Evergreen Valley College

HEAT CONDUCTION
Numerical Computations
Project Report
Fall 2022

Thuong Mai
Engineering 50 - Intro to Computing
Dr AH Tabrizi

Table of Contents:

1. Introduction:

1.1 Background

This project is a computing program in C++ to estimate the alternative surface temperature distribution of a very long rectangular solid steel bar by breaking it to various locations within the surface and using the Finite Difference approach to roughly calculate the temperature allocation within the bar. Dev C++ 5.11 and “dislin” libraries are utilized.

1.2 Project Overview:

Initially at 20 degrees Celsius, a long steel bar was placed in a temperature environment as shown in the figure. Determine the temperature distribution throughout the surface of the metal bar by chopping it into smaller areas and using a numerical differentiation method to estimate the temperature.

1.3 Objectives:

- Gain experience in writing a user-friendly C++ computer program that emphasizes an object-oriented approach.
- Develop mastery in creating and working with classes, intrinsic and extrinsic functions, program controls using loops, and if-else structure.
- Develop mastery in working with arrays and various data types.
- Gain experience in engineering problem-solving through numerical computation.
- Develop mastery in advance input/out operations including working with external data files.

2. Procedure:

2.1 Project Details.

A very long rectangular solid steel bar initially at 20 degrees Celsius is exposed to an environment that can alter the heat distribution within the bar. The top surface of the bar is under the influence of 400 degrees Celsius, where the sides are affected by 200 degrees Celsius and the

bottom facing 100 degrees heat sources as shown in the diagram. We are interested in determining the temperature variation within the bar as the bar heats up and its final steady-state temperature distribution.
And the given temperature of the steel bar is:

$T(t\ =\ 0)\ =\ 20\ ^{0}C$

$T\ top\ =\ 400\ ^{0}C$

$T\ right\ =\ 200\ ^{0}C$

$T\ left\ =\ 200\ ^{0}C$

$T\ bottom\ =\ 100\ ^{0}C$

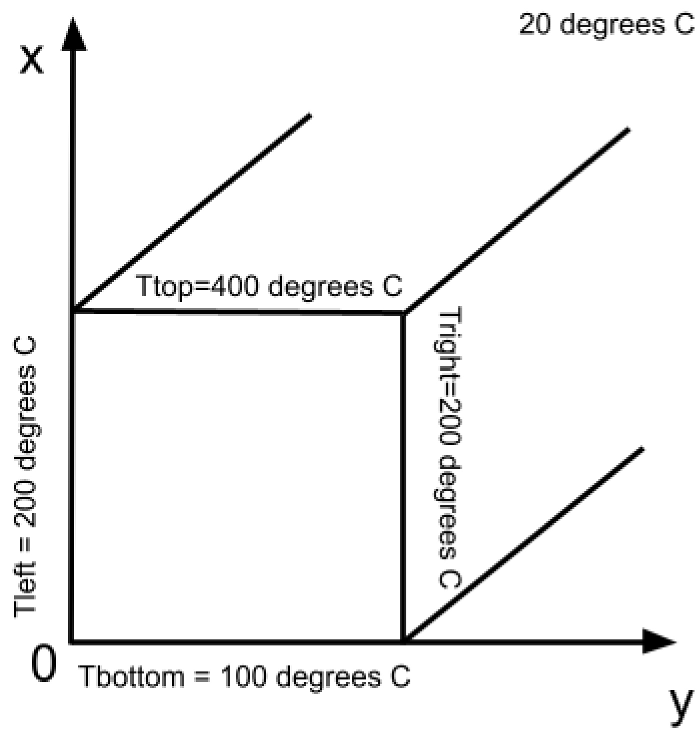

Figure 1: The coordinate system for the long bar.

By breaking down the surface in the plane XY into smaller increments in the xy directions, and applying the Finite Difference Method through the Partial Differential Equation.

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = \frac{1}{\alpha} \cdot \frac{\partial T}{\partial t} \qquad \text{(eq 1)}$$

where:
T= Temperature (C)
t= time (s)
x= Coordinate along the width of the bar (m)
y= Coordinate along the length of the bar (m)
Diffusivity (k/pc): $\alpha = 1.888 \bullet 10^{-5} (m/s^2)$

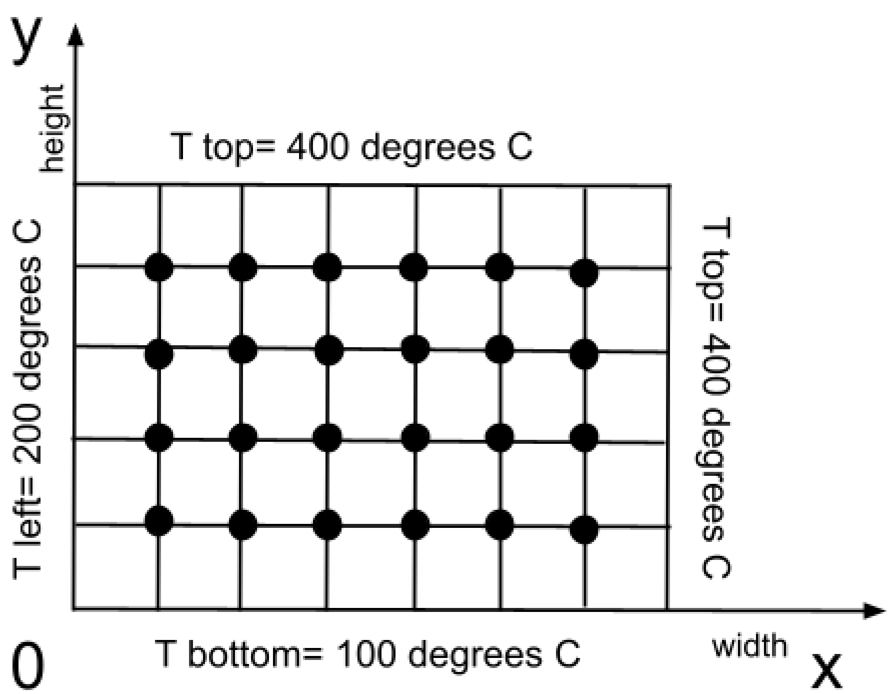

Figure 2: The coordinate surface breakdown of the metal bar.

2.2 Lists of Class, Functions, Variables.

```
class temp
{
	public:
		string file;
		void input(int &m, int &n);
		void function_1(int m, int n, float**T,float *Tcenter,
string &file );
		void function_2(int m, int n,float **T,string file);
		void temp_0(int m, int n, float **T,string file);
		void time_lot(int m,int n,float *Tcenter,string file);
		void T_contour_plot(int m, int n,float **T);
		void T_contour_shade_plot(int m, int n,float **T);
		void T_x_plot(int m, int n,float **T);
		void T_y_plot(int m, int n,float **T);
		double get_dx();
		double get_dy();
		double get_dt();
	private:
		double dx;
		double dy;
		double dt;
		float *Tcenter;
		float **T;
		float **t_half;

};
	ofstream write;
	string file;
	write.open(file.c_str());
```

3. Equations and Numerical Computation Analysis:

3.1 Taylor's Series and Finite Difference Approach.

Taylor's Series formula:

$$f(x+x_0) = f(x) + \frac{x_0}{1!}\cdot\frac{d^2 f}{dx^2} + \frac{(x_0)^2}{2!}\cdot\frac{d^2 f}{dx^2} + \ldots + \frac{(x_0)^n}{n!}\cdot\frac{d^n f}{dx^n} \quad (\text{ eq } 2\ )$$

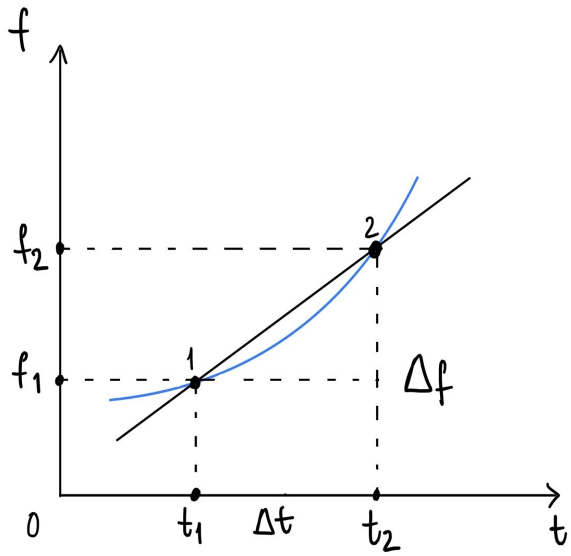

Figure 3: The slope of the derivative. Recall from Calculus, the function f(t) as shown in the figure. To equate the function, the slope of the line connecting the derivatives at point t1 and t2, which is called the tangent line can be approximated as:

$$\frac{df}{dt} = \frac{\Delta t}{\Delta x} = \frac{f(t2)-f(t1)}{t2-t1} \quad \text{(eq 3)}$$

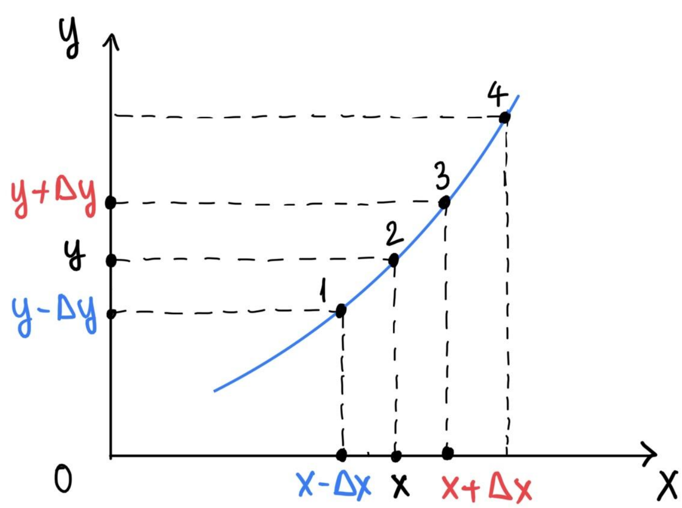

Figure 4: The series of the derivatives. Applied to the function y=T(x), as T is denoted as temperature function.To understand the error of this approximation, by looking at the Taylor's Series expansion of the T(x)=T(x+Δx) about a base point of point of T(x). Expanding the function at the two locations in the x-axis by many terms as shown below:

$$T(x + \Delta x) = T(x) + \frac{\Delta x}{1!} \cdot \frac{dT}{dx} + \frac{(\Delta x)^2}{2!} \cdot \frac{d^2T}{dx^2} + \ldots + \frac{(\Delta x)^n}{n!} \cdot \frac{d^nT}{dx^n} \quad \text{(eq 4)}$$

$$T(x - \Delta x) = T(x) - \frac{\Delta x}{1!} \cdot \frac{dT}{dx} + \frac{(-\Delta x)^2}{2!} \cdot \frac{d^2T}{dx^2} + \ldots + \frac{(-\Delta x)^n}{n!} \cdot \frac{d^nT}{dx^n} \quad \text{(eq5)}$$

The two Taylor's Series approximations are going to be used to understand the error of the expression by plugging the equations 4 and 5 into the equation in equation 3 in terms of x and T(x). This method is called forward, backward, and central difference to the first derivative.

