Computational Fluid Dynamics: Part 2

Kiara Gholizad[1]

[1]Department of Physics, Sharif University of Technology, Tehran, Iran

April 9, 2025

Problem 1

Derive the backward difference approximation formula for $\frac{\partial^3 f}{\partial x^3}$ of order Δx, using equally spaced grid points f_{i-3}, f_{i-2}, f_{i-1}, and f_i, by applying the following methods:

(a) Taylor series expansions

(b) Backward difference formulas

(c) A third-degree polynomial (cubic) interpolating four points

Solution

(a) Taylor Series Expansions

In order to get the backward difference approximation for the third derivative by Taylor series, we are required to represent f_{i-1}, f_{i-2}, and f_{i-3} in terms of f_i and its derivatives.

Let's expand f_{i-1}, f_{i-2}, and f_{i-3} in Taylor series near x_i:

$$\begin{aligned}
f_{i-1} &= f(x_i - \Delta x) \qquad (1)\\
&= f(x_i) - \Delta x \cdot f'(x_i) + \frac{(\Delta x)^2}{2!} \cdot f''(x_i) \\
&\quad - \frac{(\Delta x)^3}{3!} \cdot f'''(x_i) + \frac{(\Delta x)^4}{4!} \cdot f^{(4)}(x_i) + O((\Delta x)^5) \\
f_{i-2} &= f(x_i - 2\Delta x) \qquad (2)\\
&= f(x_i) - 2\Delta x \cdot f'(x_i) + \frac{(2\Delta x)^2}{2!} \cdot f''(x_i) \\
&\quad - \frac{(2\Delta x)^3}{3!} \cdot f'''(x_i) + \frac{(2\Delta x)^4}{4!} \cdot f^{(4)}(x_i) + O((\Delta x)^5) \\
&= f(x_i) - 2\Delta x \cdot f'(x_i) + 2(\Delta x)^2 \cdot f''(x_i) \\
&\quad - \frac{8(\Delta x)^3}{6} \cdot f'''(x_i) + \frac{16(\Delta x)^4}{24} \cdot f^{(4)}(x_i) + O((\Delta x)^5) \\
f_{i-3} &= f(x_i - 3\Delta x) \qquad (3)\\
&= f(x_i) - 3\Delta x \cdot f'(x_i) + \frac{(3\Delta x)^2}{2!} \cdot f''(x_i) \\
&\quad - \frac{(3\Delta x)^3}{3!} \cdot f'''(x_i) + \frac{(3\Delta x)^4}{4!} \cdot f^{(4)}(x_i) + O((\Delta x)^5) \\
&= f(x_i) - 3\Delta x \cdot f'(x_i) + \frac{9(\Delta x)^2}{2} \cdot f''(x_i) \\
&\quad - \frac{27(\Delta x)^3}{6} \cdot f'''(x_i) + \frac{81(\Delta x)^4}{24} \cdot f^{(4)}(x_i) + O((\Delta x)^5)
\end{aligned}$$

Our goal is to find constants a, b, c, and d such that:

$$a \cdot f_{i-3} + b \cdot f_{i-2} + c \cdot f_{i-1} + d \cdot f_i = (\Delta x)^3 \cdot f'''(x_i) + O((\Delta x)^4) \qquad (4)$$

Substituting the Taylor series expansions:

$$\begin{aligned}
& a \cdot \left[f(x_i) - 3\Delta x \cdot f'(x_i) + \frac{9(\Delta x)^2}{2} \cdot f''(x_i) - \frac{27(\Delta x)^3}{6} \cdot f'''(x_i) + \cdots \right] \\
& + b \cdot \left[f(x_i) - 2\Delta x \cdot f'(x_i) + 2(\Delta x)^2 \cdot f''(x_i) - \frac{8(\Delta x)^3}{6} \cdot f'''(x_i) + \cdots \right] \\
& + c \cdot \left[f(x_i) - \Delta x \cdot f'(x_i) + \frac{(\Delta x)^2}{2} \cdot f''(x_i) - \frac{(\Delta x)^3}{6} \cdot f'''(x_i) + \cdots \right] \\
& + d \cdot f(x_i) = (\Delta x)^3 \cdot f'''(x_i) + O((\Delta x)^4)
\end{aligned}$$

Collecting terms with the same derivative of f:

$$\begin{aligned}
&(a+b+c+d)\cdot f(x_i)\\
&+(-3a-2b-c)\cdot \Delta x\cdot f'(x_i)\\
&+\left(\frac{9a}{2}+2b+\frac{c}{2}\right)\cdot(\Delta x)^2\cdot f''(x_i)\\
&+\left(-\frac{27a}{6}-\frac{8b}{6}-\frac{c}{6}\right)\cdot(\Delta x)^3\cdot f'''(x_i)\\
&+\text{higher-order terms}=(\Delta x)^3\cdot f'''(x_i)+O((\Delta x)^4)
\end{aligned}$$

For this equation to be satisfied, we need:

$$\begin{aligned}
a+b+c+d&=0\\
-3a-2b-c&=0\\
\frac{9a}{2}+2b+\frac{c}{2}&=0\\
-\frac{27a}{6}-\frac{8b}{6}-\frac{c}{6}&=1
\end{aligned}$$

This system of equations has the solution:

$$\begin{aligned}
a&=-1\\
b&=3\\
c&=-3\\
d&=1
\end{aligned}$$

Therefore, the third derivative approximation is:

$$f'''(x_i)\approx\frac{-f_{i-3}+3f_{i-2}-3f_{i-1}+f_i}{(\Delta x)^3} \tag{5}$$

(b) Backward Difference Formulas

In the backward difference approach, we work with the backward difference operators. Let's denote the first backward difference as:

$$\nabla f_i=f_i-f_{i-1} \tag{6}$$

The second backward difference is:

$$\begin{aligned} \nabla^2 f_i &= \nabla(\nabla f_i) \\ &= \nabla(f_i - f_{i-1}) \\ &= (f_i - f_{i-1}) - (f_{i-1} - f_{i-2}) \\ &= f_i - 2f_{i-1} + f_{i-2} \end{aligned} \tag{7}$$

The third backward difference is:

$$\begin{aligned} \nabla^3 f_i &= \nabla(\nabla^2 f_i) \\ &= \nabla(f_i - 2f_{i-1} + f_{i-2}) \\ &= (f_i - 2f_{i-1} + f_{i-2}) - (f_{i-1} - 2f_{i-2} + f_{i-3}) \\ &= f_i - 2f_{i-1} + f_{i-2} - f_{i-1} + 2f_{i-2} - f_{i-3} \\ &= f_i - 3f_{i-1} + 3f_{i-2} - f_{i-3} \end{aligned} \tag{8}$$

We can prove the relation between the third backward difference and the third derivative by expressing the backward difference in terms of the derivatives. We can start by writing the function value in Taylor series near x_i:

