NAU NORTHERN ARIZONA UNIVERSITY

College of Engineering, Informatics, and Applied Sciences

To: Carson Pete

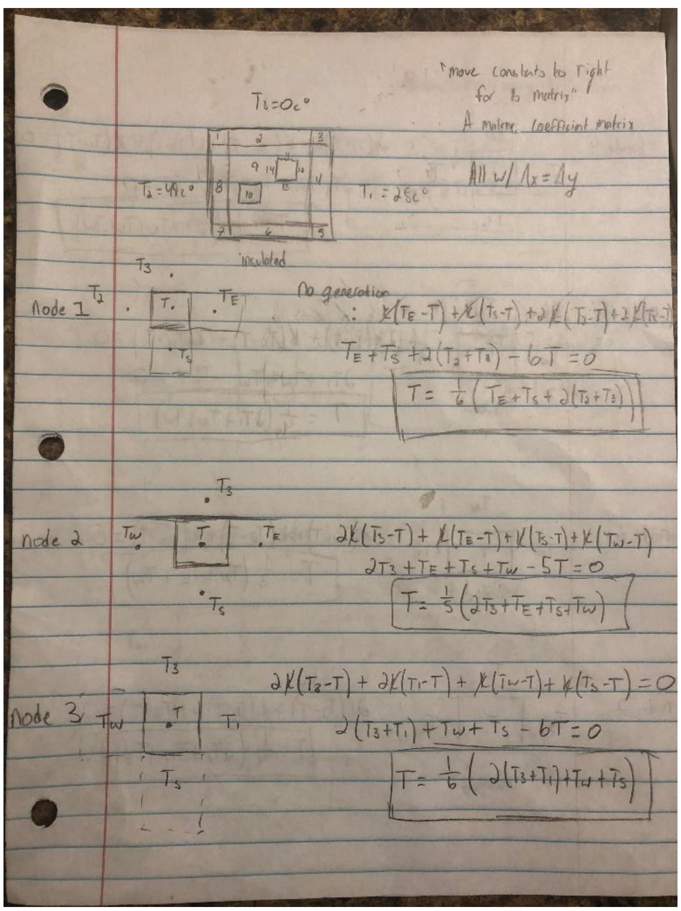
From: Abel Aldape

Due: 03/21/2021

Re: ME 450 Computer Project

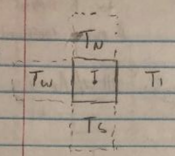
Introduction

The computer project requires us to conduct a heat transfer study on a 2D simple geometry. The geometry contains a generation and convection block, as well as conduction through the plate and boundary conditions. Three surrounding temperatures are known and the bottom edge is an insulated surface. The goal of the project is to create a temperature profile of the geometry, which requires us to know all the temperatures contained within the plate. To accurately obtain all the temperatures nodes the finite difference technique will be used. The finite differencing technique is a way to measure the temperature of a specific node by averaging the temperature of all surrounding nodes. The simple geometry is broken up into 14 nodes that can be seen in the following nodal calculations.



College of Engineering, Informatics, and Applied Sciences

node 4

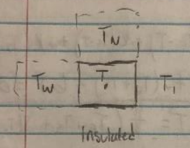


$$2k(T_1-T)+k(T_N-T)+k(T_S-T)+k(T_W-T)=0$$

$$2T_1+T_N+T_S+T_W-5T=0$$

$$T=\frac{1}{5}\left(2T_1+T_N+T_S+T_W\right)$$

node 5

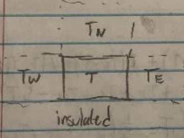


$$2k(T_1-T)+k(T_N-T)+k(T_W-T)=0$$

$$2T_1+T_N+T_W-4T=0$$

$$T=\frac{1}{4}\left(2T_1+T_N+T_W\right)$$

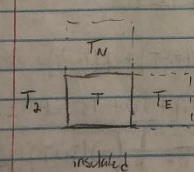
node 6



$$k(T_N-T)+k(T_E-T)+k(T_W-T)=0$$

$$T=\frac{1}{3}\left(T_N+T_E+T_W\right)$$

node 7

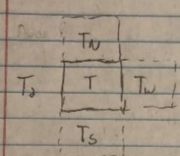


$$2k(T_2-T)+k(T_N-T)+k(T_E-T)=0$$

$$T=\frac{1}{4}\left(2T_2+T_N+T_E\right)$$

Newtons Law of cooling $h\Delta x(T_\infty-T)$

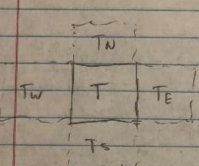
node 8



$$2k(T_2-T)+k(T_N-T)+k(T_W-T)+k(T_S-T)=0$$

$$T=\frac{1}{5}\left(2T_2+T_N+T_W+T_S\right)$$

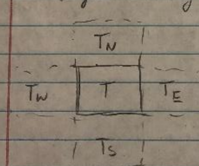
node 9



$$k(T_N-T)+k(T_E-T)+k(T_S-T)+k(T_W-T)$$

$$T=\frac{1}{4}\left(T_N+T_E+T_S+T_W\right)$$

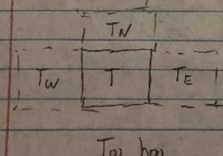
node 10 w/ generation $\dot{g}=35{,}000+\left(\frac{W}{m^3}\right)$



