

CFD Assignment-1

Name: -Swapnendu Chakraborty

Sec: - BME UG3 A2

Roll No: - 002111201176

Topic: - 2D Steady State Heat Generation Problem

Problem: -

Assignment No-1

| NXM | Tau |
|---------|-----|
| 50x50 | * |
| 100x100 | * |
| 200x200 | * |

$T_{\text{avg}}|_{x=0.5L} = \int_0^1 T(y, x) dy$

(7)

Steady state temperature contours for.

$(T_H - T_L) = \Delta T$

(a) $\Delta T = 2$

(b) $\Delta T = -2$

(c) $\Delta T = 100$

(d) $\Delta T = 0.01$

}

(Tecplot)

Formulate the problem.

- (1) G.E. $= \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{q''_g}{k} = 0$
- (2) B.C.
- (3) Dimensionless form of G.E. & B.C.
- (4) FDM - discretization.
- (5) Code development ~ MATLAB/C/C++/Python/Fortran
- (6) Mesh independence test.

Solution:-

Assignment - 1

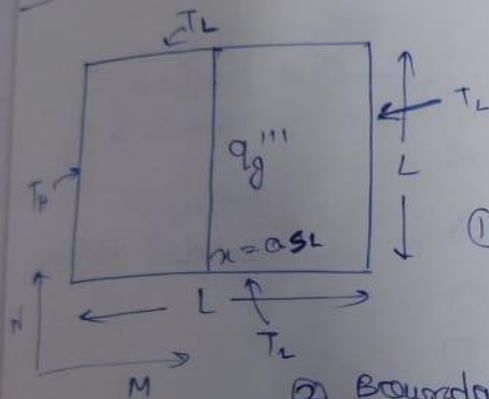
(CFD).

Name - Supriyendu Chakraborty

Sec - A2

Roll no:- 00211/201176.

Language used:- C++.



Answers

| N x M | Tau |
|-----------|-----|
| 50 x 50 | * |
| 100 x 100 | |
| 200 x 200 | |

① Governing Equations:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{q'''_y}{k} = 0.$$

② Boundary Conditions:-

BC1 At $x=0$, $T=T_H$

BC2 At $x=L$, $T=T_L$.

BC3. At $y=0$, $T=T_L$.

BC4 At $y=L$, $T=T_L$.

③ Dimensionless forms:-

$$\Theta = \frac{T - T_L}{q'''_y L^2 / k}, \quad X = \frac{x}{L}, \quad Y = \frac{y}{L}.$$

$$\Rightarrow \frac{\partial^2 \Theta}{\partial X^2} + \frac{\partial^2 \Theta}{\partial Y^2} + 1 = 0.$$

BC1 At $X=0$, $\Theta = \frac{T_H - T_L}{q'''_y L^2 / k} = \alpha$

BC3. At $Y=0$, $\Theta=0$

BC2 At $X=1$, $\Theta=0$

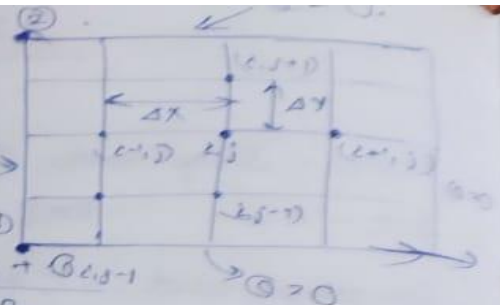
BC4 At $Y=1$, $\Theta=0$.

④ FDM Discretization:-

using Central Difference Method

$$\frac{\partial^2 \theta}{\partial x^2} + \frac{\partial^2 \theta}{\partial y^2} + 1 = 0$$

$$\Rightarrow \frac{\theta_{2M,j} - 2\theta_{2,j} + \theta_{2-1,j}}{(\Delta x)^2} + \frac{\theta_{2,j+1} - 2\theta_{2,j} + \theta_{2,j-1}}{(\Delta y)^2} + 1 = 0.$$



For uniform grid, $\Delta x = \Delta y$.

$$\Rightarrow \boxed{-\theta_{2-1,j} - \theta_{2,j-1} + 4\theta_{2,j} - \theta_{2,j+1} - \theta_{2+1,j} = (\Delta x)^2}$$

For $i=1, j=1 \rightarrow N \rightarrow \theta_{2,j} = 2$.

For $i=M, j=1 \rightarrow N \rightarrow \theta_{2,j} = 0$.

For $i=1 \rightarrow M, j=1 \rightarrow \theta_{2,j} = 0$.

For $i=1 \rightarrow M, j=N \rightarrow \theta_{2,j} = 0$.

Cross-
check Method

For handling the corner points

$$\theta_{1,1} = \frac{2}{2}, \quad \theta_{1,N} = \frac{2}{2}$$

For initial guess, $\theta_{2,j} = 0$ where $i=1 \rightarrow M$
 $j=1 \rightarrow N$.

