Computer Exercise

Noel Efrem

10 oktober 2023

Innehåll

1 Part 1 - Finite difference approximation in 1D

1.1 Introduction

In this exercise, we consider a short cylinder containing a fluid with high thermal diffusivity that moves at a slow pace. The fluid undergoes heating in a specific section, leading to the convection of heat along the cylinder. The steady-state temperature distribution, $T(z)$, within the fluid is governed by the convection-diffusion equation given by:

$$-\frac{d^2T}{dz^2} + v\frac{dT}{dz} = Q(z) \tag{1}$$

where $T(z)$ represents the temperature distribution along the length of the cylinder, v is the fluid velocity, and $Q(z)$ is the heat source function.
The heat source, $Q(z)$, is defined as:

$$Q(z) = \begin{cases} 0, & 0 \leq z < a, \\ Q_0 \sin\left(\frac{(z-a)\pi}{b-a}\right), & a \leq z \leq b, \\ 0, & b < z \leq 1. \end{cases} \tag{2}$$

The boundary conditions for this problem are set as follows: At the inlet, $z = 0$, the fluid has a known temperature T_0:

$$T(0) = T_0 \tag{3}$$

At the outlet, $z = 1$, the fluid loses heat to the surroundings, which are at temperature T_{out}. This is modeled by the boundary condition:

$$-\frac{dT(1)}{dz} = \alpha(v)(T(1) - Tout) \tag{4}$$

where $\alpha(v)$ is a function of the fluid velocity, given by:

$$\alpha(v) = \sqrt{\frac{v^2}{4} + \alpha_0^2} - \frac{v}{2} \tag{5}$$

and α_0 is the heat transfer coefficient for the non-convective case.
The objective is to determine the temperature distribution $T(z)$ within the cylinder using the convection-diffusion equation (1), the heat source function (2), and the boundary conditions (3) and (4).

1.2 Method

To solve the problem numerically, we employ the finite difference method. The convection-diffusion equation can be discretised using central difference for both the first and second derivatives.
Starting with the differential equation (1): Using central difference approximation for both derivatives:

$$-\frac{d^2T}{dz^2} \approx \frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta z^2} + \mathcal{O}(h^2) \tag{6}$$

Substituting these approximations into the PDE, we get:

$$-\frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta z^2} + v\frac{T_{i+1} - T_{i-1}}{2\Delta z} = Q_i \tag{7}$$

in which $\mathcal{O}(h^2)$ is omitted as a numerical error.
Rearranging terms:

$$\left(-\frac{1}{\Delta z^2} - \frac{v}{2\Delta z}\right) T_{i-1} + \frac{2}{\Delta z^2} T_i + \left(-\frac{1}{\Delta z^2} + \frac{v}{2\Delta z}\right) T_{i+1} = Q_i$$

From this equation, we can identify the coefficients for the matrix representation:

$$a_i = -\frac{1}{\Delta z^2} - \frac{v}{2\Delta z}$$

$$b_i = \frac{2}{\Delta z^2}$$

$$c_i = -\frac{1}{\Delta z^2} + \frac{v}{2\Delta z}$$

The boundary conditions can be discretised using central difference as well: At $z = 1$:

$$-\frac{T_{N+1} - T_{N-1}}{2\Delta z} = \alpha(v)(T_N - Tout)$$

From this, the ghost point T_{N+1} can be expressed in terms of known quantities as following.
Rearranging for the ghost point T_{N+1}:

$$T_{N+1} = T_{N-1} - 2\Delta z\alpha(v)(T_N - Tout) \tag{8}$$

Substituting this expression for T_{N+1} into the discretised equation for $i = N$, we get:

$$a_N T_{N-1} + b_N T_N + c_N(T_{N-1} - 2\Delta z\alpha(v)(T_N - Tout)) = Q_N \tag{9}$$

Expanding and collecting like terms:

$$(a_N + c_N)T_{N-1} + (b_N - 2\Delta z\alpha(v)c_N)T_N = Q_N + 2\Delta z\alpha(v)c_N Tout \tag{10}$$

Defining:

$$b' = b - 2\Delta z\alpha(v)c_N$$

$$Q'_N = Q_N + 2\Delta z\alpha(v)c_N Tout$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & b & c & 0 & \dots & 0 \\ 0 & a & b & c & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & a & b' \end{bmatrix} \begin{bmatrix} T_0 \\ T_1 \\ T_2 \\ \vdots \\ T_N \end{bmatrix} = \begin{bmatrix} T_0 \\ Q_1 - aT_0 \\ Q_2 \\ \vdots \\ Q'_N \end{bmatrix} \tag{11}$$

1.3 Results

1.3.1 Exercise a)

The focus on the first part is on understanding the impact of grid resolution on the accuracy and stability of the solution. By discretising the z-interval [0, 1] with varying grid spacings, we aim to observe the convergence behavior of the method. Specifically, we'll be examining the solutions obtained for different values of N = [10, 20, 40, 80]. The results for the various grid spacings are presented here below:

