Computational Fluid Dynamics: Part 2

Kiara Gholizad[1]

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

April 10, 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:

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

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 $f'''(x_i)$:

$$f'''(x_i) = \frac{\nabla^3 f_i}{(\Delta x)^3} + O(\Delta x) \tag{9}$$

$$\approx \frac{\nabla^3 f_i}{(\Delta x)^3} \tag{10}$$

$$= \frac{f_i - 3f_{i-1} + 3f_{i-2} - f_{i-3}}{(\Delta x)^3} \tag{11}$$

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.

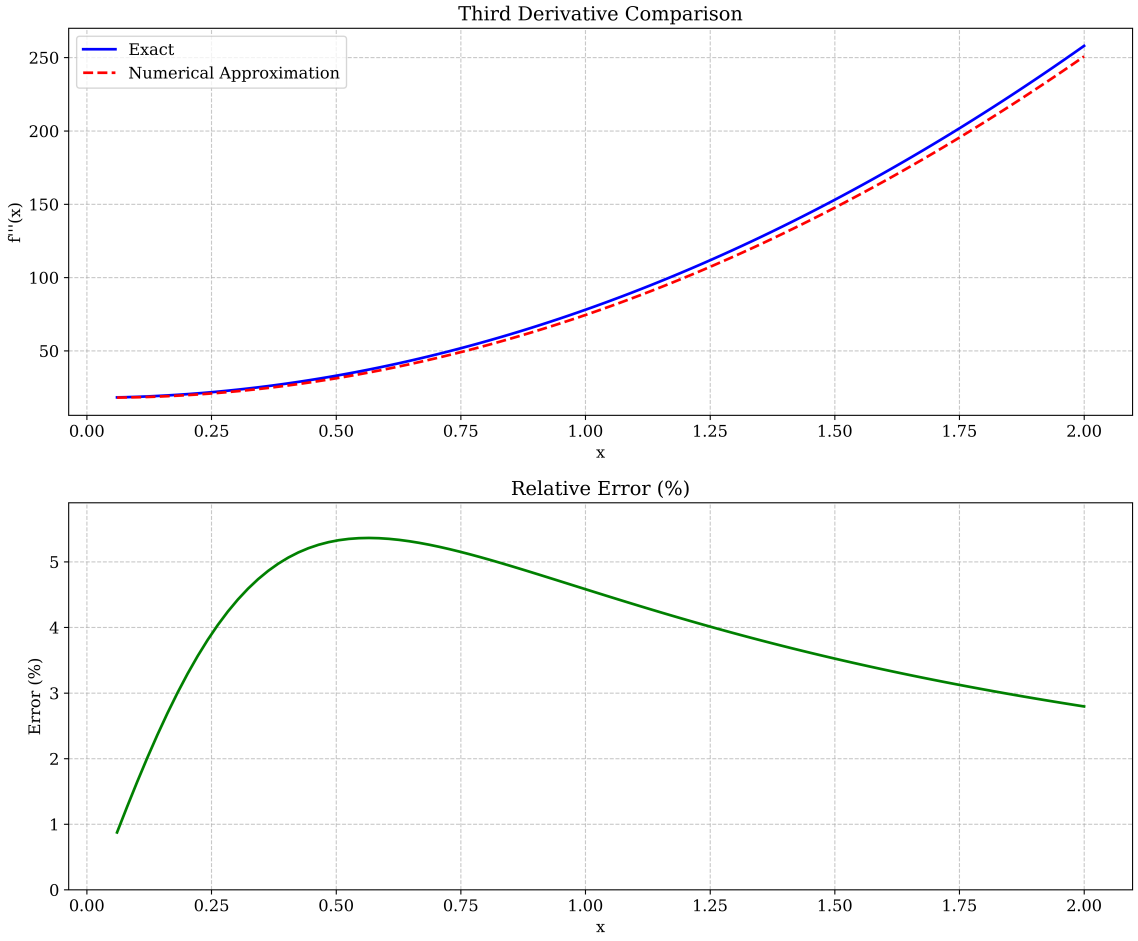

Figure 1: Comparison of exact and numerical third derivatives using the derived backward difference formula