just node 9 + $\dot{g}$

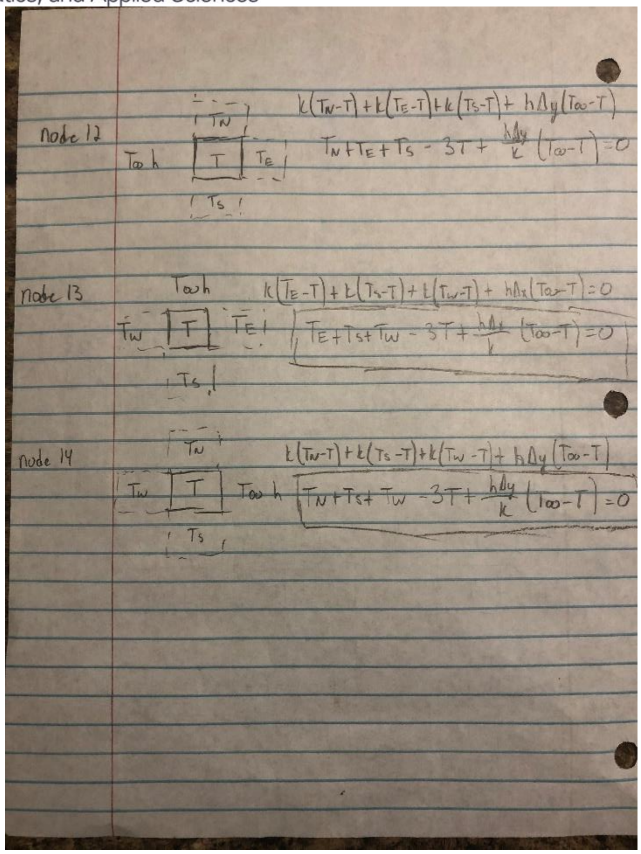
$$T=\frac{1}{4}\left(T_N+T_E+T_S+T_W\right)+\dot{g}$$

node 11



$$k(T_N-T)+k(T_E-T)+k(T_W-T)+h\Delta x(T_\infty-T)$$

$$T_N+T_E+T_W-3T+\frac{h\Delta x}{k}(T_\infty-T)=0$$



Nodal Equations: Homework 6

The equations obtained from Homework 6 will be used to create the known and unknown values in our A and b matrices. Using MATLAB's '\' function we can easily solve for temperatures at every node. Finally, we must validate our results through a number of techniques. The first will be to ensure all values inputted into the A and b matrix are correct. Next the sum of the heat transfer along the boundaries should be equal to the generation multiplied by the volume. Lastly, Dr. Pete provided us with temperature plots that our code should approximately replicate.

Code Validation

The first requirement is to visually compare our results to the heat map key provided to us. I have provided my 2D heat graph as this visual comparison because the 3D temperature map is required in the results section of this report.