For $f_{i-1} = f(x_i - \Delta x)$:

$$f_{i-1} = f(x_i) - \Delta x \cdot f'(x_i) + \frac{(\Delta x)^2}{2!} \cdot f''(x_i) - \frac{(\Delta x)^3}{3!} \cdot f'''(x_i) + O((\Delta x)^4)$$

For $f_{i-2} = f(x_i - 2\Delta x)$:

$$\begin{aligned} f_{i-2} &= f(x_i) - 2\Delta x \cdot f'(x_i) + \frac{(2\Delta x)^2}{2!} \cdot f''(x_i) - \frac{(2\Delta x)^3}{3!} \cdot f'''(x_i) + O((\Delta x)^4) \\ &= f(x_i) - 2\Delta x \cdot f'(x_i) + 2(\Delta x)^2 \cdot f''(x_i) - \frac{8(\Delta x)^3}{6} \cdot f'''(x_i) + O((\Delta x)^4) \end{aligned}$$

For $f_{i-3} = f(x_i - 3\Delta x)$:

$$\begin{aligned} f_{i-3} &= f(x_i) - 3\Delta x \cdot f'(x_i) + \frac{(3\Delta x)^2}{2!} \cdot f''(x_i) - \frac{(3\Delta x)^3}{3!} \cdot f'''(x_i) + O((\Delta x)^4) \\ &= f(x_i) - 3\Delta x \cdot f'(x_i) + \frac{9(\Delta x)^2}{2} \cdot f''(x_i) - \frac{27(\Delta x)^3}{6} \cdot f'''(x_i) + O((\Delta x)^4) \end{aligned}$$

Now, let's substitute these expansions into our expression for $\nabla^3 f_i$:

$$\begin{aligned}
\nabla^3 f_i &= f_i - 3f_{i-1} + 3f_{i-2} - f_{i-3} \\
&= f(x_i) - 3\left[f(x_i) - \Delta x \cdot f'(x_i) + \frac{(\Delta x)^2}{2} \cdot f''(x_i) - \frac{(\Delta x)^3}{6} \cdot f'''(x_i) + O((\Delta x)^4)\right] \\
&\quad + 3\left[f(x_i) - 2\Delta x \cdot f'(x_i) + 2(\Delta x)^2 \cdot f''(x_i) - \frac{8(\Delta x)^3}{6} \cdot f'''(x_i) + O((\Delta x)^4)\right] \\
&\quad - \left[f(x_i) - 3\Delta x \cdot f'(x_i) + \frac{9(\Delta x)^2}{2} \cdot f''(x_i) - \frac{27(\Delta x)^3}{6} \cdot f'''(x_i) + O((\Delta x)^4)\right]
\end{aligned}$$

Let's simplify by collecting terms:
For $f(x_i)$ terms:

$$f(x_i) - 3f(x_i) + 3f(x_i) - f(x_i) = 0$$

For $f'(x_i)$ terms:

$$\begin{aligned}
0 - 3(-\Delta x \cdot f'(x_i)) + 3(-2\Delta x \cdot f'(x_i)) - (-3\Delta x \cdot f'(x_i)) \\
= 3\Delta x \cdot f'(x_i) - 6\Delta x \cdot f'(x_i) + 3\Delta x \cdot f'(x_i) = 0
\end{aligned}$$

For $f''(x_i)$ terms:

$$\begin{aligned}
0 - 3\left(\frac{(\Delta x)^2}{2} \cdot f''(x_i)\right) + 3\left(2(\Delta x)^2 \cdot f''(x_i)\right) - \left(\frac{9(\Delta x)^2}{2} \cdot f''(x_i)\right) \\
= -\frac{3(\Delta x)^2}{2} \cdot f''(x_i) + 6(\Delta x)^2 \cdot f''(x_i) - \frac{9(\Delta x)^2}{2} \cdot f''(x_i) \\
= \left(-\frac{3}{2} + 6 - \frac{9}{2}\right)(\Delta x)^2 \cdot f''(x_i) = 0
\end{aligned}$$

For $f'''(x_i)$ terms:

$$\begin{aligned}
0 - 3\left(-\frac{(\Delta x)^3}{6} \cdot f'''(x_i)\right) + 3\left(-\frac{8(\Delta x)^3}{6} \cdot f'''(x_i)\right) - \left(-\frac{27(\Delta x)^3}{6} \cdot f'''(x_i)\right) \\
= \frac{3(\Delta x)^3}{6} \cdot f'''(x_i) - \frac{24(\Delta x)^3}{6} \cdot f'''(x_i) + \frac{27(\Delta x)^3}{6} \cdot f'''(x_i) \\
= \left(\frac{3}{6} - \frac{24}{6} + \frac{27}{6}\right)(\Delta x)^3 \cdot f'''(x_i) \\
= \frac{6}{6}(\Delta x)^3 \cdot f'''(x_i) = (\Delta x)^3 \cdot f'''(x_i)
\end{aligned}$$

Therefore:

$$\nabla^3 f_i = (\Delta x)^3 \cdot f'''(x_i) + O((\Delta x)^4)$$

Solving for $f'''(x_i)$:

$$\begin{align}
f'''(x_i) &= \frac{\nabla^3 f_i}{(\Delta x)^3} + O(\Delta x) \tag{9}\\
&\approx \frac{\nabla^3 f_i}{(\Delta x)^3} \tag{10}\\
&= \frac{f_i - 3f_{i-1} + 3f_{i-2} - f_{i-3}}{(\Delta x)^3} \tag{11}
\end{align}$$

This formally establishes that the third backward difference divided by $(\Delta x)^3$ is a first-order accurate approximation to the third derivative at point x_i.

Note that the coefficients of the backward difference expression (1, -3, 3, -1) are the binomial coefficients of the expansion of $(1-1)^3 = 1 - 3 + 3 - 1 = 0$, in a different order though, and using alternate signs. We identify the same behavior in the higher-order backward differences using Pascal's triangle, using alternate signs though.

(c) Cubic Interpolation

For the cubic interpolation method, we construct a cubic polynomial $p(x)$ that passes through the four points (x_i, f_i), (x_{i-1}, f_{i-1}), (x_{i-2}, f_{i-2}), and (x_{i-3}, f_{i-3}).