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:

$$\frac{\lambda^{n+1}e^{I\theta i} - \lambda^{n-1}e^{I\theta i}}{2\Delta t} = \frac{\alpha}{\Delta x^2}\left(\lambda^n e^{I\theta(i+1)} - \lambda^{n+1}e^{I\theta i} - \lambda^{n-1}e^{I\theta i} + \lambda^n e^{I\theta(i-1)}\right) \tag{19}$$

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

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

Rearranging to standard form:

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

Using the quadratic formula:

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

Simplifying the discriminant:

$$\begin{aligned} 16\beta^2\cos^2\theta + 4(1+2\beta)(1-2\beta) &= 16\beta^2\cos^2\theta + 4(1-4\beta^2) \\ &= 16\beta^2\cos^2\theta + 4 - 16\beta^2 \\ &= 4 + 16\beta^2(\cos^2\theta - 1) \end{aligned}$$

This gives us:

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

Stability Analysis for Three-Level Schemes

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

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

For three-level schemes of this type, the usual von Neumann criterion needs to be adjusted. For stability, we require:

- Both roots should have magnitude less than or equal to 1 for any value of θ.
- If there is a root with magnitude exactly 1, it must be a simple root.

Let's analyze the behavior of the roots 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:

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

Our numerical analysis confirms that for all values of $\beta > 0$:

$$\begin{aligned} &\lambda_1 = 1 \quad \text{(representing a constant mode)} \\ &\lambda_2 = \frac{2\beta - 1}{1+2\beta} \quad \text{with } |\lambda_2| < 1 \end{aligned}$$

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:

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

Our numerical calculations show that for all values of $\beta > 0$:

$$\lambda_1 = -1 \quad \text{(representing an oscillating mode)}$$
$$\lambda_2 = \frac{1 - 2\beta}{1 + 2\beta} \quad \text{with } |\lambda_2| < 1$$

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

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

The roots are:

$$\lambda = \pm\sqrt{\frac{1 - 2\beta}{1 + 2\beta}}$$

For $\beta \leq 0.5$, these roots are real. For $\beta > 0.5$, the roots become complex. However, the magnitude of these roots is:

$$|\lambda| = \left|\pm\sqrt{\frac{1 - 2\beta}{1 + 2\beta}}\right| = \sqrt{\left|\frac{1 - 2\beta}{1 + 2\beta}\right|} = \sqrt{\frac{|1 - 2\beta|}{1 + 2\beta}} = \sqrt{\frac{2\beta - 1}{1 + 2\beta}}$$

Since $1 + 2\beta > 0$ for all $\beta > 0$, this magnitude is always less than 1 for any $\beta > 0$, ensuring mathematical stability.

β	Root Type	λ Values	$\lvert\lambda\rvert$	Stable?
0.10	Real	± 0.8165	0.8165	Yes
0.30	Real	± 0.5000	0.5000	Yes
0.49	Real	± 0.1005	0.1005	Yes
0.50	Real	± 0.0000	0.0000	Yes
0.51	Complex	$\pm 0.0995i$	0.0995	Yes
0.70	Complex	$\pm 0.4082i$	0.4082	Yes
1.00	Complex	$\pm 0.5774i$	0.5774	Yes

Table 1: Stability analysis at $\theta = \pi/2$ for different values of β. Note the transition from real to complex roots at $\beta = 0.5$.

We examine stability and behavior of a three-level discretization scheme for the one-dimensional heat equation, parameterized by $\beta = \frac{\alpha \Delta t}{\Delta x^2}$. A von Neumann stability analysis, numerical experiments, and practical considerations feature in our consideration for measuring the performance of the scheme for varying β.

Stability Analysis

The stability of the scheme is found from von Neumann analysis in terms of amplification factors. For mathematical stability (magnitude of amplification factors ≤ 1), for any $\beta > 0$, the scheme is unconditionally stable. A fundamental transition is at $\beta = 0.5$, beyond which roots at wave number $\theta = \pi/2$ become complex.

