2024

Bauhaus Universität Weimar Computer models for physical processes Finite Difference Method for Stationary 2D heat conduction problem

SUBMITTED TO
PROFESSOR CARSTEN KÖNKE

SUBMITTED BY
EYASU BERHANE HABTE [125461]
SAYAN DATTA [125384]
ADLEY JOHN DSOUZA [125163]

Acknowledgement

At first, we would like to express our gratitude to Prof. Carsten Könke for assigning us this project on Computer models for Physical Processes, which helped us in doing a lot of research and were able to learn many new things.

We would also like to earnestly acknowledge the sincere efforts and valuable time and feedback and complete guidance during each week, which has helped us in successful completion of the project.

Declaration

We hereby declare that this project report is based on our own work, carried out during the study period of winter semester 2023.

We declare the conclusions drawn and statements made are an outcome of our research work. We further certify that,

- The report which contains our work is original and has been done by us together.
- The work has not been submitted to any other organization or universities.
- We have strictly followed the rules and guidelines issued by the university in writing the complete report.

The credit for the information, if taken from other sources are mentioned in the references.

Table of Contents

1 Problem Definition

For the shown system of a 2D heat conduction problem [Refer fig. 1] a numerical approximation solution shall be developed and implemented into MATLAB® script.

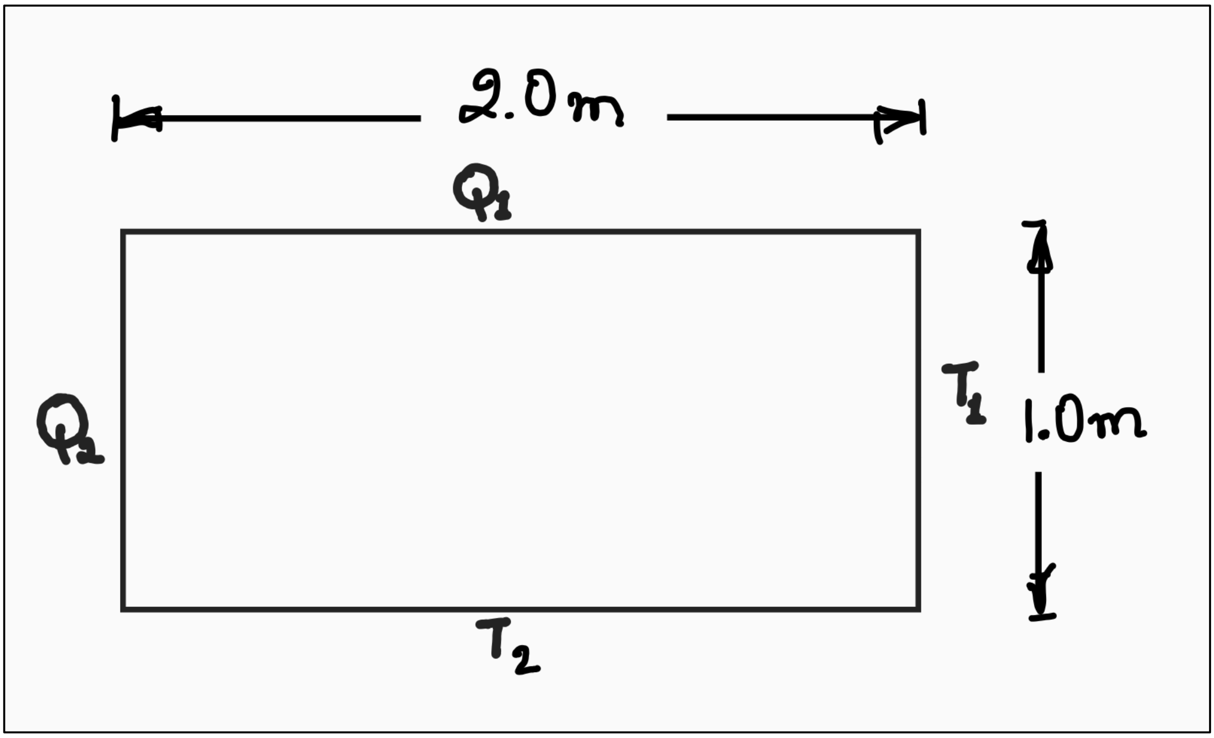


Figure 1 : 2D Plate geometry

The material parameters are as below,

$$\text{heat conductivity } c = 50\frac{W}{mK}$$
$$\text{thickness of plate } h = 0.15m$$

The boundary conditions are given as follows:

$$T_1 = 50°C$$
$$T_2 = 10°C$$
$$Q_1 = -200\frac{W}{m^2}\ (\text{heat output})$$
$$Q_2 = 0\frac{W}{m^2}$$

The dimensions of the plate are given as below,

$$2.0m \times 1.0m$$

2 Methodology

Objective: Determine the temperature distribution within the plate in a steady-state condition.

1. Physical Problem Formulation
   - **Geometry consideration:** the rectangular plate's dimensions, material properties, and boundary conditions.
   - **Heat Transfer Consideration:** heat transfer within the plate due to thermal conductivity and the specified boundary conditions (specified temperatures and heat fluxes).
2. Mathematical Formulation to establish an appropriate finite difference term.
3. Investigation of the temperature field using MATLAB® implementation
4. Analytical validation using ANSYS® Fluent
5. Evaluation of results

3 Implementation

3.1 Mathematical Formulation to establish an appropriate finite difference term.

3.1.1 General equation for heat conduction

The general equation for non-steady-state heat conduction with non-constant thermal conductivity (C) and considering internal heat generation in the system can be expressed as the transient heat conduction equation. The equation is expressed by:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} + \frac{Q}{K} = \rho \times C \times \frac{\partial T}{\partial t}$$

Since our problem is under 2D steady-state heat conduction and there is no internal heat generation

If no internal heat generation, then $Q = 0$ therefore,

$$\frac{Q}{K} = 0$$

For steady state,

$$\frac{\partial T}{\partial t} = 0$$

And the above equation will be reduced into the equation given below,

- **Heat Conduction Equation:** In 2D, the steady-state heat conduction equation is given by Laplace's equation: $\nabla^2 T = 0$,

  $$\boxed{\nabla^2 T = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0}$$

  Where T is the temperature distribution within the 2D plate.