The cubic polynomial has the form:

$$p(x) = A + B(x - x_i) + C(x - x_i)^2 + D(x - x_i)^3 \tag{12}$$

We need to solve for the coefficients A, B, C, and D using the four data points:

$$\begin{aligned}
p(x_i) &= A = f_i\\
p(x_{i-1}) &= A + B(-\Delta x) + C(-\Delta x)^2 + D(-\Delta x)^3 = f_{i-1}\\
p(x_{i-2}) &= A + B(-2\Delta x) + C(-2\Delta x)^2 + D(-2\Delta x)^3 = f_{i-2}\\
p(x_{i-3}) &= A + B(-3\Delta x) + C(-3\Delta x)^2 + D(-3\Delta x)^3 = f_{i-3}
\end{aligned}$$

This gives us a system of four equations for the four unknowns A, B, C, and D. We can write this in matrix form:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & -\Delta x & (\Delta x)^2 & -(\Delta x)^3 \\ 1 & -2\Delta x & 4(\Delta x)^2 & -8(\Delta x)^3 \\ 1 & -3\Delta x & 9(\Delta x)^2 & -27(\Delta x)^3 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} f_i \\ f_{i-1} \\ f_{i-2} \\ f_{i-3} \end{bmatrix} \tag{13}$$

We are particularly concerned with the coefficient D, as the third derivative of a cubic polynomial is:

$$p'''(x) = 6D \tag{14}$$

In order to solve for D in terms of the function values, we must solve the system of equations. We can accomplish this using Cramer's rule or by solving the system directly.

Let M be the coefficient matrix and F be the right-hand side vector. Then:

$$D = \frac{f_i - 3f_{i-1} + 3f_{i-2} - f_{i-3}}{-6(\Delta x)^3} \tag{15}$$

Therefore:

$$p'''(x_i) = 6D = 6 \cdot \frac{f_i - 3f_{i-1} + 3f_{i-2} - f_{i-3}}{-6(\Delta x)^3} = \frac{-f_i + 3f_{i-1} - 3f_{i-2} + f_{i-3}}{(\Delta x)^3}$$

Since we want to approximate $f'''(x_i)$ with $p'''(x_i)$, we get:

$$f'''(x_i) \approx \frac{f_i - 3f_{i-1} + 3f_{i-2} - f_{i-3}}{(\Delta x)^3} \tag{16}$$

This is the same formula obtained using the other two methods.

Numerical Verification

In order to check the obtained formula, we carried out a numerical test by using the function $f(x) = x^5 + 3x^3 - 2x + 5$, which possesses the precise third derivative $f'''(x) = 60x^2 + 18$.

Figure 1 plots the exact third derivative and numerical approximation based on our derived formula. We obtained the largest relative error to be around 5.37%, which validates the efficacy of our formula.

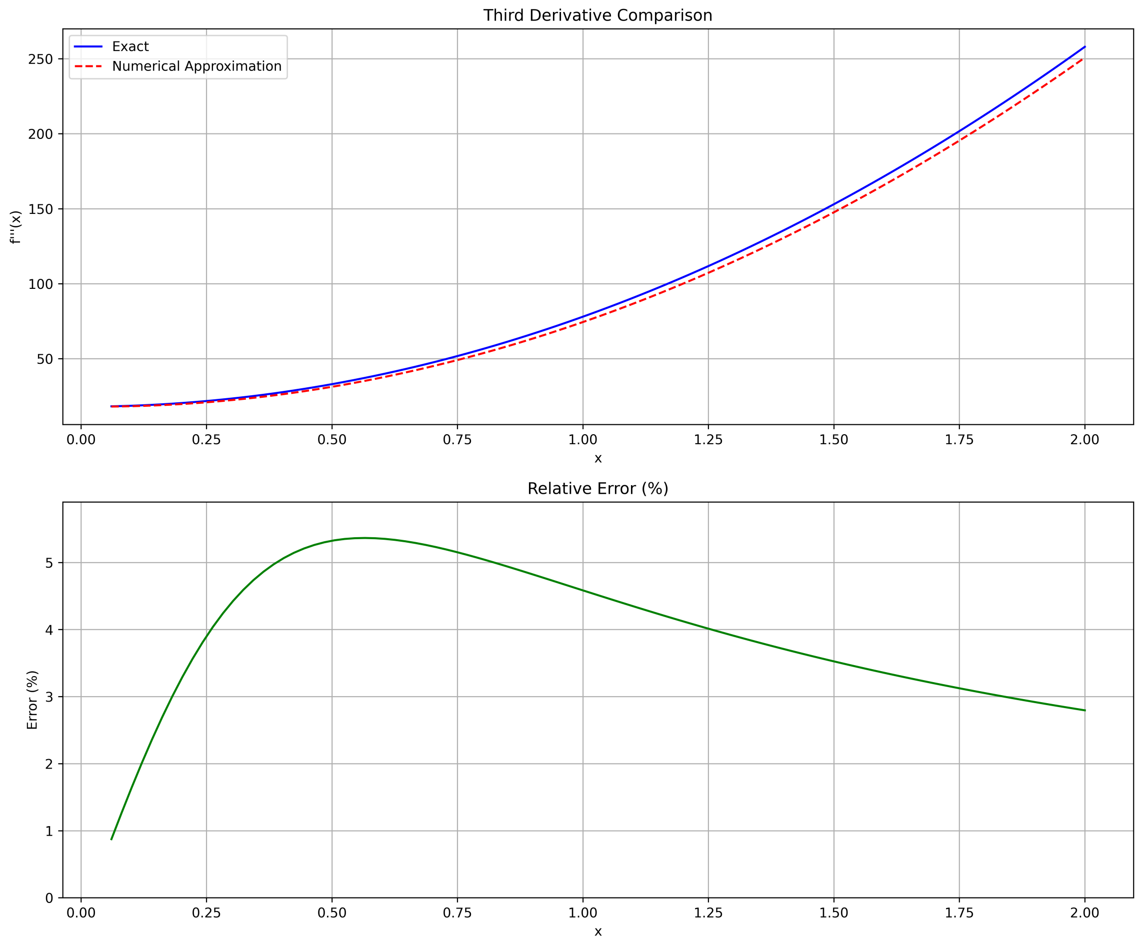

Figure 1: Comparison of exact and numerical third derivatives using the derived backward difference formula

Problem 2

A discretization method for the one-dimensional heat conduction equation with a constant thermal diffusivity α is given by:

$$\frac{\phi_i^{n+1} - \phi_i^{n-1}}{2\Delta t} = \frac{\alpha}{\Delta x^2}\left(\phi_{i+1}^n - \phi_i^{n+1} - \phi_i^{n-1} + \phi_{i-1}^n\right) \tag{17}$$

Using Fourier (von Neumann) stability analysis, determine the stability conditions for this scheme.

Solution

von Neumann Stability Analysis

To conduct a von Neumann stability analysis, we substitute a Fourier mode of the form

$$\phi_i^n = \lambda^n e^{I\theta i} \tag{18}$$

where λ is the amplification factor, θ is the wave number, and $I = \sqrt{-1}$.

