HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Project report:

2D HEAT EQUATION

Guided by:

Associate Professor Vu Van Thieu

Group members:

Do Khanh Nam - 20225988

Ngo Anh Tu - 20226005

Vu Binh Minh - 20226058

Contents

Abstract

This project examines numerical solutions for the 2-dimensional heat equation using explicit and implicit finite difference methods. The explicit method, while simple, requires small time steps for stability. The implicit method, more complex, allows larger time steps and offers unconditional stability. By deriving, implementing, and comparing both methods, we assess their accuracy, stability, and efficiency, providing guidance for selecting the appropriate method for different scenarios.

1 Introduction

1.1 The 2-D Heat Equation

The 2-D heat equation T(x, y, t) describes the distribution of heat (or temperature) in a two-dimensional domain over time. The rate of change temperature changes over time is proportional to the temperature distribution in space. Hence, we have the 2D heat equation can be expressed as:

$$\frac{\partial T}{\partial t} = \alpha \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right)$$

- T is the temperature
- t is time
- x and y are the spatial coordinates
- α is the thermal diffusivity of the material

1.2 Boundary Conditions

Boundary conditions are essential for defining the 2-D heat equation and ensuring a unique solution. The most common boundary conditions are:

- Dirichlet Boundary Condition
- Neumann Boundary Condition

1.2.1 Dirichlet Boundary Condition

This condition specifies the temperature at the boundaries of the domain, i.e..
T(n) = f(t) on the boundary, where n is the normal to the boundary.

1.2.2 Neumann Boundary Condition

This condition specifies the heat flux (rate of heat transfer) at the boundaries, i.e..
$\frac{\partial T}{\partial n} = f(t)$ on the boundary, where n is the normal to the boundary.

1.3 Initial Condition

The initial condition specify the temperature distribution over the 2-dimensional domain at the start of the heat transfer process.

$$T(x, y, 0) = f(x, y)$$

$f(x, y)$ is the initial condition function

2 Discretization

We can divide the 2D domain into a number of grid points. Each point on the domain have a coordinate (x, y). We can also separate the time domain into many time step. Hence, the equation T(x, y, t) can be represented as:

$$T(x, y, t) = T_{i,j}^n$$

- n is the time step
- i and j represent the spatial position

3 Explicit method

The explicit method uses a Forward Difference at time n and a second-order central difference for the space derivative:

$$\frac{\partial T}{\partial t} = \frac{T_{i,j}^{n+1} - T_{i,j}^n}{\Delta t} \tag{1}$$

$$\frac{\partial^2 T}{\partial x^2} = \frac{T_{i+1,j}^n - 2T_{i,j}^n + T_{i-1,j}^n}{(\Delta x)^2} \tag{2}$$

$$\frac{\partial^2 T}{\partial y^2} = \frac{T_{i,j+1}^n - 2T_{i,j}^n + T_{i,j-1}^n}{(\Delta y)^2} \tag{3}$$

Substituting (1), (2) and (3), we have

$$\frac{T_{i,j}^{n+1} - T_{i,j}^n}{\Delta t} = \alpha \left(\frac{T_{i+1,j}^n - 2T_{i,j}^n + T_{i-1,j}^n}{(\Delta x)^2} + \frac{T_{i,j+1}^n - 2T_{i,j}^n + T_{i,j-1}^n}{(\Delta y)^2} \right)$$

Let $\Delta x = \Delta y = \Delta h$, we have

$$\frac{T_{i,j}^{n+1} - T_{i,j}^n}{\Delta t} = \alpha \left(\frac{T_{i+1,j}^n + T_{i-1,j}^n + T_{i,j+1}^n + T_{i,j-1}^n - 4T_{i,j}^n}{(\Delta h)^2} \right)$$

$$<=> T_{i,j}^{n+1} = \left(1 - \frac{4\Delta t\alpha}{h^2}\right) T_{i,j}^n + \Delta t\alpha \left(\frac{T_{i+1,j}^n + T_{i-1,j}^n + T_{i,j+1}^n + T_{i,j-1}^n}{(\Delta h)^2} \right)$$

