1. Consider the isothermal Euler equations

\[ u_t + f(u)_x = 0, \quad u = \begin{bmatrix} \rho \\ \rho v \end{bmatrix}, \quad f(u) = \begin{bmatrix} \rho v \\ \rho v^2 + \rho a^2 \end{bmatrix}, \quad a = \text{const} > 0 \]

with initial conditions

\[ u(x, 0) = w(x) = \frac{1}{2}(u_1 + u_0) + \frac{1}{2}(u_1 - u_0) \tanh(100x), \quad u_0, u_1 \text{ are constant states}, \]

and the corresponding conservative time-stepping scheme given by

\[ U_{j}^{n+1} = U_{j}^{n} - \frac{\Delta t}{\Delta x} \left[ G(U_{j}^{n}, U_{j+1}^{n}) - G(U_{j-1}^{n}, U_{j}^{n}) \right], \quad U_{j}^{0} = w(x_{j}), \quad n = 0, 1, 2, \ldots \]

where \( U_{j}^{n} \approx u(x_{j}, t_{n}) \) for the grid \( x_{j} = j \Delta x \) and time steps \( t_{n} = n \Delta t \), and \( G(u_{L}, u_{R}) \) is the numerical flux function.

(a) The Roe flux (without a sonic fix) is given by

\[ G(u_{L}, u_{R}) = f(u_{L}) + \sum_{\lambda_{k} < 0} \bar{\alpha}_{k} \bar{\lambda}_{k} \bar{r}_{k} = f(u_{R}) - \sum_{\lambda_{k} > 0} \bar{\alpha}_{k} \bar{\lambda}_{k} \bar{r}_{k} \]

where the eigenvalues and eigenvectors of the Roe matrix are given by

\[ \bar{\lambda}_{1} = \bar{v} - a, \quad \bar{\lambda}_{2} = \bar{v} + a, \quad \bar{r}_{k} = \begin{bmatrix} 1 \\ \bar{\lambda}_{k} \end{bmatrix}, \quad k = 1 \text{ or } 2 \]

using the Roe average

\[ \bar{v} = \frac{\sqrt{\rho_{L} v_{L}} + \sqrt{\rho_{R} v_{R}}}{\sqrt{\rho_{L}} + \sqrt{\rho_{R}}} \]

The wave strengths, \( \bar{z} = [\bar{\alpha}_{1}, \bar{\alpha}_{2}]^T \), are found by solving the \( 2 \times 2 \) system

\[ \bar{R} \bar{z} = \bar{\alpha}_{1} \bar{r}_{1} + \bar{\alpha}_{2} \bar{r}_{2} = u_{R} - u_{L} \]

Write a code to compute solutions of the initial-value problem for the isothermal Euler equations using the conservative time-stepping scheme with Roe Riemann flux (without a sonic fix).

(b) Run your code for \( a = 1 \) and for the initial conditions with \( u_{0} = [2, 0]^T \) and \( u_{1} = [1, 0]^T \). Use a uniform grid with \( \Delta x = 1/100 \) for the interval \( x \in [-1, 1] \), and a CFL time step given by

\[ \Delta t = \frac{0.9 \Delta x}{\lambda_{\text{max}}}, \quad \lambda_{\text{max}} = \max_{j} \{|v_{j}^{n}| + a\} \]

Plot the density \( \rho(x, t) \) and velocity \( v(x, t) \) at \( t = t_{\text{final}} = 0.6 \) given by the Roe scheme in part (a), and compare the results with a numerical solution given by the conservative time-stepping scheme with Lax-Friedrichs (LF) flux using the same grid. Ideally, it would be good to plot the exact solution as determined by the solution of the Riemann problem for left and right states given by \( u_{0} \) and \( u_{1} \), but you may “cheat” and plot the solution given by the LF scheme with \( N \) grid cells, where \( N \) is really large.
(c) **[Extra credit problem]** Rerun your code for the initial conditions with $u_0 = [2, 1.6]^T$ and $u_1 = [1, 0.8]^T$. Use a uniform grid with $\Delta x = 1/100$ as before but for the interval $x \in [-1, 2]$, and with the CFL time step given above. Compare your solution at $t = t_{\text{final}} = 0.6$ as before. Note that there is a sonic rarefaction for this case. Modify your code to include a sonic fix as discussed in class, and compare the results of the codes with and without the sonic fix.

2. Consider again the isothermal Euler equations with initial conditions as in problem 1. The aim of this problem is to modify the code discussed in problem 1(a) by including slope limiting. The time stepping scheme for this is

$$U_{j}^{n+1} = U_{j}^{n} - \frac{\Delta t}{\Delta x} \left[ \mathbf{G}(U_{j+1}^{n}, U_{j-1}^{n}) - \mathbf{G}(U_{j+1}^{n}, U_{j}^{n}) \right],$$

where $U_{j,\pm}^{n}$ are slope-limited updates of the cell average $U_{j}^{n}$. The formulas for $U_{j,\pm}^{n}$ are as follows:

$$U_{j,\pm}^{n} = U_{j}^{n} + \frac{1}{2} R_{j}^{n} \left( \pm I - \frac{\Delta t}{\Delta x} \Lambda_{j}^{n} \right) z_{j}^{n},$$

where $I$ is the $2 \times 2$ identity and

$$R_{j}^{n} = \begin{bmatrix} 1 & 1 \\ v_{j}^{n} - a & v_{j}^{n} + a \end{bmatrix}, \quad \Lambda_{j}^{n} = \begin{bmatrix} v_{j}^{n} - a & 0 \\ 0 & v_{j}^{n} + a \end{bmatrix}.$$

The vector of wave strengths is given by

$$z_{j}^{n} = \text{minmod} \left( (R_{j}^{n})^{-1}(U_{j}^{n} - U_{j-1}^{n}), (R_{j}^{n})^{-1}(U_{j+1}^{n} - U_{j}^{n}) \right),$$

where minmod is the minimum-modulus function discussed in class, which is applied component-wise. (Note: in class I described the approach in terms of primitive variables, but for the purposes of this exercise it seems simplest to use conserved variables as in the formulas above.)

(a) Modify the code in problem 1(a) to include slope limiting following the formulas above.

(b) Run your code for the set up in problem 1(b). Plot the “exact” solution (maybe just LF with a ton of grid points), the solution found in problem 1(b) and the new slope-limited solution. Comment on the results.

(c) **[Extra credit problem]** It makes a nice plot to compare the density and velocity computed using LF, LW, and the Roe solver with and without slope limiting. In particular, compare the LW and Roe solver with slope limiting since both are second-order methods (for smooth solutions).