Exercise Set 4: The Finite Difference Method

Exercise 1 (Finite Differences Method)
Define the following concepts:

(a) Consistency;

(b) Stability; and,

(c) Convergence.

Discuss the importance of these concepts, how they relate to each other, and how in general they can be established.

Exercise 2 (Leapfrog Method)
The forward time, center space finite difference approximation to the advection equation $u_t + au_x = 0$ leads to a method which is *unstable*. So let us try something different. By approximating the time derivative by a centered difference, instead of the forward Euler discretization, we get the Leapfrog method,

$$v_j^{n+1} = v_j^{n-1} - \frac{ak}{h}(v_{j+1}^n - v_{j-1}^n) \; . \tag{1}$$

(i) Draw the stencil of (1).

(ii) Show that (1) is second order accurate in both space and time.

(iii) What is an obvious disadvantage of the Leapfrog method compared to the Lax-Friedrichs or Lax-Wendroff methods?

Exercise 3 (Stability of the Lax-Friedrichs Method)
As a possible numerical method for the linear transport equation $u_t + au_x = 0$, consider the Lax-Friedrichs method

$$v_j^{n+1} = \frac{1}{2}(v_{j+1}^n + v_{j-1}^n) - \frac{ak}{2h}(v_{j+1}^n - v_{j-1}^n) \tag{2}$$

Show that this method is stable in the l^∞ norm, provided that k and h satisfy the *CFL condition*

$$\frac{|a|k}{h} \leq 1. \tag{3}$$

Exercise 4 (Unconditionally Stable Method)
We have seen in the previous exercise that the Lax-Friedrichs method is stable provided k and h satisfy a CFL condition. A method which is stable for any k and h is said to be *unconditionally stable.* For $a > 0$, prove that the following backward-time backward-space method

$$v_j^{n+1} = v_j^n - \frac{ak}{h}(v_j^{n+1} - v_{j-1}^{n+1})$$

is unconditionally stable in the l^∞ norm.

Exercise 5 (Matlab Implementation)
This exercise involves some programming. Consider the scalar advection equation $u_t + au_x = 0$ with $a = 1$. Let u be the solution of $u_t + au_x = 0$ in $(-1, 1)$ that satisfies initial condition

$$u(x, 0) = u_0(x), \tag{4}$$

where

$$u_0(x) = \begin{cases} 1 & x < 0 \\ 0 & x > 0 \end{cases} , \tag{5}$$

and boundary conditions

$$u(-1,t) = 1 \qquad u(1,t) = 0 \tag{6}$$

We already know that the exact solution for $0 < t < 1$ is given by $u_0(x - at)$, but how well do numerical methods approximate such a problem? In the following consider the schemes

$$\begin{aligned}
\text{Upwind:} \quad v_j^{n+1} &= v_j^n - \frac{ak}{h}(v_j^n - v_{j-1}^n) \\
\text{Lax-Friedrichs:} \quad v_j^{n+1} &= \frac{1}{2}(v_{j+1}^n + v_{j-1}^n) - \frac{ak}{2h}(v_{j+1}^n - v_{j-1}^n) \\
\text{Lax-Wendroff:} \quad v_j^{n+1} &= v_j^n - \frac{ak}{2h}(v_{j+1}^n - v_{j-1}^n) + \frac{(ak)^2}{2h^2}(v_{j+1}^n - 2v_j^n + v_{j-1}^n) \\
\text{Beam-Warming:} \quad v_j^{n+1} &= v_j^n - \frac{ak}{2h}(3v_j^n - 4v_{j-1}^n + v_{j-2}^n) + \frac{(ak)^2}{2h^2}(v_j^n - 2v_{j-1}^n + v_{j-2}^n) \ .
\end{aligned}$$

For each of the schemes above:

1. Implement the scheme in Matlab to solve $u_t + au_x = 0$, in $(-1, 1)$, with (5), and (6) in the time interval $t \in [0, 0.5]$. For your computations use $h = 0.0025$ and $k/h = 0.5$.
2. Visualize the numerical solution and the exact solution.
3. Comment qualitatively on the solutions' behavior.
4. Write a script to generate a log-log plot of the error as a function of the resolution (either h, or the number of points in the spatial grid), while keeping the ratio $k/h = 0.5$ fixed (why is this important?).
5. Use the plot to deduce the accuracy of the method.
6. Compare your results with the accuracy you would expect on a smooth solution.