Calculation

$$\theta_{2,j} = \frac{1}{4} [(\Delta x)^2 + \theta_{2-1,j} + \theta_{2,j-1} + \theta_{2,j+1} + \theta_{2+1,j}]$$

where $i=2 \rightarrow M-1$

$j=2 \rightarrow N-1$

To recalculate T from θ

$$\boxed{T = T_L + \frac{q'' L^2 \theta}{K}}$$

For Calculation

$$q'' = 1000 \text{ W/m}^2$$

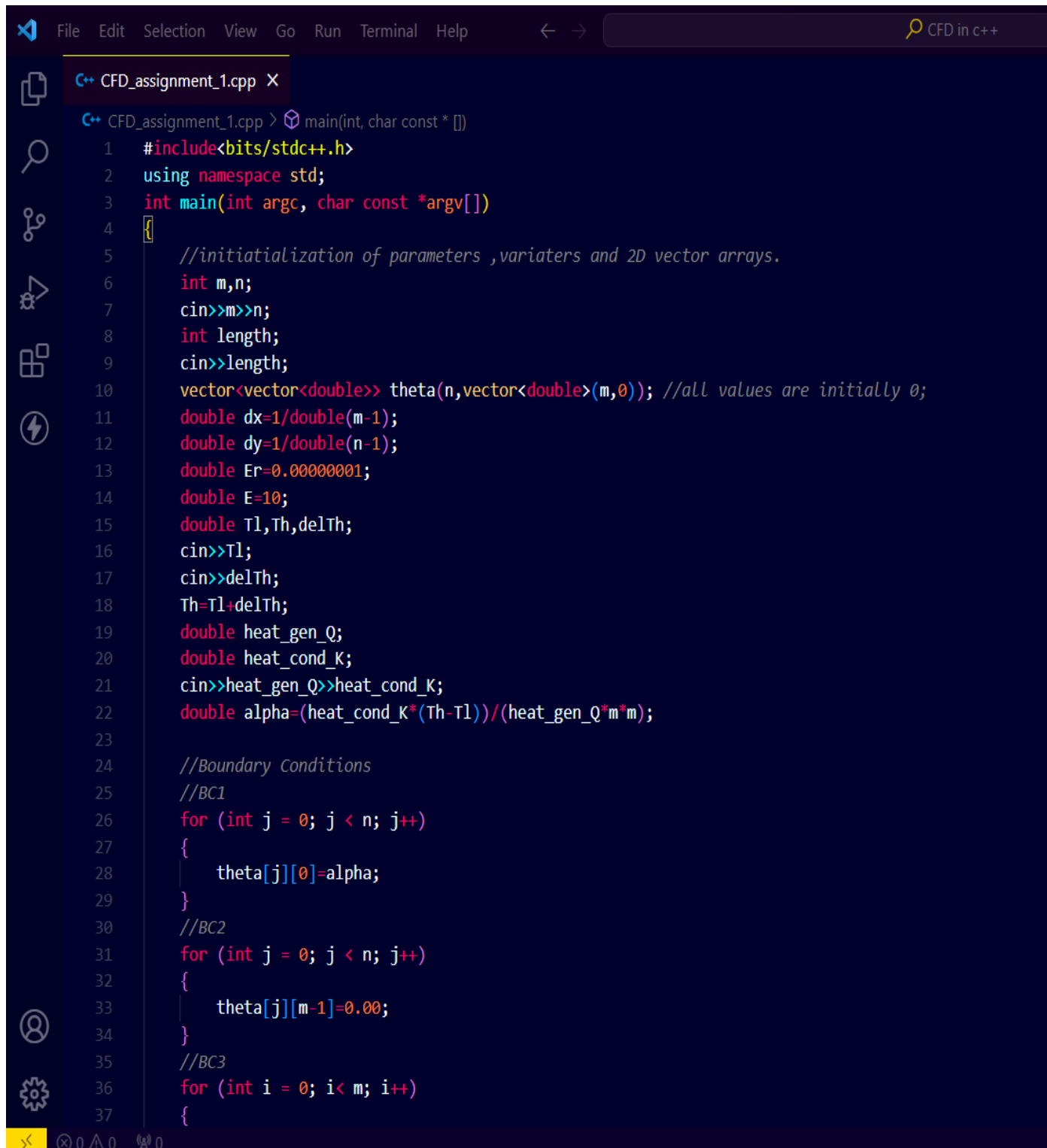
$$K = 100 \text{ W/(mK)}$$

$$T_L = 10^\circ\text{C}$$

$$\Delta T = 2, -2, 100, 001$$

$$L = 2 \text{ m}$$

Formulation and Coding in C++:-



The image shows a screenshot of a C++ code editor with a dark theme. The editor window is titled "CFD_assignment_1.cpp". The code is a C++ program for a CFD assignment, starting with a main function. It includes the `<bits/stdc++.h>` header and uses the `std` namespace. The `main` function takes `argc` and `argv` as arguments. The code initializes variables `m`, `n`, `length`, `dx`, `dy`, `Er`, `E`, `Tl`, `Th`, `delTh`, `heat_gen_Q`, `heat_cond_K`, and `alpha`. It then initializes a 2D vector array `theta` of type `vector<vector<double>>`. The code includes comments for boundary conditions and uses `for` loops to initialize the boundary values of `theta`.

```
CFD_assignment_1.cpp > main(int, char const * [])
1  #include<bits/stdc++.h>
2  using namespace std;
3  int main(int argc, char const *argv[])
4  {
5      //initiatialization of parameters ,variaters and 2D vector arrays.
6      int m,n;
7      cin>>m>>n;
8      int length;
9      cin>>length;
10     vector<vector<double>> theta(n,vector<double>(m,0)); //all values are initially 0;
11     double dx=1/double(m-1);
12     double dy=1/double(n-1);
13     double Er=0.00000001;
14     double E=10;
15     double Tl,Th,delTh;
16     cin>>Tl;
17     cin>>delTh;
18     Th=Tl+delTh;
19     double heat_gen_Q;
20     double heat_cond_K;
21     cin>>heat_gen_Q>>heat_cond_K;
22     double alpha=(heat_cond_K*(Th-Tl))/(heat_gen_Q*m*m);
23
24     //Boundary Conditions
25     //BC1
26     for (int j = 0; j < n; j++)
27     {
28         theta[j][0]=alpha;
29     }
30     //BC2
31     for (int j = 0; j < n; j++)
32     {
33         theta[j][m-1]=0.00;
34     }
35     //BC3
