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,

$$ U_t + \left( F(U) \right)_x = 0, \quad (1) $$

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

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

Here, we follow the conventional way to denote flow variables of the Euler equations. They are the mass density $\rho$, 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 $\rho e$,

$$ E = \rho \left( \frac{u^2}{2} + e \right), \quad (3) $$

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

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

We assume the ratio of specific heats $\gamma$ constant and use $\gamma = 1.4$ for our project. You might want to use different values of $\gamma$, 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 two different Riemann solvers of HLL and Roe,
4. Run benchmarked problems using FOG and PLM, respectively combined with HLL and Roe solvers,
5. Analyze your results by conducting comparison studies of FOG-HLL, FOG-Roe, PLM-HLL, and PLM-Roe, and
6. Analyze your results by performing grid convergence studies using FOG-HLL, FOG-Roe, PLM-HLL, 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 four combinations of FOG-HLL, FOG-Roe, PLM-HLL, and PLM-Roe) 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 $t^n$:

\begin{align}
  x_i &= (i - \frac{1}{2})\Delta x, \\
  t^n &= n\Delta t.
\end{align}

(5)\hspace{1cm}(6)

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

\begin{equation}
  x_{i+\frac{1}{2}} = x_i + \frac{\Delta x}{2}.
\end{equation}

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

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 $t = 0.2$.

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

The ratio of specific heats $\gamma$ 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:
\[
\mathbf{V}(x, 0) = \begin{cases} 
\left( \begin{array}{c} \rho \\ u \\ p \end{array} \right)_L = \left( \begin{array}{c} 1.0 \\ -2.0 \\ 0.4 \end{array} \right) & \text{if } x \leq 0.5, \\
\left( \begin{array}{c} \rho \\ u \\ p \end{array} \right)_R = \left( \begin{array}{c} 1.0 \\ 2.0 \\ 0.4 \end{array} \right) & \text{if } x > 0.5.
\end{cases}
\] (9)

The ratio of specific heats \( \gamma \) is chosen to be 1.4 on both sides of the interface. Please use \( t_{\text{max}} = 0.15 \). 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 \( N_{\text{ngc}} = 2 \),

\[
u_i = -u_{k-i}, \quad i = 1, 2.
\] (10)
on the left boundary with \( k = 2N_{\text{ngc}} + 1 \). Similarly on the right boundary we have

\[
u_i = -u_{k-i}, \quad i = N_x + N_{\text{ngc}} + 1, N_x + 2N_{\text{ngc}}
\] (11)
where \( k = N_{\text{ngc}} + 2N_x + 1 \).

The other primitive variables, density and pressure, are mirrored in the guardcell regions,

\[
\rho_i = \rho_{k-i}, \quad p_i = p_{k-i}, \quad i = 1, 2.
\] (12)
on the left boundary with \( k = 2N_{\text{ngc}} + 1 \). Similarly on the right boundary we have

\[
\rho_i = \rho_{k-i}, \quad p_i = p_{k-i}, \quad i = N_x + N_{\text{ngc}} + 1, N_x + 2N_{\text{ngc}}
\] (13)
where \( k = N_{\text{ngc}} + 2N_x + 1 \).

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

\[
p_L = 1000, \quad p_M = 0.01, \quad p_R = 100.
\] (14)
\[ V(x,0) = \begin{cases} 
\begin{pmatrix} \rho \\ u \\ p \end{pmatrix} & = \begin{pmatrix} 1.0 \\ 0.0 \\ 1000.0 \end{pmatrix} & \text{if } x \leq 0.1, \\
\begin{pmatrix} \rho \\ u \\ p \end{pmatrix} & = \begin{pmatrix} 1.0 \\ 0.0 \\ 0.01 \end{pmatrix} & \text{if } 0.1 < x \leq 0.9, \\
\begin{pmatrix} \rho \\ u \\ p \end{pmatrix} & = \begin{pmatrix} 1.0 \\ 0.0 \\ 100.0 \end{pmatrix} & \text{if } x > 0.9. 
\end{cases} \] (15)

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

### 2.4. Bonus Problem: 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 \([-4.5, 4.5]\),

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

where \( a_\rho \) is the amplitude and \( f_\rho \) 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 \( \gamma \) set to 1.4. The location of the initial discontinuity is at \( x_s = -4.0 \). Please use the final time \( t_{\text{max}} = 1.8 \).

### 3. Project Tasks

1. Please extend the FOG code and implement PLM, and two Riemann solvers, HLL and Roe.

2. Please setup the three problems (Sod, Rarefaction, Blast2) and one bonus problem (the Shu-Osher problem) using four combinations up to each given \( t = t_{\text{max}} \).

3. On each problem, you’re going to use a base grid resolution of \( N_x = 128 \). Please plot your primitive variables (density \( \rho \), velocity \( u \), and pressure \( p \)) at \( t = t_{\text{max}} \) using FOG+HLL, FOG+Roe, PLM+HLL, and PLM+Roe.
4. For the Blast2 and the rarefaction problems, please perform the grid resolution study using $N_x = 32, 64, 128, 256$.

   (a) For Blast2, please plot density $\rho$ on these four resolutions using PLM+Roe. Briefly discuss what you observe.

   (b) For the rarefaction wave, please conduct a convergence study in $l^1$ norm on the four resolution. Use density $\rho$ for your choice of variable to evaluate the error norm. When you do this, you would need a reference (or exact) solution in order to compute the error norm. For this, when you have no true analytic solution you can use a high-resolution solution and treat it as a reference solution. Please use $N_x = 1024$ for your reference solution using PLM+HLL. Obtain two convergence curves using (i) FOG+HLL, and (ii) PLM+HLL. What do you see?

5. Bonus: In the above runs, you can fix your slope limiter to be one of the three we studied, e.g., minmod. What happens if you use a different limiter, say, van Leer’s or MC? Do you see any better or worse solution behaviors?