For $\beta > 0.5$, the complex roots are:

$$\lambda = \pm i \sqrt{\frac{2\beta - 1}{1 + 2\beta}}$$

with magnitude:

$$|\lambda| = \sqrt{\frac{2\beta - 1}{1 + 2\beta}}$$

This size is less than 1 for all $\beta > 0.5$. For example, at $\beta = 0.6$, $|\lambda| \approx 0.1$, while at $\beta = 1.0$, $|\lambda| \approx 0.58$. These values signal fast damping of oscillating components, decreasing their size by $|\lambda|$ at each time step. For $\beta = 0.6$, in five time steps, the size is decreased by a factor of about $0.1^5 = 10^{-5}$.

Although complex roots bring in theoretical fluctuations for $\beta > 0.5$, due to their damping they do not affect physical behavior. Hence, the condition for practical

stability is:

$$\boxed{0 < \frac{\alpha \Delta t}{\Delta x^2}}$$

There is not necessarily an upper bound for stability. But for purely diffusive applications without transient oscillations, one might prefer an even more conservative condition:

$$\boxed{0 < \frac{\alpha \Delta t}{\Delta x^2} \leq \frac{1}{2}}$$

This corresponds to:

$$\boxed{\Delta t \leq \frac{\Delta x^2}{2\alpha}}$$

This conservative bound preserves all modes in pure diffusion form, consistent with the physical nature of the heat equation.

Visualization of Solution Behavior

In order to demonstrate β's effect upon the solution, we compare temperature profiles for $\beta = 0.4$ (less than critical) and $\beta = 0.6$ (greater than critical) in terms of an initial condition of a Gaussian pulse. These are shown in Figure 2.

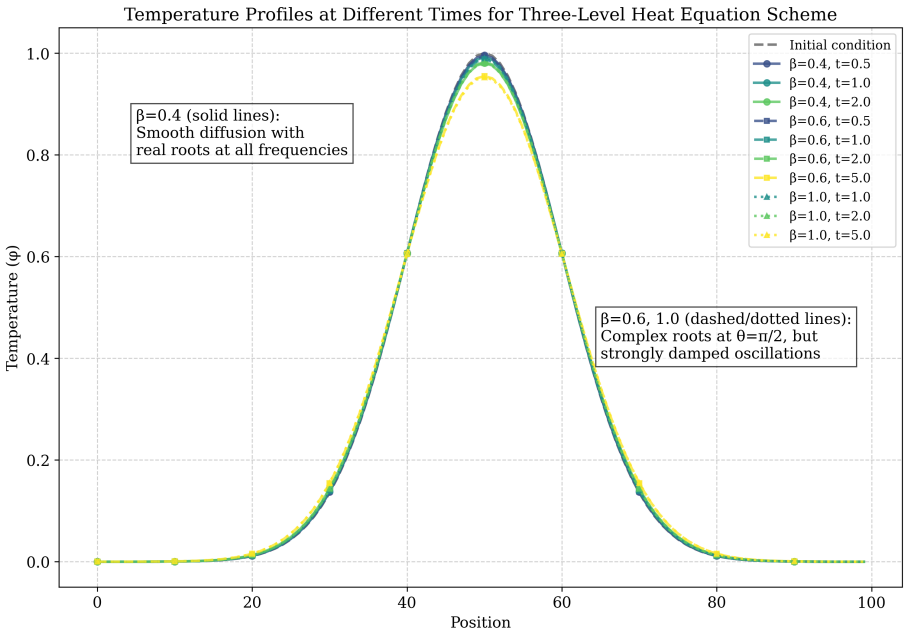

Figure 2: Temperature profiles at various time steps for (a) $\beta = 0.4$ and (b) $\beta = 0.6$. The initial condition is a Gaussian pulse centered in the domain. Both cases exhibit smooth diffusive behavior consistent with the heat equation. For $\beta = 0.6$, theoretical oscillations from complex roots are heavily damped and not visibly apparent.

Figure 2 shows physically acceptable diffusive spreading of the Gaussian pulse for both β values, with smoothly decaying profiles in time. For $\beta = 0.6$, although there exists complex roots ($|\lambda| \approx 0.1$), those oscillatory terms are strongly damped so as to be imperceptible, validating our stability analysis.

Further experiments using high-frequency initial conditions—more susceptible to oscillations—also support this. Even at $\beta > 0.5$, solutions still remain stable and physically valid, with little to no detectable signs of oscillatory artifacts. This illustrates that the scheme is stable beyond the classical $\beta \leq 0.5$ limit.