36     for (int i = 0; i < m; i++)
37     {
```

```

34     }
35     //BC3
36     for (int i = 0; i < m; i++)
37     {
38         theta[0][i]=0.00;
39     }
40     //BC4
41     for (int i = 0; i < m; i++)
42     {
43         theta[n-1][i]=0.00;
44     }
45
46     //handling the corner points;
47     theta[0][0]=alpha/2;
48     theta[n-1][0]=alpha/2;
49
50     //calculation part
51     vector<vector<double>>Theta_new(n,vector<double>(m,0));
52     Theta_new=theta;
53     while(E>Er){
54         // cout<<"hello"<<endl;
55         for(int i=1;i<m-1;i++){
56             for(int j=1;j<n-1;j++){
57                 Theta_new[i][j]=0.25*((dx*dx)+Theta_new[i-1][j]+Theta_new[i][j-1]+Theta_new[i][j+1]+Theta_new[i+1][j]);
58             }
59         }
60         //error calculation
61         double sum=0;
62         for(int i=0;i<m;i++){
63             for(int j=0;j<n;j++){
64                 sum += pow((Theta_new[i][j] - theta[i][j]), 2);
65             }
66         }
67         E=sqrt(sum/((n-1)*(m-1)));
68
69         //final=initial for next iteration
70         theta=Theta_new;

```



C++ CFD_assignment_1.cpp X



C++ CFD_assignment_1.cpp > main(int, char const * [])

```
54 // cout<< "netto <<endl;
55 for(int i=1;i<m-1;i++){
56     for(int j=1;j<n-1;j++){
57         Theta_new[i][j]=0.25*((dx*dx)+Theta_new[i-1][j]+Theta_new[i][j-1]+Theta_new[i][j+1]+Theta_new[i+1][j]);
58     }
59 }
60 //error calculation
61 double sum=0;
62 for(int i=0;i<m;i++){
63     for(int j=0;j<n;j++){
64         sum += pow((Theta_new[i][j] - theta[i][j]), 2);
65     }
66 }
67 E=sqrt(sum/((n-1)+(m-1)));
68
69 //final=initial for next iteration
70 theta=Theta_new;
71 }
72
73 for(int i=0;i<m;i++){
74     for(int j=0;j<n;j++){
75         cout<<i<<"-->"<<j<<"-->"<<theta[i][j]<<endl;
76     }
77 }
78
79 //calculation for Temp value at x=0.5L
80 double Tval;
81 double sum=0.00;
82 for (int j = 0; j < n; j++)
83 {
84     double temp=Tl+((heat_gen_Q*length*length*theta[m/2][j])/heat_cond_K);
85     sum+=temp*dy;
86 }
87 Tval=sum;
88 cout<<"The average Temperature at value X=0.5L is "<<Tval<<" degree";
89 return 0;
90 }
```

