Journal of Engineering and Technological Advances (JETA)
https://jeta.segi.edu.my/index.php/segi
2024 Vol. 10 No. 2

2D FLAT PLATE HEAT CONDUCTION WITH CONSTANT THERMAL CONDUCTIVITY: IMPLEMENTING THE ALTERNATING DIRECTION IMPLICIT METHOD

[1]Aprecio, N.V.C. *, [1]Garcia, H.R.P., [1]Igno, J.J.M., [1]Lomboy, M.J.A., [1,2] Roy, F.A.Jr.

[1] College of Engineering, Pamantasan ng Lungsod ng Maynila, Philippines

[2] Department of Computer Engineering, Polytechnic University of the Philippines, Paranaque Campus, Philippines

*Corresponding Author: nvcaprecio2022@plm.edu.ph TEL: +639663100443

Received: 15 December 2024; Accepted: 27 December 2024; Published: 31 December 2024
doi: 10.35934/segi.v9i2.118

Highlights:

- The ADI method efficiently solves 2D heat conduction problems with high accuracy
- Increasing grid resolution reduces numerical errors but increases computational cost
- MATLAB implementation of ADI demonstrates its practical applicability in thermal engineering

Abstract: The heat equation is widely used in engineering applications to predict temperature distribution in materials subjected to heating or cooling, such as in high-temperature furnaces and pipeline-based heat networks. This study applies the alternating direction implicit (ADI) method to solve a two-dimensional heat conduction problem and evaluates its accuracy through grid convergence and error analysis. Results indicate that finer grid resolutions improve numerical accuracy, with absolute errors decreasing from 15.523 (for a grid size of 110) to 0.493 (for a grid size of 190) at the centre of the plate. Computational efficiency analysis reveals a trade-off, as execution times increase from 0.089907s to 0.432780s for the same grid refinements. These findings confirm the ADI method's reliability for thermal simulations, offering a balanced approach between precision and computational cost. The study concludes that the ADI method is a robust and efficient tool for modelling heat conduction in engineering applications.

Keywords: Alternating Direction Implicit, Heat Conduction, Grid Size, Analytical Solution, Numerical Solution

---

1. Introduction

The heat equation is extensively utilized in engineering applications, particularly for predicting temperature distribution when solid materials undergo heating or cooling in high-temperature furnaces (Norzilah & Nursalasawati, 2019). Moreover, heat conduction plays a crucial role in applications that demand efficient thermal energy transfer, such as pipeline-based heat networks that distribute heat from a source to end users for residential and domestic purposes.

Solving heat conduction problems analytically can be complex, particularly for materials like alpha-beta titanium alloy (Ti6Al4V), which are challenging to machine. Computational techniques utilizing numerical methods provide an effective approach for analysing such problems with high accuracy and efficiency. While one-dimensional heat conduction problems are relatively simpler to solve analytically, two-dimensional and three-dimensional cases present significantly greater challenges (Roy & Kumar, 2021).

The integration of MATLAB, Python, and parallel computing has significantly improved the efficiency of solving high-resolution heat transfer problems in recent years. Idoko *et al.* (2024) examined the application of MATLAB, COMSOL, and Python in mathematical modelling and simulation within precision engineering, highlighting their strengths in handling various engineering challenges, including Multiphysics simulations and custom algorithm development. This study builds upon these modern approaches by providing an optimized and validated implementation of the ADI method, demonstrating its effectiveness for computational heat transfer modelling.

This study conducts a detailed numerical analysis of the two-dimensional heat conduction problem on a flat plate with constant thermal diffusivity, employing the alternating direction implicit (ADI) method in MATLAB. The novelty of this research lies in several key aspects. First, a grid convergence analysis was performed to systematically evaluate the accuracy of the solution across different grid resolutions, ensuring both numerical stability and efficiency. Specifically, an error analysis was conducted by comparing ADI results from different grid sizes with analytical solutions to quantify any numerical discrepancies. Computational performance was also assessed through a computational time analysis, measuring the efficiency of the ADI method.

Various numerical techniques have been applied to solve the two-dimensional heat equation, each with distinct advantages and limitations. The finite difference method, though simple to implement, requires small time steps for stability in its explicit form, making it computationally

expensive (Smith, 1985). The Crank-Nicolson method, a fully implicit approach, is known for its accuracy but demands significant computational resources due to the need to solve large systems of equations. In contrast, the ADI method, which is utilized in this study, offers unconditional stability, improved computational efficiency, and lower memory requirements by breaking the solution process into alternating implicit steps (Peaceman & Rachford, 1955).

1.1. Partial Differential Equations

A partial differential equation (PDE) is a mathematical equation that involves multiple independent variables, the function itself, and its partial derivatives. The standard form of a PDE is expressed as:

$$F(x, y, u(x, y), u_x(x, y), u_y(x, y), u_{xx}(x, y), u_{xy}(x, y), u_{yx}(x, y), u_{yy}(x, y) = 0 \quad (1)$$

This formula represents a second-order PDE. Solving this type of equation involves finding a function $u(x, y)$ that satisfies the equation across all valid inputs, including its derivatives. In scenarios involving ordinary differential equations (ODEs), the function u can be a vector, such as $u(x, y) = \big(u_1(x, y), u_2(x, y), \dots, u_N(x, y)\big)$. This typically results in a system of PDEs (Ivrii, 2022).