  $\frac{\partial^2 T}{\partial x^2}$ and $\frac{\partial^2 T}{\partial y^2}$ represent the second order partial derivatives of temperature with respect to x and y directions, respectively.

3.1.2 Assumptions and Simplifications:

- **Steady-State Condition:** Assumes that the temperature distribution within the plate remains constant over time $\frac{\partial T}{\partial t} = 0$.
- **Constant Thermal Conductivity:** Assumes that the material's thermal conductivity (c) remains constant within the plate.
- **No Internal Heat Generation**

3.1.3 Approximation of Derivatives using Finite Differences:

The step involving the approximation of derivatives using finite differences is crucial in discretizing the partial derivatives of the heat conduction equation to solve a 2D heat conduction problem. Let's break down the process from calculating first-order to second-order derivatives using finite differences:

Consider a plate which is discretised in the x- and y-directions [Refer fig. 2],

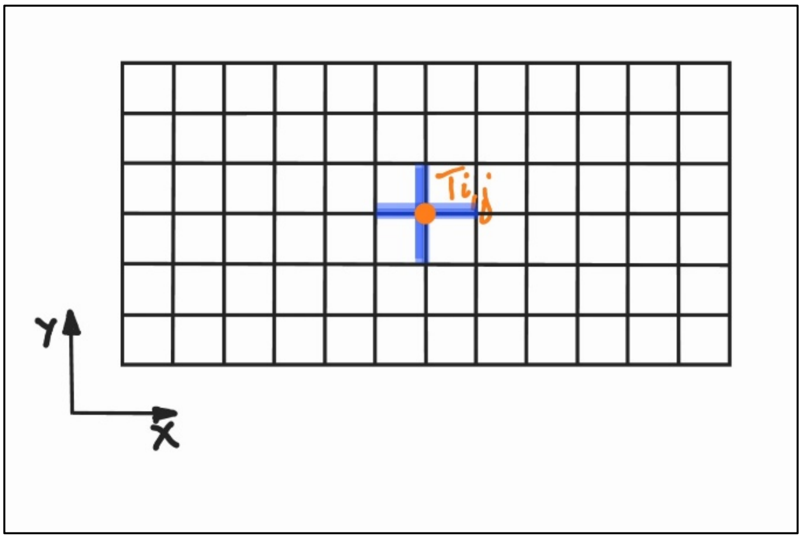


Figure 2 : 2D discretisation of the plate

The finite difference term for a point $T_{i,j}$ is derived as a function of the neighbouring grid points as represented in the figure 3.

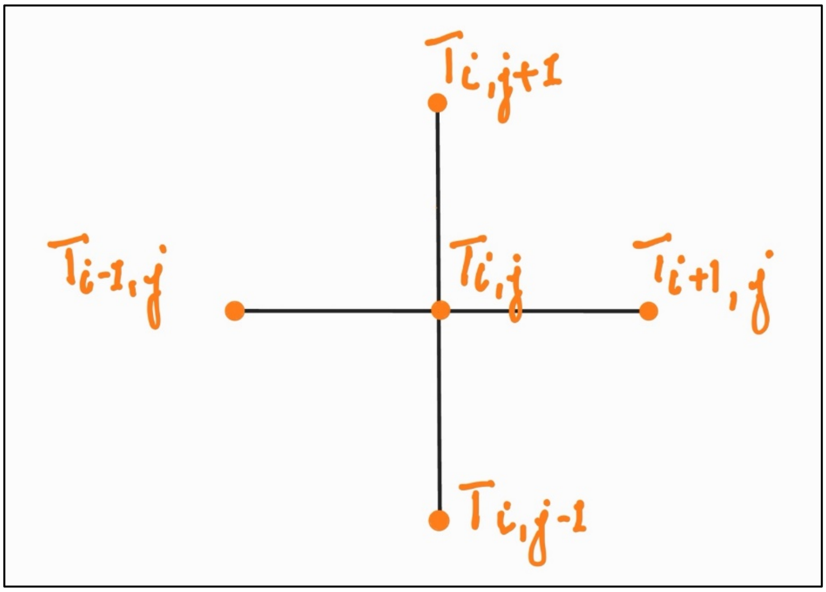


Figure 3: Neighbouring points

We proceed with the derivation using the above understanding from the figures and, arrive at the required equation to estimate the temperature value at individual grid points.

1. First-order Derivatives: Approximation of $\frac{\partial T}{\partial \mathrm{x}}$ and $\frac{\partial T}{\partial \mathrm{y}}$

   For a function T at a point (i, j) in the grid:

- Forward Difference (for $\frac{\partial T}{\partial \mathrm{x}}$):
  $\frac{\partial T}{\partial \mathrm{x}} \approx \frac{(T_{i+1,j} - T_{i,j})}{\Delta x}$
- Backward Difference (for $\frac{\partial T}{\partial \mathrm{x}}$):
  $\frac{\partial T}{\partial \mathrm{x}} \approx \frac{(T_{i,j} - T_{i-1,j})}{\Delta x}$
- Central Difference (for $\frac{\partial T}{\partial \mathrm{x}}$):
  $\frac{\partial T}{\partial \mathrm{x}} \approx \frac{(T_{i+1,j} - T_{i-1,j})}{2 \times \Delta x}$

Similarly, approximations for $\frac{\partial T}{\partial \mathrm{y}}$ using forward, backward, or central differences in the y-direction are calculated.

2. Second-order Derivatives:

   Approximation of $\frac{\partial^2 T}{\partial x^2}$ and $\frac{\partial^2 T}{\partial y^2}$:

   For a function T at a point (i, j) in the grid:

- Central Difference $(for\ \frac{\partial^2 T}{\partial x^2})$:

$$\frac{\partial^2 T}{\partial \mathrm{x}^2} \approx \frac{(T_{i+1,j} - 2T_{i,j} + T_{i-1,j})}{\Delta x^2}$$

- Central Difference $(for\ \frac{\partial^2 T}{\partial y^2})$:

$$\frac{\partial^2 T}{\partial y^2} \approx \frac{(T_{i,j+1} - 2T_{i,j} + T_{i,j-1})}{\Delta y^2}$$

The above equation is called the *Discretized Heat Conduction Equation*.

