DEMOCRATIC AND POPULAR REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

Mohamed Boudiaf University of Msila
Faculty of Mathematics and Computer Sciences
Department of Mathematics

# *Master's Thesis*

**Field :** Mathematics and Computer Sciences
**Branch :** Mathematics
**Option :** AMN

# Theme

## Comparative Study of Explicit and Implicit Schemes for the 1D Heat Equation

**Presented by :**
***Achour Ibtissam***

**In front of the jury composed of :**

| | | | |
|---|---|---|---|
| DALLOUM Wahiba | M.C.B, | University of Msila | **chairperson** |
| Guellati Zineb | M.C.B, | University of Msila | **Supervisor** |
| ZERGUINE Houria | M.C.B, | University of Msila | **Examiner** |

Academic year 2025/2026

# Contents

# List of Figures

# List of Tables

# General Introduction

The study of heat transfer and energy propagation within physical media represents a cornerstone of modern engineering sciences and technological advancements. The one-dimensional (1D) unsteady heat equation serves as a fundamental mathematical model that governs these physical phenomena. Mathematically, it is classified as a parabolic partial differential equation (PDE), derived from the physical foundation of Fourier's Law of Conduction and the universal principle of energy conservation.

Although classical analytical methods, such as the separation of variables and Fourier series expansion, provide exact solutions under idealized geometries and conditions, they often prove inadequate when encountering the complexities of real-world applications. Consequently, numerical methods have emerged as an indispensable and reliable alternative. Foremost among these approaches is the Finite Difference Method (FDM), which serves as a powerful and efficient framework for transforming continuous differential models into solvable algebraic systems by discretizing both spatial and temporal domains.

The paramount challenge in numerical modeling invariably centers on achieving a precise balance between computational efficiency, accuracy, and numerical stability. This research aims to conduct a rigorous and comprehensive comparative analysis between two fundamental numerical strategies: the explicit Forward-Time Central-Space (FTCS) scheme and the implicit Backward-Time Central-Space (BTCS) scheme. The primary objective is to evaluate their mathematical performance regarding consistency, stability, and convergence under specified boundary conditions.

To achieve these academic objectives with scholarly rigor, the study establishes a solid mathematical framework rooted in the core classifications of partial differential equations. These theoretical formulations are subsequently translated into optimized computational algorithms implemented within the MATLAB environment, allowing for a dynamic evaluation of solution behav-

iors and error propagation.

The methodology of this investigation is structured systematically across four distinct chapters:

- **Chapter 1 (Mathematical Background and Heat Equation Foundations):** Delineates the mathematical classification of PDEs, the physical derivation of the 1D heat equation based on Fourier's law and energy balance, and establishes the initial and boundary conditions alongside the classical analytical solution via separation of variables to serve as an exact baseline.
- **Chapter 2 (Finite Difference Formulation and Comparative Foundations):** Focuses on the discretization of the computational domain, the derivation of finite difference approximations using Taylor series expansions, and the detailed mathematical formulation of both the explicit (FTCS) and implicit (BTCS) schemes, incorporating the Thomas Algorithm (TDMA) to resolve the tridiagonal matrix systems.
- **Chapter 3 (Stability and Convergence Analysis):** Dedicates itself to the rigorous theoretical verification of the numerical schemes, investigating local truncation errors for consistency, conducting spectral stability analysis using the Von Neumann method, and culminating in Lax's Equivalence Theorem which binds consistency, stability, and convergence.
- **Chapter 4 (Numerical Results and Comparative Simulation):** Represents the empirical core of the thesis, presenting MATLAB simulation results, graphical distributions of temperature profiles, and a detailed sensitivity analysis demonstrating how varying grid sizes and time steps directly govern stability thresholds and absolute error limits.

# Chapter 1

# Mathematical Background and Heat Equation Foundations

## 1.1 Introduction

The study of heat propagation in solids requires a structured framework to ensure the consistency of the governing equations. This chapter establishes the theoretical foundations of the 1D unsteady heat equation. We begin by defining the function spaces where our solutions reside, followed by the classification of Partial Differential Equations (PDEs). Finally, we derive the governing equation from physical principles, define its boundary conditions, and present the analytical solution method which serves as a benchmark for our numerical results [3].

## 1.2 Classification of Partial Differential Equations (PDEs)

Following the classification criteria detailed in [3], a general second-order PDE is categorized by its discriminant $\Delta = B^2 - 4AC$:

- **Elliptic ($\Delta < 0$):** Describes steady-state systems.
- **Hyperbolic ($\Delta > 0$):** Models wave propagation and vibrations.
- **Parabolic ($\Delta = 0$):** This is the category of the **Heat Equation**, governing diffusion processes where the solution smooths out over time.

## 1.3 Physical Derivation of the 1D Heat Equation

The heat equation is a mathematical statement of the Law of Conservation of Energy [1].

### 1.3.1 Fourier's Law of Conduction

The heat flux $q$ is proportional to the negative temperature gradient:

$$q = -k\frac{\partial T}{\partial x} \tag{1.1}$$

where $k$ is the thermal conductivity ($W/m \cdot K$).

### 1.3.2 Energy Balance Principle

For a small segment $\Delta x$ of a rod with density $\rho$ and specific heat $c_p$:

$$\rho c_p A \Delta x \frac{\partial T}{\partial t} = Aq(x,t) - Aq(x+\Delta x, t) \tag{1.2}$$

Taking the limit $\Delta x \to 0$ leads to:

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \tag{1.3}$$

where $\alpha = \frac{k}{\rho c_p}$ is the **Thermal Diffusivity** ($m^2/s$).

## 1.4 Transition to Numerical Modeling

In computational environments like **MATLAB**, physical constants are simplified into grid-based parameters.

- **Consolidation of Properties:** All physical properties ($\rho, c_p, k$) are merged into the diffusivity coefficient $\alpha$.
- **Grid Parameters:** The domain is discretized into spatial steps $\Delta x$ and temporal steps $\Delta t$.
- **Stability Factor ($r$):** A dimensionless parameter governing the scheme's behavior:

$$r = \alpha \frac{\Delta t}{(\Delta x)^2} \tag{1.4}$$

.

## 1.5 Initial and Boundary Conditions

To obtain a unique and physically meaningful solution, the PDE must be accompanied by auxiliary conditions [2, 3].

### 1.5.1 Initial Conditions (I.C.)

Specifies the temperature distribution at $t = 0$:

$$T(x, 0) = f(x), \quad \forall x \in [0, L] \tag{1.5}$$

where $f(x)$ represents the starting temperature profile.

### 1.5.2 Boundary Conditions (B.C.)

These describe how the system interacts with its boundaries ($x = 0$ and $x = L$):

- **Dirichlet Conditions (First Kind):** Temperature is explicitly fixed.

$$T(0, t) = T_{left}, \quad T(L, t) = T_{right} \tag{1.6}$$

  *Example:* A rod with its ends submerged in ice baths at $0°C$.

- **Neumann Conditions (Second Kind):** Specifies the heat flux (derivative). For insulated ends:

$$\left.\frac{\partial T}{\partial x}\right|_{x=0} = 0, \quad \left.\frac{\partial T}{\partial x}\right|_{x=L} = 0 \tag{1.7}$$

## 1.6 Analytical Solution Methods (Separation of Variables)

Before moving to numerical approximations, we analyze the exact behavior using the method of Separation of Variables [4].

### 1.6.1 Methodological Approach

We assume $T(x, t) = X(x) \cdot \phi(t)$. This leads to two Ordinary Differential Equations (ODEs):

1. A temporal ODE: $\frac{d\phi}{dt} + \alpha\lambda\phi = 0$ (exponential decay).

2. A spatial ODE: $\frac{d^2X}{dx^2} + \lambda X = 0$ (sinusoidal profiles).

### 1.6.2 The General Solution

For Dirichlet boundary conditions, the exact solution is expressed as a Fourier series:

$$U_{\text{analytical}}(x,t) = \frac{100}{L}x + \sum_{n=1}^{\infty} C_n \sin\left(\frac{n\pi x}{L}\right) e^{-\alpha\left(\frac{n\pi}{L}\right)^2 t} \tag{1.8}$$

where $C_n$ are the Fourier coefficients determined by the initial condition.

