The Non-Viscous Burger's Equation

This module allows you to do a numerical study of the leap frog and Lax-Wendroff numerical techniques applied to Burger's equation in non-conservation form

$$u_t + uu_x = 0, -L < x < L, 0 < t < T$$

with initial condition

$$u(x,0) = q \arctan(sx) + r$$

and in conservation form

$$u_t + (0.5u^2)_x = 0, -L < x < L, 0 < t < T$$

with initial condition

$$u(x,0) = q \arctan(sx) + r.$$

In the codes comprising this module you are asked to input L, a_left (the value for $u(x,0)$ at $-\infty$) and a_right (the value for $u(x,0)$ at ∞). You also input the time step, dt for t and the grid size, dx for x.

There are Maple routines, Matlab routines and Fortran 90 codes for running the numerical algorithms. The Maple routines are

- burgerd-lf.mws (non-conservation form using the leap frog finite difference scheme)
- burgerd-lw.mws (non-conservation form using the Lax Wendroff finite difference scheme)
- burgerc-lf.mws (conservation form using the leap frog finite difference scheme)
- burgerc-lw.mws (conservation form using the Lax Wendroff finite difference scheme)

The Matlab routines are

- burgerdlf.m (non-conservation form using the leap frog finite difference scheme)

- burgerdlw.m (non-conservation form using the Lax Wendroff finite difference scheme)
- burgerdlf.m (conservation form using the leap frog finite difference scheme)
- burgerdlw.m (conservation form using the Lax Wendroff finite difference scheme)

The Fortran 90 codes are

- burgerd-lf.f90 (non-conservation form using the leap frog finite difference scheme)
- burgerd-lw.f90 (non-conservation form using the Lax Wendroff finite difference scheme)
- burgerc-lf.f90 (conservation form using the leap frog finite difference scheme)
- burgerc-lw.f90 (conservation form using the Lax Wendroff finite difference scheme)

Choose one or more of the above routines and run each code using a_left = -1, a_right = 1, s = 1, L = 10, T = 50*dt, dt = 0.03125 and dx = 0.25. You should convince yourself using graphics that the results you are getting are reasonable. The Maple and Matlab routines will do the graphics for you. The Fortran 90 codes output data sets. These data sets have the form name00010. The 00010 represents the 10th time step. For the Fortran 90 codes you have to load the data sets into a graphics visualizer.

Now make L larger and see what happens. Make s larger and see what happens. Then reset s to 1 and make a_left and a_right different. For example, a_left = 1, a_right = -1. What happens? Reset a_left = -1 and a_right = 1 and make dt larger. What happens? Now make dt smaller and see what happens. You should ask how these tests relates to the stability of the numerical algorithm.

Now experiment by changing a_left,a_right, dt and dx. What results do you find? Can you explain these physically and numerically?

Take the leap frog and Lax-Wendroff codes in one of the software packages and include diffusion to give the viscous Burger's equation. Study what effect this term has in the equations by varying ν. That is, consider, the leap frog and Lax-Wendroff numerical solutions to the PDE

$$u_t + uu_x - \nu u_{xx} = 0 \; ; \; -a < x < a, 0 < t < T$$

with initial condition

$$u(x, 0) = q \arctan(sx) + r$$

and the corresponding PDE in conservation form

$$u_t + (0.5u^2)_x - \nu u_{xx} = 0 \; ; \; -a < x < a, 0 < t < T$$

with initial condition

$$u(x, 0) = q \arctan(sx) + r$$

and compare the results for various inputs.