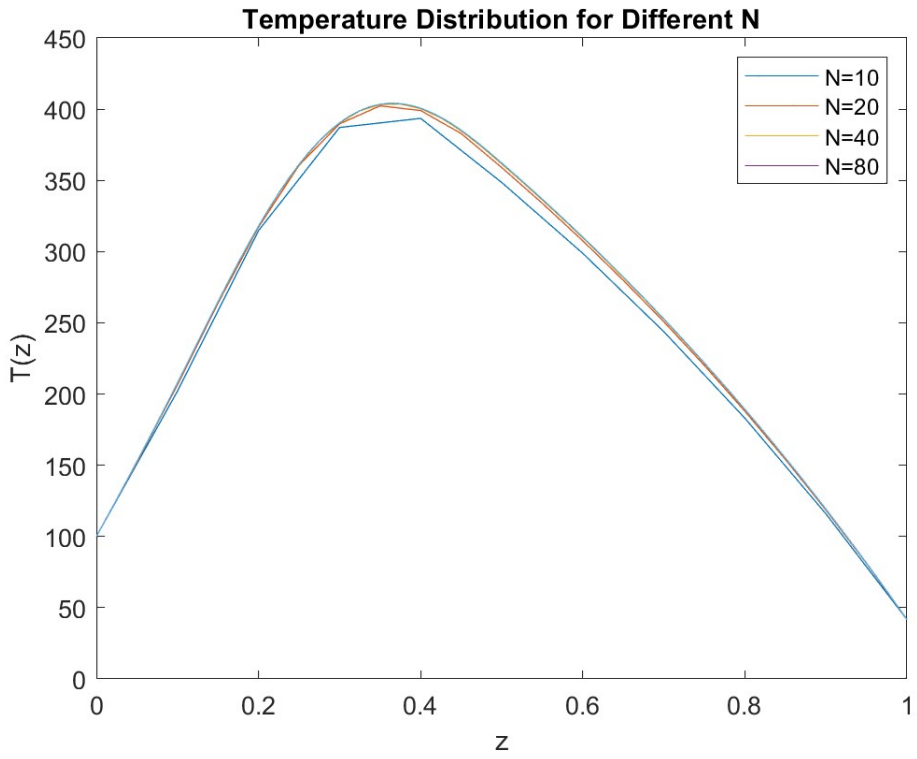

Figure 1: Temperature Distribution along the z-coordinate for grid-spacing N=[10, 20, 40, 80]

From our results, the first thing that can be noticed is the notable temperature variation between $z = 0.1$ (denoted as a) and $z = 0.4$ (denoted as b), corresponding to the positioning of the copper wire within the pipe.

For instance, with a step-size such as $N = 10$, the derived temperature curve appears irregular, suggesting a somewhat inaccurate. As the discretisation becomes finer, the deduced temperature exhibits a slight increment, around the peak temperature at z=0.4. This can be attributed to the fact that at the regions of fast rate of change (at the peaks), a finer grid spacing is crucial in evaluating the peak temperatures, thereby yielding a more accurate result. The underestimating behaviour of the temperature, especially around the peak, can be directly attributed to the central difference approximation. Using points with substantial spacing, in the coarser grids, tends to produce diminished temperature values which is

heavily accentuating in the extreme points, as the points in this region differ in value by a lot, thus failing to represent the actual curvature of the temperature distribution.

As requested in the instructions, the temperatures at z = 0.5 for various grid spacings is as follows:

N	**Temperature**
80	262.7
160	260.8
320	259.7

Table 1: Temperature at z=0.5 for various grid spacings

To validate the accuracy of our numerical solution, we tested the convergence rate of the method. The order of convergence, p, is a measure of how the error in the numerical solution decreases as the grid spacing, Δz, is halved. For second-order methods, we expect p to be approximately 2.
For our test, we focused on the temperature value at $z = 0.5$ for consecutive grid resolutions N = [10, 20, 40, 80, 160]. The order of convergence p between two consecutive resolutions was computed using the formula:

$$p = \frac{\log(E_2/E_1)}{\log(h_2/h_1)} \tag{12}$$

Where E_1 and E_2 are the absolute differences in the temperature at $z = 0.5$ for two consecutive resolutions, and h_1 and h_2 are their respective grid spacings. For instance, $E_1 = T_{10} - T_{20}$ and $E_2 = T_{20} - T_{40}$ and for the grid spacings $h_1 = \frac{1}{10}$ and $h_2 = \frac{1}{20}$, in which the subscripts for T indicate the N. The results were of the order 2 to the second decimal. For detailed implementation and result, check attached MATLAB code.

As instructed, our discretisation manifests second-order precision. This is the result of solely using the central difference approximation, which is intrinsically second-order precise for derivative approximations as can be seen in Eq6 and Eq7.

1.3.2 Exercise b)

The primary objective of this part is to comprehend the influence of fluid velocity on the temperature distribution within the cylinder. By varying the fluid velocity, we aim to observe its effect on the heat transfer dynamics, especially in the context of the convection-diffusion equation. Specifically, we'll be examining the solutions obtained for different velocities $v = [1, 5, 15, 100]$. The results for the various velocities are presented below:

