AM213B Assignment #7

Problem 1 (Theoretical)

Consider the Lax-Friedrichs method for solving $u_t + a\, u_x = 0$

$$u_i^{n+1} = \frac{u_{i+1}^n + u_{i-1}^n}{2} - \frac{ar}{2}\left(u_{i+1}^n - u_{i-1}^n\right), \qquad r = \frac{\Delta t}{\Delta x}$$

On the RHS, we write $\dfrac{u_{i+1}^n + u_{i-1}^n}{2}$ as $u_i^n + \underbrace{\frac{1}{2}\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right)}_{\text{Added viscosity}}$.

We know that the Lax-Friedrichs method has too much added viscosity. So we consider a modified version of Lax-Friedrichs

$$u_i^{n+1} = u_i^n + \frac{q}{2}\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right) - \frac{ar}{2}\left(u_{i+1}^n - u_{i-1}^n\right), \quad r = \frac{\Delta t}{\Delta x}, \quad 0 \le q \le 1 \qquad \text{(LF-2)}$$

Part 1: Find the modified PDE of (LF-2).

Part 2: Find the modified PDE of the implicit upwind method

$$u_i^{n+1} = u_i^n - ar\left(u_i^{n+1} - u_{i-1}^{n+1}\right), \qquad r = \frac{\Delta t}{\Delta x}$$

Hint: Expanding around (x_i, t_{n+1}) will make it easier.

Problem 2 (Computational)

Consider the 2D IBVP of the heat equation (similar to that in Assignment 6).

$$\begin{cases} u_t = u_{xx} + u_{yy}, \quad (x,y) \in (0,8)\times(0,8), \quad t > 0 \\ u(x,y,0) = f(x,y) \\ u_x(0,y,t) = g_L(y,t), \quad u(8,y,t) = g_R(y,t) \\ u(x,0,t) = g_B(x,t), \quad u(x,8,t) = g_T(x,t) \end{cases}$$

where $f(x,y) = 0$

$$g_L(y,t) = -2\sin(\pi y/8)\cdot\tanh(2t), \quad g_R(y,t) = 2\sin(\pi y/8)\cdot\tanh(2t)$$
$$g_B(x,t) = g_T(x,t) = -x(1-x/8)\cdot\tanh(2t)$$

Numerical grid:

We need to design the grid to accommodate $u_x(0,y,t) = g_L(y,t)$.

$$\Delta x = \frac{L_x}{N_x - 0.5}, \qquad x_i = (i-0.5)\Delta x, \quad i = 0, 1, \ldots, N_x$$

$$\Delta y = \frac{L_y}{N_y}, \qquad y_j = j\Delta x, \quad j = 0, 1, \ldots, N_y$$

The left BC: $u_x(0, y, t) = g_L(y, t)$

$$\left. \frac{u_{i+1,j}^n - u_{i,j}^n}{\Delta x} \right|_{i=0} = g_L(y_j, t_n) \qquad \text{==>} \qquad u_{0,j}^n = u_{1,j}^n - \Delta x\, g_L(y_j, t_n)$$

Caution: In the top and bottom BCs, $x_i = (i-0.5)\Delta x$, different from Assignment 5.

Use $N_x = N_y = 100$ and $\Delta t = 1.25\times10^{-3}$ to solve the IBVP to $T = 20$.

Part 1: Plot $u(x, 2)$ vs x at $t = 0.5, 5, 10$ and 20 in one figure.

Part 2: Plot u vs (x, y) at $T = 20$ as a surface.

Problem 3 (Computational)

Consider the IBVP.

$$\begin{cases} u_t + a u_x = 0, & x \in (-0.5, 2.5), \quad t > 0 \\ u(x, 0) = f(x), & x \in (-0.5, 2.5) \\ u(-0.5, t) = g(t), & t > 0 \end{cases}$$

where $a = 1.5$, $\quad f(x) = -\cos(\pi x)$, $\quad g(t) = \sin(a\pi t)$

The exact solution is expressed in terms of $f(x)$ and $g(t)$ as

$$u_{\text{ext}}(x,t) = \begin{cases} f(x - at), & x - at > -0.5 \\ g\big(t - (x+0.5)/a\big), & x - at \le -0.5 \end{cases}$$

Implement the 3 methods below to solve this IBVP.