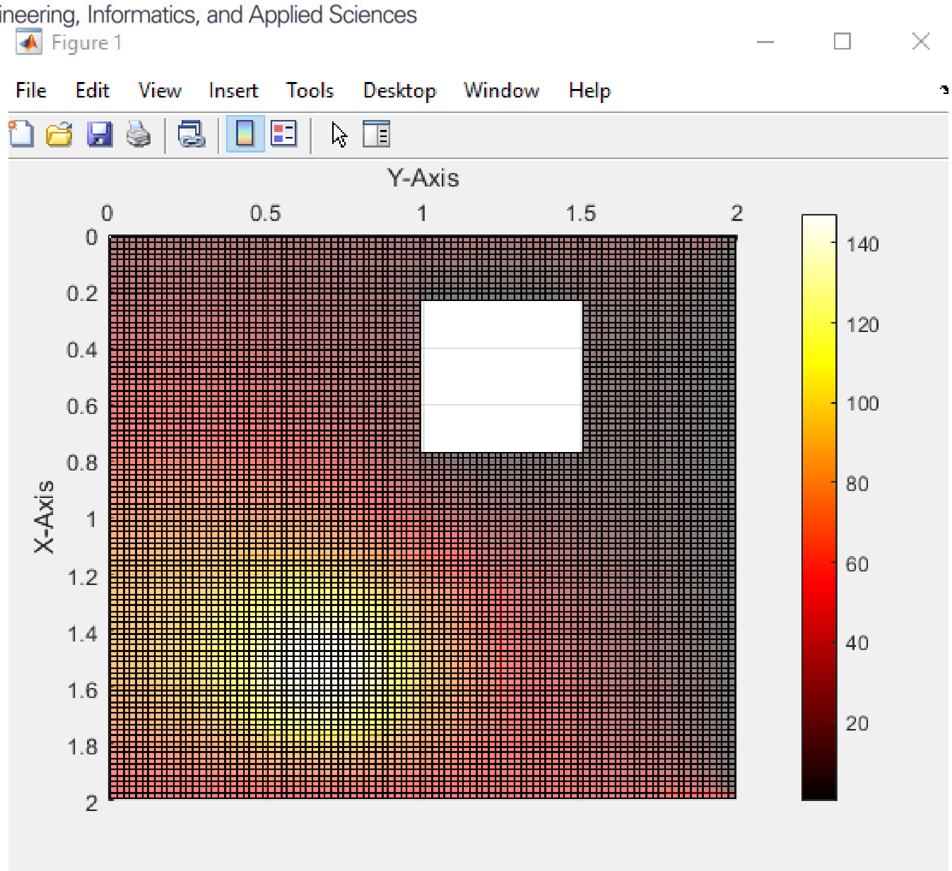


Figure 1: Generated 2D Heat Map

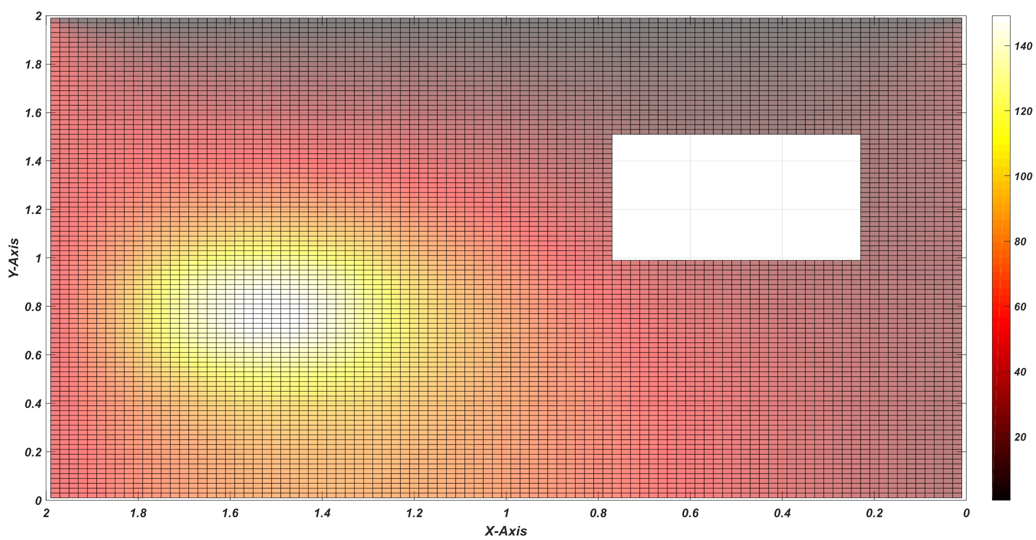


Figure 2: 2D Heat Map Key

The generated heat map accurately matches the given key and helps to validate my code.

Each boundary condition is expected to physically affect the 3D temperature graph. The four boundary conditions given in the project are

T1 = 25 degrees

T2 = 49 degrees

T3 = 0 degrees

4th boundary = Insulated

The T3 boundary condition is the easiest to visualize. The temperature along the top edge will be extremely close to zero, and will increase at each corner.

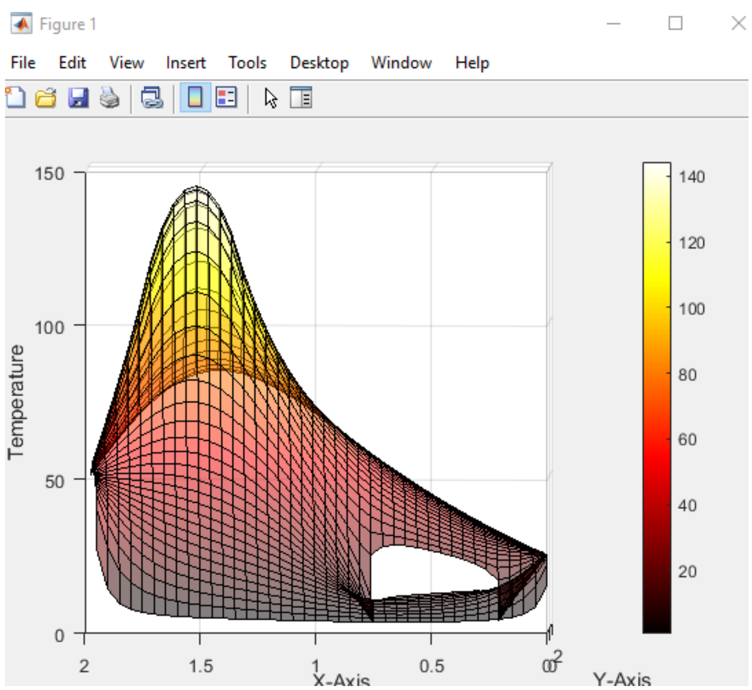


Figure 3: T3 Boundary Condition

The T2 boundary condition will also cause a majority of the boundary nodes to be close to 50 degrees, with the exception of the nodes closest to the generation block that have hotter temperatures.

