CFD Assignment-1

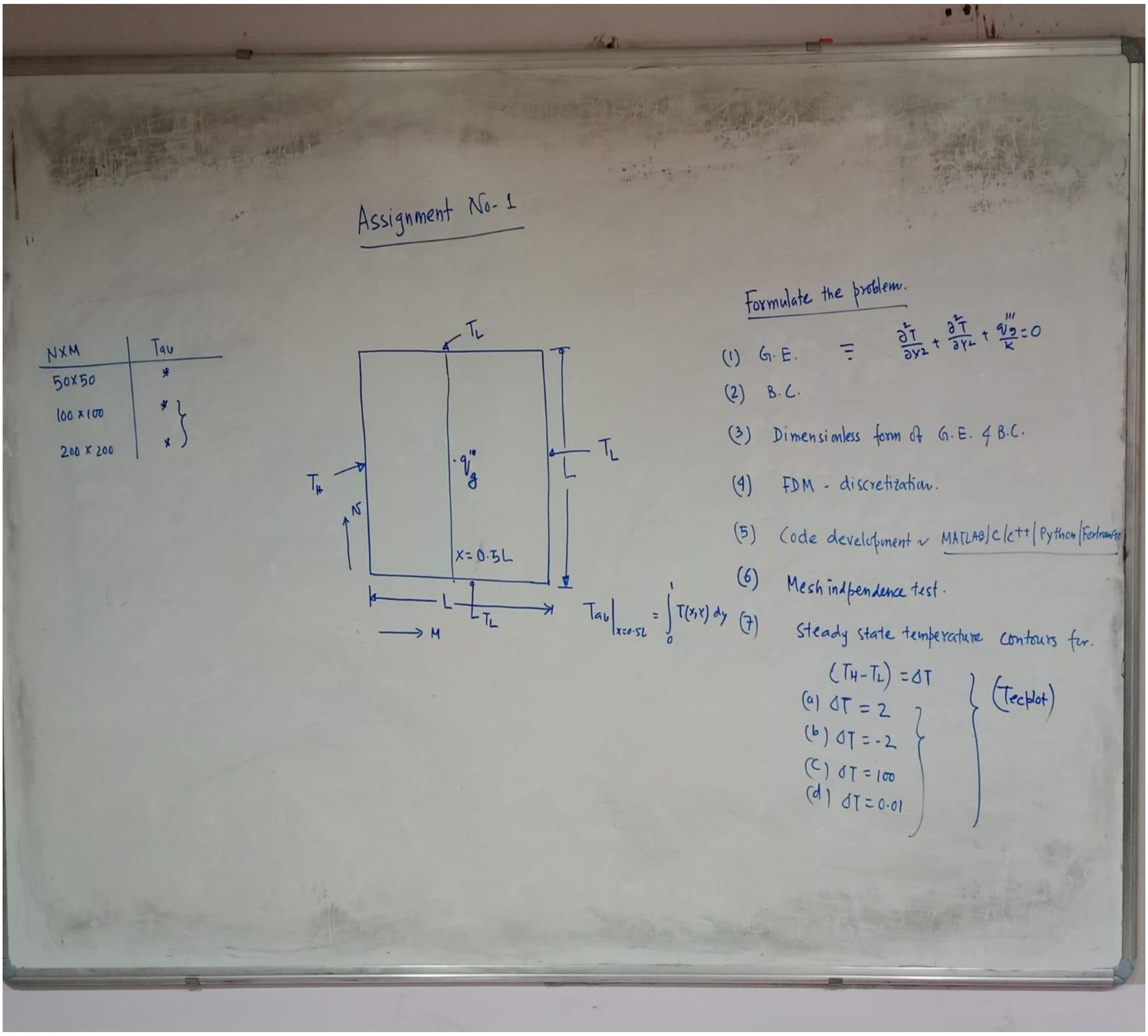
Name: -Swapnendu Chakraborty

Sec: - BME UG3 A2

Roll No: - 002111201176

Topic: - 2D Steady State Heat Generation Problem

Problem: -



Solution:-

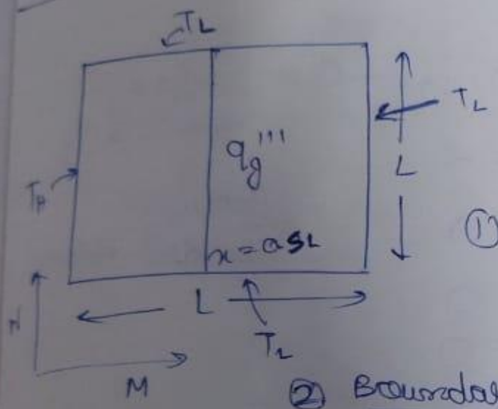
Assignment - 1

(CFD).

