



# Solution of Heat Equation using Finite Difference Method

Anirudh Singh

February 2, 2025

## Abstract

In this report, we will solve the heat equation using finite difference method. We will use both implicit and explicit schemes to solve the heat equation. We will compare the results obtained from both the schemes and discuss the stability of the schemes. We will also discuss the convergence of the schemes.

## 1 Introduction

The heat equation is a parabolic partial differential equation that describes the distribution of heat (or variation in temperature) in a given region over time. The heat equation is given by

## 2 Explicit Scheme

The heat equation is given by

$$\frac{\partial u}{\partial t} = \alpha \nabla^2 u \quad (1)$$

where  $\alpha$  is the thermal diffusivity. We will solve the heat equation using finite difference method. We will use explicit scheme to solve the heat equation. The explicit scheme is given by

## 3 Discretization Using Finite Difference Method

The derivatives are approximated as follows:

### 3.1 Time Discretization (Forward Difference)

$$\frac{\partial u}{\partial t} \approx \frac{u_i^{n+1} - u_i^n}{\Delta t} \quad (2)$$

### 3.2 Space Discretization (Central Difference)

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} \quad (3)$$

Substituting these into the heat equation:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \alpha \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} \quad (4)$$

Rearranging for  $u_i^{n+1}$ :

$$u_i^{n+1} = u_i^n + \lambda (u_{i+1}^n - 2u_i^n + u_{i-1}^n) \quad (5)$$

where  $\lambda = \frac{\alpha \Delta t}{\Delta x^2}$ .

## 4 Algorithm

1. **Initialize:** Define the spatial domain  $x \in [0, L]$  and time domain  $t \in [0, T]$ .
2. **Discretize:** Choose step sizes  $\Delta x$  and  $\Delta t$  such that the stability condition  $\lambda \leq 0.5$  is satisfied.
3. **Set Initial Condition:** Define  $u(x, 0)$  for all spatial points.
4. **Time Stepping:** For each time step  $n$ :
  - (a) Compute the new temperature values using:

$$u_i^{n+1} = u_i^n + \lambda (u_{i+1}^n - 2u_i^n + u_{i-1}^n) \quad (6)$$

- (b) Apply boundary conditions as required.

5. **Repeat Until**  $t = T$ .

## 5 Implicit Scheme

The implicit scheme for solving the heat equation is given by To numerically solve this equation, we employ the implicit finite difference method, leading to a system of linear equations that can be solved efficiently using the Thomas algorithm.

## 6 Numerical Method

### 6.1 Finite Difference Discretization

Using the implicit backward Euler method, the heat equation is discretized as:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \alpha \frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{\Delta x^2}. \quad (7)$$

Rearranging,

$$-\beta u_{i-1}^{n+1} + (1 + 2\beta)u_i^{n+1} - \beta u_{i+1}^{n+1} = u_i^n, \quad (8)$$

where  $\beta = \frac{\alpha \Delta t}{\Delta x^2}$ . This forms a tridiagonal system that can be solved using the Thomas algorithm.

### 6.2 Tridiagonal Matrix Formulation

The discretized system can be written in matrix form as:

$$\begin{bmatrix} 1 + 2\beta & -\beta & 0 & \dots & 0 \\ -\beta & 1 + 2\beta & -\beta & \dots & 0 \\ 0 & -\beta & 1 + 2\beta & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & -\beta & 1 + 2\beta \end{bmatrix} \begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \\ \vdots \\ u_N^{n+1} \end{bmatrix} = \begin{bmatrix} u_1^n \\ u_2^n \\ u_3^n \\ \vdots \\ u_N^n \end{bmatrix}. \quad (9)$$

## 6.3 Thomas Algorithm

The tridiagonal system is expressed as:

$$Au^{n+1} = u^n, \quad (10)$$

where  $A$  is a tridiagonal matrix with elements:

- Lower diagonal:  $a_i = -\beta$
- Main diagonal:  $b_i = 1 + 2\beta$
- Upper diagonal:  $c_i = -\beta$
- Right-hand side:  $d_i = u_i^n$

The Thomas algorithm consists of:

