Problem Set 11

Exercises for the Lecture Fundamentals of Simulation Methods (WS24/25)
Cornelis Dullemond and Michela Mapelli (Lecture Tuesday and Thursday 14h - 16h)
Lucia Haerer (Tutor Group T1 on Wednesday 11h - 13h)
Stefano Rinaldi (Tutor Group T2 on Thursday 11h - 13h)
Nicholas Storm (Tutor Group T3 on Friday 11h - 13h)
Submit the solution in electronic by the **Friday 6pm, January 17, 2025**.

10. Slope limiters and Riemann Solvers

10.1. Higher-order Advection schemes: Slope limiters (*10 points*)

We consider, again, the following 1D advection equation:

$$\frac{\partial q(x,t)}{\partial t} + \frac{\partial (q(x,t)u)}{\partial x} = 0 \tag{1}$$

We use a constant grid spacing Δx. We write the discretized equation in conservative form, using the fluxes at the interfaces of the cells:

$$\frac{q_i^{n+1} - q_i^n}{\Delta t} = \frac{F_{i-1/2}^{n+1/2} - F_{i+1/2}^{n+1/2}}{\Delta x} \tag{2}$$

where $F_{i\pm 1/2}^{n+1/2}$ is the flux at interface $i \pm 1/2$, time-averaged over the time step:

$$F_{i\pm 1/2}^{n+1/2} = \langle F_{i\pm 1/2}(t) \rangle_{t_n}^{t_{n+1}} = \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} F_{i\pm 1/2}(t)dt \tag{3}$$

With a linear subgrid model with slope σ_i in each cell, this mean interface flux becomes:

$$F_{i+1/2}^{n+1/2} = u_{i+1/2}\ \left[q_i\ +\ \tfrac{1}{2}\sigma_i(\Delta x - u_{i+1/2}\Delta t)\right] \tag{4}$$

for $u_{i+1/2} > 0$ and

$$F_{i+1/2}^{n+1/2} = u_{i+1/2}\ \left[q_{i+1}\ -\ \tfrac{1}{2}\sigma_{i+1}(\Delta x + u_{i+1/2}\Delta t)\right] \tag{5}$$

for $u_{i+1/2} < 0$. For simplicity you can assume $u > 0$ constant in space and time.

1. Write an advection function that computes q_i^{n+1} from q_i^n for all i in the domain, i.e., a single time step of this conservative advection equation. Allow for a prescribed slope in each cell σ_i.
2. Perform the same advection test as in the previous sheet, using $\sigma_i = 0$, and show that this is the same as the Donor-Cell algorithm.

3. Perform the same advection test as in the previous sheet, using $\sigma_i = (q^n_{i+1} - q^n_i)/\Delta x$ (downwind slope), and show that this is the same as the Lax-Wendroff algorithm.

4. Perform the same advection test as in the previous sheet, using $\sigma_i = (q^n_i - q^n_{i-1})/\Delta x$ (upwind slope). This is called the Beam-Warming algorithm.

5. Perform the same advection test as in the previous sheet, using $\sigma_i = (q^n_{i+1} - q^n_{i-1})/(2\Delta x)$ (symmetric slope). This is called Fromm's algorithm.

6. Which of these methods are TVD (total variation diminishing)?

Now let us apply so-called "slope limiters", which are actually non-linear recipes for the slopes σ_i. There exist many recipes, but here are two of the most popular ones. One method is the *minmod slope*:

$$\sigma^n_i = \text{minmod}\left(\frac{q^n_i - q^n_{i-1}}{\delta x}, \frac{q^n_{i+1} - q^n_i}{\delta x}\right) \tag{6}$$

where

$$\text{minmod}(a,b) = \begin{cases} a & \text{if } |a| < |b| \text{ and } ab > 0 \\ b & \text{if } |a| > |b| \text{ and } ab > 0 \\ 0 & \quad\quad \text{if } ab \leq 0 \end{cases} \tag{7}$$

This method chooses the smallest of the two slopes, provided they both have the same sign, and chooses 0 otherwise. The other is the *superbee slope*:

$$\sigma^n_i = \text{maxmod}(\sigma^{(1)}_i, \sigma^{(2)}_i) \tag{8}$$

with

$$\sigma^{(1)}_i = \text{minmod}\left(\frac{q^n_{i+1} - q^n_i}{\delta x}, 2\frac{q^n_i - q^n_{i-1}}{\delta x}\right) \tag{9}$$

$$\sigma^{(2)}_i = \text{minmod}\left(2\frac{q^n_{i+1} - q^n_i}{\delta x}, \frac{q^n_i - q^n_{i-1}}{\delta x}\right) \tag{10}$$

where maxmod is like minmod but with the condition $|a| < |b|$ and $|a| > |b|$ swapped.