1.2. Parabolic Partial Differential Equations

Consider a general second-order linear PDE with two independent variables, x and y, and one dependent variable, u, represented by the equation:

$$A\frac{\delta^2 u}{\delta x^2} + B\frac{\delta^2 u}{\delta x \delta y} + C\frac{\delta^2 u}{\delta y^2} + D = 0 \quad (2)$$

where A, B, and C are functions of x and y, and D can be a function of x, y, u, and their first derivatives. This equation is classified as parabolic when $B^2 - 4AC = 0$.

1.3. 2D Heat Equation

The two-dimensional heat equation is a partial differential equation (PDE) that describes the distribution of heat (or temperature) in a two-dimensional region over time. It's a fundamental equation in physics and engineering, with applications ranging from modelling heat transfer in materials to understanding diffusion processes in various systems. The equation itself is derived from the principle of conservation of energy, stating that the rate of change of heat energy within a region is equal to the net heat flux across its boundaries (Strauss, 2008).

An illustrative example of a parabolic PDE is the 2D heat conduction equation, expressed as:

$$\frac{\partial u}{\delta t} = \alpha(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}) \quad (3)$$

where $u(x, y, t)$ represents the temperature at point (x, y) and time t and α is the thermal diffusivity of the material, a constant that depends on the material's thermal conductivity, density, and specific heat capacity. A higher thermal diffusivity indicates faster heat propagation.

The thermal diffusivity, α, is defined by:

$$\alpha = \frac{k}{\rho C} \quad (4)$$

Here, k is the thermal conductivity, ρ is the density, and C is the specific heat capacity of the material. This model effectively describes the spatial and temporal distribution of heat within a metal rod, serving as a practical application of parabolic PDEs in the fields of engineering and physics (Kaw & Garapati, 2011).

The 2D heat equation states that the rate of temperature change at a point is proportional to the Laplacian of the temperature at that point. The Laplacian ($\nabla^2 u = \partial^2 u/\partial x^2 + \partial^2 u/\partial y^2$) represents the second-order spatial derivatives, indicating the curvature of the temperature field. A positive Laplacian suggests a region of relatively lower temperature surrounded by higher temperatures, leading to an increase in temperature over time. Conversely, a negative Laplacian signifies a region of higher temperature surrounded by lower temperatures, resulting in a decrease in temperature (Özişik, 2017).

The two-dimensional heat equation is a cornerstone in mathematical physics and engineering, widely used to model heat conduction in planar and semi-planar domains. This equation, classified as a parabolic partial differential equation, describes how heat energy disperses over time in a medium with spatial dimensions, making it essential for analysing processes involving thermal conductivity, such as in metals, semiconductors, and even biological tissues (Crank, 1984). Its importance lies not only in its theoretical foundation but also in its applications across various fields like material science, thermodynamics, and environmental studies (Incropera *et al.*, 2007).

The 2D heat equation assumes that the heat flow is governed by Fourier's law, stating that the rate of heat transfer is proportional to the negative gradient of the temperature. This proportionality ensures that heat diffuses from regions of higher temperature to regions of lower temperature, leading to equilibrium over time (Carslaw & Jaeger, 1959). Practical

solutions to the equation often involve initial conditions and boundary conditions tailored to specific scenarios, such as Dirichlet or Neumann conditions. These conditions reflect the nature of heat sources, insulation, or environmental influences on the material boundaries (Reddy, 2005).

From an engineering perspective, numerical methods like the finite difference method (FDM), finite element method (FEM), and alternating direction implicit (ADI) scheme have proven invaluable in solving the 2D heat equation. These techniques discretize the spatial and temporal domains into manageable grids, enabling approximate solutions for complex geometries where analytical methods are infeasible (Smith, 1985). Numerical approaches are particularly useful for simulations in irregular domains, as seen in aerospace engineering for evaluating thermal stress on aircraft structures (Patankar, 1980).

In modern applications, the 2D heat equation has been instrumental in simulating and optimizing thermal processes, such as in the design of heat exchangers and thermal barrier coatings. It is also pivotal in environmental science, modelling the dispersion of heat in oceans and the soil's surface layer (Zienkiewicz *et al.*, 2005). Beyond physical systems, it has inspired analogous models in financial mathematics and image processing, demonstrating its interdisciplinary relevance (Strikwerda, 2004).

The relevance of the 2D heat equation extends into current research, where scientists investigate the effects of variable thermal conductivity and anisotropic materials. These studies are critical for next-generation technologies like flexible electronics and nanomaterials (Cheng & Cheng, 2005). Moreover, integrating the heat equation with computational tools like MATLAB and Python has revolutionized its accessibility, enabling researchers and engineers to model and analyse thermal systems with unprecedented efficiency (Mathews & Fink, 2004).

1.4. Alternating Direction Implicit Method

The alternating direction implicit method is a widely used iterative technique in numerical linear algebra for solving Sylvester matrix equations. It is particularly favoured for addressing large matrix problems encountered in systems theory and control. ADI can be structured to generate solutions efficiently, often in a compact form that conserves memory. Its versatility extends to solving parabolic and elliptic partial differential equations numerically, making it a staple for modelling heat conduction and solving diffusion equations across multiple dimensions. This method falls under operator splitting methods, showcasing its adaptability across various computational challenges.