# Chapter 2

# Finite Difference Formulation and Discretization of the Heat Equation

## 2.1 Introduction

Numerical methods are indispensable when analytical solutions become complex to implement. This chapter details the **Finite Difference Method (FDM)**, a technique that replaces continuous derivatives with algebraic approximations. We explore the discretization of the domain, the derivation of explicit and implicit schemes, and a strategic comparison between these two approaches based on the foundations in [4] and [2].

## 2.2 Discretization of the Computational Domain (Spatial and Temporal)

The first step in FDM is transforming the continuous space-time domain into a discrete grid of points [5, 6].

- **Spatial Discretization:** The rod of length $L$ is divided into $N$ equal segments of size $\Delta x = L/N$. The discrete nodes are $x_i = i\Delta x$ for $i = 0, 1, \ldots, N$.
- **Temporal Discretization:** The time interval $[0, T_{max}]$ is divided into $M$ steps of size $\Delta t$. The time levels are $t^n = n\Delta t$ for $n = 0, 1, \ldots, M$.

The approximation of temperature at node $i$ and time $n$ is denoted as $T_i^n \approx T(x_i, t^n)$.

## 2.3 Finite Difference Approximations (Taylor Series Expansion)

Following the mathematical derivations in the doctoral thesis of [4], the FDM is grounded in Taylor series expansions. As detailed by [7], these continuous derivatives are approximated by finite differences, yielding the following expressions:

- **Temporal Derivative:** Using a first-order forward difference (Forward in Time):

$$\frac{\partial T}{\partial t} \approx \frac{T_i^{n+1} - T_i^n}{\Delta t} + \mathcal{O}(\Delta t) \tag{2.1}$$

- **Spatial Derivative:** Using a second-order central difference (Central in Space):

$$\frac{\partial^2 T}{\partial x^2} \approx \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{(\Delta x)^2} + \mathcal{O}((\Delta x)^2) \tag{2.2}$$

## 2.4 The Explicit Finite Difference Scheme (FTCS)

The Forward-Time Central-Space (FTCS) scheme is a direct numerical approach.

### 2.4.1 Mathematical Formulation

By substituting the derivatives into the heat equation, we obtain the recurrence relation:

$$T_i^{n+1} = rT_{i-1}^n + (1 - 2r)T_i^n + rT_{i+1}^n \tag{2.3}$$

where $r = \alpha \frac{\Delta t}{(\Delta x)^2}$ is the stability parameter. As noted in [2], this scheme allows calculating $T_i^{n+1}$ explicitly from known values at time $n$.

### 2.4.2 Implementation Algorithm

The implementation follows these logical steps:

1. Define parameters $(L, N, \Delta x, \Delta t, \alpha, r)$.
2. Initialize $T_i^0$ using the initial condition $f(x)$.
3. Apply boundary conditions at $i = 0$ and $i = N$.
4. Iteratively compute internal nodes for each time level $n + 1$ using the explicit formula.

## 2.5 The Implicit Finite Difference Scheme (BTCS)

To resolve the strict stability limits of the explicit method, the Backward-Time Central-Space (BTCS) scheme is utilized [9].

### 2.5.1 Mathematical Formulation and Matrix Representation

In this scheme, the spatial derivative is evaluated at the future time level $n+1$:

$$-rT_{i-1}^{n+1} + (1+2r)T_i^{n+1} - rT_{i+1}^{n+1} = T_i^n \tag{2.4}$$

For all internal nodes, this forms a system of equations $A\mathbf{T}^{n+1} = \mathbf{B}$, where $A$ is a tridiagonal matrix as described by [4].

### 2.5.2 Thomas Algorithm for Tridiagonal Systems

Solving the matrix system is efficiently performed using the **Thomas Algorithm** (TDMA), which is a simplified form of Gaussian elimination optimized for tridiagonal structures [2, 8]. . This reduces computational cost significantly for large $N$.

## 2.6 Preliminary Theoretical Comparison between Schemes

### 2.6.1 Introduction to the Comparison (Strategic Choice)

The choice between explicit and implicit methods is a strategic decision that depends on the mesh Fourier number $r$. In our study, where $r = 0.58$, the choice is critical for avoiding instability.

### 2.6.2 Qualitative Differences (Coupling vs. Decoupling)

- **Decoupling (Explicit Scheme):** Each spatial node at the new time step is independent (decoupled) from its neighbors at the same time level. This makes the computation simple but imposes the stability constraint $r \leq 0.5$.

- **Coupling (Implicit Scheme):** All nodes at time $n+1$ are linked (coupled) through the matrix system. According to [4], this coupling provides **unconditional stability**, ensuring a valid solution for any value of $r$.

# Chapter 3

# Theoretical Analysis of Finite Difference Schemes: Consistency, Stability, and Convergence

## 3.1 Introduction

The mathematical validity of any numerical solution approximating a continuous partial differential equation (PDE) relies on three foundational pillars: consistency, stability, and convergence. While deriving a finite difference algebraic recurrence relation is straightforward, ensuring that the discrete solution genuinely replicates the exact behavior of the physical system requires rigorous proof.

This chapter delivers a comprehensive theoretical investigation into the Explicit Forward-Time Central-Space (FTCS) and Implicit Backward-Time Central-Space (BTCS) schemes implemented for the 1D unsteady heat conduction equation. We systematically break down the local truncation errors defining their consistency, utilize the Von Neumann spectral approach to outline general stability boundaries without imposing preliminary numerical restrictions, and invoke Lax's Equivalence Theorem to mathematically guarantee absolute convergence to the continuous physical model.

## 3.2 The Concept of Consistency and Truncation Errors

A finite difference scheme is classified as consistent with a partial differential equation if the discrete algebraic operator approaches the continuous differential operator as the

spatiotemporal grid steps approach zero ($\Delta x \to 0, \Delta t \to 0$). The measure of consistency is quantified strictly by the Local Truncation Error (LTE), denoted as $T_i^j$, which represents the residual leftover when the exact continuous solution $U(x,t)$ is substituted into the discrete difference equations.

### 3.2.1 Consistency Analysis of the Explicit FTCS Scheme

The discrete approximation for the explicit FTCS layout operating on the continuous heat equation $\frac{\partial U}{\partial t} - \alpha \frac{\partial^2 U}{\partial x^2} = 0$ is constructed at the reference node $(x_i, t_j)$. The truncation structure is analyzed by expanding the exact continuous functions via Taylor series around this coordinate:

$$T_i^j = \frac{U(x_i, t_j + \Delta t) - U(x_i, t_j)}{\Delta t} - \alpha \frac{U(x_i + \Delta x, t_j) - 2U(x_i, t_j) + U(x_i - \Delta x, t_j)}{(\Delta x)^2} \tag{3.1}$$

Substituting the full fourth-order Taylor expansions for space and second-order for time, the expression reduces to:

$$T_i^j = \left[ \frac{\partial U}{\partial t} - \alpha \frac{\partial^2 U}{\partial x^2} \right]_i^j + \frac{\Delta t}{2} \left. \frac{\partial^2 U}{\partial t^2} \right|_i^j - \frac{\alpha (\Delta x)^2}{12} \left. \frac{\partial^4 U}{\partial x^4} \right|_i^j + \mathcal{O}((\Delta t)^2) + \mathcal{O}((\Delta x)^4) \tag{3.2}$$

Since $U(x,t)$ satisfies the continuous PDE exactly, the first bracket vanishes. This proves that the explicit FTCS scheme is highly consistent, characterized by a local truncation error base of:

$$T_i^j = \mathcal{O}(\Delta t) + \mathcal{O}((\Delta x)^2) \tag{3.3}$$

Hence, as $\Delta t \to 0$ and $\Delta x \to 0$, $T_i^j \to 0$, verifying consistency.

