
)]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b*c - a*d]*Sqrt[(b^2*c)/(b*c - a*d) - (a
*b*d)/(b*c - a*d)]*Sqrt[1 + (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c
)/(b*c - a*d) - (a*b*d)/(b*c - a*d)))])))/(b*d^2*f*Sqrt[a + b*Tan[e + f*x]]
*Sqrt[c + d*Tan[e + f*x]]*(1 + (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^
2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d))))^(3/2)) - (10*a*C*(b*c - a*d)*(b/(
(b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)))^(5/2)*((b^2*c)/(b*c - a*d) - (a
*b*d)/(b*c - a*d))^3*Sqrt[(b*(c + d*Tan[e + f*x]))/(b*c - a*d)]*(-1 - (b*d*
(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*
d))))^2*((b^2*d^2*(a + b*Tan[e + f*x])^2)/(3*(b*c - a*d)^2*((b^2*c)/(b*c -
a*d) - (a*b*d)/(b*c - a*d))^2*(-1 - (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)
*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d))))^2) - (b*d*(a + b*Tan[e + f*x
]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d))*(-1 - (b*d*(a
+ b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d))
))) - (Sqrt[b]*Sqrt[d]*ArcSinh[(Sqrt[b]*Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(
Sqrt[b*c - a*d]*Sqrt[(b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)])]*Sqrt[a +
b*Tan[e + f*x]])/(Sqrt[b*c - a*d]*Sqrt[(b^2*c)/(b*c - a*d) - (a*b*d)/(b*c -
 a*d)]*Sqrt[1 + (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d
) - (a*b*d)/(b*c - a*d)))])))/(b^2*d^2*f*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c +
d*Tan[e + f*x]]*(1 + (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/(b*c
- a*d) - (a*b*d)/(b*c - a*d))))^(3/2)))/(2*d)
```

Maple [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(fx+e))^{\frac{5}{2}}\left(A+B\tan(fx+e)+C\tan(fx+e)^2\right)}{(c+d\tan(fx+e))^{\frac{5}{2}}}dx$$

```
[In] int((a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(5/2),x)
```

```
[Out] int((a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(5/2),x)
```

Fricas [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{5/2}}\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(5/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int \frac{(a+b\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{5/2}}\,dx = \int \frac{(a+b\tan(e+fx))^{\frac{5}{2}}\left(A+B\tan(e\right.}{(c+d\tan(e+}$$

```
[In] integrate((a+b*tan(f*x+e))**(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan
(f*x+e))**(5/2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)**
2)/(c + d*tan(e + f*x))**(5/2), x)
```

Maxima [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{5/2}}\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(5/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

Giac [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^{5/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx=\text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(5/2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^{5/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx=\int \frac{(a+b\tan(e+f\,x))^{5/2}\,(C\tan(e+}{(c+d\tan(e+}$$

```
[In] int(((a + b*tan(e + f*x))^(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c
 + d*tan(e + f*x))^(5/2),x)
```

```
[Out] int(((a + b*tan(e + f*x))^(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c
 + d*tan(e + f*x))^(5/2), x)
```

3.160 $\int \frac{(a+b\tan(e+fx))^{3/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx$

Optimal result 2124
Rubi [A] (verified) 2125
Mathematica [B] (verified) 2129
Maple [F(-1)] 2130
Fricas [F(-1)] 2130
Sympy [F] 2131
Maxima [F(-1)] 2131
Giac [F(-1)] 2131
Mupad [F(-1)] 2131

Optimal result

Integrand size = 49, antiderivative size = 407

$$\int \frac{(a+b\tan(e+fx))^{3/2}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx = -\frac{(a-ib)^{3/2}(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(c-id)^{5/2}f} - \frac{(a+ib)^{3/2}(B-i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(c+id)^{5/2}f} + \frac{2b^{3/2}C\operatorname{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{d^{5/2}f} - \frac{2(c^2C-Bcd+Ad^2)(a+b\tan(e+fx))^{3/2}}{3d(c^2+d^2)f(c+d\tan(e+fx))^{3/2}} - \frac{2(b(c^4C-c^2(A-3C)d^2-2Bcd^3+Ad^4)+ad^2(2c(A-C)d-B(c^2-d^2)))\sqrt{a+b\tan(e+fx)}}{d^2(c^2+d^2)^2f\sqrt{c+d\tan(e+fx)}}$$

```
[Out] -(a-I*b)^(3/2)*(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-
I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/(c-I*d)^(5/2)/f-(a+I*b)^(3/2)*(B-I*(A-C)
)*arctanh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e
))^(1/2))/(c+I*d)^(5/2)/f+2*b^(3/2)*C*arctanh(d^(1/2)*(a+b*tan(f*x+e))^(1/2
)/b^(1/2)/(c+d*tan(f*x+e))^(1/2))/d^(5/2)/f-2*(b*(c^4*C-c^2*(A-3*C)*d^2-2*B
*c*d^3+A*d^4)+a*d^2*(2*c*(A-C)*d-B*(c^2-d^2)))*(a+b*tan(f*x+e))^(1/2)/d^2/(
c^2+d^2)^2/f/(c+d*tan(f*x+e))^(1/2)-2/3*(A*d^2-B*c*d+C*c^2)*(a+b*tan(f*x+e)
)^(3/2)/d/(c^2+d^2)/f/(c+d*tan(f*x+e))^(3/2)
```

Rubi [A] (verified)

Time = 10.00 (sec) , antiderivative size = 407, normalized size of antiderivative = 1.00, number of steps used = 14, number of rules used = 8, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.163, Rules used = {3726, 3736, 6857, 65, 223, 212, 95, 214}

$$\int \frac{(a+b\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{5/2}}\,dx = -\frac{(a-ib)^{3/2}(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(c-id)^{5/2}} - \frac{(a+ib)^{3/2}(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(c+id)^{5/2}} - \frac{2(Ad^2-Bcd+c^2C)\left(a+b\tan(e+fx)\right)^{3/2}}{3df\left(c^2+d^2\right)(c+d\tan(e+fx))^{3/2}} - \frac{2\sqrt{a+b\tan(e+fx)}(ad^2(2cd(A-C)-B(c^2-d^2))+b(-c^2d^2(A-3C)+Ad^4-2Bcd^3+c^4C))}{d^2f\left(c^2+d^2\right)^2\sqrt{c+d\tan(e+fx)}} + \frac{2b^{3/2}C\text{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{d^{5/2}f}$$

```
[In] Int[((a + b*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c
 + d*Tan[e + f*x])^(5/2),x]
```

```
[Out] -(((a - I*b)^(3/2)*(I*A + B - I*C)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e
+ f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((c - I*d)^(5/2)*f)) -
((a + I*b)^(3/2)*(B - I*(A - C))*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/((c + I*d)^(5/2)*f) + (2*
b^(3/2)*C*ArcTanh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c + d*Ta
n[e + f*x]])])/(d^(5/2)*f) - (2*(c^2*C - B*c*d + A*d^2)*(a + b*Tan[e + f*x]
)^(3/2))/(3*d*(c^2 + d^2)*f*(c + d*Tan[e + f*x])^(3/2)) - (2*(b*(c^4*C - c^
2*(A - 3*C)*d^2 - 2*B*c*d^3 + A*d^4) + a*d^2*(2*c*(A - C)*d - B*(c^2 - d^2)
))*Sqrt[a + b*Tan[e + f*x]])/(d^2*(c^2 + d^2)^2*f*Sqrt[c + d*Tan[e + f*x]])
```

Rule 65

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - a*(d/b) +
d*(x^p/b))^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 95

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
```

```
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 212

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1/(Rt[a, 2]*Rt[-b, 2]))*
ArcTanh[Rt[-b, 2]*(x/Rt[a, 2])], x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 223

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3726

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*d^2 + c*(c*C - B*d))*(a + b*Tan[e +
 f*x])^m*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3736

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[(a + b*ff*x)^m*(c + d*ff*x)^n*((A + B*ff*x + C*ff^2*x^2
)/(1 + ff^2*x^2)), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6857

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
```

[n, 0]

Rubi steps

$$\text{integral} = -\frac{2(c^2C - Bcd + Ad^2)\,(a + b\tan(e+fx))^{3/2}}{3d\,(c^2+d^2)\,f(c+d\tan(e+fx))^{3/2}} + \frac{2\int \frac{\sqrt{a+b\tan(e+fx)}\left(\frac{3}{2}(Ad(ac+bd)+(bc-ad)(cC-Bd))+\frac{3}{2}d((A-C)(bc-ad)+B(ac+bd))\tan(e+fx)+\frac{3}{2}bC(c^2+d^2)\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{3/2}}\,dx}{3d\,(c^2+d^2)}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)\,(a + b\tan(e+fx))^{3/2}}{3d\,(c^2+d^2)\,f(c+d\tan(e+fx))^{3/2}}$$
$$- \frac{2(b(c^4C - c^2(A-3C)d^2 - 2Bcd^3 + Ad^4) + ad^2(2c(A-C)d - B(c^2-d^2)))\sqrt{a+b\tan(e+fx}}{d^2\,(c^2+d^2)^2\,f\sqrt{c+d\tan(e+fx)}}$$
$$+ \frac{4\int \frac{\frac{3}{4}\left(d(ac+bd)\left(ad(Ac-cC+Bd)+b\left(c^2C-Bcd+Ad^2\right)\right)-(bc-ad)\left(ad^2(Bc-(A-C)d)-b\left(c^3C-c(A-2C)d^2-Bd^3\right)\right)\right)+\frac{3}{4}d^2((ac+b}{\sqrt{a+b\tan(e+fx)}\sqrt{c-}}}{3d^2\,(c^2+d}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)\,(a + b\tan(e+fx))^{3/2}}{3d\,(c^2+d^2)\,f(c+d\tan(e+fx))^{3/2}}$$
$$- \frac{2(b(c^4C - c^2(A-3C)d^2 - 2Bcd^3 + Ad^4) + ad^2(2c(A-C)d - B(c^2-d^2)))\sqrt{a+b\tan(e+fx}}{d^2\,(c^2+d^2)^2\,f\sqrt{c+d\tan(e+fx)}}$$
$$+ \frac{4\text{Subst}\Big(\int \frac{\frac{3}{4}\left(d(ac+bd)\left(ad(Ac-cC+Bd)+b\left(c^2C-Bcd+Ad^2\right)\right)-(bc-ad)\left(ad^2(Bc-(A-C)d)-b\left(c^3C-c(A-2C)d^2-Bd^3\right)\right)\right)+\frac{3}{4}}{\sqrt{a+bx}\sqrt{c+dx}(}}{3d^2\,(c^2}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)\,(a + b\tan(e+fx))^{3/2}}{3d\,(c^2+d^2)\,f(c+d\tan(e+fx))^{3/2}}$$
$$- \frac{2(b(c^4C - c^2(A-3C)d^2 - 2Bcd^3 + Ad^4) + ad^2(2c(A-C)d - B(c^2-d^2)))\sqrt{a+b\tan(e+fx}}{d^2\,(c^2+d^2)^2\,f\sqrt{c+d\tan(e+fx)}}$$
$$+ \frac{4\text{Subst}\Big(\int \Big(\frac{3b^2C\left(c^2+d^2\right)^2}{4\sqrt{a+bx}\sqrt{c+dx}} + \frac{3\left(-d^2\left(a^2\left(c^2C-2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-b^2\left(c^2C-2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-2ab(2c(A-C)d}{}}{3d}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)\,(a + b\tan(e+fx))^{3/2}}{3d\,(c^2+d^2)\,f(c+d\tan(e+fx))^{3/2}}$$
$$- \frac{2(b(c^4C - c^2(A-3C)d^2 - 2Bcd^3 + Ad^4) + ad^2(2c(A-C)d - B(c^2-d^2)))\sqrt{a+b\tan(e+fx}}{d^2\,(c^2+d^2)^2\,f\sqrt{c+d\tan(e+fx)}}$$
$$+ \frac{(b^2C)\,\text{Subst}\left(\int \frac{1}{\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e+fx)\right)}{d^2f}$$
$$+ \frac{\text{Subst}\Big(\int \frac{-d^2\left(a^2\left(c^2C-2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-b^2\left(c^2C-2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-2ab\left(2c(A-C)d-B\left(c^2-d^2\right)\right)\right)-d^2\left(2ab(c^2}{\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}}{d^2\,(c^2+d^2)^2}$$

$$=-\frac{2(c^2C-Bcd+Ad^2)\,(a+b\tan(e+fx))^{3/2}}{3d\,(c^2+d^2)\,f(c+d\tan(e+fx))^{3/2}}$$
$$-\frac{2(b(c^4C-c^2(A-3C)d^2-2Bcd^3+Ad^4)+ad^2(2c(A-C)d-B(c^2-d^2)))\,\sqrt{a+b\tan(e+fx)}}{d^2\,(c^2+d^2)^2\,f\sqrt{c+d\tan(e+fx)}}$$
$$+\frac{(2bC)\text{Subst}\left(\int\frac{1}{\sqrt{c-\frac{ad}{b}+\frac{dx^2}{b}}}\,dx,x,\sqrt{a+b\tan(e+fx)}\right)}{d^2f}$$
$$+\frac{\text{Subst}\left(\int\left(\frac{-id^2(a^2(c^2C-2Bcd-Cd^2-A(c^2-d^2))-b^2(c^2C-2Bcd-Cd^2-A(c^2-d^2))-2ab(2c(A-C)d-B(c^2-d^2)))+d^2(2ab(c}{2(i-x)\sqrt{a+bx}\sqrt{c+dx}}\right.\right.}{}$$

$$=-\frac{2(c^2C-Bcd+Ad^2)\,(a+b\tan(e+fx))^{3/2}}{3d\,(c^2+d^2)\,f(c+d\tan(e+fx))^{3/2}}$$
$$-\frac{2(b(c^4C-c^2(A-3C)d^2-2Bcd^3+Ad^4)+ad^2(2c(A-C)d-B(c^2-d^2)))\,\sqrt{a+b\tan(e+fx)}}{d^2\,(c^2+d^2)^2\,f\sqrt{c+d\tan(e+fx)}}$$
$$+\frac{((a-ib)^2(iA+B-iC))\,\text{Subst}\left(\int\frac{1}{(i+x)\sqrt{a+bx}\sqrt{c+dx}}\,dx,x,\tan(e+fx)\right)}{2(c-id)^2f}$$
$$+\frac{(2bC)\text{Subst}\left(\int\frac{1}{1-\frac{dx^2}{b}}\,dx,x,\frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{d^2f}$$
$$+\frac{(-id^2(a^2(c^2C-2Bcd-Cd^2-A(c^2-d^2))-b^2(c^2C-2Bcd-Cd^2-A(c^2-d^2))-2ab(2c(A-}{}$$

$$=\frac{2b^{3/2}C\text{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{d^{5/2}f}-\frac{2(c^2C-Bcd+Ad^2)\,(a+b\tan(e+fx))^{3/2}}{3d\,(c^2+d^2)\,f(c+d\tan(e+fx))^{3/2}}$$
$$-\frac{2(b(c^4C-c^2(A-3C)d^2-2Bcd^3+Ad^4)+ad^2(2c(A-C)d-B(c^2-d^2)))\,\sqrt{a+b\tan(e+fx)}}{d^2\,(c^2+d^2)^2\,f\sqrt{c+d\tan(e+fx)}}$$
$$+\frac{((a-ib)^2(iA+B-iC))\,\text{Subst}\left(\int\frac{1}{-a+ib-(-c+id)x^2}\,dx,x,\frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{(c-id)^2f}$$
$$+\frac{(-id^2(a^2(c^2C-2Bcd-Cd^2-A(c^2-d^2))-b^2(c^2C-2Bcd-Cd^2-A(c^2-d^2))-2ab(2c(A-}{}$$

$$
= -\frac{(a-ib)^{3/2}(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(c-id)^{5/2}f}
- \frac{(a+ib)^{3/2}(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(c+id)^{5/2}f}
+ \frac{2b^{3/2}C\text{arctanh}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{d^{5/2}f} - \frac{2(c^2C-Bcd+Ad^2)\,(a+b\tan(e+fx))^{3/2}}{3d\,(c^2+d^2)\,f(c+d\tan(e+fx))^{3/2}}
- \frac{2(b(c^4C-c^2(A-3C)d^2-2Bcd^3+Ad^4)+ad^2(2c(A-C)d-B(c^2-d^2)))\,\sqrt{a+b\tan(e+f}}{d^2\,(c^2+d^2)^2\,f\sqrt{c+d\tan(e+fx)}}
$$

Mathematica [B] (verified)

Both result and optimal contain complex but leaf count is larger than twice the leaf count of optimal. 1135 vs. 2(407) = 814.

Time = 7.17 (sec) , antiderivative size = 1135, normalized size of antiderivative = 2.79

$$
\int \frac{(a+b\tan(e+fx))^{3/2}\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx = \frac{(B+i(A-C))(a+b\tan(e+fx))^{3/}}{3(c-id)f(c+d\tan(e+fx))^{3/2}}
- \frac{(iA-B-iC)(a+b\tan(e+fx))^{3/2}}{3(c+id)f(c+d\tan(e+fx))^{3/2}}
+ \frac{(ia+b)(A-iB-C)\left(\frac{\sqrt{-a+ib}\text{arctanh}\left(\frac{\sqrt{-c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{-a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(-c+id)^{3/2}} + \frac{\sqrt{a+b\tan(e+fx)}}{(c-id)\sqrt{c+d\tan(e+fx)}}\right)}{(c-id)f}
+ \frac{(ia-b)(A+iB-C)\left(\frac{\sqrt{a+ib}\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(c+id)^{3/2}} - \frac{\sqrt{a+b\tan(e+fx)}}{(c+id)\sqrt{c+d\tan(e+fx)}}\right)}{(c+id)f}
- \frac{2C(bc-ad)\left(\frac{b}{\frac{b^2c}{bc-ad}-\frac{abd}{bc-ad}}\right)^{5/2}\left(\frac{b^2c}{bc-ad}-\frac{abd}{bc-ad}\right)^3\sqrt{\frac{b(c+d\tan(e+fx))}{bc-ad}}\left(-1-\frac{bd(a+b\tan(e+fx))}{(bc-ad)\left(\frac{b^2c}{bc-ad}-\frac{abd}{bc-ad}\right)}\right)^2\left(\frac{\quad}{3(bc-ad)^2\left(\cdots\right.}\right.}{b^2d^3f\sqrt{a+b\tan(e+f}}
$$

```
[In] Integrate[((a + b*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/(c + d*Tan[e + f*x])^(5/2),x]
```

```
[Out] ((B + I*(A - C))*(a + b*Tan[e + f*x])^(3/2))/(3*(c - I*d)*f*(c + d*Tan[e +
f*x])^(3/2)) - ((I*A - B - I*C)*(a + b*Tan[e + f*x])^(3/2))/(3*(c + I*d)*f*
(c + d*Tan[e + f*x])^(3/2)) + ((I*a + b)*(A - I*B - C)*((Sqrt[-a + I*b]*Arc
Tanh[(Sqrt[-c + I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[-a + I*b]*Sqrt[c + d*T
```

```
an[e + f*x]])])/(-c + I*d)^(3/2) + Sqrt[a + b*Tan[e + f*x]]/((c - I*d)*Sqrt
[c + d*Tan[e + f*x]])))/((c - I*d)*f) + ((I*a - b)*(A + I*B - C)*((Sqrt[a +
 I*b]*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + I*b]*Sqrt[
c + d*Tan[e + f*x]])])/(c + I*d)^(3/2) - Sqrt[a + b*Tan[e + f*x]]/((c + I*d
)*Sqrt[c + d*Tan[e + f*x]])))/((c + I*d)*f) - (2*C*(b*c - a*d)*(b/((b^2*c)/
(b*c - a*d) - (a*b*d)/(b*c - a*d)))^(5/2)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b
*c - a*d))^3*Sqrt[(b*(c + d*Tan[e + f*x]))/(b*c - a*d)]*(-1 - (b*d*(a + b*T
an[e + f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d))))^2*
((b^2*d^2*(a + b*Tan[e + f*x])^2)/(3*(b*c - a*d)^2*((b^2*c)/(b*c - a*d) - (
a*b*d)/(b*c - a*d))^2*(-1 - (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c
)/(b*c - a*d) - (a*b*d)/(b*c - a*d))))^2) - (b*d*(a + b*Tan[e + f*x]))/((b*
c - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d))*(-1 - (b*d*(a + b*Tan[
e + f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d))))) - (S
qrt[b]*Sqrt[d]*ArcSinh[(Sqrt[b]*Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b*c
 - a*d]*Sqrt[(b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)])]*Sqrt[a + b*Tan[e
+ f*x]])/(Sqrt[b*c - a*d]*Sqrt[(b^2*c)/(b*c - a*d) - (a*b*d)/(b*c - a*d)]*S
qrt[1 + (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) - (a*b
*d)/(b*c - a*d)))])))/(b^2*d^3*f*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e
+ f*x]]*(1 + (b*d*(a + b*Tan[e + f*x]))/((b*c - a*d)*((b^2*c)/(b*c - a*d) -
 (a*b*d)/(b*c - a*d))))^(3/2))
```

Maple [F(-1)]

Timed out.

hanged

```
[In] int((a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(5/2),x)
```

```
[Out] int((a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(5/2),x)
```

Fricas [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{5/2}}\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(5/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int \frac{(a+b\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{5/2}}\,dx = \int \frac{(a+b\tan(e+fx))^{\frac{3}{2}}\left(A+B\tan(\ldots\right.}{(c+d\tan(e+\ldots}$$

```
[In] integrate((a+b*tan(f*x+e))**(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan
(f*x+e))**(5/2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)**
2)/(c + d*tan(e + f*x))**(5/2), x)
```

Maxima [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{5/2}}\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(5/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

Giac [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{5/2}}\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(5/2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{5/2}}\,dx = \int \frac{(a+b\tan(e+f\,x))^{3/2}\left(C\tan(e+\ldots\right.}{(c+d\tan(e+\ldots}$$

```
[In] int(((a + b*tan(e + f*x))^(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c
 + d*tan(e + f*x))^(5/2),x)
```

```
[Out] int(((a + b*tan(e + f*x))^(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c
 + d*tan(e + f*x))^(5/2), x)
```

3.161 $\int \frac{\sqrt{a+b\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx$

Optimal result 2132
Rubi [A] (verified) 2133
Mathematica [A] (verified) 2136
Maple [F(-1)] 2137
Fricas [F(-1)] 2137
Sympy [F] 2137
Maxima [F(-2)] 2138
Giac [F(-1)] 2138
Mupad [F(-1)] 2138

Optimal result

Integrand size = 49, antiderivative size = 373

$$\int \frac{\sqrt{a+b\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx = -\frac{\sqrt{a-ib}(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(c-id)^{5/2}f} - \frac{\sqrt{a+ib}(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(c+id)^{5/2}f} - \frac{2(c^2C-Bcd+Ad^2)\sqrt{a+b\tan(e+fx)}}{3d(c^2+d^2)f(c+d\tan(e+fx))^{3/2}} + \frac{2(b(c^4C+2Bc^3d-c^2(5A-7C)d^2-4Bcd^3+Ad^4)+3ad^2(2c(A-C)d-B(c^2-d^2)))\sqrt{a+b\tan(e+fx)}}{3d(bc-ad)(c^2+d^2)^2f\sqrt{c+d\tan(e+fx)}}$$

```
[Out] -(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(1/2)/(c+
d*tan(f*x+e))^(1/2))*(a-I*b)^(1/2)/(c-I*d)^(5/2)/f-(B-I*(A-C))*arctanh((c+I
*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))*(a+I
*b)^(1/2)/(c+I*d)^(5/2)/f+2/3*(b*(c^4*C+2*B*c^3*d-c^2*(5*A-7*C)*d^2-4*B*c*d
^3+A*d^4)+3*a*d^2*(2*c*(A-C)*d-B*(c^2-d^2)))*(a+b*tan(f*x+e))^(1/2)/d/(-a*d
+b*c)/(c^2+d^2)^2/f/(c+d*tan(f*x+e))^(1/2)-2/3*(A*d^2-B*c*d+C*c^2)*(a+b*tan
(f*x+e))^(1/2)/d/(c^2+d^2)/f/(c+d*tan(f*x+e))^(3/2)
```

Rubi [A] (verified)

Time = 2.24 (sec) , antiderivative size = 373, normalized size of antiderivative = 1.00, number of steps used = 9, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.122, Rules used = {3726, 3730, 3697, 3696, 95, 214}

$$\int \frac{\sqrt{a+b\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx = -\frac{\sqrt{a-ib}(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(c-id)^{5/2}} - \frac{\sqrt{a+ib}(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(c+id)^{5/2}} - \frac{2(Ad^2-Bcd+c^2C)\sqrt{a+b\tan(e+fx)}}{3df(c^2+d^2)(c+d\tan(e+fx))^{3/2}} + \frac{2\sqrt{a+b\tan(e+fx)}(3ad^2(2cd(A-C)-B(c^2-d^2))+b(-c^2d^2(5A-7C)+Ad^4+2Bc^3d-4Bcd^3+c^4}{3df(c^2+d^2)^2(bc-ad)\sqrt{c+d\tan(e+fx)}}$$

```
[In] Int[(Sqrt[a + b*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c +
 d*Tan[e + f*x])^(5/2),x]
```

```
[Out] -((Sqrt[a - I*b]*(I*A + B - I*C)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((c - I*d)^(5/2)*f)) - (S
qrt[a + I*b]*(B - I*(A - C))*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]
])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/((c + I*d)^(5/2)*f) - (2*(c^2
*C - B*c*d + A*d^2)*Sqrt[a + b*Tan[e + f*x]])/(3*d*(c^2 + d^2)*f*(c + d*Tan
[e + f*x])^(3/2)) + (2*(b*(c^4*C + 2*B*c^3*d - c^2*(5*A - 7*C)*d^2 - 4*B*c*
d^3 + A*d^4) + 3*a*d^2*(2*c*(A - C)*d - B*(c^2 - d^2)))*Sqrt[a + b*Tan[e +
f*x]])/(3*d*(b*c - a*d)*(c^2 + d^2)^2*f*Sqrt[c + d*Tan[e + f*x]])
```

Rule 95

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3696

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[A^2/f, Subst[Int[(a + b*x)^m*((c + d*x)^n/(A - B*x)), x], x, Tan[e + f*x
]], x] /; FreeQ[{a, b, c, d, e, f, A, B, m, n}, x] && NeQ[b*c - a*d, 0] &&
NeQ[a^2 + b^2, 0] && EqQ[A^2 + B^2, 0]
```

Rule 3697

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[(A + I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^n*(1 - I*Ta
n[e + f*x]), x], x] + Dist[(A - I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*T
an[e + f*x])^n*(1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A,
B, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[A^2 + B^2, 0]
```

Rule 3726

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*d^2 + c*(c*C - B*d))*(a + b*Tan[e +
 f*x])^m*((c + d*Tan[e + f*x])^(n + 1)/(d*f*(n + 1)*(c^2 + d^2))), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3730

