**TEMPERATURE DISTRIBUTION SIMULATION REPORT**

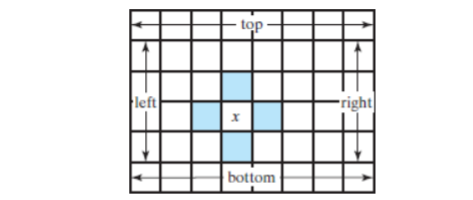
**Step 1: Problem Identification and Statement**

The objective is to design a software that helps to model a thin metal plate with constant (isothermal) temperatures on each side using a 2D grid. The software asks the user to input the row and column dimensions of the matrix (between 50 and 200), the constant temperatures for the first and last rows and columns of the matrix (between 0 and 255), and the threshold value (between 0 and 0.05). In return, the software provides four visual displays for the temperature distribution. There are two graphs which display the matrix as an image- one at time equals to zero and one at time when the loop breaks. The two other graphs show the matrix as a contour- one at time equals to zero and one at time when the loop breaks. If the user enters an out of range value for any of the inputs, the software asks the user to re-enter the values until a valid value is inputted.

**Step 2: Gathering Information**

**Relevant information:**

Typically, the number of points in the grid is specified (N number of rows by M number of columns), as are the constant temperatures on the four sides. The temperatures of the interior points are initialized to zero, but they change according to the temperatures around them. The temperature of an interior point can be computed as the average of the four adjacent temperatures; the points shaded in the figure below represent the adjacent temperatures for the pointed labelled x in the grid.



Each time that the temperature of an interior point changes, the temperatures of the points adjacent to it change. These changes continue until a thermal equilibrium is achieved and all temperatures become constant. A thermal equilibrium is reached when the difference between the previous and current temperature of every element in the grid becomes smaller than a specific threshold (entered by the user).

**Input/output Description:**

Row Dimensions

Surface Contours at final time

Heat dissipation at final time

Surface Contours at time=0

Boundary Conditions at time=0

Column Dimensions

First row temperature

Last row temperature

First column temperature

Tolerance Value

Last column temperature

The following explains how the program executes.

>> Assignment

Enter the number of rows:200

Enter the number of columns:200

Enter the boundary conditions for the top row of the matrix:0

Enter the boundary conditions for the bottom row of the matrix:0

Enter the boundary conditions for the left column of the matrix:0

Enter the boundary conditions for the right column of the matrix:0

Enter a tolerance value between 0 and 0.05:0.01

**Step 3: Test Case and Algorithm**

**Test cases:**

**Test Case 1: Invalid Dimensions**

>> Assignment

Enter the number of rows:25

Enter the number of columns:25

Enter the number of rows:

**Test Case 2: Invalid Boundary Conditions**

>> Assignment

Enter the number of rows:200

Enter the number of columns:200

Enter the boundary conditions for the top row of the matrix:300

Enter the boundary conditions for the top row of the matrix:200

Enter the boundary conditions for the bottom row of the matrix:-25

Enter the boundary conditions for the bottom row of the matrix: 10

Enter the boundary conditions for the left column of the matrix:-800

Enter the boundary conditions for the left column of the matrix: 30

Enter the boundary conditions for the right column of the matrix:700

Enter the boundary conditions for the right column of the matrix:100

**Test Case 3: Invalid Tolerance Value**

>> Assignment

Enter the number of rows:200

Enter the number of columns:200

Enter the boundary conditions for the top row of the matrix:250

Enter the boundary conditions for the bottom row of the matrix:250

Enter the boundary conditions for the left column of the matrix: 250

Enter the boundary conditions for the right column of the matrix:250

Enter a tolerance value between 0 and 0.05: 10

Tolerance value is out of range. Please re-enter value: -5

Tolerance value is out of range. Please re-enter value: 0.02

**Test Case 4: Hot edges**

>> Assignment

Enter the number of rows:200

Enter the number of columns:200

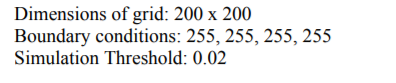
Enter the boundary conditions for the top row of the matrix:255

