Exercises for *Relativistic Hydrodynamics*

Elias R. Most [1,2]

[1] *Center for Theoretical Science & Gravity Initiative, Princeton University, Princeton, NJ 08544*

[2] *School of Natural Sciences, Institute for Advanced Study, Princeton, NJ 08540*

In this set of example problems, we want to numerically solve different hyperbolic conservation laws in flux conservative form

$$\partial_t \mathbf{U} + \nabla \cdot \mathbf{F} = \mathbf{S}\,, \tag{1}$$

where $\mathbf{U}$ is the state vector, $\mathbf{F}$ the flux and $\mathbf{S}$ the source term. To do so, we will use a number of example routines provided in this course. Schematically, the solution of (1) proceeds as follows:

1. Specify initial data (`initial_data.hh`)
2. Iterate over the main loop (time integration) (`main.cc/advance.hh`)
   (a) Enforce boundary conditions (`boundaries.hh`),
   (b) Periodically output solution (`output.hh`),
   (c) Interpolate $\mathbf{U}$ to the cell interfaces (`reconstruct.hh`),
   (d) Compute $\mathbf{F}$ using a Riemann solver (`riemann.hh & system.hh`),
   (e) Perform the flux update $\nabla \cdot \mathbf{F}$ (`advect.hh`).

While you should feel free to look up any of these steps in the source code files (`*.hh`), we will mainly modify the code in two places: *Initial data* and *flux computation.*
More specifically, we will implement initial conditions, mathematical expressions for $\mathbf{U}$, $\mathbf{F}$ and $\mathbf{S}$, as well as the characteristic speeds $c_\pm$ needed for the (approximate) solution of the Riemann problem. Consequently, we will only edit the `main.cc`, `initial_data.hh` and `system.hh` files.
Every task comes with its own folder. This demonstrator code has no external dependencies, and can be compiled using a C++14 compiler, e.g. by typing

```
clang++ -O3 -march=native -fopenmp -o example ./main.cc
```

in the respective folder. It is advised to use matplotlib (or other tools such as gnuplot) for visualisation of the data, which is stored in plain text files.

1 Advection equation

In the main part of the exercises, we will consider the advection system,

$$\partial_t u + v^i\left(u\right)\partial_i u = 0\,, \tag{2}$$

for various choices of the advection velocity $v^i\left(u\right)$ in 1-D and 2-D.

1.1 Getting started with advection

In the first part of this problem we will consider a one-dimensional setup of (2) with $v^x = \text{const}$. Complete the following tasks:

1. What is the analytical solution to (2) with $v^x = \text{const}$?

2. Compute the flux $\mathbf{F}$ and characteristic speeds $c_{\pm}$, and implement them in the labeled sections in `system.hh`. *In the following, we will adopt a computational domain from [0:1] with characteristics* $v^x = 1$.

3. Implement two initial conditions:

   (a) Smooth initial condition (`Advected_Wave1D`)

   $$u\left(x, t=0\right) = \exp(-2\cos(2\pi x))$$

   (b) Discontinuous step function (`Advected_Step1D`)

   $$u\left(x, t=0\right) = \begin{cases} 1 & \left|x - \frac{1}{2}\right| < \frac{1}{4} \\ 0 & \text{else} \end{cases}$$

   Using periodic boundary conditions, evolve both initial conditions (they can be selected in `main.cc`) for 10 periods.
   What do you notice for the step function?

4. What happens to the non-linear advection equation? Try to solve

   $$\partial_t u + u\partial_x u = 0\,, \tag{3}$$

   for the initial condition:

   $$u\left(x, t=0\right) = \cos(2\pi x)\,.$$

1.2 Understanding numerical errors

1. Using the previous initial conditions, we want to assess the quality of the numerical solution. To this end, we can compare the advected solution at time $t = nT$, where n is an integer and T is the period, to the initial condition at time $t = 0$. The error of the solution will scale with the grid resolution Δx,

   $$u\left(x, nT\right) - u\left(x, 0\right) \sim A\left(\Delta x\right)^k\,, \tag{4}$$

   where k is called the convergence order. Perform the simulation for various Δx and compute k. What does k depend on?