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 - a*(b*B - a*C))*(a + b*Tan[e +
 f*x])^(m + 1)*((c + d*Tan[e + f*x])^(n + 1)/(f*(m + 1)*(b*c - a*d)*(a^2 +
b^2))), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rubi steps

$$\text{integral} = -\frac{2(c^2C - Bcd + Ad^2)\sqrt{a+b\tan(e+fx)}}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}} + \frac{2\int \frac{\frac{1}{2}(Ad(3ac+bd)+(bc-3ad)(cC-Bd))+\frac{3}{2}d((A-C)(bc-ad)+B(ac+bd))\tan(e+fx)+\frac{1}{2}b\left(c^2C+2Bcd-(2A-3C)d^2\right)\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}\,dx}{3d\left(c^2+d^2\right)}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)\sqrt{a+b\tan(e+fx)}}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}} + \frac{2(b(c^4C+2Bc^3d-c^2(5A-7C)d^2-4Bcd^3+Ad^4)+3ad^2(2c(A-C)d-B(c^2-d^2)))\sqrt{a+}}{3d(bc-ad)\left(c^2+d^2\right)^2 f\sqrt{c+d\tan(e+fx)}} + \frac{4\int \frac{-\frac{3}{4}d(bc-ad)\left(a\left(c^2C-2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-b\left(2c(A-C)d-B\left(c^2-d^2\right)\right)\right)-\frac{3}{4}d(bc-ad)\left(2aAcd-2acCd-Ab\left(c^2-d^2\right)-aB\left(c^2\right.\right.}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}}{3d(bc-ad)\left(c^2+d^2\right)^2}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)\sqrt{a+b\tan(e+fx)}}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}} + \frac{2(b(c^4C+2Bc^3d-c^2(5A-7C)d^2-4Bcd^3+Ad^4)+3ad^2(2c(A-C)d-B(c^2-d^2)))\sqrt{a+}}{3d(bc-ad)\left(c^2+d^2\right)^2 f\sqrt{c+d\tan(e+fx)}} + \frac{((a-ib)(A-iB-C))\int \frac{1+i\tan(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx}{2(c-id)^2} + \frac{((a+ib)(A+iB-C))\int \frac{1-i\tan(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx}{2(c+id)^2}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)\sqrt{a+b\tan(e+fx)}}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}} + \frac{2(b(c^4C+2Bc^3d-c^2(5A-7C)d^2-4Bcd^3+Ad^4)+3ad^2(2c(A-C)d-B(c^2-d^2)))\sqrt{a+}}{3d(bc-ad)\left(c^2+d^2\right)^2 f\sqrt{c+d\tan(e+fx)}} + \frac{((a-ib)(A-iB-C))\text{Subst}\left(\int \frac{1}{(1-ix)\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e+fx)\right)}{2(c-id)^2 f} + \frac{((a+ib)(A+iB-C))\text{Subst}\left(\int \frac{1}{(1+ix)\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e+fx)\right)}{2(c+id)^2 f}$$

$$= -\frac{2(c^2C - Bcd + Ad^2)\sqrt{a + b\tan(e + fx)}}{3d(c^2 + d^2) f(c + d\tan(e + fx))^{3/2}}$$
$$+ \frac{2(b(c^4C + 2Bc^3d - c^2(5A - 7C)d^2 - 4Bcd^3 + Ad^4) + 3ad^2(2c(A - C)d - B(c^2 - d^2)))\sqrt{a + b}}{3d(bc - ad)(c^2 + d^2)^2 f\sqrt{c + d\tan(e + fx)}}$$
$$+ \frac{((a - ib)(A - iB - C))\text{Subst}\left(\int \frac{1}{ia+b-(ic+d)x^2}\,dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{(c - id)^2 f}$$
$$+ \frac{((a + ib)(A + iB - C))\text{Subst}\left(\int \frac{1}{-ia+b-(-ic+d)x^2}\,dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{(c + id)^2 f}$$
$$= -\frac{\sqrt{a - ib}(iA + B - iC)\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(c - id)^{5/2} f}$$
$$- \frac{\sqrt{a + ib}(B - i(A - C))\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(c + id)^{5/2} f}$$
$$- \frac{2(c^2C - Bcd + Ad^2)\sqrt{a + b\tan(e + fx)}}{3d(c^2 + d^2) f(c + d\tan(e + fx))^{3/2}}$$
$$+ \frac{2(b(c^4C + 2Bc^3d - c^2(5A - 7C)d^2 - 4Bcd^3 + Ad^4) + 3ad^2(2c(A - C)d - B(c^2 - d^2)))\sqrt{a + b}}{3d(bc - ad)(c^2 + d^2)^2 f\sqrt{c + d\tan(e + fx)}}$$

Mathematica [A] (verified)

Time = 7.11 (sec) , antiderivative size = 609, normalized size of antiderivative = 1.63

$$\int \frac{\sqrt{a + b\tan(e + fx)}(A + B\tan(e + fx) + C\tan^2(e + fx))}{(c + d\tan(e + fx))^{5/2}}\,dx =$$
$$-\frac{C\sqrt{a + b\tan(e + fx)}}{df(c + d\tan(e + fx))^{3/2}}$$
$$-\frac{-\frac{2\left(\frac{1}{2}d^2(-bcC-a(2A-3C)d)-c\left(-((Ab+aB-bC)d^2)-\frac{1}{2}c(-bcC-2bBd+aCd)\right)\right)\sqrt{a+b\tan(e+fx)}}{3(-bc+ad)(c^2+d^2)f(c+d\tan(e+fx))^{3/2}} - \frac{2\left(-\frac{3d(bc-ad)^2\left(\frac{\sqrt{-a+ib}(iA+B-iC)(c+id)}{\cdots}\right.}{\cdots}\right.}{\cdots}}{\cdots}$$

```
[In] Integrate[(Sqrt[a + b*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)
)/(c + d*Tan[e + f*x])^(5/2),x]
```

```
[Out] -((C*Sqrt[a + b*Tan[e + f*x]])/(d*f*(c + d*Tan[e + f*x])^(3/2))) - ((-2*((d
^2*(-(b*c*C) - a*(2*A - 3*C)*d))/2 - c*(-((A*b + a*B - b*C)*d^2) - (c*(-(b*
c*C) - 2*b*B*d + a*C*d))/2))*Sqrt[a + b*Tan[e + f*x]])/(3*(-(b*c) + a*d)*(c
^2 + d^2)*f*(c + d*Tan[e + f*x])^(3/2)) - (2*((-3*d*(b*c - a*d)^2*((Sqrt[-a
```

```
 + I*b]*(I*A + B - I*C)*(c + I*d)^2*ArcTanh[(Sqrt[-c + I*d]*Sqrt[a + b*Tan[
e + f*x]])/(Sqrt[-a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/Sqrt[-c + I*d] + (Sq
rt[a + I*b]*(B - I*(A - C))*(c - I*d)^2*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*T
an[e + f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/Sqrt[c + I*d]))/(2
*(-(b*c) + a*d)*(c^2 + d^2)*f) - (2*(-1/2*(d^2*(b*c - a*d)*(3*a*d*(A*c - c*
C + B*d) + b*(c^2*C - B*c*d + A*d^2))) - c*((-3*d^2*(b*c - a*d)*(A*b*c + a*
B*c - b*c*C - a*A*d + b*B*d + a*C*d))/2 + (b*c*(b*c - a*d)*(c^2*C + 2*B*c*d
 - (2*A - 3*C)*d^2))/2))*Sqrt[a + b*Tan[e + f*x]])/((-(b*c) + a*d)*(c^2 + d
^2)*f*Sqrt[c + d*Tan[e + f*x]])))/(3*(-(b*c) + a*d)*(c^2 + d^2)))/d
```

Maple [F(-1)]

Timed out.

$$\int \frac{\sqrt{a+b\tan\left(fx+e\right)}\left(A+B\tan\left(fx+e\right)+C\tan\left(fx+e\right)^{2}\right)}{\left(c+d\tan\left(fx+e\right)\right)^{\frac{5}{2}}}dx$$

```
[In] int((a+b*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(5/2),x)
```

```
[Out] int((a+b*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(5/2),x)
```

Fricas [F(-1)]

Timed out.

$$\int \frac{\sqrt{a+b\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx=\text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(5/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int \frac{\sqrt{a+b\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx=\int \frac{\sqrt{a+b\tan\left(e+fx\right)}\left(A+B\tan\left(e+\right.\right.}{\left(c+d\tan\left(e+f\right.\right.}$$

```
[In] integrate((a+b*tan(f*x+e))**(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan
(f*x+e))**(5/2),x)
```

```
[Out] Integral(sqrt(a + b*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/
(c + d*tan(e + f*x))**(5/2), x)
```

Maxima [F(-2)]

Exception generated.

$$\int \frac{\sqrt{a+b\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx = \text{Exception raised: ValueError}$$

```
[In] integrate((a+b*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(5/2),x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(((2*b*d+2*a*c)^2>0)', see `assume?`
 for mo
```

Giac [F(-1)]

Timed out.

$$\int \frac{\sqrt{a+b\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(5/2),x, algorithm="giac")
```

```
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{\sqrt{a+b\tan(e+fx)}(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^{5/2}}\,dx = \int \frac{\sqrt{a+b\tan\left(e+f\,x\right)}\,\left(C\tan(e+f\,x)^2\right.}{\left(c+d\tan\left(e+f\,x\right.\right.}$$

```
[In] int(((a + b*tan(e + f*x))^(1/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c
 + d*tan(e + f*x))^(5/2),x)
```

```
[Out] int(((a + b*tan(e + f*x))^(1/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c
 + d*tan(e + f*x))^(5/2), x)
```

3.162 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}\,dx$

Optimal result 2139
Rubi [A] (verified) 2140
Mathematica [A] (verified) 2143
Maple [F(-1)] 2143
Fricas [F(-1)] 2143
Sympy [F] 2144
Maxima [F(-1)] 2144
Giac [F(-1)] 2144
Mupad [F(-1)] 2144

Optimal result

Integrand size = 49, antiderivative size = 379

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}\,dx =$$

$$-\frac{(B+i(A-C))\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{a-ib}(c-id)^{5/2}f}$$

$$+\frac{(iA-B-iC)\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{a+ib}(c+id)^{5/2}f}+\frac{2(c^2C-Bcd+Ad^2)\sqrt{a+b\tan(e+fx)}}{3(bc-ad)(c^2+d^2)f(c+d\tan(e+fx))^{3/2}}$$

$$+\frac{2(b(2c^4C-5Bc^3d+4c^2(2A-C)d^2+Bcd^3+2Ad^4)-3ad^2(2c(A-C)d-B(c^2-d^2)))\sqrt{a+b\tan(e+}}{3(bc-ad)^2(c^2+d^2)^2f\sqrt{c+d\tan(e+fx)}}$$

```
[Out] -(B+I*(A-C))*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(1/2)/(c+
d*tan(f*x+e))^(1/2))/(c-I*d)^(5/2)/f/(a-I*b)^(1/2)+(I*A-B-I*C)*arctanh((c+I
*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/(c+I
*d)^(5/2)/f/(a+I*b)^(1/2)+2/3*(b*(2*c^4*C-5*B*c^3*d+4*c^2*(2*A-C)*d^2+B*c*d
^3+2*A*d^4)-3*a*d^2*(2*c*(A-C)*d-B*(c^2-d^2)))*(a+b*tan(f*x+e))^(1/2)/(-a*d
+b*c)^2/(c^2+d^2)^2/f/(c+d*tan(f*x+e))^(1/2)+2/3*(A*d^2-B*c*d+C*c^2)*(a+b*t
an(f*x+e))^(1/2)/(-a*d+b*c)/(c^2+d^2)/f/(c+d*tan(f*x+e))^(3/2)
```

Rubi [A] (verified)

Time = 2.06 (sec) , antiderivative size = 379, normalized size of antiderivative = 1.00, number of steps used = 9, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}} = 0.102$, Rules used = {3730, 3697, 3696, 95, 214}

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}\,dx =$$
$$-\frac{(B+i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f\sqrt{a-ib}(c-id)^{5/2}}$$
$$+\frac{(iA-B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f\sqrt{a+ib}(c+id)^{5/2}}+\frac{2(Ad^2-Bcd+c^2C)\sqrt{a+b\tan(e+fx)}}{3f(c^2+d^2)(bc-ad)(c+d\tan(e+fx))^{3/2}}$$
$$+\frac{2\sqrt{a+b\tan(e+fx)}(b(4c^2d^2(2A-C)+2Ad^4-5Bc^3d+Bcd^3+2c^4C)-3ad^2(2cd(A-C)-B(c^2-d^2)}{3f(c^2+d^2)^2(bc-ad)^2\sqrt{c+d\tan(e+fx)}}$$

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/(Sqrt[a + b*Tan[e + f*x]]*(c +
d*Tan[e + f*x])^(5/2)),x]
```

```
[Out] -(((B + I*(A - C))*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a
 - I*b]*Sqrt[c + d*Tan[e + f*x]])]])/(Sqrt[a - I*b]*(c - I*d)^(5/2)*f)) + ((
I*A - B - I*C)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + I
*b]*Sqrt[c + d*Tan[e + f*x]])]])/(Sqrt[a + I*b]*(c + I*d)^(5/2)*f) + (2*(c^2
*C - B*c*d + A*d^2)*Sqrt[a + b*Tan[e + f*x]])/(3*(b*c - a*d)*(c^2 + d^2)*f*
(c + d*Tan[e + f*x])^(3/2)) + (2*(b*(2*c^4*C - 5*B*c^3*d + 4*c^2*(2*A - C)*
d^2 + B*c*d^3 + 2*A*d^4) - 3*a*d^2*(2*c*(A - C)*d - B*(c^2 - d^2)))*Sqrt[a
+ b*Tan[e + f*x]])/(3*(b*c - a*d)^2*(c^2 + d^2)^2*f*Sqrt[c + d*Tan[e + f*x]
])
```

Rule 95

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3696

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
```

```
st[A^2/f, Subst[Int[(a + b*x)^m*((c + d*x)^n/(A - B*x)), x], x, Tan[e + f*x
]], x] /; FreeQ[{a, b, c, d, e, f, A, B, m, n}, x] && NeQ[b*c - a*d, 0] &&
NeQ[a^2 + b^2, 0] && EqQ[A^2 + B^2, 0]
```

Rule 3697

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[(A + I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^n*(1 - I*Ta
n[e + f*x]), x], x] + Dist[(A - I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*T
an[e + f*x])^n*(1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A,
B, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[A^2 + B^2, 0]
```

Rule 3730

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 - a*(b*B - a*C))*(a + b*Tan[e +
 f*x])^(m + 1)*((c + d*Tan[e + f*x])^(n + 1)/(f*(m + 1)*(b*c - a*d)*(a^2 +
b^2))), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rubi steps

$$\text{integral} = \frac{2(c^2C - Bcd + Ad^2)\sqrt{a + b\tan(e + fx)}}{3(bc - ad)(c^2 + d^2)f(c + d\tan(e + fx))^{3/2}} + \frac{2\int \frac{\frac{1}{2}(2Abd^2+3Ac(bc-ad)-(bc-3ad)(cC-Bd))+\frac{3}{2}(bc-ad)(Bc-(A-C)d)\tan(e+fx)+b(c^2C-Bcd+Ad^2)\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}\,dx}{3(bc - ad)(c^2 + d^2)}$$

$$= \frac{2(c^2C - Bcd + Ad^2)\sqrt{a + b\tan(e + fx)}}{3(bc - ad)(c^2 + d^2)f(c + d\tan(e + fx))^{3/2}} + \frac{2(b(2c^4C - 5Bc^3d + 4c^2(2A - C)d^2 + Bcd^3 + 2Ad^4) - 3ad^2(2c(A - C)d - B(c^2 - d^2)))\sqrt{a +}}{3(bc - ad)^2(c^2 + d^2)^2 f\sqrt{c + d\tan(e + fx)}} + \frac{4\int \frac{-\frac{3}{4}(bc-ad)^2(c^2C-2Bcd-Cd^2-A(c^2-d^2))-\frac{3}{4}(bc-ad)^2(2c(A-C)d-B(c^2-d^2))\tan(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx}{3(bc - ad)^2(c^2 + d^2)^2}$$

$$= \frac{2(c^2C - Bcd + Ad^2)\sqrt{a + b\tan(e + fx)}}{3(bc - ad)(c^2 + d^2) f(c + d\tan(e + fx))^{3/2}} + \frac{2(b(2c^4C - 5Bc^3d + 4c^2(2A - C)d^2 + Bcd^3 + 2Ad^4) - 3ad^2(2c(A - C)d - B(c^2 - d^2)))\sqrt{a + b}}{3(bc - ad)^2 (c^2 + d^2)^2 f\sqrt{c + d\tan(e + fx)}} + \frac{(A - iB - C)\int \frac{1 + i\tan(e + fx)}{\sqrt{a + b\tan(e + fx)}\sqrt{c + d\tan(e + fx)}}dx}{2(c - id)^2} + \frac{(A + iB - C)\int \frac{1 - i\tan(e + fx)}{\sqrt{a + b\tan(e + fx)}\sqrt{c + d\tan(e + fx)}}dx}{2(c + id)^2}$$

$$= \frac{2(c^2C - Bcd + Ad^2)\sqrt{a + b\tan(e + fx)}}{3(bc - ad)(c^2 + d^2) f(c + d\tan(e + fx))^{3/2}} + \frac{2(b(2c^4C - 5Bc^3d + 4c^2(2A - C)d^2 + Bcd^3 + 2Ad^4) - 3ad^2(2c(A - C)d - B(c^2 - d^2)))\sqrt{a + b}}{3(bc - ad)^2 (c^2 + d^2)^2 f\sqrt{c + d\tan(e + fx)}} + \frac{(A - iB - C)\text{Subst}\left(\int \frac{1}{(1 - ix)\sqrt{a + bx}\sqrt{c + dx}}dx, x, \tan(e + fx)\right)}{2(c - id)^2 f} + \frac{(A + iB - C)\text{Subst}\left(\int \frac{1}{(1 + ix)\sqrt{a + bx}\sqrt{c + dx}}dx, x, \tan(e + fx)\right)}{2(c + id)^2 f}$$

$$= \frac{2(c^2C - Bcd + Ad^2)\sqrt{a + b\tan(e + fx)}}{3(bc - ad)(c^2 + d^2) f(c + d\tan(e + fx))^{3/2}} + \frac{2(b(2c^4C - 5Bc^3d + 4c^2(2A - C)d^2 + Bcd^3 + 2Ad^4) - 3ad^2(2c(A - C)d - B(c^2 - d^2)))\sqrt{a + b}}{3(bc - ad)^2 (c^2 + d^2)^2 f\sqrt{c + d\tan(e + fx)}} + \frac{(A - iB - C)\text{Subst}\left(\int \frac{1}{ia + b - (ic + d)x^2}dx, x, \frac{\sqrt{a + b\tan(e + fx)}}{\sqrt{c + d\tan(e + fx)}}\right)}{(c - id)^2 f} + \frac{(A + iB - C)\text{Subst}\left(\int \frac{1}{-ia + b - (-ic + d)x^2}dx, x, \frac{\sqrt{a + b\tan(e + fx)}}{\sqrt{c + d\tan(e + fx)}}\right)}{(c + id)^2 f}$$

$$= -\frac{(iA + B - iC)\text{arctanh}\left(\frac{\sqrt{c - id}\sqrt{a + b\tan(e + fx)}}{\sqrt{a - ib}\sqrt{c + d\tan(e + fx)}}\right)}{\sqrt{a - ib}(c - id)^{5/2} f} - \frac{(B - i(A - C))\text{arctanh}\left(\frac{\sqrt{c + id}\sqrt{a + b\tan(e + fx)}}{\sqrt{a + ib}\sqrt{c + d\tan(e + fx)}}\right)}{\sqrt{a + ib}(c + id)^{5/2} f} + \frac{2(c^2C - Bcd + Ad^2)\sqrt{a + b\tan(e + fx)}}{3(bc - ad)(c^2 + d^2) f(c + d\tan(e + fx))^{3/2}} + \frac{2(b(2c^4C - 5Bc^3d + 4c^2(2A - C)d^2 + Bcd^3 + 2Ad^4) - 3ad^2(2c(A - C)d - B(c^2 - d^2)))\sqrt{a + b}}{3(bc - ad)^2 (c^2 + d^2)^2 f\sqrt{c + d\tan(e + fx)}}$$

Mathematica [A] (verified)

Time = 6.00 (sec) , antiderivative size = 403, normalized size of antiderivative = 1.06

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}\,dx = \frac{3(bc-ad)^2\left(\frac{(iA+B-iC)(c+id)^2\operatorname{arctanh}\left(\frac{\sqrt{-c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{-a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{-a+ib}\sqrt{-c+id}}\right.}{}$$

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/(Sqrt[a + b*Tan[e + f*x]]
*(c + d*Tan[e + f*x])^(5/2)),x]
```

```
[Out] (3*(b*c - a*d)^2*(((I*A + B - I*C)*(c + I*d)^2*ArcTanh[(Sqrt[-c + I*d]*Sqrt
[a + b*Tan[e + f*x]])/(Sqrt[-a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[-a
+ I*b]*Sqrt[-c + I*d]) + (I*(A + I*B - C)*(c - I*d)^2*ArcTanh[(Sqrt[c + I*d
]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqr
t[a + I*b]*Sqrt[c + I*d])) + (2*(b*c - a*d)*(c^2 + d^2)*(c^2*C - B*c*d + A*
d^2)*Sqrt[a + b*Tan[e + f*x]])/(c + d*Tan[e + f*x])^(3/2) + (2*(b*(2*c^4*C
- 5*B*c^3*d + 4*c^2*(2*A - C)*d^2 + B*c*d^3 + 2*A*d^4) + 3*a*d^2*(2*c*(-A +
 C)*d + B*(c^2 - d^2)))*Sqrt[a + b*Tan[e + f*x]])/Sqrt[c + d*Tan[e + f*x]])
/(3*(b*c - a*d)^2*(c^2 + d^2)^2*f)
```

Maple [F(-1)]

Timed out.

$$\int \frac{A+B\tan(fx+e)+C\tan(fx+e)^2}{\sqrt{a+b\tan(fx+e)}\,(c+d\tan(fx+e))^{\frac{5}{2}}}dx$$

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(1/2)/(c+d*tan(f*x+e))
^(5/2),x)
```

```
[Out] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(1/2)/(c+d*tan(f*x+e))
^(5/2),x)
```

Fricas [F(-1)]

Timed out.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}\,dx = \text{Timed out}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(1/2)/(c+d*tan(f
*x+e))^(5/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}\,dx = \int \frac{A+B\tan{\left(e+fx \right)}+C\tan^{2}{\left(e+fx \right)}}{\sqrt{a+b\tan{\left(e+fx \right)}}\left(c+d\tan{\left(e+fx \right)}\right)^{\frac{5}{2}}}\,dx$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e))**(1/2)/(c+d*tan
(f*x+e))**(5/2),x)
[Out] Integral((A + B*tan(e + f*x) + C*tan(e + f*x)**2)/(sqrt(a + b*tan(e + f*x))
*(c + d*tan(e + f*x))**(5/2)), x)
```

Maxima [F(-1)]

Timed out.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}\,dx = \text{Timed out}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(1/2)/(c+d*tan(f
*x+e))^(5/2),x, algorithm="maxima")
[Out] Timed out
```

Giac [F(-1)]

Timed out.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}\,dx = \text{Timed out}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(1/2)/(c+d*tan(f
*x+e))^(5/2),x, algorithm="giac")
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}\,dx = \text{Hanged}$$

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))^(1/2)*(c
+ d*tan(e + f*x))^(5/2)),x)
[Out] \text{Hanged}
```

3.163 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{5/2}}\,dx$

Optimal result . . . 2145
Rubi [A] (verified) . . . 2146
Mathematica [A] (verified) . . . 2149
Maple [F(-1)] . . . 2150
Fricas [F(-1)] . . . 2150
Sympy [F] . . . 2151
Maxima [F(-1)] . . . 2151
Giac [F(-1)] . . . 2151
Mupad [F(-1)] . . . 2151

Optimal result

Integrand size = 49, antiderivative size = 651

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{5/2}}\,dx =$$

$$-\frac{(iA+B-iC)\operatorname{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a-ib)^{3/2}(c-id)^{5/2}f}$$

$$-\frac{(B-i(A-C))\operatorname{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a+ib)^{3/2}(c+id)^{5/2}f}$$

$$-\frac{2(Ab^2-a(bB-aC))}{(a^2+b^2)(bc-ad)f\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}$$

$$-\frac{2d(b^2c(cC-Bd)-3abB(c^2+d^2)+a^2(4c^2C-Bcd+3Cd^2)+A(a^2d^2+b^2(3c^2+4d^2)))\sqrt{a+b\tan(e+}}{3(a^2+b^2)(bc-ad)^2(c^2+d^2)f(c+d\tan(e+fx))^{3/2}}$$

$$-\frac{2d(b^3c(5c^3C-8Bc^2d-cCd^2-2Bd^3)+a^2b(8c^4C-8Bc^3d+5c^2Cd^2-2Bcd^3+3Cd^4)+3a^3d^2(2cCd+}{3(a^2+b}$$

```
[Out] -(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(1/2)/(c+
d*tan(f*x+e))^(1/2))/(a-I*b)^(3/2)/(c-I*d)^(5/2)/f-(B-I*(A-C))*arctanh((c+I
*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/(a+I
*b)^(3/2)/(c+I*d)^(5/2)/f-2/3*d*(b^3*c*(-8*B*c^2*d-2*B*d^3+5*C*c^3-C*c*d^2)
+a^2*b*(-8*B*c^3*d-2*B*c*d^3+8*C*c^4+5*C*c^2*d^2+3*C*d^4)+3*a^3*d^2*(2*C*c*
d+B*(c^2-d^2))+3*a*b^2*(2*c*C*d^3-B*(c^4+c^2*d^2+2*d^4))-A*(6*a^3*c*d^3+6*a
*b^2*c*d^3-a^2*b*d^2*(11*c^2+5*d^2)-b^3*(3*c^4+17*c^2*d^2+8*d^4)))*(a+b*tan
(f*x+e))^(1/2)/(a^2+b^2)/(-a*d+b*c)^3/(c^2+d^2)^2/f/(c+d*tan(f*x+e))^(1/2)-
2*(A*b^2-a*(B*b-C*a))/(a^2+b^2)/(-a*d+b*c)/f/(a+b*tan(f*x+e))^(1/2)/(c+d*ta
n(f*x+e))^(3/2)-2/3*d*(b^2*c*(-B*d+C*c)-3*a*b*B*(c^2+d^2)+a^2*(-B*c*d+4*C*c
^2+3*C*d^2)+A*(a^2*d^2+b^2*(3*c^2+4*d^2)))*(a+b*tan(f*x+e))^(1/2)/(a^2+b^2)
/(-a*d+b*c)^2/(c^2+d^2)/f/(c+d*tan(f*x+e))^(3/2)
```

Rubi [A] (verified)

Time = 4.04 (sec) , antiderivative size = 650, normalized size of antiderivative = 1.00, number of steps used = 10, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.102, Rules used = {3730, 3697, 3696, 95, 214}

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{5/2}}\,dx =$$

$$-\frac{2d\sqrt{a+b\tan(e+fx)}(a^2Ad^2+a^2(-Bcd+4c^2C+3Cd^2)-3abB(c^2+d^2)+Ab^2(3c^2+4d^2)+b^2c(cC-}{3f(a^2+b^2)(c^2+d^2)(bc-ad)^2(c+d\tan(e+fx))^{3/2}}$$

$$-\frac{2(Ab^2-a(bB-aC))}{f(a^2+b^2)(bc-ad)\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}$$

$$-\frac{2d\sqrt{a+b\tan(e+fx)}(3a^3d^2(B(c^2-d^2)+2cCd)+a^2b(-8Bc^3d-2Bcd^3+8c^4C+5c^2Cd^2+3Cd^4)-A(}{3f(a^2}$$

$$-\frac{(iA+B-iC)\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(a-ib)^{3/2}(c-id)^{5/2}}$$

$$-\frac{(B-i(A-C))\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(a+ib)^{3/2}(c+id)^{5/2}}$$

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^(3/2)*(c
+ d*Tan[e + f*x])^(5/2)),x]
```

```
[Out] -(((I*A + B - I*C)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a
 - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a - I*b)^(3/2)*(c - I*d)^(5/2)*f)) -
((B - I*(A - C))*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a +
 I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a + I*b)^(3/2)*(c + I*d)^(5/2)*f) - (2*
(A*b^2 - a*(b*B - a*C)))/((a^2 + b^2)*(b*c - a*d)*f*Sqrt[a + b*Tan[e + f*x]
]*(c + d*Tan[e + f*x])^(3/2)) - (2*d*(a^2*A*d^2 + b^2*c*(c*C - B*d) - 3*a*b
*B*(c^2 + d^2) + A*b^2*(3*c^2 + 4*d^2) + a^2*(4*c^2*C - B*c*d + 3*C*d^2))*S
qrt[a + b*Tan[e + f*x]])/(3*(a^2 + b^2)*(b*c - a*d)^2*(c^2 + d^2)*f*(c + d*
Tan[e + f*x])^(3/2)) - (2*d*(b^3*c*(5*c^3*C - 8*B*c^2*d - c*C*d^2 - 2*B*d^3
) + a^2*b*(8*c^4*C - 8*B*c^3*d + 5*c^2*C*d^2 - 2*B*c*d^3 + 3*C*d^4) + 3*a^3
*d^2*(2*c*C*d + B*(c^2 - d^2)) + 3*a*b^2*(2*c*C*d^3 - B*(c^4 + c^2*d^2 + 2*
d^4)) - A*(6*a^3*c*d^3 + 6*a*b^2*c*d^3 - a^2*b*d^2*(11*c^2 + 5*d^2) - b^3*(
3*c^4 + 17*c^2*d^2 + 8*d^4)))*Sqrt[a + b*Tan[e + f*x]])/(3*(a^2 + b^2)*(b*c
 - a*d)^3*(c^2 + d^2)^2*f*Sqrt[c + d*Tan[e + f*x]])
```

Rule 95

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
```

```
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 214

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-a/b, 2]/a)*ArcTanh[x
/Rt[-a/b, 2]], x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3696

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[A^2/f, Subst[Int[(a + b*x)^m*((c + d*x)^n/(A - B*x)), x], x, Tan[e + f*x
]], x] /; FreeQ[{a, b, c, d, e, f, A, B, m, n}, x] && NeQ[b*c - a*d, 0] &&
NeQ[a^2 + b^2, 0] && EqQ[A^2 + B^2, 0]
```