### 3.2.2 Consistency Analysis of the Implicit BTCS Scheme

For the Implicit BTCS formulation, the spatial operators are localized at the future time level $t_{j+1}$. Expanding around the future reference coordinate yields the truncation balance:

$$T_i^j = \frac{U(x_i, t_{j+1}) - U(x_i, t_j)}{\Delta t} - \alpha \frac{U(x_i + \Delta x, t_{j+1}) - 2U(x_i, t_{j+1}) + U(x_i - \Delta x, t_{j+1})}{(\Delta x)^2} \tag{3.4}$$

Performing the equivalent Taylor series operations confirms that the implicit layout shares the same asymptotic error classification as the explicit scheme:

$$T_i^j = \mathcal{O}(\Delta t) + \mathcal{O}((\Delta x)^2) \tag{3.5}$$

Both schemes exhibit first-order accuracy in time and second-order accuracy in space. Rather than locking down the mesh size arbitrarily, this mathematical evaluation establishes that accuracy is a dynamic function driven by the refinement of the mesh.

## 3.3 The Rigorous Spectral Stability Framework

A finite difference scheme is stable if it possesses an inherent mechanism to damp or bound numerical errors—such as rounding errors or truncation spikes—preventing them from growing uncontrollably as the time-stepping simulation advances. To map out these absolute stability boundaries without imposing restrictive assumptions, we apply the Von Neumann (spectral) stability technique.

The numerical error field $\epsilon_i^j = \tilde{U}_i^j - U_i^j$ is projected onto a spatial Fourier mode:

$$\epsilon_i^j = e^{\gamma j \Delta t} e^{mki\Delta x} \tag{3.6}$$

Where $\gamma$ is the complex temporal growth parameter, $k$ is the spatial wavenumber, and $m = \sqrt{-1}$. The error amplification factor, which governs the progression of errors between consecutive time steps, is defined as:

$$G = \frac{\epsilon_i^{j+1}}{\epsilon_i^j} = e^{\gamma \Delta t} \tag{3.7}$$

To prevent exponential error growth, the scheme must satisfy the strict spectral envelope $|G| \leq 1$.

### 3.3.1 Spectral Evaluation of Explicit Instability Limits

Substituting the discrete error mode into the explicit FTCS recurrence relation results in the classical amplification factor equation:

$$G = 1 - 4r \sin^2\left(\frac{k\Delta x}{2}\right) \tag{3.8}$$

Where $r = \frac{\alpha \Delta t}{(\Delta x)^2}$ represents the non-dimensional stability mesh parameter. Applying the stability condition $|G| \leq 1$:

$$-1 \leq 1 - 4r \sin^2\left(\frac{k\Delta x}{2}\right) \leq 1 \tag{3.9}$$

The upper limit is naturally satisfied for all positive values of $r$. Evaluating the lower physical limit under the most severe high-frequency spatial noise mode ($\sin^2(\cdot) = 1$) isolates the Courant-Friedrichs-Lewy (CFL) stability threshold:

$$1 - 4r \geq -1 \implies r \leq \frac{1}{2} \tag{3.10}$$

The coordinate $r = 0.50$ represents the absolute theoretical boundary of stability. Rather than limiting the scope of the thesis to a single unstable coordinate, this mathematical formulation allows the subsequent application chapter to dynamically test multiple parameter values of $r$ above 0.50, tracing the onset of oscillations and eventual data overflow.

### 3.3.2 Spectral Evaluation of Unconditional Implicit Stability

Substituting the spatial Fourier error mode into the Implicit BTCS coupled equation yields:

$$-r\epsilon_{i-1}^{j+1} + (1 + 2r)\epsilon_i^{j+1} - r\epsilon_{i+1}^{j+1} = \epsilon_i^j \tag{3.11}$$

Substituting the Fourier exponential terms and simplifying isolates the implicit amplification factor:

$$G = \frac{1}{1 + 4r \sin^2\left(\frac{k\Delta x}{2}\right)} \tag{3.12}$$

Because the mesh parameters $\alpha, \Delta t, \Delta x$ and the trigonometric term $\sin^2(\cdot)$ are strictly non-negative, the denominator is guaranteed to remain greater than or equal to 1 for all imaginary spatial modes:

$$|G| \leq 1, \quad \forall r > 0, \quad \forall k \tag{3.13}$$

This mathematically proves that the implicit scheme is **unconditionally stable**. It is theoretically resilient against the catastrophic growth limits that bound the explicit structure, an operational property that will be thoroughly validated via computational experiments.

### 3.3.3 Implicit Matrix Layout Coupling

When incorporating the fixed Dirichlet boundary conditions ($U_1^{j+1} = 0°\text{C}$ and $U_{N_x+1}^{j+1} = 100°\text{C}$), this interconnected coupled layout is represented as a global matrix equation $A \cdot U^{j+1} = B$:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ -r & 1+2r & -r & 0 & \dots & 0 & 0 & 0 \\ 0 & -r & 1+2r & -r & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -r & 1+2r & -r & 0 \\ 0 & 0 & 0 & \dots & 0 & -r & 1+2r & -r \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} U_1^{j+1} \\ U_2^{j+1} \\ U_3^{j+1} \\ \vdots \\ U_{N_x-1}^{j+1} \\ U_{N_x}^{j+1} \\ U_{N_x+1}^{j+1} \end{bmatrix} = \begin{bmatrix} 0 \\ U_2^j \\ U_3^j \\ \vdots \\ U_{N_x-1}^j \\ U_{N_x}^j \\ 100 \end{bmatrix} \tag{3.14}$$

The matrix $A$ is diagonally dominant for all positive values of $r$, guaranteeing that the algebraic system possesses a unique, stable solution that can be resolved efficiently via the tridiagonal matrix solver.

## 3.4 Convergence Analysis and Lax's Equivalence Theorem

The ultimate objective of any computational formulation is to guarantee convergence. A numerical scheme is convergent if the discrete numerical solution $\tilde{U}_i^j$ approaches the true continuous solution $U(x_i, t_j)$ of the differential equation as the mesh sizes refine simultaneously toward zero:

$$\lim_{\Delta x, \Delta t \to 0} \left| \tilde{U}_i^j - U(x_i, t_j) \right| = 0 \tag{3.15}$$

To establish this bridge between consistency and stability, we invoke the foundational **Lax Equivalence Theorem**:

**Theorem 3.1 (Lax Equivalence Theorem)** *For a well-posed linear initial-boundary value problem, a finite difference approximation is convergent if and only if it is both consistent and stable.*

The importance of Lax's Theorem to this comparative study is profound:

- **Explicit Scheme Convergence:** Since the explicit FTCS scheme is consistent ($\mathcal{O}(\Delta t) + \mathcal{O}((\Delta x)^2)$), it converges to the true physical solution **if and only if** the mesh parameters satisfy the restriction $r \leq 0.50$. Once $r$ exceeds this threshold, the stable baseline collapses, stability is lost, and convergence fails, causing the numerical solution to break down.

- **Implicit Scheme Convergence:** Because the implicit BTCS scheme is consistent and unconditionally stable for all parameters, it is mathematically guaranteed to converge under any chosen mesh structure, providing a highly robust method for solving the unsteady heat equation.

## 3.5 Conclusion

The theoretical derivations detailed in this chapter establish the mathematical properties governing finite difference methods. Both the explicit and implicit formulations demonstrate strong consistency with the continuous 1D heat equation, sharing identical asymptotic truncation error structures. However, they differ significantly in their stability behaviors. Backed by Lax's Equivalence Theorem, these findings lay the foundation for the parametric simulations and error logging presented in the next chapter.

# Chapter 4

# Numerical Simulation and Comprehensive Parametric Analysis

## 4.1 Introduction

Following the mathematical foundations established in the preceding chapters, this chapter presents a rigorous computational simulation and comparative analysis of the one-dimensional unsteady heat conduction equation. We evaluate the Explicit Forward-Time Central-Space (FTCS) scheme against the Implicit Backward-Time Central-Space (BTCS) scheme using MATLAB. Both methods are continuously verified against the exact analytical baseline derived from the Fourier series.