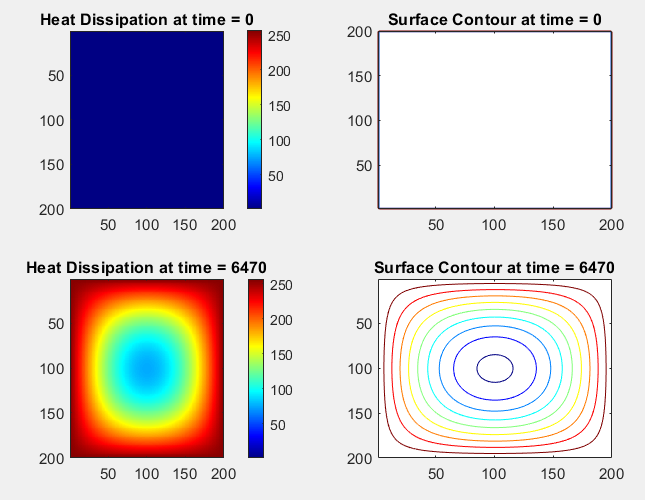
Enter the boundary conditions for the bottom row of the matrix:255

Enter the boundary conditions for the left column of the matrix: 255

Enter the boundary conditions for the right column of the matrix:255

Enter a tolerance value between 0 and 0.05: 0.02





**Test Case 5: Mixed boundary conditions**

>> Assignment

Enter the number of rows:200

Enter the number of columns:200

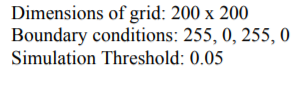
Enter the boundary conditions for the top row of the matrix:255

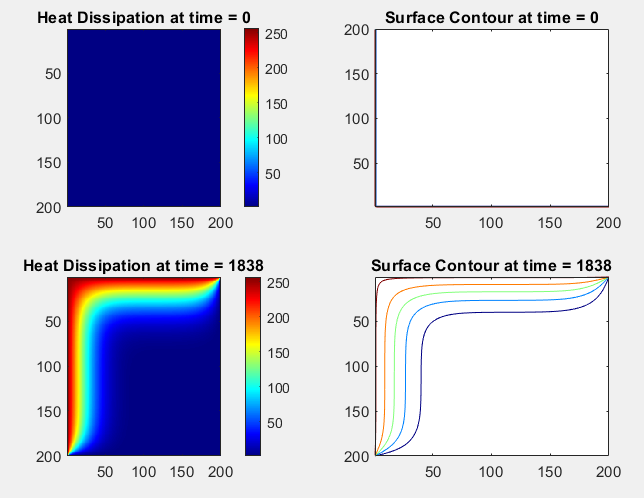
Enter the boundary conditions for the bottom row of the matrix:0

Enter the boundary conditions for the left column of the matrix: 255

Enter the boundary conditions for the right column of the matrix: 0

Enter a tolerance value between 0 and 0.05: 0.05





**Test Case 6: Zero boundary conditions**

>> Assignment

Enter the number of rows:200

Enter the number of columns:200

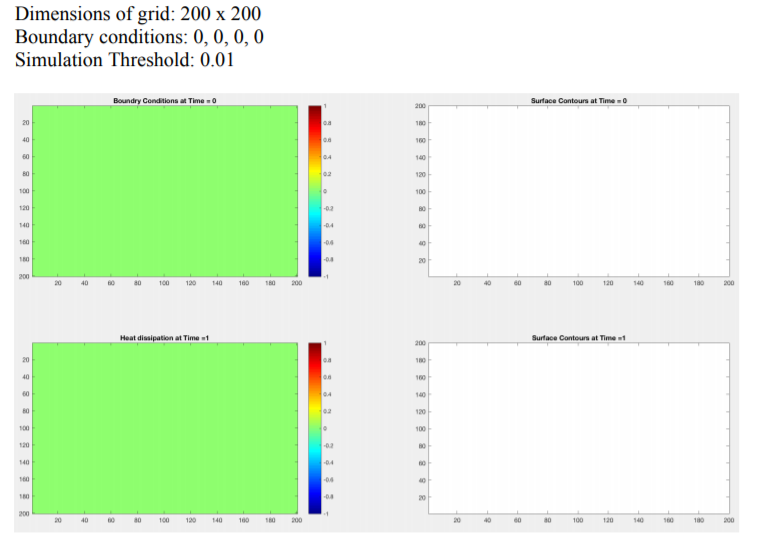
Enter the boundary conditions for the top row of the matrix:0

Enter the boundary conditions for the bottom row of the matrix:0

Enter the boundary conditions for the left column of the matrix: 0

Enter the boundary conditions for the right column of the matrix: 0

Enter a tolerance value between 0 and 0.05: 0.01



**Algorithm design**:

Assign 0 to rowsize

Assign 0 to columnsize

Assign vector with values from 50 to 200 with step 1 to dimension

While not any of the dimension values equal to rowsize or columnsize

Print 'Enter the number of rows: ' and assign input to rowsize

Print 'Enter the number of columns: ' and assign input to columnsize

End while loop

Assign a matrix of zeros of rowsize rows and columnsize columns to mat

Print 'Enter the boundary conditions for the top row of the matrix:' and assign input to all columns in row 1 in mat