Rule 3697

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[(A + I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^n*(1 - I*Ta
n[e + f*x]), x], x] + Dist[(A - I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*T
an[e + f*x])^n*(1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A,
B, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[A^2 + B^2, 0]
```

Rule 3730

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 - a*(b*B - a*C))*(a + b*Tan[e +
 f*x])^(m + 1)*((c + d*Tan[e + f*x])^(n + 1)/(f*(m + 1)*(b*c - a*d)*(a^2 +
b^2))), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rubi steps

$$\text{integral} = -\frac{2(Ab^2-a(bB-aC))}{(a^2+b^2)\,(bc-ad)f\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}} - \frac{2\int\frac{\frac{1}{2}(4Ab^2d-aA(bc-ad)-(bB-aC)(bc+3ad))+\frac{1}{2}(Ab-aB-bC)(bc-ad)\tan(e+fx)+2(Ab^2-a(bB-aC))d\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{5/2}}\,dx}{(a^2+b^2)\,(bc-ad)}$$

$$= -\frac{2(Ab^2 - a(bB - aC))}{(a^2+b^2)(bc-ad)f\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}$$
$$-\frac{2d(a^2Ad^2 + b^2c(cC - Bd) - 3abB(c^2+d^2) + Ab^2(3c^2+4d^2) + a^2(4c^2C - Bcd + 3Cd^2))\sqrt{a+}}{3(a^2+b^2)(bc-ad)^2(c^2+d^2)f(c+d\tan(e+fx))^{3/2}}$$
$$-\frac{4\int \frac{1}{4}((3bc^2-3acd+2bd^2)(a^2(A+3C)d-b^2(Bc-4Ad)-ab(Ac-cC+3Bd))-d(bc-3ad)(b^2cC+Ab(3bc+ad)+a^2(4cC-Bd)-ab(3Bc}{3}$$

$$= -\frac{2(Ab^2 - a(bB - aC))}{(a^2+b^2)(bc-ad)f\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}$$
$$-\frac{2d(a^2Ad^2 + b^2c(cC - Bd) - 3abB(c^2+d^2) + Ab^2(3c^2+4d^2) + a^2(4c^2C - Bcd + 3Cd^2))\sqrt{a+}}{3(a^2+b^2)(bc-ad)^2(c^2+d^2)f(c+d\tan(e+fx))^{3/2}}$$
$$-\frac{2d(b^3c(5c^3C - 8Bc^2d - cCd^2 - 2Bd^3) + a^2b(8c^4C - 8Bc^3d + 5c^2Cd^2 - 2Bcd^3 + 3Cd^4) + 3a^3d^2}{}$$
$$-\frac{8\int \frac{\frac{3}{8}(bc-ad)^3(a(c^2C-2Bcd-Cd^2-A(c^2-d^2))+b(2c(A-C)d-B(c^2-d^2)))+\frac{3}{8}(bc-ad)^3(2aAcd-2acCd+Ab(c^2-d^2)-aB(c^2-d^2}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}}{3(a^2+b^2)(bc-ad)^3(c^2+d^2)^2}$$

$$= -\frac{2(Ab^2 - a(bB - aC))}{(a^2+b^2)(bc-ad)f\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}$$
$$-\frac{2d(a^2Ad^2 + b^2c(cC - Bd) - 3abB(c^2+d^2) + Ab^2(3c^2+4d^2) + a^2(4c^2C - Bcd + 3Cd^2))\sqrt{a+}}{3(a^2+b^2)(bc-ad)^2(c^2+d^2)f(c+d\tan(e+fx))^{3/2}}$$
$$-\frac{2d(b^3c(5c^3C - 8Bc^2d - cCd^2 - 2Bd^3) + a^2b(8c^4C - 8Bc^3d + 5c^2Cd^2 - 2Bcd^3 + 3Cd^4) + 3a^3d^2}{}$$
$$+\frac{(A - iB - C)\int \frac{1+i\tan(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx}{2(a-ib)(c-id)^2}$$
$$+\frac{(A + iB - C)\int \frac{1-i\tan(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx}{2(a+ib)(c+id)^2}$$

$$= -\frac{2(Ab^2 - a(bB - aC))}{(a^2+b^2)(bc-ad)f\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}$$
$$-\frac{2d(a^2Ad^2 + b^2c(cC - Bd) - 3abB(c^2+d^2) + Ab^2(3c^2+4d^2) + a^2(4c^2C - Bcd + 3Cd^2))\sqrt{a+}}{3(a^2+b^2)(bc-ad)^2(c^2+d^2)f(c+d\tan(e+fx))^{3/2}}$$
$$-\frac{2d(b^3c(5c^3C - 8Bc^2d - cCd^2 - 2Bd^3) + a^2b(8c^4C - 8Bc^3d + 5c^2Cd^2 - 2Bcd^3 + 3Cd^4) + 3a^3d^2}{}$$
$$+\frac{(A - iB - C)\text{Subst}\left(\int \frac{1}{(1-ix)\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e+fx)\right)}{2(a-ib)(c-id)^2 f}$$
$$+\frac{(A + iB - C)\text{Subst}\left(\int \frac{1}{(1+ix)\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e+fx)\right)}{2(a+ib)(c+id)^2 f}$$

$$= -\frac{2(Ab^2 - a(bB - aC))}{(a^2+b^2)(bc-ad)f\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}$$
$$- \frac{2d(a^2Ad^2 + b^2c(cC - Bd) - 3abB(c^2+d^2) + Ab^2(3c^2+4d^2) + a^2(4c^2C - Bcd + 3Cd^2))\sqrt{a+}}{3(a^2+b^2)(bc-ad)^2(c^2+d^2)f(c+d\tan(e+fx))^{3/2}}$$
$$- \frac{2d(b^3c(5c^3C - 8Bc^2d - cCd^2 - 2Bd^3) + a^2b(8c^4C - 8Bc^3d + 5c^2Cd^2 - 2Bcd^3 + 3Cd^4) + 3a^3}{}$$
$$+ \frac{(A - iB - C)\text{Subst}\left(\int \frac{1}{ia+b-(ic+d)x^2}\,dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{(a-ib)(c-id)^2f}$$
$$+ \frac{(A + iB - C)\text{Subst}\left(\int \frac{1}{-ia+b-(-ic+d)x^2}\,dx, x, \frac{\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}}\right)}{(a+ib)(c+id)^2f}$$
$$= -\frac{(iA + B - iC)\text{arctanh}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a-ib)^{3/2}(c-id)^{5/2}f}$$
$$- \frac{(B - i(A-C))\text{arctanh}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a+ib)^{3/2}(c+id)^{5/2}f}$$
$$- \frac{2(Ab^2 - a(bB - aC))}{(a^2+b^2)(bc-ad)f\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}$$
$$- \frac{2d(a^2Ad^2 + b^2c(cC - Bd) - 3abB(c^2+d^2) + Ab^2(3c^2+4d^2) + a^2(4c^2C - Bcd + 3Cd^2))\sqrt{a+}}{3(a^2+b^2)(bc-ad)^2(c^2+d^2)f(c+d\tan(e+fx))^{3/2}}$$
$$- \frac{2d(b^3c(5c^3C - 8Bc^2d - cCd^2 - 2Bd^3) + a^2b(8c^4C - 8Bc^3d + 5c^2Cd^2 - 2Bcd^3 + 3Cd^4) + 3a^3}{}$$

Mathematica [A] (verified)

Time = 7.21 (sec) , antiderivative size = 903, normalized size of antiderivative = 1.39

$$\int \frac{A + B\tan(e+fx) + C\tan^2(e+fx)}{(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{5/2}}\,dx =$$
$$-\frac{2(Ab^2 - a(bB - aC))}{(a^2+b^2)(bc-ad)f\sqrt{a+b\tan(e+fx)}(c+d\tan(e+fx))^{3/2}}$$
$$-\frac{2\left(-\frac{2\left(-c\left(-2c(Ab^2-a(bB-aC))d+\frac{1}{2}(Ab-aB-bC)d(bc-ad)\right)+\frac{1}{2}d^2\left(4Ab^2d-aA(bc-ad)-(bB-aC)(bc+3ad)\right)\right)\sqrt{a+b\tan(e+fx)}}{3(-bc+ad)(c^2+d^2)f(c+d\tan(e+fx))^{3/2}} - \frac{2\left(\frac{3(bc}{}\right.}{}\right.}{}$$

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^(3/
2)*(c + d*Tan[e + f*x])^(5/2)),x]
```

```
[Out] (-2*(A*b^2 - a*(b*B - a*C)))/((a^2 + b^2)*(b*c - a*d)*f*Sqrt[a + b*Tan[e +
f*x]]*(c + d*Tan[e + f*x])^(3/2)) - (2*((-2*(-(c*(-2*c*(A*b^2 - a*(b*B - a*
C))*d + ((A*b - a*B - b*C)*d*(b*c - a*d))/2)) + (d^2*(4*A*b^2*d - a*A*(b*c
- a*d) - (b*B - a*C)*(b*c + 3*a*d)))/2)*Sqrt[a + b*Tan[e + f*x]])/(3*(-(b*c
) + a*d)*(c^2 + d^2)*f*(c + d*Tan[e + f*x])^(3/2)) - (2*((3*(b*c - a*d)^3*(
((a + I*b)*(I*A + B - I*C)*(c + I*d)^2*ArcTanh[(Sqrt[-c + I*d]*Sqrt[a + b*T
an[e + f*x]])/(Sqrt[-a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[-a + I*b]*S
qrt[-c + I*d]) + ((I*a + b)*(A + I*B - C)*(c - I*d)^2*ArcTanh[(Sqrt[c + I*d
]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqr
t[a + I*b]*Sqrt[c + I*d])))/(4*(-(b*c) + a*d)*(c^2 + d^2)*f) - (2*(d^2*(((b
*c)/2 - (3*a*d)/2)*(-2*c*(A*b^2 - a*(b*B - a*C))*d + ((A*b - a*B - b*C)*d*(
b*c - a*d))/2) + ((b*d^2 - (3*c*(-(b*c) + a*d))/2)*(4*A*b^2*d - a*A*(b*c -
a*d) - (b*B - a*C)*(b*c + 3*a*d)))/2) - c*((3*d*(-(b*c) + a*d)*(-2*(A*b^2 -
 a*(b*B - a*C))*d^2 - (c*(A*b - a*B - b*C)*(b*c - a*d))/2 + (d*(4*A*b^2*d -
 a*A*(b*c - a*d) - (b*B - a*C)*(b*c + 3*a*d)))/2))/2 - b*c*(-(c*(-2*c*(A*b^
2 - a*(b*B - a*C))*d + ((A*b - a*B - b*C)*d*(b*c - a*d))/2)) + (d^2*(4*A*b^
2*d - a*A*(b*c - a*d) - (b*B - a*C)*(b*c + 3*a*d)))/2)))*Sqrt[a + b*Tan[e +
 f*x]])/((-(b*c) + a*d)*(c^2 + d^2)*f*Sqrt[c + d*Tan[e + f*x]])))/(3*(-(b*c
) + a*d)*(c^2 + d^2))))/((a^2 + b^2)*(b*c - a*d))
```

Maple [F(-1)]

Timed out.

hanged

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(3/2)/(c+d*tan(f*x+e))
^(5/2),x)
```

```
[Out] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(3/2)/(c+d*tan(f*x+e))
^(5/2),x)
```

Fricas [F(-1)]

Timed out.

$$\int \frac{A + B\tan(e+fx) + C\tan^2(e+fx)}{(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{5/2}}\,dx = \text{Timed out}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(3/2)/(c+d*tan(f
*x+e))^(5/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

Sympy [F]

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{5/2}}\,dx=\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{\frac{3}{2}}(c+d\tan(e+fx))^{\frac{5}{2}}}\,dx$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e))**(3/2)/(c+d*tan
(f*x+e))**(5/2),x)
[Out] Integral((A + B*tan(e + f*x) + C*tan(e + f*x)**2)/((a + b*tan(e + f*x))**(3
/2)*(c + d*tan(e + f*x))**(5/2)), x)
```

Maxima [F(-1)]

Timed out.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{5/2}}\,dx=\text{Timed out}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(3/2)/(c+d*tan(f
*x+e))^(5/2),x, algorithm="maxima")
[Out] Timed out
```

Giac [F(-1)]

Timed out.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{5/2}}\,dx=\text{Timed out}$$

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(3/2)/(c+d*tan(f
*x+e))^(5/2),x, algorithm="giac")
[Out] Timed out
```

Mupad [F(-1)]

Timed out.

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{5/2}}\,dx=\text{Hanged}$$

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))^(3/2)*(c
+ d*tan(e + f*x))^(5/2)),x)
[Out] \text{Hanged}
```

3.164 $\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^n\left(A+B\tan(e+\right.$

Optimal result 2152
Rubi [A] (verified) 2153
Mathematica [F] 2155
Maple [F] 2156
Fricas [F] 2156
Sympy [F(-2)] 2156
Maxima [F] 2157
Giac [F] 2157
Mupad [F(-1)] 2157

Optimal result

Integrand size = 45, antiderivative size = 376

$$\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^n\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx =$$

$$-\frac{(B+i(A-C))\,\text{AppellF1}\left(1+m,-n,1,2+m,-\frac{d(a+b\tan(e+fx))}{bc-ad},\frac{a+b\tan(e+fx)}{a-ib}\right)(a+b\tan(e+fx))^{1+m}(c}{2(a-ib)f(1+m)}$$

$$-\frac{(A+iB-C)\,\text{AppellF1}\left(1+m,-n,1,2+m,-\frac{d(a+b\tan(e+fx))}{bc-ad},\frac{a+b\tan(e+fx)}{a+ib}\right)(a+b\tan(e+fx))^{1+m}(c+}{2(ia-b)f(1+m)}$$

$$+\frac{C\,\text{Hypergeometric2F1}\left(1+m,-n,2+m,-\frac{d(a+b\tan(e+fx))}{bc-ad}\right)(a+b\tan(e+fx))^{1+m}(c+d\tan(e+fx))^n}{bf(1+m)}$$

```
[Out] -1/2*(B+I*(A-C))*AppellF1(1+m,1,-n,2+m,(a+b*tan(f*x+e))/(a-I*b),-d*(a+b*tan
(f*x+e))/(-a*d+b*c))*(a+b*tan(f*x+e))^(1+m)*(c+d*tan(f*x+e))^n/(a-I*b)/f/(1
+m)/((b*(c+d*tan(f*x+e))/(-a*d+b*c))^n)-1/2*(A+I*B-C)*AppellF1(1+m,1,-n,2+m
,(a+b*tan(f*x+e))/(a+I*b),-d*(a+b*tan(f*x+e))/(-a*d+b*c))*(a+b*tan(f*x+e))^
(1+m)*(c+d*tan(f*x+e))^n/(I*a-b)/f/(1+m)/((b*(c+d*tan(f*x+e))/(-a*d+b*c))^n
)+C*hypergeom([-n, 1+m],[2+m],-d*(a+b*tan(f*x+e))/(-a*d+b*c))*(a+b*tan(f*x+
e))^(1+m)*(c+d*tan(f*x+e))^n/b/f/(1+m)/((b*(c+d*tan(f*x+e))/(-a*d+b*c))^n)
```

Rubi [A] (verified)

Time = 1.03 (sec) , antiderivative size = 376, normalized size of antiderivative = 1.00, number of steps used = 9, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}} = 0.133$, Rules used = {3736, 6857, 72, 71, 142, 141}

$$\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^n\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx =$$
$$-\frac{(B+i(A-C))(a+b\tan(e+fx))^{m+1}(c+d\tan(e+fx))^n\left(\frac{b(c+d\tan(e+fx))}{bc-ad}\right)^{-n}\text{AppellF1}\left(m+1,-n,\right.}{2f(m+1)(a-ib)}$$
$$-\frac{(A+iB-C)(a+b\tan(e+fx))^{m+1}(c+d\tan(e+fx))^n\left(\frac{b(c+d\tan(e+fx))}{bc-ad}\right)^{-n}\text{AppellF1}\left(m+1,-n,1,\right.}{2f(m+1)(-b+ia)}$$
$$+\frac{C(a+b\tan(e+fx))^{m+1}(c+d\tan(e+fx))^n\left(\frac{b(c+d\tan(e+fx))}{bc-ad}\right)^{-n}\text{Hypergeometric2F1}\left(m+1,-n,m+\right.}{bf(m+1)}$$

```
[In] Int[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^n*(A + B*Tan[e + f*x] + C*T
an[e + f*x]^2),x]
```

```
[Out] -1/2*((B + I*(A - C))*AppellF1[1 + m, -n, 1, 2 + m, -((d*(a + b*Tan[e + f*x
]))/(b*c - a*d)), (a + b*Tan[e + f*x])/(a - I*b)]*(a + b*Tan[e + f*x])^(1 +
 m)*(c + d*Tan[e + f*x])^n)/((a - I*b)*f*(1 + m)*((b*(c + d*Tan[e + f*x]))/
(b*c - a*d))^n) - ((A + I*B - C)*AppellF1[1 + m, -n, 1, 2 + m, -((d*(a + b*
Tan[e + f*x]))/(b*c - a*d)), (a + b*Tan[e + f*x])/(a + I*b)]*(a + b*Tan[e +
 f*x])^(1 + m)*(c + d*Tan[e + f*x])^n)/(2*(I*a - b)*f*(1 + m)*((b*(c + d*Ta
n[e + f*x]))/(b*c - a*d))^n) + (C*Hypergeometric2F1[1 + m, -n, 2 + m, -((d*
(a + b*Tan[e + f*x]))/(b*c - a*d))]*(a + b*Tan[e + f*x])^(1 + m)*(c + d*Tan
[e + f*x])^n)/(b*f*(1 + m)*((b*(c + d*Tan[e + f*x]))/(b*c - a*d))^n)
```

Rule 71

```
Int[((a_) + (b_.)*(x_))^(m_)*((c_) + (d_.)*(x_))^(n_), x_Symbol] :> Simp[((
a + b*x)^(m + 1)/(b*(m + 1)*(b/(b*c - a*d))^n))*Hypergeometric2F1[-n, m + 1
, m + 2, (-d)*((a + b*x)/(b*c - a*d))], x] /; FreeQ[{a, b, c, d, m, n}, x]
&& NeQ[b*c - a*d, 0] &&  !IntegerQ[m] &&  !IntegerQ[n] && GtQ[b/(b*c - a*d)
, 0] && (RationalQ[m] ||  !(RationalQ[n] && GtQ[-d/(b*c - a*d), 0]))
```

Rule 72

```
Int[((a_) + (b_.)*(x_))^(m_)*((c_) + (d_.)*(x_))^(n_), x_Symbol] :> Dist[(c
 + d*x)^FracPart[n]/((b/(b*c - a*d))^IntPart[n]*(b*((c + d*x)/(b*c - a*d)))
^FracPart[n]), Int[(a + b*x)^m*Simp[b*(c/(b*c - a*d)) + b*d*(x/(b*c - a*d))
, x]^n, x], x] /; FreeQ[{a, b, c, d, m, n}, x] && NeQ[b*c - a*d, 0] &&  !In
tegerQ[m] &&  !IntegerQ[n] && (RationalQ[m] ||  !SimplerQ[n + 1, m + 1])
```

Rule 141

```
Int[((a_) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_)*((e_.) + (f_.)*(x_))
^(p_), x_Symbol] :> Simp[(b*e - a*f)^p*((a + b*x)^(m + 1)/(b^(p + 1)*(m + 1
)*(b/(b*c - a*d))^n))*AppellF1[m + 1, -n, -p, m + 2, (-d)*((a + b*x)/(b*c -
 a*d)), (-f)*((a + b*x)/(b*e - a*f))], x] /; FreeQ[{a, b, c, d, e, f, m, n}
, x] &&  !IntegerQ[m] &&  !IntegerQ[n] && IntegerQ[p] && GtQ[b/(b*c - a*d),
 0] &&  !(GtQ[d/(d*a - c*b), 0] && SimplerQ[c + d*x, a + b*x])
```

Rule 142

```
Int[((a_) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_)*((e_.) + (f_.)*(x_))
^(p_), x_Symbol] :> Dist[(c + d*x)^FracPart[n]/((b/(b*c - a*d))^IntPart[n]*
(b*((c + d*x)/(b*c - a*d)))^FracPart[n]), Int[(a + b*x)^m*(b*(c/(b*c - a*d)
) + b*d*(x/(b*c - a*d)))^n*(e + f*x)^p, x], x] /; FreeQ[{a, b, c, d, e, f,
m, n}, x] &&  !IntegerQ[m] &&  !IntegerQ[n] && IntegerQ[p] &&  !GtQ[b/(b*c
- a*d), 0] &&  !SimplerQ[c + d*x, a + b*x]
```

Rule 3736

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[(a + b*ff*x)^m*(c + d*ff*x)^n*((A + B*ff*x + C*ff^2*x^2
)/(1 + ff^2*x^2)), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6857

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\text{integral} = \frac{\text{Subst}\left(\int \frac{(a+bx)^m(c+dx)^n(A+Bx+Cx^2)}{1+x^2}\,dx, x, \tan(e+fx)\right)}{f}$$

$$= \frac{\text{Subst}\left(\int \left(C(a+bx)^m(c+dx)^n + \frac{(-B+i(A-C))(a+bx)^m(c+dx)^n}{2(i-x)} + \frac{(B+i(A-C))(a+bx)^m(c+dx)^n}{2(i+x)}\right)\,dx, x, \tan(\right.}{f}$$

$$= \frac{(-B+i(A-C))\text{Subst}\left(\int \frac{(a+bx)^m(c+dx)^n}{i-x}\,dx,x,\tan(e+fx)\right)}{2f} + \frac{(B+i(A-C))\text{Subst}\left(\int \frac{(a+bx)^m(c+dx)^n}{i+x}\,dx,x,\tan(e+fx)\right)}{2f} + \frac{C\text{Subst}\left(\int (a+bx)^m(c+dx)^n\,dx,x,\tan(e+fx)\right)}{f}$$

$$= \frac{\left((-B+i(A-C))(c+d\tan(e+fx))^n\left(\frac{b(c+d\tan(e+fx))}{bc-ad}\right)^{-n}\right)\text{Subst}\left(\int \frac{(a+bx)^m\left(\frac{bc}{bc-ad}+\frac{bdx}{bc-ad}\right)^n}{i-x}\,dx,x\right.}{2f} + \frac{\left((B+i(A-C))(c+d\tan(e+fx))^n\left(\frac{b(c+d\tan(e+fx))}{bc-ad}\right)^{-n}\right)\text{Subst}\left(\int \frac{(a+bx)^m\left(\frac{bc}{bc-ad}+\frac{bdx}{bc-ad}\right)^n}{i+x}\,dx,x\right.}{2f} + \frac{\left(C(c+d\tan(e+fx))^n\left(\frac{b(c+d\tan(e+fx))}{bc-ad}\right)^{-n}\right)\text{Subst}\left(\int (a+bx)^m\left(\frac{bc}{bc-ad}+\frac{bdx}{bc-ad}\right)^n\,dx,x,\tan(e+\right.}{f}$$

$$= -\frac{(B+i(A-C))\,\text{AppellF1}\left(1+m,-n,1,2+m,-\frac{d(a+b\tan(e+fx))}{bc-ad},\frac{a+b\tan(e+fx)}{a-ib}\right)(a+b\tan(e+f}{2(a-ib)f(1+m)} - \frac{(A+iB-C)\,\text{AppellF1}\left(1+m,-n,1,2+m,-\frac{d(a+b\tan(e+fx))}{bc-ad},\frac{a+b\tan(e+fx)}{a+ib}\right)(a+b\tan(e+fx)}{2(ia-b)f(1+m)} + \frac{C\,\text{Hypergeometric2F1}\left(1+m,-n,2+m,-\frac{d(a+b\tan(e+fx))}{bc-ad}\right)(a+b\tan(e+fx))^{1+m}(c+d\tan(e}{bf(1+m)}$$

Mathematica [F]

$$\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^n\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= \int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^n\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$

```
[In] Integrate[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^n*(A + B*Tan[e + f*x]
 + C*Tan[e + f*x]^2),x]
[Out] Integrate[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^n*(A + B*Tan[e + f*x]
 + C*Tan[e + f*x]^2), x]
```

Maple [F]

$$\int (a+b\tan(fx+e))^m (c+d\tan(fx+e))^n \left(A+B\tan(fx+e)+C\tan(fx+e)^2\right) dx$$

```
[In] int((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))^n*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x
)
```

```
[Out] int((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))^n*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x
)
```

Fricas [F]

$$\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^n \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right) dx$$
$$= \int (C\tan(fx+e)^2+B\tan(fx+e)+A)(b\tan(fx+e)+a)^m(d\tan(fx+e)+c)^n dx$$

```
[In] integrate((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))^n*(A+B*tan(f*x+e)+C*tan(f*x+e
)^2),x, algorithm="fricas")
```

```
[Out] integral((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)^m*(d*
tan(f*x + e) + c)^n, x)
```

Sympy [F(-2)]

Exception generated.

$$\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^n \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right) dx$$
$$= \text{Exception raised: HeuristicGCDFailed}$$

```
[In] integrate((a+b*tan(f*x+e))**m*(c+d*tan(f*x+e))**n*(A+B*tan(f*x+e)+C*tan(f*x
+e)**2),x)
```

```
[Out] Exception raised: HeuristicGCDFailed >> no luck
```

Maxima [F]

$$\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^n \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= \int (C\tan{(fx+e)}^2+B\tan{(fx+e)}+A)(b\tan{(fx+e)}+a)^m(d\tan{(fx+e)}+c)^n\,dx$$

```
[In] integrate((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))^n*(A+B*tan(f*x+e)+C*tan(f*x+e
)^2),x, algorithm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)^m*(d
*tan(f*x + e) + c)^n, x)
```

Giac [F]

$$\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^n \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= \int (C\tan{(fx+e)}^2+B\tan{(fx+e)}+A)(b\tan{(fx+e)}+a)^m(d\tan{(fx+e)}+c)^n\,dx$$

```
[In] integrate((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))^n*(A+B*tan(f*x+e)+C*tan(f*x+e
)^2),x, algorithm="giac")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)^m*(d
*tan(f*x + e) + c)^n, x)
```

Mupad [F(-1)]

Timed out.

$$\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^n \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= \int (a+b\tan(e+f\,x))^m\,(c+d\tan(e+f\,x))^n\,\left(C\tan(e+f\,x)^2+B\tan(e+f\,x)+A\right)\,dx$$

```
[In] int((a + b*tan(e + f*x))^m*(c + d*tan(e + f*x))^n*(A + B*tan(e + f*x) + C*t
an(e + f*x)^2),x)
```