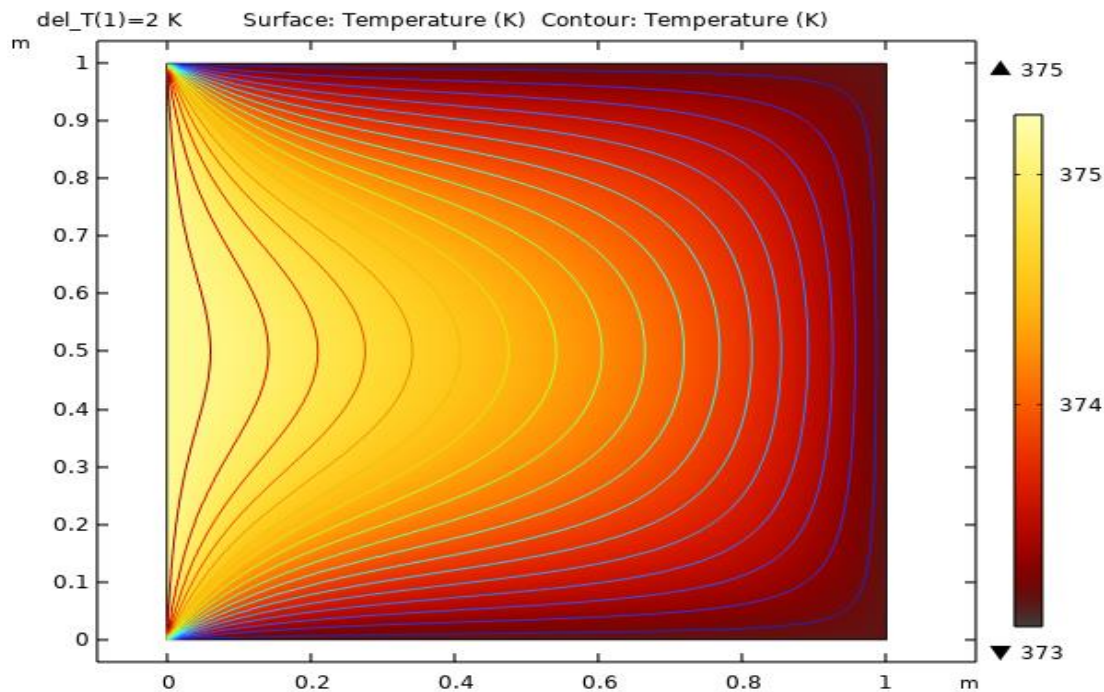


0 0 0

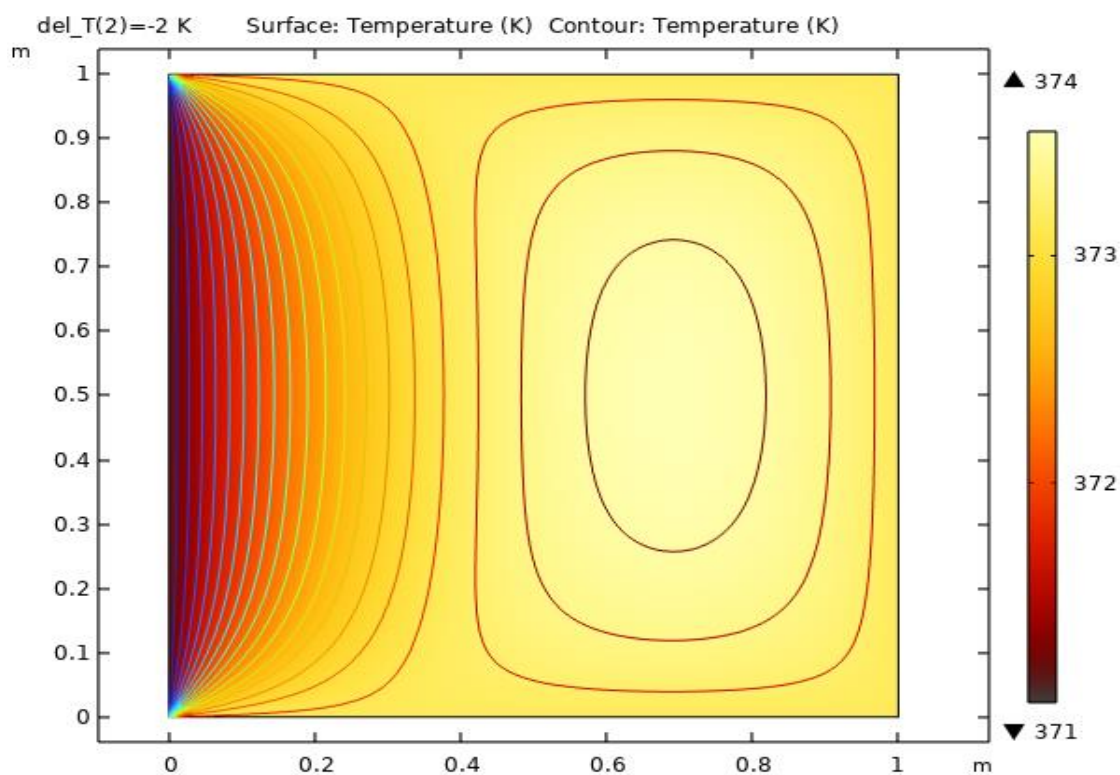
Ln 83,

