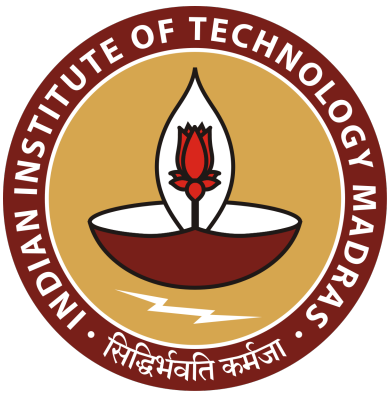
AM5630 Foundations of Computational Fluid Dynamics

Assignment 1

Heat Conduction Problem In 2 Dimensions : A Numerical Study

Submitted by:

AM21S082 Shouvik Ghorui
AE19B027 Andrea Elizabeth Biju



Indian Institute of Technology Madras

11 September, 2022

Contents

Chapter 1

Finite Volume Method for Discretization of Convection Equations in 2D Domain

1.1 Convection Equations

The general equation for transport of any physical quantity ϕ can be written as [1]:

$$\frac{\partial \rho\phi}{\partial t} + \frac{\partial \rho\phi u_j}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\Gamma \frac{\partial \phi}{\partial x_j}\right) + S_\phi \tag{1.1}$$

where ρ is the density of physical quantity per unit volume, u_j is the component of velocity of flow in the j direction, and Γ is coefficient of diffusion of the physical quantity. The first term of equation 1.1 is the unsteady term (since it is non-zero only in the case of unsteady transport), the second term is the convection term, the third is the diffusion term, and the fourth is the source term. For the case of steady diffusion, we can drop the unsteady and convection terms, to give:

$$\frac{\partial}{\partial x_j}\left(\Gamma \frac{\partial \phi}{\partial x_j}\right) + S_\phi = 0 \tag{1.2}$$

Equation 1.2 is called the differential form of the steady diffusion equation.

1.2 Finite Volume Method for Discretization

Due to large domains, nonlinearities in the equation or complex boundary conditions, analytical solutions may not exist, and it may not be possible to

integrate these equations at every point of the entire domain. Hence, we need to discretize the system of equations and solve them over a finite number of points to derive an approximate result. Integrating over a control volume ΔV:

$$\int_{\Delta V} \frac{\partial}{\partial x_j}\left(\Gamma \frac{\partial \phi}{\partial x_j}\right) dV + \int_{\Delta V} S_\phi dV = 0 \tag{1.3}$$

Using Gauss divergence theorem, we can rewrite equation 1.3 as:

$$\int_{\Delta A} \frac{\partial}{\partial x_j}\left(\Gamma \frac{\partial \phi}{\partial x_j}\right) dA + \int_{\Delta V} S_\phi dV = 0 \tag{1.4}$$

1.2.1 Meshing

For a 2D problem, we can evaluate the integral over a finite number of points sampled in a particular fashion from the domain under consideration. Let us choose a few nodal points distributed along the x and y directions. Assume the boundaries of the grid along a particular direction to be at the middle of two adjacent nodes. Now we have a mesh grid comprising of rectangular cells. A mesh where the boundaries of the cell are always at the middle of adjacent nodes is referred to as cell-centered mesh [2].

The notation followed for the mesh along x-axis is shown in the figure 1.1. The node under consideration is P, the node on its left is assigned E (east) and the one on its right is assigned as W (west). The east face is denoted by e and west face node is denoted by w. The distance between W and P is denoted as δx_{WP} or simply δx_w, between P and E is denoted as δx_{PE} or simply δx_e, and between w and e is denoted as δx_{we} or simply Δx. Similarly, along the y-axis, the node on the top of P is assigned N (north) and the one on the bottom is assigned as S (south). The north face is denoted by n and south face node is denoted by s. The distance between S and P is denoted as δy_{SP} or simply δy_s, between P and E is denoted as δx_{PN} or simply δy_n, and between s and n is denoted as δy_{sn} or simply Δy.

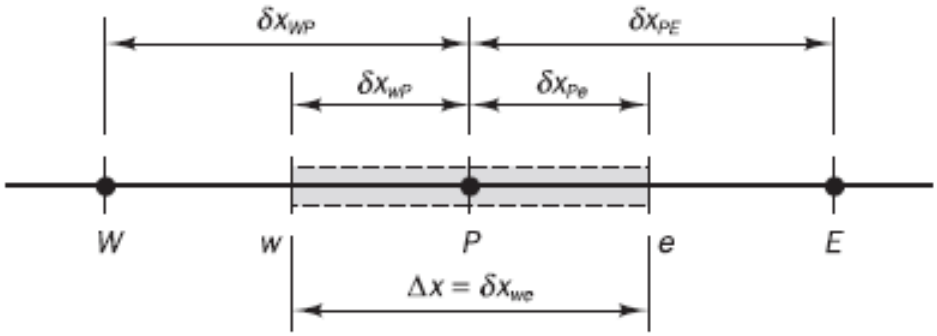


Figure 1.1: Notations followed along x-axis [1]

1.2.2 Discretization

Now the equation 1.4, when evaluated over each cell can be simplified into:

$$\left(\Gamma A\frac{d\phi}{dx}\right)_e - \left(\Gamma A\frac{d\phi}{dx}\right)_w + \left(\Gamma A\frac{d\phi}{dy}\right)_n - \left(\Gamma A\frac{d\phi}{dy}\right)_w + \bar{S}\Delta V = 0 \tag{1.5}$$