```
[Out] int((a + b*tan(e + f*x))^m*(c + d*tan(e + f*x))^n*(A + B*tan(e + f*x) + C*t
an(e + f*x)^2), x)
```

3.165 $\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^3\,(A+B\tan(e$

Optimal result 2158
Rubi [A] (verified) 2159
Mathematica [B] (verified) 2163
Maple [F] 2164
Fricas [F] 2164
Sympy [F] 2165
Maxima [F(-1)] 2165
Giac [F] 2165
Mupad [F(-1)] 2166

Optimal result

Integrand size = 45, antiderivative size = 560

$$\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$

$$= \frac{(bc(2+m)\left(b^2d(Bc+(A-C)d)(3+m)(4+m)-2(bc-ad)(3aCd-b(3cC+Bd(4+m)))\right)+d(b^3(2c(A}{}$$

$$+\frac{(A-iB-C)(c-id)^3\,\text{Hypergeometric2F1}\left(1,1+m,2+m,\frac{a+b\tan(e+fx)}{a-ib}\right)(a+b\tan(e+fx))^{1+m}}{2(ia+b)f(1+m)}$$

$$-\frac{(A+iB-C)(c+id)^3\,\text{Hypergeometric2F1}\left(1,1+m,2+m,\frac{a+b\tan(e+fx)}{a+ib}\right)(a+b\tan(e+fx))^{1+m}}{2(ia-b)f(1+m)}$$

$$+\frac{d(b^2d(Bc+(A-C)d)(3+m)(4+m)-2(bc-ad)(3aCd-b(3cC+Bd(4+m))))\tan(e+fx)(a+b\,\text{ta}}{b^3f(2+m)(3+m)(4+m)}$$

$$-\frac{(3aCd-b(3cC+Bd(4+m)))(a+b\tan(e+fx))^{1+m}(c+d\tan(e+fx))^2}{b^2f(3+m)(4+m)}$$

$$+\frac{C(a+b\tan(e+fx))^{1+m}(c+d\tan(e+fx))^3}{bf(4+m)}$$

```
[Out] (b*c*(2+m)*(b^2*d*(B*c+(A-C)*d)*(3+m)*(4+m)-2*(-a*d+b*c)*(3*C*a*d-b*(3*C*c+
B*d*(4+m))))+d*(b^3*(2*c*(A-C)*d+B*(c^2-d^2))*(2+m)*(3+m)*(4+m)-a*(b^2*d*(B
*c+(A-C)*d)*(3+m)*(4+m)-2*(-a*d+b*c)*(3*C*a*d-b*(3*C*c+B*d*(4+m))))))*(a+b*
tan(f*x+e))^(1+m)/b^4/f/(1+m)/(2+m)/(3+m)/(4+m)+1/2*(A-I*B-C)*(c-I*d)^3*hyp
ergeom([1, 1+m],[2+m],(a+b*tan(f*x+e))/(a-I*b))*(a+b*tan(f*x+e))^(1+m)/(I*a
+b)/f/(1+m)-1/2*(A+I*B-C)*(c+I*d)^3*hypergeom([1, 1+m],[2+m],(a+b*tan(f*x+e
))/(a+I*b))*(a+b*tan(f*x+e))^(1+m)/(I*a-b)/f/(1+m)+d*(b^2*d*(B*c+(A-C)*d)*(
3+m)*(4+m)-2*(-a*d+b*c)*(3*C*a*d-b*(3*C*c+B*d*(4+m))))*tan(f*x+e)*(a+b*tan(
f*x+e))^(1+m)/b^3/f/(2+m)/(3+m)/(4+m)-(3*C*a*d-b*(3*C*c+B*d*(4+m)))*(a+b*ta
n(f*x+e))^(1+m)*(c+d*tan(f*x+e))^2/b^2/f/(3+m)/(4+m)+C*(a+b*tan(f*x+e))^(1+
m)*(c+d*tan(f*x+e))^3/b/f/(4+m)
```

Rubi [A] (verified)

Time = 2.45 (sec) , antiderivative size = 551, normalized size of antiderivative = 0.98, number of steps used = 9, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}} = 0.133$, Rules used = {3728, 3718, 3711, 3620, 3618, 70}

$$\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$

$$=\frac{d\tan(e+fx)(a+b\tan(e+fx))^{m+1}\left(2(bc-ad)(-3aCd+bBd(m+4)+3bcC)+b^2d(m+3)(m+4)(d\right.}{b^3f(m+2)(m+3)(m+4)}$$

$$+\frac{(a+b\tan(e+fx))^{m+1}\left(bc(m+2)\left(2(bc-ad)(-3aCd+bBd(m+4)+3bcC)+b^2d(m+3)(m+4)(d\right.\right.}{}$$

$$+\frac{(c-id)^3(A-iB-C)(a+b\tan(e+fx))^{m+1}\operatorname{Hypergeometric2F1}\left(1,m+1,m+2,\frac{a+b\tan(e+fx)}{a-ib}\right)}{2f(m+1)(b+ia)}$$

$$-\frac{(c+id)^3(A+iB-C)(a+b\tan(e+fx))^{m+1}\operatorname{Hypergeometric2F1}\left(1,m+1,m+2,\frac{a+b\tan(e+fx)}{a+ib}\right)}{2f(m+1)(-b+ia)}$$

$$+\frac{(c+d\tan(e+fx))^2(-3aCd+bBd(m+4)+3bcC)(a+b\tan(e+fx))^{m+1}}{b^2f(m+3)(m+4)}$$

$$+\frac{C(c+d\tan(e+fx))^3(a+b\tan(e+fx))^{m+1}}{bf(m+4)}$$

```
[In] Int[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*T
an[e + f*x]^2),x]
```

```
[Out] ((b*c*(2 + m)*(b^2*d*(B*c + (A - C)*d)*(3 + m)*(4 + m) + 2*(b*c - a*d)*(3*b
*c*C - 3*a*C*d + b*B*d*(4 + m))) + d*(b^3*(2*c*(A - C)*d + B*(c^2 - d^2))*(
2 + m)*(3 + m)*(4 + m) - a*(b^2*d*(B*c + (A - C)*d)*(3 + m)*(4 + m) + 2*(b*
c - a*d)*(3*b*c*C - 3*a*C*d + b*B*d*(4 + m)))))*(a + b*Tan[e + f*x])^(1 + m
))/(b^4*f*(1 + m)*(2 + m)*(3 + m)*(4 + m)) + ((A - I*B - C)*(c - I*d)^3*Hyp
ergeometric2F1[1, 1 + m, 2 + m, (a + b*Tan[e + f*x])/(a - I*b)]*(a + b*Tan[
e + f*x])^(1 + m))/(2*(I*a + b)*f*(1 + m)) - ((A + I*B - C)*(c + I*d)^3*Hyp
ergeometric2F1[1, 1 + m, 2 + m, (a + b*Tan[e + f*x])/(a + I*b)]*(a + b*Tan[
e + f*x])^(1 + m))/(2*(I*a - b)*f*(1 + m)) + (d*(b^2*d*(B*c + (A - C)*d)*(3
 + m)*(4 + m) + 2*(b*c - a*d)*(3*b*c*C - 3*a*C*d + b*B*d*(4 + m)))*Tan[e +
f*x]*(a + b*Tan[e + f*x])^(1 + m))/(b^3*f*(2 + m)*(3 + m)*(4 + m)) + ((3*b*
c*C - 3*a*C*d + b*B*d*(4 + m))*(a + b*Tan[e + f*x])^(1 + m)*(c + d*Tan[e +
f*x])^2)/(b^2*f*(3 + m)*(4 + m)) + (C*(a + b*Tan[e + f*x])^(1 + m)*(c + d*T
an[e + f*x])^3)/(b*f*(4 + m))
```

Rule 70

```
Int[((a_) + (b_.)*(x_))^(m_)*((c_) + (d_.)*(x_))^(n_), x_Symbol] :> Simp[(b
*c - a*d)^n*((a + b*x)^(m + 1)/(b^(n + 1)*(m + 1)))*Hypergeometric2F1[-n, m
 + 1, m + 2, (-d)*((a + b*x)/(b*c - a*d))], x] /; FreeQ[{a, b, c, d, m}, x]
```

```
&& NeQ[b*c - a*d, 0] &&  !IntegerQ[m] && IntegerQ[n]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
```

```
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\text{integral} = \frac{C(a+b\tan(e+fx))^{1+m}(c+d\tan(e+fx))^3}{bf(4+m)} + \frac{\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^2\left(Abc(4+m)-C(3ad+bc(1+m))+b(Bc+(A-C)d)(4\right.}{b(4+m)}$$

$$= \frac{(3bcC-3aCd+bBd(4+m))(a+b\tan(e+fx))^{1+m}(c+d\tan(e+fx))^2}{b^2f(3+m)(4+m)} + \frac{C(a+b\tan(e+fx))^{1+m}(c+d\tan(e+fx))^3}{bf(4+m)} + \frac{\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))\left(-((2ad+bc(1+m))(3bcC-3aCd+bBd(4+m))\right.}{}$$

$$= \frac{d(b^2d(Bc+(A-C)d)(3+m)(4+m)+2(bc-ad)(3bcC-3aCd+bBd(4+m)))\tan(e+fx)(c}{b^3f(2+m)(3+m)(4+m)} + \frac{(3bcC-3aCd+bBd(4+m))(a+b\tan(e+fx))^{1+m}(c+d\tan(e+fx))^2}{b^2f(3+m)(4+m)} + \frac{C(a+b\tan(e+fx))^{1+m}(c+d\tan(e+fx))^3}{bf(4+m)} - \frac{\int (a+b\tan(e+fx))^m\left(ad(b^2d(Bc+(A-C)d)(3+m)(4+m)+2(bc-ad)(3bcC-3aCd+b\right.}{}$$

$$= \frac{(bc(2+m)\left(b^2d(Bc+(A-C)d)(3+m)(4+m)+2(bc-ad)(3bcC-3aCd+bBd(4+m))\right)+a}{} + \frac{d(b^2d(Bc+(A-C)d)(3+m)(4+m)+2(bc-ad)(3bcC-3aCd+bBd(4+m)))\tan(e+fx}{b^3f(2+m)(3+m)(4+m)} + \frac{(3bcC-3aCd+bBd(4+m))(a+b\tan(e+fx))^{1+m}(c+d\tan(e+fx))^2}{b^2f(3+m)(4+m)} + \frac{C(a+b\tan(e+fx))^{1+m}(c+d\tan(e+fx))^3}{bf(4+m)} - \frac{\int (a+b\tan(e+fx))^m\left(-b^3(Ac^3-c^3C-3Bc^2d-3Acd^2+3cCd^2+Bd^3)\right)(2+m)(3+m)(4+}{b^3(2+m)(3+m}$$

$$= \frac{(bc(2+m)\left(b^2d(Bc+(A-C)d)(3+m)(4+m)+2(bc-ad)(3bcC-3aCd+bBd(4+m))\right)+d(}{}$$

$$+\frac{d(b^2d(Bc+(A-C)d)(3+m)(4+m)+2(bc-ad)(3bcC-3aCd+bBd(4+m)))\tan(e+fx)(}{b^3f(2+m)(3+m)(4+m)}$$

$$+\frac{(3bcC-3aCd+bBd(4+m))(a+b\tan(e+fx))^{1+m}(c+d\tan(e+fx))^2}{b^2f(3+m)(4+m)}$$

$$+\frac{C(a+b\tan(e+fx))^{1+m}(c+d\tan(e+fx))^3}{bf(4+m)}$$

$$+\frac{1}{2}\left((A-iB-C)(c-id)^3\right)\int(1+i\tan(e+fx))(a+b\tan(e+fx))^m\,dx$$

$$+\frac{1}{2}\left((A+iB-C)(c+id)^3\right)\int(1-i\tan(e+fx))(a+b\tan(e+fx))^m\,dx$$

$$= \frac{(bc(2+m)\left(b^2d(Bc+(A-C)d)(3+m)(4+m)+2(bc-ad)(3bcC-3aCd+bBd(4+m))\right)+d(}{}$$

$$+\frac{d(b^2d(Bc+(A-C)d)(3+m)(4+m)+2(bc-ad)(3bcC-3aCd+bBd(4+m)))\tan(e+fx)(}{b^3f(2+m)(3+m)(4+m)}$$

$$+\frac{(3bcC-3aCd+bBd(4+m))(a+b\tan(e+fx))^{1+m}(c+d\tan(e+fx))^2}{b^2f(3+m)(4+m)}$$

$$+\frac{C(a+b\tan(e+fx))^{1+m}(c+d\tan(e+fx))^3}{bf(4+m)}$$

$$+\frac{\left((iA+B-iC)(c-id)^3\right)\text{Subst}\left(\int\frac{(a-ibx)^m}{-1+x}\,dx,x,i\tan(e+fx)\right)}{2f}$$

$$-\frac{\left(i(A+iB-C)(c+id)^3\right)\text{Subst}\left(\int\frac{(a+ibx)^m}{-1+x}\,dx,x,-i\tan(e+fx)\right)}{2f}$$

$$= \frac{(bc(2+m)\left(b^2d(Bc+(A-C)d)(3+m)(4+m)+2(bc-ad)(3bcC-3aCd+bBd(4+m))\right)+d(}{}$$

$$-\frac{(iA+B-iC)(c-id)^3\,\text{Hypergeometric2F1}\left(1,1+m,2+m,\frac{a+b\tan(e+fx)}{a-ib}\right)(a+b\tan(e+fx))^{1}}{2(a-ib)f(1+m)}$$

$$-\frac{(A+iB-C)(c+id)^3\,\text{Hypergeometric2F1}\left(1,1+m,2+m,\frac{a+b\tan(e+fx)}{a+ib}\right)(a+b\tan(e+fx))^{1+}}{2(ia-b)f(1+m)}$$

$$+\frac{d(b^2d(Bc+(A-C)d)(3+m)(4+m)+2(bc-ad)(3bcC-3aCd+bBd(4+m)))\tan(e+fx)(}{b^3f(2+m)(3+m)(4+m)}$$

$$+\frac{(3bcC-3aCd+bBd(4+m))(a+b\tan(e+fx))^{1+m}(c+d\tan(e+fx))^2}{b^2f(3+m)(4+m)}$$

$$+\frac{C(a+b\tan(e+fx))^{1+m}(c+d\tan(e+fx))^3}{bf(4+m)}$$

Mathematica [B] (verified)

Both result and optimal contain complex but leaf count is larger than twice the leaf count of optimal. 1390 vs. $2(560) = 1120$.

Time = 6.54 (sec) , antiderivative size = 1390, normalized size of antiderivative = 2.48

$$\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\frac{C(a+b\tan(e+fx))^{1+m}(c+d\tan(e+fx))^3}{bf(4+m)}$$
$$+\frac{\frac{(3bcC-3aCd+bBd(4+m))(a+b\tan(e+fx))^{1+m}(c+d\tan(e+fx))^2}{bf(3+m)}+\frac{d\left(b^2d(Bc+(A-C)d)(3+m)(4+m)+2(bc-ad)(3bcC-3aCd+bBd(4+m))\right)\tan(\cdots}{bf(2+m)}}{\cdots}$$

```
[In] Integrate[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^3*(A + B*Tan[e + f*x]
 + C*Tan[e + f*x]^2),x]
```

```
[Out] (C*(a + b*Tan[e + f*x])^(1 + m)*(c + d*Tan[e + f*x])^3)/(b*f*(4 + m)) + (((
3*b*c*C - 3*a*C*d + b*B*d*(4 + m))*(a + b*Tan[e + f*x])^(1 + m)*(c + d*Tan[
e + f*x])^2)/(b*f*(3 + m)) + ((d*(b^2*d*(B*c + (A - C)*d)*(3 + m)*(4 + m) +
 2*(b*c - a*d)*(3*b*c*C - 3*a*C*d + b*B*d*(4 + m)))*Tan[e + f*x]*(a + b*Tan
[e + f*x])^(1 + m))/(b*f*(2 + m)) - (((-(b*c*(2 + m)*(b^2*d*(B*c + (A - C)*
d)*(3 + m)*(4 + m) + 2*(b*c - a*d)*(3*b*c*C - 3*a*C*d + b*B*d*(4 + m)))) +
d*(-(b^3*(2*c*(A - C)*d + B*(c^2 - d^2))*(2 + m)*(3 + m)*(4 + m)) + a*(b^2*
d*(B*c + (A - C)*d)*(3 + m)*(4 + m) + 2*(b*c - a*d)*(3*b*c*C - 3*a*C*d + b*
B*d*(4 + m)))))*(a + b*Tan[e + f*x])^(1 + m))/(b*f*(1 + m)) + ((I/2)*(a*d*(
b^2*d*(B*c + (A - C)*d)*(3 + m)*(4 + m) + 2*(b*c - a*d)*(3*b*c*C - 3*a*C*d
+ b*B*d*(4 + m))) + b*c*(2 + m)*(b^2*d*(B*c + (A - C)*d)*(3 + m)*(4 + m) +
2*(b*c - a*d)*(3*b*c*C - 3*a*C*d + b*B*d*(4 + m))) - b*c*(2 + m)*(-((2*a*d
+ b*c*(1 + m))*(3*b*c*C - 3*a*C*d + b*B*d*(4 + m))) + b*c*(3 + m)*(A*b*c*(4
 + m) - C*(3*a*d + b*c*(1 + m)))) - d*(-(b^3*(2*c*(A - C)*d + B*(c^2 - d^2)
)*(2 + m)*(3 + m)*(4 + m)) + a*(b^2*d*(B*c + (A - C)*d)*(3 + m)*(4 + m) + 2
*(b*c - a*d)*(3*b*c*C - 3*a*C*d + b*B*d*(4 + m)))) - I*b*(2 + m)*(b^2*c*(2*
c*(A - C)*d + B*(c^2 - d^2))*(3 + m)*(4 + m) - d*(b^2*d*(B*c + (A - C)*d)*(
3 + m)*(4 + m) + 2*(b*c - a*d)*(3*b*c*C - 3*a*C*d + b*B*d*(4 + m))) + d*(-(
(2*a*d + b*c*(1 + m))*(3*b*c*C - 3*a*C*d + b*B*d*(4 + m))) + b*c*(3 + m)*(A
*b*c*(4 + m) - C*(3*a*d + b*c*(1 + m))))))*Hypergeometric2F1[1, 1 + m, 2 +
m, ((-I)*a - I*b*Tan[e + f*x])/((-I)*a + b)]*(a + b*Tan[e + f*x])^(1 + m))/
((a + I*b)*f*(1 + m)) - ((I/2)*(a*d*(b^2*d*(B*c + (A - C)*d)*(3 + m)*(4 + m
) + 2*(b*c - a*d)*(3*b*c*C - 3*a*C*d + b*B*d*(4 + m))) + b*c*(2 + m)*(b^2*d
*(B*c + (A - C)*d)*(3 + m)*(4 + m) + 2*(b*c - a*d)*(3*b*c*C - 3*a*C*d + b*B
*d*(4 + m))) - b*c*(2 + m)*(-((2*a*d + b*c*(1 + m))*(3*b*c*C - 3*a*C*d + b*
B*d*(4 + m))) + b*c*(3 + m)*(A*b*c*(4 + m) - C*(3*a*d + b*c*(1 + m)))) - d*
(-(b^3*(2*c*(A - C)*d + B*(c^2 - d^2))*(2 + m)*(3 + m)*(4 + m)) + a*(b^2*d*
```

```
(B*c + (A - C)*d)*(3 + m)*(4 + m) + 2*(b*c - a*d)*(3*b*c*C - 3*a*C*d + b*B*
d*(4 + m)))) + I*b*(2 + m)*(b^2*c*(2*c*(A - C)*d + B*(c^2 - d^2))*(3 + m)*(
4 + m) - d*(b^2*d*(B*c + (A - C)*d)*(3 + m)*(4 + m) + 2*(b*c - a*d)*(3*b*c*
C - 3*a*C*d + b*B*d*(4 + m))) + d*(-((2*a*d + b*c*(1 + m))*(3*b*c*C - 3*a*C
*d + b*B*d*(4 + m))) + b*c*(3 + m)*(A*b*c*(4 + m) - C*(3*a*d + b*c*(1 + m))
))))*Hypergeometric2F1[1, 1 + m, 2 + m, -((I*a + I*b*Tan[e + f*x])/((-I)*a
- b))]*(a + b*Tan[e + f*x])^(1 + m))/((a - I*b)*f*(1 + m)))/(b*(2 + m)))/(b
*(3 + m)))/(b*(4 + m))
```

Maple [F]

$$\int \left(a+b\tan \left(fx+e\right)\right)^{m} \left(c+d\tan \left(fx+e\right)\right)^{3} \left(A+B\tan \left(fx+e\right)+C\tan \left(fx+e\right)^{2}\right)dx$$

```
[In] int((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x
)
```

```
[Out] int((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x
)
```

Fricas [F]

$$\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\int \left(C\tan \left(fx+e\right)^{2}+B\tan \left(fx+e\right)+A\right)\left(d\tan \left(fx+e\right)+c\right)^{3}\left(b\tan \left(fx+e\right)+a\right)^{m}dx$$

```
[In] integrate((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e
)^2),x, algorithm="fricas")
```

```
[Out] integral((C*d^3*tan(f*x + e)^5 + (3*C*c*d^2 + B*d^3)*tan(f*x + e)^4 + A*c^3
 + (3*C*c^2*d + 3*B*c*d^2 + A*d^3)*tan(f*x + e)^3 + (C*c^3 + 3*B*c^2*d + 3*
A*c*d^2)*tan(f*x + e)^2 + (B*c^3 + 3*A*c^2*d)*tan(f*x + e))*(b*tan(f*x + e)
 + a)^m, x)
```

Sympy [F]

$$\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\int (a+b\tan{(e+fx)})^m\,(c+d\tan{(e+fx)})^3\left(A+B\tan{(e+fx)}+C\tan^2{(e+fx)}\right)\,dx$$

```
[In] integrate((a+b*tan(f*x+e))**m*(c+d*tan(f*x+e))**3*(A+B*tan(f*x+e)+C*tan(f*x
+e)**2),x)
[Out] Integral((a + b*tan(e + f*x))**m*(c + d*tan(e + f*x))**3*(A + B*tan(e + f*x
) + C*tan(e + f*x)**2), x)
```

Maxima [F(-1)]

Timed out.

$$\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e
)^2),x, algorithm="maxima")
[Out] Timed out
```

Giac [F]

$$\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\int \left(C\tan{(fx+e)}^2+B\tan{(fx+e)}+A\right)(d\tan{(fx+e)}+c)^3(b\tan{(fx+e)}+a)^m\,dx$$

```
[In] integrate((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e
)^2),x, algorithm="giac")
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(d*tan(f*x + e) + c)^3*(b
*tan(f*x + e) + a)^m, x)
```

Mupad [F(-1)]

Timed out.

$$\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^3 \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\int (a+b\tan(e+f\,x))^m\,(c+d\tan(e+f\,x))^3\,\left(C\tan(e+f\,x)^2+B\tan(e+f\,x)+A\right)\,dx$$

```
[In] int((a + b*tan(e + f*x))^m*(c + d*tan(e + f*x))^3*(A + B*tan(e + f*x) + C*t
an(e + f*x)^2),x)
```

```
[Out] int((a + b*tan(e + f*x))^m*(c + d*tan(e + f*x))^3*(A + B*tan(e + f*x) + C*t
an(e + f*x)^2), x)
```

3.166 $\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^2 (A+B\tan(e$

Optimal result 2167
Rubi [A] (verified) 2168
Mathematica [A] (verified) 2171
Maple [F] 2172
Fricas [F] 2172
Sympy [F] 2173
Maxima [F] 2173
Giac [F] 2173
Mupad [F(-1)] 2174

Optimal result

Integrand size = 45, antiderivative size = 363

$$\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^2 \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$= \frac{(2a^2Cd^2-abd(2cC+Bd)(3+m)+b^2(2+m)\left(2c^2C+2Bcd(3+m)+(A-C)d^2(3+m)\right))(a+b\tan(}{b^3f(1+m)(2+m)(3+m)}$$
$$+\frac{(A-iB-C)(c-id)^2\,\text{Hypergeometric2F1}\left(1,1+m,2+m,\frac{a+b\tan(e+fx)}{a-ib}\right)(a+b\tan(e+fx))^{1+m}}{2(ia+b)f(1+m)}$$
$$+\frac{(iA-B-iC)(c+id)^2\,\text{Hypergeometric2F1}\left(1,1+m,2+m,\frac{a+b\tan(e+fx)}{a+ib}\right)(a+b\tan(e+fx))^{1+m}}{2(a+ib)f(1+m)}$$
$$-\frac{d(2aCd-b(2cC+Bd(3+m)))\tan(e+fx)(a+b\tan(e+fx))^{1+m}}{b^2f(2+m)(3+m)}$$
$$+\frac{C(a+b\tan(e+fx))^{1+m}(c+d\tan(e+fx))^2}{bf(3+m)}$$

```
[Out] (2*a^2*C*d^2-a*b*d*(B*d+2*C*c)*(3+m)+b^2*(2+m)*(2*c^2*C+2*B*c*d*(3+m)+(A-C)
*d^2*(3+m)))*(a+b*tan(f*x+e))^(1+m)/b^3/f/(1+m)/(2+m)/(3+m)+1/2*(A-I*B-C)*(
c-I*d)^2*hypergeom([1, 1+m],[2+m],(a+b*tan(f*x+e))/(a-I*b))*(a+b*tan(f*x+e)
)^(1+m)/(I*a+b)/f/(1+m)+1/2*(I*A-B-I*C)*(c+I*d)^2*hypergeom([1, 1+m],[2+m],
(a+b*tan(f*x+e))/(a+I*b))*(a+b*tan(f*x+e))^(1+m)/(a+I*b)/f/(1+m)-d*(2*C*a*d
-b*(2*C*c+B*d*(3+m)))*tan(f*x+e)*(a+b*tan(f*x+e))^(1+m)/b^2/f/(2+m)/(3+m)+C
*(a+b*tan(f*x+e))^(1+m)*(c+d*tan(f*x+e))^2/b/f/(3+m)
```

Rubi [A] (verified)

Time = 1.19 (sec) , antiderivative size = 360, normalized size of antiderivative = 0.99, number of steps used = 8, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.133, Rules used = {3728, 3718, 3711, 3620, 3618, 70}

$$\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\frac{(a+b\tan(e+fx))^{m+1}\left(2a^2Cd^2-abd(m+3)(Bd+2cC)+b^2(m+2)\left(d^2(m+3)(A-C)+2Bcd(m+3\right.\right.}{b^3f(m+1)(m+2)(m+3)}$$
$$+\frac{(c-id)^2(A-iB-C)(a+b\tan(e+fx))^{m+1}\,\text{Hypergeometric2F1}\left(1,m+1,m+2,\frac{a+b\tan(e+fx)}{a-ib}\right)}{2f(m+1)(b+ia)}$$
$$+\frac{(c+id)^2(iA-B-iC)(a+b\tan(e+fx))^{m+1}\,\text{Hypergeometric2F1}\left(1,m+1,m+2,\frac{a+b\tan(e+fx)}{a+ib}\right)}{2f(m+1)(a+ib)}$$
$$+\frac{d\tan(e+fx)(-2aCd+bBd(m+3)+2bcC)(a+b\tan(e+fx))^{m+1}}{b^2f(m+2)(m+3)}$$
$$+\frac{C(c+d\tan(e+fx))^2(a+b\tan(e+fx))^{m+1}}{bf(m+3)}$$

```
[In] Int[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*T
an[e + f*x]^2),x]
```

```
[Out] ((2*a^2*C*d^2 - a*b*d*(2*c*C + B*d)*(3 + m) + b^2*(2 + m)*(2*c^2*C + 2*B*c*
d*(3 + m) + (A - C)*d^2*(3 + m)))*(a + b*Tan[e + f*x])^(1 + m))/(b^3*f*(1 +
 m)*(2 + m)*(3 + m)) + ((A - I*B - C)*(c - I*d)^2*Hypergeometric2F1[1, 1 +
m, 2 + m, (a + b*Tan[e + f*x])/(a - I*b)]*(a + b*Tan[e + f*x])^(1 + m))/(2*
(I*a + b)*f*(1 + m)) + ((I*A - B - I*C)*(c + I*d)^2*Hypergeometric2F1[1, 1
+ m, 2 + m, (a + b*Tan[e + f*x])/(a + I*b)]*(a + b*Tan[e + f*x])^(1 + m))/(
2*(a + I*b)*f*(1 + m)) + (d*(2*b*c*C - 2*a*C*d + b*B*d*(3 + m))*Tan[e + f*x
]*(a + b*Tan[e + f*x])^(1 + m))/(b^2*f*(2 + m)*(3 + m)) + (C*(a + b*Tan[e +
 f*x])^(1 + m)*(c + d*Tan[e + f*x])^2)/(b*f*(3 + m))
```

Rule 70