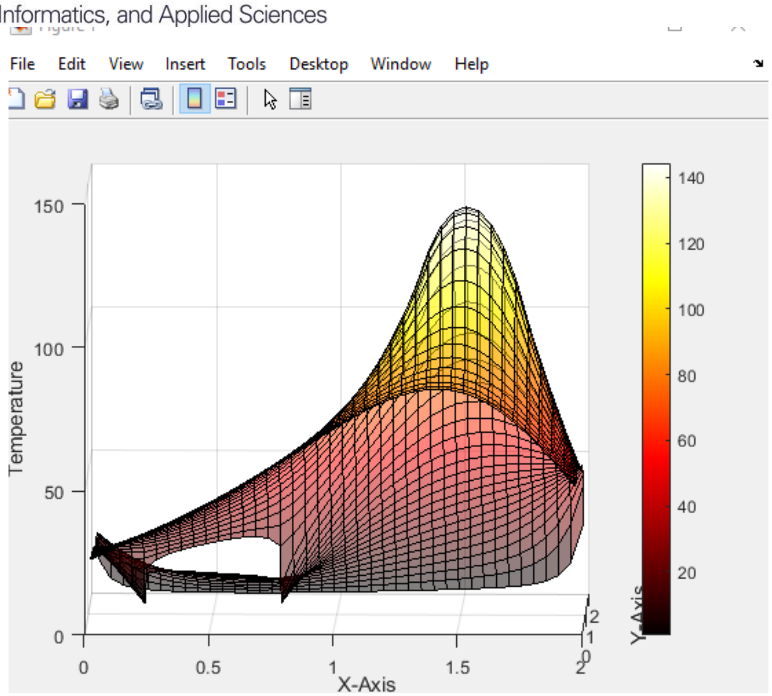


Figure 4: T2 Boundary Condition

T1 follows a similar pattern with most of the nodal temperatures being close to 25 degrees, but the top half of the right edge will be slightly cooler than the bottom half due to the 0 degrees condition at the top of the geometry and the generation being on the bottom half.

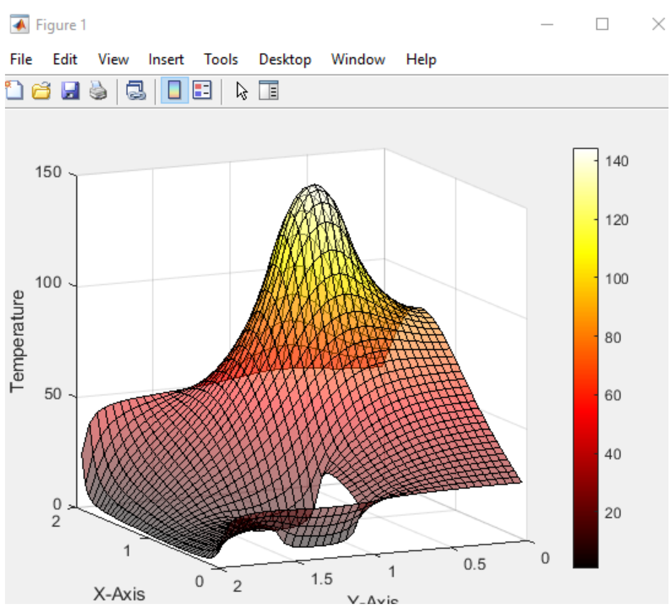


Figure 5: T1 Boundary Condition

The insulated boundary condition would mean the temperatures at the bottom edge are only influenced by the internal nodes. The ones closer to the generation block will show a large spike, followed by a decrease in temperature as you travel right across the geometry.

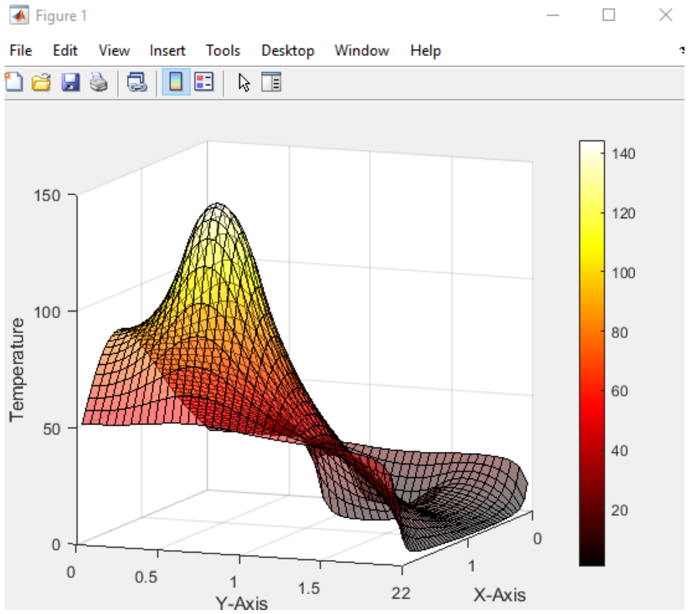


Figure 6: Insulated Boundary Condition

To ensure my calculated temperatures stayed within a .5% difference when doubling my mesh size, I took a 3x3 section of the temperature profile and calculated the percent differences of each value. As seen in the table, the calculations are within the desired range, and further helps to validate the results.

n = 20			**n = 40**		
24.62125	38.26397	43.10863	24.53411	37.99178	42.61782
11.4635	25.59	33.83864	11.21288	24.80697	32.42625
7.106254	18.79387	27.51293	6.723307	17.59697	25.35249

Percent Differences

0.003545	0.007139	0.011451
0.022104	0.031074	0.042629
0.055381	0.06578	0.081733

Results

The following graphs are the 3D temperature maps, the first being the one generated by my code, and the second being the key provided.