Here $\bar{S}$ is the averaged source term over the control volume ΔV. This equation represents the mathematical statement of the physical law that the flux entering the cell through the west and south faces minus the flux leaving the cell through the east and north faces must equal the generation of ϕ in the control volume [1]. To approximate for the terms $\frac{d\phi}{dx}$ and $\frac{d\phi}{dy}$, we use a linear interpolation, also known as central differencing method [1]. This method works best for diffusion dominated flows [2]. Hence,

$$\begin{aligned}
\Gamma_e &= \frac{\Gamma_E + \Gamma_P}{2} \\
\Gamma_w &= \frac{\Gamma_W + \Gamma_P}{2} \\
\Gamma_n &= \frac{\Gamma_N + \Gamma_P}{2} \\
\Gamma_s &= \frac{\Gamma_S + \Gamma_P}{2}
\end{aligned} \tag{1.6}$$

and diffusive flux terms can be written as:

$$\begin{aligned}
\left(\Gamma A\frac{d\phi}{dx}\right)_e &= \Gamma_e A_e\left(\frac{\phi_E-\phi_P}{\delta x_e}\right)\\
\left(\Gamma A\frac{d\phi}{dx}\right)_w &= \Gamma_w A_w\left(\frac{\phi_P-\phi_W}{\delta x_w}\right)\\
\left(\Gamma A\frac{d\phi}{dy}\right)_n &= \Gamma_n A_n\left(\frac{\phi_N-\phi_P}{\delta y_n}\right)\\
\left(\Gamma A\frac{d\phi}{dy}\right)_s &= \Gamma_s A_s\left(\frac{\phi_P-\phi_S}{\delta y_s}\right)
\end{aligned} \tag{1.7}$$

The source term can be split into two terms, one dependent on the nodal point value of ϕ, and the other independent of it.

$$\bar{S}\Delta V = S_u + S_p\phi_P \tag{1.8}$$

Substituting equations 1.6, 1.7, 1.8 into 1.5, and rearranging, we get:

$$\begin{aligned}
\left(\frac{\Gamma_e A_e}{\delta x_e}+\frac{\Gamma_w A_w}{\delta x_w}+\frac{\Gamma_n A_n}{\delta y_n}+\frac{\Gamma_s A_s}{\delta y_s}-S_p\right)\phi_P = &\left(\frac{\Gamma_e A_e}{\delta x_e}\right)\phi_E+\left(\frac{\Gamma_w A_w}{\delta x_w}\right)\phi_W\\
&+\left(\frac{\Gamma_n A_n}{\delta y_n}\right)\phi_N+\left(\frac{\Gamma_s A_s}{\delta y_s}\right)\phi_S+S_u
\end{aligned} \tag{1.9}$$

Using

$$\begin{aligned}
a_P &= \left(\frac{\Gamma_e A_e}{\delta x_e}+\frac{\Gamma_w A_w}{\delta x_w}+\frac{\Gamma_n A_n}{\delta y_n}+\frac{\Gamma_s A_s}{\delta y_s}-S_p\right)\\
a_E &= \left(\frac{\Gamma_e A_e}{\delta x_e}\right)\\
a_W &= \left(\frac{\Gamma_w A_w}{\delta x_w}\right)\\
a_N &= \left(\frac{\Gamma_n A_n}{\delta y_n}\right)\\
a_S &= \left(\frac{\Gamma_s A_s}{\delta y_s}\right)
\end{aligned} \tag{1.10}$$

we get:

$$a_P\phi_P = a_E\phi_E + a_W\phi_W + a_N\phi_N + a_S\phi_S + S_u \tag{1.11}$$

which is the final discretized equation. In our case, since the physical quantity of interest is temperature, we will henceforth use T in place of ϕ and Γ by K.

1.3 Boundary Conditions and Source / Sink Terms

1.3.1 Dirichlet Boundary Condition

Dirichlet boundary conditions are boundary conditions of the form:

$$T_A = T_0$$

where A indicates the boundary, and T_0 is a constant value. This boundary condition can be incorporated into the discretized equation for the boundary node in the following manner. Suppose the boundary under consideration is the west face of the cell, and distance of P from boundary A is δx_{AP}, then we have $a_W = 0$, and

$$\left(\frac{\Gamma_e A_e}{\delta x_e} + \frac{\Gamma_n A_n}{\delta y_n} + \frac{\Gamma_s A_s}{\delta y_s} - (-\frac{KA}{\delta x_{AP}})\right) T_P = 0T_W + a_E T_E + a_N T_N + a_S T_S + (-\frac{KA}{\delta x_{AP}})T_A \tag{1.12}$$

Note that the S_u term and the S_p terms capture the effect of the boundary conditions.

1.3.2 Neumann Boundary Conditions

Neumann boundary conditions are of the form:

$$\frac{dT}{dx_j} = 0$$

along a boundary. This boundary condition can be incorporated into the discretized equation for the boundary node in the following manner. Suppose the boundary under consideration is the west face of the cell, and distance of P from west boundary is δx_{WP}, then we have $a_W = 0$, and there will be no contribution to the source terms S_p and S_u, i.e.,

$$\left(\frac{\Gamma_e A_e}{\delta x_e} + 0 + \frac{\Gamma_n A_n}{\delta y_n} + \frac{\Gamma_s A_s}{\delta y_s} - 0\right) T_P = 0T_W + a_E T_E + a_N T_N + a_S T_S + 0 \tag{1.13}$$

which results in $\frac{dT}{dx} = 0$.

1.3.3 Source and Sink terms

As discussed in the case of 1.8, the source term has two contributions. In the case of a positive source term that doesn't depend on T_P, we can include its effect in S_u. But in the case it is negative, it is best to include it in S_p in the following manner:

$$S_p = (-K/T_P) \tag{1.14}$$

where T_P is from previous iteration (for Gauss-Seidel method, see next section).

1.4 Gauss - Seidel Numerical Scheme

In the equation 1.11, T_P can be estimated only if T_E, T_W, T_N, T_S are known. However, given sufficient boundary conditions, we can write the equations in the form:

$$AT = B$$

where A is a matrix of the coefficients of unknown temperatures T. This can be solved using matrix inversion, but it is computationally intensive and not a feasible option. Hence we adopt numerical iteration methods to arrive at a solution. The two most popular methods are the Jacobi and Gauss-Seidel Numerical Schemes. The Jacobi method utilises the initial values only to compute the unknown values, while, Gauss-Seidel method uses the values from the previous iterations to compute the unknown values in the current iteration. The Jacobi approach has lesser chance of obtaining a diverging solution, but results in slower convergence. The Gauss-Seidel method has more chances of obtaining a diverging solution, but the convergence is faster. We will use the Gauss-Seidel method.