While row 1, column 1 of mat is greater than 255 or row 1, column 1 of mat is less than 0

Print 'Enter the boundary conditions for the top row of the matrix:' and assign input to all columns in row 1 in mat

End while loop

Print 'Enter the boundary conditions for the bottom row of the matrix:’ and assign input to all columns in last row

While row rowsize, column 1 of mat is greater than 255 or row rowsize, column 1 of mat is less than 0

Print 'Enter the boundary conditions for the bottom row of the matrix:’ and assign input to all columns in last row

End while loop

Print 'Enter the boundary conditions for the left column of the matrix:’ and assign input to all rows from 2 to rowsize -1 in column 1

While row 2, column 1 of mat is greater than 255 or row 2, column 1 of mat is less than 0

Print 'Enter the boundary conditions for the left column of the matrix:’ and assign input to all rows from 2 to rowsize -1 in column 1

End while loop

Print 'Enter the boundary conditions for the right column of the matrix:’ and assign input to all rows from 2 to rowsize-1 in last column

While row rowsize and column 1 of mat is greater than 255 or row rowsize and column 1 of mat is less than 0

Print 'Enter the boundary conditions for the right column of the matrix:’ and assign input to all rows from 2 to rowsize-1 in last column

End while loop

Print 'Enter a tolerance value between 0 and 0.05: 'and assign input to tolerance

While tolerance is greater than 0.05 or tolerance is less than 0

Print 'Tolerance value is out of range. Please re-enter value: ') and assign input to tolerance

End while loop

Assign 0 to t

Break the figure window into a 2-by-2 matrix of small axes and select the first axes for the current plot

Display matrix mat as an image

Add as a title at the top of the current axis 'Heat Dissipation at time = ‘ ‘t’ after converting t to characters

Set the current figure’s colormap to jet

Display colorbar

Break the figure window into a 2-by-2 matrix of small axes and select the second axes for the current plot

Draw a contour plot of matrix mat

Add as a title at the top of the current axis ‘Surface Contour at time = ‘ ‘t’ after converting t to characters

Assign a matrix of zeros of rowsize rows and columnsize columns to newMat

Assign a matrix of zeros of rowsize rows and columnsize columns to tempdiff

While true

Assign mat from row 2 to rowsize-1 and column 2 to columnsize-1 to newMat from row 2 to rowsize-1 and column 2 to columnsize-1

Assign ((mat from row 2 to rowsize-1 and column 1 to columnsize-2) + (mat from row 1 to rowsize-2 and column 2 to columnsize-1) + (mat from row 2 to rowsize-1 and column 3 to columnsize) + (mat from row 3 to rowsize and column 2 to columnsize-1))/4 to mat from row 2 to rowsize-1 and column 2 to columnsize-1

Assign (mat from row 2 to rowsize-1 and column 2 to columnsize-1) – (newMat from row 2 to rowsize-1 and column 2 to columnsize-1) to tempdiff

Assign t+1 to t

Break the figure window into a 2-by-2 matrix of small axes and select the third axes for the current plot

Display matrix mat as an image

Add as a title at the top of the current axis 'Heat Dissipation at time = ‘ ‘t’ after converting t to characters

Set the current figure’s colormap to jet

Display colorbar

Pause for 0.00005 seconds before continuing

Break the figure window into a 2-by-2 matrix of small axes and select the fourth axes for the current plot

Draw a contour plot of matrix mat

Add as a title at the top of the current axis ‘Surface Contour at time = ‘ ‘t’ after converting t to characters

Pause for 0.00005 seconds before continuing

Reverse the graph’s y-axis

If all values in tempdiff are less than tolerance

Break while loop

End if loop

End while loop

**Step 4: Code or Implementation**

The C++ program to solve our problem is listed below.

%{

Name: Ishmal Khalid, Net ID: ik1299

Date: December 5th, 2019.

Program:Assignment.m

Description: This program models the temperature distribution in a thin

metal plate.