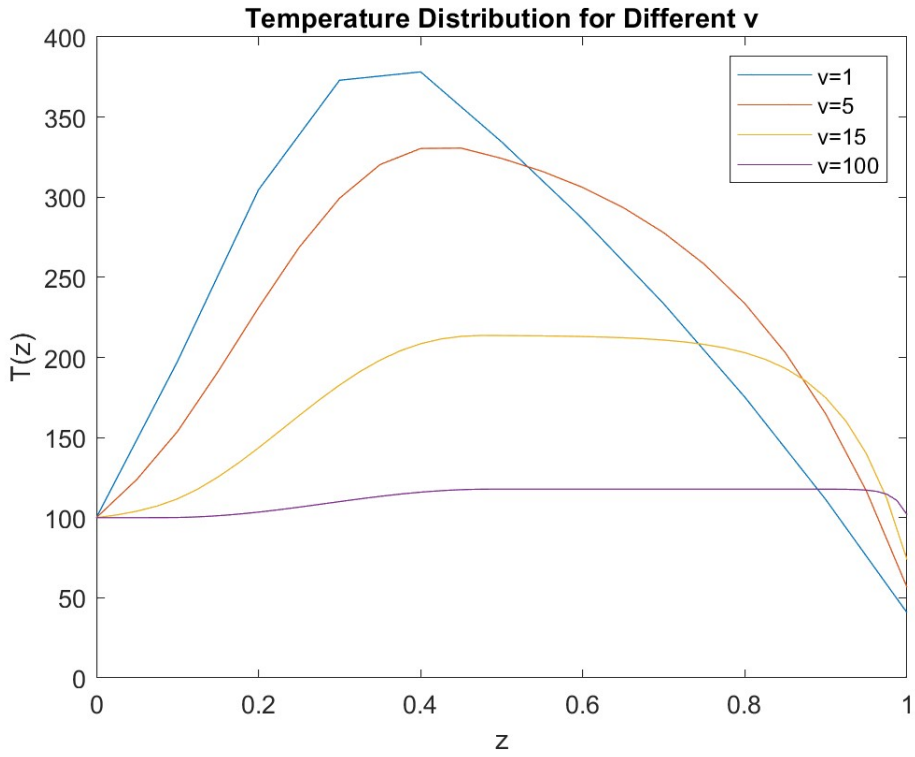

Figure 2: Temperature Distribution along the z-coordinate for velocities $v = [1, 5, 15, 100]$

The results are in line with our expectations. It's evident that the fluid velocity plays a pivotal role in determining the temperature distribution within the cylinder. At lower velocities, the fluid remains in contact with the heat source (copper wire) for a more extended period, allowing for more heat to be absorbed. This phenomenon is evident from the higher temperature values around $z = 0.1$ to $z = 0.4$ for smaller velocities.
However, as the fluid velocity increases, the time of contact with the heat source reduces, leading to a lesser amount of heat being absorbed. This is reflected in the lower peak temperatures for higher velocities. Moreover, at higher velocities, the rate of heat loss to the surroundings is also reduced, leading to a more uniform temperature distribution after $z = 0.4$.

2 Part 2 - Finite difference approximation in 2D

Introduction In this report, we study the heat conduction through a 2D rectangular metal block. The region occupied by the block is denoted by $\Omega = \{0 < x < L_x, 0 < y < L_y\}$ in the xy-plane. The bottom boundary of the block, at $y = 0$, is maintained at a temperature T_{ext}, which is the same as the surrounding air. The other three sides of the block are insulated. An external source, modeled by the function $f(x, y)$, heats the block. The temperature distribution $T(x, y)$ within the block is governed by the elliptic partial differential equation (PDE) and associated boundary conditions as described below.

2.1 Governing Equations

The temperature distribution within the block is governed by the following elliptic PDE:

$$-\Delta T = f, \quad (x, y) \in \Omega \tag{13}$$

with the boundary conditions:

$$T(x, 0) = T_{ext}, \quad 0 < x < L_x, \tag{14}$$

$$\frac{\partial T}{\partial x}(0, y) = 0, \quad 0 < y < L_y, \tag{15}$$

$$\frac{\partial T}{\partial x}(L_x, y) = 0, \quad 0 < y < L_y, \tag{16}$$

$$\frac{\partial T}{\partial y}(x, L_y) = 0, \quad 0 < x < L_x. \tag{17}$$

in which $T_{ext} = 25, L_x = 12$ and $L_y = 5$.

2.2 Method

To solve the above PDE numerically, we employ the finite difference method. The domain Ω is discretised into a quadratic mesh with a uniform step size h. The step size is defined as $h = \frac{L_x}{N} = \frac{L_y}{M}$, where N and M are the number of grid points in the x and y directions, respectively.

2.2.1 Discretisation

Using central difference approximations, the 2D Laplacian at a grid point (i, j) can be represented as:

$$\Delta T_{i,j} \approx \frac{-4T_{i,j} + T_{i+1,j} + T_{i-1,j} + T_{i,j+1} + T_{i,j-1}}{h^2} = f_{i,j} + \mathcal{O}(h^2) \tag{18}$$

in which $\mathcal{O}(h^2)$ is omitted and regarded as a numerical error of order 2.

The boundary conditions for the problem are as follows:

1. Dirichlet boundary condition at the bottom boundary:

$$T(x, 0) = T_{ext}, \quad 0 < x < L_x$$

Using the finite difference method, this condition can be directly implemented in the system of equations. For a grid point i at $y = 0$, the equation becomes:

$$T_{i,0} = T_{ext} \tag{19}$$

2. Neumann boundary condition at the left boundary:

$$\frac{\partial T}{\partial x}(0, y) = 0, \quad 0 < y < L_y$$

Using a central difference approximation at the left boundary, this condition becomes:

$$T_{0,j} = T_{2,j} \tag{20}$$

3. Neumann boundary condition at the right boundary:

$$\frac{\partial T}{\partial x}(L_x, y) = 0, \quad 0 < y < L_y$$

Using a central difference approximation at the right boundary, this condition becomes:

$$T_{N+1,j} = T_{N-1,j} \tag{21}$$

4. Neumann boundary condition at the top boundary:

$$\frac{\partial T}{\partial y}(x, L_y) = 0, \quad 0 < x < L_x$$

Using a central difference approximation at the top boundary, this condition becomes:

$$T_{i,M+1} = T_{i,M-1} \tag{22}$$

These discretised boundary conditions will be incorporated into the system of equations to ensure the solution satisfies the physical constraints at the boundaries of the domain.

