PHY 688: The Application of Simulation in Astrophysics
Homework #4
Due: Wed. Apr. 19, 2006

1. In this assignment, we will code up Godunov's method in one-dimension for the Euler equations.

Recall in class, we went through the elements that make up a hydrodynamics program (see Lecture 22 at http://www.astro.sunysb.edu/mzingale/phy688/lectures.html). It may be useful to use the `advect-godunov.f90` code from that lecture as your guide. The basic elements of Godunov's method are

- piecewise constant reconstruction of the data in each zone
- solving the Riemann problem at each interface
- updating the cell-averages using fluxes computed with the Riemann solution at the interfaces.

We will apply this method to the Euler equations. In conservative form, the Euler equations are:

$$U_t + [F(U)]_x = 0 \tag{1}$$

where

$$U = \begin{pmatrix} \rho \\ \rho u \\ \rho E \end{pmatrix} \qquad F(U) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho E u + p u \end{pmatrix} \tag{2}$$

In your code, you will need to allocate space for all three conserved variables in all zones. You will also need a variable to hold the left interface states $\{\rho_l, (\rho u)_l, (\rho E)_l\}$ and the right interface states $\{\rho_r, (\rho u)_r, (\rho E)_r\}$.

The flow of your program should follow that of `advect-godunov.f90`. Now, your `init` routine will need to initialize all the conserved variables. Likewise, `fillBC` will need to fill the guardcells for all three variables. You should use outflow boundary conditions here (i.e. give the variables a zero-gradient at the boundaries).

Our `timestep` routine needs to limit the timestep such that information traveling at the maximum wave speed does not cross more than one zone. A common way to do this in practice is as:

$$\Delta t = C \min_i \left\{ \frac{\Delta x}{|u_i| + c_i} \right\} \tag{3}$$

Here, c_i is the sound speed in zone i. You will need to compute this using the conserved variables and the equation of state. C is the Courant number, you should use $C = 0.8$ for all calculations.

The `states` routine needs to be modified to compute the interface states for all conserved variables. Here, just use piecewise constant reconstruction, as we did in class for the advection equation. A `riemann` routine for the Euler equations (based on the two-shock approximation) can be downloaded from http://www.astro.sunysb.edu/mzingale/phy688/homework.html as either `riemann.f90` or `riemann.c`. This routine will return three fluxes (one for each conserved variable). To complete the update, modify your `update` routine to conservatively difference each unknown $\{\rho, (\rho u), (\rho E)\}$ using its fluxes through the interfaces. Again, this is analogous to what we did for the linear advection equation in `advect-godunov.f90`.

Finally, modify the output program to print out $\{x_i, \rho_i, u_i, p_i\}$ for all zones. The primitive variables provide a more intuitive set of variables to understand the results.

Provide your complete code for this assignment both on paper and by e-mail.

Test Problems

We will test this code by running several shock tube problems. The exact results for these tests can be downloaded from http://www.astro.sunysb.edu/mzingale/phy688/homework.html as `test1-exact.out`, `test2-exact.out`, `test3-exact.out`, `test4-exact.out`. Run each test at 64, 128, and 256 zones, and plot the results on top of the exact solution.

Our initial conditions are specified in terms of the primitive varaibles, $W = \{\rho, u, p\}^T$. You should use $\gamma = 1.4$ in the equation of state. All of our tests are run on a domain $[0, 1]$, with a single jump in the state:

$$W(x, t = 0) = \left\{ \begin{array}{ll} W_l & x < 1/2 \\ W_r & x \geq 1/2 \end{array} \right. \tag{4}$$

For each of the tests below, run you Godunov solver at three resolutions and plot your solutions overtop the exact solutions you downloaded above.

test 1 (Sod's problem):

$$W_l = \left\{ \begin{array}{c} 1.0 \\ 0.0 \\ 1.0 \end{array} \right. \qquad W_r = \left\{ \begin{array}{c} 0.125 \\ 0.0 \\ 0.1 \end{array} \right. \tag{5}$$

$$t_{\mathrm{max}} = 0.25 \tag{6}$$

This is called *Sod's problem.* The solution consists of a shock, a contact discontinuity, and a rarefaction. It is a standard test for hydrodynamics codes, since all three hydrodynamic waves are present.

test 2 (double rarefaction):

$$W_l = \left\{ \begin{array}{c} 1.0 \\ -2.0 \\ 0.4 \end{array} \right. \qquad W_r = \left\{ \begin{array}{c} 1.0 \\ 2.0 \\ 0.4 \end{array} \right. \tag{7}$$

$$t_{\mathrm{max}} = 0.15 \tag{8}$$

This configuration results in two rarefactions moving away from the center and a trivial contact discontinuity at the center. Between the two rarefactions, the pressure tends to zero (creating a vacuum region). For this reason, this is a very difficult problem for hydro codes.

test 3 (strong shock):

$$W_l = \begin{cases} 1.0 \\ 0.0 \\ 1000.0 \end{cases} \qquad W_r = \begin{cases} 1.0 \\ 0.0 \\ 0.01 \end{cases} \tag{9}$$

$$t_{\max} = 0.012 \tag{10}$$

The solution to this initial state consists of a strong right moving shock, a contact, and a rarefaction. This strong shock is a very demanding test of a Riemann solver.

test 4 (stationary shock):
Now, we will consider the problem of a slow moving shock. Consider the following initial conditions

$$W_l = \begin{cases} 5.6698 \\ -1.4701 \\ 100.0 \end{cases} \qquad W_r = \begin{cases} 1.0 \\ -10.5 \\ 1.0 \end{cases} \tag{11}$$

$$t_{\max} = 1.0 \tag{12}$$

These initial conditions result in a single shock moving to the right, but the velocity ahead of the shock was chosen so the shock is essentially stationary.

Describe the flow behind the shock.