```
Int[((a_) + (b_.)*(x_))^(m_)*((c_) + (d_.)*(x_))^(n_), x_Symbol] :> Simp[(b
*c - a*d)^n*((a + b*x)^(m + 1)/(b^(n + 1)*(m + 1)))*Hypergeometric2F1[-n, m
 + 1, m + 2, (-d)*((a + b*x)/(b*c - a*d))], x] /; FreeQ[{a, b, c, d, m}, x]
 && NeQ[b*c - a*d, 0] &&  !IntegerQ[m] && IntegerQ[n]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
```

```
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3728

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*(a + b*Tan[e + f*x])^m*((c + d*Tan[
e + f*x])^(n + 1)/(d*f*(m + n + 1))), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\text{integral} = \frac{C(a+b\tan(e+fx))^{1+m}(c+d\tan(e+fx))^2}{bf(3+m)} + \frac{\int(a+b\tan(e+fx))^m(c+d\tan(e+fx))\left(Abc(3+m)-C(2ad+bc(1+m))+b(Bc+(A-C)d)(3+\right.}{b(3+m)}$$

$$= \frac{d(2bcC-2aCd+bBd(3+m))\tan(e+fx)(a+b\tan(e+fx))^{1+m}}{b^2f(2+m)(3+m)} + \frac{C(a+b\tan(e+fx))^{1+m}(c+d\tan(e+fx))^2}{bf(3+m)} - \frac{\int(a+b\tan(e+fx))^m\left(ad(2bcC-2aCd+bBd(3+m))-bc(2+m)(Abc(3+m)-C(2ad+bc\right.}{}$$

$$= \frac{\left(2a^2Cd^2-abd(2cC+Bd)(3+m)+b^2(2+m)\left(2c^2C+2Bcd(3+m)+(A-C)d^2(3+m)\right)\right)(a+}{b^3f(1+m)(2+m)(3+m)} + \frac{d(2bcC-2aCd+bBd(3+m))\tan(e+fx)(a+b\tan(e+fx))^{1+m}}{b^2f(2+m)(3+m)} + \frac{C(a+b\tan(e+fx))^{1+m}(c+d\tan(e+fx))^2}{bf(3+m)} - \frac{\int(a+b\tan(e+fx))^m\left(-b^2\left(Ac^2-c^2C-2Bcd-Ad^2+Cd^2\right)(2+m)(3+m)-b^2(2c(A-C)d\right.}{b^2(2+m)(3+m)}$$

$$= \frac{\left(2a^2Cd^2-abd(2cC+Bd)(3+m)+b^2(2+m)\left(2c^2C+2Bcd(3+m)+(A-C)d^2(3+m)\right)\right)(a+}{b^3f(1+m)(2+m)(3+m)} + \frac{d(2bcC-2aCd+bBd(3+m))\tan(e+fx)(a+b\tan(e+fx))^{1+m}}{b^2f(2+m)(3+m)} + \frac{C(a+b\tan(e+fx))^{1+m}(c+d\tan(e+fx))^2}{bf(3+m)} + \frac{1}{2}\left((A-iB-C)(c-id)^2\right)\int(1+i\tan(e+fx))(a+b\tan(e+fx))^m\,dx + \frac{1}{2}\left((A+iB-C)(c+id)^2\right)\int(1-i\tan(e+fx))(a+b\tan(e+fx))^m\,dx$$

$$= \frac{(2a^2Cd^2 - abd(2cC + Bd)(3 + m) + b^2(2 + m)\left(2c^2C + 2Bcd(3 + m) + (A - C)d^2(3 + m)\right))(a -}{b^3f(1 + m)(2 + m)(3 + m)}$$
$$+ \frac{d(2bcC - 2aCd + bBd(3 + m))\tan(e + fx)(a + b\tan(e + fx))^{1+m}}{b^2f(2 + m)(3 + m)}$$
$$+ \frac{C(a + b\tan(e + fx))^{1+m}(c + d\tan(e + fx))^2}{bf(3 + m)}$$
$$+ \frac{((iA + B - iC)(c - id)^2)\,\mathrm{Subst}\left(\int \frac{(a-ibx)^m}{-1+x}\,dx, x, i\tan(e + fx)\right)}{2f}$$
$$- \frac{(i(A + iB - C)(c + id)^2)\,\mathrm{Subst}\left(\int \frac{(a+ibx)^m}{-1+x}\,dx, x, -i\tan(e + fx)\right)}{2f}$$

$$= \frac{(2a^2Cd^2 - abd(2cC + Bd)(3 + m) + b^2(2 + m)\left(2c^2C + 2Bcd(3 + m) + (A - C)d^2(3 + m)\right))(a -}{b^3f(1 + m)(2 + m)(3 + m)}$$
$$- \frac{(iA + B - iC)(c - id)^2\,\mathrm{Hypergeometric2F1}\left(1, 1 + m, 2 + m, \frac{a+b\tan(e+fx)}{a-ib}\right)(a + b\tan(e + fx)}{2(a - ib)f(1 + m)}$$
$$- \frac{(A + iB - C)(c + id)^2\,\mathrm{Hypergeometric2F1}\left(1, 1 + m, 2 + m, \frac{a+b\tan(e+fx)}{a+ib}\right)(a + b\tan(e + fx))}{2(ia - b)f(1 + m)}$$
$$+ \frac{d(2bcC - 2aCd + bBd(3 + m))\tan(e + fx)(a + b\tan(e + fx))^{1+m}}{b^2f(2 + m)(3 + m)}$$
$$+ \frac{C(a + b\tan(e + fx))^{1+m}(c + d\tan(e + fx))^2}{bf(3 + m)}$$

Mathematica [A] (verified)

Time = 6.37 (sec) , antiderivative size = 505, normalized size of antiderivative = 1.39

$$\int (a + b\tan(e + fx))^m(c + d\tan(e + fx))^2\left(A + B\tan(e + fx) + C\tan^2(e + fx)\right)\,dx$$
$$= \frac{C(a + b\tan(e + fx))^{1+m}(c + d\tan(e + fx))^2}{bf(3 + m)}$$
$$+ \frac{\frac{d(2bcC-2aCd+bBd(3+m))\tan(e+fx)(a+b\tan(e+fx))^{1+m}}{bf(2+m)} - \frac{\left(-bc(2+m)(2bcC-2aCd+bBd(3+m))-d\left(b^2(Bc+(A-C)d)(2+m)(3+m)-a(2bcC-\right.\right.}{bf(1+m)}}{}$$

```
[In] Integrate[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^2*(A + B*Tan[e + f*x]
 + C*Tan[e + f*x]^2),x]
```

```
[Out] (C*(a + b*Tan[e + f*x])^(1 + m)*(c + d*Tan[e + f*x])^2)/(b*f*(3 + m)) + ((d
*(2*b*c*C - 2*a*C*d + b*B*d*(3 + m))*Tan[e + f*x]*(a + b*Tan[e + f*x])^(1 +
 m))/(b*f*(2 + m)) - (((-(b*c*(2 + m)*(2*b*c*C - 2*a*C*d + b*B*d*(3 + m)))
```

```
- d*(b^2*(B*c + (A - C)*d)*(2 + m)*(3 + m) - a*(2*b*c*C - 2*a*C*d + b*B*d*(
3 + m))))*(a + b*Tan[e + f*x])^(1 + m))/(b*f*(1 + m)) + ((I/2)*(-(b^2*(A*c^
2 - c^2*C - 2*B*c*d - A*d^2 + C*d^2)*(2 + m)*(3 + m)) - I*b^2*(2*c*(A - C)*
d + B*(c^2 - d^2))*(2 + m)*(3 + m))*Hypergeometric2F1[1, 1 + m, 2 + m, ((-I
)*a - I*b*Tan[e + f*x])/((-I)*a + b)]*(a + b*Tan[e + f*x])^(1 + m))/((a + I
*b)*f*(1 + m)) - ((I/2)*(-(b^2*(A*c^2 - c^2*C - 2*B*c*d - A*d^2 + C*d^2)*(2
 + m)*(3 + m)) + I*b^2*(2*c*(A - C)*d + B*(c^2 - d^2))*(2 + m)*(3 + m))*Hyp
ergeometric2F1[1, 1 + m, 2 + m, -((I*a + I*b*Tan[e + f*x])/((-I)*a - b))]*(
a + b*Tan[e + f*x])^(1 + m))/((a - I*b)*f*(1 + m)))/(b*(2 + m)))/(b*(3 + m)
)
```

Maple [F]

$$\int (a+b\tan (fx+e))^{m} (c+d\tan (fx+e))^{2} \left(A+B\tan (fx+e)+C\tan (fx+e)^{2}\right) dx$$

```
[In] int((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x
)
```

```
[Out] int((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x
)
```

Fricas [F]

$$\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^2 \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\int \left(C\tan (fx+e)^{2}+B\tan (fx+e)+A\right)(d\tan (fx+e)+c)^{2}(b\tan (fx+e)+a)^{m}\,dx$$

```
[In] integrate((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e
)^2),x, algorithm="fricas")
```

```
[Out] integral((C*d^2*tan(f*x + e)^4 + (2*C*c*d + B*d^2)*tan(f*x + e)^3 + A*c^2 +
 (C*c^2 + 2*B*c*d + A*d^2)*tan(f*x + e)^2 + (B*c^2 + 2*A*c*d)*tan(f*x + e))
*(b*tan(f*x + e) + a)^m, x)
```

Sympy [F]

$$\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^2 \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right) dx$$
$$= \int (a+b\tan(e+fx))^m (c+d\tan(e+fx))^2 \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right) dx$$

```
[In] integrate((a+b*tan(f*x+e))**m*(c+d*tan(f*x+e))**2*(A+B*tan(f*x+e)+C*tan(f*x
+e)**2),x)
[Out] Integral((a + b*tan(e + f*x))**m*(c + d*tan(e + f*x))**2*(A + B*tan(e + f*x
) + C*tan(e + f*x)**2), x)
```

Maxima [F]

$$\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^2 \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right) dx$$
$$= \int (C\tan(fx+e)^2+B\tan(fx+e)+A)(d\tan(fx+e)+c)^2(b\tan(fx+e)+a)^m\, dx$$

```
[In] integrate((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e
)^2),x, algorithm="maxima")
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(d*tan(f*x + e) + c)^2*(b
*tan(f*x + e) + a)^m, x)
```

Giac [F]

$$\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^2 \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right) dx$$
$$= \int (C\tan(fx+e)^2+B\tan(fx+e)+A)(d\tan(fx+e)+c)^2(b\tan(fx+e)+a)^m\, dx$$

```
[In] integrate((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e
)^2),x, algorithm="giac")
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(d*tan(f*x + e) + c)^2*(b
*tan(f*x + e) + a)^m, x)
```

Mupad [F(-1)]

Timed out.

$$\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\int (a+b\tan(e+f\,x))^m\,(c+d\tan(e+f\,x))^2\,\left(C\tan(e+f\,x)^2+B\tan(e+f\,x)+A\right)\,dx$$

```
[In] int((a + b*tan(e + f*x))^m*(c + d*tan(e + f*x))^2*(A + B*tan(e + f*x) + C*t
an(e + f*x)^2),x)
```

```
[Out] int((a + b*tan(e + f*x))^m*(c + d*tan(e + f*x))^2*(A + B*tan(e + f*x) + C*t
an(e + f*x)^2), x)
```

3.167 $\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))\left(A+B\tan(e+\right.$

Optimal result 2175
Rubi [A] (verified) 2176
Mathematica [A] (verified) 2178
Maple [F] 2179
Fricas [F] 2179
Sympy [F] 2179
Maxima [F] 2180
Giac [F] 2180
Mupad [F(-1)] 2180

Optimal result

Integrand size = 43, antiderivative size = 247

$$\begin{aligned}
&\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx \\
&= -\frac{(aCd-b(cC+Bd)(2+m))(a+b\tan(e+fx))^{1+m}}{b^2f(1+m)(2+m)} \\
&\quad+\frac{(A-iB-C)(c-id)\,\text{Hypergeometric2F1}\left(1,1+m,2+m,\frac{a+b\tan(e+fx)}{a-ib}\right)(a+b\tan(e+fx))^{1+m}}{2(ia+b)f(1+m)} \\
&\quad-\frac{(A+iB-C)(c+id)\,\text{Hypergeometric2F1}\left(1,1+m,2+m,\frac{a+b\tan(e+fx)}{a+ib}\right)(a+b\tan(e+fx))^{1+m}}{2(ia-b)f(1+m)} \\
&\quad+\frac{Cd\tan(e+fx)(a+b\tan(e+fx))^{1+m}}{bf(2+m)}
\end{aligned}$$

```
[Out] -(C*a*d-b*(B*d+C*c)*(2+m))*(a+b*tan(f*x+e))^(1+m)/b^2/f/(1+m)/(2+m)+1/2*(A-
I*B-C)*(c-I*d)*hypergeom([1, 1+m],[2+m],(a+b*tan(f*x+e))/(a-I*b))*(a+b*tan(
f*x+e))^(1+m)/(I*a+b)/f/(1+m)-1/2*(A+I*B-C)*(c+I*d)*hypergeom([1, 1+m],[2+m
],(a+b*tan(f*x+e))/(a+I*b))*(a+b*tan(f*x+e))^(1+m)/(I*a-b)/f/(1+m)+C*d*tan(
f*x+e)*(a+b*tan(f*x+e))^(1+m)/b/f/(2+m)
```

Rubi [A] (verified)

Time = 0.59 (sec) , antiderivative size = 247, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.116, Rules used = {3718, 3711, 3620, 3618, 70}

$$\int (a+b\tan(e+fx))^m(c+d\tan(e+fx)) \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right) dx$$
$$=\frac{(c-id)(A-iB-C)(a+b\tan(e+fx))^{m+1}\,\text{Hypergeometric2F1}\left(1,m+1,m+2,\frac{a+b\tan(e+fx)}{a-ib}\right)}{2f(m+1)(b+ia)}$$
$$-\frac{(c+id)(A+iB-C)(a+b\tan(e+fx))^{m+1}\,\text{Hypergeometric2F1}\left(1,m+1,m+2,\frac{a+b\tan(e+fx)}{a+ib}\right)}{2f(m+1)(-b+ia)}$$
$$-\frac{(aCd-b(m+2)(Bd+cC))(a+b\tan(e+fx))^{m+1}}{b^2f(m+1)(m+2)}$$
$$+\frac{Cd\tan(e+fx)(a+b\tan(e+fx))^{m+1}}{bf(m+2)}$$

```
[In] Int[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan
[e + f*x]^2),x]
```

```
[Out] -(((a*C*d - b*(c*C + B*d)*(2 + m))*(a + b*Tan[e + f*x])^(1 + m))/(b^2*f*(1
+ m)*(2 + m))) + ((A - I*B - C)*(c - I*d)*Hypergeometric2F1[1, 1 + m, 2 + m
, (a + b*Tan[e + f*x])/(a - I*b)]*(a + b*Tan[e + f*x])^(1 + m))/(2*(I*a + b
)*f*(1 + m)) - ((A + I*B - C)*(c + I*d)*Hypergeometric2F1[1, 1 + m, 2 + m,
(a + b*Tan[e + f*x])/(a + I*b)]*(a + b*Tan[e + f*x])^(1 + m))/(2*(I*a - b)*
f*(1 + m)) + (C*d*Tan[e + f*x]*(a + b*Tan[e + f*x])^(1 + m))/(b*f*(2 + m))
```

Rule 70

```
Int[((a_) + (b_.)*(x_))^(m_)*((c_) + (d_.)*(x_))^(n_), x_Symbol] :> Simp[(b
*c - a*d)^n*((a + b*x)^(m + 1)/(b^(n + 1)*(m + 1)))*Hypergeometric2F1[-n, m
 + 1, m + 2, (-d)*((a + b*x)/(b*c - a*d))], x] /; FreeQ[{a, b, c, d, m}, x]
 && NeQ[b*c - a*d, 0] &&  !IntegerQ[m] && IntegerQ[n]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
```

```
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3718

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[b*C*Tan[e + f*x]*((c + d*Tan[e + f*x])^(n +
 1)/(d*f*(n + 2))), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rubi steps

$$\text{integral} = \frac{Cd\tan(e+fx)(a+b\tan(e+fx))^{1+m}}{bf(2+m)} - \frac{\int(a+b\tan(e+fx))^m\left(aCd-Abc(2+m)-b(Bc+(A-C)d)(2+m)\tan(e+fx)+(aCd-b(cC+\right.}{b(2+m)}$$

$$= -\frac{(aCd-b(cC+Bd)(2+m))(a+b\tan(e+fx))^{1+m}}{b^2f(1+m)(2+m)} + \frac{Cd\tan(e+fx)(a+b\tan(e+fx))^{1+m}}{bf(2+m)} - \frac{\int(a+b\tan(e+fx))^m(-b(Ac-cC-Bd)(2+m)-b(Bc+(A-C)d)(2+m)\tan(e+fx))\,d}{b(2+m)}$$

$$= -\frac{(aCd-b(cC+Bd)(2+m))(a+b\tan(e+fx))^{1+m}}{b^2f(1+m)(2+m)} + \frac{Cd\tan(e+fx)(a+b\tan(e+fx))^{1+m}}{bf(2+m)} + \frac{1}{2}((A-iB-C)(c-id))\int(1+i\tan(e+fx))(a+b\tan(e+fx))^m\,dx + \frac{1}{2}((A+iB-C)(c+id))\int(1-i\tan(e+fx))(a+b\tan(e+fx))^m\,dx$$

$$= -\frac{(aCd - b(cC + Bd)(2+m))(a + b\tan(e+fx))^{1+m}}{b^2 f(1+m)(2+m)} + \frac{Cd\tan(e+fx)(a+b\tan(e+fx))^{1+m}}{bf(2+m)} - \frac{(i(A+iB-C)(c+id))\text{Subst}\left(\int \frac{(a+ibx)^m}{-1+x}\,dx, x, -i\tan(e+fx)\right)}{2f} + \frac{((A-iB-C)(ic+d))\text{Subst}\left(\int \frac{(a-ibx)^m}{-1+x}\,dx, x, i\tan(e+fx)\right)}{2f}$$

$$= -\frac{(aCd - b(cC + Bd)(2+m))(a + b\tan(e+fx))^{1+m}}{b^2 f(1+m)(2+m)} - \frac{(A-iB-C)(ic+d)\,\text{Hypergeometric2F1}\left(1, 1+m, 2+m, \frac{a+b\tan(e+fx)}{a-ib}\right)(a+b\tan(e+fx))^{1+}}{2(a-ib)f(1+m)} - \frac{(A+iB-C)(c+id)\,\text{Hypergeometric2F1}\left(1, 1+m, 2+m, \frac{a+b\tan(e+fx)}{a+ib}\right)(a+b\tan(e+fx))^{1+}}{2(ia-b)f(1+m)} + \frac{Cd\tan(e+fx)(a+b\tan(e+fx))^{1+m}}{bf(2+m)}$$

Mathematica [A] (verified)

Time = 3.27 (sec) , antiderivative size = 202, normalized size of antiderivative = 0.82

$$\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \frac{(a+b\tan(e+fx))^{1+m}\left(\frac{-2aCd+2b(cC+Bd)(2+m)}{b(1+m)} - \frac{ib(A-iB-C)(c-id)(2+m)\,\text{Hypergeometric2F1}\left(1,1+m,2+m,\frac{a+b\tan(e+fx)}{a-ib}\right)}{(a-ib)(1+m)}\right.}{2bf(2+m)}$$

```
[In] Integrate[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])*(A + B*Tan[e + f*x] +
 C*Tan[e + f*x]^2),x]
```

```
[Out] ((a + b*Tan[e + f*x])^(1 + m)*((-2*a*C*d + 2*b*(c*C + B*d)*(2 + m))/(b*(1 +
 m)) - (I*b*(A - I*B - C)*(c - I*d)*(2 + m)*Hypergeometric2F1[1, 1 + m, 2 +
 m, (a + b*Tan[e + f*x])/(a - I*b)])/((a - I*b)*(1 + m)) + (I*b*(A + I*B -
C)*(c + I*d)*(2 + m)*Hypergeometric2F1[1, 1 + m, 2 + m, (a + b*Tan[e + f*x]
)/(a + I*b)])/((a + I*b)*(1 + m)) + 2*C*d*Tan[e + f*x]))/(2*b*f*(2 + m))
```

Maple [F]

$$\int (a+b\tan (fx+e))^{m} (c+d\tan (fx+e)) \left(A+B\tan (fx+e)+C\tan (fx+e)^{2}\right) dx$$

```
[In] int((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x)
[Out] int((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x)
```

Fricas [F]

$$\int (a+b\tan(e+fx))^m(c+d\tan(e+fx)) \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right) \,dx = \int (C\tan (fx+e)^{2}+B\tan (fx+e)+A)(d\tan (fx+e)+c)(b\tan (fx+e)+a)^{m} \,dx$$

```
[In] integrate((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x, algorithm="fricas")
[Out] integral((C*d*tan(f*x + e)^3 + (C*c + B*d)*tan(f*x + e)^2 + A*c + (B*c + A*
d)*tan(f*x + e))*(b*tan(f*x + e) + a)^m, x)
```

Sympy [F]

$$\int (a+b\tan(e+fx))^m(c+d\tan(e+fx)) \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right) \,dx = \int \left(a+b\tan\left(e+fx\right)\right)^{m} \left(c+d\tan\left(e+fx\right)\right) \left(A+B\tan\left(e+fx\right)+C\tan^{2}\left(e+fx\right)\right)\, dx$$

```
[In] integrate((a+b*tan(f*x+e))**m*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)
**2),x)
[Out] Integral((a + b*tan(e + f*x))**m*(c + d*tan(e + f*x))*(A + B*tan(e + f*x) +
 C*tan(e + f*x)**2), x)
```

Maxima [F]

$$\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\int (C\tan(fx+e)^2+B\tan(fx+e)+A)(d\tan(fx+e)+c)(b\tan(fx+e)+a)^m\,dx$$

```
[In] integrate((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x, algorithm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(d*tan(f*x + e) + c)*(b*t
an(f*x + e) + a)^m, x)
```

Giac [F]

$$\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\int (C\tan(fx+e)^2+B\tan(fx+e)+A)(d\tan(fx+e)+c)(b\tan(fx+e)+a)^m\,dx$$

```
[In] integrate((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x, algorithm="giac")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(d*tan(f*x + e) + c)*(b*t
an(f*x + e) + a)^m, x)
```

Mupad [F(-1)]

Timed out.

$$\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\int (a+b\tan(e+f\,x))^m\,(c+d\tan(e+f\,x))\,\left(C\tan(e+f\,x)^2+B\tan(e+f\,x)+A\right)\,dx$$

```
[In] int((a + b*tan(e + f*x))^m*(c + d*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan
(e + f*x)^2),x)
```

```
[Out] int((a + b*tan(e + f*x))^m*(c + d*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan
(e + f*x)^2), x)
```

3.168 $\int (a+b\tan(e+fx))^m \left(A+B\tan(e+fx)+C\tan^2(e\right.$

Optimal result 2181
Rubi [A] (verified) 2182
Mathematica [A] (verified) 2183
Maple [F] 2184
Fricas [F] 2184
Sympy [F] 2184
Maxima [F] 2185
Giac [F] 2185
Mupad [F(-1)] 2185

Optimal result

Integrand size = 33, antiderivative size = 178

$$\int (a+b\tan(e+fx))^m \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right) dx = \frac{C(a+b\tan(e+fx))^{1+m}}{bf(1+m)} + \frac{(A-iB-C)\,\text{Hypergeometric2F1}\left(1,1+m,2+m,\frac{a+b\tan(e+fx)}{a-ib}\right)(a+b\tan(e+fx))^{1+m}}{2(ia+b)f(1+m)} + \frac{(iA-B-iC)\,\text{Hypergeometric2F1}\left(1,1+m,2+m,\frac{a+b\tan(e+fx)}{a+ib}\right)(a+b\tan(e+fx))^{1+m}}{2(a+ib)f(1+m)}$$

```
[Out] C*(a+b*tan(f*x+e))^(1+m)/b/f/(1+m)+1/2*(A-I*B-C)*hypergeom([1, 1+m],[2+m],(
a+b*tan(f*x+e))/(a-I*b))*(a+b*tan(f*x+e))^(1+m)/(I*a+b)/f/(1+m)+1/2*(I*A-B-
I*C)*hypergeom([1, 1+m],[2+m],(a+b*tan(f*x+e))/(a+I*b))*(a+b*tan(f*x+e))^(1
+m)/(a+I*b)/f/(1+m)
```

Rubi [A] (verified)

Time = 0.22 (sec) , antiderivative size = 178, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 4, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.121, Rules used = {3711, 3620, 3618, 70}

$$\int (a+b\tan(e+fx))^m \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\frac{(A-iB-C)(a+b\tan(e+fx))^{m+1}\,\text{Hypergeometric2F1}\left(1,m+1,m+2,\frac{a+b\tan(e+fx)}{a-ib}\right)}{2f(m+1)(b+ia)}$$
$$+\frac{(iA-B-iC)(a+b\tan(e+fx))^{m+1}\,\text{Hypergeometric2F1}\left(1,m+1,m+2,\frac{a+b\tan(e+fx)}{a+ib}\right)}{2f(m+1)(a+ib)}$$
$$+\frac{C(a+b\tan(e+fx))^{m+1}}{bf(m+1)}$$

```
[In] Int[(a + b*Tan[e + f*x])^m*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] (C*(a + b*Tan[e + f*x])^(1 + m))/(b*f*(1 + m)) + ((A - I*B - C)*Hypergeomet
ric2F1[1, 1 + m, 2 + m, (a + b*Tan[e + f*x])/(a - I*b)]*(a + b*Tan[e + f*x]
)^(1 + m))/(2*(I*a + b)*f*(1 + m)) + ((I*A - B - I*C)*Hypergeometric2F1[1,
1 + m, 2 + m, (a + b*Tan[e + f*x])/(a + I*b)]*(a + b*Tan[e + f*x])^(1 + m))
/(2*(a + I*b)*f*(1 + m))
```

Rule 70

```
Int[((a_) + (b_.)*(x_))^(m_)*((c_) + (d_.)*(x_))^(n_), x_Symbol] :> Simp[(b
*c - a*d)^n*((a + b*x)^(m + 1)/(b^(n + 1)*(m + 1)))*Hypergeometric2F1[-n, m
 + 1, m + 2, (-d)*((a + b*x)/(b*c - a*d))], x] /; FreeQ[{a, b, c, d, m}, x]
 && NeQ[b*c - a*d, 0] &&  !IntegerQ[m] && IntegerQ[n]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3711

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[C*((a +
 b*Tan[e + f*x])^(m + 1)/(b*f*(m + 1))), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rubi steps

$$
\begin{aligned}
\text{integral} &= \frac{C(a+b\tan(e+fx))^{1+m}}{bf(1+m)}+\int(a+b\tan(e+fx))^m(A-C+B\tan(e+fx))\,dx\\
&= \frac{C(a+b\tan(e+fx))^{1+m}}{bf(1+m)}\\
&\quad+\frac{1}{2}(A-iB-C)\int(1+i\tan(e+fx))(a+b\tan(e+fx))^m\,dx\\
&\quad+\frac{1}{2}(A+iB-C)\int(1-i\tan(e+fx))(a+b\tan(e+fx))^m\,dx\\
&= \frac{C(a+b\tan(e+fx))^{1+m}}{bf(1+m)}+\frac{(iA+B-iC)\text{Subst}\left(\int\frac{(a-ibx)^m}{-1+x}\,dx,x,i\tan(e+fx)\right)}{2f}\\
&\quad+\frac{(i(-A-iB+C))\text{Subst}\left(\int\frac{(a+ibx)^m}{-1+x}\,dx,x,-i\tan(e+fx)\right)}{2f}\\
&= \frac{C(a+b\tan(e+fx))^{1+m}}{bf(1+m)}\\
&\quad-\frac{(iA+B-iC)\,\text{Hypergeometric2F1}\left(1,1+m,2+m,\frac{a+b\tan(e+fx)}{a-ib}\right)(a+b\tan(e+fx))^{1+m}}{2(a-ib)f(1+m)}\\
&\quad+\frac{(iA-B-iC)\,\text{Hypergeometric2F1}\left(1,1+m,2+m,\frac{a+b\tan(e+fx)}{a+ib}\right)(a+b\tan(e+fx))^{1+m}}{2(a+ib)f(1+m)}
\end{aligned}
$$

Mathematica [A] (verified)