The ADI method is a two-step iterative process for updating an approximate solution to the equation $AX - XB = C$ (Wachspress, 2008). One iteration works as follows:

1. Solve for $X^{\left(j+\frac{1}{2}\right)}$ using the equation $\left(A + \beta_{j+1}I\right)X^{\left(j+\frac{1}{2}\right)} = X^{(j)}\left(\beta - \beta_{J+1}I\right) + C.$

2. Solve for $X^{(j+1)}$ using the equation $X^{(j+1)}\left(\beta - \alpha_{J+1}I\right) = \left(A - \alpha_{j+1}I\right)X^{\left(j+\frac{1}{2}\right)} - C.$

The convergence of ADI heavily depends on choosing suitable shift parameters $\left(\alpha_{j+1}, \beta_{j+1}\right)$ (Lu & Wachspress, 1991; Beckermann & Townsend, 2017). To perform K iterations of ADI, an initial guess $X^{(0)}$ and sets of shift parameters $\left(\alpha_j, \beta_j\right)_{j=1}^{K}$ are needed.

Originally, the ADI method was designed to tackle the 2D diffusion equation on a square domain through finite differences (Peaceman & Rachford, 1955). Unlike ADI applied to matrix equations, the version used for parabolic equations does not need shift parameters chosen upfront. Instead, the shift in each iteration is determined by factors like the timestep, diffusion coefficient, and grid spacing. The link to ADI for matrix equations becomes evident when examining how the ADI iteration affects the system during steady-state conditions.

The standard approach for numerically solving the heat conduction equation is through the Crank-Nicolson method. However, this method leads to complex equations in multiple dimensions that are time-consuming to solve. In contrast, the ADI method offers an advantage because the equations encountered in each step have a simpler structure, making them amenable to efficient solution using the tridiagonal matrix algorithm.

The ADI method is based on the concept of splitting finite difference equations into two parts: one where the x-derivative is treated implicitly, and the other where the y-derivative is treated implicitly.

$$\frac{u_{ij}^{n+\frac{1}{2}} - u_{ij}^{n}}{\Delta\frac{t}{2}} = \frac{\left(\delta_x^2 u_{ij}^{n+\frac{1}{2}} + \delta_y^2 u_{ij}^{n}\right)}{\Delta x^2} \tag{5}$$

$$\frac{u_{ij}^{n+1} - u_{ij}^{n+\frac{1}{2}}}{\Delta\frac{t}{2}} = \frac{\left(\delta_x^2 u_{ij}^{n+\frac{1}{2}} + \delta_y^2 u_{ij}^{n+1}\right)}{\Delta y^2} \tag{6}$$

The system of equations in the ADI method is symmetric and tridiagonal, with a bandwidth of 3, making it suitable for solving using the tridiagonal matrix algorithm. It has been demonstrated that this method is unconditionally stable and achieves second-order accuracy in both time and space.

1.5. Problem Statement

A solid square plate has an initial temperature of $50\ °C$. The edges of the plate are maintained at a constant temperature of $0\ °C$, as shown in **Figure 1**. The thermal conductivity of the material is $50\ \frac{W}{m\ K}$, the density is $1500\ \frac{kg}{m^3}$, and the specific heat capacity is $0.1333\ \frac{J}{kg\ K}$. The objective is to determine the temperature distribution within the plate after 10 seconds.

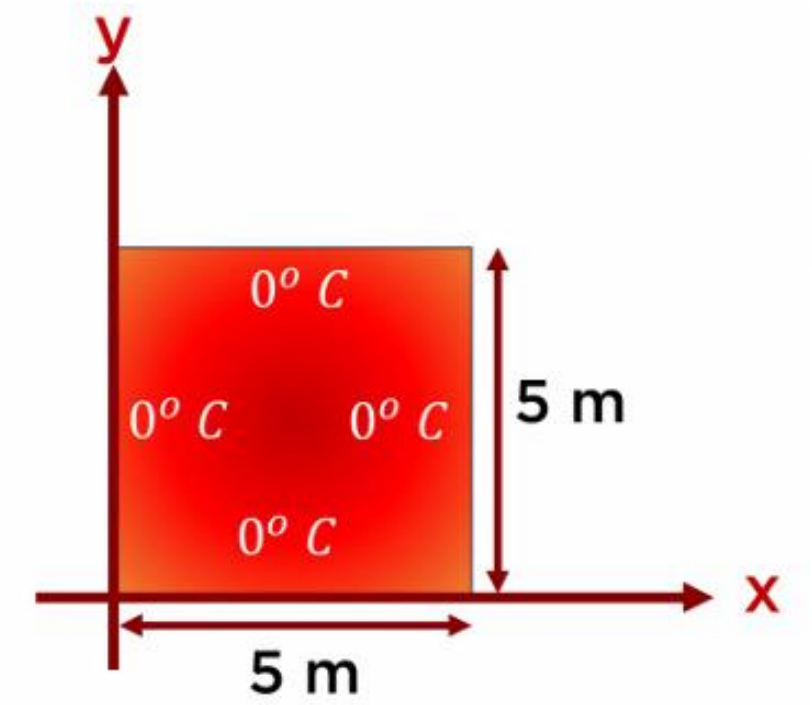

Figure 1. Schematic diagram of the problem

2. Methodology

2.1. Thermal Diffusivity

The thermal diffusivity is an important value in the 2D heat equation. To calculate it, Equation 4 was used. Given a thermal conductivity of $k = 50\ \frac{W}{m\ K}$; a density of $\rho = 1500\frac{kg}{m^3}$; and a specific heat capacity $c = 0.1333\ \frac{J}{kg\ K}$, the aforementioned values were substituted into Equation (4)