Name - Swapnendu Chakraborty

Sec - A2 Roll no:- 002111201176.

Language used :- C++.



Answers

$N \times M$	T_{av}
$50 * 50$	*
100×100	
$200 * 200$	

① Governing Equations:

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

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:-

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

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

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

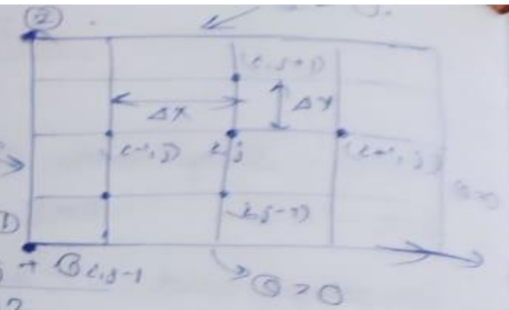
BC2 At $X = 1$, $\theta = 0$

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

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

4) FDM Discretization :-

Using Central Difference Method



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

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

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

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

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

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

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

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

Gauss-Seidal Method

For Handling the corner points

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

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

Calculation

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

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

$q'' = 1000\ W/m^3$

$K = 100\ W/(mK)$

$T_L = 100°C$

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

$L = 1\ m.$

Formulation and Coding in C++:-

```cpp
#include<bits/stdc++.h>
using namespace std;
int main(int argc, char const *argv[])
{
    //initiatialization of parameters ,variaters and 2D vector arrays.
    int m,n;
    cin>>m>>n;
    int length;
    cin>>length;
    vector<vector<double>> theta(n,vector<double>(m,0)); //all values are initially 0;
    double dx=1/double(m-1);
    double dy=1/double(n-1);
    double Er=0.00000001;
    double E=10;
    double Tl,Th,delTh;
    cin>>Tl;
    cin>>delTh;
    Th=Tl+delTh;
    double heat_gen_Q;
    double heat_cond_K;
    cin>>heat_gen_Q>>heat_cond_K;
    double alpha=(heat_cond_K*(Th-Tl))/(heat_gen_Q*m*m);

    //Boundary Conditions
    //BC1
    for (int j = 0; j < n; j++)
    {
        theta[j][0]=alpha;
    }
    //BC2
    for (int j = 0; j < n; j++)
    {
        theta[j][m-1]=0.00;
    }
    //BC3
    for (int i = 0; i< m; i++)
    {
```

CFD_assignment_1.cpp X

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

```cpp
    }
    //BC3
    for (int i = 0; i< m; i++)
    {
        theta[0][i]=0.00;
    }
    //BC4
    for (int i = 0; i< m; i++)
    {
        theta[n-1][i]=0.00;
    }

    //handling the corner points;
    theta[0][0]=alpha/2;
    theta[n-1][0]=alpha/2;

    //calculation part
    vector<vector<double>>Theta_new(n,vector<double>(m,0));
    Theta_new=theta;
    while(E>Er){
        // cout<<"hello"<<endl;
        for(int i=1;i<m-1;i++){
            for(int j=1;j<n-1;j++){
                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]);
            }
        }
        //error calculation
        double sum=0;
        for(int i=0;i<m;i++){
            for(int j=0;j<n;j++){
                sum += pow((Theta_new[i][j] - theta[i][j]), 2);
            }
        }
        E=sqrt(sum/((n-1)+(m-1)));

        //final=initial for next iteration
        theta=Theta_new;
```