3.1.1 Forward, Backward, and Central Differencing:

a/ Forward Differencing

$$\frac{dT}{dt}=\frac{T(x+\Delta x)-T(x)}{\Delta x}=\frac{dT}{dx}+\frac{\Delta x}{2!}\cdot\frac{d^2T}{dx^2}+\frac{(\Delta x)^2}{3!}\cdot\frac{d^3T}{dx^3}+\dots \quad \text{(eq 6)}$$

From the equation 6 (eq 6), the error is dependent on Δx. In order to optimize the error, by controlling the value of the Δx to be smaller, the error $\frac{\Delta x}{2!}\cdot\frac{d^2T}{dx^2}+\dots+\frac{(\Delta x)^{n-1}}{n!}\cdot\frac{d^nT}{dx^n}$ getting towards 0 and becomes neglectable.

The equation then:

$$\frac{dT}{dx}=\frac{T(x+\Delta x)-T(x)}{\Delta x} \quad \text{(eq 7)}$$

Same for y and T(y):

$$\frac{dT}{dy}=\frac{T(y+\Delta y)-T(y)}{\Delta y} \quad \text{(eq 8)}$$

b/ Backward Differencing:

$$\frac{dT}{dt}=\frac{T(x)-T(x-\Delta x)}{\Delta x}=\frac{dT}{dx}-\frac{\Delta x}{2!}\cdot\frac{d^2T}{dx^2}+\dots-\frac{(\Delta x)^{n-1}}{n!}\cdot\frac{d^nT}{dx^n} \quad \text{(eq 9)}$$

Repeat the process, the error is dependent on Δx. Same as in forward differencing, backward differencing also optimizes the error by controlling the value of the Δx to be smaller, the error $\frac{\Delta x}{2!}\cdot\frac{d^2T}{dx^2}+\dots+\frac{(\Delta x)^{n-1}}{n!}\cdot\frac{d^nT}{dx^n}$ getting towards 0 and becomes so small that we can neglect it.

The equation then:

$$\frac{dT}{dx}=\frac{T(x)-T(x-\Delta x)}{\Delta x} \quad \text{(eq 10)}$$

Same for y and T(y):

$$\frac{dT}{dy} = \frac{T(y) - T(y-\Delta y)}{\Delta y} \text{ (eq 11)}$$

c/ Central Differencing:

$$\frac{dT}{dx} = \frac{T(x+\Delta x) - T(x-\Delta x)}{2\Delta x} \quad \text{(eq 12)}$$

$$\frac{dT}{dy} = \frac{T(y+\Delta y) - T(y-\Delta y)}{2\Delta y} \quad \text{(eq 13)}$$

3.1.2 Second Derivatives:

As the equation 1 requires to include the second derivatives and the error is neglectable, the partial derivative is extracted by adding the two Taylor's Series expansion equation (4) and (5) shown below:

$$T(x + \Delta x) + T(x - \Delta x) = 2T(x) + 2(\frac{(-\Delta x)^2}{2!} \cdot \frac{d^2T}{dx^2})$$

$$= 2T(x) + (\Delta x)^2 \cdot \frac{d^2T}{dx^2} \qquad \text{(eq 14)}$$

Move the derivative values to the left hand side:

$$\frac{d^2T}{dx^2} = \frac{T(x+\Delta x) - 2T(x) + T(x-\Delta x)}{(\Delta x)^2} \quad \text{(eq 15)}$$

Same for Δy:

$$\frac{d^2T}{dy^2} = \frac{T(y+\Delta y) - 2T(y) + T(y-\Delta y)}{(\Delta y)^2} \qquad \text{(eq 16)}$$

The derivative of temperature in terms of time is:

$$\frac{dT}{dt} = \frac{T(x)_{t+\Delta t} - T(x)_t}{\Delta t} \qquad \text{(eq 17)}$$

Interpreting the partial derivative equation (eq1) in terms of the expansions, the analytical equation to estimate the heat distribution within the region of the bar's surface is:

$$\frac{T(x+\Delta x) - 2T(x) + T(x-\Delta x)}{(\Delta x)^2} + \frac{T(y+\Delta y) - 2T(y) + T(y-\Delta y)}{(\Delta y)^2} = \frac{1}{\alpha} \cdot \frac{T(x)_{t+\Delta t} - T(x)_t}{\Delta t}$$

3.2 . Applied Program Analysis.

The program is written to assist the users in estimating the temperature changing in linear time and predict the behavior of the metal bar being 4 sides heated by different values of temperature by applied theorized differential equations. The magnitude of the long bar is given as height of the bar is 0.05m and width of the bar is 0.02m. The temperature 4 sides are distributed as the figure. Breaking down the 2 dimensional surface (2D matrix) to the grid containing points known as numerical allocations.

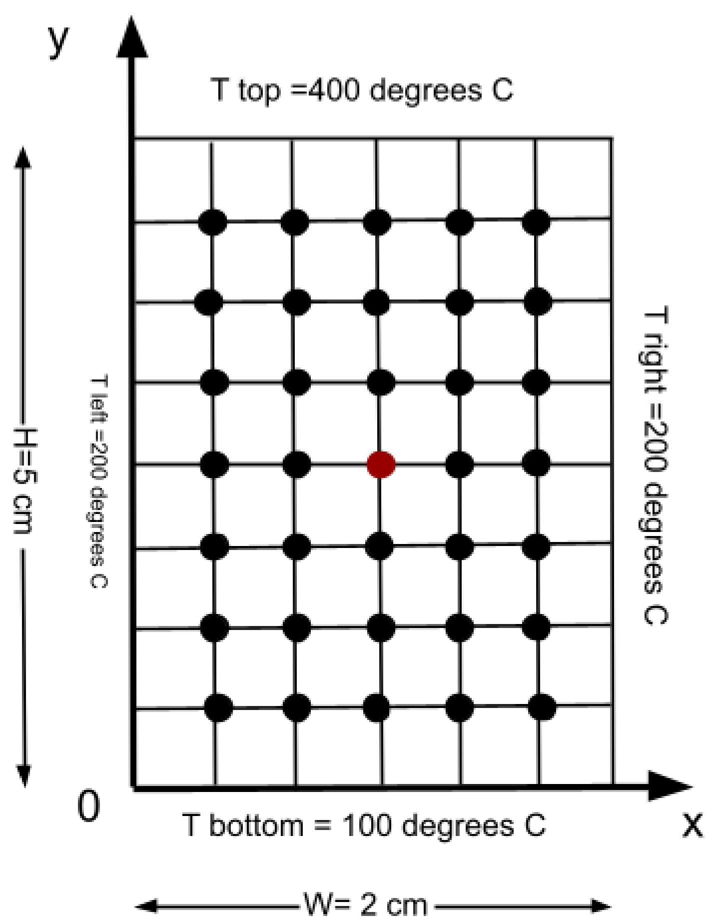

Figure 5 : The fractionated surface of the metal bar.

The program has the users input the value of the number of rows and number of columns of the surface as they wish to test. In fact, the magnitude of the material can be input as the users develop to experiment more with different sizes and materials in the future.The program is robust, dynamic, and user-friendly.

The relationships between Δx *(dx)* and Δy *(dy)* versus numbers of row *(m)* and number of column *(n)* as declared below:

```
void  temp::input(int &m, int &n)
            {
                  float width=0.02, height=0.05;
                  dx=width/(n-1);
                  dy=height/(m-1);
            }
```

As the result of applying the forward and backward differentiation, we equalize the equation to approximate the inside surface temperature $T(x)_{t+\Delta t}$ through each allocation as shown (figure 5) .

$$T(x)_{t+\Delta t} = \alpha \cdot \Delta t \cdot \left(\frac{T(x+\Delta x) - 2T(x) + T(x-\Delta x)}{(\Delta x)^2} + \frac{T(y+\Delta y) - 2T(y) + T(y-\Delta y)}{(\Delta y)^2}\right)$$

By calling a 2D dynamic array `double **T` , the size of the array does not need to be initialized and fixed. This is a very user-friendly and efficient method of calling an array since the users have more than one input to test out with different sizes of the 2D array to reduce the inaccuracy and uncertainty.

```
double **T= new double *[m];
            for (int i=0;i<m;i++)
            {
                  T[i]=new double [n];
            }
```

Initialize the primary temperature of the inner surface at time t=0, where i starts from 1 to the number of rows minus 1, and j starts from 1 to the number of columns minus 1. The program will set the values of each point on the side. In this case the number of rows is m-1 because the loop index is at 0 and the same is applied for the number of columns.

```
for (int i=1;i<m-1;i++)
      {
            for (int j=1; j<n-1;j++)
            {
                T[i][j]=20;
        }
    }
```

Then, initialize the temperature of the 4 sides of the metal bar.

```
for (int i=0;i<m;i++)
      {
            T[i][0]=200;
            T[i][n-1]=200;
      }
      for (int j=0;j<n ;j++)
      {
            T[0][j]=100;
            T[m-1][j]=400;
      }
```

In order to test the equation will be able to function, iteratively doing the calculation by using 2 nested for loops, in which i is for row and j is for column, inside a while loop acts as time.

```
while (time<=1300)
{
      for (int i=1;i<m-1;i++)
      {
            for (int j=1; j<n-1;j++)
            {
            T[i][j]=(( T[i+1][j] - 2*T[i][j] + T[i-1][j]
)/pow(dx,2) +    ( T[i][j+1] - 2*T[i][j] + T[i][j-1]
)/pow(dy,2))*a*dt+T[i][j];
            }

      }
      time++;
            Tcenter[time]=T[m/2][n/2];
}
}
```

4. Final Results.

The program is convenient, accessible, and adaptable in terms of user's commands and needs. As the heat is steadily distributed through the surface of the long metal bar, the breakdown of the surface can be adjustable for users to output as much data as they need to certain the errors and accurate approximations.