Applying the discretized second-order derivatives in the original 2D heat conduction equation:

$$\nabla^2 T = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0$$

Substituting the discretized forms of the derivatives in the above equation, we get:

$$\frac{\left(T_{i+1,j} - 2T_{i,j} + T_{i-1,j}\right)}{\Delta x^2} + \frac{\left(T_{i,j+1} - 2T_{i,j} + T_{i,j-1}\right)}{\Delta y^2} = 0$$

$$\frac{\left[(\Delta y^2) \times \left(T_{i+1,j} - 2T_{i,j} + T_{i-1,j}\right)\right] + \left[(\Delta x^2) \times \left(T_{i,j+1} - 2T_{i,j} + T_{i,j-1}\right)\right]}{(\Delta x^2 \times \Delta y^2)} = 0$$

$$\left[(\Delta y^2) \times \left(T_{i+1,j} - 2T_{i,j} + T_{i-1,j}\right)\right] + \left[(\Delta x^2) \times \left(T_{i,j+1} - 2T_{i,j} + T_{i,j-1}\right)\right] = 0$$

$$\begin{aligned}&\left(\Delta y^2 \times 2T_{i,j}\right) + (\Delta x^2 \times 2T_{i,j}) \\ &\qquad = (\Delta y^2 \times T_{i+1,j}) + (\Delta y^2 \times T_{i-1,j}) + (\Delta x^2 \times T_{i,j+1}) + (\Delta x^2 \times T_{i,j-1})\end{aligned}$$

$$(\Delta x^2 + \Delta y^2) \times 2T_{i,j} = (\Delta y^2) \times (T_{i+1,j} + T_{i-1,j}) + (\Delta x^2) \times (T_{i,j+1} + T_{i,j-1})$$

$$\boxed{\boldsymbol{T_{i,j}} = \frac{(\boldsymbol{\Delta y^2}) \times (\boldsymbol{T_{i+1,j}} + \boldsymbol{T_{i-1,j}}) + (\boldsymbol{\Delta x^2}) \times (\boldsymbol{T_{i,j+1}} + \boldsymbol{T_{i,j-1}})}{\mathbf{2} \times (\boldsymbol{\Delta x^2} + \boldsymbol{\Delta y^2})}}$$

If we have the same grid spacing in x and y direction ($\Delta x = \Delta y$) above equation is simplified to,

$$\boxed{\boldsymbol{T_{i,j}} = \frac{(\boldsymbol{T_{i+1,j}}) + (\boldsymbol{T_{i-1,j}}) + (\boldsymbol{T_{i,j+1}}) + (\boldsymbol{T_{i,j-1}})}{\mathbf{4}}}$$

The above equations represent a discrete estimation for each interior grid point (i, j) based on its neighbouring grid points. It characterizes the temperature distribution within the rectangular plate using finite differences.

3.1.4 Boundary Conditions

Additionally, the discretized equations for the boundary grid points (edges of the plate) need to reflect the specified boundary conditions. For instance:

1. At points where temperatures are specified (Dirichlet boundary conditions):

   Directly the temperature values at these grid points are set to the specified values

$$right\ side\ column\ of\ grid\ points = T_1$$

$$bottom\ side\ row\ of\ grid\ points = T_2$$

2. At points where heat fluxes are specified (Neumann boundary conditions):

   Finite difference approximations consistent with the given heat fluxes at the boundary based on the previous grid temperature are applied.

   For left side boundary condition:

$$q = -C \times \frac{\partial T}{\partial \mathrm{x}}$$

$$q = -C \times \frac{(T_{i,j} - T_{i+1,j})}{\Delta x}$$

   Therefore, the temperature values are given by:

$$T_{i,j} = (T_{i+1,j}) - \frac{(q \times \Delta x)}{C}$$

   For top side boundary condition:

$$q = -C \times \frac{\partial T}{\partial \mathrm{y}}$$

$$q = -C \times \frac{(T_{i,j} - T_{i,j-1})}{\Delta y}$$

   Therefore, the temperature values are given by:

$$T_{i,j} = (T_{i,j-1}) - \frac{(q \times \Delta y)}{C}$$

   Where:

   q is the predefined heat output or input (heat flux) at the boundary

   C is the thermal conductivity of the material

   Δx and Δy are grid spacing with respect to x and y direction respectively

   $\frac{\partial T}{\partial \mathrm{x}}\ and \frac{\partial T}{\partial \mathrm{y}}$ are temperature gradient (rate of change of temperature with respect to x and y direction respectively).

3.2 Investigation of the temperature field using MATLAB®

3.2.1 Discretization:

The 2D space is discretised into a grid $(N_x \times N_y)$ [Refer fig. 4] and derivatives using finite differences are approximated.