⊗ 0 ⚠ 0 0 Ln 75, Col

CFD_assignment_1.cpp

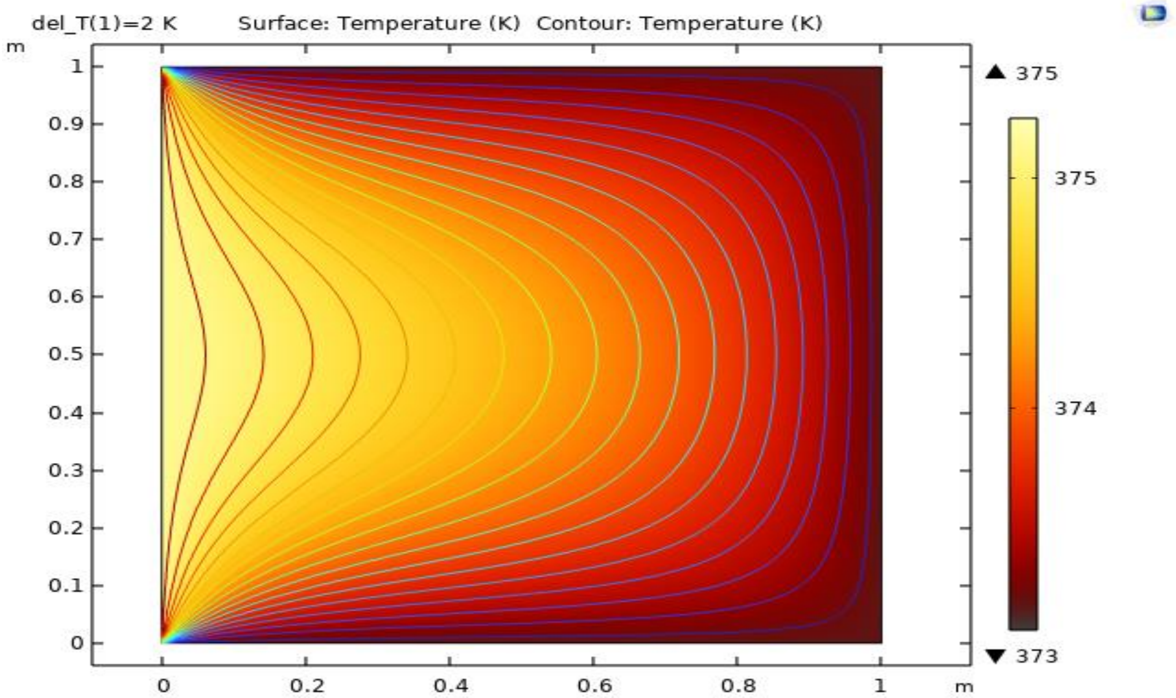
CFD_assignment_1.cpp > main(int, char const * [])

```cpp
        // cout<< hello <<endl;
        for(int i=1;i<m-1;i++){
            for(int j=1;j<n-1;j++){
                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]);
            }
        }
        //error calculation
        double sum=0;
        for(int i=0;i<m;i++){
            for(int j=0;j<n;j++){
                sum += pow((Theta_new[i][j] - theta[i][j]), 2);
            }
        }
        E=sqrt(sum/((n-1)+(m-1)));

        //final=initial for next iteration
        theta=Theta_new;
    }

    for(int i=0;i<m;i++){
        for(int j=0;j<n;j++){
            cout<<i<<"-->"<<j<<"-->"<<theta[i][j]<<endl;
        }
    }

    //calculation for Temp value at x=0.5L
    double Tval;
    double sum=0.00;
    for (int j = 0; j < n; j++)
    {
       double temp=Tl+((heat_gen_Q*length*length*theta[m/2][j])/heat_cond_K);
        sum+=temp*dy;
    }
    Tval=sum;
    cout<<"The average Temperature at value X=0.5L is "<<Tval<<" degree";
    return 0;
}
```