4.1 Tables and Graphs for 10x10 Matrix Surfaces.

Case 1: Table and graph of the 10 by 10 matrix.

Suppose user was to input number of rows and columns as m=10,n=10.

The temperature at time t = 0 is:									
100	100	100	100	100	100	100	100	100	100
200	20	20	20	20	20	20	20	20	200
200	20	20	20	20	20	20	20	20	200
200	20	20	20	20	20	20	20	20	200
200	20	20	20	20	20	20	20	20	200
200	20	20	20	20	20	20	20	20	200
200	20	20	20	20	20	20	20	20	200
200	20	20	20	20	20	20	20	20	200
200	20	20	20	20	20	20	20	20	200
400	400	400	400	400	400	400	400	400	400

Table 1: The initial temperature of the metal bar's surface.

The temperature at half time t = 3.25 is:									
100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
200.00	129.95	117.84	114.18	112.91	112.91	114.19	117.87	129.98	200.00
200.00	151.88	136.43	130.47	128.23	128.23	130.49	136.47	151.94	200.00
200.00	170.90	157.38	150.80	148.07	148.08	150.83	157.45	170.98	200.00
200.00	190.56	182.49	176.79	174.07	174.07	176.83	182.57	190.65	200.00
200.00	213.57	213.23	209.47	207.16	207.17	209.51	213.31	213.66	200.00
200.00	242.40	250.55	249.13	247.48	247.49	249.17	250.62	242.48	200.00
200.00	279.88	294.74	295.20	294.24	294.25	295.23	294.79	279.94	200.00
200.00	330.12	345.20	346.24	345.84	345.84	346.25	345.23	330.15	200.00
400.00	400.00	400.00	400.00	400.00	400.00	400.00	400.00	400.00	400.00

Table 2: The temperature of the metal bar's surface at half time.

The temperature at time final t = 6.5 is:

100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
200.00	136.66	129.65	128.98	128.96	128.97	128.98	129.66	136.67	200.00
200.00	164.48	158.62	158.28	158.41	158.42	158.29	158.64	164.50	200.00
200.00	187.90	187.32	188.32	188.79	188.80	188.34	187.34	187.92	200.00
200.00	209.92	216.59	219.53	220.46	220.47	219.55	216.61	209.94	200.00
200.00	232.96	247.39	252.30	253.66	253.66	252.32	247.41	232.98	200.00
200.00	259.48	280.64	286.86	288.44	288.45	286.88	280.66	259.50	200.00
200.00	292.56	317.09	323.23	324.68	324.68	323.25	317.11	292.58	200.00
200.00	336.87	357.10	361.15	362.04	362.04	361.16	357.11	336.87	200.00
400.00	400.00	400.00	400.00	400.00	400.00	400.00	400.00	400.00	400.00

Table 3: The temperature of the metal bar's surface at steady state.

The temperature of the transient center versus t is:

t=0.5s	t=1s	t=1.5s	t=2s	t=2.5s	t=3s	t=3.5s	t=4s	t=4.5s	t=5s	t=5.5s	t=6s	t=6.5s
20.00	39.53	84.61	124.37	155.74	180.22	199.29	214.12	225.60	234.47	241.30	246.54	250.56
20.00	39.94	85.05	124.72	156.01	180.43	199.46	214.25	225.70	234.55	241.36	246.59	250.59
20.00	40.36	85.49	125.07	156.29	180.65	199.63	214.38	225.80	234.62	241.42	246.63	250.63
20.00	40.78	85.93	125.42	156.56	180.86	199.79	214.51	225.90	234.70	241.48	246.68	250.66
20.00	41.20	86.37	125.77	156.84	181.08	199.96	214.64	226.00	234.78	241.53	246.72	250.70
20.00	41.62	86.80	126.12	157.11	181.29	200.13	214.76	226.10	234.86	241.59	246.77	250.73
20.00	42.05	87.24	126.47	157.38	181.50	200.29	214.89	226.20	234.93	241.65	246.81	250.77
20.00	42.47	87.67	126.82	157.66	181.71	200.46	215.02	226.30	235.01	241.71	246.86	250.80
20.01	42.90	88.11	127.17	157.93	181.93	200.62	215.15	226.40	235.08	241.77	246.90	250.84
20.01	43.33	88.54	127.52	158.20	182.14	200.79	215.28	226.50	235.16	241.83	246.95	250.87
20.01	43.77	88.97	127.86	158.47	182.35	200.95	215.40	226.60	235.24	241.89	246.99	250.91
20.02	44.20	89.40	128.21	158.74	182.56	201.11	215.53	226.69	235.31	241.94	247.04	250.94
20.03	44.64	89.84	128.55	159.01	182.77	201.28	215.66	226.79	235.39	242.00	247.08	250.97
20.04	45.07	90.27	128.89	159.27	182.98	201.44	215.78	226.89	235.46	242.06	247.13	251.01
20.05	45.51	90.70	129.24	159.54	183.18	201.60	215.91	226.99	235.54	242.12	247.17	251.04
20.07	45.95	91.12	129.58	159.81	183.39	201.76	216.03	227.08	235.61	242.17	247.21	251.08

20.08	46.39	91.55	129.92	160.07	183.60	201.92	216.16	227.18	235.69	242.23	247.26	251.11
20.11	46.84	91.98	130.26	160.34	183.81	202.08	216.28	227.28	235.76	242.29	247.30	251.14
20.13	47.28	92.40	130.60	160.60	184.01	202.25	216.41	227.37	235.83	242.35	247.34	251.18
20.16	47.73	92.83	130.94	160.87	184.22	202.41	216.53	227.47	235.91	242.40	247.39	251.21
20.19	48.17	93.25	131.27	161.13	184.42	202.56	216.66	227.56	235.98	242.46	247.43	251.24
20.22	48.62	93.68	131.61	161.39	184.63	202.72	216.78	227.66	236.06	242.52	247.48	251.28
20.26	49.07	94.10	131.94	161.66	184.83	202.88	216.90	227.76	236.13	242.57	247.52	251.31
20.31	49.52	94.52	132.28	161.92	185.04	203.04	217.03	227.85	236.20	242.63	247.56	251.34
20.36	49.97	94.94	132.61	162.18	185.24	203.20	217.15	227.94	236.28	242.68	247.60	251.37
20.41	50.42	95.36	132.95	162.44	185.44	203.36	217.27	228.04	236.35	242.74	247.65	251.41
20.47	50.88	95.78	133.28	162.70	185.65	203.51	217.39	228.13	236.42	242.80	247.69	251.44
20.53	51.33	96.20	133.61	162.96	185.85	203.67	217.51	228.23	236.49	242.85	247.73	251.47
20.60	51.78	96.62	133.94	163.22	186.05	203.83	217.64	228.32	236.57	242.91	247.78	251.51
20.67	52.24	97.04	134.27	163.47	186.25	203.98	217.76	228.41	236.64	242.96	247.82	251.54
20.75	52.70	97.45	134.60	163.73	186.45	204.14	217.88	228.51	236.71	243.02	247.86	251.57
20.84	53.15	97.87	134.93	163.99	186.65	204.30	218.00	228.60	236.78	243.07	247.90	251.60
20.93	53.61	98.28	135.26	164.24	186.85	204.45	218.12	228.69	236.85	243.13	247.94	251.63
21.03	54.07	98.69	135.58	164.50	187.05	204.61	218.24	228.79	236.92	243.18	247.99	251.67
21.13	54.52	99.11	135.91	164.75	187.25	204.76	218.36	228.88	236.99	243.24	248.03	251.70
21.24	54.98	99.52	136.23	165.01	187.44	204.91	218.48	228.97	237.07	243.29	248.07	251.73
21.35	55.44	99.93	136.56	165.26	187.64	205.07	218.60	229.06	237.14	243.35	248.11	251.76
21.47	55.90	100.34	136.88	165.51	187.84	205.22	218.71	229.15	237.21	243.40	248.15	251.79
21.60	56.36	100.75	137.21	165.76	188.03	205.37	218.83	229.25	237.28	243.45	248.19	251.83
21.73	56.82	101.16	137.53	166.01	188.23	205.52	218.95	229.34	237.35	243.51	248.24	251.86
21.87	57.28	101.56	137.85	166.26	188.42	205.68	219.07	229.43	237.42	243.56	248.28	251.89
22.02	57.74	101.97	138.17	166.51	188.62	205.83	219.18	229.52	237.49	243.62	248.32	251.92
22.17	58.21	102.37	138.49	166.76	188.81	205.98	219.30	229.61	237.56	243.67	248.36	251.95
22.33	58.67	102.78	138.81	167.01	189.01	206.13	219.42	229.70	237.63	243.72	248.40	251.98
22.49	59.13	103.18	139.13	167.26	189.20	206.28	219.54	229.79	237.70	243.78	248.44	252.01
22.66	59.59	103.59	139.44	167.51	189.39	206.43	219.65	229.88	237.76	243.83	248.48	252.05
22.84	60.05	103.99	139.76	167.76	189.59	206.58	219.77	229.97	237.83	243.88	248.52	252.08
23.02	60.52	104.39	140.08	168.00	189.78	206.73	219.88	230.06	237.90	243.93	248.56	252.11
23.21	60.98	104.79	140.39	168.25	189.97	206.88	220.00	230.15	237.97	243.99	248.60	252.14
23.40	61.44	105.19	140.70	168.49	190.16	207.03	220.11	230.24	238.04	244.04	248.64	252.17
23.60	61.90	105.59	141.02	168.74	190.35	207.17	220.23	230.32	238.11	244.09	248.68	252.20