- Upwind method
- Lax-Friedrichs method
- Lax-Wendroff method

Use the numerical grid

$$\Delta x = \frac{3}{N}, \quad x_j = -0.5 + j\Delta x, \quad x_0 = -0.5, \quad x_N = 2.5$$

The IBVP specifies a boundary condition only at $x_0 = -0.5$.

In the Lax-Friedrichs and Lax-Wendroff methods, we need an *artificial numerical boundary condition* at $x_N = 2.5$. In this problem, we use the exact solution.

$$u_N^n = u_{\text{ext}}(x_N, t_n)$$

In simulations, use $N = 300$ and $\Delta t = r\,\Delta x$ with, respectively, $r = 0.3$ and $r = 0.6$.

Use the exact solution to calculate the error of each method.

$$E_i^n = \left|u_{\text{ext}}(x_i, t_n) - u_i^n\right|$$

Part 1: Plot errors vs x of the 3 methods at $t = 1.08$ in one figure. Plot two figures, one for $r = 0.3$ and the other for $r = 0.6$. Use log scale for errors.

Part 2: Which method has the smallest error? Which r value yields a smaller error?

Problem 4 (Computational)

Continue with the IBVP in Problem 3. Change the initial value to

$$f(x) = \cos(\pi x), \quad x \in (-0.5, 2.5)$$

Change the artificial numerical boundary condition at $x_N = 2.5$ to

Condition 1: $u_N^n = 2u_{N-1}^n - u_{N-2}^n$ (extrapolation)

In simulations, use $N = 300$ and $r = 0.3$.

Part 1: Plot numerical solutions of the 3 methods and the exact solution at $t = 1.08$ in one figure. Use linear scale for solutions.

Part 2: Plot errors vs x of the 3 methods at $t = 1.08$ in one figure. Use log scale for errors.

Part 3: Observe the connection between the location of maximum error and the location of the cusp in the exact solution.

Observe whether or not the effect of artificial numerical boundary condition encroaches into the interior of computational region.

Problem 5 (Computational)

Continue with the IBVP in Problem 4. Change the artificial BC at $x_N = 2.5$ to

Condition 2: $u_N^n = 0$ (an ad hoc boundary condition)

Part 1: Use $N = 300$ and $r = 0.3$ in simulations of the 3 methods.

Plot the exact errors vs x of the 3 methods at $t = 1.08$ in one figure. Use log scale for errors. Observe whether or not the ad hoc BC affects the deep interior of computational region, far away from $x_N = 2.5$.

Part 2: We study only the Lax-Wendroff method in this part. Keep r = 0.3. Use the numerical solutions of N = 300 and N = 600 to estimate the error in $u_{\{N=300\}}$ at t = 1.08.

Plot the estimated error vs x and the exact error vs x at t =1.08 in one figure for comparison. Use log scale for the errors.

Problem 6 (Computational)

Consider the IVP of linear hyperbolic PDE with variable coefficients.

$$\begin{cases} u_t + \left(\sin(x)+\cos(x)\right)u_x = -\cos(x)u\,, & x \in (-\infty, +\infty)\,, \quad t > 0 \\ u(x,0) = \cos^2(x)\,, & x \in (-\infty, +\infty) \end{cases}$$

Implement the method of characteristics. Write a code to calculate $u(x, t)$ at any given point ($x = \xi$, $t = T$). In your implementation, use RK4 ODE solver with h = 0.005 (h = –0.005 when tracing back characteristics).

Test your code at (x =1 and t = 0.5). You should get

$u(1, 0.5) \approx 0.6129394$

Use your code to calculate $u(x, t)$ at three time levels: t = 0.2, t = 0.5, t = 1. At each time level, use about 400 points to represent u for $x \in [0, 4\pi]$.

Plot $u(x, t)$ vs x for $x \in [0, 4\pi]$ at these 3 time levels in one figure.