Chapter 2

Results

The following results are obtained for the domain and boundary conditions as shown in figure 2.1.

2.1 Equidistant Mesh

For studying the case of equidistant mesh and studying mesh convergence, we look at 4 different meshes - 5×5, 10×10, 20×20, 40×40. The meshes are as shown below:

Solving the problem using these different meshes, we get contour plots of temperature over the 2D domain.

We can see that the contours are different for a 5×5 mesh than the rest, and the contours become smoother and converge to a single solution as the mesh becomes finer.

Now, we look at the temperature distribution at $y = 0.3$ along x direction and compare them for different mesh sizes (Figure 2.4).

We can see clearly from Figure 2.4 that the solution does not converge for 5×5 mesh, due to lack of sufficient number of nodal points. As the mesh is made finer, we get mesh convergence (20×20 and 40×40 meshes have overlapping results). Thus we can say that the minimum size of an equidistant mesh for convergence is 20×20.

We also plot the residual error as a function of the number of iterations in

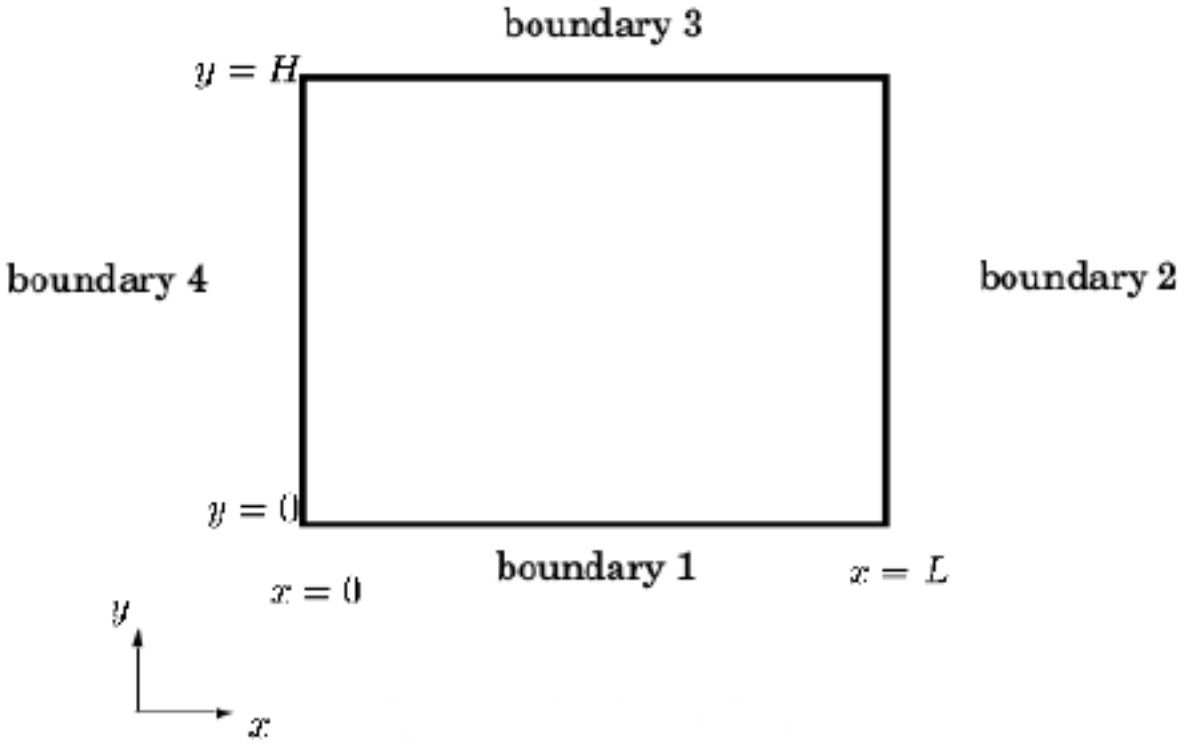


Figure 1: Computational domain.

T1	T2	T3
15	$-5y/H + 15cos(2\pi y/H)$	10

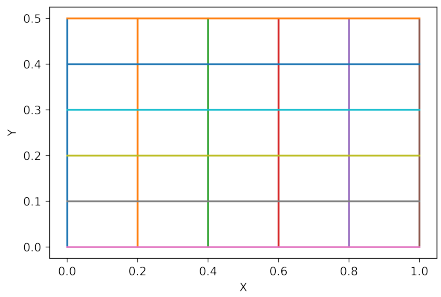
Figure 2.1: Domain and Boundary conditions

all 4 cases, considering an error tolerance of 10^{-5}. The error is calculated as:

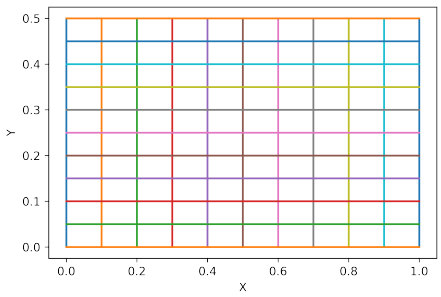
$$R = \frac{|\sum_N (T_i - T_{i-1})|}{N} \tag{2.1}$$

and must be less than the tolerance.