**Forward elimination:**

$$w = \frac{a_i}{b'_{i-1}} \quad (11)$$

$$b'_i = b_i - wc_{i-1} \quad (12)$$

$$d'_i = d_i - wd'_{i-1} \quad (13)$$

**Backward substitution:**

$$u_N = \frac{d'_N}{b'_N} \quad (14)$$

$$u_i = \frac{d'_i - c_i u_{i+1}}{b'_i} \quad (\text{for } i = N - 1 \text{ to } 1). \quad (15)$$

The Thomas algorithm efficiently solves the tridiagonal system arising from the implicit finite difference formulation of the heat equation. This method ensures stability, allowing larger time steps compared to explicit schemes. Further extensions include two-dimensional cases and adaptive time stepping.

## Questions

1. Compare the residual of the equation with time for implicit and explicit methods for different  $\beta$  values.
2. Can you get a solution for any  $\Delta t$ ? Show this by taking another value for  $\Delta x$ .
3. For a given  $\Delta x$  and  $\Delta t$ , how much time does the explicit method and implicit method take to reach a steady state? Compare.

## Answers

**Ans 1.** The problem asks to compare the residual of the equation over time for both the implicit and explicit methods, considering different values of  $\beta$ . The residual indicates the error between the numerical solution and the exact solution at each time step. It provides a measure of how well the numerical method is approximating the actual solution.

## Implicit Method

In the implicit method, the solution at time step  $n + 1$  is determined by solving a system of equations that involve values at both the current and next time steps. The method is unconditionally stable, meaning that it can handle larger time steps ( $\Delta t$ ) without becoming unstable. As a result, the residual for the implicit method tends to decrease over time, even for larger values of  $\beta$ . However, this method is computationally more expensive due to the need for solving a system of equations at each time step.

## Explicit Method

The explicit method computes the solution at the next time step based on known values at the current time step. Although it is computationally less expensive than the implicit method, the explicit method is conditionally stable, meaning that the time step  $\Delta t$  must be sufficiently small to avoid instability. If  $\beta$  is large or if  $\Delta t$  is too large, the residual in the explicit method can become large, reflecting poor accuracy and possible instability.

## Effect of $\beta$ on Residuals

The parameter  $\beta$  (which could be related to the CFL number or thermal diffusivity) affects the stability and accuracy of both methods:

- For large values of  $\beta$ , the explicit method may become unstable if  $\Delta t$  is not small enough, leading to large residuals. In contrast, the implicit method can handle larger values of  $\beta$  without instability, and the residuals remain relatively small.
- For small values of  $\beta$ , both methods can maintain stability, and the residuals in both cases will be small. The difference between the methods becomes less significant.

## Comparison of Residuals Over Time

The residuals over time for the implicit and explicit methods behave differently:

- The residual for the implicit method decreases over time, even for larger values of  $\beta$ . This is because the implicit method is stable and can handle larger time steps.
- The residual for the explicit method may initially be small, but as time progresses, especially for larger  $\beta$  values, the residual can increase or oscillate, reflecting instability or poor accuracy unless a very small time step is used.

The explicit method is computationally less expensive, but its residual increases or oscillates for larger values of  $\beta$ , especially when the time step is not small enough. The implicit method, while more expensive to compute, remains stable and has smaller residuals even for larger  $\beta$  values and larger time steps.

Thus, the implicit method generally performs better in terms of stability and accuracy for larger values of  $\beta$ , while the explicit method requires smaller time steps to maintain stability and low residuals.

**Ans 2.** The second question asks whether a solution can be obtained for any time step  $\Delta t$ . Specifically, it asks us to demonstrate this by taking another value for  $\Delta x$ .

## Solution Existence for Any $\Delta t$

In numerical methods for solving partial differential equations such as the heat equation, the existence of a solution depends on the stability of the method. The stability of the explicit and implicit methods is affected by the choice of the time step  $\Delta t$  and spatial step  $\Delta x$ .

### Explicit Method Stability

For the explicit method, the time step  $\Delta t$  is constrained by a stability condition. The well-known Courant–Friedrichs–Lewy (CFL) condition for the explicit method is given by:

$$\lambda = \frac{\alpha \Delta t}{\Delta x^2} \leq \frac{1}{2}$$

where  $\alpha$  is the thermal diffusivity, and  $\Delta x$  and  $\Delta t$  are the spatial and temporal steps, respectively. This condition ensures that the explicit method remains stable.