Let $\gamma = \frac{\alpha \Delta t}{(\Delta h)^2}$, we have

$$T_{i,j}^{n+1} = (1 - 4\gamma)\, T_{i,j}^n + \gamma \left(T_{i+1,j}^n + T_{i-1,j}^n + T_{i,j+1}^n + T_{i,j-1}^n \right) \tag{4}$$

With this recurrence relation, and knowing the values at time n, we can obtain the corresponding values at time n+1.

The explicit method is known to be numerically stable and convergent whenever:

$$\gamma \leq \frac{1}{4}$$

$$<=> \Delta t \leq \frac{(\Delta h)^2}{4\alpha}$$

The truncation errors are proportional to the time step and the square of the space step:

$$\epsilon = O(\Delta t) + O((\Delta h)^2)$$

4 Implicit method

The implicit method Uses a Backward Difference at time n+1 and a second-order central difference for the space derivative:

$$\frac{\partial T}{\partial t} = \frac{T_{i,j}^{n+1} - T_{i,j}^{n}}{\Delta t} \tag{5}$$

$$\frac{\partial^2 T}{\partial x^2} = \frac{T_{i+1,j}^{n+1} - 2T_{i,j}^{n+1} + T_{i-1,j}^{n+1}}{(\Delta x)^2} \tag{6}$$

$$\frac{\partial^2 T}{\partial y^2} = \frac{T_{i,j+1}^{n+1} - 2T_{i,j}^{n+1} + T_{i,j-1}^{n+1}}{(\Delta y)^2} \tag{7}$$

Substituting (5), (6) and (7), we have

$$\frac{T_{i,j}^{n+1} - T_{i,j}^{n}}{\Delta t} = \alpha \left(\frac{T_{i+1,j}^{n+1} - 2T_{i,j}^{n+1} + T_{i-1,j}^{n+1}}{(\Delta x)^2} + \frac{T_{i,j+1}^{n+1} - 2T_{i,j}^{n+1} + T_{i,j-1}^{n+1}}{(\Delta y)^2} \right)$$

Let $\Delta x = \Delta y = \Delta h$, we have

$$\frac{T_{i,j}^{n+1} - T_{i,j}^{n}}{\Delta t} = \alpha \left(\frac{T_{i+1,j}^{n+1} + T_{i-1,j}^{n+1} + T_{i,j+1}^{n+1} + T_{i,j-1}^{n+1} - 4T_{i,j}^{n+1}}{(\Delta h)^2} \right)$$

$$<=> T_{i,j}^{n+1} = T_{i,j}^{n} + \Delta t \alpha \left(\frac{T_{i+1,j}^{n+1} + T_{i-1,j}^{n+1} + T_{i,j+1}^{n+1} + T_{i,j-1}^{n+1} - 4T_{i,j}^{n+1}}{(\Delta h)^2} \right)$$

Let $\gamma = \frac{\alpha \Delta t}{(\Delta h)^2}$, we have

$$T_{i,j}^{n+1} = T_{i,j}^{n} + \gamma \left(T_{i+1,j}^{n+1} + T_{i-1,j}^{n+1} + T_{i,j+1}^{k+1} + T_{i,j-1}^{n+1} - 4T_{i,j}^{n+1} \right)$$

$$<=> (1+4\gamma)T_{i,j}^{n+1} - \gamma(T_{i+1,j}^{n+1} + T_{i-1,j}^{n+1} + T_{i,j+1}^{n+1} + T_{i,j-1}^{n+1}) = T_{i,j}^{n}$$