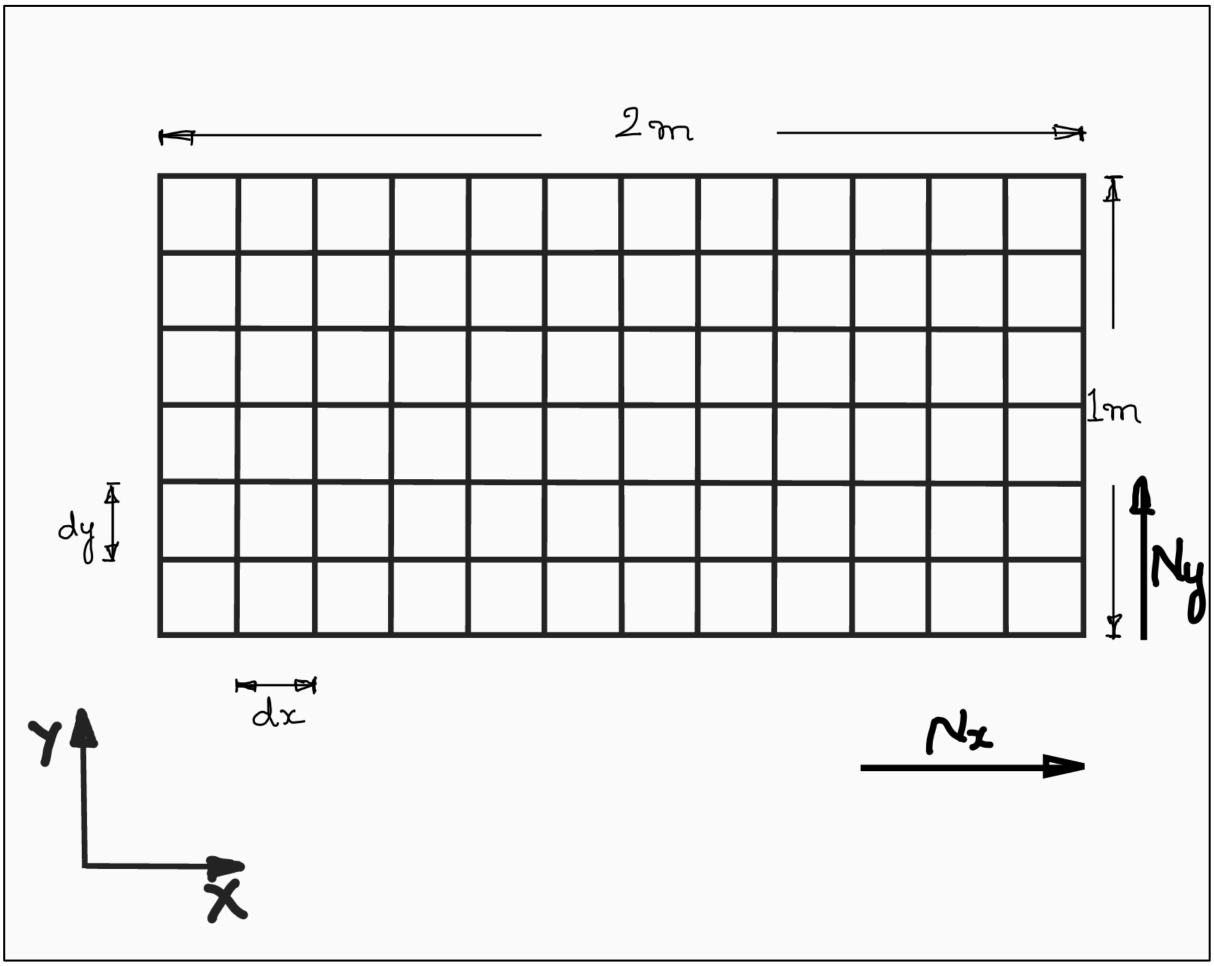


Figure 4: 2D discretization into N_x&N_y

3.2.2 Iterative Solver:

We employed an iterative technique known as Gauss-Seidel for addressing our problems. This method systematically resolves a system of discrete equations through iterative updates until convergence. It commences with an initial approximation for the solution and progressively refines it until reaching the actual solution. The determination of convergence relies on assessing the solution against a predefined tolerance level.

Let's break down the problem step by step:

The number of grids in x-direction is given by N_x.

The number of grids in y-direction is given by N_y.

Consider the length of the plate as L_x and the breadth of the plate as L_y.

The grid spacing is calculated in the x-direction, dx, and y-direction, dy, as follows,

$$dx = \frac{L_x}{N_x - 1} \qquad dy = \frac{L_y}{N_y - 1}$$

The temperature values of each grid point are set to a initial value and stored in T.

Given boundary conditions are used to set the temperatures of the grid points as below,

$$T_{rightside\ end\ column} = T_1$$

$$T_{bottom\ side\ end\ row} = T_2$$

And subsequently the value of the boundary grid points with heat flux [Refer fig. 5] are calculated using the equations as defined before,

$$T_{top\ side\ end\ row} = T_{top\ side\ end\ row-1} - \frac{(Q_1 \times d_y)}{c}$$

$$T_{left\ side\ end\ column} = T_{left\ side\ end\ column-1} - \frac{(Q_2 \times d_x)}{c}$$

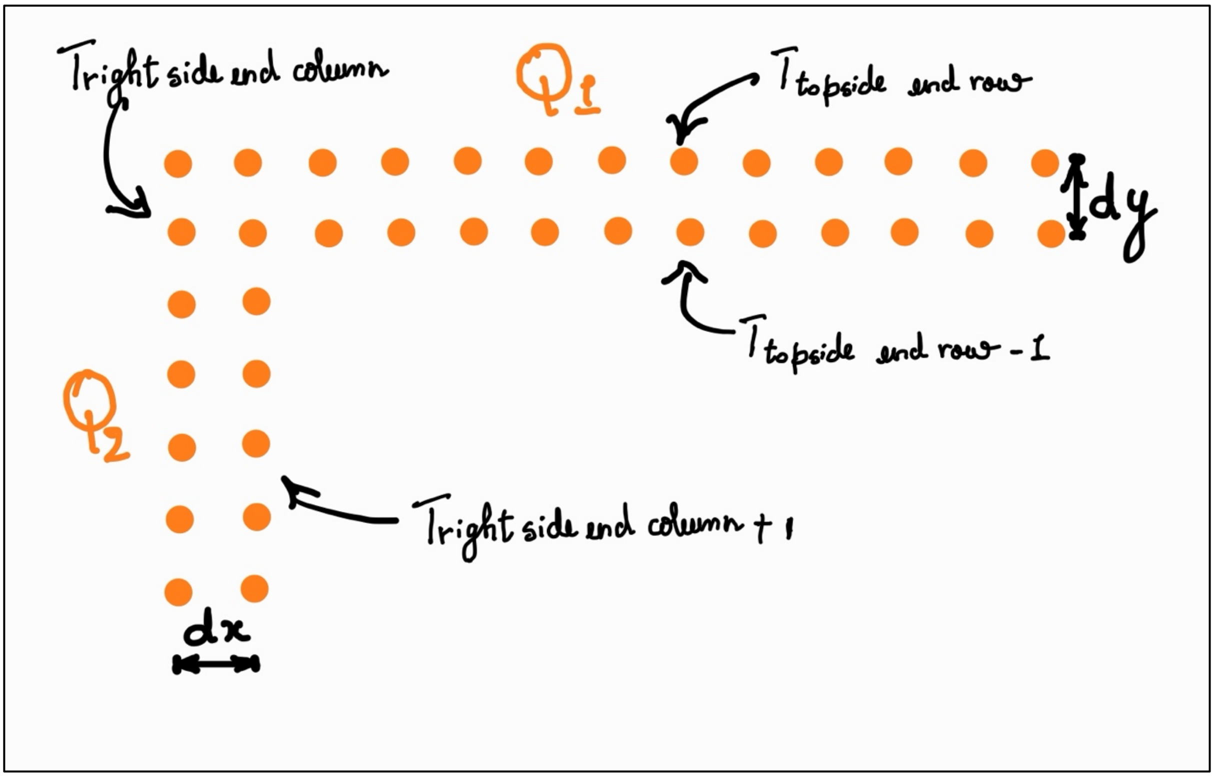


Figure 5: Heat flux boundary condition

Once the boundary conditions are set the value of each grid point as indicated in the figure apart from $T_{rightside\ end\ column}$ and $T_{bottom\ side\ end\ \ row}$ are iteratively calculated.