7. Implement these two slope recipes into your advection function. In the file called `slope_tools.py` the functions for the slope limiters are provided for your convenience.

8. Perform, as before, the advection test using these two slopes.

9. Are they TVD?

10.2. A first 1D Riemann solver for hydrodynamics (*14 points*)

We will now implement the simplest Riemann solver for 1D *non*-isothermal hydrodynamics, i.e., we will include the energy equation this time:

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{f}}{\partial x} = 0 \tag{11}$$

where

$$\mathbf{q} = \begin{pmatrix} \rho \\ \rho u \\ \rho e_{\text{tot}} \end{pmatrix} \qquad \text{and} \qquad \mathbf{f} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho h_{\text{tot}} u \end{pmatrix} \tag{12}$$

with

$$e_{\text{tot}} = e_{\text{therm}} + \tfrac{1}{2}u^2, \qquad p = (\gamma - 1)\rho e_{\text{therm}}, \qquad h_{\text{tot}} = e_{\text{tot}} + p/\rho \tag{13}$$

The simplest Riemann solver of this kind is the HLL solver (from Harten, Lax and van Leer). It ignores the middle wave (the mass-advection wave) and decomposes $\Delta\mathbf{q}$ into *two* waves $\Delta_{(-)}\mathbf{q}$ and $\Delta_{(+)}\mathbf{q}$ which are the backward and forward moving sound waves. At each cell interface $i + 1/2$ this gives us a left state $\mathbf{q}_{L,i+1/2} = \mathbf{q}_i$, a right state $\mathbf{q}_{R,i+1/2} = \mathbf{q}_{i+1}$ and a middle state $\mathbf{q}_{M,i+1/2}$ which is assumed to be:

$$\mathbf{q}_M = \frac{\lambda_{(+)}\mathbf{q}_R - \lambda_{(-)}\mathbf{q}_L + \mathbf{f}_L - \mathbf{f}_R}{\lambda_{(+)} - \lambda_{(-)}} \tag{14}$$

where $\mathbf{f}_L \equiv \mathbf{f}_{L,i+1/2} = \mathbf{f}_i$ and $\mathbf{f}_R \equiv \mathbf{f}_{R,i+1/2} = \mathbf{f}_{i+1}$. Using the above expressions one can derive that the flux at the interface is then:

$$\bar{\mathbf{f}}_{i+1/2} = \begin{cases} \mathbf{f}_L & \text{if} \quad \lambda_{(-)} \geq 0 \\ \frac{\lambda_{(+)}\mathbf{f}_L - \lambda_{(-)}\mathbf{f}_R + \lambda_{(+)}\lambda_{(-)}(\mathbf{q}_R - \mathbf{q}_L)}{\lambda_{(+)} - \lambda_{(-)}} & \text{if} \quad \lambda_{(-)} < 0 < \lambda_{(+)} \\ \mathbf{f}_R & \text{if} \quad \lambda_{(+)} \leq 0 \end{cases} \tag{15}$$

The values of the signal speeds $\lambda_{(-)}$ and $\lambda_{(+)}$ are given by:

$$\lambda_{(-),i+1/2} = u_L - a_L \qquad \lambda_{(+),i+1/2} = u_R + a_R \tag{16}$$

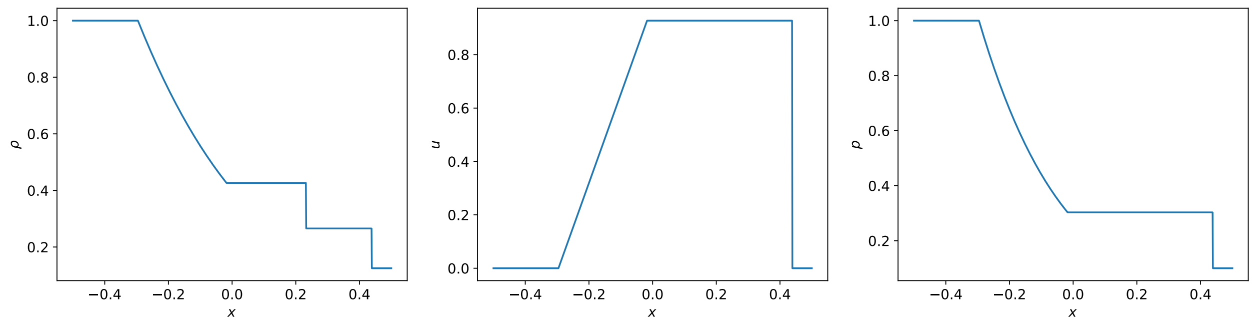
where a is the adiabatic sound speed $a = \sqrt{\gamma p/\rho}$.