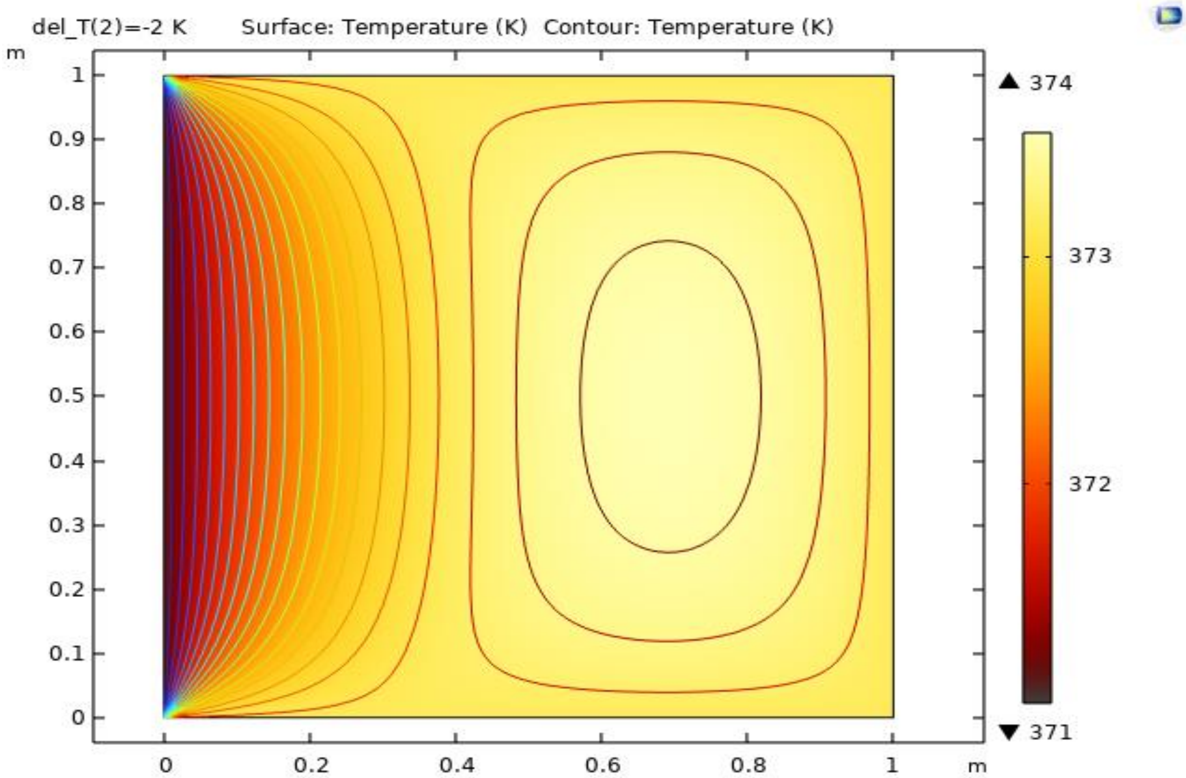
Ln 83,

Contour Plots using TechPlot 360: -

