AMS260 Final Term Coding Project - Due 6pm, March 24, 2017

1D Finite Volume Shock Capturing Code for the Euler Equations

---

In this project we implement a finite volume conservative code to solve the compressible 1D Euler equations,

$$\mathbf{U}_t + \Big(\mathbf{F}(\mathbf{U})\Big)_x = 0, \tag{1}$$

where the conservative variables $\mathbf{U}$ and the associated flux function $\mathbf{F}(\mathbf{U})$ are given by

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix}, \text{ and } \mathbf{F} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ u(E+p) \end{pmatrix}. \tag{2}$$

Here, we follow the conventional way to denote flow variables of the Euler equations. They are the mass density ρ, the x-momentum $m = \rho u$, and the total energy per unit mass as a sum of the kinetic energy $\rho u^2/2$ and the internal energy ρe,

$$E = \rho\Big(\frac{u^2}{2} + e\Big), \tag{3}$$

with the specific internal energy e given by a simple ideal gas law equation of state (EoS),

$$e = e(\rho, p) = \frac{p}{(\gamma - 1)\rho}. \tag{4}$$

We assume the ratio of specific heats γ constant and use $\gamma = 1.4$ for our project. You might want to use different values of γ, for instance, $\gamma = 5/3 = 1.666667$ to see what differences you get.

The goals in the project are the following:

1. Complete the first-order Godunov (FOG) template code,
2. Extend the FOG code to the second-order piecewise linear method (PLM),
3. Implement three different slope limiters, minmod, van Leer, and MC for PLM,
4. Implement three different Riemann solvers of HLL, HLLC and Roe,
5. Run benchmarked problems using FOG and PLM, respectively combined with HLL, HLLC and Roe solvers (6 cases), and
6. Analyze your results by conducting comparison studies of the six cases (FOG-HLL, FOG-HLLC, FOG-Roe, PLM-HLL, PLM-HLLC, and PLM-Roe).

You're given one example problem, Sod's shock tube, as a template. Using this template code, you should be able to check and see if all your implementations (the six combinations) are correctly working on the Sod problem.

Once you see all your implementations are correct on the Sod's shock tube problem, you're going to setup and run three new test problems.

1. Grid Discretization

We adopt our 1D grid as before, following the cell-centered (rather than cell interface-centered) notation for discrete cells x_i and the conventional temporal discretization t^n:

$$x_i = (i - \frac{1}{2})\Delta x, \tag{5}$$

$$t^n = n\Delta t. \tag{6}$$

Then the cell interface-centered grid points are written using the 'half-integer' indices:

$$x_{i+\frac{1}{2}} = x_i + \frac{\Delta x}{2}. \tag{7}$$

In this project, we take the number of guardcells $N_{ngc} = 2$ on each side of the domain, resulting the following grid configuration with N_x numbers of interior grid resolutions:

- Two first guardcells on the left: $x_i, 1 \le i \le 2$,
- Interior points: $x_i, N_{ngc} + 1 \le i \le N_{ngc} + N_x$,
- Two last guardcells on the right: $x_i, N_{ngc} + N_x + 1 \le i \le 2N_{ngc} + N_x$.

2. Three Test Problems plus one bonus problem

2.1. Example: Sod's Shock Tube Problem

The Sod problem (Sod 1978) is a one-dimensional flow discontinuity problem that provides a good test of a compressible code's ability to capture shocks and contact discontinuities with a small number of cells and to produce the correct profile in a rarefaction. It also tests a code's ability to correctly satisfy the Rankine-Hugoniot shock jump conditions.

We construct the initial conditions for the Sod problem on the computational domain $[0, 1]$ by establishing a single jump discontinuity. The fluid is initially at rest on either side of the interface, and the density and pressure jumps are chosen so that all three types of nonlinear, hydrodynamic waves (shock, contact, and rarefaction) develop. To the "left" and "right" of the interface we have