1. Implement a 1D HLL hydrodynamics solver. Because the implementation of these equations can be a bit tricky, you are allowed to make use of `hydro_tools.py`, which contains a few functions that might be of help. If you like a challenge, you can also try to merely use these as an inspiration or not look at them at all. Up to you.

A standard test is called the *Sod shock tube test*, which is a Riemann problem defined by

$$\rho_L = 1, u_L = 0, p_L = 1, \rho_R = 0.125, u_R = 0, p_R = 0.1 \tag{17}$$

with $\gamma = 1.4$. In the file `hydro_tools.py` you will find a function called `analytic_riemann_problem_solution()` that computes the analytic solution to this problem, that you can use to compare your numerical result to. The analytic solution for $t = 0.25$ looks like this:



2. Run the Sod test on a domain $x \in [-0.5, 0.5]$ with 100 grid cells, with your Riemann solver from $t = 0$ to $t = 0.25$ using 400 time steps, and overplot the analytic solution at $t = 0.25$. You will see that the contact discontinuity and the shock wave are rather strongly smeared out, because the Riemann solver is still only first-order accurate.

To include slope limiters into the Riemann solver, we will use the MUSCL-Hancock algorithm. First we linearly extrapolate $\mathbf{q}$ from the cell centers to the cell walls. For cell wall $i + 1/2$ we then have a state $\mathbf{q}_{L,i+1/2}$ to the left of the cell wall and a state $\mathbf{q}_{R,i+1/2}$ to the right of the cell wall, given by

$$\begin{aligned} \tilde{\mathbf{q}}_{L,i+1/2} &= \mathbf{q}_i + \tfrac{1}{2}\mathbf{s}_i \Delta x & (18) \\ \tilde{\mathbf{q}}_{R,i+1/2} &= \mathbf{q}_{i+1} - \tfrac{1}{2}\mathbf{s}_{i+1} \Delta x & (19) \end{aligned}$$

where $\mathbf{s}$ is the slope (a vector of three elements, just like $\mathbf{q}$). Compare the similarity to Eqs. (4,5). Note that if you do not use the Lax-Wendroff or Beam-Warming slope recipe, you can compute $\mathbf{s}$ without reference to an advection velocity. That is why we do not use Lax-Wendroff or Beam-Warming slopes for Riemann solvers.
In the above comparison we also note that we have not yet accounted for the half time step terms of Eqs. (4,5). These arose because the linearly slanted subgrid solutions shifts over time. We have to account for this here too. This is done with the MUSCL-Hancock scheme, where we write:

$$\begin{aligned} \mathbf{q}_{L,i+1/2} &= \tilde{\mathbf{q}}_{L,i+1/2} + \tfrac{1}{2}(\mathbf{f}_{R,i-1/2} - \mathbf{f}_{L,i+1/2})\Delta t/\Delta x & (20) \\ \mathbf{q}_{R,i+1/2} &= \tilde{\mathbf{q}}_{R,i+1/2} + \tfrac{1}{2}(\mathbf{f}_{R,i+1/2} - \mathbf{f}_{L,i+3/2})\Delta t/\Delta x & (21) \end{aligned}$$

Now, for each cell interface $i + 1/2$, you pass the $\mathbf{q}_{L,i+1/2}$ and $\mathbf{q}_{R,i+1/2}$ state vectors to the function that computes the HLL flux $\bar{\mathbf{f}}_{i+1/2}$ (Eqs. 15, 16), and use these fluxes to update the old state q_i^n to the new state q_i^{n+1}.

3. Implement the MUSCL-Hancock scheme with the SuperBee flux limiter into your HLL solver.

4. Run the Sod test with the SuperBee slope limiter and show that this gives a much better match to the analytic solution as the one without slope limiter.