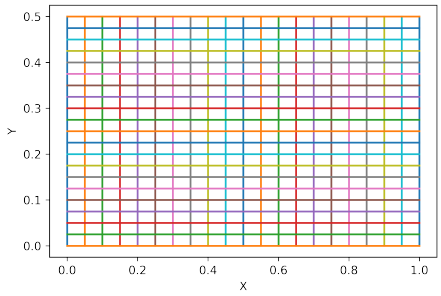
We can see from the figure that even though the convergence is faster for



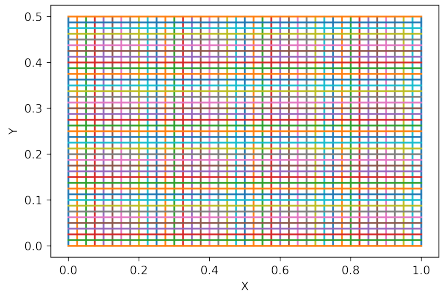
(a) 5×5 mesh



(b) 10×10 mesh



(c) 20×20 mesh



(d) 40×40 mesh

coarser mesh due to lesser number of nodal points, the initial errors incurred are much higher. Hence choosing a mesh of the right size is important for accurate results with the lower number of iterations until convergence.

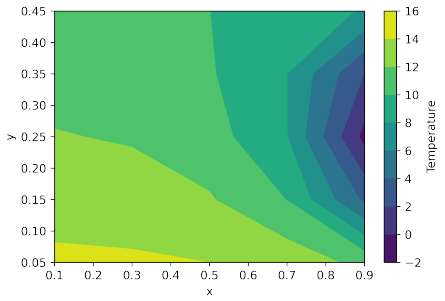
The effect of changing the error tolerance ϵ can be observed by plotting the number of iterations until convergence for various values of ϵ on a log-log plot (Figure 2.6).

We can see that roughly, the variation of $\log(N)$ with $\log(\epsilon)$ is linear with a negative slope, or in turn, N decreases exponentially with the increase in ϵ. This means that the number of iterations increases as the error tolerance becomes smaller, which is as expected. We also plot the variation of temperature as a function of x at $y = 0.3$ for various error tolerances.

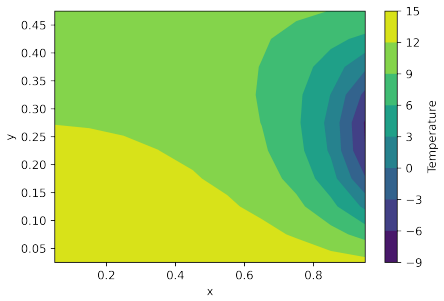
We can see from figure 2.7 that as the value of ϵ is decreased, the solution converges to the correct value.

The heat flux vector at every boundary is also plotted over the domain for various mesh sizes.

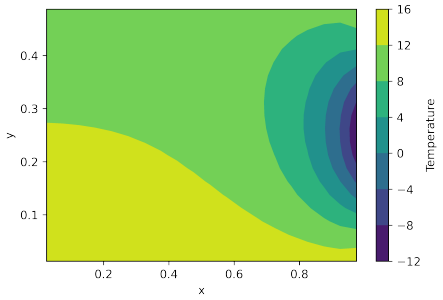
Heat flux vector plots become independent of mesh size once mesh indepen-



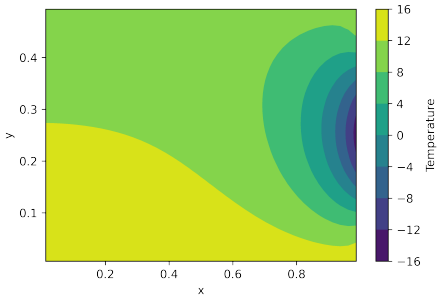
(a) Contour Plot of Temperature for 5×5 mesh



(b) Contour Plot of Temperature for 10×10 mesh



(c) Contour Plot of Temperature for 20×20 mesh



(d) Contour Plot of Temperature for 40×40 mesh

dence is achieved.

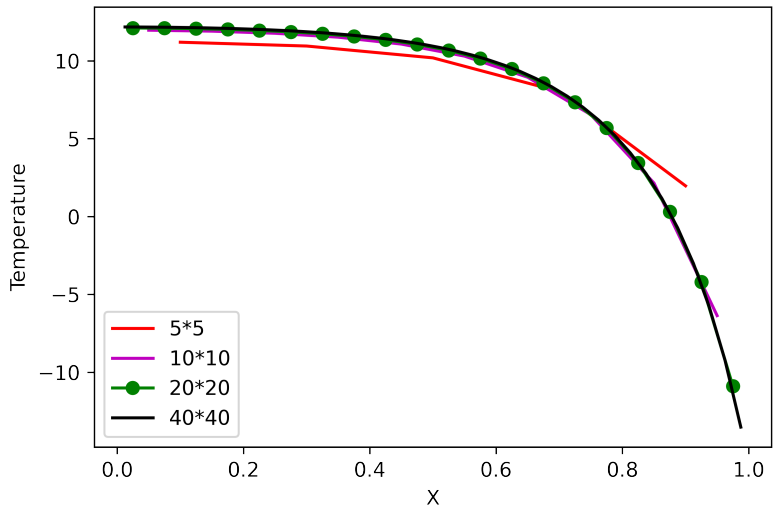
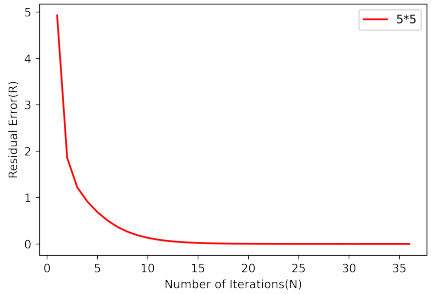
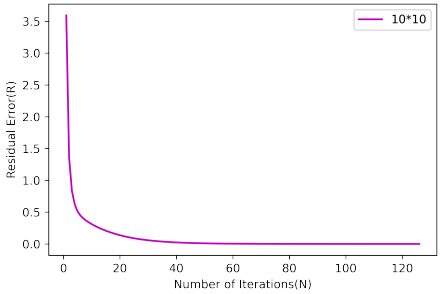