23.81	62.36	105.98	141.33	168.98	190.54	207.32	220.34	230.41	238.18	244.14	248.72	252.23
24.02	62.83	106.38	141.64	169.22	190.73	207.47	220.46	230.50	238.24	244.20	248.76	252.26
24.24	63.29	106.77	141.95	169.47	190.92	207.62	220.57	230.59	238.31	244.25	248.80	252.29
24.47	63.75	107.17	142.26	169.71	191.11	207.76	220.68	230.68	238.38	244.30	248.84	252.32
24.70	64.21	107.56	142.57	169.95	191.30	207.91	220.80	230.76	238.45	244.35	248.88	252.35
24.94	64.68	107.96	142.88	170.19	191.48	208.05	220.91	230.85	238.51	244.40	248.92	252.38
25.18	65.14	108.35	143.19	170.43	191.67	208.20	221.02	230.94	238.58	244.45	248.96	252.41
25.43	65.60	108.74	143.50	170.67	191.86	208.35	221.13	231.02	238.65	244.51	249.00	252.44
25.68	66.06	109.13	143.80	170.91	192.04	208.49	221.25	231.11	238.71	244.56	249.04	252.47
25.94	66.52	109.52	144.11	171.15	192.23	208.63	221.36	231.20	238.78	244.61	249.08	252.50
26.20	66.98	109.91	144.41	171.39	192.42	208.78	221.47	231.28	238.85	244.66	249.12	252.53
26.48	67.44	110.29	144.72	171.62	192.60	208.92	221.58	231.37	238.91	244.71	249.16	252.56
26.75	67.90	110.68	145.02	171.86	192.79	209.06	221.69	231.45	238.98	244.76	249.20	252.59
27.03	68.36	111.07	145.32	172.10	192.97	209.21	221.80	231.54	239.04	244.81	249.23	252.62
27.32	68.82	111.45	145.63	172.33	193.15	209.35	221.91	231.62	239.11	244.86	249.27	252.65
27.61	69.28	111.84	145.93	172.57	193.34	209.49	222.02	231.71	239.17	244.91	249.31	252.68
27.91	69.74	112.22	146.23	172.80	193.52	209.63	222.13	231.79	239.24	244.96	249.35	252.71
28.21	70.20	112.60	146.53	173.04	193.70	209.78	222.24	231.88	239.30	245.01	249.39	252.74
28.51	70.66	112.98	146.83	173.27	193.88	209.92	222.35	231.96	239.37	245.06	249.43	252.77
28.82	71.12	113.36	147.12	173.50	194.06	210.06	222.46	232.05	239.43	245.11	249.46	252.80
29.14	71.58	113.74	147.42	173.73	194.24	210.20	222.57	232.13	239.50	245.16	249.50	252.83
29.46	72.03	114.12	147.72	173.97	194.42	210.34	222.68	232.21	239.56	245.21	249.54	252.86
29.78	72.49	114.50	148.01	174.20	194.60	210.48	222.79	232.30	239.63	245.26	249.58	252.88
30.11	72.95	114.88	148.31	174.43	194.78	210.62	222.89	232.38	239.69	245.31	249.62	252.91
30.45	73.40	115.25	148.60	174.66	194.96	210.76	223.00	232.46	239.75	245.36	249.65	252.94
30.78	73.86	115.63	148.90	174.89	195.14	210.89	223.11	232.55	239.82	245.41	249.69	252.97
31.12	74.31	116.00	149.19	175.11	195.32	211.03	223.22	232.63	239.88	245.45	249.73	253.00
31.47	74.77	116.37	149.48	175.34	195.50	211.17	223.32	232.71	239.94	245.50	249.76	253.03
31.82	75.22	116.75	149.78	175.57	195.67	211.31	223.43	232.79	240.01	245.55	249.80	253.06
32.17	75.68	117.12	150.07	175.80	195.85	211.45	223.54	232.88	240.07	245.60	249.84	253.08
32.53	76.13	117.49	150.36	176.02	196.03	211.58	223.64	232.96	240.13	245.65	249.88	253.11
32.89	76.58	117.86	150.65	176.25	196.20	211.72	223.75	233.04	240.20	245.70	249.91	253.14
33.26	77.03	118.23	150.94	176.47	196.38	211.85	223.85	233.12	240.26	245.74	249.95	253.17
33.63	77.48	118.60	151.22	176.70	196.55	211.99	223.96	233.20	240.32	245.79	249.99	253.20
34.00	77.93	118.96	151.51	176.92	196.73	212.13	224.06	233.28	240.38	245.84	250.02	253.23

34.37	78.38	119.33	151.80	177.15	196.90	212.26	224.17	233.36	240.45	245.89	250.06	253.25
34.75	78.83	119.69	152.08	177.37	197.07	212.40	224.27	233.44	240.51	245.93	250.10	253.28
35.13	79.28	120.06	152.37	177.59	197.25	212.53	224.37	233.52	240.57	245.98	250.13	253.31
35.52	79.73	120.42	152.65	177.81	197.42	212.66	224.48	233.60	240.63	246.03	250.17	253.34
35.91	80.18	120.79	152.94	178.03	197.59	212.80	224.58	233.68	240.69	246.08	250.20	253.36
36.30	80.62	121.15	153.22	178.26	197.76	212.93	224.68	233.76	240.75	246.12	250.24	253.39
36.69	81.07	121.51	153.50	178.48	197.94	213.06	224.79	233.84	240.81	246.17	250.28	253.42
37.09	81.51	121.87	153.78	178.70	198.11	213.20	224.89	233.92	240.88	246.22	250.31	253.45
37.49	81.96	122.23	154.06	178.91	198.28	213.33	224.99	234.00	240.94	246.26	250.35	253.47
37.89	82.40	122.59	154.35	179.13	198.45	213.46	225.10	234.08	241.00	246.31	250.38	253.50
38.30	82.85	122.94	154.62	179.35	198.62	213.59	225.20	234.16	241.06	246.36	250.42	253.53
38.70	83.29	123.30	154.90	179.57	198.79	213.72	225.30	234.24	241.12	246.40	250.45	253.55
39.11	83.73	123.66	155.18	179.79	198.96	213.86	225.40	234.31	241.18	246.45	250.49	253.58
39.53	84.17	124.01	155.46	180.00	199.12	213.99	225.50	234.39	241.24	246.49	250.52	253.61

Table 4: The transient center temperature of the metal bar’s surface.

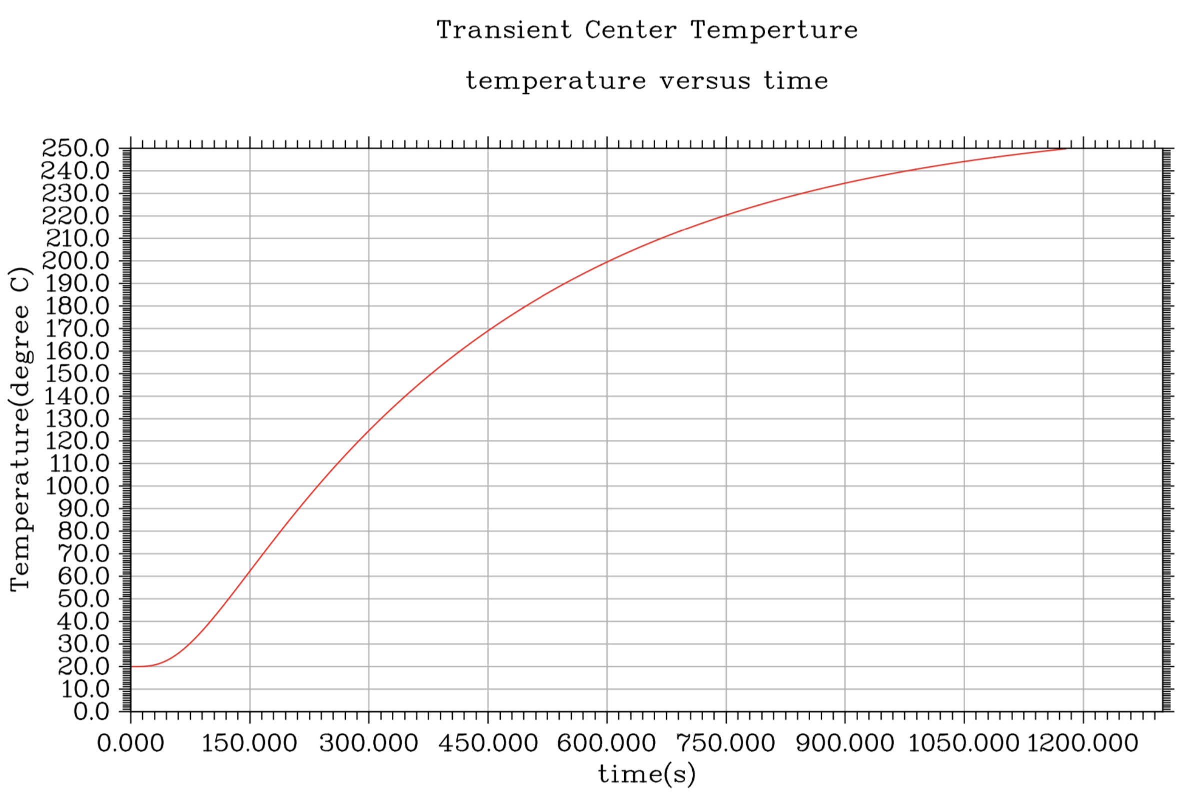

Figure 6: The Center Temperature of a (10x10).