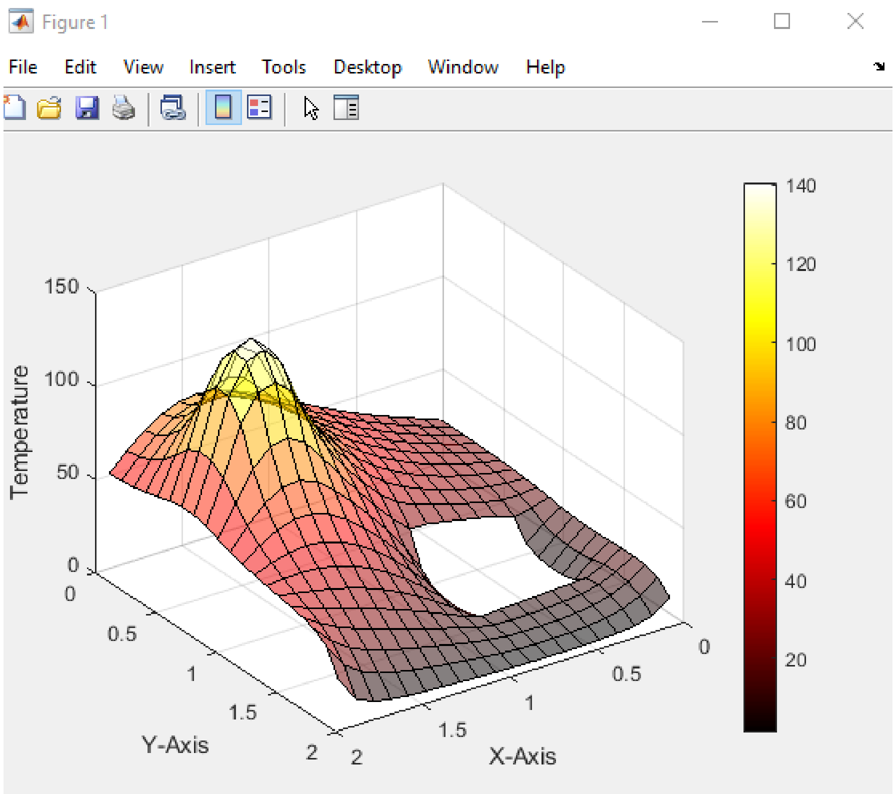


Figure 7: Generated 3D Temperature Map

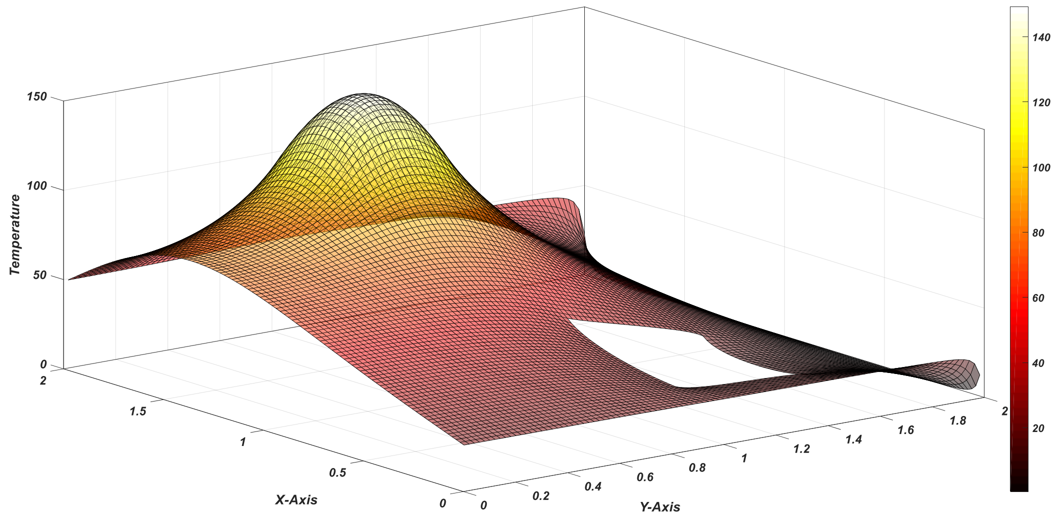


Figure 8: Key 3D Temperature Map

The generated graph matches the given plot. This leads me to believe my code is sufficient in mapping temperatures across the 2D plate.

Plots of the boundaries are also provided, and match the temperature profiles discussed in the validation portion of the assignment. The provided key is also attached to confirm the results.