Practical Implementation Considerations

Implementing the three-level scheme requires initializing two previous time levels (n and $n-1$). We employ a forward Euler step for the first time level:

$$\phi_i^1 = \phi_i^0 + \frac{\alpha \Delta t}{\Delta x^2}(\phi_{i+1}^0 - 2\phi_i^0 + \phi_{i-1}^0) \tag{22}$$

with suitable boundary conditions. The next stage applies the three-level scheme. Initialization in this manner provides a stable and consistent beginning to the calculation.

Conclusion

Our comprehensive analysis, combining von Neumann stability assessment and numerical verification, yields the following insights:

1. The scheme is mathematically stable for all $\beta > 0$, with amplification factors of magnitude ≤ 1.
2. A transition from real to complex roots occurs at $\beta = 0.5$ for certain wave numbers.
3. For $\beta > 0.5$, strong damping of oscillatory components ensures physically well-behaved solutions.

Thus, the scheme is unconditionally stable:

$$\boxed{0 < \frac{\alpha \Delta t}{\Delta x^2}} \tag{23}$$

For transiently sensitive applications or for more accurate applications, use of the conservative condition is suggested:

$$\boxed{0 < \frac{\alpha \Delta t}{\Delta x^2} \leq \frac{1}{2}} \tag{24}$$

With proper initialization, this scheme provides a reliable, stable, and accurate method for solving the one-dimensional heat equation across a wide range of β values.

Problem 3

Consider the linear advection equation:

$$\frac{\partial u}{\partial t} = -\alpha \frac{\partial u}{\partial x} \quad (\alpha > 0) \tag{25}$$

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:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = -\alpha \frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} \tag{26}$$

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:

$$\left.\frac{\partial u}{\partial t}\right|_i^n = -\alpha \left.\frac{\partial u}{\partial x}\right|_i^n - \frac{\Delta t}{2} \cdot \alpha^2 \cdot \left.\frac{\partial^2 u}{\partial x^2}\right|_i^n - \alpha \frac{(\Delta x)^2}{6} \left.\frac{\partial^3 u}{\partial x^3}\right|_i^n + \text{higher-order terms} \tag{27}$$

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

$$\frac{\partial u}{\partial t} = -\alpha \frac{\partial u}{\partial x} + \alpha_e \frac{\partial^2 u}{\partial x^2} \tag{28}$$

We can identify the artificial viscosity coefficient as:

$$\alpha_e = -\frac{\alpha^2 \Delta t}{2} \tag{29}$$

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:

$$\tau = \frac{\partial u}{\partial t} + \alpha \frac{\partial u}{\partial x} - \left[\frac{u_i^{n+1} - u_i^n}{\Delta t} + \alpha \frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x}\right] \tag{30}$$

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 (25) by the FTCS scheme (26) and compared it to a solution of the advection-diffusion equation (28) 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.