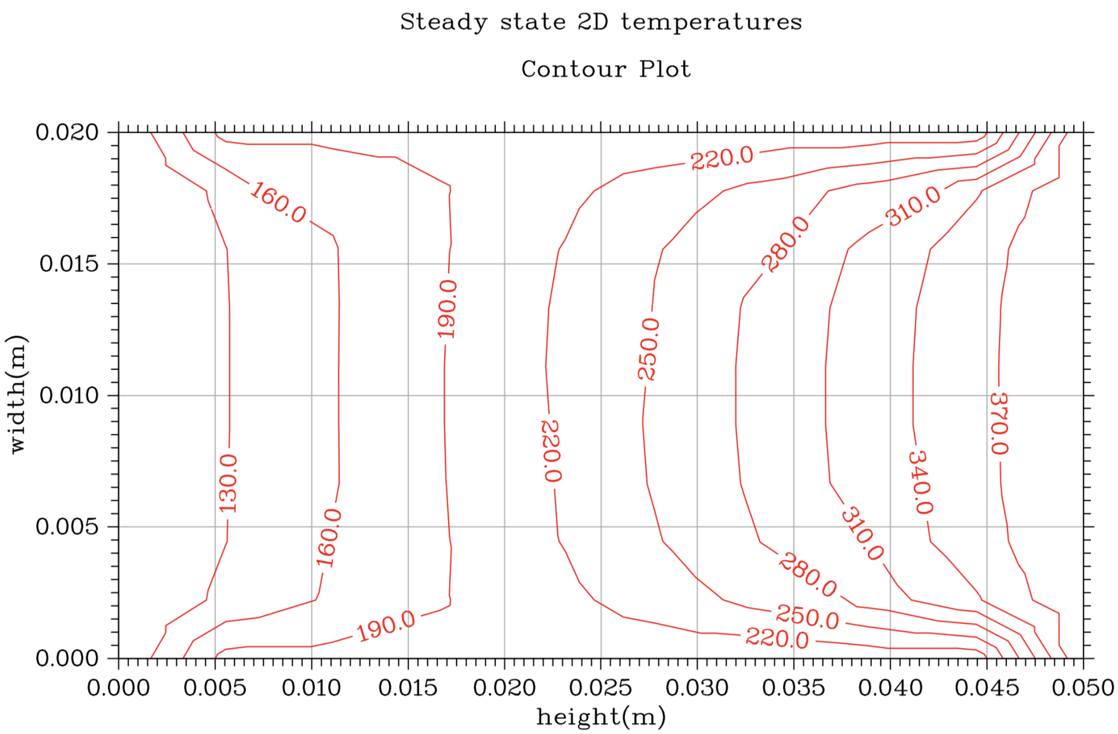

Figure 7: The contour plot of the 2D temperature of a (10x10).

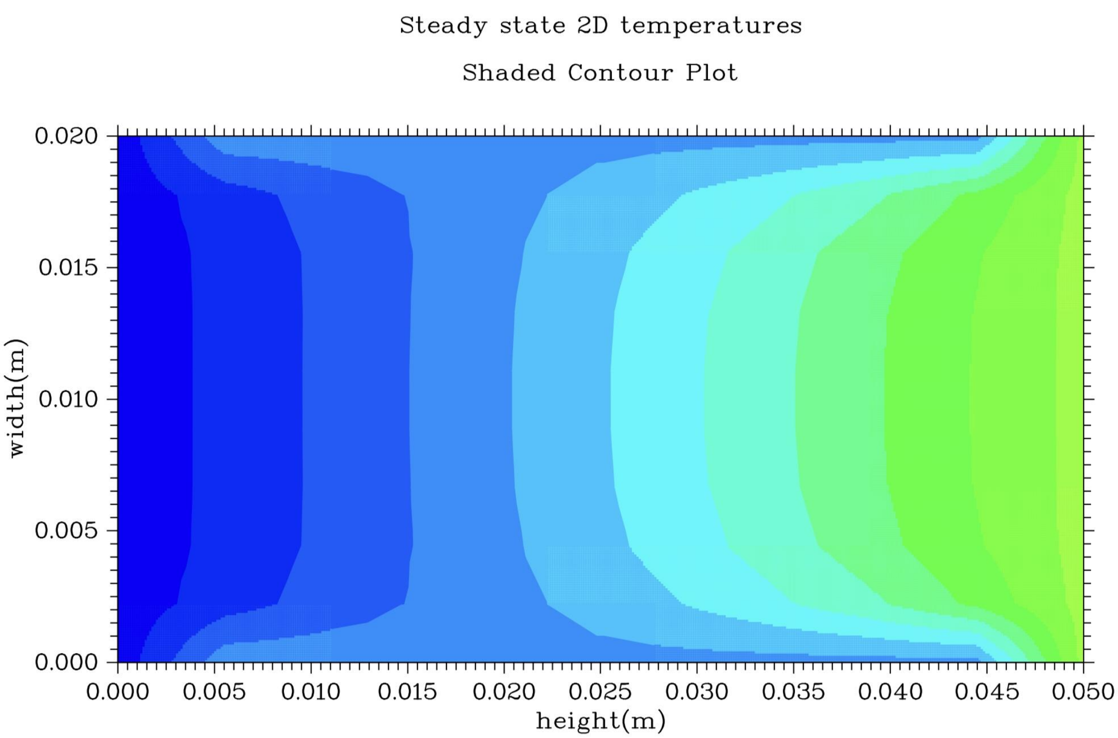

Figure 8: The shaded contour plot of the 2D temperature of (10x10).

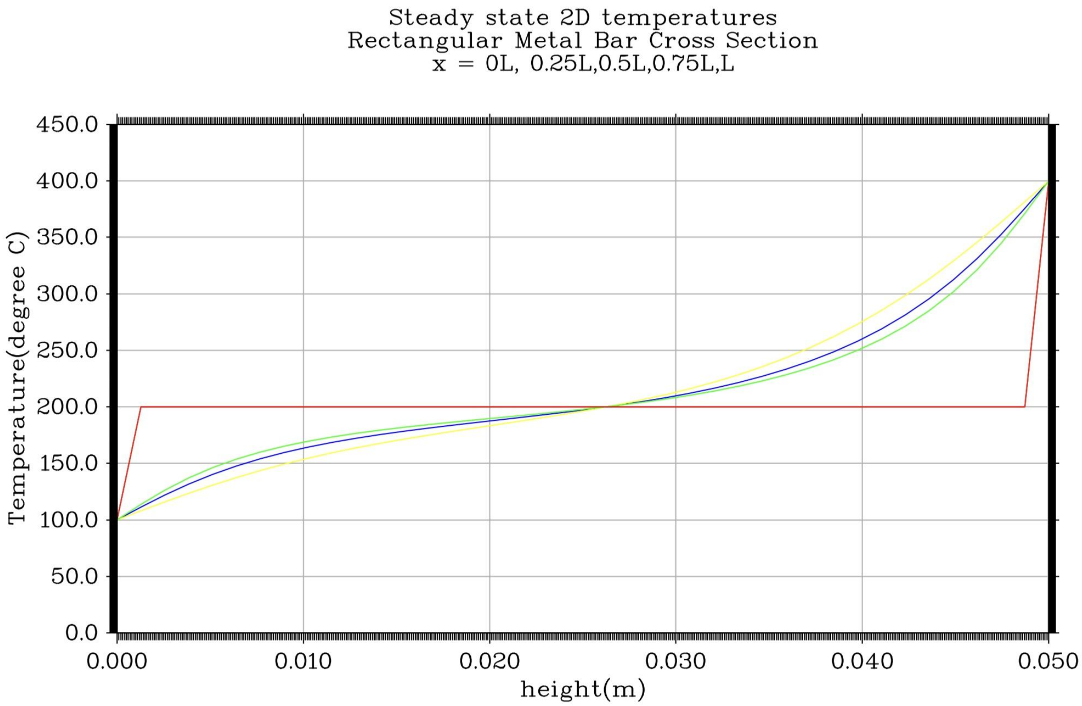

Figure 9: The temperature in the direction of x (10x10).

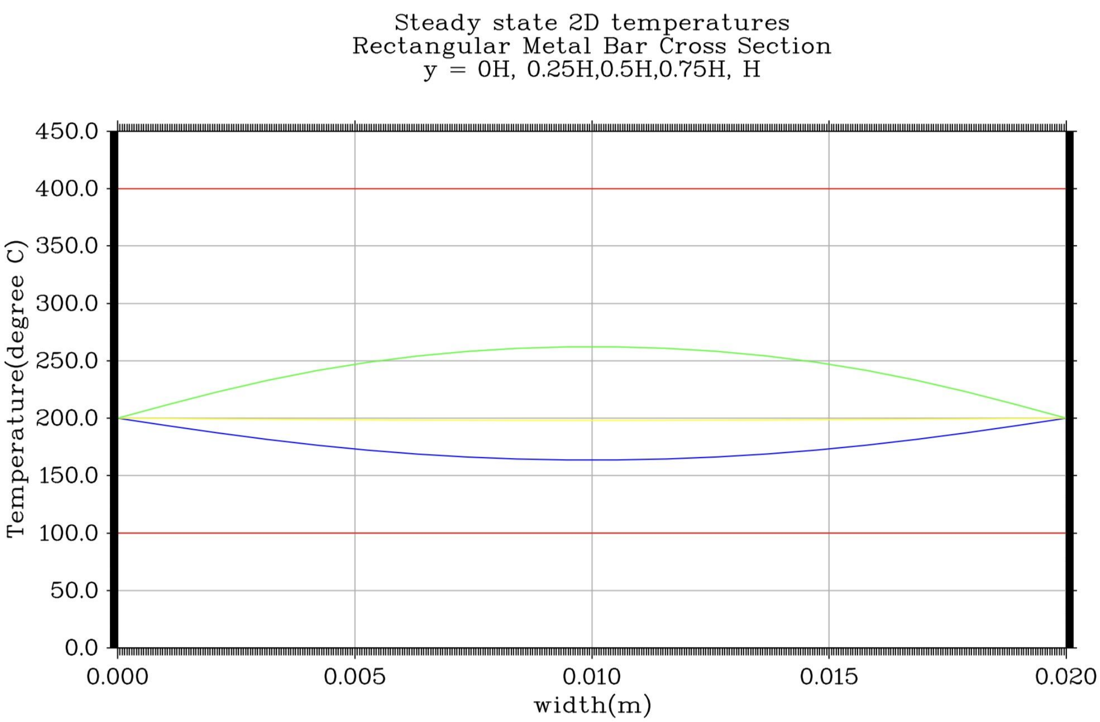

Figure 10: The temperature in the direction of y (10x10).

4.2 Graphs and Error Analysis.

Case 2:

Suppose the user input the number of rows and columns as:

m=40,n=20;

These plots below demonstrate the behavior of the temperature distributions within the surface as the grid was changed. As the height of the bar is longer than the width of the bar, to extract the smaller value of Δy, the number of rows should be more than the number of columns.