Consider a grid point as given below in figure 6,

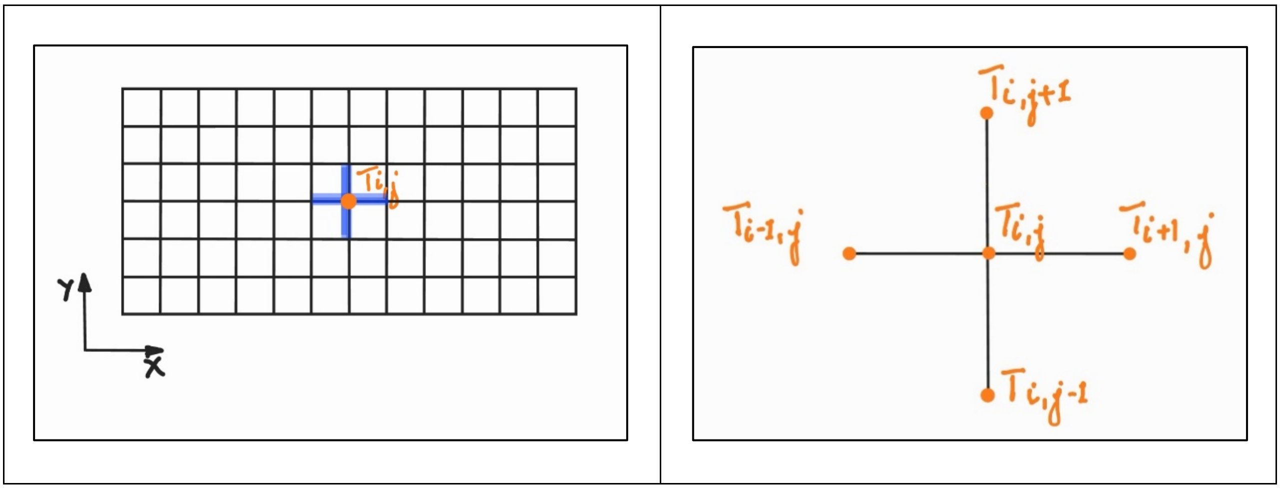


Figure 6: 2D discretization and neighbouring points

The value of $T_{i,j}$ is calculated in MATLAB® going over each grid point left to right and through each row, and from bottom to top. Each point in the grid is calculated according to the below derived equation,

$$\boxed{\boldsymbol{T_{i,j}} = \frac{(\boldsymbol{\Delta y^2}) \times (\boldsymbol{T_{i+1,j}} + \boldsymbol{T_{i-1,j}}) + (\boldsymbol{\Delta x^2}) \times (\boldsymbol{T_{i,j+1}} + \boldsymbol{T_{i,j-1}})}{\boldsymbol{2} \times (\boldsymbol{\Delta x^2} + \boldsymbol{\Delta y^2})}}$$

$$T_{top\ side\ end\ \ row} = T_{top\ side\ end\ row-1} - \frac{(Q_1 \times d_y)}{c}$$

Since $Q_2 = 0$

$$T_{left\ side\ end\ \ column} = T_{left\ side\ end\ column-1}$$

The calculation is done iteratively until the maximum number of iterations or when the value stabilises. The stabilisation is evaluated by calculating the error after each iteration. Once these conditions are achieved, we obtain the temperature values of the 2D stable system.

The calculation of error is done as below,

$$Error = maximum\ of\ the\ absolute\ value\ of\ (T_{new} - T_{old})$$

Step 1: The temperature values of individual grid points are set to the initial values as represented in the below figure 7. As can be seen from the figure the values on the right end column and the bottom side row are constant throughout the iterations according to the given boundary conditions.

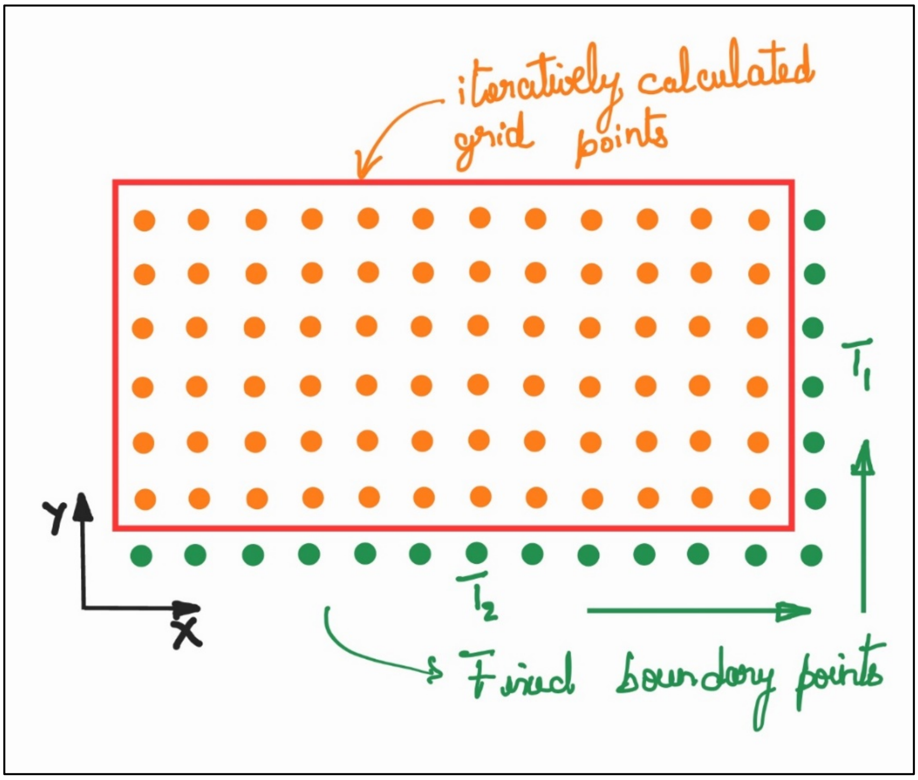


Figure 7: Iterative method for stabilization of the temperature values

Step 2: The calculation of temperature values is executed from the bottom and moves step by step, from the left to right and through individual rows to complete all the grid points [Refer fig. 8].