$$\alpha = \frac{50\ \frac{W}{m\ K}}{1500\frac{kg}{m^3} \cdot 0.1333\ \frac{J}{kg\ K}} \tag{7}$$

Equation 7 can be further simplified to:

$$\alpha = 0.25\ \frac{m^2}{s} \tag{8}$$

Equation 8 is the obtained thermal diffusivity which was used to solve the 2D heat equation.

2.2. The Analytical Solution via Separation of Variables

The governing equation is the 2D heat equation is given in Equation 9:

$$\partial T/\partial t = \alpha(\partial^2 T/\partial x^2 + \partial^2 T/\partial y^2) \tag{9}$$

To solve this equation using the separation of variables method, we assume that the temperature function can be expressed as the product of separate functions of x, y, and t.

$$T(x, y, t) = X(x)Y(y)T(t) \tag{10}$$

Substituting into Equation 9:

$$\alpha\left(\left(1/X(x)\right)(d^2X(x)/dx^2) + \left(1/Y(y)\right)(d^2Y(y)/dy^2)\right) = \left(1/T(t)\right)(dT(t)/dt) \tag{11}$$

Since the left-hand side depends only on x, y and the right-hand side depends only on t, both sides must be equal to a constant, −λ.

$$dT/dt\ +\ \lambda T\ =\ 0 \tag{12}$$

For the spatial components:

$$\left(1/X(x)\right)(d^2X(x)/dx^2) + \left(1/Y(y)\right)(d^2Y(y)/dy^2) = -\lambda/\alpha \tag{13}$$

Rearranging:

$$dT/dt\ =\ -\lambda T \tag{14}$$

Separating variables and integrating:

$$\int\ (dT/T)\ =\ -\lambda \int\ dt \tag{15}$$

$$ln\ T\ =\ -\lambda t\ +\ C \tag{16}$$

Exponentiating both sides:

$$T(t)\ =\ C\ e\text{^}(-\lambda t) \tag{17}$$

Since the solution should remain bounded:

$$T(t)\ =\ A\_n\ e\text{^}(-\lambda\_n\ t) \tag{18}$$

Rearranging:

$$d^2X(x)/dx^2 + k^2 X(x) = 0 \quad (19)$$

where $k^2 = \lambda/\alpha$. The general solution is:

$$X(x) = C_1 cos(kx) + C_2 sin(kx) \quad (20)$$

Applying boundary conditions at $x = 0$ and $x = L$:

$$X(0) = 0 \Rightarrow C_1 = 0 \quad (21)$$

$X(L) = 0 \Rightarrow sin(kL) = 0 \Rightarrow k_n = (n\pi)/L$, where $n = 1,2,3,\ldots$:

$$X_n(x) = sin(\,(n\pi x)/L\,) \quad (22)$$

Similarly, solving for $Y(y)$:

$$Y_n(y) = sin\big((n\pi y)/L\big) \quad (23)$$

The full solution is:

$$T(x,y,t) = \Sigma_{n=1}^{\infty} A_n sin\big((n\pi x)/L\big) sin\big((n\pi y)/L\big) e^{\left(-((n\pi)/L)^2 \alpha t\right)} \quad (24)$$

Using the initial condition $T(x,y,0) = 50°C$:

$$50 = \Sigma_{n=1}^{\infty} A_n sin\big((n\pi x)/L\big) sin\big((n\pi y)/L\big) \quad (25)$$

Multiplying both sides by $sin(\,(m\pi y)/L\,)$ and integrating:

$$A_n = (4/L^2) \int_0^L \int_0^L 50\, sin\big((n\pi x)/L\big) sin\big((n\pi y)/L\big) dx\, dy \quad (26)$$

Computing the integral:

$$A_n = (4 \times 50)/L^2 \quad (27)$$

Substituting $L = 5$:

$$A_n = (4 \times 50)/5^2 = 8 \quad (28)$$

Thus, the final solution is:

$$T(x,y,t) = \Sigma_{n=1}^{\infty} 8 sin\big((n\pi x)/5\big) sin\big((n\pi y)/5\big) e^{\left(-((n\pi)/5)^2 \alpha t\right)} \quad (29)$$

where $\alpha = 0.25\ {}^{m^2}/_{s}$.

2.3. MATLAB

To solve the given problem involving 2D heat diffusion equation, the ADI method is implemented using MATLAB. The flowchart of MATLAB coding is shown in **Figure 2**.