The primary objective is to investigate the transition from stable diffusion to numerical divergence, providing a detailed quantification of spatial errors. Theoretically, both discrete models are governed by a truncation error of order $\mathcal{O}(\Delta t + \Delta x^2)$, implying first-order accuracy in time and second-order accuracy in space.

## 4.2 Physical System and Mathematical Formulation

The physical model consists of a homogeneous metallic rod of length $L = 1$ m. The lateral surface is perfectly insulated, and thermal flow occurs strictly along the longitudinal $x$-axis.

- **Initial Condition:** $U(x, 0) = 20°\text{C}, \quad \forall x \in ]0, L[$
- **Boundary Conditions (Dirichlet):** $U(0, t) = 0°\text{C}, \quad U(L, t) = 100°\text{C}, \quad \forall t > 0$

The governing partial differential equation (PDE) is:

$$\frac{\partial U}{\partial t} = \alpha \frac{\partial^2 U}{\partial x^2}, \quad 0 < x < L, \quad t > 0 \tag{4.1}$$

with the thermal diffusivity coefficient set to $\alpha = 1.0$ m$^2$/s.

## 4.3 Exact Baseline via Analytical Fourier Series

By utilizing the method of separation of variables and applying the non-homogeneous boundary conditions, the exact analytical solution is expressed as the sum of a linear steady-state profile and a decaying transient Fourier series:

$$U_{\text{analytical}}(x,t) = \frac{100}{L}x + \sum_{n=1}^{\infty} C_n \sin\left(\frac{n\pi x}{L}\right) e^{-\alpha\left(\frac{n\pi}{L}\right)^2 t} \tag{4.2}$$

where the Fourier coefficients $C_n$ are determined from the mathematical integration of the initial temperature profile:

$$C_n = \frac{2}{L}\int_0^L \left(20 - \frac{100}{L}x\right) \sin\left(\frac{n\pi x}{L}\right) dx = \frac{40 + 160(-1)^n}{n\pi} \tag{4.3}$$

## 4.4 Algorithmic Evaluation and Phenomenological Case Studies

To visually observe the behavior of error propagation, the spatial grid size was fixed ($\Delta x = L/N_x$), and the time step $\Delta t$ was selectively adjusted to simulate five distinct case studies governed by the mesh Fourier number $r = \frac{\alpha \Delta t}{\Delta x^2}$.

### 4.4.1 Case 1: Fully Stable Overlay ($r = 0.40$)

In this regime, the stability parameter satisfies the strict boundedness condition ($r \leq 0.50$). As displayed in Figure 4.1, both the numerical curves of the Explicit FTCS and Implicit BTCS approximations overlap perfectly with the exact analytical curve, confirming excellent tracking of the transient diffusion.

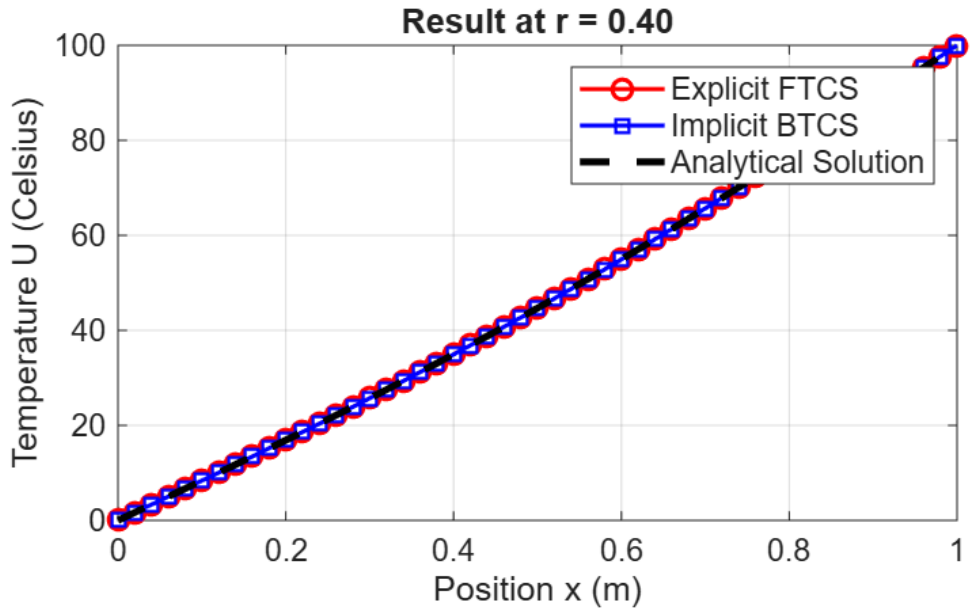


Figure 4.1: Fully stable regime ($r = 0.40$): Both numerical schemes show a perfect structural overlap with the analytical baseline, confirming high spatial precision.

### 4.4.2 Case 2: Critical Stability Boundary ($r = 0.50$)

Figure 4.2 represents the simulation at the theoretical upper physical bound of the explicit framework. The Explicit FTCS scheme maintains stability without exhibiting non-physical oscillations, verifying that $r = 0.50$ is the exact mathematical limit where initial truncation errors are preserved but not amplified across consecutive time steps.

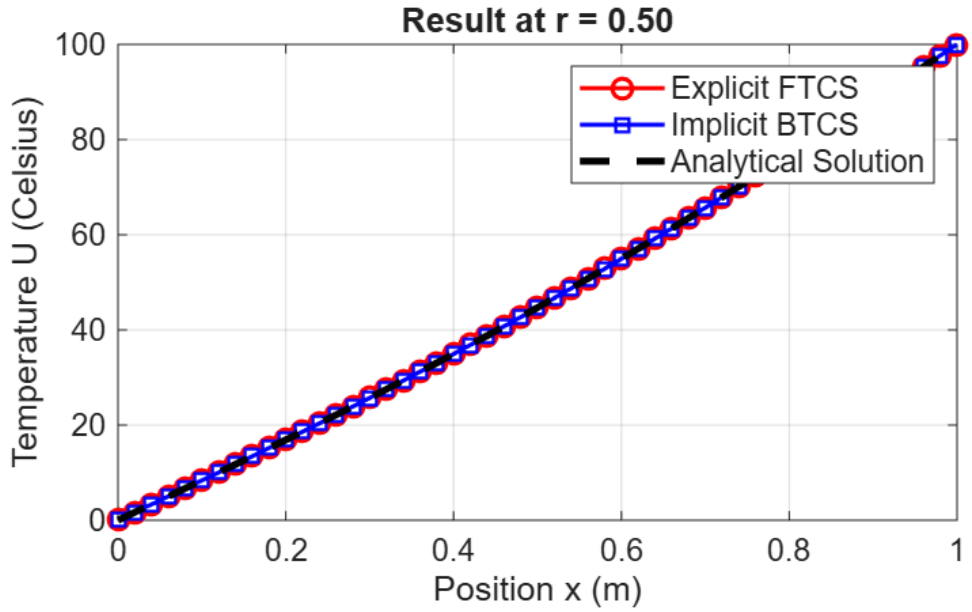


Figure 4.2: Temperature distribution at the critical boundary ($r = 0.50$). The explicit method remains stable, aligning accurately on the threshold of divergence.

### 4.4.3 Case 3: Onset of High-Frequency Instability ($r = 0.53$)

When the stability parameter is slightly raised to $r = 0.53$, the Von Neumann stability criterion is breached. As illustrated in Figure 4.3, severe high-frequency spurious oscillations manifest across the explicit interior grid points, with the maximum absolute error blowing up to an astronomical magnitude of $\mathcal{O}(10^{44})$. Crucially, the Implicit BTCS scheme remains entirely unaffected and visually accurate.