Substituting into the given discretization scheme:

$$\frac{\lambda^{n+1}e^{I\theta i} - \lambda^{n-1}e^{I\theta i}}{2\Delta t} = \frac{\alpha}{\Delta x^2}\left(\lambda^n e^{I\theta(i+1)} - \lambda^{n+1}e^{I\theta i} - \lambda^{n-1}e^{I\theta i} + \lambda^n e^{I\theta(i-1)}\right)$$

Dividing both sides by $\lambda^{n-1}e^{I\theta i}$:

$$\frac{\lambda^2 - 1}{2\Delta t} = \frac{\alpha}{\Delta x^2}\left(\lambda \cdot e^{I\theta} - \lambda^2 - 1 + \lambda \cdot e^{-I\theta}\right)$$

Using the identity $e^{I\theta} + e^{-I\theta} = 2\cos\theta$:

$$\frac{\lambda^2 - 1}{2\Delta t} = \frac{\alpha}{\Delta x^2}\left(2\lambda\cos\theta - \lambda^2 - 1\right)$$

Multiplying both sides by $2\Delta t$:

$$\lambda^2 - 1 = \frac{2\alpha\Delta t}{\Delta x^2}\left(2\lambda\cos\theta - \lambda^2 - 1\right)$$

Defining $\beta = \frac{\alpha\Delta t}{\Delta x^2}$:

$$\begin{aligned} \lambda^2 - 1 &= 2\beta\left(2\lambda\cos\theta - \lambda^2 - 1\right) \\ \lambda^2 - 1 &= 4\beta\lambda\cos\theta - 2\beta\lambda^2 - 2\beta \\ \lambda^2 + 2\beta\lambda^2 &= 1 + 4\beta\lambda\cos\theta + 2\beta \end{aligned}$$

$$(1+2\beta)\lambda^2 - 4\beta\lambda\cos\theta - (1+2\beta) = 0 \tag{19}$$

This is the characteristic equation of our three-level scheme. Using the quadratic formula:

$$\lambda = \frac{4\beta\cos\theta \pm \sqrt{16\beta^2\cos^2\theta + 4(1+2\beta)^2}}{2(1+2\beta)} \tag{20}$$

Stability Analysis for Three-Level Schemes

An important observation about this characteristic equation is that the product of its roots is:

$$\lambda_1 \cdot \lambda_2 = \frac{-(1+2\beta)}{1+2\beta} = -1 \tag{21}$$

This means that for any value of $\beta > 0$, the product of the roots will always be -1. A key consequence is that it is impossible for both roots to have magnitude less than 1 simultaneously.

For three-level schemes of this type, the usual von Neumann criterion needs to be adjusted. For this scheme, the appropriate condition of stability is:

- There is exactly one root that is greater in magnitude than 1 for any value of θ.
- The remaining root should be of magnitude less than or equal to 1 for any value of θ.
- Also, the scheme needs to be correctly initialized to inhibit the unstable mode's contribution.

Let us analyze the behavior of the roots at critical values of θ and for different values of β.

Analysis at Critical Points

Case 1: $\theta = 0$ (lowest frequency mode) At $\theta = 0$, $\cos\theta = 1$, and the characteristic equation becomes:

$$(1+2\beta)\lambda^2 - 4\beta\lambda - (1+2\beta) = 0$$

The roots are:

$$\lambda = \frac{4\beta \pm \sqrt{16\beta^2 + 4(1+2\beta)^2}}{2(1+2\beta)}$$

Let's examine these roots more closely. For $\beta > 0$:

$$\lambda_1 = \frac{4\beta + \sqrt{16\beta^2 + 4(1+2\beta)^2}}{2(1+2\beta)} > 1$$

$$\lambda_2 = \frac{4\beta - \sqrt{16\beta^2 + 4(1+2\beta)^2}}{2(1+2\beta)} < 0$$

Since $\lambda_1 \cdot \lambda_2 = -1$ and $\lambda_1 > 1$, we must have $-1 < \lambda_2 < 0$. Therefore, at $\theta = 0$, we always have one root with magnitude greater than 1 and one with magnitude less than 1.

Case 2: $\theta = \pi$ (highest frequency mode) At $\theta = \pi$, $\cos\theta = -1$, and the characteristic equation becomes:

$$(1 + 2\beta)\lambda^2 + 4\beta\lambda - (1 + 2\beta) = 0$$

The roots are:

$$\lambda = \frac{-4\beta \pm \sqrt{16\beta^2 + 4(1 + 2\beta)^2}}{2(1 + 2\beta)}$$

For all $\beta > 0$:

$$\lambda_1 = \frac{-4\beta + \sqrt{16\beta^2 + 4(1 + 2\beta)^2}}{2(1 + 2\beta)} > 0$$
$$\lambda_2 = \frac{-4\beta - \sqrt{16\beta^2 + 4(1 + 2\beta)^2}}{2(1 + 2\beta)} < 0$$

Again, since $\lambda_1 \cdot \lambda_2 = -1$, we need to determine whether both roots can have magnitude greater than 1.

Case 3: $\theta = \frac{\pi}{2}$ (mid frequency mode) At $\theta = \frac{\pi}{2}$, $\cos\theta = 0$, and the characteristic equation becomes:

$$(1 + 2\beta)\lambda^2 - (1 + 2\beta) = 0$$
$$\lambda^2 = 1$$

The roots are $\lambda = \pm 1$, both with magnitude exactly 1, regardless of the value of β.

Theoretical Analysis of the Roots

Let's derive the condition under which both roots could have magnitude greater than 1. For this to happen, we would need:

$$|\lambda_1| > 1 \quad \text{and} \quad |\lambda_2| > 1$$

Since $\lambda_1 \cdot \lambda_2 = -1$, for both roots to have magnitude greater than 1, they would need to be real numbers with opposite signs, with $|\lambda_1| > 1$ and $|\lambda_2| > 1$. This would mean:

$$\lambda_1 > 1 \quad \text{and} \quad \lambda_2 < -1 \quad \text{or}$$
$$\lambda_1 < -1 \quad \text{and} \quad \lambda_2 > 1$$

However, this creates a contradiction because $\lambda_1 \cdot \lambda_2 = -1$ would require $|\lambda_1| \cdot |\lambda_2| = 1$. If both $|\lambda_1| > 1$ and $|\lambda_2| > 1$, then $|\lambda_1| \cdot |\lambda_2| > 1$, which contradicts our constraint.

Therefore, for any value of $\beta > 0$ and any value of θ, it is mathematically impossible for both roots to have magnitude greater than 1 simultaneously.

Behavior as β Increases

As β approaches infinity, the characteristic equation approaches:

$$2\beta\lambda^2 - 4\beta\lambda\cos\theta - 2\beta = 0 \tag{22}$$

Dividing by 2β:

$$\lambda^2 - 2\lambda\cos\theta - 1 = 0$$

The roots of this limiting equation are:

$$\lambda = \cos\theta \pm \sqrt{\cos^2\theta + 1}$$

Let's examine these roots at different values of θ:

For $\theta = 0$ ($\cos\theta = 1$):

$$\lambda = 1 \pm \sqrt{2}$$
$$\lambda_1 = 1 + \sqrt{2} \approx 2.414$$
$$\lambda_2 = 1 - \sqrt{2} \approx -0.414$$

Here, $|\lambda_1| > 1$ and $|\lambda_2| < 1$.

For $\theta = \frac{\pi}{2}$ ($\cos\theta = 0$):

$$\lambda = \pm 1$$

Both roots have magnitude exactly 1.