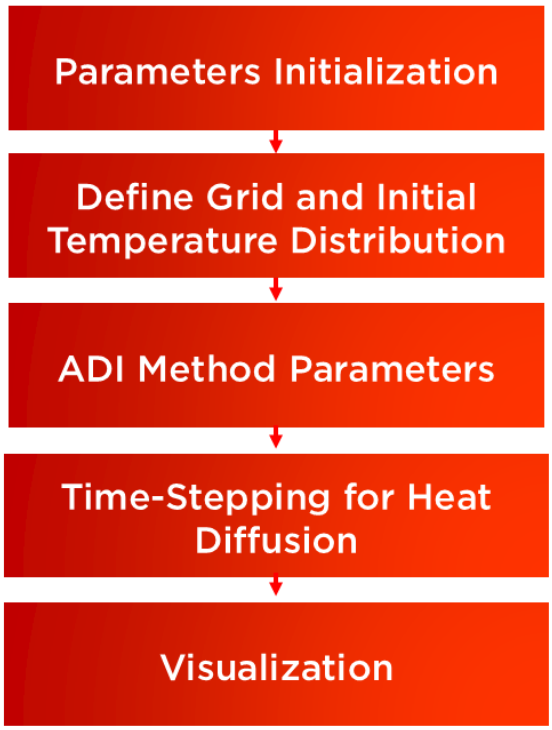

Figure 2. Flowchart of MATLAB coding

MATLAB is first used to define the physical and simulation parameters necessary for the model. This includes the dimensions of the plate, the number of grid points for the spatial discretization, the time step for the simulation, and the thermal diffusivity of the material. These parameters are crucial for setting up the simulation environment and ensuring that the numerical solution is accurate and stable.

The initial temperature distribution across the plate is set up in MATLAB using matrix operations. MATLAB's efficient handling of matrices makes it ideal for setting up the initial conditions and applying boundary conditions. The temperature matrix is initialized, and boundary conditions are applied to simulate the physical constraints of the problem.

MATLAB's core strength in numerical analysis is leveraged to implement the ADI method, a numerical method used to solve partial differential equations like the heat equation. The ADI method is particularly suited for problems like this because it is stable and efficient for

multidimensional problems. MATLAB is used to construct and solve the tridiagonal matrix equations that arise in the ADI method, updating the temperature distribution at each time step.

After computing the temperature distribution, MATLAB is also used to visualize the results. The meshgrid and surf functions are employed to create a 3D surface plot of the temperature distribution over the plate. This visualization is crucial for analysing the behaviour of the heat distribution over time and can be used to verify the correctness of the numerical solution and to present the results in a visually appealing format.

2.4. The Coding Process

Below is the detailed discussion of the coding process.

2.4.1. Initialization of Parameters

L, N, dx, dt, alpha, total time. These lines define the simulation's physical and computational parameters. L is the plate's length, N is the number of grid points per dimension, dx is the distance between grid points, dt is the time step, alpha is the thermal diffusivity, and total time is the duration of the simulation.

2.4.2. Temperature Matrix Initialization

T. Initializes the temperature matrix with a uniform initial temperature of $50° C$, except at the boundaries where the temperature is set to $0° C$ due to the specified boundary conditions.

2.4.3. ADI Method Parameters

a, b. These coefficients are used in the tridiagonal matrix setup for the ADI method, where a is the diagonal element and b is the off-diagonal element, adjusted for the thermal diffusivity and grid spacing.

2.4.4. Time-Stepping Loop

The loop represents the time evolution of the temperature distribution. It uses two nested loops to apply the ADI method alternately in the x and y directions. Each iteration updates the temperature matrix T.

2.4.5. 3D Visualization

Meshgrid and Surface Plot. The meshgrid function generates matrices X and Y that correspond to the coordinates of the plate. The surf function then plots these coordinates against the temperature matrix T, resulting in a 3D surface plot. The plot is enhanced with labels and a colour bar to indicate temperature values.

2.5. Error Analysis

This was performed to determine the accuracy of the ADI method relative to the analytical solution. Equation K was used to determine the absolute error.

$$absolute\ error = |\ true\ value\ -\ measured\ value\ | \tag{30}$$

Here, the true value is represented by the value obtained from the analytical solution while the measured value is the value obtained by the numerical solution via ADI method.

3. Results and Discussion

Figure 3 illustrates the temperature distribution of the given problem after 10 seconds with respect to the initial and boundary conditions.