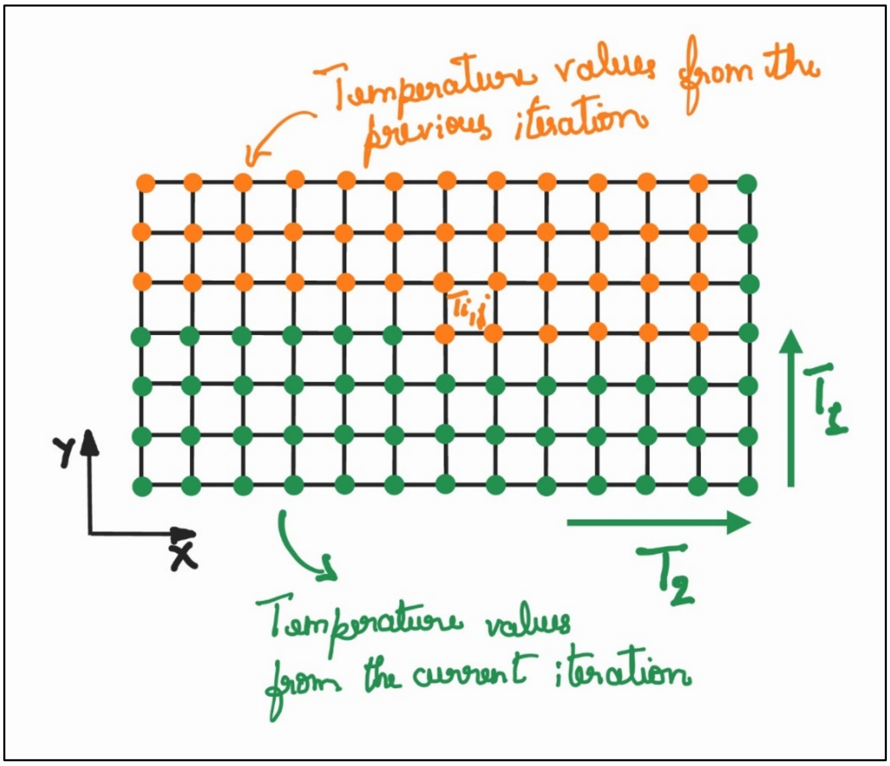


Figure 8: Iterative method in progress

Step 3: The calculations repeat until required tolerance of error is achieved or the maximum number of iterations have elapsed [Refer fig. 9].

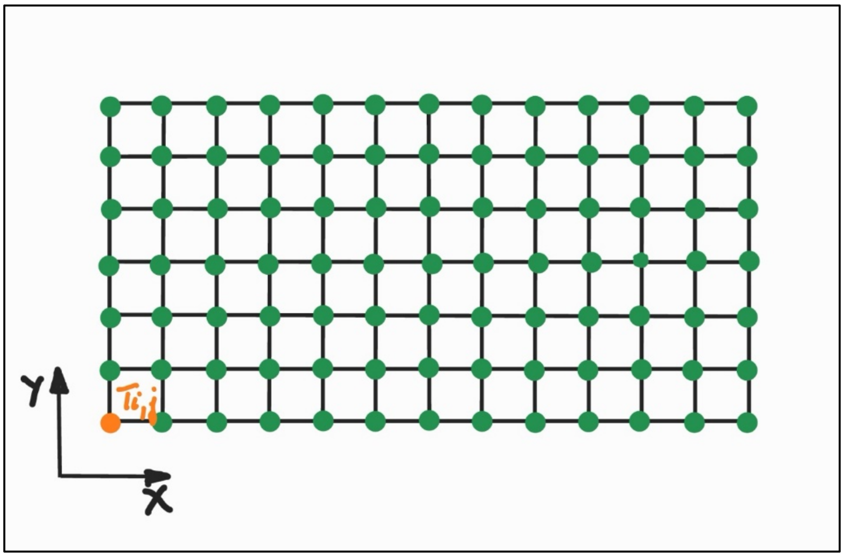


Figure 9: Final estimation of temperature values

4 Analytical validation using ANSYS®

4.1 Methodology followed:

1. The Fluid Flow (Fluent) in ANSYS® Workbench [Refer fig.10] is used to do the required simulation for analytical validation.

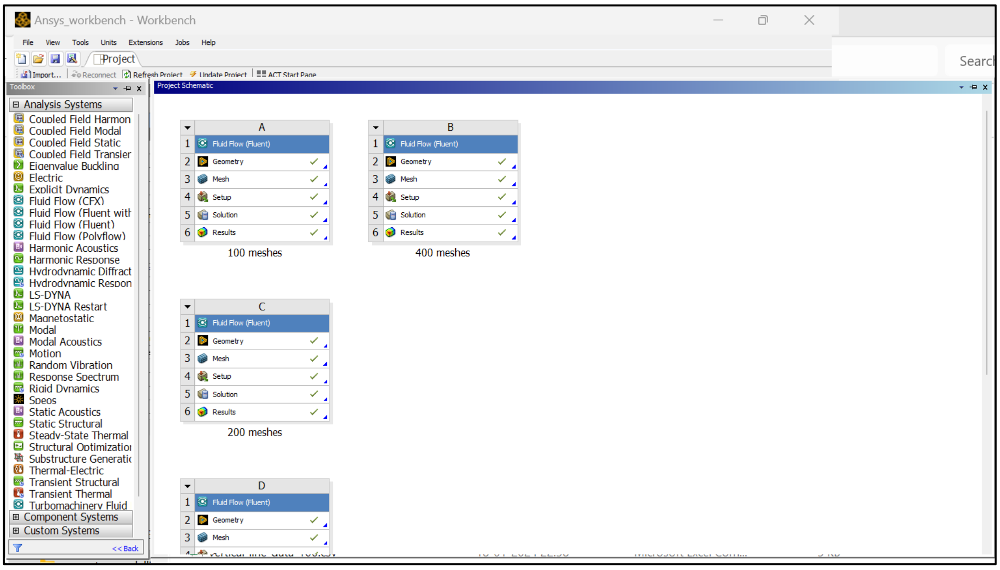


Figure 10: ANSYS® Workbench (Fluid Flow)

2. Simulation is done for 100, 200 and 300 meshes and the result is compared with the MATLAB® output with respect to the corresponding number of meshes.