Time = 0.28 (sec) , antiderivative size = 135, normalized size of antiderivative = 0.76

$$
\int(a+b\tan(e+fx))^m\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx
$$

$$
=\frac{\left(\frac{2C}{b}-\frac{i(A-iB-C)\,\text{Hypergeometric2F1}\left(1,1+m,2+m,\frac{a+b\tan(e+fx)}{a-ib}\right)}{a-ib}+\frac{i(A+iB-C)\,\text{Hypergeometric2F1}\left(1,1+m,2+m,\frac{a+b\tan(e+fx)}{a+ib}\right)}{a+ib}\right)}{2f(1+m)}
$$

```
[In] Integrate[(a + b*Tan[e + f*x])^m*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] (((2*C)/b - (I*(A - I*B - C)*Hypergeometric2F1[1, 1 + m, 2 + m, (a + b*Tan[
e + f*x])/(a - I*b)])/(a - I*b) + (I*(A + I*B - C)*Hypergeometric2F1[1, 1 +
 m, 2 + m, (a + b*Tan[e + f*x])/(a + I*b)])/(a + I*b))*(a + b*Tan[e + f*x])
^(1 + m))/(2*f*(1 + m))
```

Maple [F]

$$\int (a+b\tan (fx+e))^m \left(A+B\tan (fx+e)+C\tan (fx+e)^2\right) dx$$

```
[In] int((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x)
```

```
[Out] int((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x)
```

Fricas [F]

$$\int (a+b\tan(e+fx))^m \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\int \left(C\tan (fx+e)^2+B\tan (fx+e)+A\right)(b\tan (fx+e)+a)^m\,dx$$

```
[In] integrate((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorithm="
fricas")
```

```
[Out] integral((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)^m, x)
```

Sympy [F]

$$\int (a+b\tan(e+fx))^m \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx$$
$$=\int (a+b\tan (e+fx))^m \left(A+B\tan (e+fx)+C\tan^2 (e+fx)\right)\,dx$$

```
[In] integrate((a+b*tan(f*x+e))**m*(A+B*tan(f*x+e)+C*tan(f*x+e)**2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**m*(A + B*tan(e + f*x) + C*tan(e + f*x)**2),
x)
```

Maxima [F]

$$\int (a+b\tan(e+fx))^m \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right) dx = \int (C\tan(fx+e)^2 + B\tan(fx+e) + A)(b\tan(fx+e)+a)^m dx$$

```
[In] integrate((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorithm="
maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)^m, x
)
```

Giac [F]

$$\int (a+b\tan(e+fx))^m \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right) dx = \int (C\tan(fx+e)^2 + B\tan(fx+e) + A)(b\tan(fx+e)+a)^m dx$$

```
[In] integrate((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorithm="
giac")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)^m, x
)
```

Mupad [F(-1)]

Timed out.

$$\int (a+b\tan(e+fx))^m \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right) dx = \int (a+b\tan(e+f\,x))^m \left(C\tan(e+f\,x)^2+B\tan(e+f\,x)+A\right) dx$$

```
[In] int((a + b*tan(e + f*x))^m*(A + B*tan(e + f*x) + C*tan(e + f*x)^2),x)
```

```
[Out] int((a + b*tan(e + f*x))^m*(A + B*tan(e + f*x) + C*tan(e + f*x)^2), x)
```

3.169 $\int \frac{(a+b\tan(e+fx))^m(A+B\tan(e+fx)+C\tan^2(e+fx))}{c+d\tan(e+fx)}\,dx$

Optimal result 2186
Rubi [A] (verified) 2186
Mathematica [A] (verified) 2189
Maple [F] 2189
Fricas [F] 2190
Sympy [F] 2190
Maxima [F] 2190
Giac [F] 2191
Mupad [F(-1)] 2191

Optimal result

Integrand size = 45, antiderivative size = 258

$$\int \frac{(a+b\tan(e+fx))^m\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{c+d\tan(e+fx)}\,dx = -\frac{(iA+B-iC)\,\text{Hypergeometric2F1}\left(1,1+m,2+m,\frac{a+b\tan(e+fx)}{a-ib}\right)(a+b\tan(e+fx))^{1+m}}{2(a-ib)(c-id)f(1+m)} - \frac{(A+iB-C)\,\text{Hypergeometric2F1}\left(1,1+m,2+m,\frac{a+b\tan(e+fx)}{a+ib}\right)(a+b\tan(e+fx))^{1+m}}{2(ia-b)(c+id)f(1+m)} + \frac{(c^2C-Bcd+Ad^2)\,\text{Hypergeometric2F1}\left(1,1+m,2+m,-\frac{d(a+b\tan(e+fx))}{bc-ad}\right)(a+b\tan(e+fx))^{1+m}}{(bc-ad)\,(c^2+d^2)\,f(1+m)}$$

```
[Out] -1/2*(I*A+B-I*C)*hypergeom([1, 1+m],[2+m],(a+b*tan(f*x+e))/(a-I*b))*(a+b*ta
n(f*x+e))^(1+m)/(a-I*b)/(c-I*d)/f/(1+m)-1/2*(A+I*B-C)*hypergeom([1, 1+m],[2
+m],(a+b*tan(f*x+e))/(a+I*b))*(a+b*tan(f*x+e))^(1+m)/(I*a-b)/(c+I*d)/f/(1+m
)+(A*d^2-B*c*d+C*c^2)*hypergeom([1, 1+m],[2+m],-d*(a+b*tan(f*x+e))/(-a*d+b*
c))*(a+b*tan(f*x+e))^(1+m)/(-a*d+b*c)/(c^2+d^2)/f/(1+m)
```

Rubi [A] (verified)

Time = 0.60 (sec) , antiderivative size = 258, normalized size of antiderivative = 1.00, number of steps used = 8, number of rules used = 5, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.111, Rules used

= {3734, 3620, 3618, 70, 3715}

$$\int \frac{(a+b\tan(e+fx))^m\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{c+d\tan(e+fx)}\,dx$$
$$=\frac{(Ad^2-Bcd+c^2C)\,(a+b\tan(e+fx))^{m+1}\,\mathrm{Hypergeometric2F1}\left(1,m+1,m+2,-\frac{d(a+b\tan(e+fx))}{bc-ad}\right)}{f(m+1)\,(c^2+d^2)\,(bc-ad)}$$
$$-\frac{(iA+B-iC)(a+b\tan(e+fx))^{m+1}\,\mathrm{Hypergeometric2F1}\left(1,m+1,m+2,\frac{a+b\tan(e+fx)}{a-ib}\right)}{2f(m+1)(a-ib)(c-id)}$$
$$-\frac{(A+iB-C)(a+b\tan(e+fx))^{m+1}\,\mathrm{Hypergeometric2F1}\left(1,m+1,m+2,\frac{a+b\tan(e+fx)}{a+ib}\right)}{2f(m+1)(-b+ia)(c+id)}$$

```
[In] Int[((a + b*Tan[e + f*x])^m*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c + d
*Tan[e + f*x]),x]
```

```
[Out] -1/2*((I*A + B - I*C)*Hypergeometric2F1[1, 1 + m, 2 + m, (a + b*Tan[e + f*x
])/(a - I*b)]*(a + b*Tan[e + f*x])^(1 + m))/((a - I*b)*(c - I*d)*f*(1 + m))
 - ((A + I*B - C)*Hypergeometric2F1[1, 1 + m, 2 + m, (a + b*Tan[e + f*x])/(
a + I*b)]*(a + b*Tan[e + f*x])^(1 + m))/(2*(I*a - b)*(c + I*d)*f*(1 + m)) +
 ((c^2*C - B*c*d + A*d^2)*Hypergeometric2F1[1, 1 + m, 2 + m, -((d*(a + b*Ta
n[e + f*x]))/(b*c - a*d))]*(a + b*Tan[e + f*x])^(1 + m))/((b*c - a*d)*(c^2
+ d^2)*f*(1 + m))
```

Rule 70

```
Int[((a_) + (b_.)*(x_))^(m_)*((c_) + (d_.)*(x_))^(n_), x_Symbol] :> Simp[(b
*c - a*d)^n*((a + b*x)^(m + 1)/(b^(n + 1)*(m + 1)))*Hypergeometric2F1[-n, m
 + 1, m + 2, (-d)*((a + b*x)/(b*c - a*d))], x] /; FreeQ[{a, b, c, d, m}, x]
 && NeQ[b*c - a*d, 0] &&  !IntegerQ[m] && IntegerQ[n]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3715

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_) + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :>
 Dist[A/f, Subst[Int[(a + b*x)^m*(c + d*x)^n, x], x, Tan[e + f*x]], x] /; F
reeQ[{a, b, c, d, e, f, A, C, m, n}, x] && EqQ[A, C]
```

Rule 3734

```
Int[(((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2))/((a_.) + (b_.)*tan[(e_.)
+ (f_.)*(x_)]), x_Symbol] :> Dist[1/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n
*Simp[b*B + a*(A - C) + (a*B - b*(A - C))*Tan[e + f*x], x], x], x] + Dist[(
A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n*((1 + Tan[e
+ f*x]^2)/(a + b*Tan[e + f*x])), x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C
, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&
 !GtQ[n, 0] &&  !LeQ[n, -1]
```

Rubi steps

$$\text{integral} = \frac{\int (a+b\tan(e+fx))^m (Ac-cC+Bd+(Bc-(A-C)d)\tan(e+fx))\,dx}{c^2+d^2} + \frac{(c^2C-Bcd+Ad^2)\int \frac{(a+b\tan(e+fx))^m(1+\tan^2(e+fx))}{c+d\tan(e+fx)}\,dx}{c^2+d^2}$$

$$= \frac{(A-iB-C)\int (1+i\tan(e+fx))(a+b\tan(e+fx))^m\,dx}{2(c-id)} + \frac{(A+iB-C)\int (1-i\tan(e+fx))(a+b\tan(e+fx))^m\,dx}{2(c+id)} + \frac{(c^2C-Bcd+Ad^2)\,\text{Subst}\left(\int \frac{(a+bx)^m}{c+dx}\,dx, x, \tan(e+fx)\right)}{(c^2+d^2)\,f}$$

$$= \frac{(c^2C-Bcd+Ad^2)\,\text{Hypergeometric2F1}\left(1,1+m,2+m,-\frac{d(a+b\tan(e+fx))}{bc-ad}\right)(a+b\tan(e+fx))^{1+m}}{(bc-ad)\,(c^2+d^2)\,f(1+m)} + \frac{(iA+B-iC)\text{Subst}\left(\int \frac{(a-ibx)^m}{-1+x}\,dx, x, i\tan(e+fx)\right)}{2(c-id)f} - \frac{(i(A+iB-C))\text{Subst}\left(\int \frac{(a+ibx)^m}{-1+x}\,dx, x, -i\tan(e+fx)\right)}{2(c+id)f}$$

$$= -\frac{(iA+B-iC)\,\text{Hypergeometric2F1}\left(1,1+m,2+m,\frac{a+b\tan(e+fx)}{a-ib}\right)(a+b\tan(e+fx))^{1+m}}{2(a-ib)(c-id)f(1+m)} - \frac{(A+iB-C)\,\text{Hypergeometric2F1}\left(1,1+m,2+m,\frac{a+b\tan(e+fx)}{a+ib}\right)(a+b\tan(e+fx))^{1+m}}{2(ia-b)(c+id)f(1+m)} + \frac{(c^2C-Bcd+Ad^2)\,\text{Hypergeometric2F1}\left(1,1+m,2+m,-\frac{d(a+b\tan(e+fx))}{bc-ad}\right)(a+b\tan(e+fx)}{(bc-ad)(c^2+d^2)f(1+m)}$$

Mathematica [A] (verified)

Time = 1.27 (sec) , antiderivative size = 204, normalized size of antiderivative = 0.79

$$\int \frac{(a+b\tan(e+fx))^m\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{c+d\tan(e+fx)}\,dx = \frac{\left(\frac{(A-iB-C)(-ic+d)\,\text{Hypergeometric2F1}\left(1,1+m,2+m,\frac{a+b\tan(e+fx)}{a-ib}\right)}{a-ib} + \frac{(A+iB-C)(ic+d)\,\text{Hypergeometric2F1}\left(1,1+m,2+m,\frac{a+b\tan(e+fx}{a+ib}\right.}{a+ib}\right.}{2(c^2+d^2)f(1+m)}$$

```
[In] Integrate[((a + b*Tan[e + f*x])^m*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(c + d*Tan[e + f*x]),x]
```

```
[Out] ((((A - I*B - C)*((-I)*c + d)*Hypergeometric2F1[1, 1 + m, 2 + m, (a + b*Tan
[e + f*x])/(a - I*b)])/(a - I*b) + ((A + I*B - C)*(I*c + d)*Hypergeometric2
F1[1, 1 + m, 2 + m, (a + b*Tan[e + f*x])/(a + I*b)])/(a + I*b) + (2*(c^2*C
- B*c*d + A*d^2)*Hypergeometric2F1[1, 1 + m, 2 + m, (d*(a + b*Tan[e + f*x])
)/(-(b*c) + a*d)])/(b*c - a*d))*(a + b*Tan[e + f*x])^(1 + m))/(2*(c^2 + d^2
)*f*(1 + m))
```

Maple [F]

$$\int \frac{(a+b\tan(fx+e))^m\left(A+B\tan(fx+e)+C\tan(fx+e)^2\right)}{c+d\tan(fx+e)}\,dx$$

```
[In] int((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e)),x)
```

```
[Out] int((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e)),x)
```

Fricas [F]

$$\int \frac{(a+b\tan(e+fx))^m\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{c+d\tan(e+fx)}\,dx = \int \frac{\left(C\tan\left(fx+e\right)^2+B\tan\left(fx+e\right)+A\right)\left(b\tan\left(fx+e\right)+a\right)^m}{d\tan\left(fx+e\right)+c}\,dx$$

```
[In] integrate((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
)),x, algorithm="fricas")
```

```
[Out] integral((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)^m/(d*
tan(f*x + e) + c), x)
```

Sympy [F]

$$\int \frac{(a+b\tan(e+fx))^m\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{c+d\tan(e+fx)}\,dx = \int \frac{\left(a+b\tan\left(e+fx\right)\right)^m\left(A+B\tan\left(e+fx\right)+C\tan^2\left(e+fx\right)\right)}{c+d\tan\left(e+fx\right)}\,dx$$

```
[In] integrate((a+b*tan(f*x+e))**m*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x
+e)),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**m*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/(
c + d*tan(e + f*x)), x)
```

Maxima [F]

$$\int \frac{(a+b\tan(e+fx))^m\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{c+d\tan(e+fx)}\,dx = \int \frac{\left(C\tan\left(fx+e\right)^2+B\tan\left(fx+e\right)+A\right)\left(b\tan\left(fx+e\right)+a\right)^m}{d\tan\left(fx+e\right)+c}\,dx$$

```
[In] integrate((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
)),x, algorithm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)^m/(d
*tan(f*x + e) + c), x)
```

Giac [F]

$$\int \frac{(a+b\tan(e+fx))^m (A+B\tan(e+fx)+C\tan^2(e+fx))}{c+d\tan(e+fx)} dx = \int \frac{\left(C\tan\left(fx+e\right)^2+B\tan\left(fx+e\right)+A\right)\left(b\tan\left(fx+e\right)+a\right)^m}{d\tan\left(fx+e\right)+c} dx$$

```
[In] integrate((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
)),x, algorithm="giac")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)^m/(d
*tan(f*x + e) + c), x)
```

Mupad [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^m (A+B\tan(e+fx)+C\tan^2(e+fx))}{c+d\tan(e+fx)} dx = \int \frac{(a+b\tan(e+f\,x))^m\,\left(C\tan(e+f\,x)^2+B\tan(e+f\,x)+A\right)}{c+d\tan\left(e+f\,x\right)} dx$$

```
[In] int(((a + b*tan(e + f*x))^m*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d
*tan(e + f*x)),x)
```

```
[Out] int(((a + b*tan(e + f*x))^m*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d
*tan(e + f*x)), x)
```

3.170 $\int \frac{(a+b\tan(e+fx))^m\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^2}\,dx$

Optimal result 2192
Rubi [A] (verified) 2193
Mathematica [A] (verified) 2196
Maple [F] 2197
Fricas [F] 2197
Sympy [F(-2)] 2197
Maxima [F] 2198
Giac [F] 2198
Mupad [F(-1)] 2198

Optimal result

Integrand size = 45, antiderivative size = 403

$$\int \frac{(a+b\tan(e+fx))^m\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^2}\,dx$$
$$= \frac{(A-iB-C)\,\text{Hypergeometric2F1}\left(1,1+m,2+m,\frac{a+b\tan(e+fx)}{a-ib}\right)(a+b\tan(e+fx))^{1+m}}{2(ia+b)(c-id)^2f(1+m)}$$
$$+ \frac{(iA-B-iC)\,\text{Hypergeometric2F1}\left(1,1+m,2+m,\frac{a+b\tan(e+fx)}{a+ib}\right)(a+b\tan(e+fx))^{1+m}}{2(a+ib)(c+id)^2f(1+m)}$$
$$- \frac{(ad^2(2c(A-C)d-B(c^2-d^2))-b(Ad^2(c^2(2-m)-d^2m)-Bcd(c^2(1-m)-d^2(1+m))-c^2C(c^2m}{(bc-ad)^2\,(c^2+d^2)}$$
$$+ \frac{(c^2C-Bcd+Ad^2)\,(a+b\tan(e+fx))^{1+m}}{(bc-ad)\,(c^2+d^2)\,f(c+d\tan(e+fx))}$$

```
[Out] 1/2*(A-I*B-C)*hypergeom([1, 1+m],[2+m],(a+b*tan(f*x+e))/(a-I*b))*(a+b*tan(f
*x+e))^(1+m)/(I*a+b)/(c-I*d)^2/f/(1+m)+1/2*(I*A-B-I*C)*hypergeom([1, 1+m],[
2+m],(a+b*tan(f*x+e))/(a+I*b))*(a+b*tan(f*x+e))^(1+m)/(a+I*b)/(c+I*d)^2/f/(
1+m)-(a*d^2*(2*c*(A-C)*d-B*(c^2-d^2))-b*(A*d^2*(c^2*(2-m)-d^2*m)-B*c*d*(c^2
*(1-m)-d^2*(1+m))-c^2*C*(c^2*m+d^2*(2+m))))*hypergeom([1, 1+m],[2+m],-d*(a+
b*tan(f*x+e))/(-a*d+b*c))*(a+b*tan(f*x+e))^(1+m)/(-a*d+b*c)^2/(c^2+d^2)^2/f
/(1+m)+(A*d^2-B*c*d+C*c^2)*(a+b*tan(f*x+e))^(1+m)/(-a*d+b*c)/(c^2+d^2)/f/(c
+d*tan(f*x+e))
```

Rubi [A] (verified)

Time = 1.32 (sec) , antiderivative size = 402, normalized size of antiderivative = 1.00, number of steps used = 9, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.133, Rules used = {3730, 3734, 3620, 3618, 70, 3715}

$$\int \frac{(a+b\tan(e+fx))^m\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^2}\,dx$$

$$= \frac{\left(Ad^2-Bcd+c^2C\right)(a+b\tan(e+fx))^{m+1}}{f\left(c^2+d^2\right)(bc-ad)(c+d\tan(e+fx))}$$

$$-\frac{(a+b\tan(e+fx))^{m+1}\left(ad^2\left(2cd(A-C)-B\left(c^2-d^2\right)\right)-b\left(Ac^2d^2(2-m)-Ad^4m-B(c^3d(1-m)-\right.\right.}{f(m+1)\left(c^2+a\right.}$$

$$+\frac{(A-iB-C)(a+b\tan(e+fx))^{m+1}\,\text{Hypergeometric2F1}\left(1,m+1,m+2,\frac{a+b\tan(e+fx)}{a-ib}\right)}{2f(m+1)(b+ia)(c-id)^2}$$

$$+\frac{(iA-B-iC)(a+b\tan(e+fx))^{m+1}\,\text{Hypergeometric2F1}\left(1,m+1,m+2,\frac{a+b\tan(e+fx)}{a+ib}\right)}{2f(m+1)(a+ib)(c+id)^2}$$

```
[In] Int[((a + b*Tan[e + f*x])^m*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c + d
*Tan[e + f*x])^2,x]
```

```
[Out] ((A - I*B - C)*Hypergeometric2F1[1, 1 + m, 2 + m, (a + b*Tan[e + f*x])/(a -
 I*b)]*(a + b*Tan[e + f*x])^(1 + m))/(2*(I*a + b)*(c - I*d)^2*f*(1 + m)) +
((I*A - B - I*C)*Hypergeometric2F1[1, 1 + m, 2 + m, (a + b*Tan[e + f*x])/(a
 + I*b)]*(a + b*Tan[e + f*x])^(1 + m))/(2*(a + I*b)*(c + I*d)^2*f*(1 + m))
- ((a*d^2*(2*c*(A - C)*d - B*(c^2 - d^2)) - b*(A*c^2*d^2*(2 - m) - c^4*C*m
- A*d^4*m - c^2*C*d^2*(2 + m) - B*(c^3*d*(1 - m) - c*d^3*(1 + m))))*Hyperge
ometric2F1[1, 1 + m, 2 + m, -((d*(a + b*Tan[e + f*x]))/(b*c - a*d))]*(a + b
*Tan[e + f*x])^(1 + m))/((b*c - a*d)^2*(c^2 + d^2)^2*f*(1 + m)) + ((c^2*C -
 B*c*d + A*d^2)*(a + b*Tan[e + f*x])^(1 + m))/((b*c - a*d)*(c^2 + d^2)*f*(c
 + d*Tan[e + f*x]))
```

Rule 70

```
Int[((a_) + (b_.)*(x_))^(m_)*((c_) + (d_.)*(x_))^(n_), x_Symbol] :> Simp[(b
*c - a*d)^n*((a + b*x)^(m + 1)/(b^(n + 1)*(m + 1)))*Hypergeometric2F1[-n, m
 + 1, m + 2, (-d)*((a + b*x)/(b*c - a*d))], x] /; FreeQ[{a, b, c, d, m}, x]
 && NeQ[b*c - a*d, 0] &&  !IntegerQ[m] && IntegerQ[n]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
```

```
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3715

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_) + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :>
 Dist[A/f, Subst[Int[(a + b*x)^m*(c + d*x)^n, x], x, Tan[e + f*x]], x] /; F
reeQ[{a, b, c, d, e, f, A, C, m, n}, x] && EqQ[A, C]
```

Rule 3730

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 - a*(b*B - a*C))*(a + b*Tan[e +
 f*x])^(m + 1)*((c + d*Tan[e + f*x])^(n + 1)/(f*(m + 1)*(b*c - a*d)*(a^2 +
b^2))), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3734

```
Int[(((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2))/((a_.) + (b_.)*tan[(e_.)
+ (f_.)*(x_)]), x_Symbol] :> Dist[1/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n
*Simp[b*B + a*(A - C) + (a*B - b*(A - C))*Tan[e + f*x], x], x], x] + Dist[(
A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n*((1 + Tan[e
+ f*x]^2)/(a + b*Tan[e + f*x])), x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C
, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&
 !GtQ[n, 0] &&  !LeQ[n, -1]
```

Rubi steps

$$\text{integral} = \frac{(c^2C - Bcd + Ad^2)(a + b\tan(e + fx))^{1+m}}{(bc - ad)(c^2 + d^2)f(c + d\tan(e + fx))}$$

$$+ \frac{\int \frac{(a+b\tan(e+fx))^m((cC-Bd)(ad-bc(1+m))-A(acd-b(c^2-d^2m))+(bc-ad)(Bc-(A-C)d)\tan(e+fx)-b(c^2C-Bcd+Ad^2)m\tan^2(e+fx}{c+d\tan(e+fx)}}{(bc - ad)(c^2 + d^2)}$$

$$= \frac{(c^2C - Bcd + Ad^2)(a + b\tan(e + fx))^{1+m}}{(bc - ad)(c^2 + d^2)f(c + d\tan(e + fx))}$$

$$+ \frac{\int (a + b\tan(e + fx))^m (-((bc - ad)(c^2C - 2Bcd - Cd^2 - A(c^2 - d^2))) - (bc - ad)(2c(A - C}{(bc - ad)(c^2 + d^2)^2}$$

$$+ \frac{(-cd(bc - ad)(Bc - (A - C)d) - bc^2(c^2C - Bcd + Ad^2)m + d^2((cC - Bd)(ad - bc(1 + m)) -}{(bc - ad)(c^2 + d^2)^2}$$

$$= \frac{(c^2C - Bcd + Ad^2)(a + b\tan(e + fx))^{1+m}}{(bc - ad)(c^2 + d^2)f(c + d\tan(e + fx))}$$

$$+ \frac{(A - iB - C)\int (1 + i\tan(e + fx))(a + b\tan(e + fx))^m \, dx}{2(c - id)^2}$$

$$+ \frac{(A + iB - C)\int (1 - i\tan(e + fx))(a + b\tan(e + fx))^m \, dx}{2(c + id)^2}$$

$$+ \frac{(-cd(bc - ad)(Bc - (A - C)d) - bc^2(c^2C - Bcd + Ad^2)m + d^2((cC - Bd)(ad - bc(1 + m)) -}{(bc - ad)(c^2 + d^2)^2 f}$$

$$=$$

$$- \frac{(ad^2(2c(A - C)d - B(c^2 - d^2)) - b(Ac^2d^2(2 - m) - c^4Cm - Ad^4m - c^2Cd^2(2 + m) - B(c^3d(}{(bc - ad)^2}$$

$$+ \frac{(c^2C - Bcd + Ad^2)(a + b\tan(e + fx))^{1+m}}{(bc - ad)(c^2 + d^2)f(c + d\tan(e + fx))}$$

$$+ \frac{(iA + B - iC)\text{Subst}\left(\int \frac{(a-ibx)^m}{-1+x}\,dx, x, i\tan(e + fx)\right)}{2(c - id)^2 f}$$

$$- \frac{(i(A + iB - C))\text{Subst}\left(\int \frac{(a+ibx)^m}{-1+x}\,dx, x, -i\tan(e + fx)\right)}{2(c + id)^2 f}$$

$$= -\frac{(iA+B-iC)\,\text{Hypergeometric2F1}\left(1,1+m,2+m,\frac{a+b\tan(e+fx)}{a-ib}\right)(a+b\tan(e+fx))^{1+m}}{2(a-ib)(c-id)^2f(1+m)} - \frac{(A+iB-C)\,\text{Hypergeometric2F1}\left(1,1+m,2+m,\frac{a+b\tan(e+fx)}{a+ib}\right)(a+b\tan(e+fx))^{1+m}}{2(ia-b)(c+id)^2f(1+m)} - \frac{(ad^2(2c(A-C)d-B(c^2-d^2))-b(Ac^2d^2(2-m)-c^4Cm-Ad^4m-c^2Cd^2(2+m)-B(c^3d(1}{(bc-ad)^2\,(} + \frac{(c^2C-Bcd+Ad^2)\,(a+b\tan(e+fx))^{1+m}}{(bc-ad)\,(c^2+d^2)\,f(c+d\tan(e+fx))}$$

Mathematica [A] (verified)