If  $\Delta t$  is too large relative to  $\Delta x$ , the explicit method may become unstable, leading to divergent solutions. Therefore, the solution cannot be obtained for arbitrary values of  $\Delta t$  if this stability condition is violated.

### Implicit Method Stability

The implicit method, on the other hand, is unconditionally stable. This means that no matter how large the time step  $\Delta t$  is, the method will always provide a stable solution, as long as the system of equations can be solved. The implicit method does not have the strict stability constraint that the explicit method does, so it can handle much larger values of  $\Delta t$  compared to the explicit method.

### Effect of Changing $\Delta x$

When we change  $\Delta x$ , we affect the spatial resolution of the grid. Smaller values of  $\Delta x$  lead to more grid points, which generally result in a more accurate solution. If  $\Delta x$  is reduced, the CFL condition for the explicit method requires that  $\Delta t$  also be reduced to maintain stability.

For the implicit method, changing  $\Delta x$  has less of an impact on the stability or solution process, as it is unconditionally stable. However, reducing  $\Delta x$  will require solving a larger system of equations at each time step, making the computation more expensive.

- The explicit method has a stability condition that limits the size of  $\Delta t$  relative to  $\Delta x$ . Therefore, a solution can only be obtained for time steps that satisfy the CFL condition. If  $\Delta t$  is too large for a given  $\Delta x$ , the solution will not be stable.
- The implicit method, being unconditionally stable, allows for larger  $\Delta t$  values, meaning a solution can be obtained for any  $\Delta t$ , regardless of the choice of  $\Delta x$ .

Thus, a solution can be obtained for any  $\Delta t$  using the implicit method, but the explicit method requires careful selection of  $\Delta t$  to ensure stability.

**Ans 3.** The question asks: "For a given  $\Delta x$  and  $\Delta t$ , how much time does the explicit method and implicit method take to reach the steady state? Compare."

## Steady State in Heat Equation

In the context of the heat equation, the *steady state* is reached when the temperature distribution no longer changes with time. Mathematically, this occurs when the solution at the next time step is nearly identical to the solution at the current time step, meaning the residual is close to zero. The time required for both methods to reach the steady state depends on the stability and accuracy of the method, as well as the chosen values for  $\Delta x$  and  $\Delta t$ .

## Time to Reach Steady State Using Explicit Method

The explicit method requires smaller time steps for stability, which can significantly slow down the convergence to the steady state. The time step  $\Delta t$  is constrained by the CFL condition:

$$\lambda = \frac{\alpha \Delta t}{\Delta x^2} \leq \frac{1}{2}$$

Due to this condition, the explicit method tends to take more time to reach the steady state when compared to the implicit method, especially if  $\Delta x$  is small. This means that for large grids or high-resolution problems, the explicit method will require numerous time steps to achieve the steady state.

## Time to Reach Steady State Using Implicit Method

The implicit method, on the other hand, is unconditionally stable, which means it can take larger time steps without violating stability. As a result, it typically reaches the steady state much faster than the explicit method. Even with larger  $\Delta t$ , the implicit method can achieve stable solutions without the need for excessive time step reductions, making it more efficient in terms of computational time for reaching steady state.

While the implicit method may require more computation per time step due to solving a system of equations, the overall number of time steps is generally smaller, leading to faster convergence to the steady state.

## Comparison

- The **explicit method** requires smaller time steps to ensure stability and takes more time to reach the steady state. The number of time steps is higher because of the strict stability condition.
- The **implicit method** can take larger time steps and reaches the steady state more quickly. While each time step is more computationally expensive due to solving a system of equations, the total computational effort is typically smaller because fewer time steps are required.

In summary, the implicit method generally reaches the steady state more quickly than the explicit method for a given  $\Delta x$  and  $\Delta t$ . This is due to the larger permissible time steps and the unconditionally stable nature of the implicit method. While the explicit method requires smaller time steps for stability, it requires more time steps overall to converge to the steady state, making it slower in comparison.