2.2.2 Matrix Representation

To solve the system of equations arising from the discretised Laplacian, we represent it in matrix form. Given the 2D nature of our problem, we need a strategy to represent this system in a 1D vector form. We achieve this by employing a column-major ordering. In column-major ordering, we traverse the grid column by column. For a 3×3 grid, the ordering is as follows:

$$\begin{aligned}
T_{1,1} &\rightarrow T_1 \\
T_{2,1} &\rightarrow T_2 \\
T_{3,1} &\rightarrow T_3 \\
T_{1,2} &\rightarrow T_4 \\
T_{2,2} &\rightarrow T_5 \\
T_{3,2} &\rightarrow T_6 \\
T_{1,3} &\rightarrow T_7 \\
T_{2,3} &\rightarrow T_8 \\
T_{3,3} &\rightarrow T_9
\end{aligned}$$

For instance, considering the point $T_{1,1}$ (or T_1 in our ordering), the discretised Laplacian equation in y becomes:

$$\frac{-T_{1,0} + 2T_{1,1} + T_{1,2}}{h^2} = f_{1,1}$$

Here, the term corresponding to $T_{1,0}$ is handled by Eq. 19 in which the Dirichlet BC states that this is equal to T_{ext}. This is then moved to left side as a
Still considering the point $T_{1,1}$ (or T_1 in our ordering), the discretised Laplacian equation in x becomes:

$$\frac{-T_{0,1} + 2T_{1,1} + T_{2,1}}{h^2} = f_{1,1}$$

Here, the term corresponding to $T_{0,1}$ is handled using Eq. 20, the Neumann BC on the left side, resulting in $T_{0,1} = T_{2,1}$.

Therefore, the matrices A_x and A_y are tridiagonal matrices. Their main diagonal is populated with 2's, while the sub-diagonal and super-diagonal contain -1's. This structure arises from the second-order central difference scheme.

2.2.3 Constructing the Matrix A

Instead of manually constructing the system matrix for each grid point, we use matrices A_x and A_y which correspond to the discretised Laplacian in the x and y directions, respectively. The global matrix A representing the 2D Laplacian over the entire domain is then constructed using the Kronecker product:

$$A = I_y \otimes A_x + A_y \otimes I_x \tag{23}$$

where I_x and I_y are identity matrices of size $N \times N$ and $M \times M$, respectively. This matrix A encapsulates the effects of the differential operators in both the x and y directions and can be used to set up and solve the discrete version of the PDE.

2.2.4 Constructing the Vector b

The vector b represents the non-homogeneous term in our system of equations. It's initialized with the function values, representing the source term in our PDE. Additionally, the Dirichlet condition at the bottom boundary is incorporated into the vector b to account for the external temperature T_{ext}. Since column major ordering is used, the implementation of the Dirichlet boundary entails adding $\frac{T_{ext}}{h^2}$ to the first N elements. These are the elements corresponding to the $(i, j) = (i, 1)$ FDM equations, which are the equations that include the ghost point $T(i, 0$, which by Eq. 19 is equal to T_{ext}

2.2.5 Sparse Representation

For large values of N and M, the matrix A is sparse, meaning most of its entries are zero. Using a sparse matrix representation can save significant memory and computational effort. In MATLAB, the 'spdiags' function can be used to construct these matrices in a sparse format.

2.3 Results

2.3.1 Exercise a) - Computation and Visualization of $T(x, y)$

Given the constant $f \equiv 2$ and a step size $h = 0.2$ (yielding $N = 60$), we compute the solution $T(x, y)$ for the provided PDE.

To visualize the solution, we employ the `mesh` function in MATLAB. This provides a 3D representation of the temperature distribution across the domain.

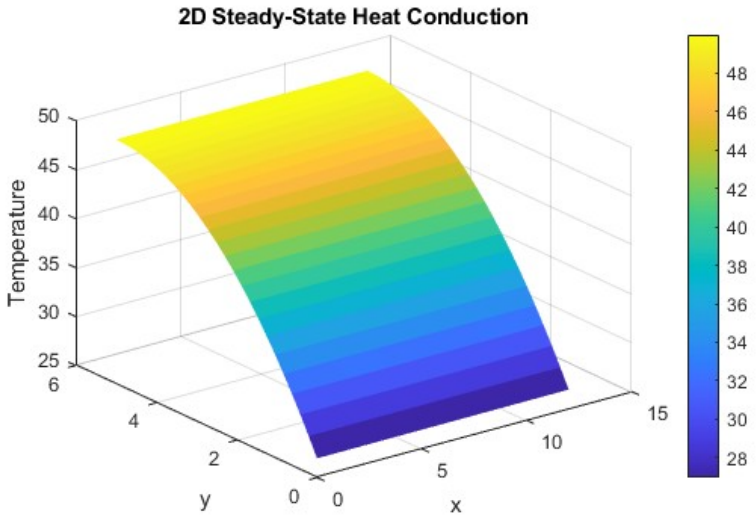

Figure 3: 3D mesh plot of the temperature distribution $T(x, y)$.

From the computed solution, the temperature value at the point $(x, y) = (6, 2)$ inside the block is 42.16.

The obtained results align with our expectations. Given that only the bottom part of the block is not insulated, while all other edges are, the temperature distribution reflects this setup. Since the heat source is constant everywhere, the top boundary, being the furthest from the surrounding air (at y=0)), exhibits the highest temperature, as evident in the 3D plot (Figure 3). However, the solution did not demonstrate a second order of accuracy, which is inconsistent with our use of ghost points and second-order derivative approximations throughout the computation. The same method here as Section 1.3.1 was used to test convergence numerically with the help of Eq12.

2.3.2 Exercise b) Determining the Exact Solution T(x,y)

To find the exact solution for the case when f is constant, we can utilize the provided PDE and boundary conditions.