Time = 6.24 (sec) , antiderivative size = 563, normalized size of antiderivative = 1.40

$$\int \frac{(a+b\tan(e+fx))^m\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^2}\,dx = -\frac{(Ad^2-c(-cC+Bd))\,(a+b\tan(e+fx))^{1+m}}{(-bc+ad)\,(c^2+d^2)\,f(c+d\tan(e+fx))} - \frac{-\frac{(-cd(bc-ad)(Bc-(A-C)d)-bc^2(c^2C-Bcd+Ad^2)m+d^2((cC-Bd)(ad-bc(1+m))-A(acd-b(c^2-d^2m))))\,\text{Hypergeometric2F1}\left(1,1+m,2+\right.}{(-bc+ad)(c^2+d^2)f(1+m)}}{}$$

```
[In] Integrate[((a + b*Tan[e + f*x])^m*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(c + d*Tan[e + f*x])^2,x]
```

```
[Out] -(((A*d^2 - c*(-(c*C) + B*d))*(a + b*Tan[e + f*x])^(1 + m))/((-(b*c) + a*d)
*(c^2 + d^2)*f*(c + d*Tan[e + f*x]))) - (-(((-(c*d*(b*c - a*d)*(B*c - (A -
C)*d)) - b*c^2*(c^2*C - B*c*d + A*d^2)*m + d^2*((c*C - B*d)*(a*d - b*c*(1 +
 m)) - A*(a*c*d - b*(c^2 - d^2*m))))*Hypergeometric2F1[1, 1 + m, 2 + m, (d*
(a + b*Tan[e + f*x]))/(-(b*c) + a*d)]*(a + b*Tan[e + f*x])^(1 + m))/((-(b*c
) + a*d)*(c^2 + d^2)*f*(1 + m))) + (((I/2)*(-((b*c - a*d)*(c^2*C - 2*B*c*d
- C*d^2 - A*(c^2 - d^2))) - I*(b*c - a*d)*(2*c*(A - C)*d - B*(c^2 - d^2)))*
Hypergeometric2F1[1, 1 + m, 2 + m, ((-I)*a - I*b*Tan[e + f*x])/((-I)*a + b)
]*(a + b*Tan[e + f*x])^(1 + m))/((a + I*b)*f*(1 + m)) - ((I/2)*(-((b*c - a*
d)*(c^2*C - 2*B*c*d - C*d^2 - A*(c^2 - d^2))) + I*(b*c - a*d)*(2*c*(A - C)*
d - B*(c^2 - d^2)))*Hypergeometric2F1[1, 1 + m, 2 + m, -((I*a + I*b*Tan[e +
 f*x])/((-I)*a - b))]*(a + b*Tan[e + f*x])^(1 + m))/((a - I*b)*f*(1 + m)))/
(c^2 + d^2))/((-(b*c) + a*d)*(c^2 + d^2))
```

Maple [F]

$$\int \frac{(a+b\tan(fx+e))^m \left(A+B\tan(fx+e)+C\tan(fx+e)^2\right)}{(c+d\tan(fx+e))^2}\,dx$$

```
[In] int((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^2,x
)
```

```
[Out] int((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^2,x
)
```

Fricas [F]

$$\int \frac{(a+b\tan(e+fx))^m (A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^2}\,dx = \int \frac{\left(C\tan(fx+e)^2+B\tan(fx+e)+A\right)(b\tan(fx+e)+a)^m}{(d\tan(fx+e)+c)^2}\,dx$$

```
[In] integrate((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^2,x, algorithm="fricas")
```

```
[Out] integral((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)^m/(d^
2*tan(f*x + e)^2 + 2*c*d*tan(f*x + e) + c^2), x)
```

Sympy [F(-2)]

Exception generated.

$$\int \frac{(a+b\tan(e+fx))^m (A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^2}\,dx = \text{Exception raised: HeuristicGCDFailed}$$

```
[In] integrate((a+b*tan(f*x+e))**m*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x
+e))**2,x)
```

```
[Out] Exception raised: HeuristicGCDFailed >> no luck
```

Maxima [F]

$$\int \frac{(a+b\tan(e+fx))^m\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^2}\,dx = \int \frac{\left(C\tan\left(fx+e\right)^2+B\tan\left(fx+e\right)+A\right)\left(b\tan\left(fx+e\right)+a\right)^m}{\left(d\tan\left(fx+e\right)+c\right)^2}\,dx$$

[In] integrate((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^2,x, algorithm="maxima")

[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)^m/(d*tan(f*x + e) + c)^2, x)

Giac [F]

$$\int \frac{(a+b\tan(e+fx))^m\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^2}\,dx = \int \frac{\left(C\tan\left(fx+e\right)^2+B\tan\left(fx+e\right)+A\right)\left(b\tan\left(fx+e\right)+a\right)^m}{\left(d\tan\left(fx+e\right)+c\right)^2}\,dx$$

[In] integrate((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^2,x, algorithm="giac")

[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)^m/(d*tan(f*x + e) + c)^2, x)

Mupad [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^m\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^2}\,dx = \int \frac{(a+b\tan(e+f\,x))^m\,\left(C\tan(e+f\,x)^2+B\tan(e+f\,x)+A\right)}{\left(c+d\tan\left(e+f\,x\right)\right)^2}\,dx$$

[In] int(((a + b*tan(e + f*x))^m*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d*tan(e + f*x))^2,x)

[Out] int(((a + b*tan(e + f*x))^m*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d*tan(e + f*x))^2, x)

3.171 $\int \frac{(a+b\tan(e+fx))^m(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^3}\,dx$

Optimal result 2199
Rubi [A] (verified) 2200
Mathematica [B] (verified) 2203
Maple [F] 2205
Fricas [F] 2205
Sympy [F] 2205
Maxima [F(-1)] 2206
Giac [F] 2206
Mupad [F(-1)] 2206

Optimal result

Integrand size = 45, antiderivative size = 702

$$\int \frac{(a+b\tan(e+fx))^m\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^3}\,dx$$
$$= \frac{(A-iB-C)\,\text{Hypergeometric2F1}\left(1,1+m,2+m,\frac{a+b\tan(e+fx)}{a-ib}\right)(a+b\tan(e+fx))^{1+m}}{2(ia+b)(c-id)^3f(1+m)}$$
$$+ \frac{(A+iB-C)\,\text{Hypergeometric2F1}\left(1,1+m,2+m,\frac{a+b\tan(e+fx)}{a+ib}\right)(a+b\tan(e+fx))^{1+m}}{2(a+ib)(ic-d)^3f(1+m)}$$
$$+ \frac{(2a^2d^3((A-C)d(3c^2-d^2)-B(c^3-3cd^2))-2abd^2(B(6c^2d^2-c^4(2-m)-d^4m)+2c(A-C)d(c^2(3}{}$$
$$+ \frac{(c^2C-Bcd+Ad^2)\,(a+b\tan(e+fx))^{1+m}}{2(bc-ad)\,(c^2+d^2)\,f(c+d\tan(e+fx))^2}$$
$$- \frac{(2ad^2(2c(A-C)d-B(c^2-d^2))-b(c^4C(1-m)+Ad^4(1-m)-Bc^3d(3-m)+Bcd^3(1+m)+c^2d}{2(bc-ad)^2\,(c^2+d^2)^2\,f(c+d\tan(e+fx))}$$

```
[Out] 1/2*(A-I*B-C)*hypergeom([1, 1+m],[2+m],(a+b*tan(f*x+e))/(a-I*b))*(a+b*tan(f
*x+e))^(1+m)/(I*a+b)/(c-I*d)^3/f/(1+m)+1/2*(A+I*B-C)*hypergeom([1, 1+m],[2+
m],(a+b*tan(f*x+e))/(a+I*b))*(a+b*tan(f*x+e))^(1+m)/(a+I*b)/(I*c-d)^3/f/(1+
m)+1/2*(2*a^2*d^3*((A-C)*d*(3*c^2-d^2)-B*(c^3-3*c*d^2))-2*a*b*d^2*(B*(6*c^2
*d^2-c^4*(2-m)-d^4*m)+2*c*(A-C)*d*(c^2*(3-m)-d^2*(1+m)))-b^2*(A*d^2*(d^4*(1
-m)*m+2*c^2*d^2*(-m^2+3*m+1)-c^4*(m^2-5*m+6))+B*c*d*(d^4*m*(1+m)-2*c^2*d^2*
(-m^2+m+3)+c^4*(m^2-3*m+2))+c^2*C*(c^4*(1-m)*m+2*c^2*d^2*(-m^2-m+3)-d^4*(m^
2+3*m+2))))*hypergeom([1, 1+m],[2+m],-d*(a+b*tan(f*x+e))/(-a*d+b*c))*(a+b*t
an(f*x+e))^(1+m)/(-a*d+b*c)^3/(c^2+d^2)^3/f/(1+m)+1/2*(A*d^2-B*c*d+C*c^2)*(
a+b*tan(f*x+e))^(1+m)/(-a*d+b*c)/(c^2+d^2)/f/(c+d*tan(f*x+e))^2-1/2*(2*a*d^
2*(2*c*(A-C)*d-B*(c^2-d^2))-b*(c^4*C*(1-m)+A*d^4*(1-m)-B*c^3*d*(3-m)+B*c*d^
```

```
3*(1+m)+c^2*d^2*(A*(5-m)-C*(3+m))))*(a+b*tan(f*x+e))^(1+m)/(-a*d+b*c)^2/(c^
2+d^2)^2/f/(c+d*tan(f*x+e))
```

Rubi [A] (verified)

Time = 3.24 (sec) , antiderivative size = 702, normalized size of antiderivative = 1.00, number of steps used = 10, number of rules used = 6, $\frac{\text{number of rules}}{\text{integrand size}} = 0.133$, Rules used = {3730, 3734, 3620, 3618, 70, 3715}

$$\int \frac{(a+b\tan(e+fx))^m\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^3}\,dx$$

$$= \frac{(a+b\tan(e+fx))^{m+1}\,(2a^2d^3(d(A-C)\,(3c^2-d^2)-B(c^3-3cd^2))-2abd^2(2cd(A-C)\,(c^2(3-m)-d^2(}{}$$

$$+\frac{(Ad^2-Bcd+c^2C)\,(a+b\tan(e+fx))^{m+1}}{2f\,(c^2+d^2)\,(bc-ad)(c+d\tan(e+fx))^2}$$

$$-\frac{(a+b\tan(e+fx))^{m+1}\,(2ad^2(2cd(A-C)-B(c^2-d^2))-b(c^2d^2(A(5-m)-C(m+3))+Ad^4(1-m}{2f\,(c^2+d^2)^2\,(bc-ad)^2(c+d\tan(e+fx))}$$

$$+\frac{(A-iB-C)(a+b\tan(e+fx))^{m+1}\,\text{Hypergeometric2F1}\left(1,m+1,m+2,\frac{a+b\tan(e+fx)}{a-ib}\right)}{2f(m+1)(b+ia)(c-id)^3}$$

$$+\frac{(A+iB-C)(a+b\tan(e+fx))^{m+1}\,\text{Hypergeometric2F1}\left(1,m+1,m+2,\frac{a+b\tan(e+fx)}{a+ib}\right)}{2f(m+1)(a+ib)(-d+ic)^3}$$

```
[In] Int[((a + b*Tan[e + f*x])^m*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c + d
*Tan[e + f*x])^3,x]
```

```
[Out] ((A - I*B - C)*Hypergeometric2F1[1, 1 + m, 2 + m, (a + b*Tan[e + f*x])/(a -
 I*b)]*(a + b*Tan[e + f*x])^(1 + m))/(2*(I*a + b)*(c - I*d)^3*f*(1 + m)) +
((A + I*B - C)*Hypergeometric2F1[1, 1 + m, 2 + m, (a + b*Tan[e + f*x])/(a +
 I*b)]*(a + b*Tan[e + f*x])^(1 + m))/(2*(a + I*b)*(I*c - d)^3*f*(1 + m)) +
((2*a^2*d^3*((A - C)*d*(3*c^2 - d^2) - B*(c^3 - 3*c*d^2)) - 2*a*b*d^2*(B*(6
*c^2*d^2 - c^4*(2 - m) - d^4*m) + 2*c*(A - C)*d*(c^2*(3 - m) - d^2*(1 + m))
) - b^2*(A*d^2*(d^4*(1 - m)*m + 2*c^2*d^2*(1 + 3*m - m^2) - c^4*(6 - 5*m +
m^2)) + B*(c*d^5*m*(1 + m) - 2*c^3*d^3*(3 + m - m^2) + c^5*d*(2 - 3*m + m^2
)) + c^2*C*(c^4*(1 - m)*m + 2*c^2*d^2*(3 - m - m^2) - d^4*(2 + 3*m + m^2)))
)*Hypergeometric2F1[1, 1 + m, 2 + m, -((d*(a + b*Tan[e + f*x]))/(b*c - a*d)
)]*(a + b*Tan[e + f*x])^(1 + m))/(2*(b*c - a*d)^3*(c^2 + d^2)^3*f*(1 + m))
+ ((c^2*C - B*c*d + A*d^2)*(a + b*Tan[e + f*x])^(1 + m))/(2*(b*c - a*d)*(c^
2 + d^2)*f*(c + d*Tan[e + f*x])^2) - ((2*a*d^2*(2*c*(A - C)*d - B*(c^2 - d^
2)) - b*(c^4*C*(1 - m) + A*d^4*(1 - m) - B*c^3*d*(3 - m) + B*c*d^3*(1 + m)
+ c^2*d^2*(A*(5 - m) - C*(3 + m))))*(a + b*Tan[e + f*x])^(1 + m))/(2*(b*c -
 a*d)^2*(c^2 + d^2)^2*f*(c + d*Tan[e + f*x]))
```

Rule 70

```
Int[((a_) + (b_.)*(x_))^(m_)*((c_) + (d_.)*(x_))^(n_), x_Symbol] :> Simp[(b
*c - a*d)^n*((a + b*x)^(m + 1)/(b^(n + 1)*(m + 1)))*Hypergeometric2F1[-n, m
 + 1, m + 2, (-d)*((a + b*x)/(b*c - a*d))], x] /; FreeQ[{a, b, c, d, m}, x]
 && NeQ[b*c - a*d, 0] &&  !IntegerQ[m] && IntegerQ[n]
```

Rule 3618

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[c*(d/f), Subst[Int[(a + (b/d)*x)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3620

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3715

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_) + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :>
 Dist[A/f, Subst[Int[(a + b*x)^m*(c + d*x)^n, x], x, Tan[e + f*x]], x] /; F
reeQ[{a, b, c, d, e, f, A, C, m, n}, x] && EqQ[A, C]
```

Rule 3730

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[(A*b^2 - a*(b*B - a*C))*(a + b*Tan[e +
 f*x])^(m + 1)*((c + d*Tan[e + f*x])^(n + 1)/(f*(m + 1)*(b*c - a*d)*(a^2 +
b^2))), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3734

```
Int[(((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2))/((a_.) + (b_.)*tan[(e_.)
+ (f_.)*(x_)]), x_Symbol] :> Dist[1/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n
```

```
*Simp[b*B + a*(A - C) + (a*B - b*(A - C))*Tan[e + f*x], x], x], x] + Dist[(
A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n*((1 + Tan[e
+ f*x]^2)/(a + b*Tan[e + f*x])), x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C
, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&
 !GtQ[n, 0] &&  !LeQ[n, -1]
```

Rubi steps

$$\text{integral} = \frac{(c^2C - Bcd + Ad^2)\,(a + b\tan(e + fx))^{1+m}}{2(bc - ad)\,(c^2 + d^2)\,f(c + d\tan(e + fx))^2} + \frac{\int \frac{(a+b\tan(e+fx))^m\,(A(2c(bc-ad)+bd^2(1-m))+(cC-Bd)(2ad-bc(1+m))+2(bc-ad)(Bc-(A-C)d)\tan(e+fx)+b(c^2C-Bcd+Ad^2)(1-m)\,t}{(c+d\tan(e+fx))^2}}{2(bc - ad)\,(c^2 + d^2)}$$

$$= \frac{(c^2C - Bcd + Ad^2)\,(a + b\tan(e + fx))^{1+m}}{2(bc - ad)\,(c^2 + d^2)\,f(c + d\tan(e + fx))^2} - \frac{(2ad^2(2c(A - C)d - B(c^2 - d^2)) - b(c^4C(1 - m) + Ad^4(1 - m) - Bc^3d(3 - m) + Bcd^3(1 + m)}{2(bc - ad)^2\,(c^2 + d^2)^2\,f(c + d\tan(e + fx}$$
$$+ \frac{\int (a+b\tan(e+fx))^m\,(-((2d(bc-ad)(Bc-(A-C)d)-bc(c^2C-Bcd+Ad^2)(1-m))(ad-bc(1+m)))-(acd-b(c^2-d^2m))\,(A(2c(bc-ad}{}$$

$$= \frac{(c^2C - Bcd + Ad^2)\,(a + b\tan(e + fx))^{1+m}}{2(bc - ad)\,(c^2 + d^2)\,f(c + d\tan(e + fx))^2} - \frac{(2ad^2(2c(A - C)d - B(c^2 - d^2)) - b(c^4C(1 - m) + Ad^4(1 - m) - Bc^3d(3 - m) + Bcd^3(1 + m)}{2(bc - ad)^2\,(c^2 + d^2)^2\,f(c + d\tan(e + fx}$$
$$+ \frac{\int (a + b\tan(e + fx))^m\,(2(bc - ad)^2\,(Ac^3 - c^3C + 3Bc^2d - 3Acd^2 + 3cCd^2 - Bd^3) - 2(bc - ad)^2}{2(bc - ad)^2\,(c^2 + d^2)^3}$$
$$+ \frac{(2a^2d^3((A - C)d(3c^2 - d^2) - B(c^3 - 3cd^2)) - 2abd^2(B(6c^2d^2 - c^4(2 - m) - d^4m) + 2c(A - C)}{}$$

$$= \frac{(c^2C - Bcd + Ad^2)\,(a + b\tan(e + fx))^{1+m}}{2(bc - ad)\,(c^2 + d^2)\,f(c + d\tan(e + fx))^2} - \frac{(2ad^2(2c(A - C)d - B(c^2 - d^2)) - b(c^4C(1 - m) + Ad^4(1 - m) - Bc^3d(3 - m) + Bcd^3(1 + m)}{2(bc - ad)^2\,(c^2 + d^2)^2\,f(c + d\tan(e + fx}$$
$$+ \frac{(A - iB - C)\int (1 + i\tan(e + fx))(a + b\tan(e + fx))^m\,dx}{2(c - id)^3} + \frac{(A + iB - C)\int (1 - i\tan(e + fx))(a + b\tan(e + fx))^m\,dx}{2(c + id)^3}$$
$$+ \frac{(2a^2d^3((A - C)d(3c^2 - d^2) - B(c^3 - 3cd^2)) - 2abd^2(B(6c^2d^2 - c^4(2 - m) - d^4m) + 2c(A - C)}{}$$

$$= \frac{(2a^2d^3((A-C)d(3c^2-d^2)-B(c^3-3cd^2))-2abd^2(B(6c^2d^2-c^4(2-m)-d^4m)+2c(A-C)d}{}$$

$$+\frac{(c^2C-Bcd+Ad^2)\,(a+b\tan(e+fx))^{1+m}}{2(bc-ad)\,(c^2+d^2)\,f(c+d\tan(e+fx))^2}$$

$$-\frac{(2ad^2(2c(A-C)d-B(c^2-d^2))-b(c^4C(1-m)+Ad^4(1-m)-Bc^3d(3-m)+Bcd^3(1+m}{2(bc-ad)^2\,(c^2+d^2)^2\,f(c+d\tan(e+f}$$

$$-\frac{(i(A+iB-C))\text{Subst}\left(\int\frac{(a+ibx)^m}{-1+x}\,dx,x,-i\tan(e+fx)\right)}{2(c+id)^3f}$$

$$+\frac{(A-iB-C)\text{Subst}\left(\int\frac{(a-ibx)^m}{-1+x}\,dx,x,i\tan(e+fx)\right)}{2(ic+d)^3f}$$

$$=$$

$$-\frac{(A-iB-C)\,\text{Hypergeometric2F1}\left(1,1+m,2+m,\frac{a+b\tan(e+fx)}{a-ib}\right)(a+b\tan(e+fx))^{1+m}}{2(a-ib)(ic+d)^3f(1+m)}$$

$$-\frac{(A+iB-C)\,\text{Hypergeometric2F1}\left(1,1+m,2+m,\frac{a+b\tan(e+fx)}{a+ib}\right)(a+b\tan(e+fx))^{1+m}}{2(ia-b)(c+id)^3f(1+m)}$$

$$+\frac{(2a^2d^3((A-C)d(3c^2-d^2)-B(c^3-3cd^2))-2abd^2(B(6c^2d^2-c^4(2-m)-d^4m)+2c(A-C}{}$$

$$+\frac{(c^2C-Bcd+Ad^2)\,(a+b\tan(e+fx))^{1+m}}{2(bc-ad)\,(c^2+d^2)\,f(c+d\tan(e+fx))^2}$$

$$-\frac{(2ad^2(2c(A-C)d-B(c^2-d^2))-b(c^4C(1-m)+Ad^4(1-m)-Bc^3d(3-m)+Bcd^3(1+m}{2(bc-ad)^2\,(c^2+d^2)^2\,f(c+d\tan(e+f}$$

Mathematica [B] (verified)

Both result and optimal contain complex but leaf count is larger than twice the leaf count of optimal. 2238 vs. 2(702) = 1404.

Time = 6.33 (sec) , antiderivative size = 2238, normalized size of antiderivative = 3.19

$$\int\frac{(a+b\tan(e+fx))^m\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^3}\,dx$$
$$= \text{Result too large to show}$$

```
[In] Integrate[((a + b*Tan[e + f*x])^m*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(c + d*Tan[e + f*x])^3,x]

[Out] -1/2*((A*d^2 - c*(-(c*C) + B*d))*(a + b*Tan[e + f*x])^(1 + m))/((-(b*c) + a
*d)*(c^2 + d^2)*f*(c + d*Tan[e + f*x])^2) - (-(((-(c*(2*d*(b*c - a*d)*(B*c
- (A - C)*d) - b*c*(c^2*C - B*c*d + A*d^2)*(1 - m))) + d^2*(A*(2*c*(b*c - a
```

```
*d) + b*d^2*(1 - m)) + (c*C - B*d)*(2*a*d - b*c*(1 + m))))*(a + b*Tan[e + f
*x])^(1 + m))/((-(b*c) + a*d)*(c^2 + d^2)*f*(c + d*Tan[e + f*x]))) - (-(((-
(c*d*(-(b*c) + a*d)*(-2*c*(b*c - a*d)*(B*c - (A - C)*d) - b*d*(c^2*C - B*c*
d + A*d^2)*(1 - m) + d*(A*(2*c*(b*c - a*d) + b*d^2*(1 - m)) + (c*C - B*d)*(
2*a*d - b*c*(1 + m))))) - b*c^2*m*(-(c*(2*d*(b*c - a*d)*(B*c - (A - C)*d) -
 b*c*(c^2*C - B*c*d + A*d^2)*(1 - m))) + d^2*(A*(2*c*(b*c - a*d) + b*d^2*(1
 - m)) + (c*C - B*d)*(2*a*d - b*c*(1 + m)))) + d^2*((2*d*(b*c - a*d)*(B*c -
 (A - C)*d) - b*c*(c^2*C - B*c*d + A*d^2)*(1 - m))*(-(a*d) + b*c*(1 + m)) +
 (-(c*(-(b*c) + a*d)) - b*d^2*m)*(A*(2*c*(b*c - a*d) + b*d^2*(1 - m)) + (c*
C - B*d)*(2*a*d - b*c*(1 + m)))))*Hypergeometric2F1[1, 1 + m, 2 + m, (d*(a
+ b*Tan[e + f*x]))/(-(b*c) + a*d)]*(a + b*Tan[e + f*x])^(1 + m))/((-(b*c) +
 a*d)*(c^2 + d^2)*f*(1 + m))) + (((I/2)*(d*(-(b*c) + a*d)*(-2*c*(b*c - a*d)
*(B*c - (A - C)*d) - b*d*(c^2*C - B*c*d + A*d^2)*(1 - m) + d*(A*(2*c*(b*c -
 a*d) + b*d^2*(1 - m)) + (c*C - B*d)*(2*a*d - b*c*(1 + m)))) + c*((2*d*(b*c
 - a*d)*(B*c - (A - C)*d) - b*c*(c^2*C - B*c*d + A*d^2)*(1 - m))*(-(a*d) +
b*c*(1 + m)) + (-(c*(-(b*c) + a*d)) - b*d^2*m)*(A*(2*c*(b*c - a*d) + b*d^2*
(1 - m)) + (c*C - B*d)*(2*a*d - b*c*(1 + m))) + b*m*(-(c*(2*d*(b*c - a*d)*(
B*c - (A - C)*d) - b*c*(c^2*C - B*c*d + A*d^2)*(1 - m))) + d^2*(A*(2*c*(b*c
 - a*d) + b*d^2*(1 - m)) + (c*C - B*d)*(2*a*d - b*c*(1 + m))))) + I*(c*(-(b
*c) + a*d)*(-2*c*(b*c - a*d)*(B*c - (A - C)*d) - b*d*(c^2*C - B*c*d + A*d^2
)*(1 - m) + d*(A*(2*c*(b*c - a*d) + b*d^2*(1 - m)) + (c*C - B*d)*(2*a*d - b
*c*(1 + m)))) - d*((2*d*(b*c - a*d)*(B*c - (A - C)*d) - b*c*(c^2*C - B*c*d
+ A*d^2)*(1 - m))*(-(a*d) + b*c*(1 + m)) + (-(c*(-(b*c) + a*d)) - b*d^2*m)*
(A*(2*c*(b*c - a*d) + b*d^2*(1 - m)) + (c*C - B*d)*(2*a*d - b*c*(1 + m))) +
 b*m*(-(c*(2*d*(b*c - a*d)*(B*c - (A - C)*d) - b*c*(c^2*C - B*c*d + A*d^2)*
(1 - m))) + d^2*(A*(2*c*(b*c - a*d) + b*d^2*(1 - m)) + (c*C - B*d)*(2*a*d -
 b*c*(1 + m)))))))*Hypergeometric2F1[1, 1 + m, 2 + m, ((-I)*a - I*b*Tan[e +
 f*x])/((-I)*a + b)]*(a + b*Tan[e + f*x])^(1 + m))/((a + I*b)*f*(1 + m)) -
((I/2)*(d*(-(b*c) + a*d)*(-2*c*(b*c - a*d)*(B*c - (A - C)*d) - b*d*(c^2*C -
 B*c*d + A*d^2)*(1 - m) + d*(A*(2*c*(b*c - a*d) + b*d^2*(1 - m)) + (c*C - B
*d)*(2*a*d - b*c*(1 + m)))) + c*((2*d*(b*c - a*d)*(B*c - (A - C)*d) - b*c*(
c^2*C - B*c*d + A*d^2)*(1 - m))*(-(a*d) + b*c*(1 + m)) + (-(c*(-(b*c) + a*d
)) - b*d^2*m)*(A*(2*c*(b*c - a*d) + b*d^2*(1 - m)) + (c*C - B*d)*(2*a*d - b
*c*(1 + m))) + b*m*(-(c*(2*d*(b*c - a*d)*(B*c - (A - C)*d) - b*c*(c^2*C - B
*c*d + A*d^2)*(1 - m))) + d^2*(A*(2*c*(b*c - a*d) + b*d^2*(1 - m)) + (c*C -
 B*d)*(2*a*d - b*c*(1 + m))))) - I*(c*(-(b*c) + a*d)*(-2*c*(b*c - a*d)*(B*c
 - (A - C)*d) - b*d*(c^2*C - B*c*d + A*d^2)*(1 - m) + d*(A*(2*c*(b*c - a*d)
 + b*d^2*(1 - m)) + (c*C - B*d)*(2*a*d - b*c*(1 + m)))) - d*((2*d*(b*c - a*
d)*(B*c - (A - C)*d) - b*c*(c^2*C - B*c*d + A*d^2)*(1 - m))*(-(a*d) + b*c*(
1 + m)) + (-(c*(-(b*c) + a*d)) - b*d^2*m)*(A*(2*c*(b*c - a*d) + b*d^2*(1 -
m)) + (c*C - B*d)*(2*a*d - b*c*(1 + m))) + b*m*(-(c*(2*d*(b*c - a*d)*(B*c -
 (A - C)*d) - b*c*(c^2*C - B*c*d + A*d^2)*(1 - m))) + d^2*(A*(2*c*(b*c - a*
d) + b*d^2*(1 - m)) + (c*C - B*d)*(2*a*d - b*c*(1 + m)))))))*Hypergeometric
2F1[1, 1 + m, 2 + m, -((I*a + I*b*Tan[e + f*x])/((-I)*a - b))]*(a + b*Tan[e
 + f*x])^(1 + m))/((a - I*b)*f*(1 + m)))/(c^2 + d^2))/((-(b*c) + a*d)*(c^2
```

```
+ d^2)))/(2*(-(b*c) + a*d)*(c^2 + d^2))
```

Maple [F]

$$\int \frac{(a+b\tan(fx+e))^m \left(A+B\tan(fx+e)+C\tan(fx+e)^2\right)}{(c+d\tan(fx+e))^3}\,dx$$

```
[In] int((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^3,x
)
```

```
[Out] int((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^3,x
)
```

Fricas [F]

$$\int \frac{(a+b\tan(e+fx))^m\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^3}\,dx = \int \frac{\left(C\tan(fx+e)^2+B\tan(fx+e)+A\right)(b\tan(fx+e)+a)^m}{(d\tan(fx+e)+c)^3}\,dx$$