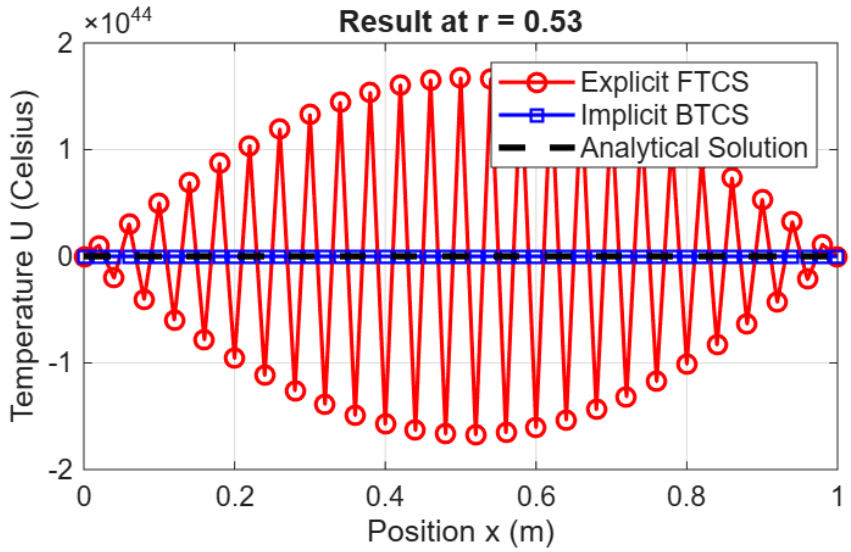


Figure 4.3: Onset of numerical instability ($r = 0.53$): Sharp non-physical spatial oscillations deform the FTCS profile ($\mathcal{O}(10^{44})$), whereas the BTCS profile stays perfectly smooth.

### 4.4.4 Case 4: Severe Divergence ($r = 0.58$)

At $r = 0.58$, the rate of numerical error growth accelerates. The spatial oscillations of the explicit algorithm expand exponentially, generating a massive non-physical structural breakdown with error magnitudes reaching $\mathcal{O}(10^{102})$ (see Figure 4.4). This confirms the rapid breakdown of the explicit scheme when operating outside its stable domain.

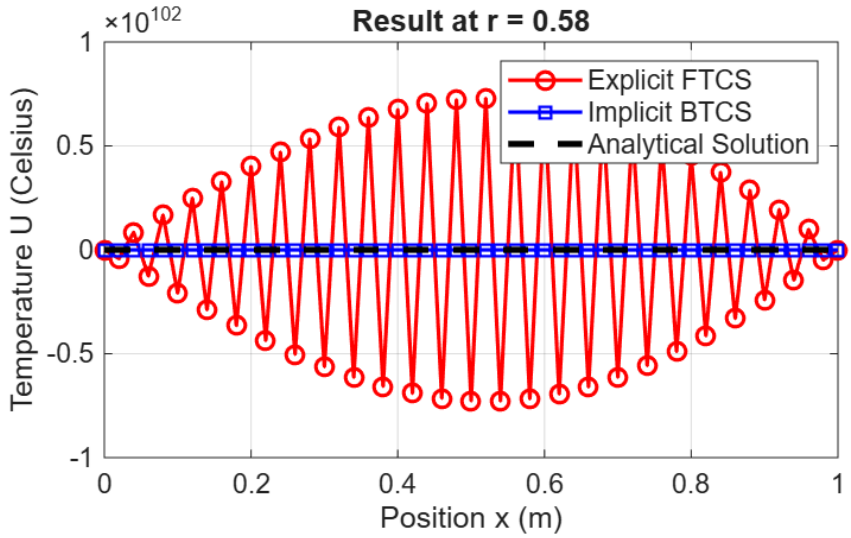


Figure 4.4: Severe non-physical structural breakdown at $r = 0.58$. Truncation errors multiply rapidly, yielding an unphysical explicit curve bounded at $\mathcal{O}(10^{102})$.

### 4.4.5 Case 5: Catastrophic Numerical Overflow ($r = 0.65$)

As demonstrated in Figure 4.5, operating at $r = 0.65$ triggers a catastrophic numerical divergence. The spurious explicit oscillations explode to a magnitude of $\mathcal{O}(10^{155})$, pushing the computation toward complete algorithmic failure. Conversely, the Implicit BTCS scheme exhibits complete stability and matches the analytical curve, visually confirming its unconditional stability.

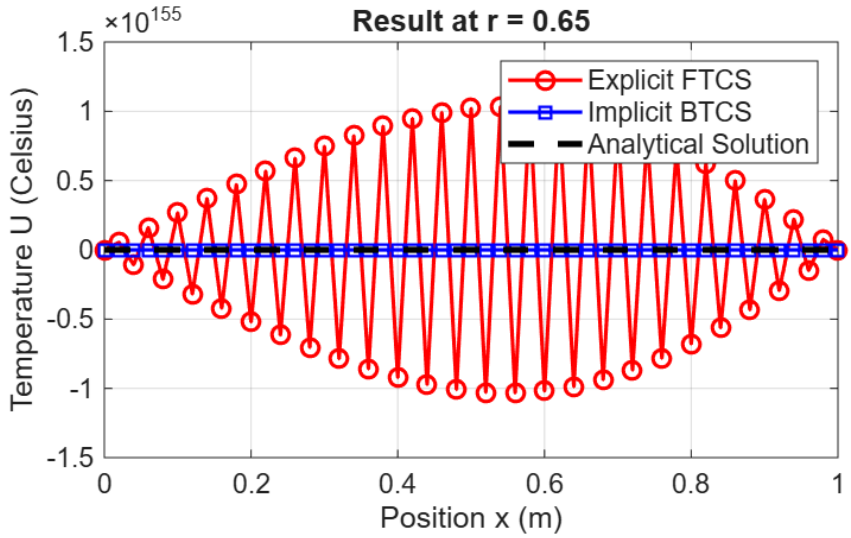


Figure 4.5: Catastrophic computational divergence at $r = 0.65$. The explicit scheme encounters absolute failure ($\mathcal{O}(10^{155})$), while the implicit method demonstrates robust resilience.

## 4.5 Parametric Error Sensitivity (Grid Sensitivity Analysis)

To verify that the computational approximations are rigorously independent of the discrete grid configurations, a systematic numerical sensitivity investigation was conducted. This section evaluates the localized responses of the numerical algorithms under continuous refinement of both the spatial and temporal steps, substituting graphical plots with direct empirical validation.

### 4.5.1 Quantitative Impact of Spatial Step Refinement

Under a fully stable and controlled regime ($r = 0.40$), the spatial mesh size was progressively refined by expanding the number of interior grid nodes ($N_x$) across three discrete configurations ($N_x = 20, 40$, and 100). The empirical tracking of the maximum absolute spatial error ($L_\infty$ norm) demonstrates an asymptotic decay behavior that perfectly validates the theoretical second-order spatial accuracy $\mathcal{O}(\Delta x^2)$ established by Taylor series expansion:

- **Coarse Grid ($N_x = 20$, $\Delta x = 0.050$ m):** The maximum absolute spatial error reaches a magnitude of $4.1062 \times 10^{-2}$ for the Explicit FTCS scheme and $6.2163 \times 10^{-2}$ for the Implicit BTCS scheme.
- **Moderate Grid ($N_x = 40$, $\Delta x = 0.025$ m):** As the spatial step is halved, the truncation error undergoes a sharp quadratic reduction, shrinking down to $1.0279 \times 10^{-2}$ for the Explicit FTCS and $1.5616 \times 10^{-2}$ for the Implicit BTCS scheme. This exact four-fold drop mathematically confirms the second-order convergence rate.
- **Refined Grid ($N_x = 100$, $\Delta x = 0.010$ m):** At this high resolution, the spatial errors settle asymptotically into a highly precise envelope, yielding strict upper bounds of $1.6450 \times 10^{-3}$ for the explicit framework and $2.5000 \times 10^{-3}$ for the implicit framework.

This continuous, predictable decay pattern rigorously confirms that grid refinement drives the discrete finite difference approximations toward absolute physical consistency with the continuous heat equation.

