

Computer Problem #5: Linear Convective Transport
Distributed: Tuesday, October 11, 2005
Due: Thursday, November 3, 2005

Consider 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. This equation can be solved numerically using the two-step MacCormack method:

Predictor: $(u_i^{n+1})^* = u_i^n - c\Delta t \frac{u_{i+1}^n - u_i^n}{\Delta x}$

Corrector: $u_i^{n+1} = \frac{1}{2} \left[u_i^n + (u_i^{n+1})^* - c\Delta t \frac{(u_i^{n+1})^* - (u_{i-1}^{n+1})^*}{\Delta x} \right]$

- Derive the modified equation for this two-step scheme and determine the anticipated error type (dispersive or dissipative). When trying to eliminate (), make sure you use of FDE not PDE.
- Use the von Neumann technique to assess the stability of this scheme, and plot the phase and amplitude errors as a function of $k\Delta x$ for several Courant numbers, including a few for which linear stability is violated.
- Write a computer code for this scheme, using as initial conditions the following function:

$$u(x, t=0) = 2 + u_0(x) \left[1 + 0.3 \sin\left(\frac{2\pi x}{9\Delta x}\right) \right] \left[1 + 0.4 \sin\left(\frac{2\pi x}{10\Delta x}\right) \right]$$

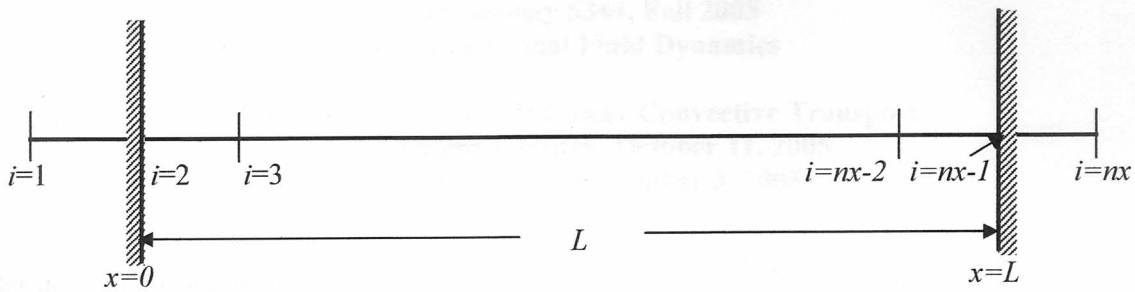
where

$$u_0(x) = \begin{cases} -1 & \text{if } 8 \leq x \leq 28 \\ 1 & \text{if } 28 < x \leq 39 \\ 0 & \text{otherwise} \end{cases}$$

with $\Delta x = 1.0$ in a periodic domain of length 50. This somewhat unconventional initial condition provides a stringent test of advective schemes because it contains sharp gradients and other spatial irregularities.

The periodic boundary condition means that $u(x=0) = u(x=L)$, $u(x=-l) = u(x=(L-l))$ and $u(x=l) = u(x=(L+l))$, where L is the length of the physical domain.

To facilitate the implementation of periodic conditions at the lateral boundaries, we usually define an extra grid point outside each physical boundary, so for a physical domain of length L , we need $nx = L/\Delta x + 3$ number of grid points, as illustrated below:



In the discrete form, the boundary conditions are: $u(1) = u(nx-2)$ and $u(nx) = u(3)$. When these conditions are used, $u(2) = u(nx-1)$ should be automatically satisfied (note a typo with the previous version of this formula). For actual implementation, you integrate the finite difference equation forward in time for $i = 2$ to $nx-1$, and set boundary conditions at $i = 1$ and nx .

Assume $c = 1.0$, run your code with Courant numbers of 0.10, 0.25, 0.50, and 1.00, show the plots for the numerical solutions at time $t = 50.0, 100.0, 200.0$, and 400.0, and discuss your results in light of what you know about this scheme based on the amplitude and phase error analyses. Is there anything special about solution with Courant number = 1?

1. For Courant numbers of 0.10, 0.25, 0.50, and 1.00 and for $t = 50.0, 100.0, 200.0$, and 400.0, use Takacs' method to compute the amplitude and phase errors of numerical solutions relative to the exact solution (which is simply the initial condition shifted to the right by the number of time steps multiplied by the Courant number), and compare them with the theoretical predictions made in part b. Is the predominant error type similar to that anticipated from the modified equation? Comment on your findings.