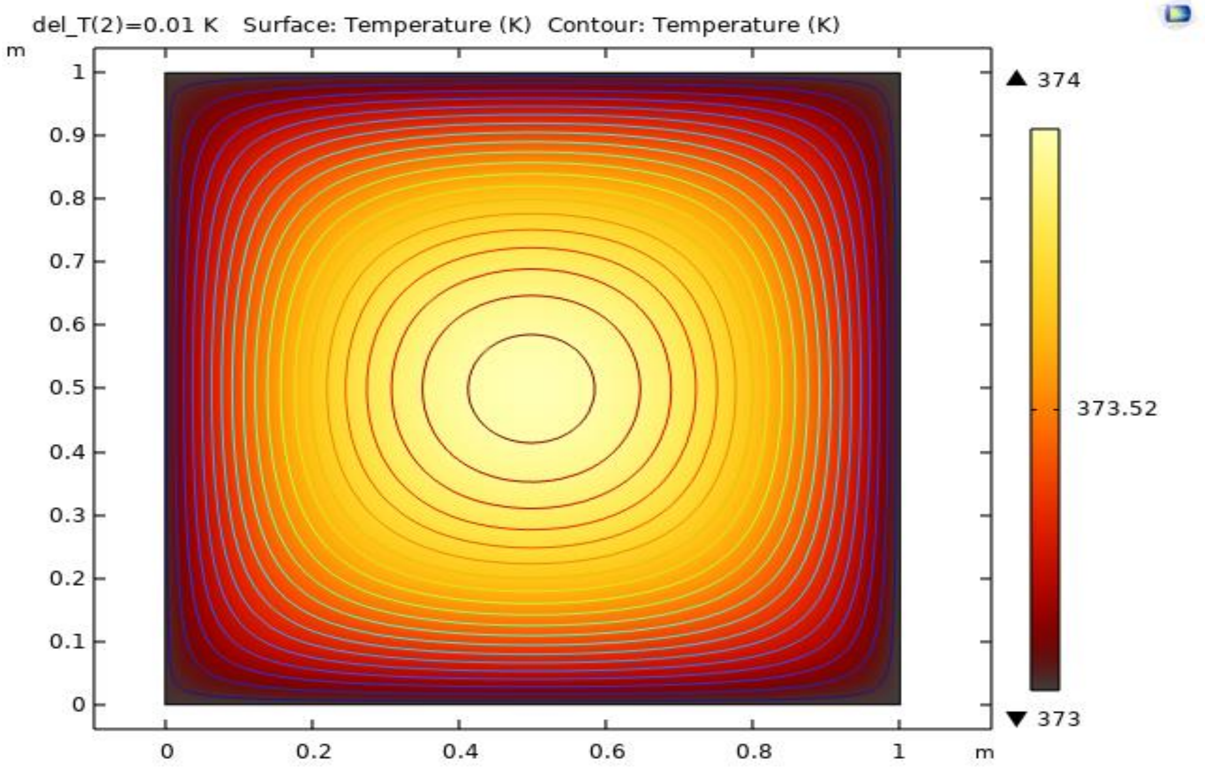
For del_T=2 deg



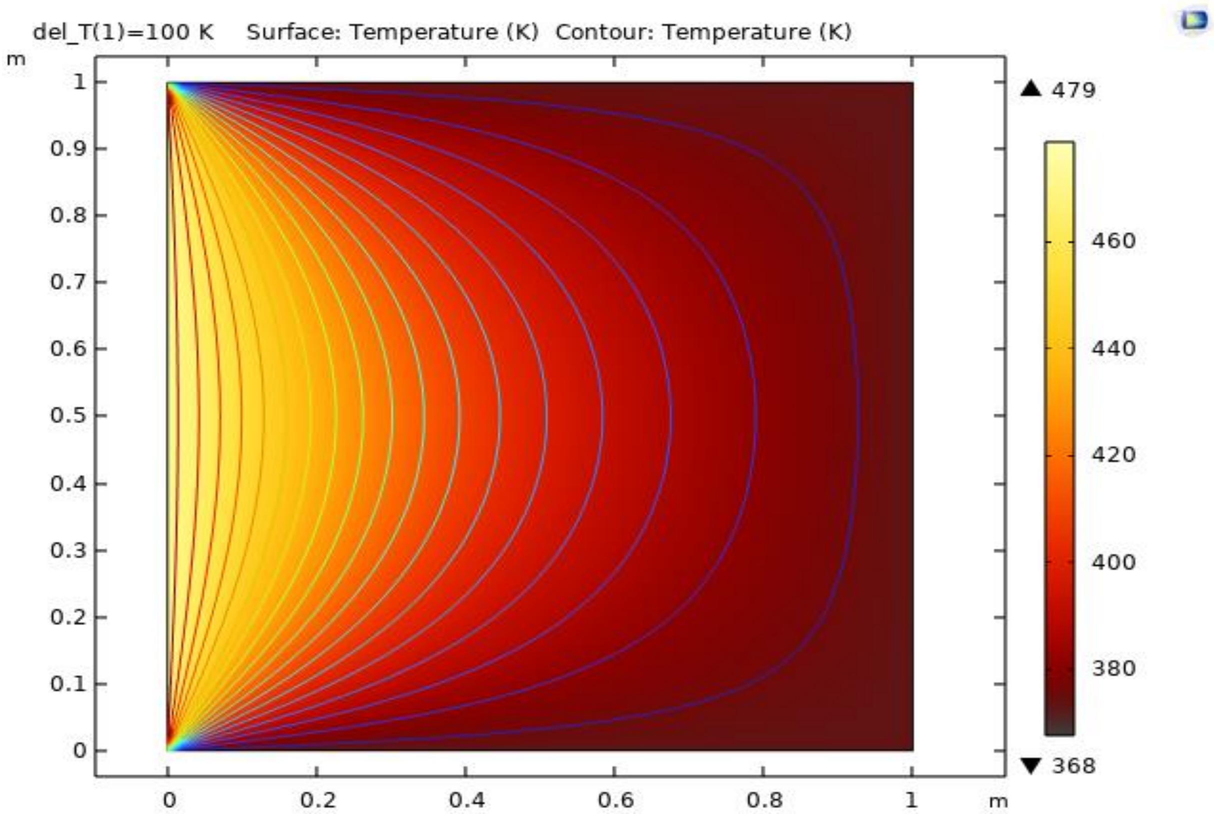
For del_T=-2 deg