3. To compare the data, temperature data of horizontal mid-point line and vertical mid-point line of the plate is used [Refer fig. 11].

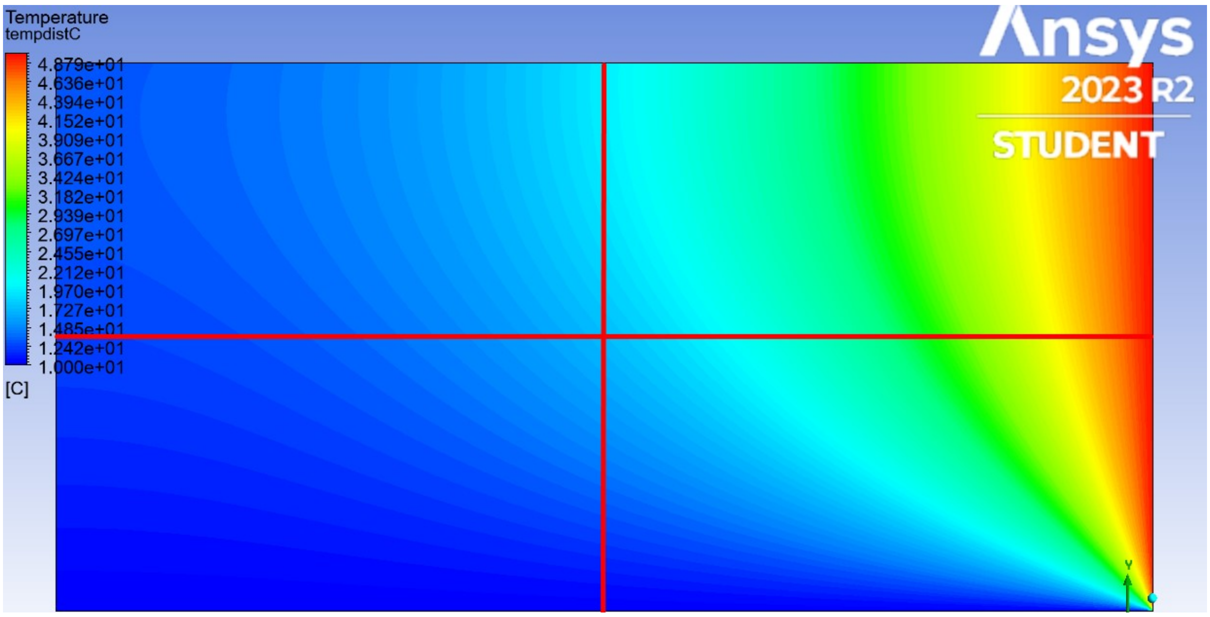


Figure 11: Vertical and horizontal midpoint lines for extracting temperature data

4. We divide the mid-point lines into grid points and extract the temperature value for each grid point. The grid points correspond to the number of meshes [Refer fig. 12].

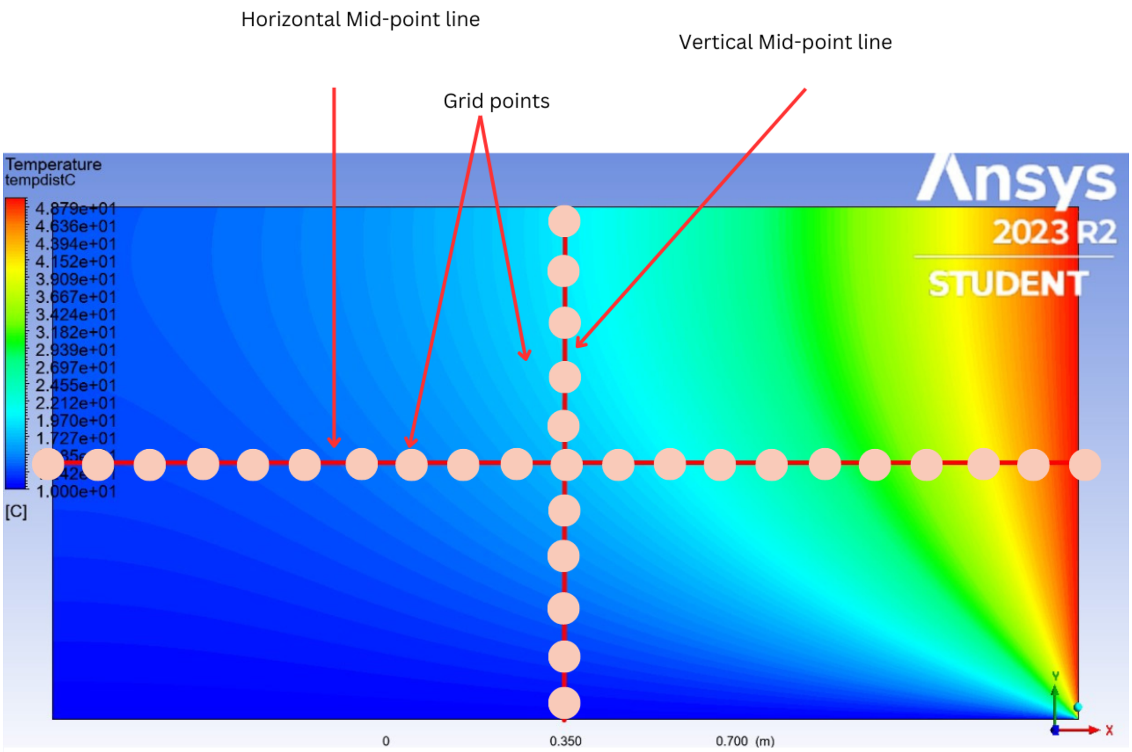


Figure 12: Grid points on the vertical and horizontal midpoint lines

5. We then compare the ANSYS® simulation data with the output of MATLAB®. To get the output from the MATLAB® we follow the same principle and we find the vertical and horizontal mid-point line data.

