Finite-Difference Methods

Oliver Zhao

1 Background

Finite-difference methods (FDM) are a set of techniques often used to numerically solve partial differential equations by approximating the derivatives with finite differences.

1.1 Taylor Series Expansion

Recall that the Taylor series expansion can be written as

$$u(x + \Delta x) = u(x) + \frac{\partial u}{\partial x}\Delta x + \frac{1}{2!}\frac{\partial^2 u}{\partial x^2}(\Delta x)^2 + \dots \tag{1.1}$$

1.2 1st-Order Approximation

Forward difference:

$$\frac{\partial u}{\delta x} = \frac{u(x + \Delta x) - u(x)}{\Delta x} + O(\Delta x) \tag{1.2}$$

Backward difference:

$$\frac{\partial u}{\delta x} = \frac{u(x) - u(x - \Delta x)}{\Delta x} + O(\Delta x) \tag{1.3}$$

Central difference:

$$\frac{\partial u}{\delta x} = \frac{u(x + \Delta x) - u(x - \Delta x)}{2\Delta x} + O(\Delta x^2) \tag{1.4}$$

1.3 2nd-Order Approximation

Forward difference:

$$\frac{\partial^2 u}{\partial x^2} = \frac{u(x + 2\Delta x) - 2u(x + \Delta x) + u(x)}{(\Delta x)^2} + O(\Delta x) \tag{1.5}$$

Backward difference:

$$\frac{\partial^2 u}{\partial x^2} = \frac{u(x) - 2u(x - \Delta x) + u(x - 2\Delta x)}{(\Delta x)^2} + O(\Delta x) \tag{1.6}$$

Central difference:

$$\frac{\partial^2 u}{\partial x^2} = \frac{u(x + \Delta x) - 2u(x) + u(x - \Delta x)}{(\Delta x)^2} + O(\Delta x^2) \tag{1.7}$$

1.4 Higher-Order Approximations

nth-order forward difference:

$$\frac{\partial^n u}{\partial x^n} = \left[\sum_{i=0}^{n} (-1)^{n-i} \binom{n}{i} u(x + (n - i)\Delta x)\right] / (\Delta x)^n \tag{1.8}$$

nth-order backward difference:

$$\frac{\partial^n u}{\partial x^n} = \left[\sum_{i=0}^{n} (-1)^{i} \binom{n}{i} u(x - i\Delta x)\right] / (\Delta x)^n \tag{1.9}$$

nth-order central difference:

$$\frac{\partial^n u}{\partial x^n} = \left[\sum_{i=0}^{n} (-1)^{i} \binom{n}{i} u\left(x + \left(\frac{n}{2} - i\right) \Delta x\right)\right] / (\Delta x)^n \tag{1.10}$$

1.5 Solving Differential Equations

When solving differential equations, these finite difference approximations are substituted into the original differential equation to derive the numerical solution.

There are three general schemes for solving differential equations through FDMs. The explicit method relies on a forward difference at time t_n, while the implicit method relies on a backward difference at time t_{n+1}. Meanwhile, the Crank-Nicholson method uses a central difference at time $t_{n+1/2}$.

2 Examples: One-Dimension Problems

2.1 1D Laplace Equation

Note that the 1D Laplace equation can be easily solved analytically. Here we use the FDM method for demonstration purposes.

$$U_{xx} = 0. \tag{2.1}$$

By using the central difference for the space derivative at points x_i, we get

$$\frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} = 0. \tag{2.2}$$

2.2 1D Heat Equation

Consider the normalized 1D heat equation problem with homogeneous Dirichlet boundary conditions:

$$\begin{aligned} &U_t = U_{xx} \\ &U(0,t) = U(1,t) = 0 \qquad \text{(boundary condition)} \\ &U(x,0) = U_0(x) \qquad\qquad \text{(initial condition)} \end{aligned} \tag{2.3}$$

This problem can be solved by approximating the derivatives with their respective finite differences. We partition the domain in space using a mesh $x_0, \ldots, x_J$ and in time using a mesh $t_0, \ldots, t_N$, where the difference between two adjacent space points is h and the difference between two consecutive time points is k. Then we use the notation below for the numerical approximations of $u(x_j, t_n)$.

$$u(x_j, t_n) = u_j^n. \tag{2.4}$$

2.2.1 Explicit Method

By using the forward difference for the time derivative at time t_n and second-order central difference for the space derivative at position x_j (FTCS method), we get

$$\frac{u_j^{n+1} - u_j^n}{k} = \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{h^2}. \tag{2.5}$$

This can be rearranged into

$$u_j^{n+1} = (1 - 2r)u_j^n + ru_{j-1}^n + ru_{j+1}^n, \qquad r = k/h^2. \tag{2.6}$$