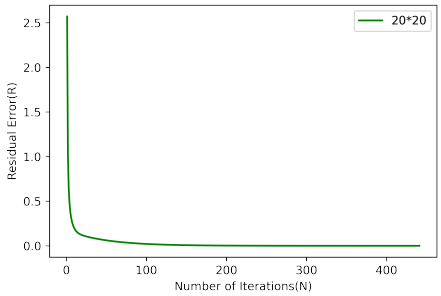
Figure 2.4: Temperature as a function of x at y = 0.3 for various mesh sizes



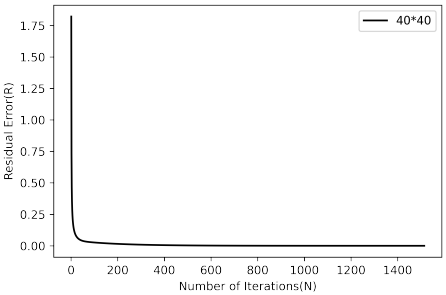
(a) Residual error R vs number of iterations N for 5×5 mesh



(b) Residual error R vs number of iterations N for 10×10 mesh



(c) Residual error R vs number of iterations N for 20×20 mesh



(d) Residual error R vs number of iterations N for 40×40 mesh

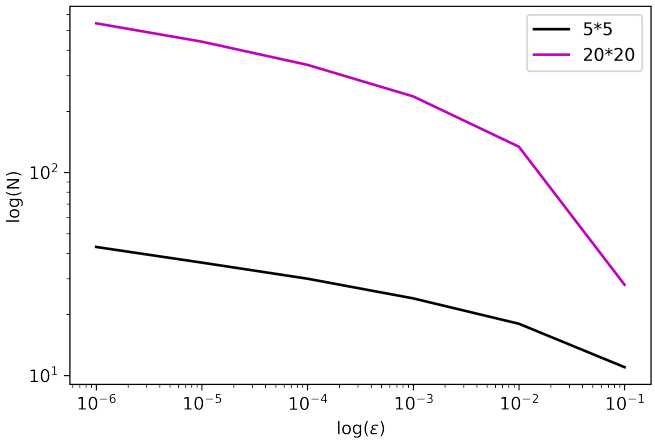


Figure 2.6: Plot of $\log(N)$ vs $\log(\epsilon)$ for various mesh sizes

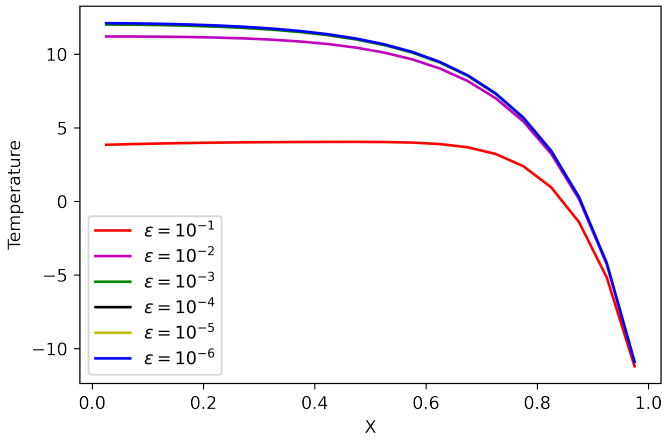
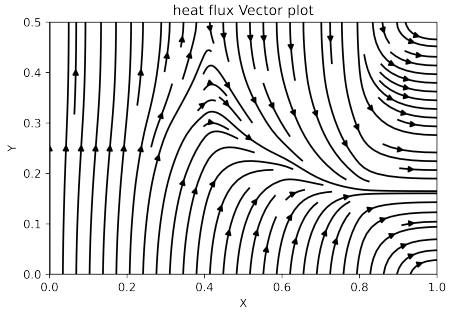
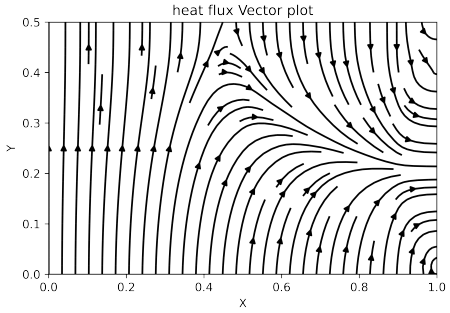


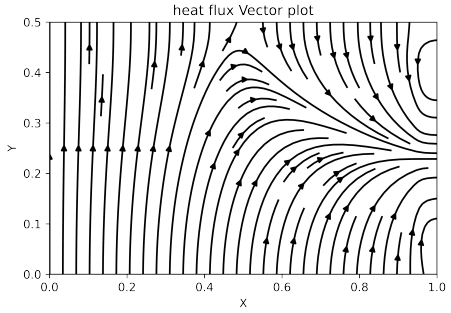
Figure 2.7: Plot of Temperature vs x at $y = 0.3$ for 20×20 mesh as error tolerance is varied



