

Seminar 2

Implement the numerical methods using a programming language of your choice (Python, MATLAB, C++, etc.). For each problem, present the numerical results graphically and briefly discuss their accuracy and stability.

Problem 1. Advection Equation

Consider the one-dimensional linear advection equation

$$\frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = 0, \quad a = 1, \quad (1)$$

on the periodic domain $0 \leq x \leq 1$, with the initial condition

$$u(x, 0) = \begin{cases} 1, & 0.3 \leq x \leq 0.5, \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

Tasks:

1. Implement the first-order upwind method.
2. Implement the Lax–Wendroff method.
3. Use $\Delta x = 0.01$ and the Courant number $C = a\Delta t/\Delta x = 0.8$.
4. Plot both numerical solutions at $t = 0.5$ together with the analytical solution.
5. Compare numerical diffusion and oscillations near the discontinuities.

Questions: Which method preserves the discontinuities more sharply? Which method produces numerical oscillations?

Problem 2. Heat Equation

Consider the one-dimensional heat equation

$$\frac{\partial u}{\partial t} = \kappa \frac{\partial^2 u}{\partial x^2}, \quad \kappa = 0.01, \quad (3)$$

on $0 \leq x \leq 1$, with initial and boundary conditions

$$u(x, 0) = \sin(\pi x), \quad u(0, t) = u(1, t) = 0. \quad (4)$$

Tasks:

1. Implement the explicit forward-time, centered-space (FTCS) scheme

$$u_j^{n+1} = u_j^n + r \left(u_{j+1}^n - 2u_j^n + u_{j-1}^n \right), \quad r = \frac{\kappa \Delta t}{\Delta x^2}. \quad (5)$$

2. Use $\Delta x = 0.02$.

3. Compute the solution up to $t = 1$.
4. Compare the numerical solution with

$$u(x, t) = e^{-\kappa\pi^2 t} \sin(\pi x). \quad (6)$$

5. Repeat the calculation for $r = 0.4$ and $r = 0.6$.

Questions: How does the numerical solution depend on the time step? What happens when $r > 0.5$?

Problem 3. Wave Equation

Consider the one-dimensional wave equation

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}, \quad c = 1, \quad (7)$$

on $0 \leq x \leq 1$, with initial conditions

$$u(x, 0) = \sin(\pi x), \quad \left. \frac{\partial u}{\partial t} \right|_{t=0} = 0, \quad (8)$$

and boundary conditions

$$u(0, t) = u(1, t) = 0. \quad (9)$$

Tasks:

1. Implement the centered finite-difference scheme

$$u_j^{n+1} = 2u_j^n - u_j^{n-1} + C^2 (u_{j+1}^n - 2u_j^n + u_{j-1}^n), \quad C = \frac{c\Delta t}{\Delta x}. \quad (10)$$

2. Use $\Delta x = 0.01$ and $C = 0.8$.
3. Initialize the first time step using the prescribed initial conditions.
4. Plot the solutions at $t = 0$, $t = 0.5$, and $t = 1$.
5. Compare with the analytical solution

$$u(x, t) = \cos(\pi t) \sin(\pi x). \quad (11)$$

Questions: Does the numerical solution preserve the wave amplitude? Are there any phase errors?

Problem 4. Burgers' Equation

Consider the inviscid Burgers equation

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} \left(\frac{u^2}{2} \right) = 0, \quad (12)$$

on $0 \leq x \leq 1$, with the initial condition

$$u(x, 0) = \begin{cases} 1, & x < 0.5, \\ 0, & x \geq 0.5. \end{cases} \quad (13)$$

Use the boundary conditions $u(0, t) = 1$ and $u(1, t) = 0$.

Tasks:

1. Implement a first-order conservative upwind (Godunov) scheme.
2. Use $\Delta x = 0.01$ and choose the time step such that

$$\frac{\max_j |u_j^n| \Delta t}{\Delta x} \leq 0.8. \quad (14)$$

3. Compute the solution up to $t = 0.3$.
4. Plot the numerical solution at several times.
5. Compare the shock position with the analytical shock speed $s = 0.5$.

Questions: Does the numerical method reproduce the correct shock propagation speed? How does numerical diffusion affect the shock width?

Problem 5. Poisson Equation

Consider the one-dimensional Poisson equation

$$\frac{d^2 u}{dx^2} = -\pi^2 \sin(\pi x), \quad 0 \leq x \leq 1, \quad (15)$$

with boundary conditions

$$u(0) = u(1) = 0. \quad (16)$$

Tasks:

1. Approximate the second derivative using centered finite differences.
2. Solve the resulting system using Jacobi iteration, starting from $u_j^{(0)} = 0$ at interior nodes.
3. Stop the iteration when

$$\max_j \left| u_j^{(k+1)} - u_j^{(k)} \right| < 10^{-10}. \quad (17)$$

4. Compare the numerical solution with

$$u(x) = \sin(\pi x). \quad (18)$$

5. Repeat the calculation using $N = 20, 40$, and 80 grid intervals.
6. Plot the maximum error as a function of the grid spacing.

Questions: How does the numerical error depend on the grid spacing? How does the number of Jacobi iterations change when the grid is refined?

Submission Requirements

For each problem, submit:

1. The program source code.
2. Plots of the numerical solutions.
3. Comparison with the analytical solution, where available.
4. Brief answers to the questions.