%}

%initialize row and column size

rowsize = 0;

columnsize = 0;

%create a vector for dimension

dimension = 50:1:200;

%validate row and column size inputs by repeating loop until input is valid

while ~any(rowsize == dimension | columnsize == dimension)

%take user input for row and column size

rowsize= input('Enter the number of rows:');

columnsize= input('Enter the number of columns:');

end

%specify the dimensions of a matrix and initialize it to 0

mat = zeros(rowsize, columnsize);

%validate input for top row of matrix by repeating loop until input is valid

mat(1,:) = input('Enter the boundary conditions for the top row of the matrix:');

while (mat(1,1) > 255 || mat(1,1) < 0)

mat(1,:) = input('Enter the boundary conditions for the top row of the matrix:');

end

%validate input for bottom row of matrix by repeating loop until input is valid

mat(rowsize,:) = input('Enter the boundary conditions for the bottom row of the matrix:');

while (mat(rowsize,1)> 255 || mat(rowsize,1) < 0)

mat(rowsize,:) = input('Enter the boundary conditions for the bottom row of the matrix: ');

end

%validate input for left column of matrix by repeating loop until input is valid

mat(2:(rowsize-1),1)= input('Enter the boundary conditions for the left column of the matrix: ');

while (mat(2,1) > 255 || mat(2,1) < 0)

mat(2:rowsize-1,1)= input('Enter the boundary conditions for the left column of the matrix: ');

end

%validate input for right column of matrix by repeating loop until input is valid

mat(2:(rowsize-1),columnsize)= input('Enter the boundary conditions for the right column of the matrix: ');

while (mat(2,columnsize) > 255 || mat(2,columnsize) < 0)

mat(2:rowsize-1,columnsize)= input('Enter the boundary conditions for the right column of the matrix: ');

end

%validate input for tolerance value by repeating loop until input is valid

tolerance = input('Enter a tolerance value between 0 and 0.05: ');

while (tolerance > 0.05 || tolerance < 0)

tolerance = input('Tolerance value is out of range. Please re-enter value: ');

end

%initialize time to 0

t = 0;

%plot the first graph for Heat Dissipation at initial time

subplot(2,2,1)

image(mat) %create graph of matrix as an image

title(['Boundary Conditions at time = ', num2str(t)]) %give title

colormap(jet),colorbar %set color

%plot the second graph for surface contours at initial time

subplot(2,2,2)

contour(mat) %create contour graph

title(['Surface Contour at time = ', num2str(t)]) %give title

%specify the dimensions of matrices and initialize them to 0

newMat = zeros(rowsize, columnsize);

tempdiff = zeros(rowsize, columnsize);

%repeat loop while true

while 1

%copy mat into newMat

newMat(2:(rowsize-1),2:(columnsize-1))= mat(2:(rowsize-1),2:(columnsize-1));

%calculate values for each cell in mat

mat(2:(rowsize-1),2:(columnsize-1))= (mat(2:(rowsize-1),1:(columnsize-2)) + mat(1:(rowsize-2), 2:(columnsize-1)) + mat(2:(rowsize-1),3:columnsize) + mat(3:(rowsize),2:(columnsize-1)))/4;

%calculate the difference between all cells in mat and newMat

tempdiff(2:(rowsize-1),2:(columnsize-1)) = mat(2:(rowsize-1),2:(columnsize-1))-newMat(2:(rowsize-1),2:(columnsize-1));

%increment time after every loop

t = t + 1;

%plot the third graph for Heat Dissipation at final time

subplot(2,2,3)

image(mat) %create graph of matrix as an image

title(['Heat Dissipation at time = ', num2str(t)]) %give title

colormap(jet),colorbar %set color

pause(0.00005) %pause

%plot the fourth graph for surface contours at final time

subplot(2,2,4)

contour(mat) %create contour graph

title(['Surface Contour at time = ', num2str(t)]) %give title

pause(0.00005) %pause

set(gca, 'ydir', 'reverse') %reverse y axis

%break while loop if all tempdiff values are less than tolerance

if all(tempdiff < tolerance)

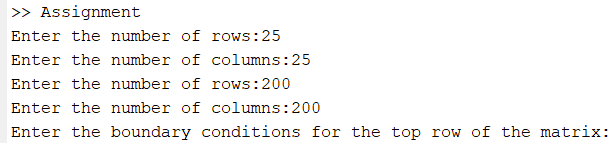
break;