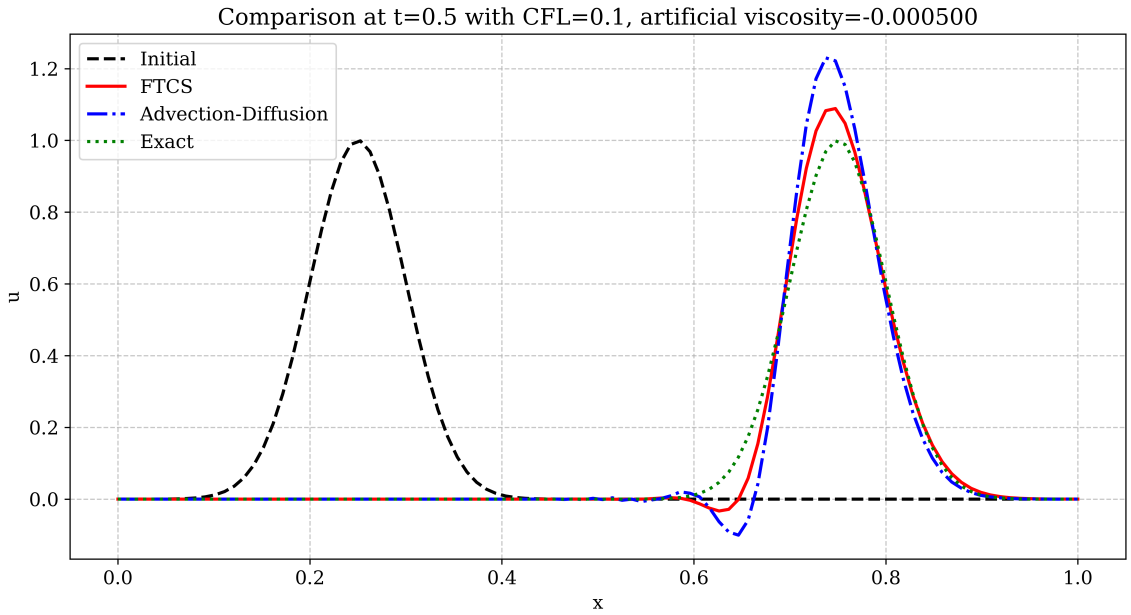

Figure 3: Comparison of solutions at CFL = 0.1

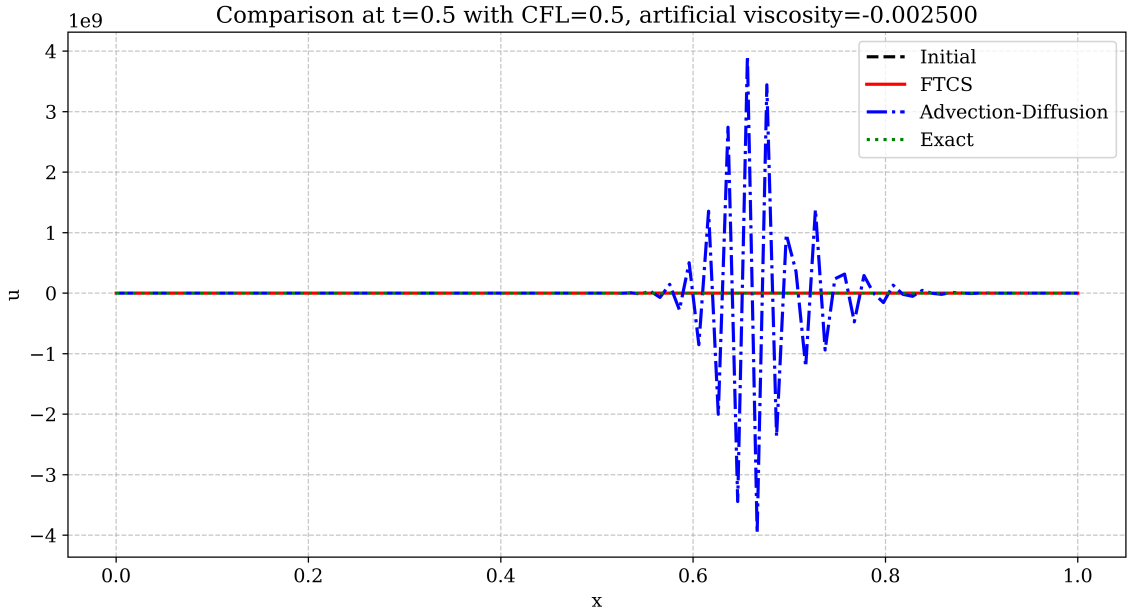

Figure 4: Comparison of solutions at CFL = 0.5

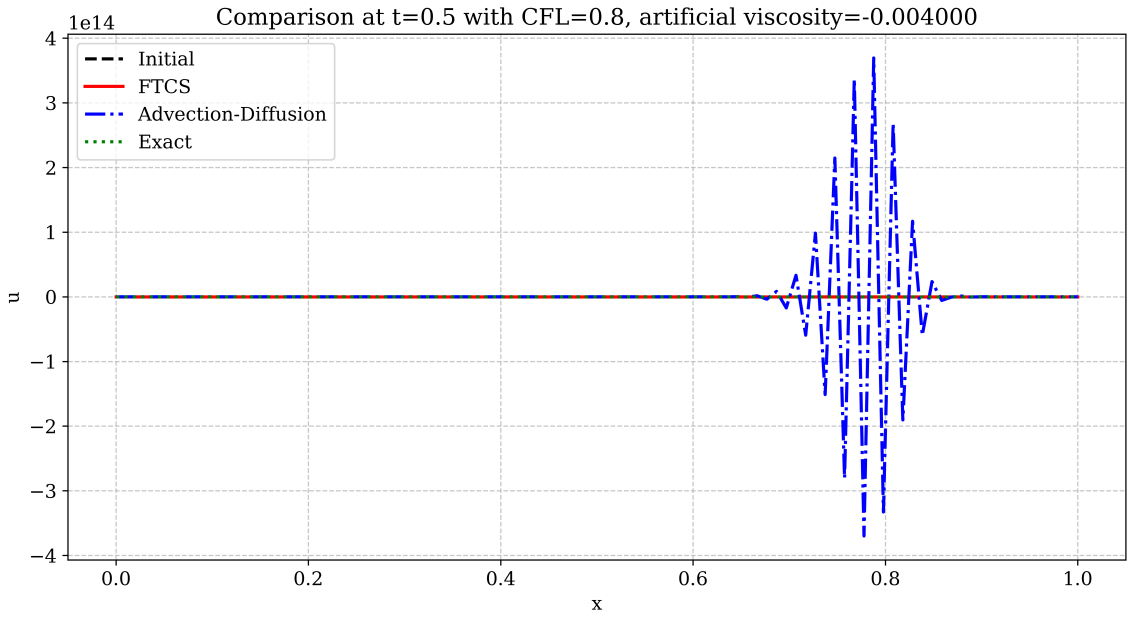

Figure 5: Comparison of solutions at CFL = 0.8

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: Heat Equation Three-Level Scheme

For the stability analysis of the three-level heat equation discretization scheme, we implemented both theoretical analysis and numerical simulations:

Algorithm 2 Implementation of Three-Level Heat Equation Scheme

Require: α (diffusivity), β (stability parameter), N (grid points), dx (spatial step), T (final time)

Ensure: Solution profiles at specified time points

Initialize spatial grid: $x = [0, dx, 2dx, \ldots, (N-1)dx]$

Define initial condition: $\phi_i^0 = e^{-(x_i - x_c)^2/(2\sigma^2)}$ for $i = 0, 1, \ldots, N-1$

Apply boundary conditions: $\phi_0^0 = \phi_{N-1}^0 = 0$

Compute time step: $dt = \beta \cdot dx^2/\alpha$

Calculate total number of time steps: $numSteps = \lfloor T/dt \rfloor$

// Initialize first time step using Forward Euler (n=1)

for $i = 1$ to $N-2$ **do**