Contour Plots using TechPlot 360: -

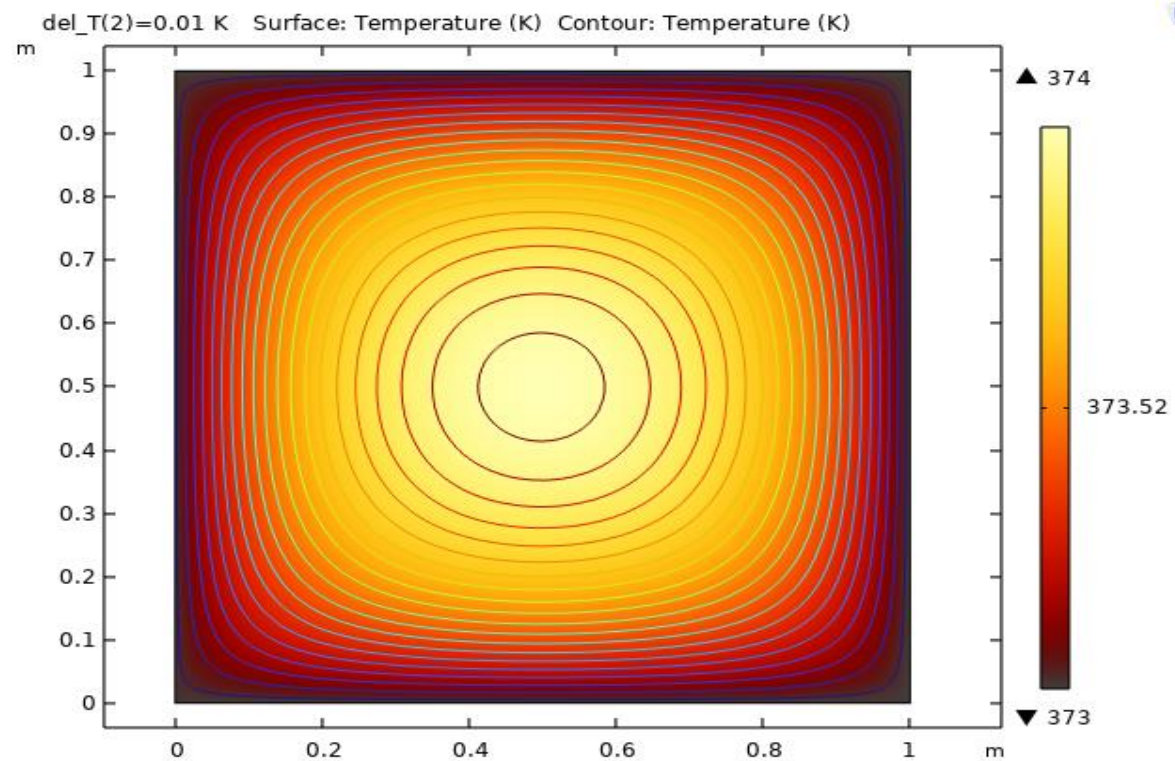
For $\Delta T = 2$ deg