Continue this for all i and j in the domain. We obtain this system of linear equation:

$$A[T^{n+1}] = [T^n]$$

$$[T^{n+1}] = \begin{bmatrix} T_{1,1}^{n+1} \\ T_{1,2}^{n+1} \\ \vdots \\ T_{i,j}^{n+1} \\ T_{i,j+1}^{n+1} \\ \vdots \\ T_{N_x,N_y-1}^{n+1} \\ T_{N_x,N_y}^{n+1} \end{bmatrix} \qquad [T^{n}] = \begin{bmatrix} T_{1,1}^{n} \\ T_{1,2}^{n} \\ \vdots \\ T_{i,j}^{n} \\ T_{i,j+1}^{n} \\ \vdots \\ T_{N_x,N_y-1}^{n} \\ T_{N_x,N_y}^{n} \end{bmatrix}$$

The solution $[T^{n+1}]$ and $[T^n]$ are:
The coefficient matrix A depends on specify boundary condition.

The right-hand side matrix $[T^n]$ is known, so we can obtain the n+1 step temperature as:

$$[T^{n+1}] = A^{-1}[T^n]$$

The implicit method is always numerically stable and convergent. The truncation errors are linear over the time step and quadratic over the space step:

$$\epsilon = O(\Delta t) + O((\Delta h)^2)$$

5 Stability analysis

Define the error in the numerical approximation as

$$\epsilon_{i,j}^n = T_{i,j}^n - t_{i,j}^n$$

- $T_{i,j}^n$ is the solution without rounding error at grid point (i, j, n)
- $t_{i,j}^n$ is the approximate solution at grid point (i, j, n)

For linear differential equations with periodic boundary condition, the spatial variation of error may be expanded in a finite Fourier series, in the interval L, as:

$$\epsilon(x, y) = \sum_{m=-N}^{N} \sum_{n=-N}^{N} E_{m,n}(t) e^{i\omega_m x} e^{i\omega_n y}$$

- $\omega_m = \pi m/L$
- $\omega_n = \pi n/L$
- $N = L/\Delta h$

The time dependence of the error is included by assuming the amplitude of error $E_{m,n}$ is a function of time.

Since the difference equation for error is linear (the behavior of each term of the series is the same as the series itself), it is enough to consider the growth of error of a typical term:

$$\epsilon_{m,n}(x,y,t) = E_{m,n}(t)e^{i\omega_m x}e^{i\omega_n y}$$

The stability characteristics can be studied using just this form for the error with no loss in generality.

Define the amplification factor:

$$G = \frac{E_{m,n}(t+\Delta t)}{E_{m,n}(t)}$$

Define a Stability Criterion: The goal is to show that the error incurred by a particular numerical scheme doesn't grow in the evolving steps of time. Then, the necessary and sufficient condition for the error to remain bounded, maintaining numerical stability is:

$$|G| \leq 1$$

5.1 Explicit method

Both the numerical solution $t^n_{i,j}$ and the solution without rounding error $T^n_{i,j}$ satisfy the heat equation, therefore, the error $\epsilon^n_{i,j}$ also follows the discretized PDE equation:

$$\epsilon^{n+1}_{i,j} = (1-4\gamma)\,\epsilon^n_{i,j} + \gamma\left(\epsilon^n_{i+1,j} + \epsilon^n_{i-1,j} + \epsilon^n_{i,j+1} + \epsilon^n_{i,j-1}\right)$$

Apply Fourier Transform to both side of the equation:

$$\epsilon^n_{i,j} = E_{m,n}(t)e^{i\omega_m x}e^{i\omega_n y}$$

$$\epsilon^{n+1}_{i,j} = E_{m,n}(t+\Delta t)e^{i\omega_m x}e^{i\omega_n y}$$

$$\epsilon^n_{i-1,j} = E_{m,n}(t)e^{i\omega_m (x-\Delta h)}e^{i\omega_n y}$$

$$\epsilon^n_{i+1,j} = E_{m,n}(t)e^{i\omega_m (x+\Delta h)}e^{i\omega_n y}$$

$$\epsilon^n_{i,j-1} = E_{m,n}(t)e^{i\omega_m x}e^{i\omega_n (y-\Delta h)}$$

$$\epsilon^n_{i,j+1} = E_{m,n}(t)e^{i\omega_m x}e^{i\omega_n (y+\Delta h)}$$

We have:

$$G = (1-4\gamma) + \gamma(e^{-i\omega_m \Delta h} + e^{i\omega_m \Delta h} + e^{-i\omega_n \Delta h} + e^{i\omega_n \Delta h})$$