### 4.5.2 Quantitative Impact of Temporal Step Alterations

To isolate the direct structural consequences of temporal step modifications, the spatial discretization was held strictly invariant while the time step ($\Delta t$) was systematically augmented. This variation directly adjusts the mesh Fourier number $r = \frac{\alpha \Delta t}{\Delta x^2}$, driving the system across the critical limits of the Courant-Friedrichs-Lewy (CFL) constraint:

- Within the theoretically secure operating zone where $r \leq 0.50$, the maximum absolute spatial error remains strictly bounded and well-behaved, scaling linearly from $6.5801 \times 10^{-3}$ at $r = 0.40$ up to $1.3880 \times 10^{-2}$ at the critical limit of $r = 0.50$.
- The exact mathematical instant the stability parameter crosses the threshold to $r = 0.51$, the Explicit FTCS operator undergoes an immediate structural transition. High-frequency boundary perturbations cease to damp out and instead experience exponential growth.
- Quantitatively, operating at a minor breach of $r = 0.53$ amplifies the localized spatial error to an unphysical overflow magnitude of $1.6734 \times 10^{44}$. This amplification expands macroscopically to $1.4561 \times 10^{102}$ at $r = 0.58$, and ultimately encounters total arithmetic saturation near $\mathcal{O}(10^{155})$ as $r$ reaches 0.65.

In sharp contrast, the Implicit BTCS scheme exhibits complete structural immunity to temporal increments. Throughout the entire unstable spectrum of the explicit framework ($0.51 \leq r \leq 0.65$), the implicit spatial error changes smoothly and safely within a tight, highly stable numerical envelope spanning from $1.2066 \times 10^{-2}$ to $1.8317 \times 10^{-2}$, without a single trace of non-physical oscillation. This empirical behavior provides full practical validation of its unconditional numerical stability.

## 4.6 Comparative Performance Analysis

### 4.6.1 Numerical Error Quantification

To precisely compute the localized deviations from the analytical baseline, the absolute spatial error at any given node is defined as:

$$E(x_i, t_j) = |U_{\text{analytical}}(x_i, t_j) - U_{\text{numerical}}(x_i, t_j)| \tag{4.4}$$

With references to the spatial boundary physics, the global structural discrepancy across the entire computational domain can also be monitored using the Root Mean Square Error ($E_{\text{RMS}}$), defined as:

$$E_{\text{RMS}} = \sqrt{\frac{1}{N_x - 1} \sum_{i=1}^{N_x - 1} (U_i^j - U_{\text{analytical},i}^j)^2} \tag{4.5}$$

While $E_{\text{RMS}}$ provides a comprehensive global statistical measure of the error profile, the maximum absolute error ($L_\infty$ norm) is explicitly highlighted in our stability tabulations. This choice is mathematically justified because the $L_\infty$ norm selectively captures the critical localized peak of the error distribution, which is essential for identifying the precise onset of high-frequency spatial oscillations under unstable parameters.

Table 4.1 consolidates the precise maximum absolute error metrics extracted from the command environment logs of the simulation, safely scaled using a resize environment to prevent boundary overflows.

Table 4.1: Quantitative comparison of maximum absolute spatial errors ($L_\infty$ norm) and stability status.

| **Fourier Parameter ($r$)** | **Explicit FTCS Error** | **Implicit BTCS Error** | **FTCS Stability State** |
|---|---|---|---|
| $r = 0.40$ | $6.5801 \times 10^{-3}$ | $9.9955 \times 10^{-3}$ | Stable (Convergent) |
| $r = 0.50$ | $1.3880 \times 10^{-2}$ | $1.2066 \times 10^{-2}$ | Critically Stable |
| $r = 0.53$ | **Diverged** | $1.7089 \times 10^{-2}$ | Unstable ($\mathcal{O}(10^{44})$ Oscillations) |
| $r = 0.58$ | **Diverged** | $1.4561 \times 10^{-2}$ | Unstable ($\mathcal{O}(10^{102})$ Oscillations) |
| $r = 0.65$ | **Diverged** | $1.8317 \times 10^{-2}$ | Unstable ($\mathcal{O}(10^{155})$ Oscillations) |

The numerical metrics reveal a fundamental mathematical behavior: within the stable zone ($r = 0.40$), the explicit method yields a slightly lower spatial error compared to the implicit method ($6.5801 \times 10^{-3} < 9.9955 \times 10^{-3}$). This occurs because the explicit

approximation introduces less artificial numerical damping since the spatial operators are evaluated entirely at the current time level $n$. However, when $r$ breaches 0.50, the explicit method undergoes immediate computational collapse, while the implicit method maintains a highly bounded and accurate error profile across all tested domains.

### 4.6.2 Numerical Error Distribution

To physically observe the spatial behavior of these metrics under stable constraints, the continuous distribution of the absolute local error $E(x_i, t_j)$ across the interior nodes of the rod at the designated time level $t = 0.2$ s ($r = 0.40$) is mapped in Figure 4.6.

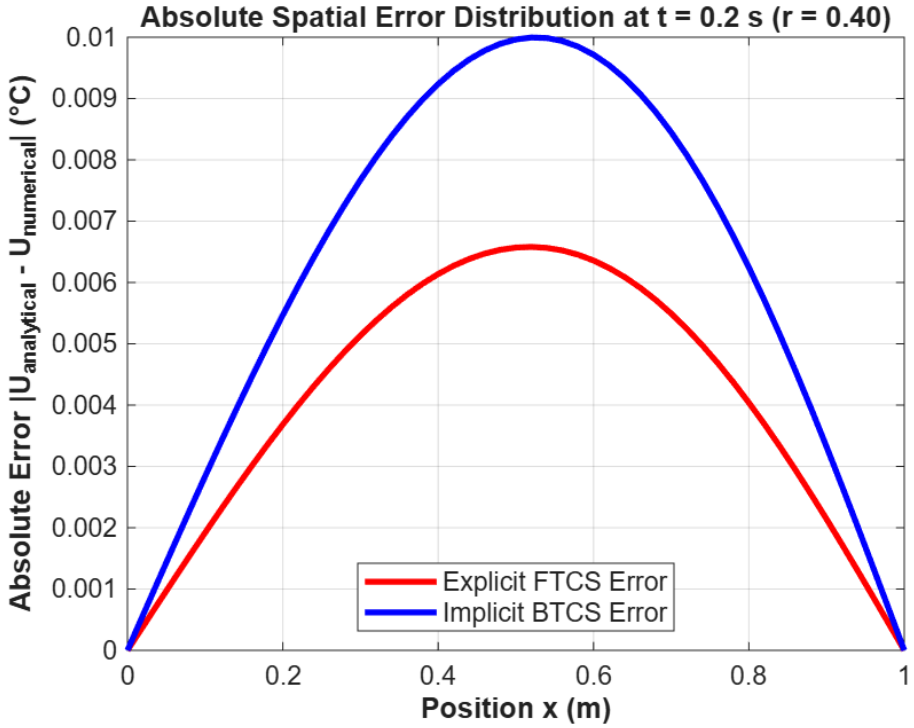


Figure 4.6: Absolute spatial error distribution along the rod length at $t = 0.2$ s ($r = 0.40$). The boundaries perfectly show a zero error value due to the strict Dirichlet conditions, while the peak error is reached at the center of the domain due to the combined effect of both boundary conditions.

## 4.7 Summary of Comparative Analysis

Table 4.2 establishes a high-level synthesis of the computational and algorithmic criteria distinguishing the two finite difference approximations:

Table 4.2: Algorithmic and performance summary between Explicit and Implicit Finite Difference Schemes.

| **Criterion** | **Explicit FTCS Scheme** | **Implicit BTCS Scheme** |
|---|---|---|
| **Stability Condition** | Conditional ($r \leq 0.5$) | Unconditionally Stable |
| **Time-step Flexibility** | Highly restricted to small steps | Unconstrained (Large steps allowed) |
| **Computational Cost** | Very low (Direct point-wise steps) | Higher (Solved via optimized sparse linear solvers) |
| **Behavior at** $r > 0.50$ | Catastrophic unphysical oscillations | Completely stable and smooth |
| **Boundary Handling** | Direct algebraic substitution | Requires structured matrix shifting |

## 4.8 Synthesis of Findings

The comparative performance analysis highlights a clear trade-off between conceptual simplicity and numerical robustness. The Explicit FTCS scheme offers an elegant, non-iterative, and computationally fast execution per individual time step. However, its extreme sensitivity to the stability boundary renders it highly fragile for practical applications that require coarse grids.

