MPO 762 – Problem Set 3

1 Quest for a stable time integration scheme

The advection equation $u_t + cu_x = 0$ has been discretized in space using the fifth-order upwind differencing scheme to get the following semi-discrete finite difference equation

$$\frac{\mathrm{d}u_j}{\mathrm{d}t} = -c\frac{-2\,u_{j-3} + 15\,u_{j-2} - 60\,u_{j-1} + 20\,u_j + 30u_{j+1} - 3\,u_{j+2}}{60\Delta x} \tag{1}$$

A user is wondering what type of time-discretization would be useful for this scheme, and what the stability constraints might be. The user decides on a Discrete Fourier decomposition $u_j(t) = \hat{u}(t)e^{ikx_j}$ for the analysis, where $\hat{u}(t)$ is the (complex) wave amplitude and k is the wavenumber.

- Derive the ordinary differential equation governing the time-evolution of the wave amplitude $\hat{u}(t)$. If one were to write this equation in the form of $\frac{\mathrm{d}\hat{u}}{\mathrm{d}t} = \kappa\hat{u}$ what would be the real and imaginary parts of κ?
- Plot the ratio $\frac{\kappa\Delta t}{\mu}$ as a function of $k\Delta x$, and identify in particular, the minima and maxima of the real and imaginary parts.
- Based on the aforementioned analysis, and on the stability diagrams discussed in class, propose some *explicit* scheme for a stable time integration, and identify stability constraints if any.
- Discuss the numerical dissipation, if any, created by this scheme and how it affects the different wavelengths.
- What would happen to the time-integration scheme if c turns out to be negative?

2 Programming FDA to advection equation

Modify the matlab script given to solve the advection equation in a periodic domain using the 3 different spatial differences listed above. Use Runge Kutta of 3rd order for the time stepping. Using this script to solve the following initial condition problems using the 3 different discretizations:

- Smooth initial condition:

$$u(x,0) = \tanh\alpha\left(\cos\pi x + \frac{1}{2}\right) + e^{-\cos^2\pi x} \tag{2}$$

  with $\alpha = 4$ on the interval $-1 \le x \le 1$.
- Inverted parabola

$$u(x,0) = \max(0, 1 - 4x^2) \tag{3}$$

Use the above to discretize the interval $-1 \le x \le 1$ into 64 cells. Set $c = 1$ and integrate until time $t = 20$ by which time the exact solution would have performed 10 circuits of the domain, and would look exactly like the initial condition. Set the Courant number to 1/2 in your experiments. Do the following in your numerical experimentation:

- Plot the solutions on the same graph and compare them to the initial conditions.
- Tabulate the maximum and rms errors in the each of the solutions at the final time.
- Comment on the differences in the numerical solutions, if any, and try to explain the discrepancies.

- Repeat the calculation using 128 cells, do you see an improvement? and is it as expected? Try to explain the sources of the numerical artifacts as best you can.

Please don't write a whole new code for each experiment. Use the template provided on blackboard. Structure your code so that a change in spatial discretization or initial conditions means commenting/uncommenting only a few lines of code.