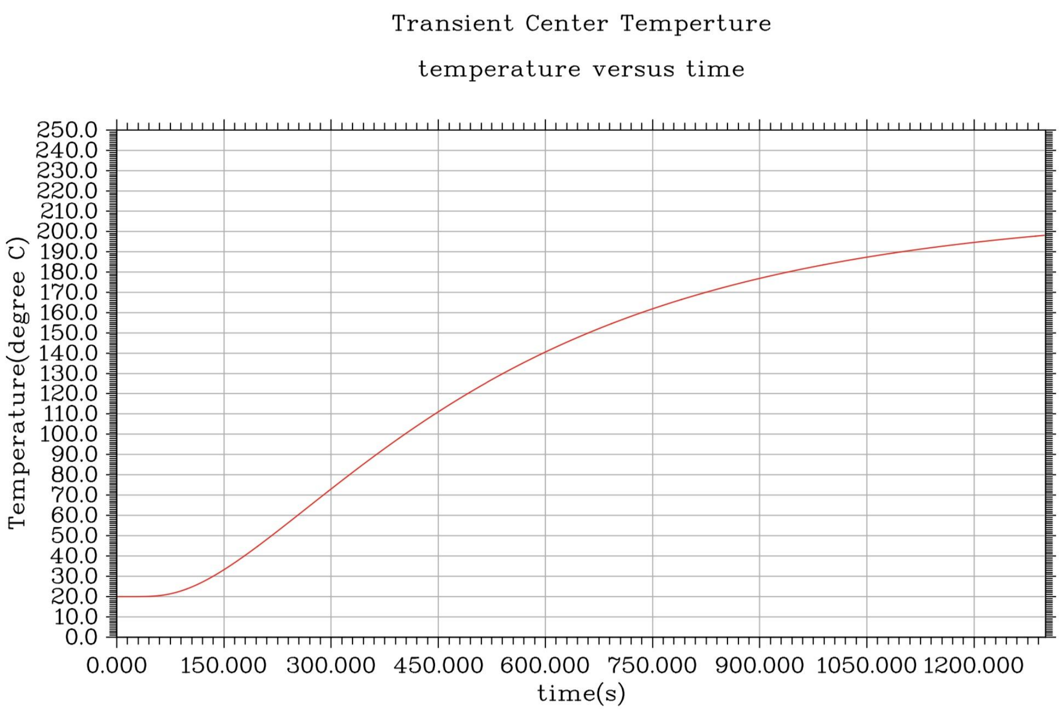

Figure 11 : The Center Temperature of a (40x20).

As in the figure , the 2D temperature of the surface is significantly rising from 20 degrees C to around 140 degrees Celsius in 3 seconds and gradually reaching to approximately 200 degrees Celsius.

Since the number of rows and columns are respectively reciprocal related to the increment Δx and Δy. And according to Taylor's polynomials, the error is neglectable if the value of Δ is small. And in order to have more accuracy, the grid should go through many trials to conclude the exact tendency of the heat distribution within the surface.

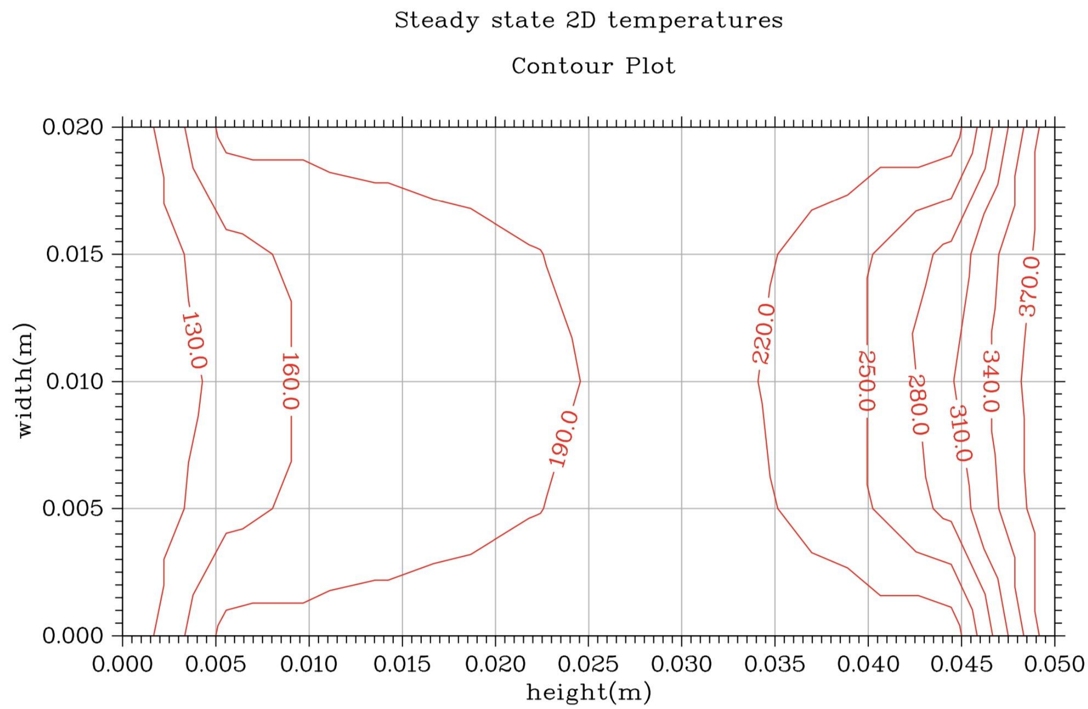

Figure 12: The contour plot of the 2D temperature of (40x20).

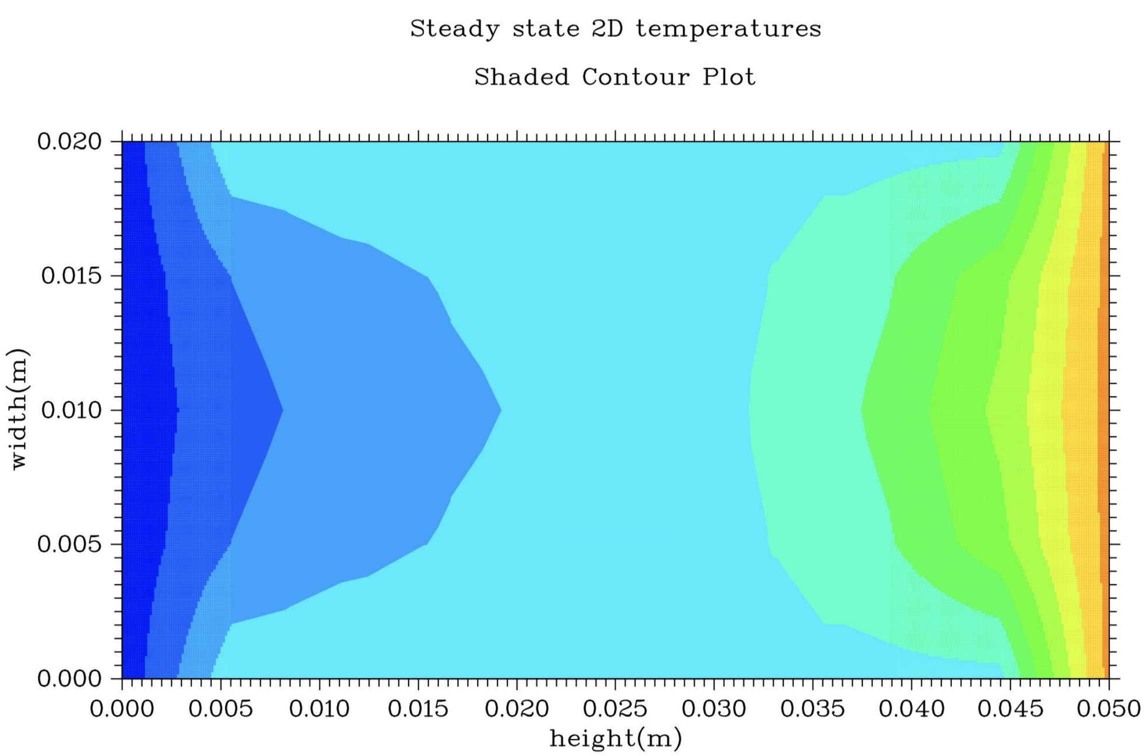

Figure 13 : The shaded contour plot of the 2D temperature of (40x20).

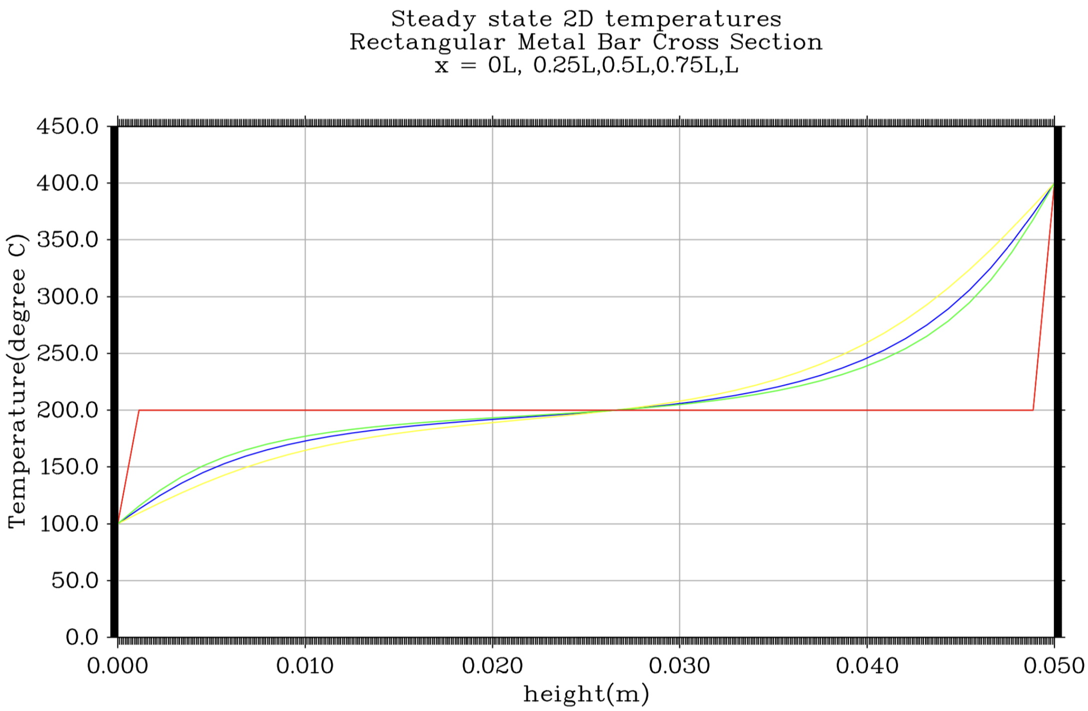

Figure 14: The temperature in the direction of x (40x20).