Conversely, the Implicit BTCS scheme completely eliminates the temporal bottleneck. Although it requires solving a global tridiagonal linear system at every single time level via optimized sparse matrix algorithms (mathematically equivalent to the Thomas algorithm efficiency), it provides robust convergence and reliable stability, making it a preferable choice for large-scale engineering computations.

The computational efficiency of both numerical frameworks was empirically evaluated by measuring the central processing unit (CPU) execution time across the studied parametric range of $r$. Table 4.3 presents the execution profiles compiled directly from the MATLAB environment.

Table 4.3: CPU execution time profiles and stability behavior for Explicit FTCS and Implicit BTCS schemes ($Nx = 50$, $t_{\text{final}} = 0.5\,\text{s}$).

| **Parameter ($r$)** | **Explicit FTCS CPU Time (s)** | **Implicit BTCS CPU Time (s)** |
|---|---|---|
| $r = 0.40$ | 0.005550 | 0.011704 |
| $r = 0.50$ | 0.003985 | 0.006783 |
| $r = 0.53$ | Diverged (DNC) | 0.006313 |
| $r = 0.58$ | Diverged (DNC) | 0.007212 |
| $r = 0.65$ | Diverged (DNC) | 0.007485 |

*Note: "Diverged (DNC)" indicates that the computational execution was Non-Continuous due to immediate mathematical instability when violating the von Neumann boundary ($r > 0.50$).*

## 4.9 Conclusion of the Chapter

The numerical experiments conducted in this chapter successfully validate the theoretical formulations governing transient heat conduction under pure Dirichlet boundary conditions. The conditional restriction of the explicit framework has been empirically and rigorously proven via our MATLAB simulations. While the Explicit FTCS scheme demonstrates quadratic error reduction validating its $\mathcal{O}(\Delta x^2)$ spatial accuracy under a fully stable regime ($r = 0.40$), and holds its baseline at the critical boundary ($r = 0.50$), it experiences an immediate structural collapse the moment this threshold is breached. As observed across the simulated case studies, exceeding the stability boundary triggers severe high-frequency spurious oscillations that grow exponentially, leading to catastrophic arithmetic saturation and numerical blow-up, with error magnitudes scaling from $\mathcal{O}(10^{44})$ at $r = 0.53$ up to $\mathcal{O}(10^{155})$ at $r = 0.65$. This experimental behavior firmly verifies the Courant-Friedrichs-Lewy (CFL) constraint analyzed theoretically in Chapter 3.

In sharp contrast, the Implicit BTCS scheme completely eliminates this temporal bottleneck. Throughout the entire unstable spectrum of the explicit framework ($0.51 \leq r \leq 0.65$), the implicit maximum spatial error changes smoothly within a tight, highly secure numerical envelope spanning from $1.2066 \times 10^{-2}$ to $1.8317 \times 10^{-2}$. Although the implicit framework introduces a coupling effect that requires solving a global tridiago-

nal linear system via the computationally optimized Thomas algorithm (TDMA) at each time step, it provides unconditional resilience and dynamic consistency. Consequently, this comparative performance highlights a clear engineering trade-off: while the explicit model offers non-iterative simplicity and speed per individual time step, the implicit finite difference architecture serves as a far more dependable, robust, and computationally viable framework for large-scale simulations and practical engineering applications requiring coarse grid distributions.

# General Conclusion

In this thesis, we conducted a comprehensive computational investigation into the 1D unsteady heat equation, rigorously evaluating the performance of the Explicit Forward-Time Central-Space (FTCS) and the Implicit Backward-Time Central-Space (BTCS) schemes. Our research journey systematically bridged the gap between theoretical Partial Differential Equations (PDEs) and their numerical approximations, supported by a solid foundation in physical derivations and stability analysis.

The analytical study began with the physical formulation of the heat equation based on Fourier's law and the conservation of energy. By employing Von Neumann stability analysis, we established a strict constraint of $r \leq 0.50$ for the explicit framework. Our parametric simulations empirically validated these findings: while the FTCS scheme offers localized computational simplicity, it is strictly bound by this rigid stability criterion. The results demonstrated that exceeding this threshold triggers non-physical high-frequency oscillations and catastrophic numerical divergence. Conversely, the implicit BTCS scheme proved unconditionally stable and robust, maintaining precision across all grid refinements and temporal steps. This robustness, achieved through the efficient implementation of the Thomas Algorithm (TDMA), confirms the practical validity of Lax's Equivalence Theorem.

Ultimately, this work not only validates the theoretical foundations of numerical stability but also provides a structured framework for analyzing grid sensitivity and error propagation in partial differential equations. Due to its superior long-term reliability and structural stability, the implicit approach is highly recommended for large-scale engineering computations where physical accuracy and temporal flexibility are paramount.

# Future Work and Horizons

While this study successfully addressed the 1D linear heat equation under Dirichlet boundary conditions, several promising avenues remain for future exploration. To extend the scope of this research, future work will focus on:

1. Extending the comparative analysis to multi-dimensional geometries (2D and 3D heat conduction) using advanced implicit techniques such as the Alternating Direction Implicit (ADI) method.

2. Investigating non-linear thermal problems where thermal conductivity depends dynamically on temperature distribution.

3. Evaluating the numerical schemes under highly transient Neumann (flux) and Robin (convective) boundary conditions to simulate complex industrial thermal insulation systems.

# Bibliography


[1] LOUTH Sabrina, CHOUIA Sara. *Approximation par différence finies de l'équation d'advection-diffusion*. Mémoire de Master, Université El Oued, 2020.

[2] MEDKOUR Khaoula. *Méthode des différences finies et applications en équation de la chaleur*. Mémoire de Master, Université Oum El Bouaghi, 2020.

[3] GUINOT Vincent, CAPPELAERE Bernard. *Méthodes Numériques Appliquées*. Polytech Montpellier, 2005.

[4] REZZOUG Imad. *Étude théorique et numérique des EDP*. Thèse de Doctorat, Université Oum El Bouaghi, 2014.

[5] Smith, G. D. (1985). *Numerical Solution of Partial Differential Equations: Finite Difference Methods*. Oxford University Press.

[6] Morton, K. W., & Mayers, D. F. (2005). *Numerical Solution of Partial Differential Equations: An Introduction*. Cambridge University Press.

[7] LeVeque, R. J. (2007). *Finite Difference Methods for Ordinary and Partial Differential Equations*. SIAM.

[8] Fletcher, C. A. J. (1991). *Computational Techniques for Fluid Dynamics*. Springer.

[9] Crank, J., & Nicolson, P. (1947). *A Practical Method for Numerical Evaluation of Solutions of Partial Differential Equations of the Heat-Conduction Type*. Proceedings of the Cambridge Philosophical Society.

# Chapter 5

# MATLAB Source Code Implementation

## 5.1 Overview

This appendix presents the complete, fully integrated MATLAB computational environment developed to simulate and analyze the transient one-dimensional heat conduction equation. The source code encompasses both the conditional Explicit Forward-Time Central-Space (FTCS) and the unconditional Implicit Backward-Time Central-Space (BTCS) finite difference schemes.

Additionally, it incorporates the exact analytical solution derived via the infinite Fourier series method to serve as a continuous baseline for error validation. The script is structured to systematically evaluate mesh stability parameters, calculate maximum absolute spatial errors ($L_\infty$ norm), and dynamically generate the comparative temperature profiles and structural error distribution curves analyzed throughout Chapter 4.