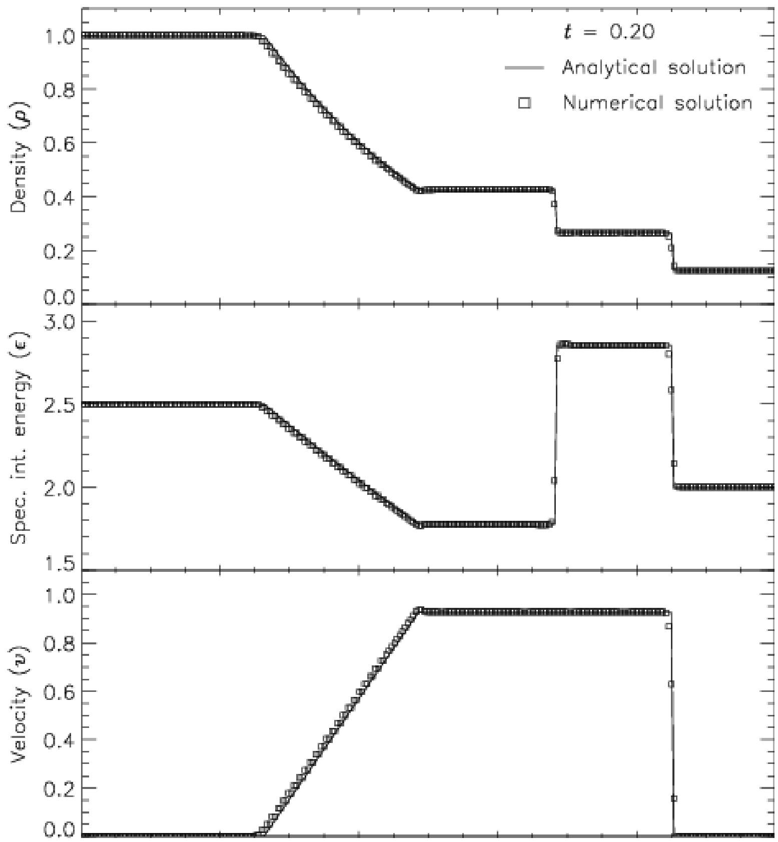

Figure 1. Comparison of numerical and analytical solutions to the Sod problem using the FLASH code. The simulated result is sampled at $t = 0.2$.

$$\mathbf{V}(x,0) = \begin{cases} \begin{pmatrix} \rho \\ u \\ p \end{pmatrix}_L = \begin{pmatrix} 1.0 \\ 0.0 \\ 1.0 \end{pmatrix} & \text{if } x \leq 0.5, \\ \\ \begin{pmatrix} \rho \\ u \\ p \end{pmatrix}_R = \begin{pmatrix} 0.125 \\ 0.0 \\ 0.1 \end{pmatrix} & \text{if } x > 0.5. \end{cases} \tag{8}$$

The ratio of specific heats γ is chosen to be 1.4 on both sides of the interface. The outflow boundary condition is used.

2.2. Rarefaction Wave

This problem does not contain any jump discontinuities and is smooth, hence it is a good test problem for convergence test. The initial condition on the computational domain $[0, 1]$ is given by:

$$\mathbf{V}(x,0) = \begin{cases} \begin{pmatrix} \rho \\ u \\ p \end{pmatrix}_L = \begin{pmatrix} 1.0 \\ -2.0 \\ 0.4 \end{pmatrix} & \text{if } x \le 0.5, \\ \begin{pmatrix} \rho \\ u \\ p \end{pmatrix}_R = \begin{pmatrix} 1.0 \\ 2.0 \\ 0.4 \end{pmatrix} & \text{if } x > 0.5. \end{cases} \tag{9}$$

The ratio of specific heats γ is chosen to be 1.4 on both sides of the interface. Please use $t_{\max} = 0.15$. The outflow boundary condition is applied to this problem.

2.3. Interacting Blast-Wave: Blast2

This Blast2 problem was originally used by Woodward and Colella (1984) to compare the performance of several different hydrodynamical methods on problems involving strong shocks and narrow features. It has no analytical solution (except at very early times), but since it is one-dimensional, it is easy to produce a converged solution by running the code with a very large number of cells, permitting a reference solution to compare with.

Reflecting boundary conditions are used, where the velocity u is negated in the guard-cell regions in a symmetric way, i.e., assuming $N_{ngc} = 2$,

$$u_i = -u_{k-i}, i = 1, 2. \tag{10}$$

on the left boundary with $k = 2N_{ngc} + 1$. Similarly on the right boundary we have

$$u_i = -u_{k-i}, i = N_x + N_{ngc} + 1, N_x + 2N_{ngc} \tag{11}$$

where $k = N_{ngc} + 2N_x + 1$.
The other primitive variables, density and pressure, are mirrored in the guardcell regions,

$$\rho_i = \rho_{k-i}, p_i = p_{k-i}, i = 1, 2. \tag{12}$$

on the left boundary with $k = 2N_{ngc} + 1$. Similarly on the right boundary we have

$$\rho_i = \rho_{k-i}, p_i = p_{k-i}, i = N_x + N_{ngc} + 1, N_x + 2N_{ngc} \tag{13}$$

where $k = N_{ngc} + 2N_x + 1$.