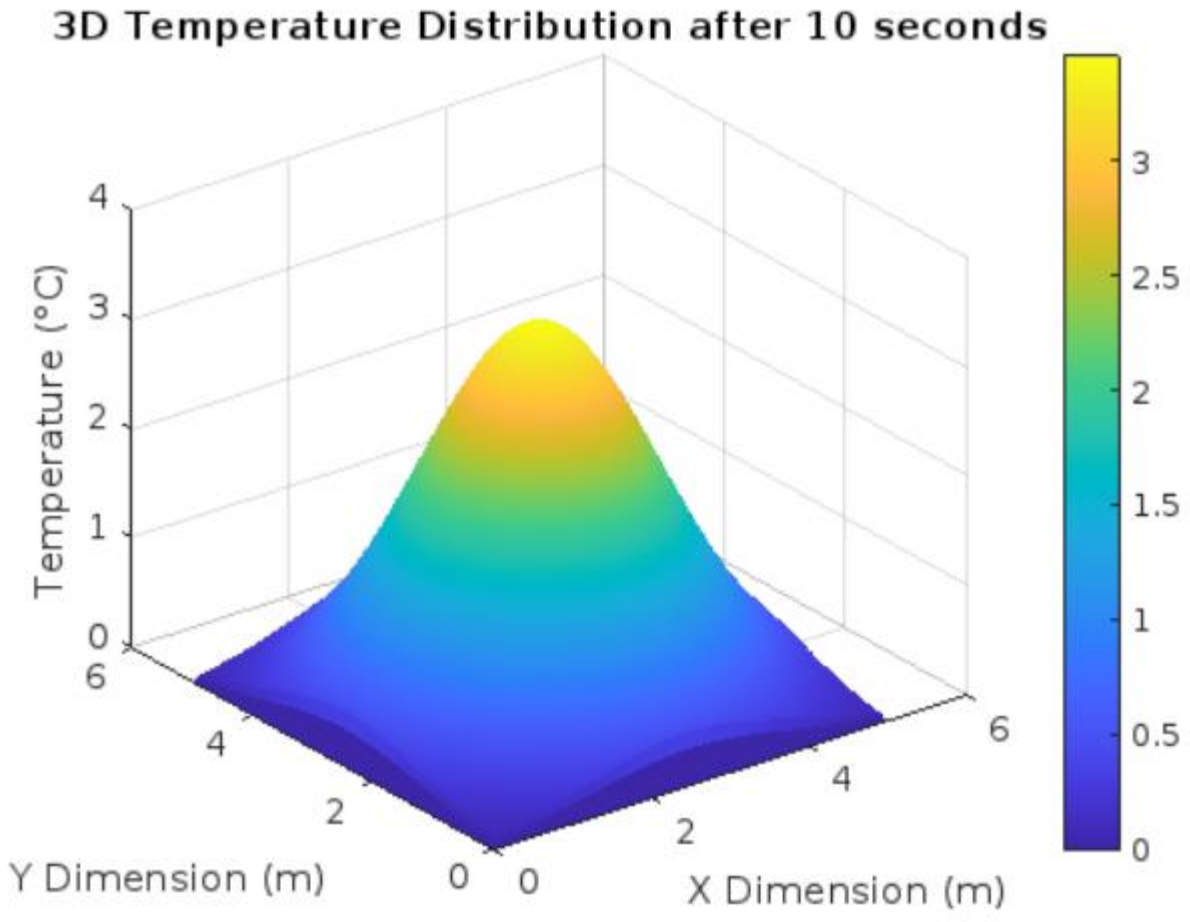

Figure 3. 3D temperature distribution using alternating direction implicit

The x-axis and y-axis both represent spatial dimensions measured in meters (m). On the other hand, the z-axis represents the temperature. The colour scale on the right side of the plot indicates the temperature values, where dark blue represents the lowest temperature and bright

yellow represents the highest temperature. The highest temperatures are shown in yellow and are located at the centre of the plot, indicating a concentration of heat in this area. The temperatures decrease towards the edges of the plot, transitioning through green to dark blue. The plot features a bell-shaped surface. At the centre, the graph features a prominent peak, corresponding to the highest temperature in the simulation, which gradually decreases when away from the centre.

3.1. Error Analysis

To determine the accuracy of the obtained results via ADI method, an error analysis was performed. Equation 30 was utilized. Temperature at specific x and y values were obtained from the analytical solution and the ADI method at different grid sizes. Afterwards, the absolute error was obtained. The analysis is conducted along both the x-direction at $y = 3$ and the y-direction at $x = 2$. Additionally, the computational efficiency of different grid sizes was assessed.

The absolute error for different grid sizes was summarized in **Table 1**. As expected, the error decreases as the grid resolution increases. Notably, the largest errors occur at the centre of the plate, this is due to a combination of factors. The ADI method introduces a splitting error by solving the problem in two half-steps (implicit in the $x-$and$-y$-directions separately), which accumulates more significantly in the centre where the solution is less constrained by boundary conditions. Near the boundaries, the solution is strongly influenced by well-defined boundary conditions, reducing errors, but at the centre, errors propagate and amplify due to symmetry and numerical dispersion. Additionally, the centre experiences slower transient response to temperature changes, making it more sensitive to truncation errors arising from spatial and temporal discretization. Coarser grids and larger time steps exacerbate these errors, as the smooth variations in the centre are not captured accurately. Thus, the combination of error accumulation, symmetry, slower diffusion, and discretization limitations leads to higher errors in the central region compared to the edges.