Given the PDE:

$$-\Delta T = f \tag{24}$$

where Δ represents the Laplacian operator. In a 2D scenario, the Laplacian is expressed as:

$$\Delta T = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \tag{25}$$

For the proposed solution form $T(x, y) = c_0 + c_1 y + c_2 y^2$:
1. The second partial derivative with respect to x is zero since T is independent of x:

$$\frac{\partial^2 T}{\partial x^2} = 0 \tag{26}$$

2. The second partial derivative with respect to y yields:

$$\frac{\partial^2 T}{\partial y^2} = 2c_2 \tag{27}$$

Substituting these derivatives into the PDE, we obtain:

$$-2c_2 = f \tag{28}$$

which implies:

$$c_2 = -\frac{f}{2} = -1 \tag{29}$$

Utilizing the Dirichlet boundary condition:

$$T(x, 0) = T_{ext} \tag{30}$$

and substituting $y = 0$ into our solution form, we deduce:

$$c_0 = T_{ext} = 25 \tag{31}$$

The Neumann boundary conditions on the left and right sides are satisfied for any polynomial in y due to the absence of x dependence. Therefore, they don't offer additional constraints for the coefficients.
However, the Neumann boundary condition at the top is:

$$\frac{\partial T}{\partial y}(x, L_y) = 0 \tag{32}$$

Differentiating our solution with respect to y, we get:

$$\frac{\partial T}{\partial y} = c_1 + 2c_2 y = c_1 - 2y \tag{33}$$

Setting this to zero at $y = L_y = 5$, we can solve for c_1:

$$c_1 - 2(5) = 0 \implies c_1 = 10 \tag{34}$$

In conclusion, we obtain the following second degree polynomial:

$$T(x, y) = 25 + 10y - y^2 \tag{35}$$

Using MATLAB's polyfit, the exact solution was compared to the coefficient compared numerically and the values were relatively in agreement, however, they differed due to numerical error as a result of the coarse discretisation using $h = 0.2$. This was validated using smaller step sizes and the coefficients approached to theoretical values as would be expected. This was outside of the scope of this exercise so the result of a finer mesh will not be presented. The following figure shows the data at $x = 0$ with curvefit obtained from polyfit.

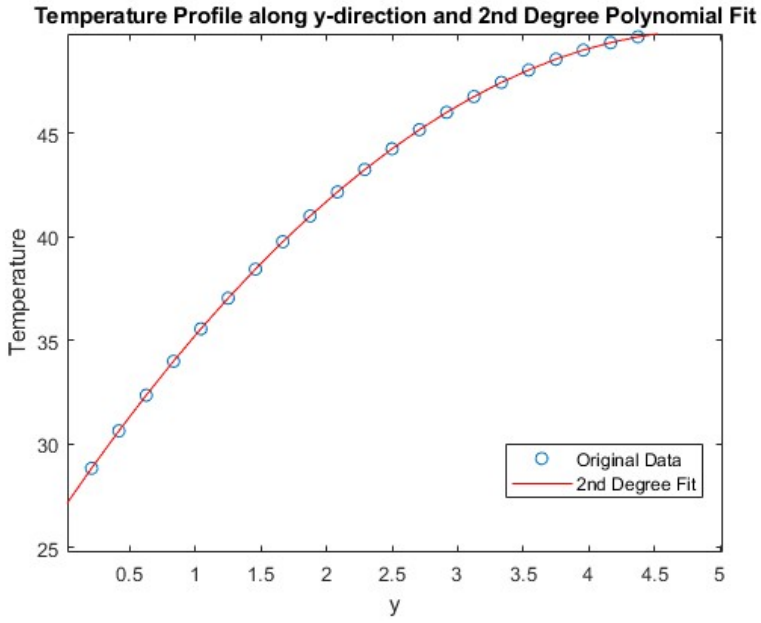

Figure 4: Data of temperature profile and 2^{nd} degree fit with $y = -0.92y^2 + 9.22y + 26.96$

2.3.3 Exercise c) Exactness of the Numerical Solution

For the given problem, the numerical solution obtained using the central difference method is theoretically exact for the second-order polynomial solution. This is because the central difference approximation of the second derivative is exact for second-order polynomials. To see this, consider the central difference approximation for $T''(y)$:

$$\frac{T(y+h) - 2T(y) + T(y-h)}{h^2} = T''(y) + R(y) \tag{36}$$

where $R(y)$ is the error term. To understand the nature of $R(y)$, we can expand $T(y+h)$ and $T(y-h)$ using Taylor series about the point y:

$$T(y+h) = T(y) + hT'(y) + \frac{h^2}{2}T''(y) + \frac{h^3}{6}T'''(y) + \mathcal{O}(h^4) \tag{37}$$

$$T(y-h) = T(y) - hT'(y) + \frac{h^2}{2}T''(y) - \frac{h^3}{6}T'''(y) + \mathcal{O}(h^4) \tag{38}$$

Substituting these into the central difference formula, we find that the h^3 terms cancel out, leaving us with:

$$R(y) = \frac{h^2}{12}T'''(y) + \mathcal{O}(h^4) \tag{39}$$

For a second-order polynomial $T(y) = c_0 + c_1 y + c_2 y^2$, the third derivative $T'''(y)$ is zero. Thus, the error term $R(y)$ vanishes, making the approximation exact.

However, this is not to mean the numerical values obtained will identically match the analytical. While the central difference method perfectly captures the second-order polynomial nature of the solution, the grid resolution (i.e., the choice of h) will still impact the accuracy of the solution when it comes to representing the domain and boundary conditions. A coarse grid might not represent the boundary conditions or the domain accurately, leading to errors. Thus, in practice, a sufficiently fine grid is necessary to capture the details of the problem accurately.
Comparing the solutions from parts (a) and (b) at the point $(x, y) = (6, 2)$, we obtain for the analytical that $T(6, 2) = 41$ and, recalling from a), the numerical solution for the given grid resolution is $T(6, 2) = 41.59$. As mentioned in b), this gets closer to 41 as we refined the grid, once again highlighting the fact that the of the accuracy is also affected by mesh quality.