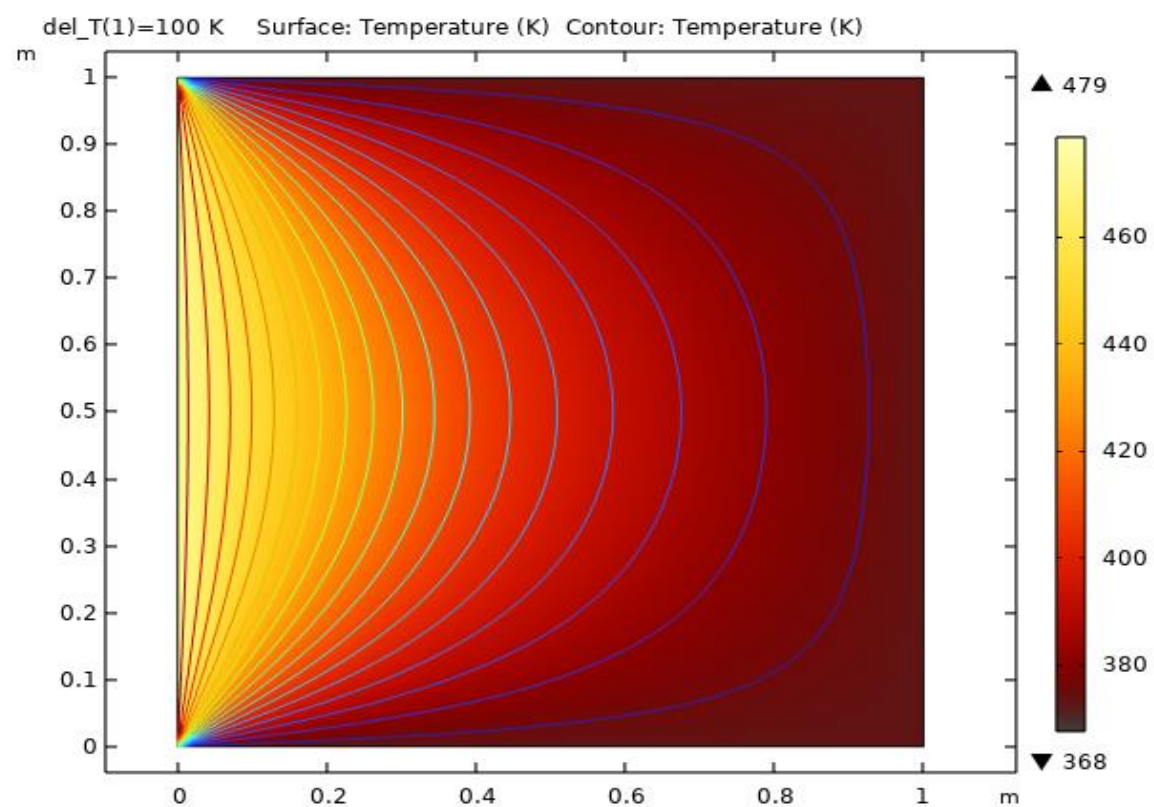
For $\Delta T = -2$ deg



For $\Delta T = 0.01$ deg