For $\theta = \pi$ ($\cos\theta = -1$):

$$\lambda = -1 \pm \sqrt{2}$$
$$\lambda_1 = -1 + \sqrt{2} \approx 0.414$$
$$\lambda_2 = -1 - \sqrt{2} \approx -2.414$$

Here, $|\lambda_1| < 1$ and $|\lambda_2| > 1$.

This analysis confirms that as $\beta \to \infty$, we still maintain the property that exactly one root has magnitude greater than 1 and one root has magnitude less than or equal to 1 for all values of θ.

Unconditional Stability of the Scheme

Based on the theoretical analysis we have conducted, we can now state the following conclusions about the stability of the given discretization scheme:

1. For any value of $\beta > 0$, the product of the roots of the characteristic equation is always -1.

2. Due to this constraint, it is mathematically impossible for both roots to have magnitude greater than 1 simultaneously.

3. For all values of $\beta > 0$ and all wave numbers θ, exactly one root has magnitude greater than 1, and one root has magnitude less than or equal to 1.

4. This property holds true even as β approaches infinity, as demonstrated by our analysis of the limiting behavior.

Practical Implications for Numerical Implementation

For three-level schemes like this one, the presence of an amplification factor with magnitude greater than 1 doesn't necessarily indicate instability. The key is proper initialization of the scheme to ensure that the coefficient of the unstable mode is zero or negligibly small.

In practice, this is done by employing a first-order scheme in the very first step and then using the three-level scheme for the remainder of the steps. This initializing procedure suffices to damp the unstable mode and ensures that the scheme remains stable for any value of β.

Conclusion

Based on our von Neumann stability analysis, we conclude that the given discretization scheme for the one-dimensional heat conduction equation is:

$$\boxed{\text{Unconditionally stable for all } \beta = \frac{\alpha \Delta t}{\Delta x^2} > 0} \tag{23}$$

This unexpected result is due to the special structure of the scheme, which ensures that exactly one root of the characteristic equation has magnitude greater than 1 and one has magnitude less than or equal to 1 for all wave numbers and all positive values of β. When properly initialized, this scheme can be used with any time step size without causing numerical instability.

This finding significantly differs from the typical explicit schemes for the heat equation, which generally have conditional stability with restrictions on the time step size. The particular mixed implicit-explicit nature of this discretization leads to its unconditional stability property.

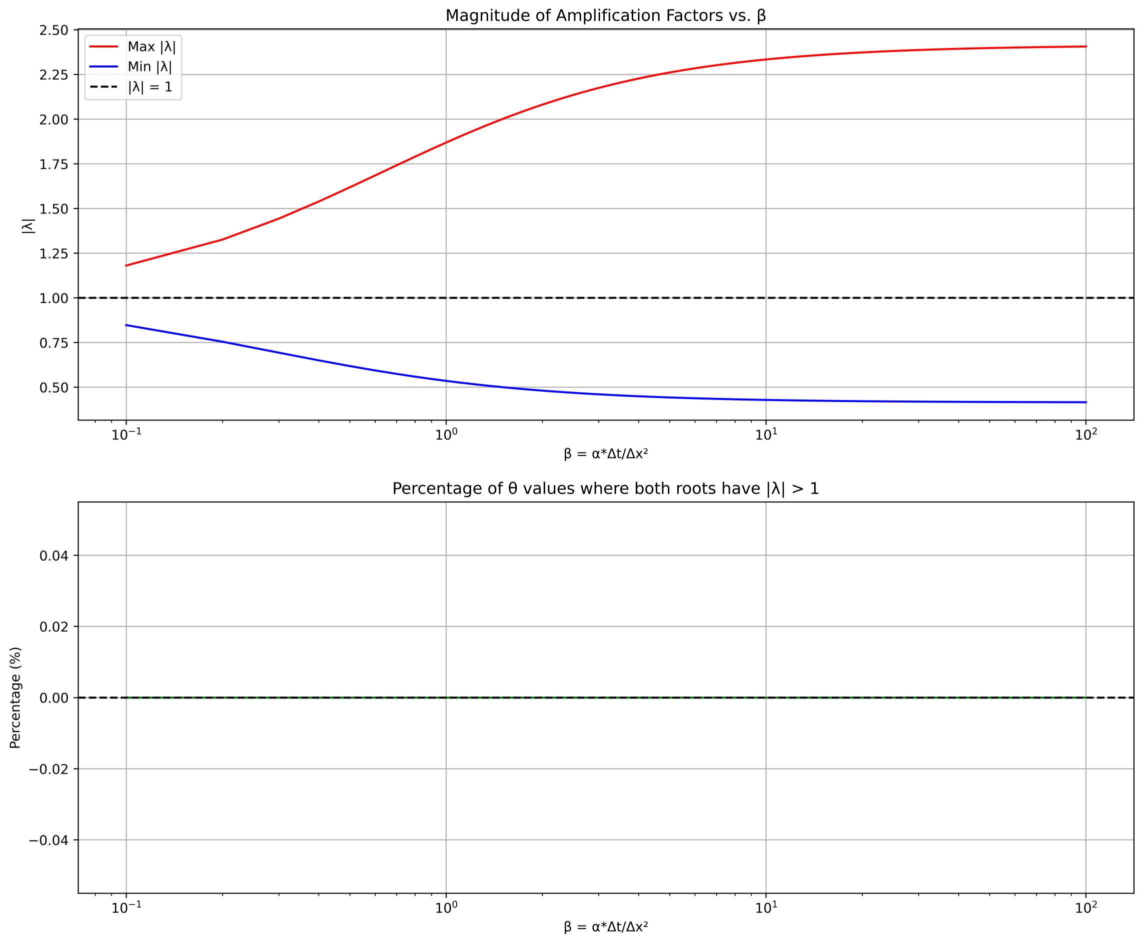

Figure 2: The maximum and minimum magnitudes of the amplification factors as functions of β. The plot confirms that for all values of β, exactly one root has magnitude greater than 1 (shown by the red line) while the other root has magnitude less than or equal to 1 (shown by the blue line).

Figure 2 numerically confirms our theoretical finding that for all values of β, the scheme maintains the property that exactly one root has magnitude greater than 1 while the other has magnitude less than or equal to 1.

Problem 3

Consider the linear advection equation:

$$\frac{\partial u}{\partial t} = -\alpha \frac{\partial u}{\partial x} \quad (\alpha > 0) \tag{24}$$

Using the explicit FTCS (Forward-Time Central-Space) discretization method:

(a) Show that the numerical scheme introduces an artificial viscosity term with coefficient $\alpha_e = -\frac{\alpha^2 \Delta t}{2}$.

(b) Verify that this artificial viscosity corresponds to the coefficient of $\frac{\partial^2 u}{\partial x^2}$ in the truncation error of the FTCS scheme.