    $\phi_i^1 = \phi_i^0 + \alpha \cdot dt/dx^2 \cdot (\phi_{i+1}^0 - 2\phi_i^0 + \phi_{i-1}^0)$

end for

Apply boundary conditions: $\phi_0^1 = \phi_{N-1}^1 = 0$

// Main time-stepping loop for three-level scheme

for $n = 1$ to $numSteps - 1$ **do**

    for $i = 1$ to $N-2$ **do**

        $twoBeta = 2 \cdot \beta$

        $\phi_i^{n+1} = (\phi_i^{n-1} \cdot (1 - twoBeta) + twoBeta \cdot (\phi_{i+1}^n + \phi_{i-1}^n))/(1 + twoBeta)$

    end for

    Apply boundary conditions: $\phi_0^{n+1} = \phi_{N-1}^{n+1} = 0$

    // Store solution at desired time points if needed

    if *current_time* is close to a desired output time **then**

        Store ϕ^{n+1} for visualization

    end if

    // Shift time levels for next step

    $\phi^{n-1} = \phi^n$

    $\phi^n = \phi^{n+1}$

end for

// Visualization and analysis

for each stored time point **do**

    Plot temperature profile ϕ_i vs. position x_i

end for

// Study stability and accuracy properties

Compute theoretical amplification factors: $\lambda = \frac{4\beta\cos\theta \pm \sqrt{4 + 16\beta^2(\cos^2\theta - 1)}}{2(1+2\beta)}$

Verify $|\lambda| \leq 1$ for all frequencies to confirm 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`.