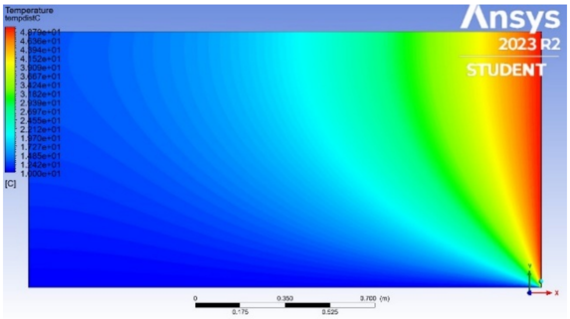
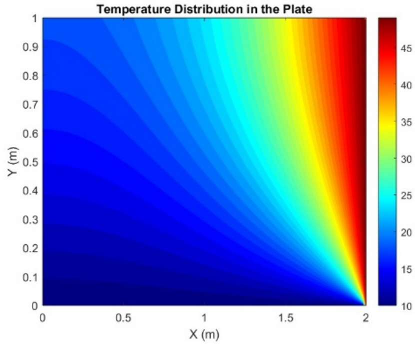
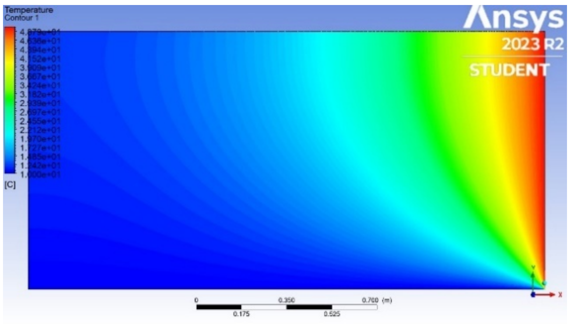
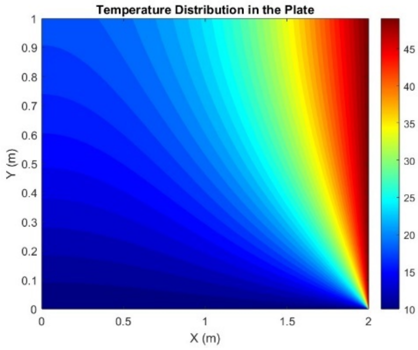
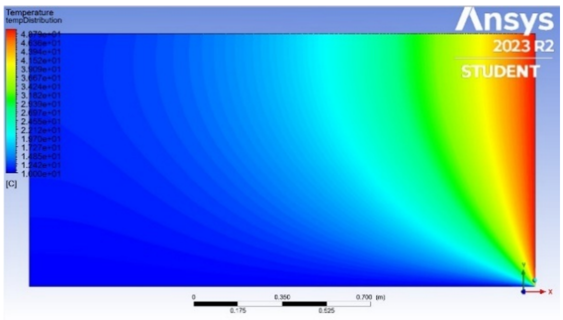
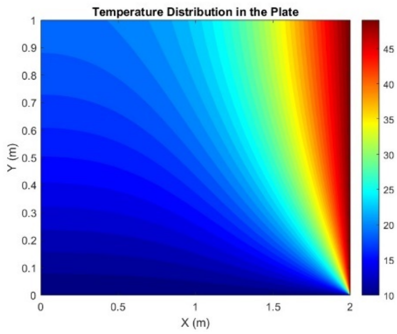
Code:
horizontal_midpoint_data = T(51,:);
verticle_midpoint_data = T(:,51);
Where: T --→ Temperature matrix

Note: This code is used when there is 100 meshes. The number need to be changed based on the number of meshes.

5 Results

5.1 Plot comparison

Table 1: MATLAB® Vs ANSYS® temperature distribution plot

No. of mesh	ANSYS® Plot	MATLAB® Plot
100		
200		
300		

From Table 1, it can be seen that the temperature distribution is visually similar between MATLAB® implementation and ANSYS®. The extreme values are approximately having the same colour representation. Furthermore to have an analytical comparison of the results the grid point values are extracted from MATLAB® and ANSYS® respectively. The absolute difference of the extracted values is computed.

5.2 Data Analysis

For data analysis we find the maximum and average difference between ANSYS® and MATLAB® result. We see a very high degree of correlation between the two results as shown in the Table 2 below which gives more confidence on the MATLAB® implementation.

Table 2: Analytical evaluation of the temperature values

No. of Meshes	Horizontal Midpoint Line		Vertical Midpoint Line	
	Max difference	Average difference	Max difference	Average difference
100	2.51276381	1.832376204	04.3920025	2.147666243
200	2.07700129	1.726038221	04.1900000	2.052288557
300	0.55606093	0.237455986	0.94312382	0.225470157

6 Conclusion

1. We have successfully established a finite difference term for the 2D heat conduction problem according to the boundary conditions and it is given as below x- and y-directions,

   For unequal grid spacing

$$\boxed{T_{i,j} = \frac{(\Delta y^2) \times (T_{i+1,j} + T_{i-1,j}) + (\Delta x^2) \times (T_{i,j+1} + T_{i,j-1})}{2 \times (\Delta x^2 + \Delta y^2)}}$$

   For equal grid spacing,

$$\boxed{T_{i,j} = \frac{(T_{i+1,j}) + (T_{i-1,j}) + (T_{i,j+1}) + (T_{i,j-1})}{4}}$$

2. Furthermore the derived equation is implemented in MATLAB® and the stabilised temperature values using the Gauss Seidel iterative method are computed.
3. Given 2D heat conduction problem is defined and simulated using the ANSYS® Software to have a thorough analysis and verification of the MATLAB® implemented code.
4. The evaluation of the results reveal a very high degree of correlation between the implemented MATLAB® code and the ANSYS® simulation.
5. Thus the implementation can be extended to solve other 2D heat conduction problems and will lead to accurate results.

7 References

1. Könke, Carsten. "Numerical Discretization Methods " Computer Models for Physical Processes. Bauhaus-Universität. Weimar. WiSe 2023.
2. Powell, Adam. "Finite Difference Solution of the Heat Equation" dspace.mit.edu, 15 December 2023, www.dspace.mit.edu/bitstream/handle/1721.1/35256/22-00JSpring-2002/NR/rdonlyres/Nuclear-Engineering/22-00JIntroduction-to-Modeling-and-SimulationSpring2002/55114EA2-9B81-4FD8-90D5-5F64F21D23D0/0/lecture_16.pdf
3. dos Santos, Vânia Gonçalves de Brito, and Paula Tamires Gomes dos Anjos. "Finite Difference Method Applied in Two-Dimensional Heat Conduction Problem in the Permanent Regime in Rectangular Coordinates." *Advances in Pure Mathematics* 12.9 (2022): 505-518.