2.4 Exercise d) Localised non-constant heat source

In this section, we are tasked with solving the same problem in Section 2.3.1, but with a non-constant heat source with the following formula:

$$f = 100exp(-\frac{1}{2}(x-4)^2 - 4(y-1)^2) \tag{40}$$

2.5 Exercise d) Localised non-constant heat source

In this exercise, we aim to determine the temperature distribution $T(x, y)$ for different mesh sizes and visualize the results. The given PDE, boundary conditions, and source term remain consistent with the previous sections.

Mesh Size (h)	Temperature at $(6, 2)$
0.2	[Temperature Value for $h = 0.2$]
0.1	[Temperature Value for $h = 0.1$]
0.05	[Temperature Value for $h = 0.05$]

For visualization, the solution for $h = 0.1$ was chosen due to its intermediate granularity. The temperature distribution was plotted using three different visualization techniques: a 3D surface plot, a 2D heatmap, and contour lines.

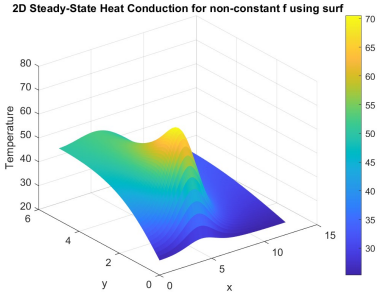

Figure 5: 3D surface plot of the temperature distribution for $h = 0.1$.

The temperature distribution visualisations provide insights into how heat propagates through the domain. The heat source is at its highest at (4,1) which is evident in the results obtained.

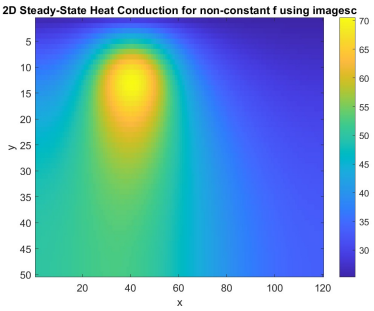

Figure 6: 2D heatmap of the temperature distribution for $h = 0.1$.

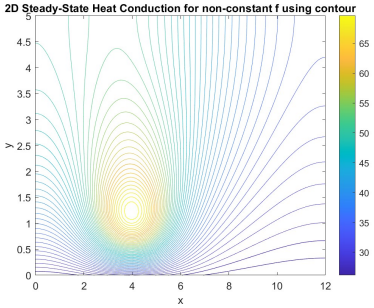

Figure 7: Contour plot of the temperature distribution for $h = 0.1$.

3 Comsol Multiphysics Analysis

3.1 Introduction

The problem geometry remains to be similar as that in Section 2, representing a block subjected to specific boundary conditions. However, in this section, several variations to the problem will be introduced, including changing boundary conditions and introducing cooling mechanisms like holes in the block. These modifications aim to simulate more realistic scenarios, such as convective cooling due to airflow or enhanced cooling due to ventilation holes.

3.2 Exercise a) Method and Result

In COMSOL, the *Heat Transfer in Solids* module was used to model our problem. A rectangular domain with dimensions L_x and L_y was defined. The differential equation was set up with the source term as described in Section 2.3.4, where f was non-constant. Boundary conditions were applied: the bottom boundary was set to a Dirichlet condition using "Initial Values", while the other boundaries were insulating and were set using Thermal Insulation". The standard and finer mesh options available in *Comsol* were used for our computations, ensuring the accuracy of our results.

The obtained results seem to be in close agreement with the results in Section 2.3.4. As can

be seen by comparing Fig 3 and Fig 8, the results are identical to the eye.

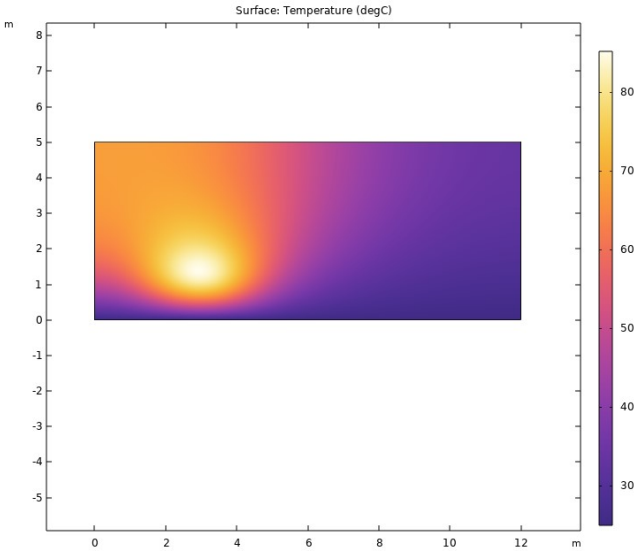

Figure 8: 2D colour scale plot of the temperature distribution $T(x, y)$.

By obtaining the temperature values at (x,y) = (6,2), it can be observed that for the CFD solution in COMSOL T(6,2) = 47.225(obtained 'normal' mesh using 264 domain elements and 44 boundary elements) and T(6,2) = 47.224 (obtained 'finer' mesh using 802 domain elements and 78 boundary elements). On the other hand, for the numerical solution using Finite Difference Method with the step size h = 0.2 (1334 internal elements and 166 boundary elements) the value at T(6,2) = 50.612. However, it was clear that with finer grid spacing the temperature were converging to that obtained using COMSOL. For instance, h = 0.01, gave T(6,2) = 47.396.