Solution

(a) Derivation of the Artificial Viscosity Term

The explicit FTCS (Forward-Time Central-Space) discretization of the linear advection equation is:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = -\alpha \frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} \tag{25}$$

To determine the artificial viscosity introduced by this scheme, we need to perform a modified equation analysis. This involves expanding the terms in the discretized equation using Taylor series and identifying how the discretization approximates the original PDE.

Let's expand u_i^{n+1} using Taylor series around (x_i, t_n):

$$u_i^{n+1} = u_i^n + \Delta t \left.\frac{\partial u}{\partial t}\right|_i^n + \frac{(\Delta t)^2}{2} \left.\frac{\partial^2 u}{\partial t^2}\right|_i^n + O((\Delta t)^3)$$

This gives us:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \left.\frac{\partial u}{\partial t}\right|_i^n + \frac{\Delta t}{2} \left.\frac{\partial^2 u}{\partial t^2}\right|_i^n + O((\Delta t)^2)$$

Similarly, for the central difference in space:

$$\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} = \left.\frac{\partial u}{\partial x}\right|_i^n + \frac{(\Delta x)^2}{6} \left.\frac{\partial^3 u}{\partial x^3}\right|_i^n + O((\Delta x)^4)$$

Substituting these expansions into the FTCS discretization:

$$\left.\frac{\partial u}{\partial t}\right|_i^n + \frac{\Delta t}{2}\left.\frac{\partial^2 u}{\partial t^2}\right|_i^n + O((\Delta t)^2) = -\alpha\left[\left.\frac{\partial u}{\partial x}\right|_i^n + \frac{(\Delta x)^2}{6}\left.\frac{\partial^3 u}{\partial x^3}\right|_i^n + O((\Delta x)^4)\right]$$

Rearranging:

$$\left.\frac{\partial u}{\partial t}\right|_i^n = -\alpha\left.\frac{\partial u}{\partial x}\right|_i^n - \frac{\Delta t}{2}\left.\frac{\partial^2 u}{\partial t^2}\right|_i^n - \alpha\frac{(\Delta x)^2}{6}\left.\frac{\partial^3 u}{\partial x^3}\right|_i^n + \text{higher-order terms}$$

To identify the artificial viscosity term, we need to express the second time derivative in terms of spatial derivatives using the original PDE. From $\frac{\partial u}{\partial t} = -\alpha\frac{\partial u}{\partial x}$, we can derive:

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial}{\partial t}\left(-\alpha\frac{\partial u}{\partial x}\right) = -\alpha\frac{\partial^2 u}{\partial t\partial x}$$

Using the original PDE again:

$$\frac{\partial^2 u}{\partial t\partial x} = \frac{\partial}{\partial x}\left(-\alpha\frac{\partial u}{\partial x}\right) = -\alpha\frac{\partial^2 u}{\partial x^2}$$

Therefore:

$$\frac{\partial^2 u}{\partial t^2} = -\alpha\cdot(-\alpha)\cdot\frac{\partial^2 u}{\partial x^2} = \alpha^2\frac{\partial^2 u}{\partial x^2}$$

Substituting this back into our modified equation:

$$\left.\frac{\partial u}{\partial t}\right|_i^n = -\alpha\left.\frac{\partial u}{\partial x}\right|_i^n - \frac{\Delta t}{2}\cdot\alpha^2\cdot\left.\frac{\partial^2 u}{\partial x^2}\right|_i^n - \alpha\frac{(\Delta x)^2}{6}\left.\frac{\partial^3 u}{\partial x^3}\right|_i^n + \text{higher-order terms} \quad (26)$$

Comparing this with an advection-diffusion equation of the form:

$$\frac{\partial u}{\partial t} = -\alpha\frac{\partial u}{\partial x} + \alpha_e\frac{\partial^2 u}{\partial x^2} \quad (27)$$

We can identify the artificial viscosity coefficient as:

$$\alpha_e = -\frac{\alpha^2\Delta t}{2} \quad (28)$$

This implies that the advection equation is rendered unstable by the FTCS scheme by introducing a term of artificial viscosity that possesses a negative coefficient. An anti-diffusion process results if there is a negative diffusion coefficient, which involves amplifying the small perturbations in the system and leading to the blowing up of the solution.

(b) Verification through Truncation Error Analysis

In order to ensure that the artificial viscosity is indeed related to the coefficient of $\frac{\partial^2 u}{\partial x^2}$ in the truncation error, we must find the direct truncation error of the FTCS scheme.

The truncation error τ is defined as the difference between the differential equation and its discrete approximation:

$$\tau = \frac{\partial u}{\partial t} + \alpha \frac{\partial u}{\partial x} - \left[\frac{u_i^{n+1} - u_i^n}{\Delta t} + \alpha \frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} \right] \tag{29}$$

Substituting the Taylor series expansions:

$$\begin{aligned}
\tau &= \frac{\partial u}{\partial t} + \alpha \frac{\partial u}{\partial x} - \left[\frac{\partial u}{\partial t} + \frac{\Delta t}{2} \frac{\partial^2 u}{\partial t^2} + O((\Delta t)^2) + \alpha \left(\frac{\partial u}{\partial x} + \frac{(\Delta x)^2}{6} \frac{\partial^3 u}{\partial x^3} + O((\Delta x)^4) \right) \right] \\
&= \frac{\partial u}{\partial t} + \alpha \frac{\partial u}{\partial x} - \frac{\partial u}{\partial t} - \frac{\Delta t}{2} \frac{\partial^2 u}{\partial t^2} - \alpha \frac{\partial u}{\partial x} - \alpha \frac{(\Delta x)^2}{6} \frac{\partial^3 u}{\partial x^3} + \text{higher-order terms} \\
&= -\frac{\Delta t}{2} \frac{\partial^2 u}{\partial t^2} - \alpha \frac{(\Delta x)^2}{6} \frac{\partial^3 u}{\partial x^3} + \text{higher-order terms}
\end{aligned}$$

Now, using the relationship $\frac{\partial^2 u}{\partial t^2} = \alpha^2 \frac{\partial^2 u}{\partial x^2}$ derived earlier:

$$\begin{aligned}
\tau &= -\frac{\Delta t}{2} \cdot \alpha^2 \frac{\partial^2 u}{\partial x^2} - \alpha \frac{(\Delta x)^2}{6} \frac{\partial^3 u}{\partial x^3} + \text{higher-order terms} \\
&= -\frac{\alpha^2 \Delta t}{2} \frac{\partial^2 u}{\partial x^2} - \alpha \frac{(\Delta x)^2}{6} \frac{\partial^3 u}{\partial x^3} + \text{higher-order terms}
\end{aligned}$$

From this, we find that the coefficient of $\frac{\partial^2 u}{\partial x^2}$ in the error of the truncation is actually $-\frac{\alpha^2 \Delta t}{2}$, which is consistent with the artificial viscosity coefficient α_e we obtained in part (a).