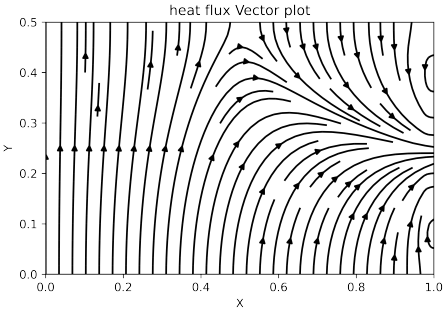
(a) Heat Flux Vector Plot for 5×5 mesh



(b) Heat Flux Vector Plot for 10×10 mesh

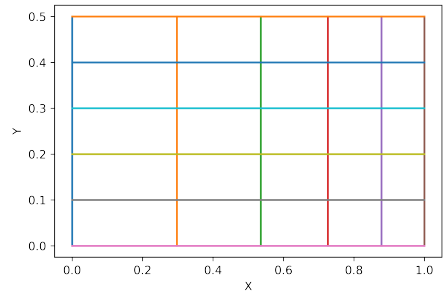


(c) Heat Flux Vector Plot for 20×20 mesh

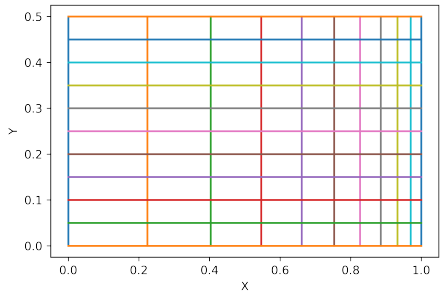


(d) Heat Flux Vector Plot for 40×40 mesh

2.2 Stretched Meshes



(a) Stretched 5×5 mesh $SF_x = 1.1$

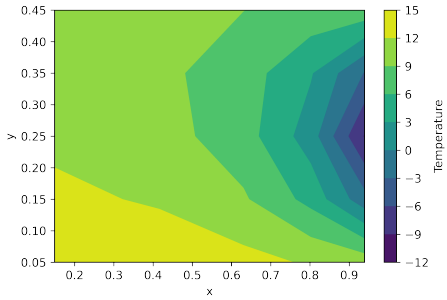


(b) Stretched 5×5 mesh $SF_x = 1.1$

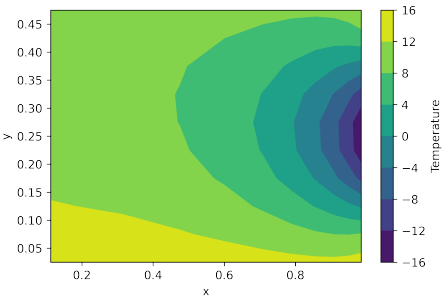
It can often be advantageous to use a stretched mesh over an equidistant one when the gradients are concentrated onto one side (which is the case in the current problem, as seen in the contour plots), as we can stretch the mesh to use more number of cells in the region where there are larger gradients. Define the stretch factor as:

$$SF_x = \frac{0.5\Delta x}{\delta x_e}$$

and similarly for y direction also. Here, we use a stretch factor $SF_x = 1.1$ and $SF_y = 1$ (since gradients are more pronounced along the x-axis towards the right, and smaller gradients are present along y). The stretched mesh is shown.



(a) Contour plot of temperature for stretched 5×5 mesh $SF_x = 1.1$



(b) Contour plot of temperature for stretched 10×10 mesh $SF_x = 1.1$

The contour plots for the stretched 5×5 mesh and 10×10 mesh are shown.

Here we can see that the contour plot of temperature for stretched 5×5 mesh $SF_x = 1.1$ is different from the one for 10×10 mesh due to too less number of nodal points. We also plot the temperature along x at $y = 0.3$ for various stretched meshes and compare them to the case without stretching in figure 2.11.

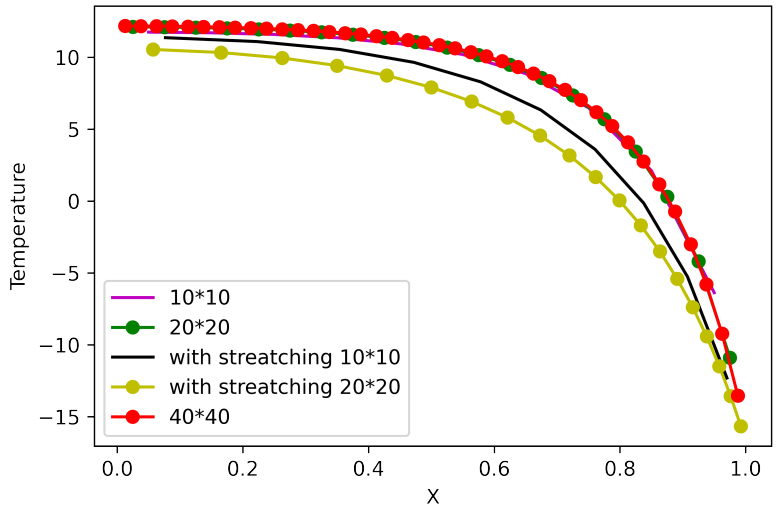
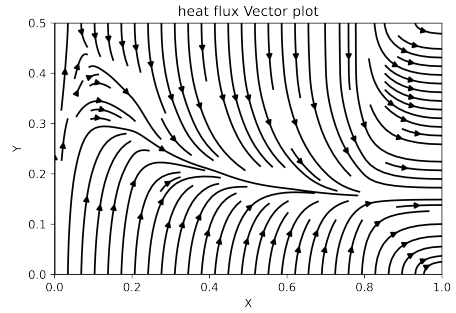


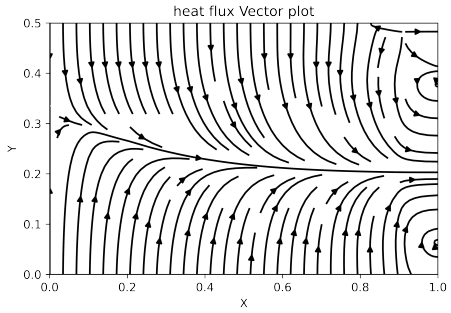
Figure 2.11: Temperature along x at $y = 0.3$ for various meshes with and without stretching

In figure 2.11, we can see that the solutions given by stretched and unstretched meshes are significantly different. This can be understood by noting the fact that for any mesh, we can expect convergence as mesh becomes finer; we can see that as the stretched mesh becomes finer, the solution moves away from the converged solution of the unstretched case. Mesh convergence can be achieved if we use finer unstretched meshes, but the final solutions for both types of meshes are different.

The heat flux plot for various stretched meshes are shown.