Note that if the condition $r \leq 1/2$ is satisfied, the method will be numerically stable and convergent. Meanwhile, the numerical errors are proportional to the time step and square of the space step, or

$$\Delta u = O(k) + O(h^2). \tag{2.7}$$

2.2.2 Implicit Method

By using the backward difference for the time derivative at time t_{n+1} and second-order central difference for the space derivative at position x_j (BTCS method), we get

$$\frac{u_j^{n+1} - u_j^n}{k} = \frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{h^2}. \tag{2.8}$$

This can be rearranged into

$$u_j^n = (1 + 2r)u_j^{n+1} - ru_{j-1}^{n+1} - ru_{j+1}^{n+1}, \qquad r = k/h^2. \tag{2.9}$$

Like the FTCS method, if the condition $r \leq 1/2$ is satisfied, the method will be numerically stable and convergent. Meanwhile, the numerical errors are proportional to the time step and square of the space step, or

$$\Delta u = O(k) + O(h^2). \tag{2.10}$$

2.2.3 Crank-Nicholson Method

By using the central difference for the time derivative at time $t_{n+1/2}$ and second-order central difference for the space derivative at position x_j (CTCS method), we get

$$\frac{u_j^{n+1} - u_j^n}{k} = \frac{1}{2}\left(\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{h^2} + \frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{h^2}\right). \tag{2.11}$$

This can be rearranged into

$$(2+2r)u_j^{n+1} - ru_{j-1}^{n+1} - ru_{j+1}^{n+1} = (2-2r)u_j^n + ru_{j-1}^n + ru_{j+1}^n. \tag{2.12}$$

This scheme is always numerically stable and convergent, but requires more computational power. The errors are quadratic over both the time step and space step, or

$$\Delta u = O(k^2) + O(h^2). \tag{2.13}$$

3 Examples: Two-Dimension Problems

3.1 2D Laplace Equation

Consider the 2D Laplace equation problem

$$U_{xx} + U_{yy} = 0. \tag{3.1}$$

By using the central difference for the space derivative at points x_i and y_j, we get

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta y^2} = 0. \tag{3.2}$$

If we assume that $\Delta x = \Delta y$, then the problem can be further simplified into

$$4u_{i,j} - u_{i-1,j} - u_{i,j-1} - u_{i+1,j} - u_{i,j+1} = 0. \tag{3.3}$$

3.2 2D Heat Equation

Consider the normalized 2D heat equation problem with homogeneous Dirichlet boundary conditions:

$$\begin{aligned}
&U_t = U_{xx} + U_{yy} \\
&U(0, y, t) = U(1, y, t) = 0 \quad \text{(boundary condition)} \\
&U(x, 0, t) = U(x, 1, t) = 0 \quad \text{(boundary condition)} \\
&U(x, y, 0) = U_0(x, y) \qquad \text{(initial condition)}
\end{aligned} \tag{3.4}$$

Let i indicate the node position in the x-dimension and j indicate the node position in the y-dimension, and k indicate the time point. For simplicity, we assume that $\Delta x = \Delta y$.

3.2.1 Explicit Solution

Analogous to the FTCS method used to solve the 1D heat equation, we get:

$$\frac{T_{i,j}^{k+1} - T_{i,j}^{k}}{\Delta t} = \frac{T_{i,j-1}^{k} + T_{i-1,j}^{k} - 4T_{i,j}^{k} + T_{i+1,j}^{k} + T_{i,j+1}^{k}}{\Delta x^2}. \tag{3.5}$$

Which can be rearranged into

$$T_{i,j}^{k+1} = T_{i,j}^{k} + \Delta t \left(\frac{T_{i,j-1}^{k} + T_{i-1,j}^{k} - 4T_{i,j}^{k} + T_{i+1,j}^{k} + T_{i,j+1}^{k}}{\Delta x^2} \right) \tag{3.6}$$

3.2.2 Implicit Solution

Analogous to the BTCS method used to solve the 1D heat equation, we get:

$$\frac{T_{i,j}^{k+1} - T_{i,j}^{k}}{\Delta t} = \frac{T_{i,j-1}^{k+1} + T_{i-1,j}^{k+1} - 4T_{i,j}^{k+1} + T_{i+1,j}^{k+1} + T_{i,j+1}^{k+1}}{\Delta x^2}. \tag{3.7}$$

Which can be rearranged into

$$T_{i,j}^{k+1} = T_{i,j}^{k} + \Delta t \left(\frac{T_{i,j-1}^{k+1} + T_{i-1,j}^{k+1} - 4T_{i,j}^{k+1} + T_{i+1,j}^{k+1} + T_{i,j+1}^{k+1}}{\Delta x^2} \right) \tag{3.8}$$