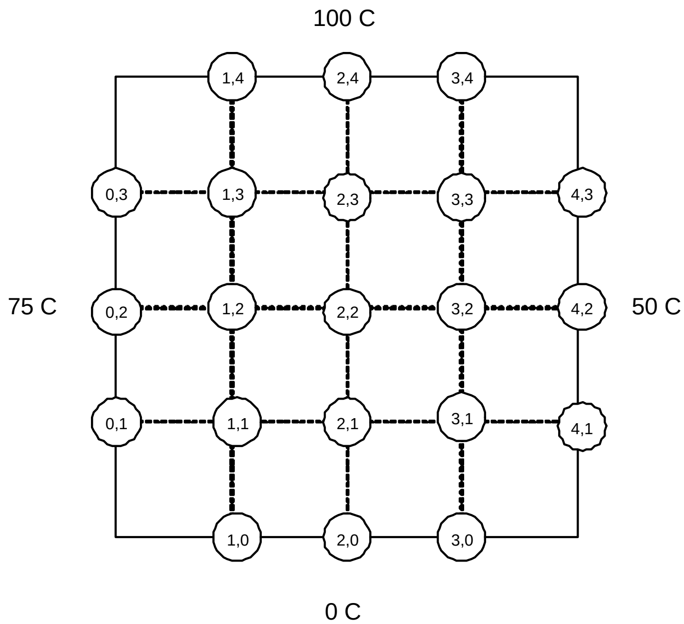

Figure 1: Initial state of system

Problem 29.2

Detrmine the distrubtion of heat flux for the shown hearted plate. Assume the plate is 40×40 cm and is made of aluminum ($k' = 0.49$ cal/(s.cm.C^0).

Solution

The heat flux vector is found as follows.

For each point in the grid, the x-component of the vector originating at the point is found by taking the difference of the solution T values located at the adjacent 2 points along the x-direction, divided by the distance between those 2 points. Similarly, the y-component is found, but now we consider the 2 points along the y-directions, one above and one below. Once the x and y component is found, the vector slope is determined.

The final solution from problem 29.1 is found to be:

```
A =


   77.6961   74.8137   69.2085
   60.4724   53.6169   51.4969
   46.6481   29.1632   33.9286
```

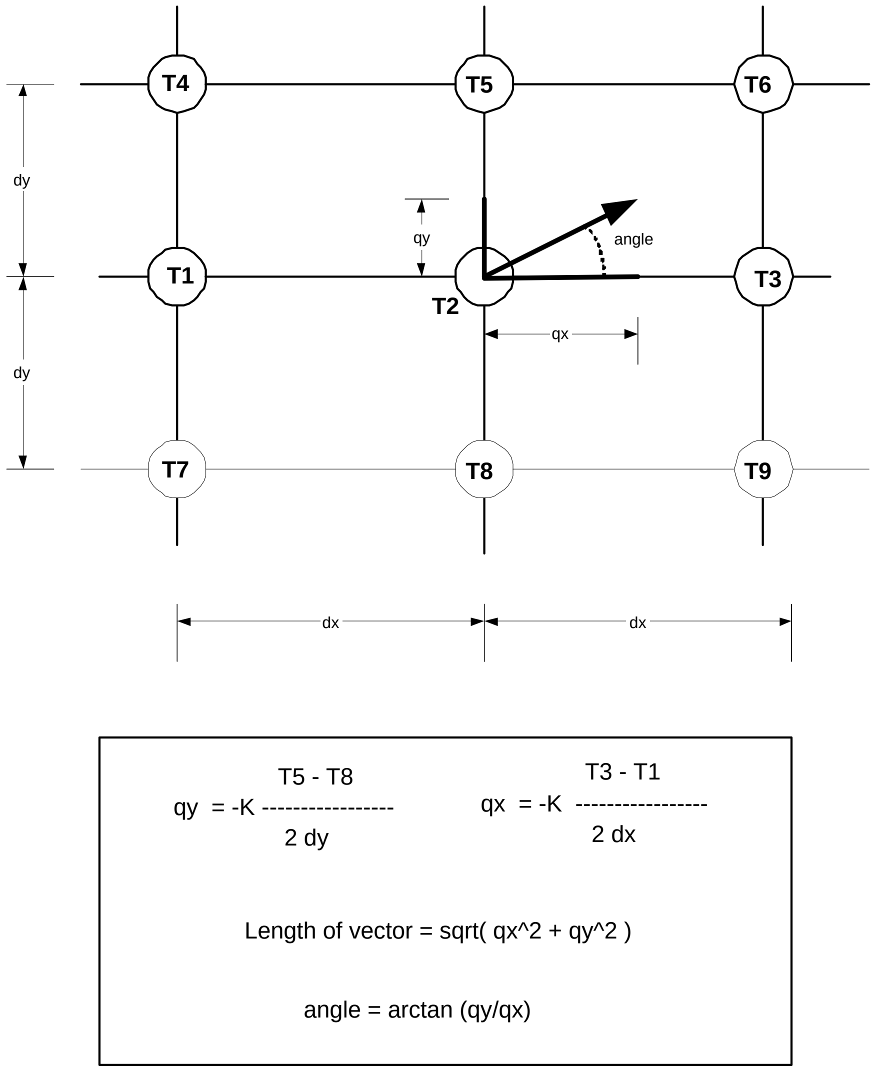

Figure 2: Method of calcuating heat flux

And the above is what will be used to find the flux.
The following diagram shows how the flux vector is calculated for one point T_2. These calculation are made for each point and the final flux vector field is plotted.