3.3 Exercise b) Method and Result

Just as in the previous exercise the same setup was used. However, an insulating Neumann boundary condition was used at y=0 using Thermal Insulationänd the following Robin boundary condition was implemented on the remaining three boundaries using "Heat Flux".

$$-\frac{\partial T}{\partial n} = \alpha(T - T_{ext}) \tag{41}$$

The temperature distribution for this setup is presented in Fig 9.
The average temperature at y = 5 was found using Evaluation Group $\implies$ Line Average to be T = 193.98.

3.4 Exercise c) Method and Result

In this exercise, the model is expanded by introducing a cooling mechanism: a circular hole cente red at $(x, y) = (6.0, 3.5)$ with a radius of 1.0. This hole allows air to flow through,

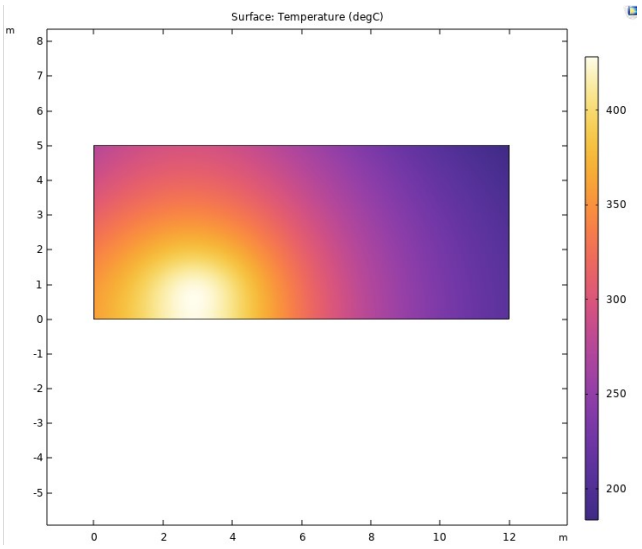

Figure 9: 2D colour scale plot of the temperature distribution $T(x, y)$ using combination of Neumann and Robin distribution.

leading to cooling effects. The edge of this hole is set to have the Robin boundary condition, as given by Eq. 41. The objective is to compute and visualise the temperature distribution, and analyse if it effects the average temperature at the top boundary $y = 5$.

The hole was implemented by creating a simple circular geometry with the given specifications and this was then subtracted from the rectangle using the boolean difference"in COMSOL.

The results seem to be reasonable as the hole seems to have cooling effect on the overall temperature distribution, but the top left side specifically, which is situated at the opposite side of where the heat source is the strongest, as can be seen in Fig. 10

The cooling mechanism is further solidified after the average temperature at $y = 5$ is extracted and is found to be $T = 147.85$, which compared to the results found in Section 3.3, is lower.

3.5 Exercise d) Method and Result

In this iteration of the model, the cooling mechanism is further refined by replacing the single hole with four smaller holes. These holes, each with a radius of 0.4, are centered at $(x, y) = (6.0 \pm 0.5, 3.5 \pm 0.5)$. The presence of multiple holes is expected to provide a more efficient cooling effect. Similar to the previous exercise, the edges of these holes are governed by the Robin boundary condition, as described by Eq. 41. The goal remains to compute and visualize the temperature distribution, with a particular emphasis on understanding the impact of these multiple holes on the average temperature at the top boundary $y = 5$.

The results, as visualized in Fig. 11, indicate a more pronounced cooling effect, especially

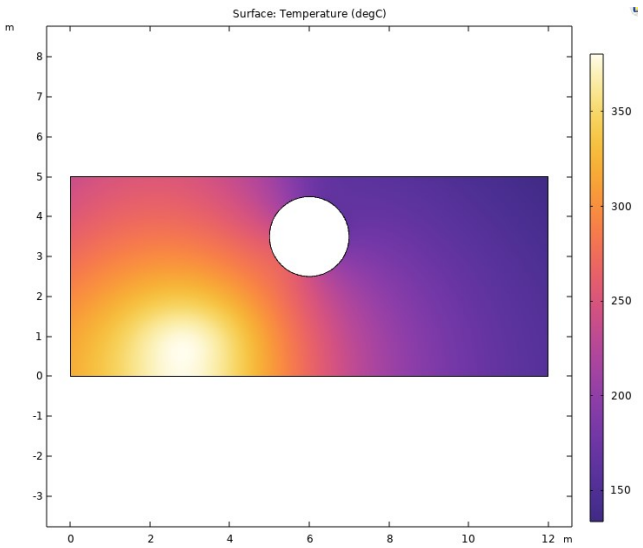

Figure 10: 2D colour scale plot of the temperature distribution $T(x, y)$ using combination of Neumann and Robin distribution and cooling mechanism in the form of a hole with radius 1 and at (6, 3.5).

around the regions of the holes. The temperature distribution appears more uniform, and the cooling effect of the multiple holes is evident.