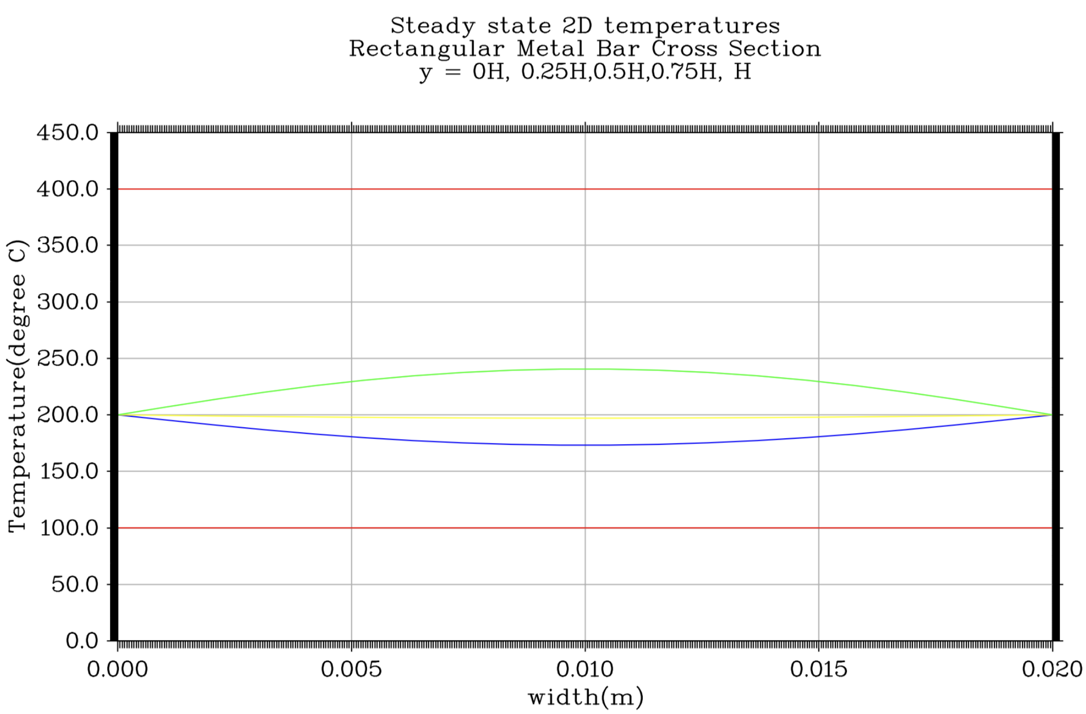

Figure 15: The temperature in the direction of y (40x20).

5. Conclusion.

The project is the approximation and the investigation of the heat conduction of an object. In particular, the long metal bar is placed in a 4 side heated environment, and to determine the temperature and the time it reaches the steady state, the involvement of numerical differentiation computing contributes a huge major in calculating the heat expansion of a surface.

The theoretical part of the project is applying the Taylor's Series to create polynomial calculations of temperature. In real-life, by dealing with a huge database, engineers tend to consider an infinitesimal limit, and it is convenient not to take the limit but to approximate as the Δ values approaching 0. Utilizing the Taylor's series expansion to theorize the temperature equation helps reduce the uncertainty and get the picture of the tendency.

To get a smaller error and more accurate data approximations, the delta values should be small, which means the users should test with reasonably big numbers of points in the xy plane since number of points and Δ values are reciprocal. The number of rows and numbers of columns will be keyboard input by the users.

In terms of programming, the program is object-oriented and user-friendly for corporations and users to apply on a daily basis. The program is coded in C++ language, featuring a variety of user-functions, class to return multiple properties of an object, 1D and 2D dynamic arrays to minimize the memory space, output stream to the file the users wish to save to. The program is also dynamic and adaptable in terms of user's changing.

The program also embraces the dislin library that helps the users to picture the behaviors of the data more comprehensively. The plotting features include the transient center temperature with time, the regular and shaded contour plot of the surface, temperature versus x-axis and y-axis to understand the tendency of heat conduction.