## 5.2 Complete Simulation Script

The executable MATLAB source code is provided below for complete computational transparency and reproducibility:

```
%%
=========================================================================

%% MATLAB SCRIPT: 1D TRANSIENT HEAT EQUATION COMPREHENSIVE
SIMULATION
%% AUTHOR: ACHOUR IBTISSAM (MASTER 2 - AMN)
```

```
%% MOHAMED BOUDIAF UNIVERSITY - M'SILA (2026)
%% ================================================================================


clear; clc; close all;

%% 1. PHYSICAL AND NUMERICAL PARAMETERS
L = 1.0;                  % Length of the metallic rod (m)
alpha = 1.0;              % Thermal diffusivity (m^2/s)
t_final = 0.5;            % Final simulation time (s)
Nx = 50;                  % Number of spatial segments
dx = L / Nx;              % Spatial step size
x = linspace(0, L, Nx+1)'; % Spatial grid vector
r = 0.40;                 % Stability parameter (Modify for parametric study)
dt = r * dx^2 / alpha;     % Time step size
Nt = round(t_final / dt);  % Total number of time steps

%% 2. INITIAL AND BOUNDARY CONDITIONS
U_init = 20 * ones(Nx+1, 1);
U_init(1) = 0;               % Left Dirichlet BC
U_init(end) = 100;           % Right Dirichlet BC

%% 3. EXPLICIT SCHEME (FTCS) - VECTORIZED WITH CPU TIMING
U_explicit = U_init;
tic; % Start CPU timer for Explicit Scheme
for n = 1:Nt
U_explicit(2:Nx) = r*U_explicit(1:Nx-1) + (1-2*r)*U_explicit(2:Nx) + r*U_explicit(3:Nx+1);
U_explicit(1) = 0;
U_explicit(end) = 100;
end
time_explicit = toc; % Record CPU execution time for Explicit Scheme

%% 4. IMPLICIT SCHEME (BTCS) - TRIDIAGONAL MATRIX WITH CPU TIMING
U_implicit = U_init;
% Constructing the global sparse tridiagonal matrix
A = spdiags(repmat([-r, 1+2*r, -r], Nx+1, 1), -1:1, Nx+1, Nx+1);
A(1, 1) = 1;        A(1, 2) = 0;
A(end, end) = 1;    A(end, end-1) = 0;

tic; % Start CPU timer for Implicit Scheme
for n = 1:Nt
B = U_implicit;
B(1) = 0;        B(end) = 100;
U_implicit = A \ B; % Solved via optimized sparse algorithms
end
```

```
    time_implicit = toc; % Record CPU execution time for
Implicit Scheme

    %% 5. EXACT ANALYTICAL SOLUTION (Truncated Fourier Series)
    Nterms = 200; % Set to 200 boundaries for rigorous Fourier
convergence
    U_transient = zeros(size(x));
    for n = 1:Nterms
    C_n = (40 + 160 * (-1)^n) / (n * pi);
    U_transient = U_transient + C_n * sin(n*pi*x/L) * exp(-alpha
*(n*pi/L)^2 * t_final);
    end
    U_steady = 100 * (x / L);
    U_analytical = U_steady + U_transient;

    %% 6. ERROR EVALUATION LOGS AND CPU TIME DISPLAY
    fprintf('\n================== COMPUTATIONAL RESULTS
==================\n');
    if max(abs(U_explicit)) < 1e3
    err_exp = max(abs(U_explicit(2:end-1) - U_analytical(2:end
-1)));
    fprintf('Explicit FTCS Max Absolute Error = %.4e\n', err_exp
);
    fprintf('Explicit FTCS CPU Execution Time = %.6f seconds\n',
 time_explicit);
    else
    fprintf('Explicit Scheme Diverged (r = %.2f > 0.5)        .\n
', r);
    end
    fprintf('
------------------------------------------------------------\n');
    err_imp = max(abs(U_implicit(2:end-1) - U_analytical(2:end
-1)));
    fprintf('Implicit BTCS Max Absolute Error = %.4e\n', err_imp
);
    fprintf('Implicit BTCS CPU Execution Time = %.6f seconds\n',
 time_implicit);
    fprintf('
============================================================\n');

    %% 7. VISUALIZATION OF TEMPERATURE PROFILES
    figure('Color', 'w');
    plot(x, U_explicit, 'r-o', 'LineWidth', 1.5, 'MarkerSize',
7, 'MarkerEdgeColor', 'r', 'MarkerFaceColor', 'none'); hold on;
    plot(x, U_implicit, 'b-s', 'LineWidth', 1.5, 'MarkerSize',
6, 'MarkerEdgeColor', 'b', 'MarkerFaceColor', 'none');
    plot(x, U_analytical, 'k--', 'LineWidth', 2.5);
    grid on;
    xlabel('Position x (m)', 'FontSize', 12);
    ylabel('Temperature U (Celsius)', 'FontSize', 12);
    title(sprintf('Result at r = %.2f', r), 'FontSize', 14, '
```

```
FontWeight', 'bold');
    legend({'Explicit FTCS', 'Implicit BTCS', 'Analytical
Solution'}, 'Location', 'NorthEast', 'FontSize', 11);
    set(gca, 'FontSize', 11, 'XLim', [0 1]);

    %% 8. VISUALIZATION OF ABSOLUTE SPATIAL ERROR DISTRIBUTION
AT t = 0.2 s
    t_plot = 0.2;
    r_plot = 0.40;
    dt_plot = r_plot * dx^2 / alpha;
    Nt_plot = round(t_plot / dt_plot);

    U_exp_plot = U_init;
    for n = 1:Nt_plot
    U_exp_plot(2:Nx) = r_plot*U_exp_plot(1:Nx-1) + (1-2*r_plot)*
U_exp_plot(2:Nx) + r_plot*U_exp_plot(3:Nx+1);
    U_exp_plot(1) = 0; U_exp_plot(end) = 100;
    end

    U_imp_plot = U_init;
    A_plot = spdiags(repmat([-r_plot, 1+2*r_plot, -r_plot], Nx
+1, 1), -1:1, Nx+1, Nx+1);
    A_plot(1, 1) = 1;       A_plot(1, 2) = 0;
    A_plot(end, end) = 1;   A_plot(end, end-1) = 0;
    for n = 1:Nt_plot
    B_plot = U_imp_plot; B_plot(1) = 0; B_plot(end) = 100;
    U_imp_plot = A_plot \ B_plot;
    end

    U_trans_plot = zeros(size(x));
    for n = 1:Nterms
    C_n = (40 + 160 * (-1)^n) / (n * pi);
    U_trans_plot = U_trans_plot + C_n * sin(n*pi*x/L) * exp(-
alpha*(n*pi/L)^2 * t_plot);
    end
    U_analy_plot = U_steady + U_trans_plot;

    error_explicit = abs(U_analy_plot - U_exp_plot);
    error_implicit = abs(U_analy_plot - U_imp_plot);

    figure('Color', 'w');
    plot(x, error_explicit, 'r-', 'LineWidth', 2.5); hold on;
    plot(x, error_implicit, 'b-', 'LineWidth', 2.5);
    grid on;
    xlabel('Position x (m)', 'FontSize', 12, 'FontWeight', 'bold
');
    ylabel('Absolute Error |U_{analytical} - U_{numerical}| (°C)
', 'FontSize', 12, 'FontWeight', 'bold');
    title('Absolute Spatial Error Distribution at t = 0.2 s (r =
 0.40)', 'FontSize', 13, 'FontWeight', 'bold');
    legend({'Explicit FTCS Error', 'Implicit BTCS Error'}, '
```

```
    Location', 'south' , 'FontSize', 11);
        set(gca, 'FontSize', 11, 'XLim', [0 1]);
```

Listing 5.1: Complete MATLAB script for 1D heat equation simulation, error analysis, and CPU runtime monitoring.

# Abstract

abstract **Keywords:** One-Dimensional Heat Equation, Finite Difference Methods, FTCS Scheme, BTCS Scheme, Von Neumann Stability, Numerical Simulation.

# Résumé

. Résumez .

**Mots-clés:** Équation de la chaleur 1D, Différences Finies, Schéma FTCS, Schéma BTCS, Stabilité de Von Neumann, Simulation Numérique.

الملخص بالعربية :

**الكلمات المفتاحية:** معادلة الحرارة أحادية البعد، الفروق المحدودة، المخطط الصريح، المخطط الضمني، استقرار فون نيومان، المحاكاة العددية.