

**Radiative Gas Dynamics**  
**Problem Set 4: A Minimal 1-d Hydro Code**  
**Due Thursday, Feb. 15**

The goal of this problem set is to write a minimal 1-d hydrodynamics code and apply it to some simple test problems. The code is minimal in several senses: (1) it is 1-d plane-parallel, (2) it does not incorporate gravity, (3) it does not incorporate viscosity and therefore cannot handle shocks, (4) since it does not incorporate viscosity, it does not need to evolve the gas temperature but just uses a polytropic or isothermal equation of state. The range of problems this code can be applied to is quite limited because of these restrictions, but it is relatively easy to add at least some of these other processes.

The differential equations to be integrated are the continuity and momentum equations:

$$\frac{\partial \rho}{\partial t} = -u \frac{\partial \rho}{\partial x} - \rho \frac{\partial u}{\partial x} \quad (1)$$

$$\frac{\partial u}{\partial t} = -u \frac{\partial u}{\partial x} - \rho^{-1} \frac{\partial P}{\partial x}. \quad (2)$$

In this code, the density  $\rho$  and velocity  $u$  should be represented as 1-dimensional arrays. The initial conditions will be small perturbations about a uniform density, and we'll adopt units in which this unperturbed density is  $\rho = 1$ . We'll also adopt units in which the length of the computational box is  $L = 1$ . Finally, we'll adopt a time unit equal to the length of the box divided by the isothermal sound speed at some fiducial temperature  $T_0$ , implying that

$$L/a_{T_0} = 1, \quad a_{T_0} = \left( \frac{kT_0}{m} \right)^{1/2}. \quad (3)$$

Where does the time unit enter our equations? It determines the equation of state. Specifically, since  $L = 1$ , equation (3) implies that in this system of units  $a_{T_0} = 1$ , and therefore that

$$P = \frac{\rho}{m} kT = a_{T_0}^2 \rho \frac{T}{T_0} = T_1 \rho^\gamma, \quad (4)$$

where  $\gamma$  is the adiabatic index and  $T_1$  is the temperature of gas with  $\rho = 1$  in units where the temperature  $T_0 = 1$ .

The dimension  $N$  of the  $\rho$  and  $u$  arrays determines the spatial resolution of the calculation, since structures on scale  $\sim L/N$  or smaller cannot be represented. The fundamental numerical parameters of the calculation are therefore  $N$  and the timestep  $\Delta t$ . As a default for this problem set, we'll adopt  $N = 200$ , and we'll experiment with  $\Delta t$ .

To estimate spatial derivatives, we will use centered finite differences, e.g.,

$$u'_i = \frac{u_{i+1} - u_{i-1}}{2\Delta x} = \frac{N}{2}(u_{i+1} - u_{i-1}), \quad (5)$$

where  $u'_i$  is the estimate of  $\frac{\partial u}{\partial x}$  in cell  $i$ ,  $u_{i-1}$  and  $u_{i+1}$  are the values of  $u$  in cells  $i-1$  and  $i+1$ , and the second equality uses the fact that the spacing between cells is  $\Delta x = L/N = 1/N$ . With the equation of state (4), the estimate of the pressure gradient is

$$P'_i = \frac{N}{2} T_1 (\rho_{i+1}^\gamma - \rho_{i-1}^\gamma). \quad (6)$$

There is one additional wrinkle here – we need to decide what the boundary conditions are at the edge of our box. In general, the appropriate boundary condition depends on the problem one is trying to solve. Here we will adopt the convenient and relatively simple case of a periodic boundary condition – material that leaves one side of the box reenters at the other side. This means that in estimating the spatial derivative in the first or last cell, one must make use of the cell on the other side of the box, e.g.,

$$u'_1 = \frac{N}{2}(u_2 - u_N), \quad u'_N = \frac{N}{2}(u_1 - u_{N-1}). \quad (7)$$

My suggestion for implementing this is to have your array indices run from 0 to  $N+1$ , with elements 1 to  $N$  being the “meaningful” ones, but each time you update the arrays at a new timestep, you copy  $\rho[N]$  to  $\rho[0]$  and  $\rho[1]$  to  $\rho[N+1]$  (and the same for  $u$ ). Then when you need to evaluate derivatives, you just loop over the indices 1 to  $N$  and the periodic values will get used when they are needed. (If you are programming in fortran, you can either define your arrays so that they start at index 0 or go from 1 to  $N+2$  instead of 0 to  $N+1$ .)