For del_T=0.01 deg



For del_T=100 deg

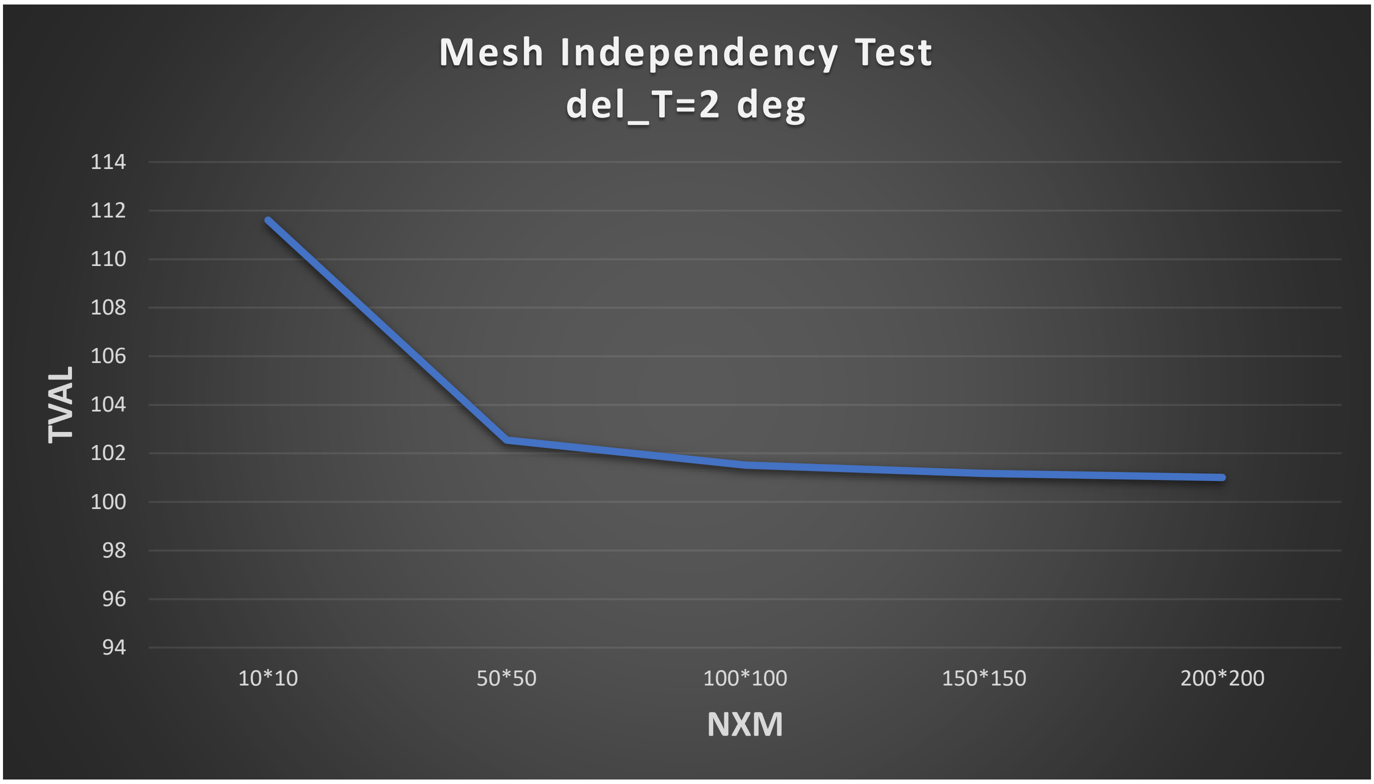


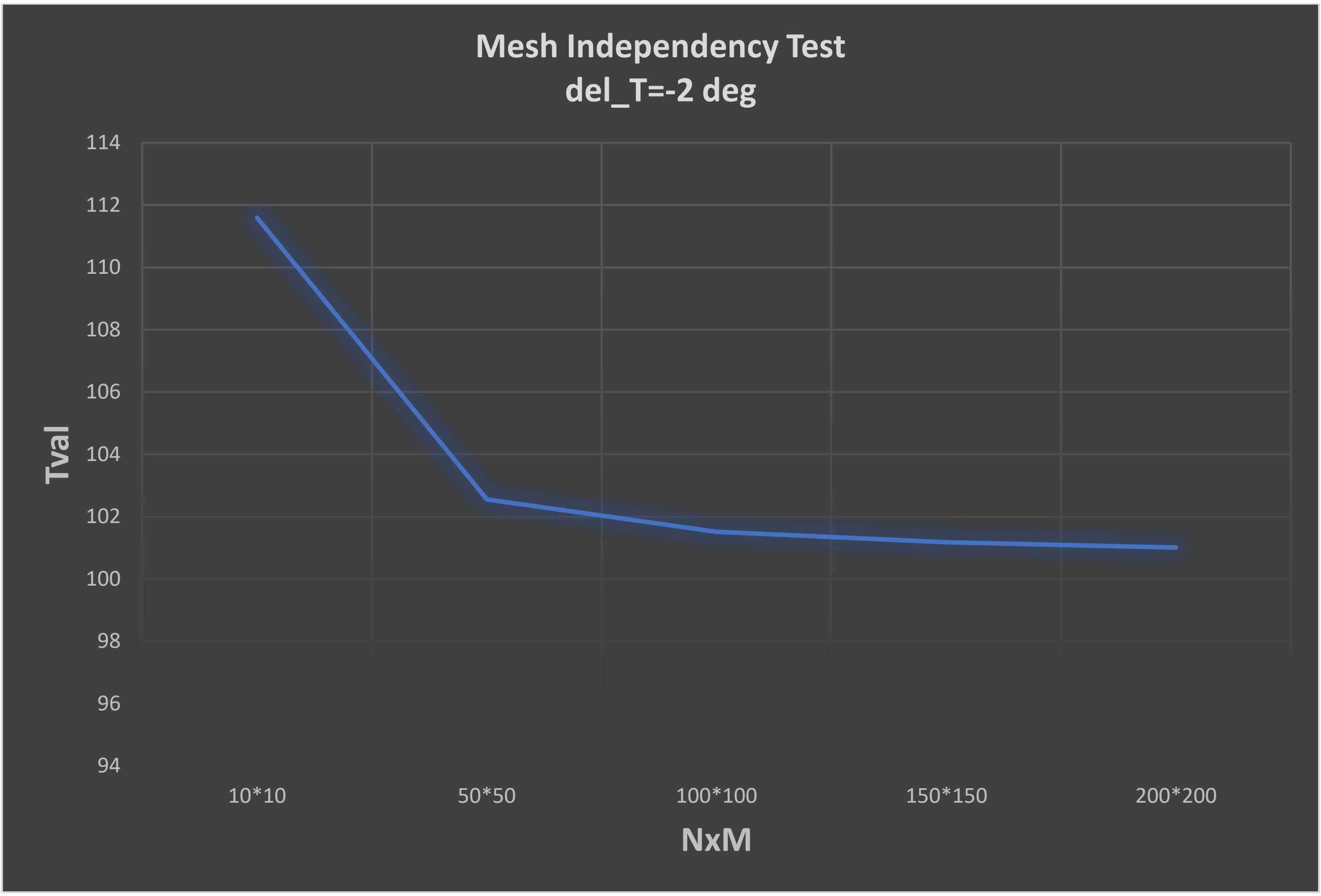