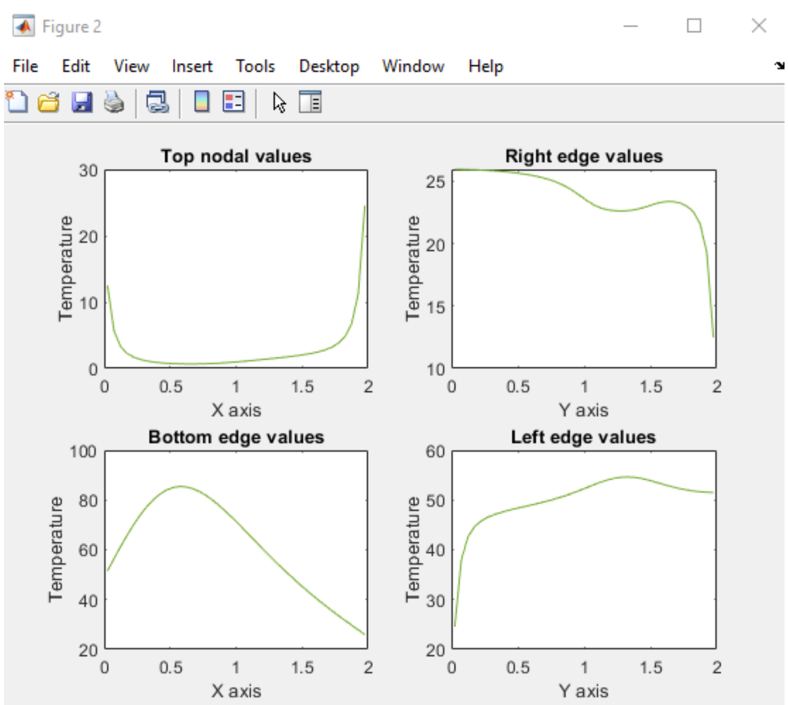


Figure 9: Temperature Profiles

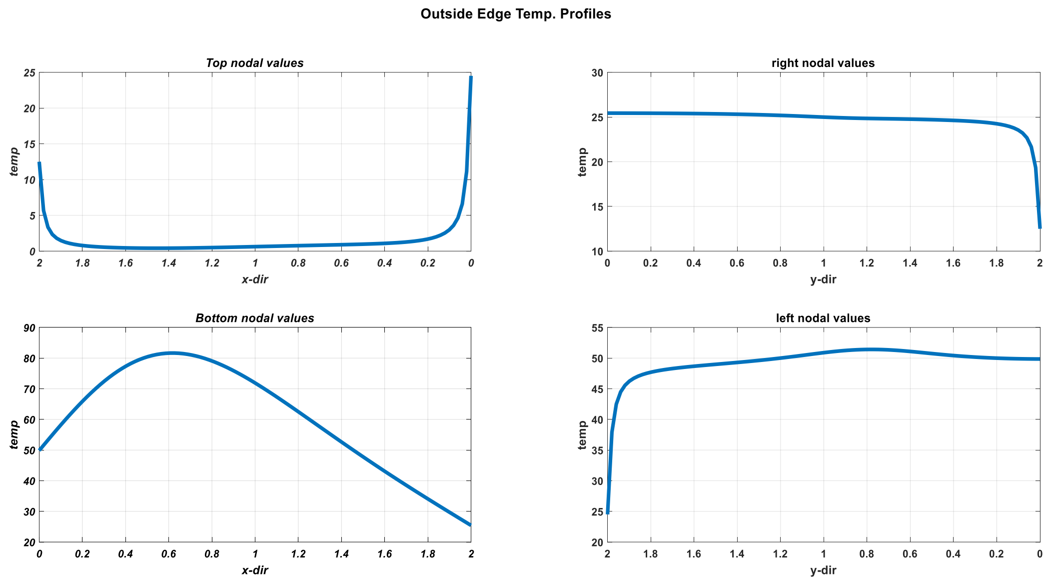


Figure 10: Key Temperature Profiles

Complications

Part B of the code validation was the first major issue I ran into during the project.

```
LeftEdge = sum(2*k*(T2-T(:,1)))
TopEdge = sum(2*k*(T3-T(1,end)))
RightEdge = sum(2*k*(T1-T(:,1)))

qboundary = LeftEdge + TopEdge + RightEdge
```

My initial attempt was to do this portion similar to the nodal equations, but was sure there was a more efficient way to do it. The results from this portion of code does not equal the generation in the geometry. I believe to get the correct results I would also need to include each corner node of the geometry.

The second complication in the project was correctly mapping the convective edge temperature profiles. Below are the figures I obtained when attempting to plot the edges.