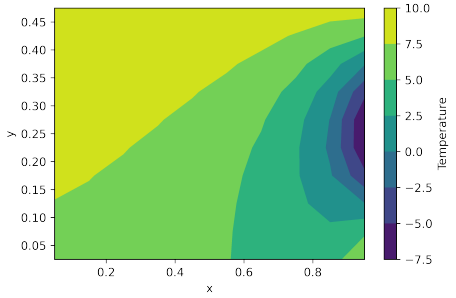


(a) Heat flux vector plot for stretched 5×5 mesh $SF_x = 1.1$

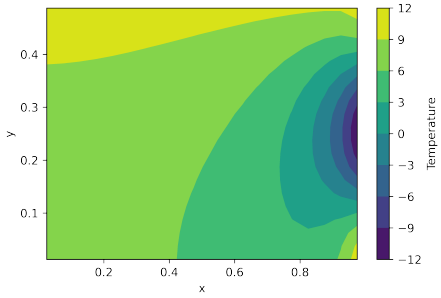


(b) Heat flux vector plot for stretched 10×10 mesh $SF_x = 1.1$

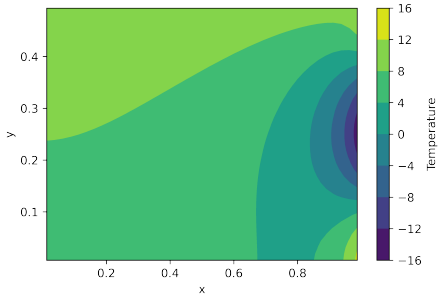
2.3 Effect of Boundary Conditions



(a) Contour Plot with Neumann conditions on 10×10 mesh



(b) Contour Plot with Neumann conditions on 20×20 mesh



(c) Contour Plot with Neumann conditions on 40×40 mesh

We now study the effect of changing the boundary conditions on boundary 1 from Dirichlet to Neumann. The contour plots for various mesh sizes are shown in figure.

The solutions can be observed to be significantly different from the case with Dirichlet boundary conditions. The plot of the temperature distribution along x for $y = 0.3$ can be seen from figure2.14.

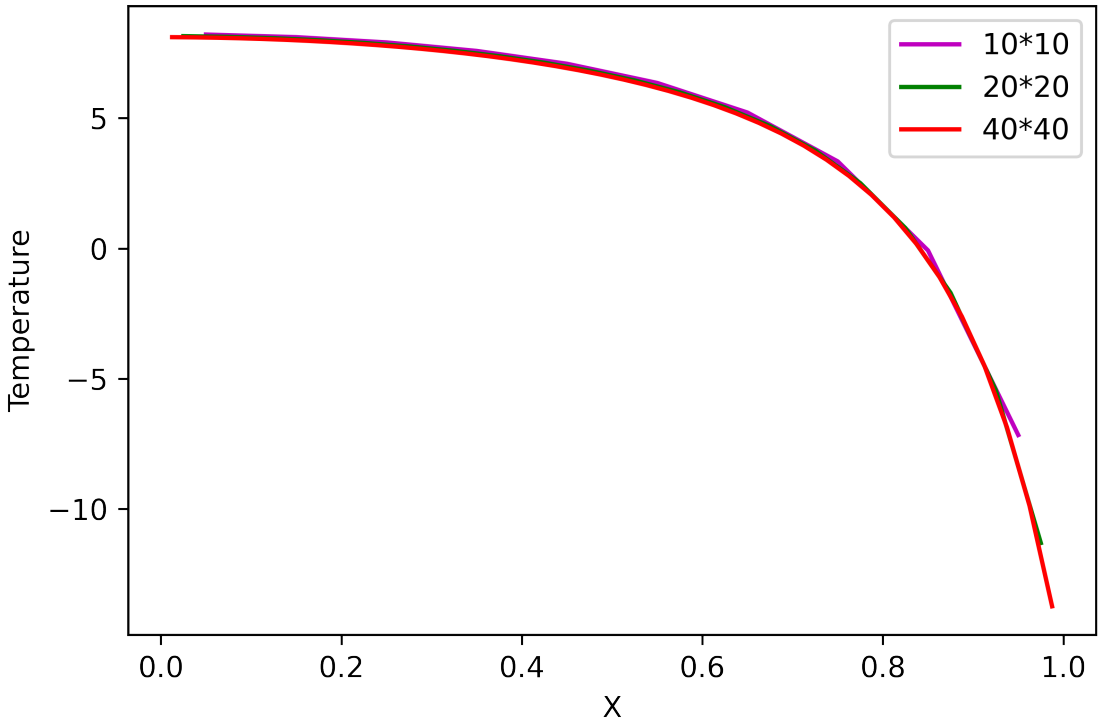


Figure 2.14: temperature distribution along x for $y = 0.3$ for various mesh sizes with Neumann BC

Here we can see that solutions for all mesh sizes overlap, indicating convergence at 10×10 mesh size itself. For 20×20 mesh, we compare the Dirichlet and Neumann solutions.

The heat flux vector plots are also plotted for the same cases.

