AM213B Assignment #8

Problem 1 (Computational)

Consider the IVP of Burgers' equation

$$\begin{cases} u_t + \left(\frac{1}{2}u^2\right)_x = 0, \quad t > 0 \\ u(x,0) = \begin{cases} \frac{-1}{2}, & x \le 0 \\ 1, & 0 < x \le 1 \\ 0, & x > 1 \end{cases} \end{cases} \qquad \text{(IVP-1)}$$

For $t \le 2$, the exact solution of (IVP-1) is

$$u_{\text{ext}}(x,t) = \begin{cases} \frac{-1}{2}, & x \le \frac{-1}{2}t \\ \frac{x}{t}, & \frac{-1}{2}t < x \le t \\ 1 & t < x \le 1 + \frac{1}{2}t \\ 0 & x > 1 + \frac{1}{2}t \end{cases}$$

We implement the three methods below to solve (IVP-1).

- Upwind method 1 (with no entropy fix)

$$F_{i+1/2}^{(\text{Up})} = \frac{1}{2}\left(F(u_{i+1}^n) + F(u_i^n)\right) - \frac{1}{2}\alpha(u_i^n, u_{i+1}^n)(u_{i+1}^n - u_i^n)$$

$$\alpha(u_i^n, u_{i+1}^n) = \frac{F(u_{i+1}^n) - F(u_i^n)}{u_{i+1}^n - u_i^n} = \frac{\frac{1}{2}\left(u_{i+1}^n\right)^2 - \frac{1}{2}\left(u_i^n\right)^2}{u_{i+1}^n - u_i^n} = \frac{1}{2}(u_i^n + u_{i+1}^n)$$

- Upwind method 2 (with LeVeque entropy fix)

$$F_{i+1/2}^{(\text{Up})} = \frac{1}{2}\left(F(u_{i+1}^n) + F(u_i^n)\right) - \frac{1}{2}\psi_{i+1/2}(u_{i+1}^n - u_i^n)$$

$$\psi_{i+1/2} = \max\left\{\left|\alpha(u_i^n, u_{i+1}^n)\right|, -F'(u_i^n), F'(u_{i+1}^n)\right\}$$

- Lax-Wendroff method

$$F_{i+1/2}^{(\text{LW})} = \frac{1}{2}\left(F(u_{i+1}^n) + F(u_i^n)\right) - \frac{\Delta t}{2\Delta x}\alpha(u_i^n, u_{i+1}^n)^2(u_{i+1}^n - u_i^n)$$

We select $[L_1, L_2]$ with $L_1 = -1$ and $L_2 = 2$ as the computational domain.

<u>We use the finite volume discretization</u>: viewing x_i as the center of cell i.

$$\Delta x = \frac{L_2 - L_1}{N}, \qquad x_i = L_1 + (i - 0.5)\Delta x, \quad i = 0, 1, \ldots, N+1$$

$$x_0 = L_1 - 0.5\Delta x\,,\quad x_1 = L_1 + 0.5\Delta x\,,\ldots,\quad x_N = L_2 - 0.5\Delta x\,,\quad x_{N+1} = L_2 + 0.5\Delta x$$

To calculate $\left\{u_i^{n+1}, 1 \le i \le N\right\}$ in each time step, we need u^n at x_0 and at x_{N+1}. In this problem, we use artificial boundary conditions: $u_0^n = u_1^n\,,\quad u_{N+1}^n = u_N^n$.

Use N = 300 and $r = \Delta t/\Delta x$ = 0.5 in simulations.

Part 1: Plot in one figure, the exact solution and numerical solutions of the three methods at t = 1. Which methods deviate substantially from the exact solution?

Part 2: Plot in one figure, numerical solutions of upwind method 2 at t = 0, t = 1, t = 1.5, t = 3, and t = 6 to show the time evolution.

Does any characteristic at boundaries ever go into the computational domain?

Remark: When all characteristics at boundaries are going out of the computational domain, the artificial boundary conditions will not affect the interior of the domain.

Problem 2 (Computational)

Continue with (IVP-1) and the upwind method 2 in Problem 1.

We use the same $[L_1, L_2]$, the same BCs and N = 300 as in Problem 1.

Part 1: Use $r = \Delta t/\Delta x = 10/8$, which is above the CFL condition ($r \le 1$). Plot the numerical solution of upwind method 2 at t = 0.5. You will see huge oscillations.

Part 2: Use $r = \Delta t/\Delta x = 10/8.5$, which is above the CFL condition ($r \le 1$). Plot in one figure the numerical solution of upwind method 2 and the exact solution at t = 1.6. You will see a different effect of violating the CFL condition.

Problem 3 (Computational)

Consider the IVP of conservation law

$$\begin{cases} u_t + \left(\frac{1}{4}u^4\right)_x = 0\,, & t > 0 \\ u(x,0) = \sin(\pi x) \end{cases} \qquad \text{(IVP-2)}$$

Implement the upwind method 2 to solve (IVP-2). Note that for (IVP-2), we have

$$\alpha(u_i^n, u_{i+1}^n) = \frac{F(u_{i+1}^n) - F(u_i^n)}{u_{i+1}^n - u_i^n} = \frac{\frac{1}{4}\left(u_{i+1}^n\right)^4 - \frac{1}{4}\left(u_i^n\right)^4}{u_{i+1}^n - u_i^n}$$

$$= \frac{1}{4}\left[\left(u_{i+1}^n\right)^3 + \left(u_{i+1}^n\right)^2\left(u_i^n\right) + \left(u_{i+1}^n\right)\left(u_i^n\right)^2 + \left(u_i^n\right)^3\right]$$

We select $[L_1, L_2]$ with $L_1 = 0$ and $L_2 = 4$ as the computational domain.