Since this is a 40cm square, the x and y distance between any 2-points is
10-cm.

Calculations of flux follows.
$i = 1, j = 1$
$qx = -0.49\frac{T_{i+1,j} - T_{i-1,j}}{2\Delta x} = -0.49\frac{T_{2,1} - T_{0,1}}{2\Delta x} = -0.49\frac{29.163-75}{2(10)} = 1.1230 \ \frac{cal}{cm^2 \sec}$

$qy = -0.49 \frac{T_{i,j+1} - T_{i,j-1}}{2\Delta y} = -0.49 \frac{T_{1,2} - T_{1,0}}{2\Delta y} = -0.49 \frac{60.4724 - 0}{2(10)} = -1.4816 \frac{cal}{cm^2 \sec}$

length of flux vector at this point is $\sqrt{q_x^2 + q_y^2} = \sqrt{1.1230^2 + (-1.4816)^2} = 1.8591$

The angle the vector makes with the x-direction is

$\theta = \arctan\left(\frac{q_y}{q_x}\right) = \arctan\left(\frac{-1.4816}{1.1230}\right) = -52.8392^0$

This process is repeated for each point.

It is clear then how the flux vector field is calculated.

To display the final result, I wrote a matlab fucntion called nma_getFlux1 which accepts as input the matrix (solution T), and the distance between each x-point, and the distance between each y-point. It reurns a matrix which contains the length of the flux at each grid point, and another matrix which contains the angle of the flux at each grid point.

This below is the result of running this function. First obtain the Solution:

```
>> A=nma_laplaceRectDirchlet(xpoints, ypoints, bottom, right, top, left, lambda, 1);
A =

   75.0000  100.0000  100.0000  100.0000   50.0000
   75.0000   77.6961   74.8137   69.2085   50.0000
   75.0000   60.4724   53.6169   51.4969   50.0000
   75.0000   46.6481   29.1632   33.9286   50.0000
         0         0         0         0   50.0000


epsilonA =

    0.1517
```

Now pass the solution to the flux calculation function. use $k = 0.49,\ dx = 10, dy = 10$.

```
>> [flux,beta]=nma_getFlux1(A,0.49,10,10)

flux =
```

```
         0         0         0         0         0
         0    0.9684    1.1553    1.3348         0
         0    0.9236    1.1399    0.8689         0
         0    1.8591    1.3501    1.3610         0
         0         0         0         0         0


beta =

         0         0         0         0         0
         0  -89.7300  -79.6302  -62.9061         0
         0  -55.4445  -78.8768  -84.1464         0
         0  -52.8386  -76.6545 -112.0294         0
         0         0         0         0         0
```

The above shows the solution. It shows the magntude of the flux at each internal grid point, and the angle to the x-axis of the vector. I plot (approximatly) by hand the above flux at each point. The result shown in the diagram

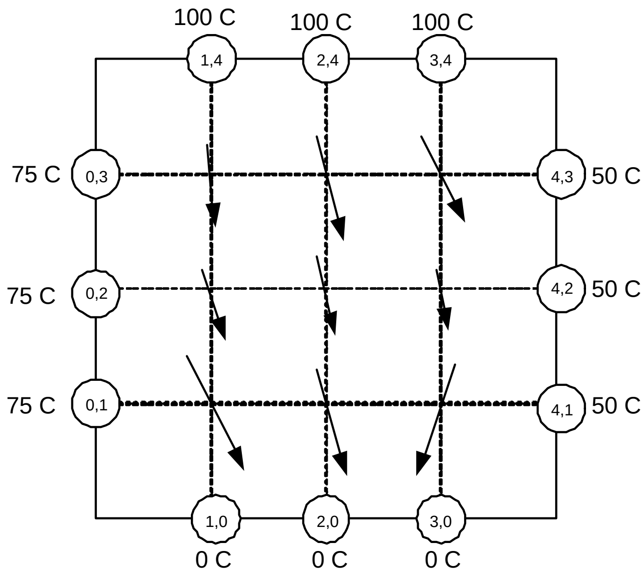

Figure 3: Plot of heat flux

Source code:

```
function [flux,beta]=nma_getFlux1(A,k,dx,dy)

TRUE  = 1;
FALSE = 0;

[row,col]=size(A);
flux  = zeros(row,col);
beta  = zeros(row,col);
nx=row-2;
ny=col-2;

for(i=2+nx-1:-1:2)
    for(j=2:1:2+ny-1)
        qx=-k*(A(i,j+1)-A(i,j-1))/(2*dx);
        qy=-k*(A(i-1,j)-A(i+1,j))/(2*dy);
        flux(i,j)=sqrt(qx^2+qy^2);
        beta(i,j)=atan2(qy,qx)*180/pi;
```

```
    end
end

end
```