end

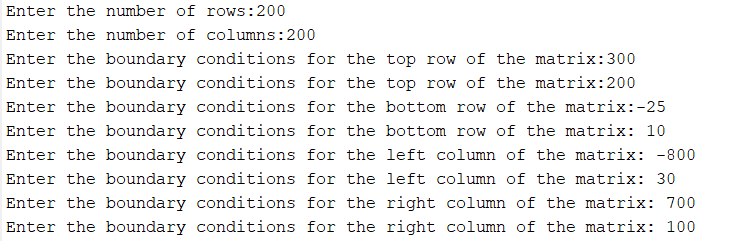
end

**Step 5: Test and Verification**

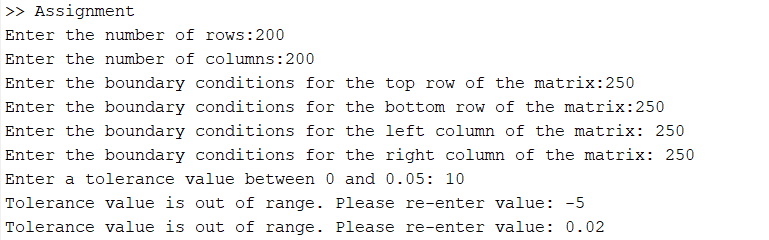
**Test Case 1: Invalid Dimensions**



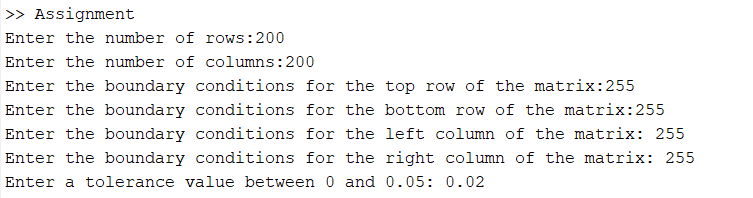
**Test Case 2: Invalid Boundary Conditions**

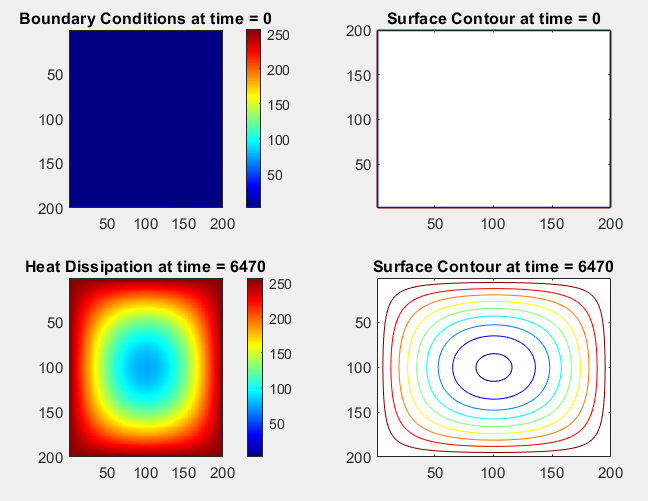


**Test Case 3: Invalid Tolerance Value**

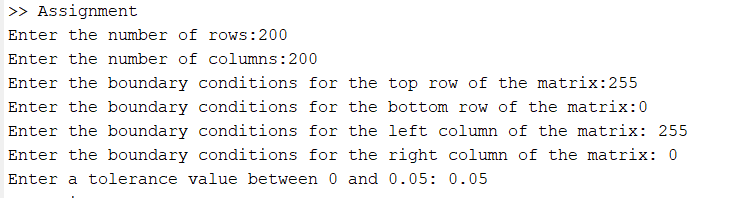


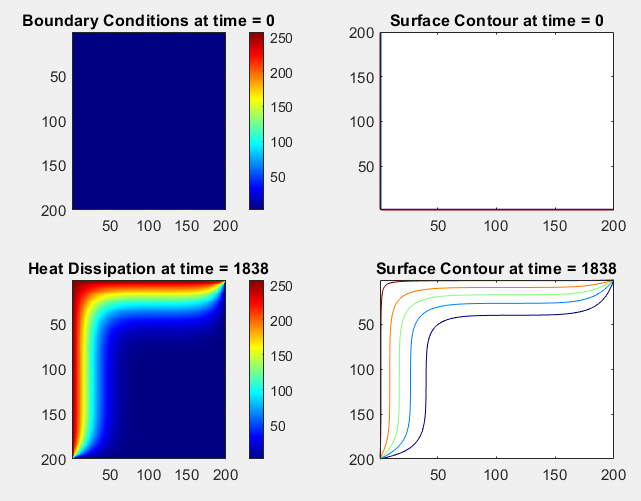
**Test Case 4: Hot edges**





**Test Case 5: Mixed boundary conditions**





**Test Case 6: Zero boundary conditions**