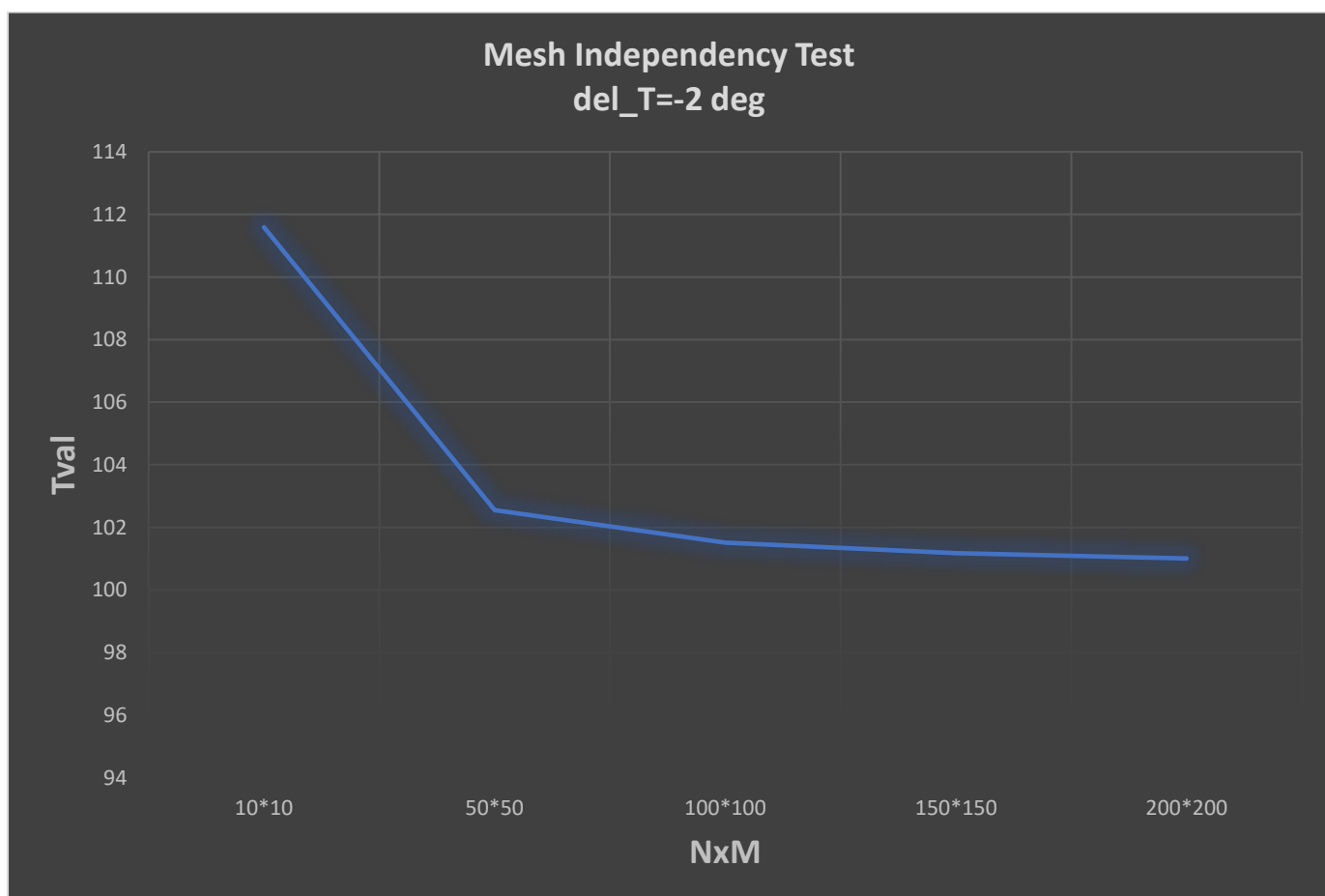
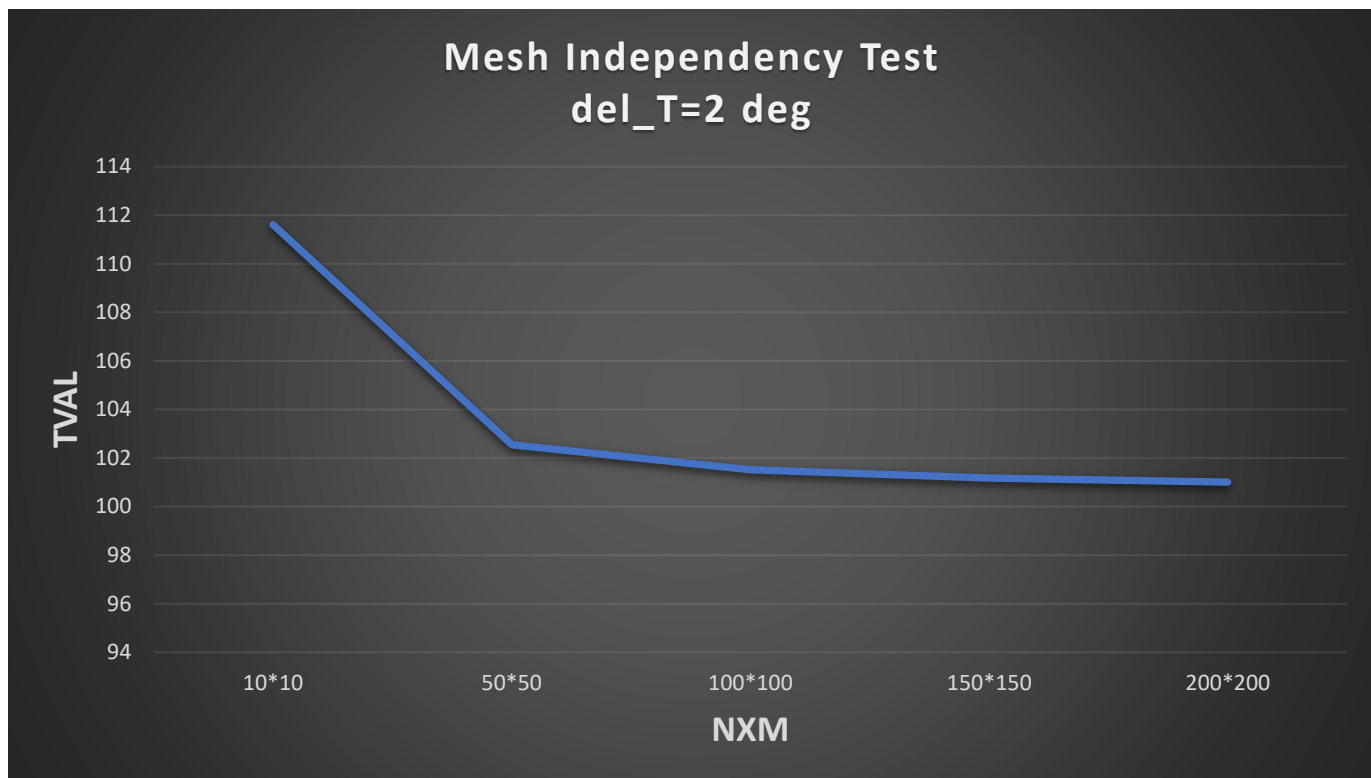