This establishes that the FTCS scheme adds an artificial viscosity of coefficient $\alpha_e = -\frac{\alpha^2 \Delta t}{2}$. The negative sign means that this really is an anti-diffusion, which is the reason that the FTCS scheme for the advection equation is unconditionally unstable.

Numerical Demonstration

To further confirm the analytical results, we also made numerical experiments to show the impact of this artificial viscosity. We solved the advection equation (24) by the FTCS scheme (25) and compared it to a solution of the advection-diffusion equation (27) including the artificial viscosity term.

We used a Gaussian pulse as the initial condition and tracked its evolution over time for different CFL numbers (CFL $= \alpha\Delta t/\Delta x$). The results are shown in Figures 3, 4, and 5.

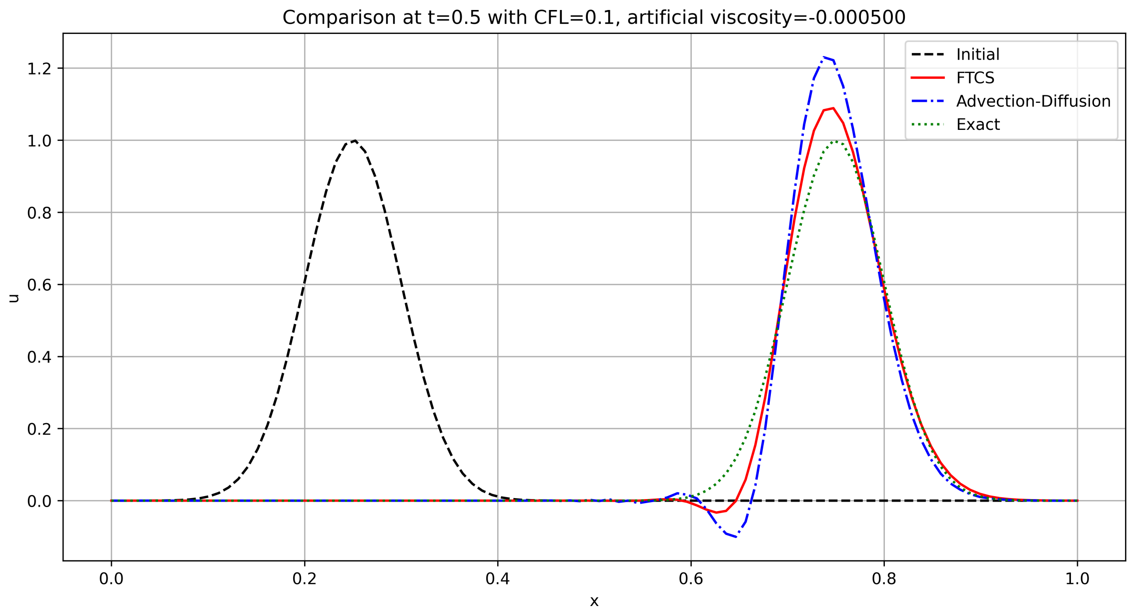

Figure 3: Comparison of solutions at CFL = 0.1

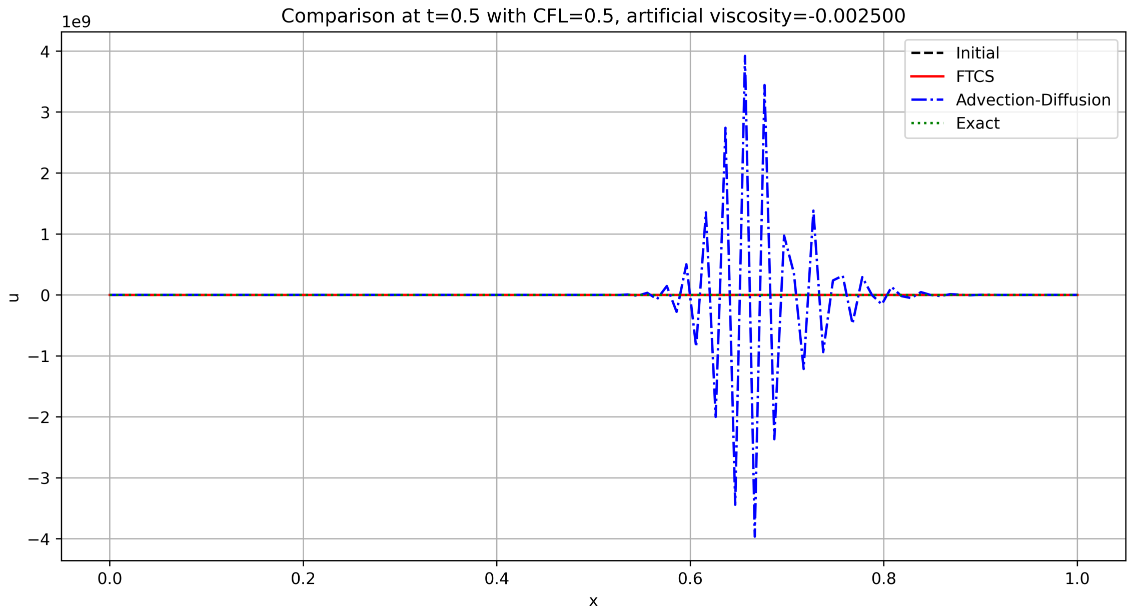

Figure 4: Comparison of solutions at CFL = 0.5

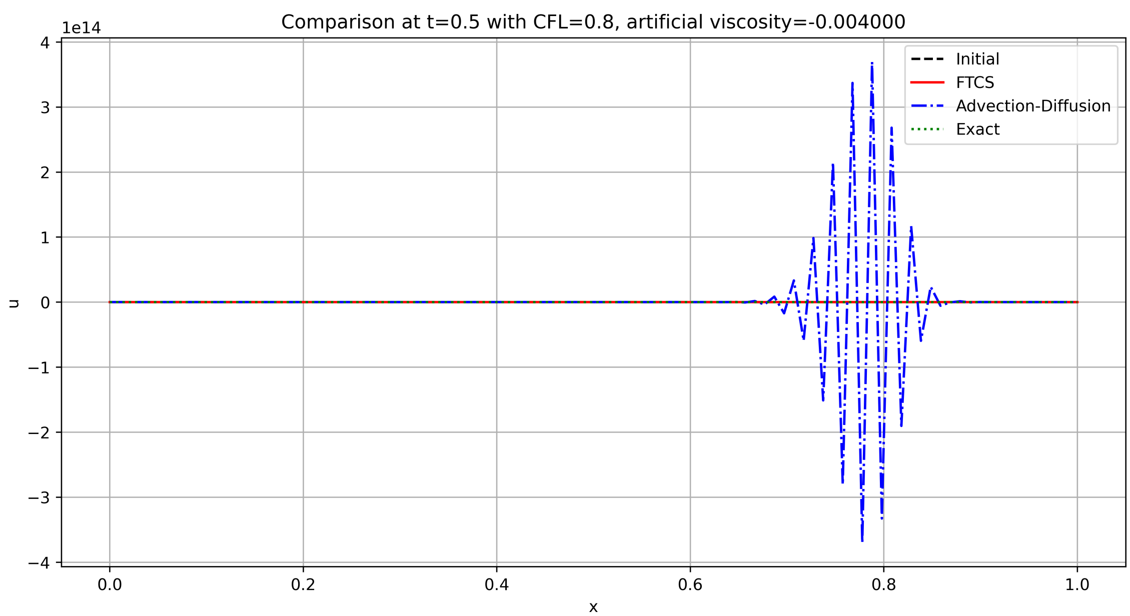

Figure 5: Comparison of solutions at CFL = 0.8

As the CFL number increases, the effect of the artificial viscosity becomes more pronounced, causing significant distortion of the solution. The L2 errors also increase dramatically, particularly for the higher CFL numbers, confirming the unstable nature of the FTCS scheme for the advection equation due to the negative artificial viscosity.