2. In the main part of this problem, we want to study the impact of different reconstruction algorithms. In particular, we will consider several approaches commonly used in the literature,

   - MinMod
   - XPPM4
   - MP5
   - WENO-Z

   To have a more challenging test problem, we will make use the slotted cylinder test (see Figs. 13 & 14 in `https://portal.nersc.gov/project/edge_mdl/cogent_files/Papers/Colella_JCP_11.pdf`). This 2-D problem consists of a cylinder, out of which a vertical stripe has been excised. The cylinder is placed a distance d away from the origin and rotated at constant angular speed. This problem comes fully implemented in `Advection2D/SlottedCylinder`.

   Choosing the different reconstruction algorithms in `main.cc`, perform simulations for three different grid sizes (64^2, 128^2, 256^2). What do you notice? Which one performs better?
   (Optional) Compute the convergence order using the L^2 norm with (4).

2 Relativistic hydrodynamics

After gaining some familiarity with the numerical methods, we are ready to tackle the hydrodynamics system. This is given by

$$\mathbf{U}^i = \left(\rho u^0, \rho h \left(u^0\right)^2 - p - \rho u^0, \rho h u^0 u_i\right) \ , \ \mathbf{F}^i = \left(\rho u^i, \rho h u^0 u^i - \rho u^i, \rho h u^i u_j - p\delta^i_j\right) \ , \ \mathbf{S}^i = 0$$

Here ρ is the rest-mass density, u^μ the four velocity, p the pressure, and $h = 1 + \varepsilon + p/\rho$ is the specific enthalpy, with $e = \rho(1+\varepsilon)$ being the total energy density, and ε being the specific internal energy, respectively. For simplicity of this test, we assume a flat Minkowski spacetime, with $\eta^{\mu\nu} = \mathrm{diag}(-1,1,1,1)$ as the metric. We will further use the ideal fluid equation of state $p = \rho\varepsilon\,(\Gamma - 1)$, where $\Gamma = 4/3$ is a constant.

Different from the previous problem, we can see that the conserved variables $\mathbf{U}^i$ are non-linearly related to the primitives $\mathcal{P}^i = (\rho, \rho\varepsilon, u_i)$. While $\mathbf{U}^i$ can be trivially computed from $\mathcal{P}^i$, the inverse is not true and requires non-linear root-finding (see, e.g., `arXiv:1712.07538`). This primitive inversion is slightly more involved, and is already implemented in the example code.

1. Implement the relativistic hydrodynamics system in the example code `SRHD`.
   Hints: To maintain the correct ordering of the variables in the state and primitive vector, make use of the global indices `(RHOB, EPS, WVX, WVY, WVZ)`, and `(RHOSTAR, TAUENERGY, STX, STY, STZ)` for the primitives and conservatives, respectively. You can compare with a solution file `system_solution.hh`, that can also be used for the remainder of this problem.

2. Now that the system is implemented, we want to test it on some one dimensional Riemann problems (`Shocktube`). Some are provided as examples, but feel free to change them.
   (Optional) If you really want to know that you are getting the right solution (and you have a Fortran compiler installed), feel free to check out the exact solver for the relativistic (magneto-)hydrodynamic Riemann problem (`https://www.brunogiacomazzo.org/ExactSolver/`).

3. Repeat the convergence analysis from 1.2 for a Shocktube problem. You can either compare to the exact solution, or you use a third (higher resolution) solution as reference.

4. (Optional) A standard test problem for a relativistic hydrodynamics is the Kelvin-Helmholtz test (e.g., `arXiv:1101.3573`). This test involves a strong shear flow, which is Kelvin-Helmholtz unstable. A setup can be found in the `SRHD/KH` folder.

   (a) Run the test with increasing values of the relative shear velocity `Vshear`.
   What happens if `Vshear` gets too large? *(Hint: Plot the sonic Mach number $\mathcal{M}_s = v/c_s$, where c_s is the sound speed.)*

   (b) Depending on your available compute power, run the test at increasing resolution, doubling at each iteration. What do you notice in terms of the vortices that form?

   (c) (Absolutely optional!) Perform a Fourier transformation of the kinetic energy for the final time at various resolutions, and reconstruction algorithms in Problem 1.2. What does the result tell you about the presence/absence of a physical small-scale cut-off?