```
[In] integrate((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^3,x, algorithm="fricas")
```

```
[Out] integral((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)^m/(d^
3*tan(f*x + e)^3 + 3*c*d^2*tan(f*x + e)^2 + 3*c^2*d*tan(f*x + e) + c^3), x)
```

Sympy [F]

$$\int \frac{(a+b\tan(e+fx))^m\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^3}\,dx = \int \frac{(a+b\tan(e+fx))^m\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^3}\,dx$$

```
[In] integrate((a+b*tan(f*x+e))**m*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x
+e))**3,x)
```

```
[Out] Integral((a + b*tan(e + f*x))**m*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/(
c + d*tan(e + f*x))**3, x)
```

Maxima [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^m\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^3}\,dx = \text{Timed out}$$

```
[In] integrate((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^3,x, algorithm="maxima")
```

```
[Out] Timed out
```

Giac [F]

$$\int \frac{(a+b\tan(e+fx))^m\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^3}\,dx = \int \frac{\left(C\tan\left(fx+e\right)^2+B\tan\left(fx+e\right)+A\right)\left(b\tan\left(fx+e\right)+a\right)^m}{\left(d\tan\left(fx+e\right)+c\right)^3}\,dx$$

```
[In] integrate((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^3,x, algorithm="giac")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)^m/(d
*tan(f*x + e) + c)^3, x)
```

Mupad [F(-1)]

Timed out.

$$\int \frac{(a+b\tan(e+fx))^m\,(A+B\tan(e+fx)+C\tan^2(e+fx))}{(c+d\tan(e+fx))^3}\,dx = \int \frac{(a+b\tan(e+f\,x))^m\,\left(C\tan(e+f\,x)^2+B\tan(e+f\,x)+A\right)}{\left(c+d\tan\left(e+f\,x\right)\right)^3}\,dx$$

```
[In] int(((a + b*tan(e + f*x))^m*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d
*tan(e + f*x))^3,x)
```

```
[Out] int(((a + b*tan(e + f*x))^m*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d
*tan(e + f*x))^3, x)
```

CHAPTER 4

APPENDIX

4.1 Listing of Grading functions 2207

4.1 Listing of Grading functions

The following are the current version of the grading functions used for grading the quality of the antiderivative with reference to the optimal antiderivative included in the test suite.

There is a version for Maple and for Mathematica/Rubi. There is a version for grading Sympy and version for use with Sagemath.

The following are links to the current source code.

The following are the listings of source code of the grading functions.

Mathematica and Rubi grading function

```
(* Original version thanks to Albert Rich emailed on 03/21/2017 *)
(* ::Package:: *)

(* Nasser: April 7,2022. add second output which gives reason for the grade *)
(*                       Small rewrite of logic in main function to make it*)
(*                       match Maple's logic. No change in functionality otherwise*)

(* ::Subsection:: *)
(*GradeAntiderivative[result,optimal]*)


(* ::Text:: *)
(*If result and optimal are mathematical expressions, *)
```

```
(*          GradeAntiderivative[result,optimal] returns*)
(*  "F" if the result fails to integrate an expression that*)
(*      is integrable*)
(*  "C" if result involves higher level functions than necessary*)
(*  "B" if result is more than twice the size of the optimal*)
(*      antiderivative*)
(*  "A" if result can be considered optimal*)


GradeAntiderivative[result_,optimal_] := Module[{expnResult,expnOptimal,leafCountResult,leafCou
    expnResult  = ExpnType[result];
    expnOptimal = ExpnType[optimal];
    leafCountResult = LeafCount[result];
    leafCountOptimal = LeafCount[optimal];

    (*Print["expnResult=",expnResult," expnOptimal=",expnOptimal];*)
    If[expnResult<=expnOptimal,
        If[Not[FreeQ[result,Complex]], (*result contains complex*)
            If[Not[FreeQ[optimal,Complex]], (*optimal contains complex*)
                If[leafCountResult<=2*leafCountOptimal,
                    finalresult={"A",""}
                    ,(*ELSE*)
                    finalresult={"B","Both result and optimal contain complex but leaf count is
                  ]
            ,(*ELSE*)
                finalresult={"C","Result contains complex when optimal does not."}
            ]
        ,(*ELSE*)(*result does not contains complex*)
            If[leafCountResult<=2*leafCountOptimal,
               finalresult={"A",""}
            ,(*ELSE*)
               finalresult={"B","Leaf count is larger than twice the leaf count of optimal. $"<
              ]
        ]
    ,(*ELSE*) (*expnResult>expnOptimal*)
        If[FreeQ[result,Integrate] && FreeQ[result,Int],
            finalresult={"C","Result contains higher order function than in optimal. Order "<>T
            ,
            finalresult={"F","Contains unresolved integral."}
        ]
    ];

    finalresult
]

(* ::Text:: *)
(*The following summarizes the type number assigned an *)
```

```
(*expression based on the functions it involves*)
(*1 = rational function*)
(*2 = algebraic function*)
(*3 = elementary function*)
(*4 = special function*)
(*5 = hyperpergeometric function*)
(*6 = appell function*)
(*7 = rootsum function*)
(*8 = integrate function*)
(*9 = unknown function*)


ExpnType[expn_] :=
  If[AtomQ[expn],
    1,
  If[ListQ[expn],
    Max[Map[ExpnType,expn]],
  If[Head[expn]===Power,
    If[IntegerQ[expn[[2]]],
      ExpnType[expn[[1]]],
    If[Head[expn[[2]]]===Rational,
      If[IntegerQ[expn[[1]]] || Head[expn[[1]]]===Rational,
        1,
      Max[ExpnType[expn[[1]]],2]],
    Max[ExpnType[expn[[1]]],ExpnType[expn[[2]]],3]]],
  If[Head[expn]===Plus || Head[expn]===Times,
    Max[ExpnType[First[expn]],ExpnType[Rest[expn]]],
  If[ElementaryFunctionQ[Head[expn]],
    Max[3,ExpnType[expn[[1]]]],
  If[SpecialFunctionQ[Head[expn]],
    Apply[Max,Append[Map[ExpnType,Apply[List,expn]],4]],
  If[HypergeometricFunctionQ[Head[expn]],
    Apply[Max,Append[Map[ExpnType,Apply[List,expn]],5]],
  If[AppellFunctionQ[Head[expn]],
    Apply[Max,Append[Map[ExpnType,Apply[List,expn]],6]],
  If[Head[expn]===RootSum,
    Apply[Max,Append[Map[ExpnType,Apply[List,expn]],7]],
  If[Head[expn]===Integrate || Head[expn]===Int,
    Apply[Max,Append[Map[ExpnType,Apply[List,expn]],8]],
  9]]]]]]]]]]]


ElementaryFunctionQ[func_] :=
  MemberQ[{
  Exp,Log,
  Sin,Cos,Tan,Cot,Sec,Csc,
  ArcSin,ArcCos,ArcTan,ArcCot,ArcSec,ArcCsc,
```

```
  Sinh,Cosh,Tanh,Coth,Sech,Csch,
  ArcSinh,ArcCosh,ArcTanh,ArcCoth,ArcSech,ArcCsch
},func]


SpecialFunctionQ[func_] :=
  MemberQ[{
  Erf, Erfc, Erfi,
  FresnelS, FresnelC,
  ExpIntegralE, ExpIntegralEi, LogIntegral,
  SinIntegral, CosIntegral, SinhIntegral, CoshIntegral,
  Gamma, LogGamma, PolyGamma,
  Zeta, PolyLog, ProductLog,
  EllipticF, EllipticE, EllipticPi
},func]


HypergeometricFunctionQ[func_] :=
  MemberQ[{Hypergeometric1F1,Hypergeometric2F1,HypergeometricPFQ},func]


AppellFunctionQ[func_] :=
  MemberQ[{AppellF1},func]
```

Maple grading function

```
# File: GradeAntiderivative.mpl
# Original version thanks to Albert Rich emailed on 03/21/2017

#Nasser 03/22/2017  Use Maple leaf count instead since buildin
#Nasser 03/23/2017  missing 'ln' for ElementaryFunctionQ added
#Nasser 03/24/2017  corrected the check for complex result
#Nasser 10/27/2017  check for leafsize and do not call ExpnType()
#                   if leaf size is "too large". Set at 500,000
#Nasser 12/22/2019  Added debug flag, added 'dilog' to special functions
#                   see problem 156, file Apostol_Problems
#Nasser 4/07/2022   add second output which gives reason for the grade

GradeAntiderivative := proc(result,optimal)
local leaf_count_result,
        leaf_count_optimal,
        ExpnType_result,
        ExpnType_optimal,
        debug:=false;

        leaf_count_result:=leafcount(result);
```

```
        #do NOT call ExpnType() if leaf size is too large. Recursion problem
        if leaf_count_result > 500000 then
            return "B","result has leaf size over 500,000. Avoiding possible recursion issues
        fi;

        leaf_count_optimal := leafcount(optimal);
        ExpnType_result  := ExpnType(result);
        ExpnType_optimal := ExpnType(optimal);

        if debug then
            print("ExpnType_result",ExpnType_result," ExpnType_optimal=",ExpnType_optimal);
        fi;

# If result and optimal are mathematical expressions,
#  GradeAntiderivative[result,optimal] returns
#   "F" if the result fails to integrate an expression that
#       is integrable
#   "C" if result involves higher level functions than necessary
#   "B" if result is more than twice the size of the optimal
#       antiderivative
#   "A" if result can be considered optimal

    #This check below actually is not needed, since I only
    #call this grading only for passed integrals. i.e. I check
    #for "F" before calling this. But no harm of keeping it here.
    #just in case.


    if not type(result,freeof('int')) then
        return "F","Result contains unresolved integral";
    fi;


    if ExpnType_result<=ExpnType_optimal then
        if debug then
             print("ExpnType_result<=ExpnType_optimal");
        fi;
        if is_contains_complex(result) then
            if is_contains_complex(optimal) then
                if debug then
                        print("both result and optimal complex");
                fi;
                if leaf_count_result<=2*leaf_count_optimal then
                    return "A"," ";
                else
                   return "B",cat("Both result and optimal contain complex but leaf count of
                                convert(leaf_count_result,string)," vs. $2 (",
```

```
                                convert(leaf_count_optimal,string)," ) = ",convert(2*leaf_co
                end if
            else #result contains complex but optimal is not
                if debug then
                        print("result contains complex but optimal is not");
                fi;
                return "C","Result contains complex when optimal does not.";
            fi;
        else # result do not contain complex
             # this assumes optimal do not as well. No check is needed here.
            if debug then
                   print("result do not contain complex, this assumes optimal do not as well")
            fi;
            if leaf_count_result<=2*leaf_count_optimal then
                if debug then
                    print("leaf_count_result<=2*leaf_count_optimal");
                fi;
                return "A"," ";
            else
                if debug then
                    print("leaf_count_result>2*leaf_count_optimal");
                fi;
                return "B",cat("Leaf count of result is larger than twice the leaf count of opt
                                convert(leaf_count_result,string),"$ vs. $2(",
                                convert(leaf_count_optimal,string),")=",convert(2*leaf_count
            fi;
        fi;
    else #ExpnType(result) > ExpnType(optimal)
        if debug then
            print("ExpnType(result) > ExpnType(optimal)");
        fi;
        return "C",cat("Result contains higher order function than in optimal. Order ",
                    convert(ExpnType_result,string)," vs. order ",
                    convert(ExpnType_optimal,string),".");
    fi;

end proc:

#
# is_contains_complex(result)
# takes expressions and returns true if it contains "I" else false
#
#Nasser 032417
is_contains_complex:= proc(expression)
  return (has(expression,I));
end proc:
```

```
# The following summarizes the type number assigned an expression
# based on the functions it involves
# 1 = rational function
# 2 = algebraic function
# 3 = elementary function
# 4 = special function
# 5 = hyperpergeometric function
# 6 = appell function
# 7 = rootsum function
# 8 = integrate function
# 9 = unknown function

ExpnType := proc(expn)
  if type(expn,'atomic') then
    1
  elif type(expn,'list') then
    apply(max,map(ExpnType,expn))
  elif type(expn,'sqrt') then
    if type(op(1,expn),'rational') then
       1
    else
       max(2,ExpnType(op(1,expn)))
    end if
  elif type(expn,'`^`') then
    if type(op(2,expn),'integer') then
      ExpnType(op(1,expn))
    elif type(op(2,expn),'rational') then
      if type(op(1,expn),'rational') then
         1
      else
         max(2,ExpnType(op(1,expn)))
      end if
    else
         max(3,ExpnType(op(1,expn)),ExpnType(op(2,expn)))
    end if
  elif type(expn,'`+`') or type(expn,'`*`') then
    max(ExpnType(op(1,expn)),max(ExpnType(rest(expn))))
  elif ElementaryFunctionQ(op(0,expn)) then
    max(3,ExpnType(op(1,expn)))
  elif SpecialFunctionQ(op(0,expn)) then
    max(4,apply(max,map(ExpnType,[op(expn)])))
  elif HypergeometricFunctionQ(op(0,expn)) then
    max(5,apply(max,map(ExpnType,[op(expn)])))
  elif AppellFunctionQ(op(0,expn)) then
    max(6,apply(max,map(ExpnType,[op(expn)])))
  elif op(0,expn)='int' then
    max(8,apply(max,map(ExpnType,[op(expn)]))) else
```

```
  9
  end if
end proc:


ElementaryFunctionQ := proc(func)
  member(func,[
        exp,log,ln,
        sin,cos,tan,cot,sec,csc,
        arcsin,arccos,arctan,arccot,arcsec,arccsc,
        sinh,cosh,tanh,coth,sech,csch,
        arcsinh,arccosh,arctanh,arccoth,arcsech,arccsch])
end proc:

SpecialFunctionQ := proc(func)
  member(func,[
        erf,erfc,erfi,
        FresnelS,FresnelC,
        Ei,Ei,Li,Si,Ci,Shi,Chi,
        GAMMA,lnGAMMA,Psi,Zeta,polylog,dilog,LambertW,
        EllipticF,EllipticE,EllipticPi])
end proc:

HypergeometricFunctionQ := proc(func)
  member(func,[Hypergeometric1F1,hypergeom,HypergeometricPFQ])
end proc:

AppellFunctionQ := proc(func)
  member(func,[AppellF1])
end proc:

# u is a sum or product.  rest(u) returns all but the
# first term or factor of u.
rest := proc(u) local v;
  if nops(u)=2 then
     op(2,u)
  else
     apply(op(0,u),op(2..nops(u),u))
  end if
end proc:

#leafcount(u) returns the number of nodes in u.
#Nasser 3/23/17  Replaced by build-in leafCount from package in Maple
leafcount := proc(u)
   MmaTranslator[Mma][LeafCount](u);
end proc:
```

Sympy grading function

```
#Dec 24, 2019. Nasser M. Abbasi:
#              Port of original Maple grading function by
#              Albert Rich to use with Sympy/Python
#Dec 27, 2019  Nasser. Added `RootSum`. See problem 177, Timofeev file
#              added 'exp_polar'
from sympy import *

def leaf_count(expr):
    #sympy do not have leaf count function. This is approximation
    return round(1.7*count_ops(expr))

def is_sqrt(expr):
    if isinstance(expr,Pow):
        if expr.args[1] == Rational(1,2):
            return True
        else:
            return False
    else:
        return False

def is_elementary_function(func):
    return func in [exp,log,ln,sin,cos,tan,cot,sec,csc,
            asin,acos,atan,acot,asec,acsc,sinh,cosh,tanh,coth,sech,csch,
            asinh,acosh,atanh,acoth,asech,acsch
        ]

def is_special_function(func):
    return func in [ erf,erfc,erfi,
             fresnels,fresnelc,Ei,Ei,Li,Si,Ci,Shi,Chi,
             gamma,loggamma,digamma,zeta,polylog,LambertW,
             elliptic_f,elliptic_e,elliptic_pi,exp_polar
         ]

def is_hypergeometric_function(func):
    return func in [hyper]

def is_appell_function(func):
    return func in [appellf1]

def is_atom(expn):
    try:
        if expn.isAtom or isinstance(expn,int) or isinstance(expn,float):
          return True
        else:
          return False
```

```
    except AttributeError as error:
        return False

def expnType(expn):
    debug=False
    if debug:
        print("expn=",expn,"type(expn)=",type(expn))

    if is_atom(expn):
       return 1
    elif isinstance(expn,list):
        return max(map(expnType, expn))    #apply(max,map(ExpnType,expn))
    elif  is_sqrt(expn):
        if isinstance(expn.args[0],Rational): #type(op(1,expn),'rational')
            return 1
        else:
            return max(2,expnType(expn.args[0]))   #max(2,ExpnType(op(1,expn)))
    elif isinstance(expn,Pow):    #type(expn,'`^`')
        if isinstance(expn.args[1],Integer):   #type(op(2,expn),'integer')
            return expnType(expn.args[0])    #ExpnType(op(1,expn))
        elif isinstance(expn.args[1],Rational):   #type(op(2,expn),'rational')
            if isinstance(expn.args[0],Rational): #type(op(1,expn),'rational')
                return 1
            else:
                return max(2,expnType(expn.args[0]))   #max(2,ExpnType(op(1,expn)))
        else:
            return max(3,expnType(expn.args[0]),expnType(expn.args[1])) #max(3,ExpnType(op(1,expn)),ExpnTy
    elif isinstance(expn,Add) or isinstance(expn,Mul): #type(expn,'`+`') or type(expn,'`*`')
        m1 = expnType(expn.args[0])
        m2 = expnType(list(expn.args[1:]))
        return max(m1,m2)   #max(ExpnType(op(1,expn)),max(ExpnType(rest(expn))))
    elif is_elementary_function(expn.func):   #ElementaryFunctionQ(op(0,expn))
        return max(3,expnType(expn.args[0]))   #max(3,ExpnType(op(1,expn)))
    elif is_special_function(expn.func): #SpecialFunctionQ(op(0,expn))
        m1 = max(map(expnType, list(expn.args)))
        return max(4,m1)    #max(4,apply(max,map(ExpnType,[op(expn)])))
    elif is_hypergeometric_function(expn.func): #HypergeometricFunctionQ(op(0,expn))
        m1 = max(map(expnType, list(expn.args)))
        return max(5,m1)    #max(5,apply(max,map(ExpnType,[op(expn)])))
    elif is_appell_function(expn.func):
        m1 = max(map(expnType, list(expn.args)))
        return max(6,m1)    #max(5,apply(max,map(ExpnType,[op(expn)])))
    elif isinstance(expn,RootSum):
        m1 = max(map(expnType, list(expn.args))) #Apply[Max,Append[Map[ExpnType,Apply[List,expn]],7]],
        return max(7,m1)
    elif str(expn).find("Integral") != -1:
```

```
        m1 = max(map(expnType, list(expn.args)))
        return max(8,m1)    #max(5,apply(max,map(ExpnType,[op(expn)])))
    else:
        return 9

#main function
def grade_antiderivative(result,optimal):

    #print ("Enter grade_antiderivative for sagemath")
    #print("Enter grade_antiderivative, result=",result," optimal=",optimal)

    leaf_count_result  = leaf_count(result)
    leaf_count_optimal = leaf_count(optimal)

    #print("leaf_count_result=",leaf_count_result)
    #print("leaf_count_optimal=",leaf_count_optimal)

    expnType_result  = expnType(result)
    expnType_optimal = expnType(optimal)

    if str(result).find("Integral") != -1:
        grade = "F"
        grade_annotation =""
    else:
        if expnType_result <= expnType_optimal:
            if result.has(I):
                if optimal.has(I): #both result and optimal complex
                    if leaf_count_result <= 2*leaf_count_optimal:
                        grade = "A"
                        grade_annotation =""
                    else:
                        grade = "B"
                        grade_annotation ="Both result and optimal contain complex but leaf count of result is larg
                else: #result contains complex but optimal is not
                    grade = "C"
                    grade_annotation ="Result contains complex when optimal does not."
            else: # result do not contain complex, this assumes optimal do not as well
                if leaf_count_result <= 2*leaf_count_optimal:
                    grade = "A"
                    grade_annotation =""
                else:
                    grade = "B"
                    grade_annotation ="Leaf count of result is larger than twice the leaf count of optimal. "+str(le
        else:
            grade = "C"
            grade_annotation ="Result contains higher order function than in optimal. Order "+str(ExpnType_
```

```
    #print("Before returning. grade=",grade, " grade_annotation=",grade_annotation)

    return grade, grade_annotation
```

SageMath grading function

```
#Dec 24, 2019. Nasser: Ported original Maple grading function by
#            Albert Rich to use with Sagemath. This is used to
#            grade Fricas, Giac and Maxima results.
#Dec 24, 2019. Nasser: Added 'exp_integral_e' and 'sng', 'sin_integral'
#            'arctan2','floor','abs','log_integral'
#June 4, 2022  Made default grade_annotation "none" instead of "" due
#            issue later when reading the file.
#July 14, 2022. Added ellipticF. This is until they fix sagemath, then remove it.

from sage.all import *
from sage.symbolic.operators import add_vararg, mul_vararg

debug=False;

def tree_size(expr):
    r"""
    Return the tree size of this expression.
    """
    #print("Enter tree_size, expr is ",expr)

    if expr not in SR:
        # deal with lists, tuples, vectors
        return 1 + sum(tree_size(a) for a in expr)
    expr = SR(expr)
    x, aa = expr.operator(), expr.operands()
    if x is None:
        return 1
    else:
        return 1 + sum(tree_size(a) for a in aa)

def is_sqrt(expr):
    if expr.operator() == operator.pow:   #isinstance(expr,Pow):
        if expr.operands()[1]==1/2: #expr.args[1] == Rational(1,2):
            if debug: print ("expr is sqrt")
            return True
        else:
            return False
    else:
        return False
```

```
def is_elementary_function(func):
    #debug=False
    m = func.name() in ['exp','log','ln',
            'sin','cos','tan','cot','sec','csc',
            'arcsin','arccos','arctan','arccot','arcsec','arccsc',
            'sinh','cosh','tanh','coth','sech','csch',
            'arcsinh','arccosh','arctanh','arccoth','arcsech','arccsch','sgn',
        'arctan2','floor','abs'
        ]
    if debug:
        if m:
            print ("func ", func , " is elementary_function")
        else:
            print ("func ", func , " is NOT elementary_function")


    return m

def is_special_function(func):
    #debug=False
    if debug:
        print ("type(func)=", type(func))

    m= func.name() in ['erf','erfc','erfi','fresnel_sin','fresnel_cos','Ei',
          'Ei','Li','Si','sin_integral','Ci','cos_integral','Shi','sinh_integral'
          'Chi','cosh_integral','gamma','log_gamma','psi,zeta',
          'polylog','lambert_w','elliptic_f','elliptic_e','ellipticF',
          'elliptic_pi','exp_integral_e','log_integral']

    if debug:
        print ("m=",m)
        if m:
            print ("func ", func ," is special_function")
        else:
            print ("func ", func ," is NOT special_function")


    return m


def is_hypergeometric_function(func):
    return func.name() in ['hypergeometric','hypergeometric_M','hypergeometric_U']

def is_appell_function(func):
    return func.name() in ['hypergeometric']   #[appellf1] can't find this in sagemath
```

```
def is_atom(expn):

    #debug=False
    if debug:
        print ("Enter is_atom, expn=",expn)

    if not hasattr(expn, 'parent'):
        return False


    #thanks to answer at https://ask.sagemath.org/question/49179/what-is-sagemath-equivalent-to-atomic-t
    try:
        if expn.parent() is SR:
            return expn.operator() is None
        if expn.parent() in (ZZ, QQ, AA, QQbar):
            return expn in expn.parent() # Should always return True
        if hasattr(expn.parent(),"base_ring") and hasattr(expn.parent(),"gens"):
            return expn in expn.parent().base_ring() or expn in expn.parent().gens()

        return False

    except AttributeError as error:
        print("Exception,AttributeError in  is_atom")
        print ("cought exception" , type(error).__name__ )
        return False


def expnType(expn):

    if debug:
        print (">>>>>Enter expnType, expn=", expn)
        print (">>>>>is_atom(expn)=", is_atom(expn))

    if is_atom(expn):
        return 1
    elif type(expn)==list:   #isinstance(expn,list):
        return max(map(expnType, expn))    #apply(max,map(ExpnType,expn))
    elif  is_sqrt(expn):
        if  type(expn.operands()[0])==Rational:  #type(isinstance(expn.args[0],Rational):
            return 1
        else:
            return max(2,expnType(expn.operands()[0]))   #max(2,expnType(expn.args[0]))
    elif expn.operator() == operator.pow:    #isinstance(expn,Pow)
        if type(expn.operands()[1])==Integer:   #isinstance(expn.args[1],Integer)
            return expnType(expn.operands()[0])    #expnType(expn.args[0])
        elif type(expn.operands()[1])==Rational:   #isinstance(expn.args[1],Rational)
            if type(expn.operands()[0])==Rational:  #isinstance(expn.args[0],Rational)
```

```
            return 1
        else:
            return max(2,expnType(expn.operands()[0]))  #max(2,expnType(expn.args[0]))
    else:
        return max(3,expnType(expn.operands()[0]),expnType(expn.operands()[1])) #max(3,expnType(expn.
elif expn.operator() == add_vararg or expn.operator() == mul_vararg: #isinstance(expn,Add) or isinsta
    m1 = expnType(expn.operands()[0]) #expnType(expn.args[0])
    m2 = expnType(expn.operands()[1:]) #expnType(list(expn.args[1:]))
    return max(m1,m2)  #max(ExpnType(op(1,expn)),max(ExpnType(rest(expn))))
elif is_elementary_function(expn.operator()):  #is_elementary_function(expn.func)
    return max(3,expnType(expn.operands()[0]))
elif is_special_function(expn.operator()): #is_special_function(expn.func)
    m1 = max(map(expnType, expn.operands()))      #max(map(expnType, list(expn.args)))
    return max(4,m1)   #max(4,m1)
elif is_hypergeometric_function(expn.operator()): #is_hypergeometric_function(expn.func)
    m1 = max(map(expnType, expn.operands()))       #max(map(expnType, list(expn.args)))
    return max(5,m1)   #max(5,m1)
elif is_appell_function(expn.operator()):
    m1 = max(map(expnType, expn.operands()))       #max(map(expnType, list(expn.args)))
    return max(6,m1)   #max(6,m1)
elif str(expn).find("Integral") != -1: #this will never happen, since it
            #is checked before calling the grading function that is passed.
            #but kept it here.
    m1 = max(map(expnType, expn.operands()))       #max(map(expnType, list(expn.args)))
    return max(8,m1)   #max(5,apply(max,map(ExpnType,[op(expn)])))
else:
    return 9

#main function
def grade_antiderivative(result,optimal):



    if debug:
        print ("Enter grade_antiderivative for sagemath")
        print("Enter grade_antiderivative, result=",result)
        print("Enter grade_antiderivative, optimal=",optimal)
        print("type(anti)=",type(result))
        print("type(optimal)=",type(optimal))

    leaf_count_result  = tree_size(result) #leaf_count(result)
    leaf_count_optimal = tree_size(optimal) #leaf_count(optimal)

    #if debug: print ("leaf_count_result=", leaf_count_result, "leaf_count_optimal=",leaf_count_optimal)



    expnType_result  = expnType(result)
    expnType_optimal = expnType(optimal)
```

```
if debug: print ("expnType_result=", expnType_result, "expnType_optimal=",expnType_optimal)

if expnType_result <= expnType_optimal:
    if result.has(I):
        if optimal.has(I): #both result and optimal complex
            if leaf_count_result <= 2*leaf_count_optimal:
                grade = "A"
                grade_annotation ="none"
            else:
                grade = "B"
                grade_annotation ="Both result and optimal contain complex but leaf count of result is larger th
        else: #result contains complex but optimal is not
            grade = "C"
            grade_annotation ="Result contains complex when optimal does not."
    else: # result do not contain complex, this assumes optimal do not as well
        if leaf_count_result <= 2*leaf_count_optimal:
            grade = "A"
            grade_annotation ="none"
        else:
            grade = "B"
            grade_annotation ="Leaf count of result is larger than twice the leaf count of optimal. "+str(leaf_c
else:
    grade = "C"
    grade_annotation ="Result contains higher order function than in optimal. Order "+str(expnType_resul


print("Before returning. grade=",grade, " grade_annotation=",grade_annotation)

return grade, grade_annotation
```