The initial conditions consist of two parallel, planar flow discontinuities on the computational domain $[0, 1]$. The density is unity and the velocity is initially zero everywhere. The pressure is large at the left and right and small in the center

$$p_{\rm L} \;=\; 1000, \qquad p_{\rm M} \;=\; 0.01, \qquad p_{\rm R} \;=\; 100\ . \tag{14}$$

$$\mathbf{V}(x,0) = \begin{cases} \begin{pmatrix} \rho \\ u \\ p \end{pmatrix}_L = \begin{pmatrix} 1.0 \\ 0.0 \\ 1000.0 \end{pmatrix} & \text{if } x \le 0.1, \\ \begin{pmatrix} \rho \\ u \\ p \end{pmatrix}_M = \begin{pmatrix} 1.0 \\ 0.0 \\ 0.01 \end{pmatrix} & \text{if } 0.1 < x \le 0.9, \\ \begin{pmatrix} \rho \\ u \\ p \end{pmatrix}_R = \begin{pmatrix} 1.0 \\ 0.0 \\ 100.0 \end{pmatrix} & \text{if } x > 0.9. \end{cases} \tag{15}$$

Please use $\gamma = 1.4$ and the final time $t_{\max} = 0.038$.

2.4. The Shu-Osher Problem

The problem description is given in Chapter 10 of the lecture note. Please also see computed results and their comparisons in Fig. 5 therein.

This problem requires to use a special boundary condition that keeps the values of the primitive variables in the guardcells unchanged from the initial conditions. That is, using the computational domain of $[-4.5, 4.5]$,

$$\mathbf{V}(x,t) = \begin{cases} \begin{pmatrix} \rho \\ u \\ p \end{pmatrix}_L = \begin{pmatrix} 3.857143 \\ 2.629369 \\ 10.33333 \end{pmatrix} & \text{if } x < -4.0, \\ \begin{pmatrix} \rho \\ u \\ p \end{pmatrix}_R = \begin{pmatrix} 1 + a_\rho \sin(f_\rho x) \\ 0.0 \\ 1.0 \end{pmatrix} & \text{if } x > -4.0, \end{cases} \tag{16}$$

where a_ρ is the amplitude and f_ρ is the frequency of the density perturbations, for which we take $a_\rho = 0.2$ and $f_\rho = 5.0$. The ideal equation of state is used with γ set to 1.4. The location of the initial discontinuity is at $x_s = -4.0$. Please use the final time $t_{\max} = 1.8$.

3. Project Tasks

Write a scientific report using LaTeX no more than 15 pages with font size 11. Your report should have three parts: introduction, main code results, and conclusion. Submit your report and source code to your Git repository. Report any differences you may observe and explain why.

- Coding:
    1. Study the template code.
    2. Please complete the FOG code and extend it to implement PLM, and three Riemann solvers, HLL, HLLC and Roe.

3. Implement three different slope limiters: minmod, van Leer's, and MC for PLM.
4. Please setup the above four problems: Sod, Rarefaction, Blast2, and the Shu-Osher problem.

- Code Results: Run the four problems to conduct the following comparison tests.

1. [**FOG vs. PLM**] Run the Sod shock tube problem on $N_x = 128$ using HLLC. Plot primitive variables (density ρ, velocity u, and pressure p) at $t = 0.2$ for each FOG and PLM+minmod.
2. [**Riemann Solvers**] Run the rarefaction problem on $N_x = 128$ using PLM with minmod. Plot primitive variables (density ρ, velocity u, and pressure p) at $t = 0.15$ for each HLL, HLLC, and Roe.
3. [**Slope Limiters**] Run the Blast2 problem on $N_x = 128$ PLM with HLLC. Plot primitive variables (density ρ, velocity u, and pressure p) at $t = 0.038$ for each minmod, van Leer's, and MC slope limiters. In addition, compare these results with FOG+HLLC.
4. [**Grid Resolutions**] Run the Shu-Osher problem using PLM+MC+Roe. on $N_x = 32, 64, 128$, and 256. Plot primitive variables (density ρ, velocity u, and pressure p) at $t = 1.8$ for each grid resolution. In addition, compare these results with FOG+Roe.
5. [**CFL**] Run the Shu-Osher problem using PLM+minmod+HLLC on $N_x = 128$ using CFL $= 0.2, 0.4, 0.6, 0.8, 1.0$, and 1.4. Plot primitive variables (density ρ, velocity u, and pressure p) at $t = 1.8$ for each CFL number.
6. [**Extra Bonus Problem**] Extend the 1D Euler code to one of the following options – Please let me know if you wish to do this optional problem:
   - 1D MHD to solve the 1D Brio-Wu problem (see papers by Brio and Wu; Roe and Balsara), or
   - ADER method for high-order in both space and time (see Chapters 19 and 20 in Toro; and papers by Titarev and Toro)