Table 1. Absolute error analysis at y = 3 along the x-direction

x (m)	$T_{Analytical}$	$T_{N=110}$	$T_{N=130}$	$T_{N=150}$	$T_{N=170}$	$T_{N=190}$
0.5	0.8501	4.2213	2.6872	1.4910	0.6249	0.0378
1.0	1.6139	7.2610	4.5765	2.4831	0.9675	0.0599
1.5	2.2134	11.0220	7.0187	3.8969	1.6366	0.1042
2.0	2.5875	14.2340	9.1464	5.1788	2.3060	0.3582
2.5	2.6978	15.5230	10.0120	5.7147	2.6029	0.4931
3.0	2.5875	14.2340	9.1464	5.1788	2.3060	0.3582
3.5	2.2134	11.0220	7.0187	3.8969	1.6366	0.1042
4.0	1.6139	7.2610	4.5765	2.4831	0.9675	0.0599
4.5	0.8501	4.2213	2.6872	1.4910	0.6249	0.0378

Table 2 presents the range of absolute errors in temperature calculations along the x-direction at $y = 3$. As grid size increases, the minimum and maximum errors decrease significantly, demonstrating the improvement in numerical accuracy. For instance, at a grid size of 110, the maximum error reaches 15.523, whereas at 190, it reduces to 0.493. This trend highlights the expected behaviour of the ADI method, where finer spatial discretization minimizes truncation and discretization errors. The results confirm that increasing grid resolution enhances computational precision, albeit at the cost of increased computational time.

Similar trends in error behaviour are observed along the y-direction at $x = 2$, as shown in **Table 3**. The largest errors again appear near the centre, reinforcing the observation that numerical errors accumulate due to the centrality of the diffusion process in the domain.

Table 2. Error range at y = 3 along the x-direction

Grid Size	Min Error	Max Error
110	4.2213	15.523
130	2.6872	10.012
150	1.4910	5.715
170	0.6249	2.603
190	0.0378	0.493

Table 3. Absolute error analysis at $x = 2$ along the y-direction

y (m)	$T_{Analytical}$	$T_{N=110}$	$T_{N=130}$	$T_{N=150}$	$T_{N=170}$	$T_{N=190}$
0.5	0.8501	4.2213	2.6872	1.4910	0.6249	0.0378
1.0	1.6139	7.2610	4.5765	2.4831	0.9675	0.0599
1.5	2.2134	11.0220	7.0187	3.8969	1.6366	0.1042
2.0	2.5875	14.2340	9.1464	5.1788	2.3060	0.3582
2.5	2.6978	15.5230	10.0120	5.7147	2.6029	0.4931
3.0	2.5875	14.2340	9.1464	5.1788	2.3060	0.3582
3.5	2.2134	11.0220	7.0187	3.8969	1.6366	0.1042
4.0	1.6139	7.2610	4.5765	2.4831	0.9675	0.0599
4.5	0.8501	4.2213	2.6872	1.4910	0.6249	0.0378

Table 4 presents a similar error analysis along the y-direction at $x = 2$. The results align with the observations from **Table 2**, where larger grid sizes lead to smaller errors. The highest error at the lowest resolution (grid size 110) is 15.523, which reduces to 0.493 at the finest resolution (grid size 190). This consistency in error reduction along both spatial directions indicates the robustness of the ADI method. Additionally, the error distribution suggests that numerical

discrepancies are more pronounced in the centre of the plate due to the iterative nature of the ADI scheme, where errors propagate from boundary conditions inward.

Table 4. Error range at $x = 2$ along the y-direction

Grid Size	Min Error	Max Error
110	4.2213	15.523
130	2.6872	10.012
150	1.4910	5.715
170	0.6249	2.603
190	0.0378	0.493

3.2. Computational Efficiency Analysis

The execution time for different grid sizes is recorded in **Table 5**. As expected, finer grids result in increased computation times due to the larger number of grid points and the corresponding increase in the number of calculations.

Table 5. Computational efficiency analysis

Grid Size	Execution Time (s)
110	0.089907
130	0.140320
150	0.155160
170	0.340320
190	0.432780

The results indicate that increasing the grid resolution improves accuracy, but at the cost of higher computational time. The accumulation of error at the centre of the plate is attributed to the numerical diffusion and truncation errors inherent in finite difference approximations. Since heat diffuses symmetrically from the boundaries, central points accumulate more propagated errors from surrounding nodes. This suggests that adaptive meshing techniques or

higher-order numerical schemes may be needed to reduce central error accumulation while maintaining computational efficiency.

The ADI method offers several advantages over other numerical techniques, such as the Crank-Nicolson and explicit finite difference methods, particularly in solving PDEs like the heat conduction equation. One of the primary benefits of the ADI method is its computational efficiency. Unlike the Crank-Nicolson method, which requires solving large, banded linear systems, the ADI method simplifies these computations by splitting the finite difference equations into two stages. In each stage, only one spatial derivative is treated implicitly, resulting in systems that are easier and faster to solve using algorithms like the tridiagonal matrix algorithm (Peaceman & Rachford, 1955). This reduction in computational complexity makes the ADI method particularly suitable for large-scale problems.

Both the ADI and Crank-Nicolson methods are unconditionally stable and second-order accurate in time and space. However, the ADI method's structure allows for more efficient computations without compromising stability or accuracy. This efficiency becomes increasingly significant as the problem's dimensionality and grid size grow (Saqib *et al.*, 2017).

Furthermore, Ajeel & Gaftan (2023) observed that the Crank-Nicolson method exhibits higher accuracy in the initial time steps, while the ADI method shows improved accuracy in later stages. Moreover, the explicit finite difference method is straightforward to implement but suffers from conditional stability, necessitating small time steps to maintain accuracy. In contrast, the ADI method's unconditional stability permits larger time steps, enhancing computational efficiency. Additionally, the ADI method's approach of solving tridiagonal systems reduces memory requirements compared to the Crank-Nicolson method, which often involves handling larger matrices (Peaceman & Rachford, 1955).