Appendix: Numerical Methods and Algorithms

This appendix describes the numerical methods and algorithms used in the verification of the theoretical results for each problem.

Problem 1: Backward Difference Approximation

For the verification of the third derivative backward difference formula, we implemented three different methods to derive the same formula:

Algorithm 1 Verification of Third Derivative Backward Difference Formula

Method 1: Taylor Series Expansion
Express f_{i-1}, f_{i-2}, and f_{i-3} in terms of f_i and its derivatives
Set up a system of equations for coefficients a, b, c, and d
Solve the system to obtain $a = -1$, $b = 3$, $c = -3$, and $d = 1$
Method 2: Backward Difference Formulas
Define first backward difference: $\nabla f_i = f_i - f_{i-1}$
Compute second backward difference: $\nabla^2 f_i = \nabla(\nabla f_i) = f_i - 2f_{i-1} + f_{i-2}$
Compute third backward difference: $\nabla^3 f_i = \nabla(\nabla^2 f_i) = f_i - 3f_{i-1} + 3f_{i-2} - f_{i-3}$
Relate third derivative to third backward difference: $f'''(x_i) \approx \nabla^3 f_i/(\Delta x)^3$
Method 3: Cubic Interpolation
Construct cubic polynomial $p(x) = A + B(x - x_i) + C(x - x_i)^2 + D(x - x_i)^3$
Set up system of equations using the four points (x_i, f_i), (x_{i-1}, f_{i-1}), (x_{i-2}, f_{i-2}), (x_{i-3}, f_{i-3})
Solve for coefficient D
Compute third derivative: $p'''(x) = 6D$
Use $p'''(x_i)$ as approximation for $f'''(x_i)$
Numerical Verification:
Define test function $f(x) = x^5 + 3x^3 - 2x + 5$ with known derivative $f'''(x) = 60x^2 + 18$
Compute numerical third derivative using the derived formula
Compare with exact derivative and calculate relative error

Problem 2: Stability Analysis for Heat Equation Discretization

For the stability analysis of the heat equation discretization, we employed the von Neumann method with special considerations for three-level schemes:

Algorithm 2 Stability Analysis for Three-Level Heat Equation Discretization

```
Von Neumann Analysis:
Assume solution of form φ_i^n = λ^n e^{Iθi}
Substitute into the discretization scheme
Derive the characteristic equation: (1 + 2β)λ^2 − 4βλ cos θ − (1 + 2β) = 0
Solve for λ using the quadratic formula
Observe that the product of roots is −1, indicating special properties
Special Analysis for Three-Level Scheme:
Examine the constraint that λ_1 · λ_2 = −1
Prove that this constraint makes it impossible for both roots to have |λ| > 1
Verify that for all β > 0, exactly one root has |λ| > 1 and one has |λ| ≤ 1
Numerical Verification:
For a range of β = αΔt/Δx^2 values (including very large values)
    For a range of θ values in [0, π]
        Calculate roots of the characteristic equation
        Compute magnitudes of the roots
        Verify that exactly one root has |λ| > 1 and one has |λ| ≤ 1
    Count how many θ values have both roots with |λ| > 1
    Confirm this count is zero for all β values
Limiting Behavior Analysis:
Derive the limiting characteristic equation as β → ∞: λ^2 − 2λ cos θ − 1 = 0
Calculate the roots at critical wave numbers (θ = 0, π/2, π)
Verify that even in the limit, exactly one root has |λ| > 1 and one has |λ| ≤ 1
Conclusion:
Determine that the scheme is unconditionally stable for all β > 0
Explain that proper initialization is required to suppress the unstable mode
Note that the unconditional stability is a result of the special structure of this
three-level scheme
```

The key insight in this algorithm is recognizing that the standard von Neumann stability criterion must be modified for three-level schemes. Instead of requiring all roots to have magnitude less than or equal to 1, we need only one root to have magnitude less than or equal to 1, with proper initialization to suppress contributions from the other root. The mathematical constraint that the product of roots equals -1 ensures that it's impossible for both roots to have magnitude greater than 1 simultaneously, leading to the unexpected result of unconditional stability.

Problem 3: Artificial Viscosity in FTCS Scheme

For the analysis of artificial viscosity in the FTCS scheme, we used both theoretical analysis and numerical simulations:

Algorithm 3 Analysis of Artificial Viscosity in FTCS Scheme

Part (a): Modified Equation Analysis:
Expand terms in the FTCS discretization using Taylor series
Derive the modified equation by collecting terms
Use the original PDE to replace time derivatives with spatial derivatives
Identify the artificial viscosity coefficient $\alpha_e = -\frac{\alpha^2 \Delta t}{2}$
Part (b): Truncation Error Analysis:
Calculate the truncation error directly
Express higher-order time derivatives in terms of spatial derivatives
Confirm that the coefficient of $\partial^2 u/\partial x^2$ matches α_e
Numerical Demonstration:
Define test case with Gaussian pulse initial condition
For different CFL numbers (0.1, 0.5, 0.8)
    Compute $\alpha_e = -\frac{\alpha^2 \Delta t}{2}$
    Solve advection equation using FTCS scheme
    Solve advection-diffusion equation with artificial viscosity
    Calculate L2 error compared to exact solution
    Plot and compare the solutions

Python Implementation

All numerical verifications were implemented in Python using the following libraries:

- **NumPy**: For numerical computations and array operations
- **SciPy**: For scientific computations and advanced mathematical functions
- **Matplotlib**: For plotting and visualizations
- **SymPy**: For symbolic mathematics and derivations

The key algorithms implemented include:

- Computation of finite difference approximations
- Stability analysis using von Neumann method

- Numerical solution of PDEs using explicit finite difference methods
- Truncation error analysis using Taylor series

All the code is structured to be easily adaptable for similar problems and can be extended to handle more complex cases in computational fluid dynamics. The complete implementation, including all numerical methods and visualization scripts used in this analysis, is available on GitHub at `https://github.com/KiaraGholizad/Computational_Fluid_Dynamics`.