Using Euler's formula:

$$sin^2(\theta/2) = -\frac{e^{-i\theta} + e^{i\theta} - 2}{4}$$

$$=> G = 1 - 4\gamma(sin^2(\omega_m \Delta h/2) + sin^2(\omega_n \Delta h/2)$$

For this scheme to stable, $|G| \leq 1$

$$|1 - 4\gamma(sin^2(\omega_m \Delta h/2) + sin^2(\omega_n \Delta h/2)| \leq 1$$

$$<=> 4\gamma(sin^2(\omega_m \Delta h/2) + sin^2(\omega_n \Delta h/2) \leq 2$$

$$<=> \gamma \leq \frac{1}{4}$$

5.2 Implicit method

Both the numerical solution $t^n_{i,j}$ and the solution without rounding error $T^n_{i,j}$ satisfy the heat equation, therefore, the error $\epsilon^n_{i,j}$ also follows the discretized PDE equation:

$$(1+4\gamma)\epsilon^{n+1}_{i,j} - \gamma(\epsilon^{n+1}_{i+1,j} + \epsilon^{n+1}_{i-1,j} + \epsilon^{n+1}_{i,j+1} + \epsilon^{n+1}_{i,j-1}) = \epsilon^n_{i,j}$$

Apply Fourier Transform to both side of the equation:

$$\epsilon^n_{i,j} = E_{m,n}(t)e^{i\omega_m x}e^{i\omega_n y}$$

$$\epsilon^{n+1}_{i,j} = E_{m,n}(t+\Delta t)e^{i\omega_m x}e^{i\omega_n y}$$

$$\epsilon^{n+1}_{i-1,j} = E_{m,n}(t+\Delta t)e^{i\omega_m (x-\Delta h)}e^{i\omega_n y}$$

$$\epsilon^{n+1}_{i+1,j} = E_{m,n}(t+\Delta t)e^{i\omega_m (x+\Delta h)}e^{i\omega_n y}$$

$$\epsilon^{n+1}_{i,j-1} = E_{m,n}(t+\Delta t)e^{i\omega_m x}e^{i\omega_n (y-\Delta h)}$$

$$\epsilon^{n+1}_{i,j+1} = E_{m,n}(t+\Delta t)e^{i\omega_m x}e^{i\omega_n (y+\Delta h)}$$

We have:

$$(1+4\gamma)G - \gamma G(e^{-i\omega_m\Delta h} + e^{i\omega_m\Delta h} + e^{-i\omega_n\Delta h} + e^{i\omega_n\Delta h}) = 1$$

Using Euler's formula:

$$sin^2(\theta/2) = -\frac{e^{-i\theta} + e^{i\theta} - 2}{4}$$

$$=> G + 4\gamma G(sin^2(\omega_m\Delta h/2) + sin^2(\omega_n\Delta h/2)) = 1$$

$$=> G = \frac{1}{1 + 4\gamma(sin^2(\omega_m\Delta h/2) + sin^2(\omega_n\Delta h/2))}$$

The right hand side is always positive and smaller than 1. Therefore, This scheme is unconditionally stable.