4. Conclusion

The study successfully demonstrated the effectiveness of the ADI method in solving the two-dimensional heat conduction problem. The numerical results closely align with the analytical solution, with errors significantly decreasing as grid resolution improves. The computational efficiency analysis confirms that while finer grids yield higher accuracy, they also demand increased processing time. These findings affirm the ADI method's utility in thermal simulations, making it a reliable choice for engineering applications requiring precise heat distribution modelling. For future work, it is recommended to extend the analysis to problems

involving variable thermal conductivity and complex geometries, as well as to explore hybrid numerical methods that may further improve accuracy and efficiency.

Acknowledgement

The authors would like to thank Pamantasan ng Lungsod ng Maynila and Polytechnic University of the Philippines for supporting the research of this study.

Credit Author Statement

Conceptualization and methodology, Aprecio, N.V.C. and Roy, F.A.Jr.; review of related literature, Aprecio, N.V.C., Lomboy, M.J.A. and Roy, F.A.Jr.; validation of solution, Aprecio, N.V.C. and Roy, F.A.Jr.; writing¬—original draft preparation, Aprecio, N.V.C., Garcia, H.R.P., and Igno, J.J.M.; writing—review and editing, Aprecio, N.V.C. and Roy, F.A.Jr.

Conflicts of Interest

The authors declare no conflict of interest.

References

Ajeel, O. A., & Gaftan, A. M. (2023). Using Crank-Nicolson numerical method to solve heat-diffusion problem. *Tikrit Journal of Pure Science*, 28(3), 101-104.

Beckermann, B., & Townsend, A. (2017). On the singular values of matrices with displacement structure. *SIAM Journal on Matrix Analysis and Applications*, 38(4), 1227-1248.

Carslaw, H. S., & Jaeger, J. C. (1959). *Conduction of heat in solids* (2nd ed.). Oxford University Press.

Cheng, A. H. D., & Cheng, D. T. (2005). Heritage and early history of the boundary element method. *Engineering analysis with boundary elements*, 29(3), 268-302.

Crank, J. (1984). *Free and moving boundary problems*. Clarendon Press.

Idoko, I. P., Ezeamii, G. C., Idogho, C., Peter, E., Obot, U. S., & Iguoba, V. A. (2024). Mathematical modeling and simulations using software like MATLAB, COMSOL and Python. *Magna Scientia Advanced Research and Reviews*, 12(2), 062-095.

Incropera, F. P., DeWitt, D. P., Bergman, T. L., & Lavine, A. S. (2007). *Fundamentals of heat and mass transfer* (6th ed.). Wiley.

Ivrii, V. (2022). *Partial differential equations*. https://www.math.toronto.edu/ivrii/PDE-textbook/PDE-textbook.pdf

Kaw, A., & Garapati, S. H. (2011). *Parabolic differential equations*. https://mathforcollege.com/nm/mws/gen/10pde/mws_gen_pde_txt_parabolic.pdf

Lu, A., & Wachspress, E. L. (1991). Solution of Lyapunov equations by alternating direction implicit iteration. *Computers & Mathematics with Applications,* 21(9), 43-58.

Mathews, J. H., & Fink, K. D. (2004). *Numerical methods using MATLAB*. Pearson.

Norzilah, A. H., & Nursalasawati, R. (2019). Alternating direction implicit (ADI) method for solving two dimensional (2-D) transient heat equation. *ASM Science Journal, Special for SKSM,* 26(6), 28-33.

Özişik, M. N. (2017). *Heat conduction*. John Wiley & Sons.

Patankar, S. V. (1980). *Numerical heat transfer and fluid flow*. Taylor & Francis.

Peaceman, D. W., & Rachford, Jr, H. H. (1955). The numerical solution of parabolic and elliptic differential equations. *Journal of the Society for industrial and Applied Mathematics,* 3(1), 28-41.

Reddy, J. N. (2005). *An introduction to the finite element method* (3rd ed.). McGraw-Hill.

Roy, A. K., & Kumar, K. (2021). 2D heat conduction on a flat plate with Ti6Al4V alloy under steady state conduction: A numerical analysis. *Materials Today: Proceedings*, *46*, 896-902.

Saqib, M., Hasnain, S., & Mashat, D. S. (2017). Computational solutions of two dimensional convection diffusion equation using crank-nicolson and time efficient ADI. *American Journal of Computational Mathematics,* 7(03), 208–227.

Smith, G. D. (1985). *Numerical solution of partial differential equations: Finite difference methods*. Clarendon Press.

Strauss, W. A. (2008). *Partial differential equations: An introduction*. John Wiley & Sons.

Strikwerda, J. C. (2004). *Finite difference schemes and partial differential equations* (2nd ed.). Society for Industrial and Applied Mathematics.

Wachspress, E. L. (2008). Trail to a Lyapunov equation solver. *Computers & Mathematics with Applications*, *55*(8), 1653-1659.

Zienkiewicz, O. C., Taylor, R. L., & Zhu, J. Z. (2005). *The finite element method: Its basis and fundamentals* (6th ed.). Butterworth-Heinemann.