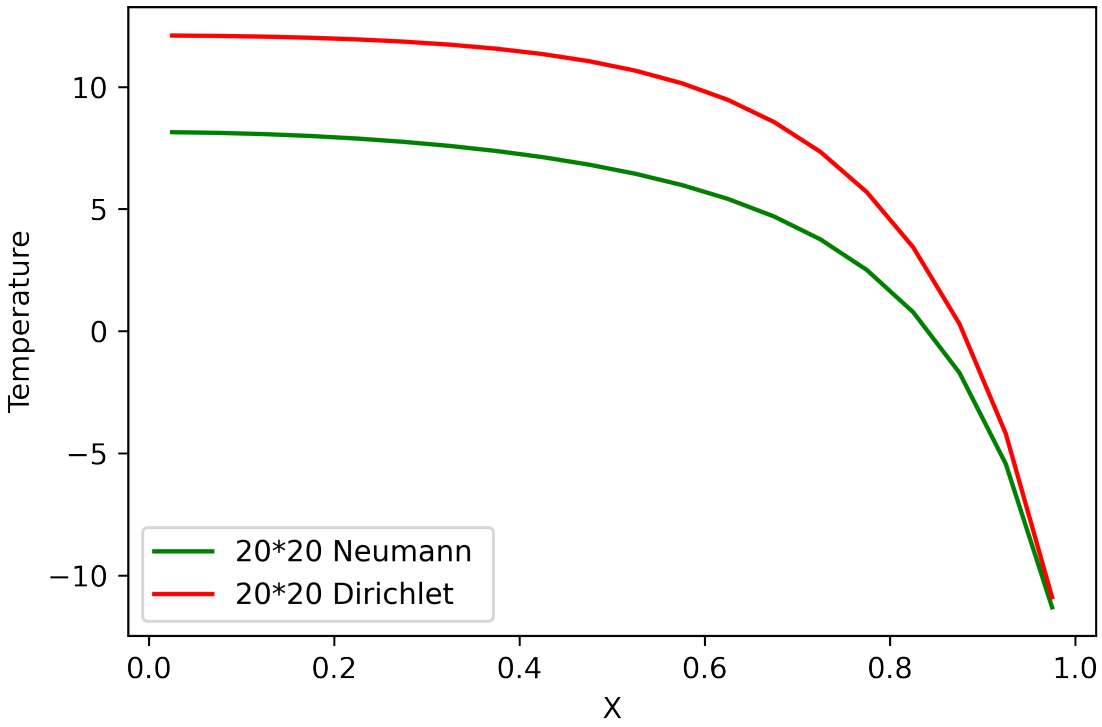
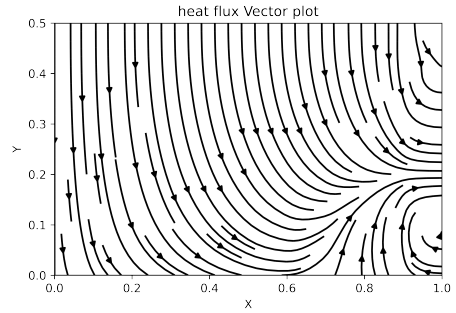
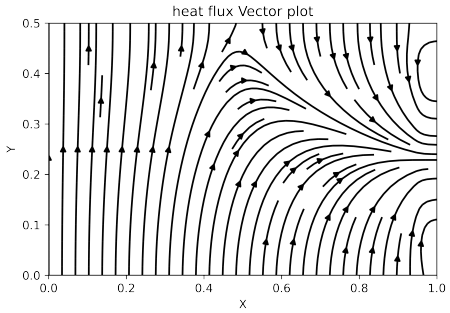


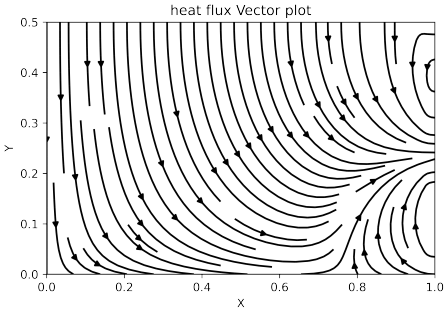
Figure 2.15: Dirichlet and Neumann solutions for 20×20 mesh



(a) Heat flux vector Plot with Neumann conditions on 10×10 mesh



(b) Heat flux vector Plot with Neumann conditions on 20×20 mesh



(c) Heat flux vector Plot with Neumann conditions on 40×40 mesh

2.4 Conclusions

We have analysed the problem of 2D heat conduction, under the variation of mesh size, stretch factor, error tolerance and boundary conditions. For equidistant mesh, we can see that the contours are different for a 5×5 mesh than the rest, and the contours become smoother and converge to a single solution as the mesh becomes finer. The solution does not converge for 5×5 mesh, due to lack of sufficient number of nodal points. As the mesh is made finer, we get mesh convergence (20×20 and 40×40 meshes have overlapping results). Thus we can say that the minimum size of an equidistant mesh for convergence is 20×20. We can see that even though the convergence is faster for coarser mesh due to lesser number of nodal points, the initial errors incurred are much higher. Hence choosing a mesh of the right size is important for accurate results with the lower number of iterations until convergence. The effect of changing the error tolerance can be observed by plotting the number of iterations until convergence for various values of ϵ on a log-log plot. We can see that roughly, the variation of log(N) with log(ϵ) is linear with a negative slope. This means that the number of iterations increases as the error tolerance becomes smaller, which is as expected. We also plot the variation of temperature as a function of x at $y = 0.3$ for various error tolerances. As the value of ϵ is decreased, the solution converges to the correct value. One can also use a stretched mesh over an equidistant one when the gradients are concentrated onto one side (which is the case in the current problem, as seen in the contour plots), as we can have more number of cells in the region where there are larger gradients.

Here, we use a stretch factor $SFx = 1.1$ and $SFy = 1$. We can see that the contour plot of temperature for stretched 5×5 mesh is different from the one for 10×10 mesh due to too less number of nodal points. We can see that the solutions given by stretched and unstretched meshes are significantly different. This can be understood by noting the fact that for any mesh, we can expect convergence as mesh becomes finer; we can see that as the stretched mesh becomes finer, the solution moves away from the converged solution of the unstretched case. We also investigated the effect of changing the boundary conditions on boundary 1 from Dirichlet to Neumann. The solutions can be observed to be significantly different from the case with Dirichlet boundary

conditions. We can see that solutions for all mesh sizes overlap, indicating convergence at 10×10 mesh size itself.
Overall, we can see that the mesh sizing, stretching, error tolerance and boundary conditions play a significant role in the type of solution and convergence of solutions. A CFD engineer must keep these variations in mind while studying similar diffusion problems.

Bibliography

[1] H. K. Versteeg and W. Malalasekera. *An introduction to computational fluid dynamics: the finite volume method*. 2nd ed. OCLC: ocm76821177. Harlow, England ; New York: Pearson Education Ltd, 2007. ISBN: 9780131274983.

[2] Vagesh D Narasimhamurthy. *AM5630 Foundations of Computational Fluid Dynamics Course*. English. Jul - Nov 2022. AM5630 Foundations of Computational Fluid Dynamics Course AM5630. Chennai, 2022.