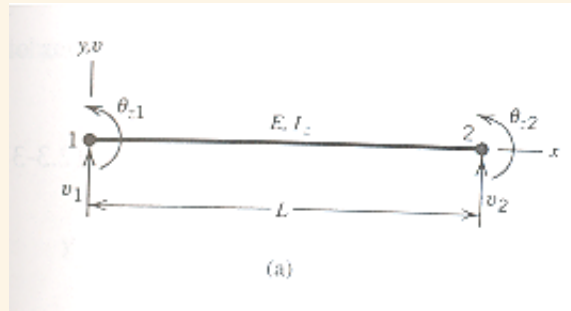


Generating 4-degree of freedom beam stiffness matrix

by Nasser M. Abbasi oct 26,2009

This diagram below is from the book "Concepts and applications of finite element methods" 4th edition. The goal is to find the stiffness matrix for this beam. It has 2 degrees of freedom per node. I will start by assuming the deflection $w(x)$ to be approximated by a cubic polynomial as a function of x , and then use minimum potential energy approach to find the stiffness matrix.



Assume that the deflection is approximated by $w = a_1 + a_2 x + a_3 x^2 + a_4 x^3$ and from this we need to obtain the

shape function N so that we can write $w = N \begin{pmatrix} v_1 \\ \theta_1 \\ v_2 \\ \theta_2 \end{pmatrix}$ hence we see that N must be a 1 by 4 row vector. We call

a_1, a_2, a_3, a_4 the generalized coordinates, and $v_1, v_2, \theta_1, \theta_2$ as the nodal degrees of freedom.

```
In[1]:= w = a1 + a2 x + a3 x^2 + a4 x^3;
```

Now find an expression for each of the above 4 nodal unknowns in terms of the deflection. Notice that the angle $\theta = \frac{dw}{dx}$ as it is being approximated for small deflection (as in the angle of an arc (in radian) is the arc length over radius)

```
In[6]:= eq1 = v1 == w /. x -> 0;
eq2 = v2 == w /. x -> L;
eq3 = theta1 == D[w, x] /. x -> 0;
eq4 = theta2 == D[w, x] /. x -> L;
```

Now to solve for the generalized d.o.f a_1, a_2, a_3, a_4

```
In[42]:= generalDegreeOfFreedomVars = {a1, a2, a3, a4};
First@Solve[{eq1, eq2, eq3, eq4}, generalDegreeOfFreedomVars];
wNew = w /. %;
Print["w(x) = " <> ToString[wNew, FormatType -> TraditionalForm]]
```

$$w(x) = -\frac{x^3 (-L \theta_1 - L \theta_2 - 2 v_1 + 2 v_2)}{L^3} - \frac{x^2 (2 L \theta_1 + L \theta_2 + 3 v_1 - 3 v_2)}{L^2} + v_1 + x \theta_1$$

To write the above in the form $w = N d$ we do

```
In[46]:= nodalsDegreeOfFreedomVars = {v1, theta1, v2, theta2};
{b, shapeFunctions} = Normal[CoefficientArrays[wNew, nodalsDegreeOfFreedomVars]];
wNew = shapeFunctions.nodalsDegreeOfFreedomVars;
Print["w(x) = ", wNew]
```

$$w(x) = \left(1 - \frac{3x^2}{L^2} + \frac{2x^3}{L^3}\right)v_1 + \left(\frac{3x^2}{L^2} - \frac{2x^3}{L^3}\right)v_2 + \left(x - \frac{2x^2}{L} + \frac{x^3}{L^2}\right)\theta_1 + \left(-\frac{x^2}{L} + \frac{x^3}{L^2}\right)\theta_2$$

Now that we have found the shape functions, we can find the B matrix. Notice that for beam, the strain matrix B is given by $\frac{d^2 w}{dx^2}$

```
In[50]:= B = D[shapeFunctions, {x, 2}]
```

```
Out[50]= { - 6/L^2 + 12x/L^3, - 4/L + 6x/L^2, 6/L^2 - 12x/L^3, - 2/L + 6x/L^2 }
```

and now Π the element potential energy is found (here I only consider strain energy, no loading is needed since we only need to find the element stiffness matrix, not solve a system for unknowns).

```
In[53]:= N = {shapeFunctions};
B = {B};
d = Transpose[{nodalsDegreeOfFreedomVars}];
Dimensions[N]
Dimensions[d]
Dimensions[B]
```

```
Out[56]= {1, 4}
```

```
Out[57]= {4, 1}
```

```
Out[58]= {1, 4}
```

```
In[59]:= Pi = 1/2 Integrate[E I Transpose[d].Transpose[B].B.d dx];
Pi = Pi[[1, 1]]
```

```
Out[60]= 1/L^3 2 E I (3 v1^2 + 3 v2^2 - 3 L v2 (theta1 + theta2) + L^2 (theta1^2 + theta1 theta2 + theta2^2) + 3 v1 (-2 v2 + L (theta1 + theta2)))
```

Now that we have the element potential energy, we can generate the element stiffness matrix

```

In[61]:= kmat = Integrate[E I Transpose[B].B] dx;
Print[ToString[ $\frac{E I}{L^3}$ , FormatType -> TraditionalForm] <>
ToString[kmat / ( $\frac{E I}{L^3}$ ), FormatType -> TraditionalForm] <>
ToString[d, FormatType -> TraditionalForm]]

```

$$\frac{E I}{L^3} \begin{pmatrix} 12 & 6 L & -12 & 6 L \\ 6 L & 4 L^2 & -6 L & 2 L^2 \\ -12 & -6 L & 12 & -6 L \\ 6 L & 2 L^2 & -6 L & 4 L^2 \end{pmatrix} \begin{pmatrix} v_1 \\ \theta_1 \\ v_2 \\ \theta_2 \end{pmatrix}$$