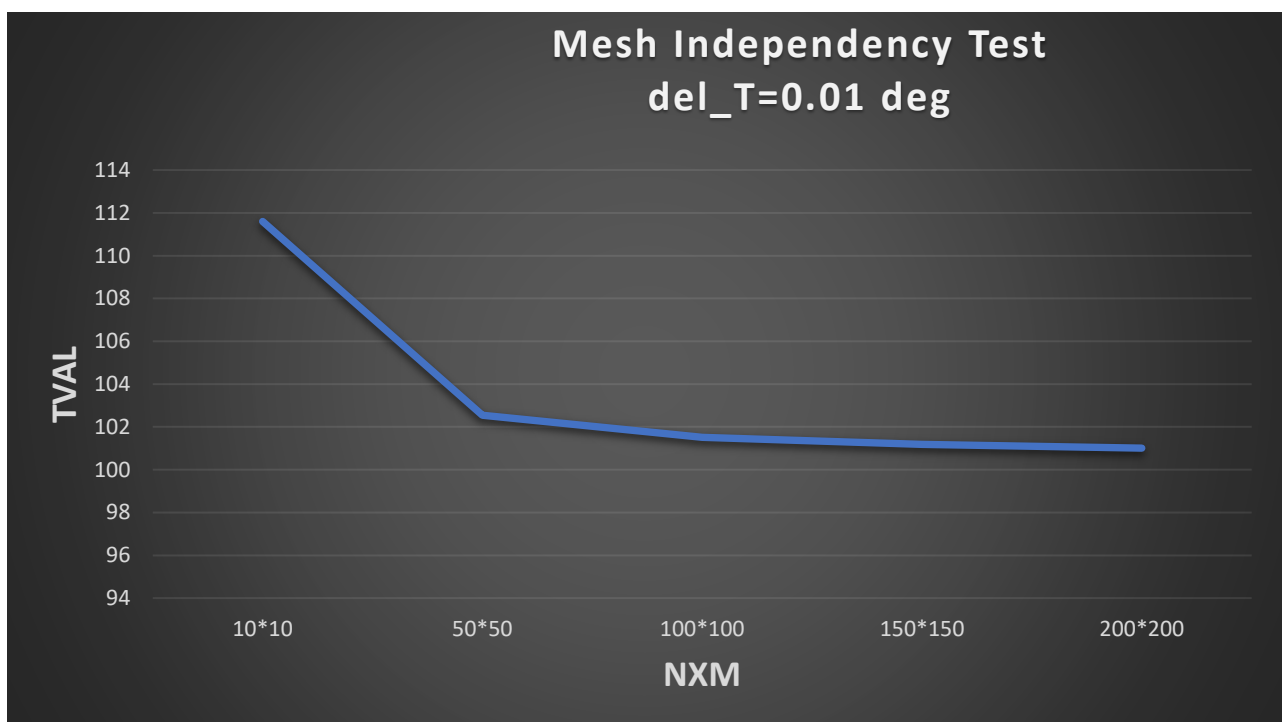
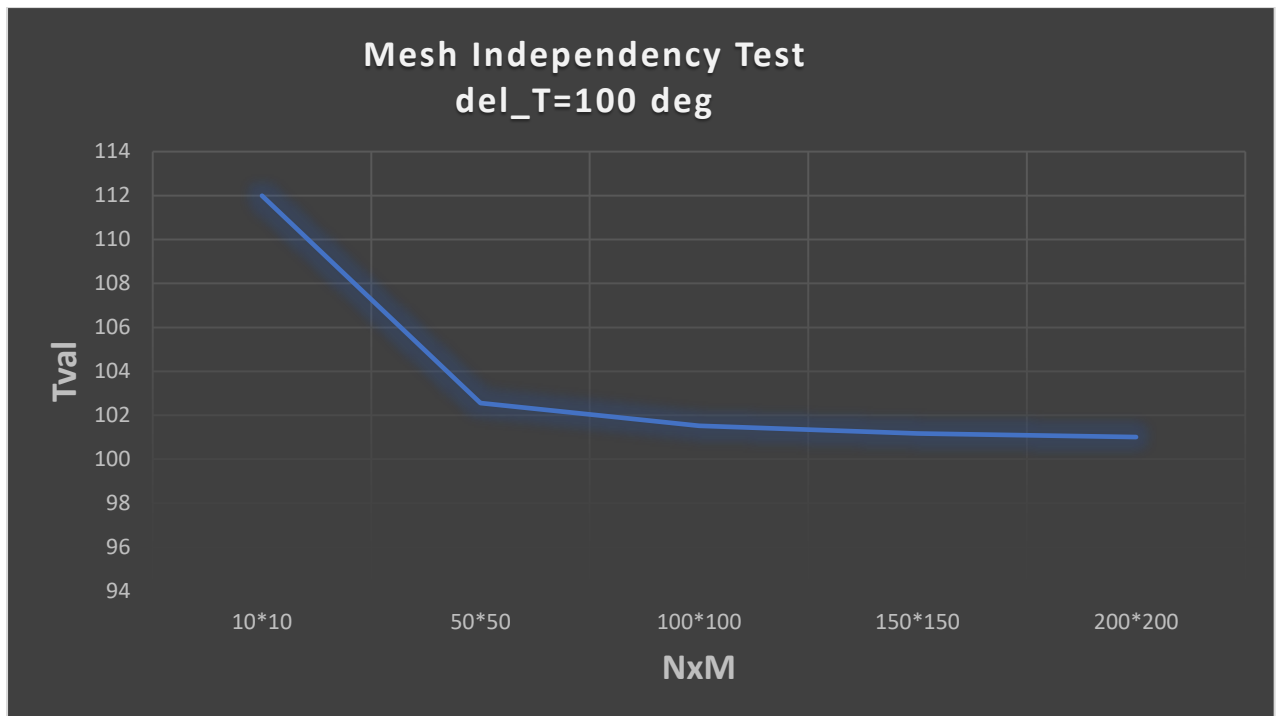
For $\Delta T = 100$ deg



Mesh Independency Test: -

| Mesh Independency Test: | |
|---------------------------|------------------------|
| $N \times M$ | $T_{on} \cdot x=0.5L$ |
| $\Delta T = 2^\circ C$ | 111.608 deg |
| | 102.546 deg |
| | 101.516 deg |
| | 101.177 deg |
| | 101.008 deg |
| $\Delta T = -2^\circ C$ | 111.592 deg |
| | 102.546 deg |
| | 101.516 deg |
| | 101.177 deg |
| | 101.008 deg |
| $\Delta T = 100^\circ C$ | 111.992 deg |
| | 102.56 deg |
| | 101.519 deg |
| | 101.178 deg |
| | 101.009 deg |
| $\Delta T = 0.01^\circ C$ | 111.6 deg |
| | 102.546 deg |
| | 101.516 deg |
| | 101.177 deg |
| | 101.008 deg |





Thus, we can conclude that on finding the Temperature at $x=0.5L$ is more or less constant on taking a mesh above 100×100 .