The efficacy of this enhanced cooling mechanism is further corroborated when the average temperature at $y = 5$ is found to be $T = 134.10$. This temperature, when compared to results from Section 3.3 and the previous exercise, is notably lower, underscoring the improved cooling efficiency of the four-hole configuration.
The mesh plot for the configuration is presented in Fig. 12.
From the mesh plot, it's evident that the triangle sizes are notably smaller around the circular holes compared to the rectangular ones. This is due to the fact that curved geometries are inherently more complex. Circular boundaries, being non-linear, often require a finer mesh to accurately capture their shape. In contrast, linear or rectangular boundaries can be represented with larger mesh elements without compromising the solution's accuracy. This can be seen in the following example in which a rectangular and circular holes were implemented:

3.6 Exercise e) Method and Result

In this final exercise, the objective is to design a unique configuration that further reduces the temperature, specifically aiming for an average temperature on the top boundary $y = 5$ to be less than 100 degrees. Given the constraints, the chosen geometry is an array of small rectangles, arranged on an 8x8 grid, within the allowed region of the rectangle defined by $[6.0 \pm 1.5, 3.5 \pm 1]$. This design choice is influenced by the principle in thermal physics that states the surface area to volume ratio is an important factor in determining the cooling rate.

The results, as visualized in Fig. 14, show a significant reduction in temperature, especially

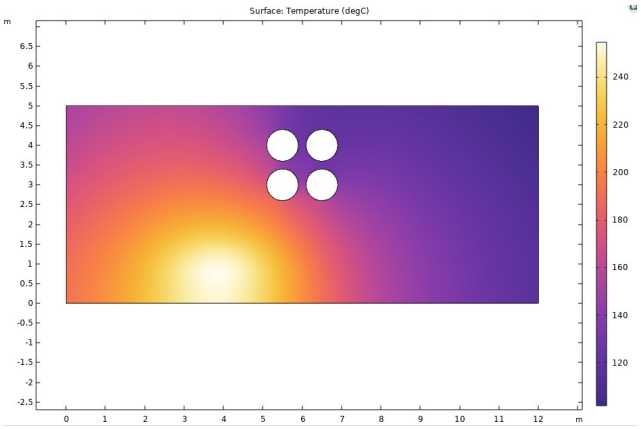

Figure 11: 2D colour scale plot of the temperature distribution $T(x, y)$ using a combination of Neumann and Robin boundary conditions, enhanced by the presence of four smaller cooling holes.

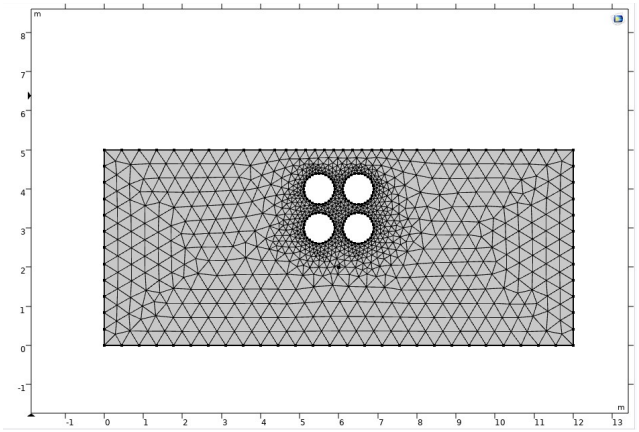

Figure 12: Mesh plot for the given configuration.

around the regions populated by the small rectangles. The increased surface area provided by these rectangles enhances the cooling effect.

The average temperature at $y = 5$ for this configuration is found to be $T = 88.10$, achieving the target of being below 100 degrees. The configuration was inspired by maximizing surface area for enhanced cooling. Furthermore, the practicality of this design can be seen in present non-complex products, which are ventilation systems or cooling systems in hardware components that employ the same structure. Further improvements could involve refining the size of the small triangles, and this could pose a challenge practically if we are to assume that our hypothesis maximizing surface area gives enhanced cooling". The numerical challenge would be if one were to minimise to temperature to extreme levels, for instance under 1 degree in this task. In that sense, an extremely fine mesh would be needed and therefore would lead very computational heavy calculations.

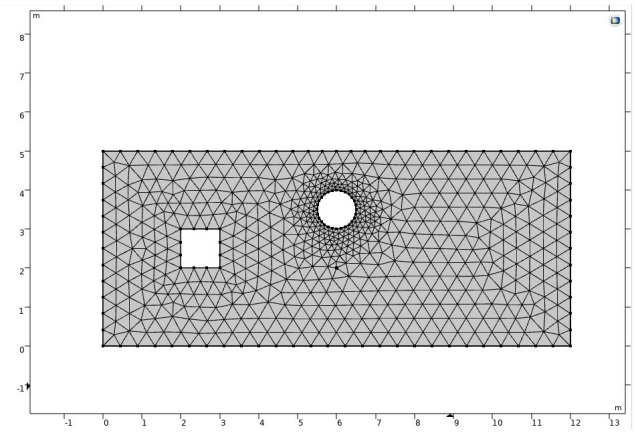

Figure 13: Mesh plot comparing the complexities of two holes with differing geometries.

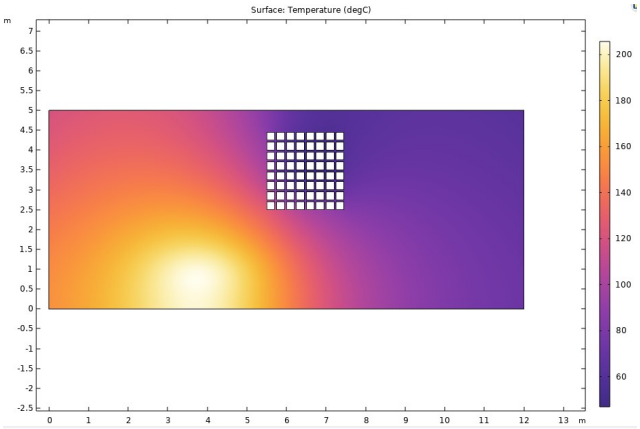

Figure 14: 2D colour scale plot of the temperature distribution $T(x, y)$ with a custom configuration of small rectangles to optimize cooling.