For a time integration scheme, use a midpoint method similar to the one you used (for space integration) in the isothermal sphere integration. Specifically, to advance the values of  $\rho_i$  and  $u_i$  from time step  $j$  to step  $j+1$ , the equations are

$$\rho_{i,j+1} = \rho_{i,j} + \dot{\rho}_{i,j+\frac{1}{2}} \Delta t \quad (8)$$

$$u_{i,j+1} = u_{i,j} + \dot{u}_{i,j+\frac{1}{2}} \Delta t \quad (9)$$

where (from equations 1 and 2)

$$\dot{\rho}_{i,j+\frac{1}{2}} = -u_{i,j+\frac{1}{2}} \rho'_{i,j+\frac{1}{2}} - \rho_{i,j+\frac{1}{2}} u'_{i,j+\frac{1}{2}} \quad (10)$$

$$\dot{u}_{i,j+\frac{1}{2}} = -u_{i,j+\frac{1}{2}} u'_{i,j+\frac{1}{2}} - \rho_{i,j+\frac{1}{2}}^{-1} P'_{i,j+\frac{1}{2}} \quad (11)$$

and the values at the midstep ( $j + \frac{1}{2}$ ) are estimated by extrapolation from the beginning of the step,

$$\rho_{i,j+\frac{1}{2}} = \rho_{i,j} + \dot{\rho}_{i,j} \frac{\Delta t}{2} \quad (12)$$

$$u_{i,j+\frac{1}{2}} = u_{i,j} + \dot{u}_{i,j} \frac{\Delta t}{2}. \quad (13)$$

Note that although I have used a second index,  $j$ , to denote the timestep, you do not need two-dimensional arrays because you do not need to save the values of  $\rho$  and  $u$  at previous times. My

implementation of this program has two sets of  $\rho$  and  $u$  arrays, one set that has values on the integer values of  $j$  and one set that has values at the midsteps.

Write a minimal hydro program using this integration scheme. I recommend using double precision (real\*8). This program needn't be very complicated. My version has 140 lines, but most of that is comments, declarations, inputs, and so forth – the integration itself is only 35 lines of code, and could be shorter.

(a) For the first run, use  $\gamma = 1$  (an isothermal equation of state), an initial density field

$$\rho(x) = 1 + 0.02 \exp \left[ -\frac{(x - 0.5)^2}{2 \times 0.05^2} \right],$$

initial velocity field  $u(x) = 0.0$ , and temperature  $T_1 = 1$ . Use a timestep  $\Delta t = 10^{-3}$ .

Plot the density field  $\rho(x)$  at  $t = 0.1, 0.3, 0.5, 0.8, 1.0, 1.5,$  and  $2.0$ .

Write a brief interpretation of your result.

(b) Repeat the calculation with temperature  $T_1 = 4$ . How is the result different, and why? (You don't need to attach more plots, just describe the difference in words and explain it.)

(c) Repeat the calculation with  $T_1 = 1$  but  $\gamma = 5/3$ . How is the result different and why?

(d) Now take initial conditions  $\rho(x) = 1.0$  and

$$u(x) = 0.02 \exp \left[ -\frac{(x - 0.5)^2}{2 \times 0.05^2} \right],$$

with  $\gamma = 1, T_1 = 1$ .

Plot the density field at  $t = 0.0, 0.1, 0.3, 0.5, 0.8, 1.0, 1.5, 2.0$ .

Write a brief interpretation of the result.

Your initial velocity perturbation has a maximum amplitude of 0.02 in some unit. What (physically) is the velocity unit in your calculation?

(e) Repeat the calculation of part (a) but with a perturbation amplitude of 0.1 instead of 0.02, and only plot the results to  $t = 1.0$ . How are they different from those of (a) and why?

(f) Repeat the calculation of part (a), with the perturbation amplitude returned to 0.02, but with  $\Delta t = 2 \times 10^{-3}$ , and only going to  $t = 1.0$ . How do the results differ from (a)?

Repeat the calculation with  $\Delta t = 4 \times 10^{-3}$ . Now what happens?

(g) From part (e), you found that there is a maximum  $\Delta t$  for which you can get stable results. Why do you think this is? Suppose that you had used  $T_1 = 4$  instead of  $T_1 = 1$ . Would the maximum  $\Delta t$  be smaller, larger, or the same? Suppose that you had kept  $T_1 = 1$  but used 400 grid cells instead of 200. Would the maximum  $\Delta t$  be smaller, larger, or the same?