Mesh Independency Test: -

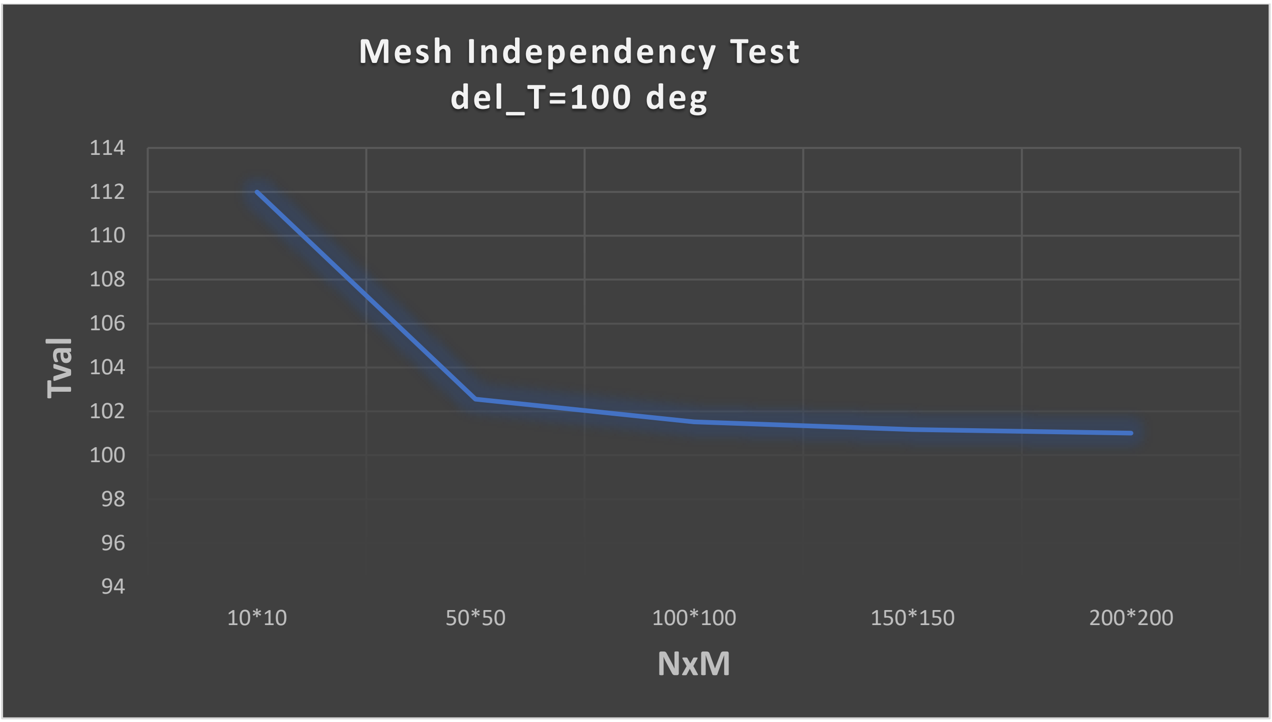
Mesh Independency Test:

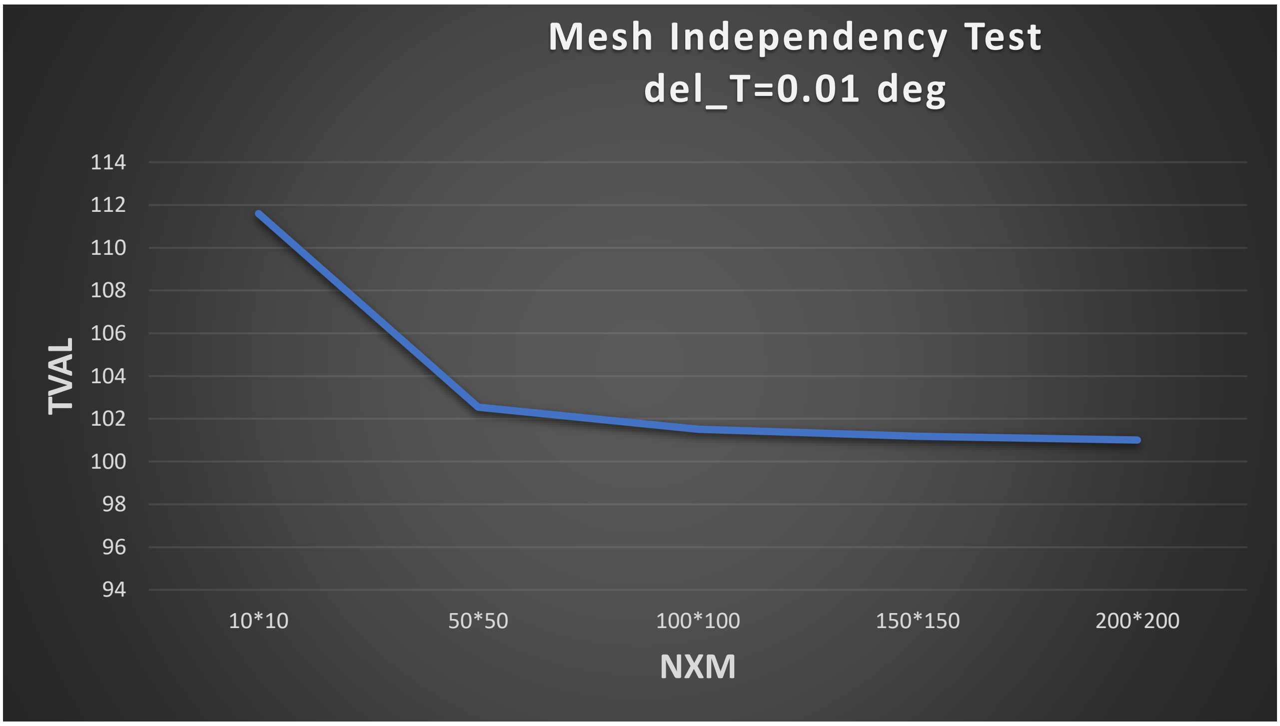
	N × M	Tav | x=0.5L
$\Delta T = 2°C$	10 × 10	111.608 deg
	50 × 50	102.546 deg
	100 × 100	101.516 deg
	150 × 150	101.177 deg
	200 × 200	101.008 deg
$\Delta T = -2°C$	10 × 10	111.592 deg
	50 × 50	102.546 deg
	100 × 100	101.516 deg
	150 × 150	101.177 deg
	200 × 200	101.006 deg
$\Delta T = 100°C$	10 × 10	111.992 deg
	50 × 50	102.56 deg
	100 × 100	101.519 deg
	150 × 150	101.178 deg
	200 × 200	101.009 deg
$\Delta T = 0.01°C$	10 × 10	111.6 deg
	50 × 50	102.546 deg
	100 × 100	101.516 deg
	150 × 150	101.177 deg
	200 × 200	101.008 deg

REDMI 9 POWER | SWAPNENDU
2023.10.27 00:16


Mesh Independency Test
del_T=2 deg
TVAL
114
112
110
108
106
104
102
100
98
96
94
10*10
50*50
100*100
150*150
200*200
NXM


Mesh Independency Test
del_T=-2 deg
Tval
114
112
110
108
106
104
102
100
98
96
94
10*10
50*50
100*100
150*150
200*200
NxM





Thus, we can conclude that on finding the Temperature at x=0.5L is more or less constant on taking a mesh above 100 x100.