6 Example

- Domain:

$$x \in [0,1], \quad y \in [0,1].$$

- Initial condition function:

$$T(x,y) = 100\sin(\pi x)\sin(\pi y).$$

- Boundary condition:

$$T(0,y,t) = T(1,y,t) = T(x,0,t) = T(x,1,t) = 0.$$

- Exact solution:

$$T(x,y,t) = 100\sin(\pi x)\sin(\pi y)e^{-2a\pi^2 t}.$$

- Thermal diffusivity:

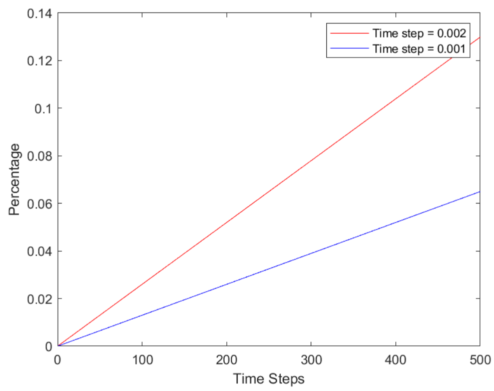
$$\alpha = 0.1.$$

- For this boundary condition, the coefficient matrix A of the implicit method is:

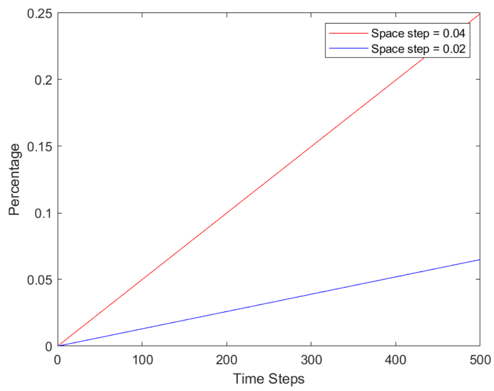
$$A = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & & & & & & & & & & & \vdots & \vdots \\ 0 & 0 & & -\gamma & \cdots & -\gamma & 1+4\gamma & -\gamma & \cdots & -\gamma & 0 & & 0 & 0 \\ 0 & 0 & & 0 & -\gamma & \cdots & -\gamma & 1+4\gamma & -\gamma & \cdots & -\gamma & & 0 & 0 \\ \vdots & \vdots & & & & & & & & & & & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 \end{bmatrix}$$

- The implicit method is more computationally demanding than the explicit method as it requires solving a system of numerical equations on each time step. The explicit method is easier to implement and less computationally demanding.

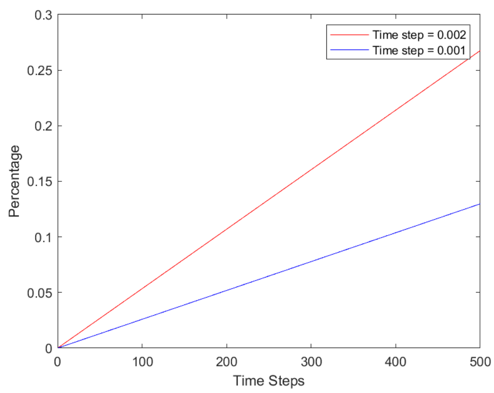
- Both methods have comparable precision with respect to time step and space step.



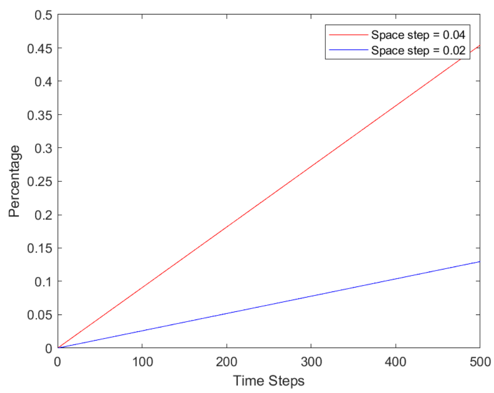
(a) Error w.r.t to Δt of explicit method



(b) Error w.r.t to Δh of explicit method



(c) Error w.r.t to Δt of implicit method



(d) Error w.r.t to Δh of implicit method