Since (IVP-2) is periodic, we use periodic boundary conditions: $u_0^n = u_N^n$, $u_{N+1}^n = u_1^n$.

Use $N = 400$ and $r = \Delta t/\Delta x = 0.5$ in simulations.

Part 1: Plot in one figure, $u(x, t)$ vs x at $t = 0, 1, 3, 10$, and 40.

Part 2: Plot in one figure, $\left(u(x,t) / \max_x u(x,t) \right)$ vs x at $t = 0, 1, 3, 10$, and 40. Do your results support the assertion that $u(x, t)$ vs x has a similar shape for large t?

Problem 4 (Computational)

We use the method of characteristics to solve the 2D IVP below

$$\begin{cases} \dfrac{\partial u(x,y,t)}{\partial t} + \nabla \cdot \left(\vec{a}(x,y) u(x,y,t) \right) = 0 \\ u(x,y,0) = u_0(x,y) \equiv \sin^2(x+y) \end{cases} \qquad \text{(IVP-2D)}$$

where

$$\vec{a}(x,y) = \begin{pmatrix} a_1(x,y) \\ a_2(x,y) \end{pmatrix} \equiv \begin{pmatrix} \sin(x)\sin(y) \\ 1 - \exp\left(\sin(x+y)\right) \end{pmatrix}$$

The whole problem is periodic in both x and y directions with period = 2π.

We first write out the divergence and write the PDE as

$$\frac{\partial u}{\partial t} + a_1(x,y)\frac{\partial u}{\partial x} + a_2(x,y)\frac{\partial u}{\partial y} = b(x,y)u$$

where

$$b(x,y) = -\frac{\partial a_1}{\partial x} - \frac{\partial a_2}{\partial y} = -\cos(x)\sin(y) + \exp\left(\sin(x+y)\right)\cos(x+y)$$

Our goal is to calculate the solution of (IVP-2D) at any given point (ξ, η, T).

The method of characteristics consists of the two steps below.

- Tracing back the C-line from (ξ, η, T) to time 0

$$\begin{aligned} &\frac{dX}{dt} = a_1(X,Y) \\ &\frac{dY}{dt} = a_2(X,Y) \\ &X(T) = \xi, \quad Y(T) = \eta \end{aligned} \qquad \text{(FVP-C)}$$

  We use an ODE solve to solve (FVP-C) from $t = T$ to $t = 0$.

With the solution of (FVP-C), we set $x_0 = X(0)$ and $y_0 = Y(0)$.

- Advancing from $(x_0, y_0, 0)$ to (ξ, η, T).

$$\begin{aligned}
&\frac{dx}{dt} = a_1(x, y) \\
&\frac{dy}{dt} = a_2(x, y) \\
&\frac{dv}{dt} = b(x, y)v \\
&x(0) = x_0, \quad y(0) = y_0, \quad v(0) = u_0(x_0, y_0)
\end{aligned} \qquad \text{(IVP-C)}$$

We use an ODE solve to solve (IVP-C) from $t = 0$ to $t = T$.

The solution of the (IVP-2D) at (ξ, η, T) is given by $u(\xi, \eta, T) = v(T)$.

Write a code to calculate $u(x, y, T)$ at any given point (x, y, T).

In your implementation, use RK4 with $h = 0.01$ ($h = -0.01$ in tracing back).

Test your code at $(x, y, T) = (3.9, 2.3, 1.2)$. You should get $u(3.9, 2.3, 1.2) \approx 5.340824$

Part 1: Set $x_1 = 3.9$. Calculate and plot $u(x_1, y, T)$ as a function of y for $T = 0.75$, 1.0, and 1.25 in one figure. Use about 300 points for y in $[0, 2\pi]$.

Part 2: Set $x_1 = 2.5$. Calculate and plot $u(x_1, y, T)$ as a function of y for $T = 0.75$, 1.0, and 1.25 in one figure. Use about 300 points for y in $[0, 2\pi]$.

Problem 5 (Computational)

Continue with (IVP-2D) in problem 4.

For each set of $(x_1, y_1) = (3.9, 2.3)$, $(2.7, 4.0)$, and $(2.0, 3.0)$, calculate $u(x_1, y_1, t)$ as a function of t. Plot $u(x_1, y_1, t)$ vs t for the 3 sets of (x_1, y_1) in one figure.

Problem 6 (Computational)

Continue with (IVP-2D) in problem 4. Consider the numerical grid on (x, y):

$$\Delta x = \Delta y = \frac{2\pi}{N}, \qquad x_i = i\Delta x, \quad 0 \le i \le N, \qquad y_j = j\Delta x, \quad 0 \le j \le N$$

Use $N = 80$ and calculate $u(x, y)$ on the grid for $T = 0.0$, 0.5, 1.0, and 1.25.

Plot $u(x, y)$ using contourf with colorbar (see sample code).

Plot 4 panels, one panel for each time level specified above.