6. List of Figures, Official Program, Oath.

6.1 List of Figures

6.2 Official Program

```
//Thuong Mai
//Engineering 50_Fall2022
#include<iostream>
#include<cmath>
#include<iomanip>
#include<fstream>
#include<string>
#include <cstdio>
#include <windows.h>
#include "c:\dislin\dislin.h"
using namespace std;
class temp
{
	public:
		string file;
		void input(int &m, int &n);
		void function_1(int m, int n, float**T,float *Tcenter,
string &file );
		void function_2(int m, int n,float **T,string file);
		void temp_0(int m, int n, float **T,string file);
```

```
            void time_lot(int m,int n,float *Tcenter,string file);
            void T_contour_plot(int m, int n,float **T);
            void T_contour_shade_plot(int m, int n,float **T);
            void T_x_plot(int m, int n,float **T);
            void T_y_plot(int m, int n,float **T);
            double get_dx();
            double get_dy();
            double get_dt();
      private:
            double dx;
            double dy;
            double dt;
            float *Tcenter;
            float **T;
            float **t_half;

};
int main()
{
      string file;
      ofstream write;
      temp TEMP;
      int m, n, time=1300;
            cout<<"\n Hello, welcome to my 2D transient_Heat
Conduction estimate program!"<<endl;
            cout<<"\n This program will assist you in estimating
the surface temperature of the environment heat
distribution."<<endl;
            cout<<"\n To output this program in your text file,
enter the file name."<<endl;
      cin>>file;
      cout<<"\n Enter the number of rows and columns of the
surface: "<<endl;
      cin>>m>>n;
      write.open(file.c_str());
            cout<<"\n Number of rows is  : "<<m<<endl;
            cout<<"\n Number of colums is: "<<n<<endl;
            write<<"\n Number of rows is  : "<<m<<endl;
            write<<"\n Number of colums is: "<<n<<endl;
      TEMP.input(m,n);
            cout<<"\n The value of delta x is dx =
"<<TEMP.get_dx()<<" m "<<endl;
            cout<<"\n The value of delta y is dy =
"<<TEMP.get_dy()<<" m "<<endl;
```

```
            cout<<"\n The value of delta t is dt = 
"<<TEMP.get_dt()<<" m "<<endl;
            write<<"\n The value of delta x is dx = 
"<<TEMP.get_dx()<<" m "<<endl;
            write<<"\n The value of delta y is dy = 
"<<TEMP.get_dy()<<" m "<<endl;
            write<<"\n The value of delta t is dt = 
"<<TEMP.get_dt()<<" s "<<endl;
      double total_time= time*TEMP.get_dt();
      double deltat=TEMP.get_dt();
            cout<<"\n The estimate time for the suface temperature 
to be steady is: "<<total_time<<" seconds."<<endl;
            write<<"\n The estimate time for the suface temperature 
to be steady is: "<<total_time<<" seconds."<<endl;

      float **t_half= new float *[m];
      for (int i=0;i<m;i++)
            {
                  t_half[i]=new float [n];
            }
      float *Tcenter = new float [time];
      float **T= new float *[m];
            for (int i=0;i<m;i++)
            {
                  T[i]=new float [n];
            }
            cout<<"\n The temperature at time t = 0 is:"<<endl;
            TEMP.temp_0(m,n,T,file);
            cout<<"\n The temperature at half time t = 
"<<total_time/2<<" is: "<<endl;
            TEMP.function_1(m,n,T,Tcenter,file);
            cout<<"\n The temperature at time final t = 
"<<total_time<<" is: "<<endl;
            TEMP.function_2(m,n,T,file);
            cout<<"\n The temperature of the midpoint is: "<<endl;
            TEMP.time_lot(m,n,Tcenter,file);
            cout<<"\n The contour plot is :"<<endl;
            //write<<"\n The contour plot is :"<<endl;
            TEMP.T_contour_plot(m,n,T);
            cout<<"\n The contour shade plot is: "<<endl;
            //write<<"\n The contour shade plot is: "<<endl;
            TEMP.T_contour_shade_plot(m, n, T);
            cout<<"\n The plot for the x- axis:"<<endl;
            //write<<"\n The plot for the x- axis:"<<endl;
```

```
            TEMP.T_x_plot(m,n,T);
            cout<<"\n The plot for the y- axis:"<<endl;
            //write<<"\n The plot for the y- axis:"<<endl;
            TEMP.T_y_plot(m,n,T);
            cout<<"\n Thank you for using my program!"<<endl;
      delete [] T;
      return 0;
}
void  temp::input(int &m, int &n)
            {
                  float width=0.02, height=0.05;
                  dx=width/(n-1);
                  dy=height/(m-1);
                  dt=0.005;
            }
void temp::function_1( int m, int n, float **T, float *Tcenter
,string &file)
{
      ofstream write;
      write.open(file.c_str(),ios::app);
      write<<"\n The temperature at half time t = "<<6.5/2<<" is:
"<<endl;
      float t_half[m][n];

      const double a=1.88E-05 ;
      int time=0;
      for (int i=1;i<m-1;i++)
      {
            for (int j=1; j<n-1;j++)
            {
                T[i][j]=20;
         }
     }
      for (int i=0;i<m;i++)
      {
            T[i][0]=200.000;
            T[i][n-1]=200.000;
      }
      for (int j=0;j<n ;j++)
      {
            T[0][j]=100.000;
            T[m-1][j]=400.000;
      }
while (time<=1300)
```

```
{
      for (int i=1;i<m-1;i++)
      {
            for (int j=1; j<n-1;j++)
            {
            T[i][j]=(( T[i+1][j] - 2*T[i][j] + T[i-1][j]
)/pow(dx,2) +    ( T[i][j+1] - 2*T[i][j] + T[i][j-1]
)/pow(dy,2))*a*dt+T[i][j];
            }
      }
      time++;
      if (time==650)
      {
            for (int i=0;i<m;i++)
            {
                  for (int j=0;j<n;j++)
                  {
                        t_half[i][j]=T[i][j];
                  }
            }
      }
            Tcenter[time]=T[m/2][n/2];
}

for (int i=0;i<m;i++)
            {
                  for (int j=0;j<n;j++)
                  {
                        cout<<setw(20)<<t_half[i][j];
                        write<<setw(20)<<t_half[i][j];
                  }
                  cout<<endl;
                  write<<endl;
            }
}
void temp::function_2 (int m, int n,float **T,string file)  //this
is for the final time
{
      ofstream write;
      write.open(file.c_str(), ios::app);
      write<<"\n The temperature at time final t = "<<1300*dt<<"
is: "<<endl;
      for (int i=0;i<m;i++)
      {
```

```
            for (int j=0;j<n;j++)
            {
                  cout<<setw(20)<<T[i][j];
                  write<<setw(20)<<T[i][j];
            }
            cout<<endl;
            write<<endl;
      }
}
void temp::temp_0(int m, int n, float **T,string file) // this
function is to determine the temperature at the time=0
{
      ofstream write;
      write.open(file.c_str(),ios::app);
      write<<"\n The temperature at time t = 0 is:"<<endl;
            for (int i=1;i<m-1;i++)
      {
            for (int j=1; j<n-1;j++)
            {
                T[i][j]=20;
         }
     }
      for (int i=0;i<m;i++)
      {
            T[i][0]=200.000;
            T[i][n-1]=200.000;
      }
      for (int j=0;j<n ;j++)
      {
            T[0][j]=100.000;
            T[m-1][j]=400.000;
      }
            for (int i=0;i<m;i++)
            {
                   for (int j=0;j<n;j++)
                  {
                  cout<<setw(20)<<T[i][j];
                  write<<setw(20)<<T[i][j];
                  }
                  cout<<endl;
                  write<<endl;
            }
      }
double temp::get_dx()
```

```
{
      return dx;
}
double temp::get_dy()
{
      return dy;
}
double temp::get_dt()
{
      return dt;
}
void temp::time_lot(int m,int n,float *Tcenter,string file)
{
      ofstream write;
      write.open(file.c_str(), ios::app);
      write<<"\n The temperature of the midpoint is: "<<endl;
      float t[1300];
      for (int i = 1 ;i<=1300; i++)
      {
            t[i]=i-1;
            cout<<Tcenter[i]<<endl;
            write<<Tcenter[i]<<endl;
      }
//**************Plot Configuration*****************

  metafl("pdf");//Creates screen output. To create PDF output use
"pdf"
  disini();
  pagera();
  complx();
  axspos(450,1800);
  axslen(2200,1200);
  name("time(s)","x");
  name("Temperature(degree C)","y");
  labdig(3,"x");
  labdig(1,"y");
  ticks(10,"xy");
  titlin("Transient Center Temperture",1);
  titlin("temperature versus time",3);
  graf(0.,1300.,0.,150.,0.,250.,0.,10); // what is the step of the
temperature
  //question, why does this work fine as well
graf(0.,1300.,0.,150.,0.,250.,20.,10);
  setrgb(0.7,0.7,0.7);
```

```
  grid(1,1);
  color("white");

  title();
//*******************Create Plot*******************
  color("red");
  curve(t,Tcenter,1300); //plots cos curve
  color("red");
  disfin();
}
void temp::T_contour_plot(int m, int n,float **T)
{
           float x[n], y[m], z[m][n];
      cout<<"does this function merge in?"<<endl;


for (int i=0;i<m;i++)
{
      for(int j=0;j<n;j++)
      {
       z[i][j]=T[i][j];
      }
}

//***************Plot Configuration*****************
  metafl("pdf");//Creates screen output. To create PDF output use
"pdf"
  disini();
  pagera();
  complx();
  axspos(450,1800);
  axslen(2200,1200);
  name("height(m)","x");
  name("width(m)","y");
  labdig(3,"x");
  labdig(3,"y");
  ticks(10,"xy");
  titlin("Steady state 2D temperatures",1);
  titlin(" Contour Plot",3);
   graf(0.,0.05,0.,0.005,0.,0.02,0.,0.005); // what is the step of
the temperature
  setrgb(0.7,0.7,0.7);
  grid(1,1);
  color("white");
```

```
  title();
//*******************Create Plot*******************
           for (int i=0;i<n;i++)
           {
                x[i]=i*dx;
           }
           for (int j=0;j<m;j++)
           {
                y[j]=j*dy;
           }
      float zlevel=100;

      for (int g=0;g<9;g++)
      {
           zlevel=zlevel+30;
           labels("float","contur");
           color("red");
           contur(y,m,x,n,(const float *)z,zlevel);
      }
  disfin();
}
```

```
void temp::T_contour_shade_plot(int m, int n,float **T)
{
      float x[n], y[m], z[m][n];

for (int i=0;i<m;i++)
{
      for(int j=0;j<n;j++)
      {
       z[i][j]=T[i][j];
      }
}

//**************Plot Configuration*****************
  metafl("pdf");//Creates screen output. To create PDF output use
"pdf"
  disini();
  pagera();
  complx();
  axspos(450,1800);
  axslen(2200,1200);
  name("height(m)","x");
  name("width(m)","y");
  labdig(3,"x");
```

```
  labdig(3,"y");
  ticks(10,"xy");
  titlin("Steady state 2D temperatures",1);
  titlin("Shaded Contour Plot",3);
   graf(0.,0.05,0.,0.005,0.,0.02,0.,0.005); // what is the step of
the temperature
  setrgb(0.7,0.7,0.7);
  grid(1,1);
  color("white");
  title();
//*******************Create Plot*******************
           for (int i=0;i<n;i++)
           {
                 x[i]=i*dx;
           }
           for (int j=0;j<m;j++)
           {
                 y[j]=j*dy;
           }
           int num=m+n;
      float zlevel[num];

      zlevel[0]={100};

      for (int g=0;g<num;g++)
      {
           zlevel[g]=zlevel[g-1]+30;
           labels("float","contur");
           color("yellow");
           conshd(y,m,x,n,(const float *)z,zlevel,num);
      }
  disfin();
}
void temp::T_x_plot(int m, int n,float **T)
{
      cout<<"This is the function for the x-plot"<<endl;
      float x0[1000],x1[1000],x2[1000],x3[1000],x4[1000];
      int x0_25=0.25*m,x0_5 = 0.5*m,x0_75=0.75*m,x1_0=m-1;
      //that means that it does need some calculations here
      float l[n];
                 for (int i=0;i<n;i++)
                 {
                       l[i]=dx*i;
                 }
```

```
                for (int j=0;j<n;j++)
                {
                     x0[j]=T[0][j];
                     x1[j]=T[x0_25][j];
                     x2[j]=T[x0_5][j];
                     x3[j]=T[x0_75][j];
                     x4[j]=T[x1_0][j];
                }
//***************Plot Configuration*****************
  metafl("pdf");//Creates screen output. To create PDF output use
"pdf"
  disini();
  pagera();
  complx();
  axspos(450,1800);
  axslen(2200,1200);
  name("width(m)","x");
  name("Temperature(degree C)","y");
  labdig(3,"x");
  labdig(1,"y");
  ticks(100,"xy");
  titlin("Steady state 2D temperatures",1);
  titlin("Rectangular Metal Bar Cross Section",2);
  titlin("y = 0H, 0.25H,0.5H,0.75H, H",3);
  graf(0.,0.02,0.,0.005,0.,450.,0.,50.);
  setrgb(0.7,0.7,0.7);
  grid(1,1);
  color("white");
  title();
//*******************Create Plot*******************
      color("red");
      curve(l,x0,n);
      color("blue");
      curve(l,x1,n);
      color("yellow");
      curve(l,x2,n);
      color("green");
      curve(l,x3,n);
      color("red");
      curve(l,x4,n);
      disfin();
}

void temp::T_y_plot(int m, int n,float **T)
```

```
{
cout<<"This is the function for the y-plot"<<endl;
      float y0[1000],y1[1000],y2[1000],y3[1000],y4[1000];
      int y0_25=0.25*n,y0_5 = 0.5*n,y0_75=0.75*n,y1_0=n-1;
      float l[m]; //only works for m=20, n=10
                  for (int j=0;j<m;j++)
                  {
                        l[j]=dy*j;
                  }
                  for (int i=0;i<m;i++)
                  {
                        y0[i]=T[i][0];
                        y1[i]=T[i][y0_25];
                        y2[i]=T[i][y0_5];
                        y3[i]=T[i][y0_75];
                        y4[i]=T[i][y1_0];
                  }
//***************Plot Configuration*****************
  metafl("pdf");//Creates screen output. To create PDF output use
"pdf"
  disini();
  pagera();
  complx();
  axspos(450,1800);
  axslen(2200,1200);
  name("height(m)","x");
  name("Temperature(degree C)","y");
  labdig(3,"x");
  labdig(1,"y");
  ticks(100,"xy");
  titlin("Steady state 2D temperatures",1);
  titlin("Rectangular Metal Bar Cross Section",2);
  titlin("x = 0L, 0.25L,0.5L,0.75L,L",3);
  graf(0.,0.05,0.,0.01,0.,450.,0.,50.);
  setrgb(0.7,0.7,0.7);
  grid(1,1);
  color("white");
  title();
//*******************Create Plot*******************
      color("red");
      curve(l,y0,m); //plots cos curve
      color("blue");
      curve(l,y1,m);
      color("yellow");
```

```
    curve(l,y2,m);
    color("green");
    curve(l,y3,m);
    color("red");
    curve(l,y4,m);
    disfin();
}
```

Oath

"This project has been completed in its entirety, including the program design/development and the written report, by me. I have not shared any portion of my program with anyone. I know that plagiarism of any kind may result in a grade of ' F ' in the course."

Name: Thuong Mai

[redacted]

Date: 12/14/2022