Computer Problem #5 Part II: Linear Advection Problem
Distributed: November 14, 2017
Due: Tuesday, November 30, 2017

Consider again the 1-D linear convection equation

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0$$

where c is a positive and constant advection speed.

Consider a periodic domain of width 1.0 that is divided into 32 grid intervals with $\Delta x = 1/32$. The initial condition given by

$$u_0(x) = \begin{cases} \{64[(x-1/2)^2 - 1/64]\}^2 & \text{if } 3/8 \le x \le 5/8 \\ 0 & \text{otherwise} \end{cases}$$

Also, let c = 1.0. With these conditions, the feature being advected completes one pass through the domain in a time of 1.0.

a. Using the leapfrog time-differencing scheme and a second-order centered-in-space discretization for the advection term, run the solution to t = 3 using Courant numbers of 0.7, 0.5, and 0.1. Compute the dispersion, dissipation, and diffusion errors using Tackacs' method.

b. Repeat part a, this time adding the Asselin-Robert time filter with a filter coefficient of 0.1. Discuss the results.

c. Perform a linear stability analysis on the leapfrog fourth-order centered-in-space scheme and determine its phase and amplitude errors, and compare them with those of leapfrog second-order-in-space scheme discussed in class.

d. Repeat part a, this time using a fourth-order centered-in-space difference for the advection term, given by

$$\frac{u_i^{n+1} - u_i^{n-1}}{2\Delta t} = -c\left[\frac{4}{3}\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} - \frac{1}{3}\frac{u_{i+2}^n - u_{i-2}^n}{4\Delta x}\right].$$

e. Repeat-part c, this time using the third-order Adams-Bashforth time differencing scheme. Discuss your results and comment on the possible advantages and disadvantages of this scheme.

The periodic boundary condition means that $u(x=0) = u(x=1)$, $u(-m\Delta x) = u(1.0 - m\Delta x)$ and $u(m\Delta x) = u(1.0 + m\Delta x)$, where m is any number, and 1.0 is the length of the physical domain.

To facilitate the implementation of periodic conditions at the lateral boundaries, we usually define extra grid points outside the physical boundary. For the 4th order difference scheme, two extra points are needed outside each boundary, as illustrated below:
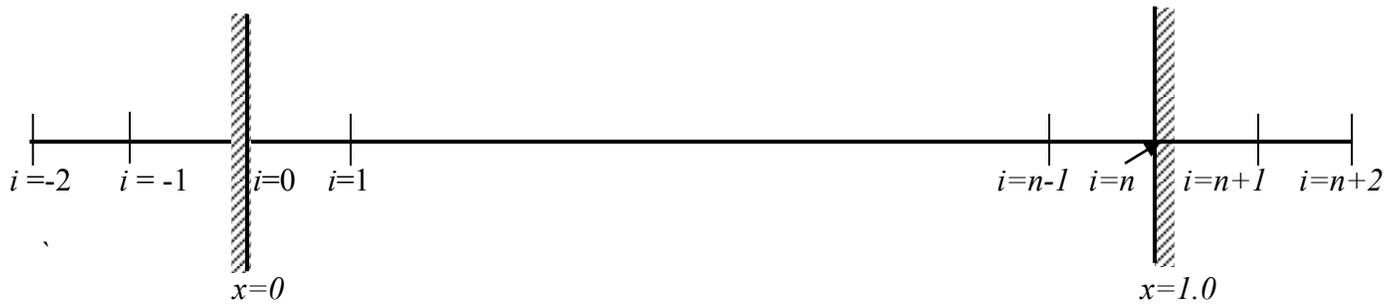
In your Fortran code, it is convenient to define arrays as u(-2:n+2) where n=32 in this case. Then you boundary conditions will be like

u(-1)=u(n-1)
u(-2)=u(n-2)
u(n+1)=u(1)
u(n+2)=u(2)

u at i=0 and n should be solved from the FDE. When the above conditions are used, $u(n) = u(0)$) should be automatically satisfied. For actual implementation, you integrate the finite difference equation forward in time for $i = 0$ to n, and set boundary conditions at $i = -1, -2$ and $n+1, n+2$.