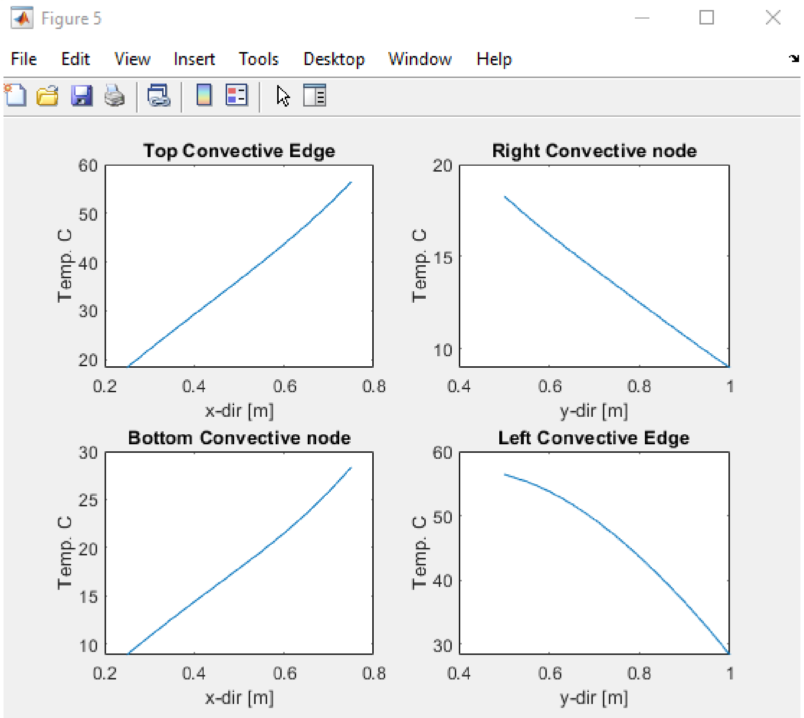


Figure 11: Generated Convective Temperature Edges

Code

```
% Abel Aldape
% ME 450 Heat Transfer
% Dr. Carson Pete
% Computer Project


close all, clc
n = 40 %Initialize matrix size
C = zeros(n);%Creates matrix of zeroes
A =zeros(n*n); %Creates A matrix of unknown variables
b = zeros(n*n,1);%Creates b matrix of known variables
T1 = 25;
T2 = 49;
T3 = 0;
Tinf = 15;
k = 15; %(W/mK)
h_inf = 52; %(W/m^2K)
GEN = 35000; %(W/m^2)
dx=2/n;
[X,Y]=meshgrid(2-dx/2:-dx:0,dx/2:dx:2);
xx=reshape(X',n^2,1);                  % x's as a vector
yy=reshape(Y',n^2,1);                  % y's as a vector


%Fill in matrix with approporate nodal equations
for i = 1:n^2
```

```
if i == 1 %Top Left Corner
    A(i,i) = -6; %T
    A(i,i+1) = 1; %TE
    A(i,i+n) = 1; %TS
    b(i) = (-2*T2)-(2*T3);%KNOWNS
    C(i) = 1; %NODE

 elseif i > 1 && i < n %Top Edge
    A(i,i) = -5;%T
    A(i,i+1) = 1;%TE
    A(i,i-1) = 1;%TW
    A(i,i+n) = 1;%TS
    b(i) = (-2*T3);%KNOWNS
    C(i) = 2; %NODE

 elseif i == n %Top Right Corner
    A(i,i) = -6;%T
    A(i,i-1) = 1;%TW
    A(i,i+n) = 1;%TS
    b(i) = (-2*T3)-(2*T1);%KNOWNS
    C(i) = 3; %NODE

 elseif mod(i,n) == 0 && i ~=n && i ~=n^2 %Right Edge
    A(i,i) = -5;%T
    A(i,i-1) = 1;%TW
    A(i,i+n) = 1 ;%TS
    A(i,i-n) = 1;%TN
    b(i) = (-2*T1);%KNOWNS
    C(i) = 4; %NODE

 elseif i == n^2 %Bottom Right Corner
    A(i,i) = -4;%T
    A(i,i-1) = 1;%TW
    A(i,i-n) = 1;%TN
    b(i) = -2*T1;%KNOWNS
    C(i) = 5;%NODE

  elseif i > n^2-n+1 && i < n^2 %Bottom Edge
    A(i,i) = -3;%T
    A(i,i+1) = 1;%TE
    A(i,i-1) = 1;%TW
    A(i,i-n) = 1;%TN
    b(i) = 0;%KNOWNS (INSULATED)
    C(i) = 6;%NODE

   elseif i == n^2-n+1 %Bottom Left Corner
    A(i,i) = -4;%T
    A(i,i+1) = 1;%TE
    A(i,i-n) = 1;%TN
    b(i) = (-2*T2);%KNOWNS
    C(i) = 7;%NODE

   elseif mod(i,n) == 1 && i ~= 1 && i ~= n^2-n+1 %Left Edge
    A(i,i) = -5;%T
```

```
        A(i,i+1) = 1;%TE
        A(i,i+n) = 1;%TS
        A(i,i-n) = 1;%TN
        b(i) = (-2*T2); %KNOWNS
        C(i) = 8; %NODE

        elseif xx(i) >= 1.35 && xx(i)<=1.75 && yy(i)>=1.1 && yy(i)<=1.5
%Gerneation Block
        A(i,i) = -4;%T
        A(i,i+1) = 1;%TE
        A(i,i-1) = 1;%TW
        A(i,i+n) = 1;%TS
        A(i,i-n) = 1;%TN
        b(i) = -GEN*(dx^2/k);
        C(i) = 10;

        elseif xx(i) >= 0.25 && xx(i)<=0.75 && yy(i)>=0.5 && yy(i)<=1
%Convection Block (empty)
        A(i,i)=1;
        b(i) =0; %KNOWNS
        C(i) = 15;%NODE



        elseif xx(i) >= 0.25 && xx(i)<=0.75 && yy(i)>=0.5-dx && yy(i)<=0.5
%Top of convection block
        A(i,i) = -(h_inf*dx/k)-3;%T
        A(i,i+1) = 1;%TE
        A(i,i-1) = 1;%TW
        A(i,i-n) = 1;%TN
        b(i) = -((h_inf*dx)/k)*Tinf;
       C(i) = 11; %NODE

        elseif xx(i) >= 0.75 && xx(i)<=0.75+dx && yy(i)>=0.5 && yy(i)<=1
%Left of convection block
        A(i,i) = -(3 +(h*dx/k));%T
        A(i,i+1) = 1;%TE
        A(i,i+n) = 1;%TS
        A(i,i-n) = 1;%TN
        b(i) = -((h_inf*dx)/k)*Tinf; %KNOWNS
       C(i) = 12; %NODE

       elseif xx(i) >= 0.25 && xx(i)<=0.75 && yy(i)>=1 && yy(i)<=1+dx %Bottom
of convection block
        A(i,i) = -(h_inf*dx/k)-3;%T
        A(i,i+1) = 1;%TE
        A(i,i+n) = 1;%TS
        A(i,i-1) = 1;%TW
        b(i) = -((h_inf*dx)/k)*Tinf; %KNOWNS
       C(i) = 13; %NODE

       elseif xx(i) >= 0.25-dx && xx(i)<=0.25 && yy(i)>=0.5 && yy(i)<=1
%Right of convection block
        A(i,i) =-(3+(h_inf*dx/k));%T
        A(i,i+n) = 1;%TS
        A(i,i-1) = 1;%TW
```

```
        A(i,i-n) = 1;%TN
        b(i) = -((h_inf*dx)/k)*Tinf; %KNOWNS
       C(i) = 14; %NODE

        else %Internal Nodes
        A(i,i) = -4;%T
        A(i,i-1) = 1;%TW
        A(i,i+n) = 1;%TS
        A(i,i-n) = 1;%TN
        A(i,i+1) = 1;%TE
        b(i)=0; %KNOWNS
         C(i) = 9;%NODE


    end
end
    C' %Displays the matrix with node locations


T = A\b; %Solve for Temperatures
B=b;     %For Boundary Profiles
b  = reshape(b,[n,n])'; %scales b plot
T = reshape(T,[n,n]);   %scales temp plot
T(T==0)=NaN;

figure(1)
rotate(surf(X,Y,T,'FaceAlpha',0.5),[0 0 1],90)
colorbar
colormap hot
xlabel({'X-Axis'});
ylabel({'Y-Axis'});
zlabel({'Temperature'});
hold on

T=A\B;
T = reshape(T,[n,n]);

figure(2)
p1 = subplot(2,2,1);
plot(Y,flip(((T(:,1))))); %1st subplot
title('Top nodal values');
ylabel('Temperature');
xlabel('X axis');

sb2 = subplot(2,2,2);
plot(Y,flip((T(end,:)))); %2nd subplot
title('Right edge values');
ylabel('Temperature');
xlabel('Y axis');

sb3 = subplot(2,2,3); plot(Y,((T(:,end)))); %3rd subplot
title('Bottom edge values');
ylabel('Temperature');
xlabel('X axis');
```

```
sb4 = subplot(2,2,4); plot(Y,((T(1,:)))); %4th subplot
title('Left edge values');
ylabel('Temperature');
xlabel('Y axis');

T1 = reshape(T,[n,n]);
 T(T ==0) = NaN;

 figure(5)
 subplot (2,2,4)%Left Convective Edge
 plot(.5:dx:1,(T(.25*n:.5*n,.375*n)))
 %plot(0:dx:.4,flip(Treshape(.8*n/2:n/2,round(LRow*n,0))))
 title('Left Convective Edge')
 xlabel('y-dir [m]');
 ylabel('Temp. C');

%Top Convective Edge
subplot (2,2,1)
plot(0.25:dx:0.75,T(.25*n,.125*n:.375*n))
title('Top Convective Edge')
xlabel('x-dir [m]');
ylabel('Temp. C');

%Subplot 7 Right Convective Edge
subplot (2,2,2)
plot(.5:dx:1,T(.25*n:.5*n,.125*n))
title('Right Convective node')
xlabel('y-dir [m]');
ylabel('Temp. C');

subplot (2,2,3)
plot(0.25:dx:0.75,T(.5*n,.125*n:.375*n))
title('Bottom Convective node')
xlabel('x-dir [m]');
ylabel('Temp. C');
%Validation

LeftEdge = sum(2*k*(T2-T(:,1)))
TopEdge = sum(2*k*(T3-T(1,end)))
RightEdge = sum(2*k*(T1-T(:,1)))

qboundary = LeftEdge + TopEdge + RightEdge
```