Relativistic Hydrodynamics: Fundamentals

David Radice

May 12, 2018

Introduction

The goal of this exercise sheet is to become familiar with the equations of relativistic hydrodynamics and their basic properties, and to learn about the most important building blocks required to assemble the simulations codes used in numerical relativity. We will write a small python code that can solve the equations of special-relativistic hydrodynamics in 1D (1+1) starting from a toy code for non-relativistic gasdynamics. We will use units where $c = 1$ throughout these notes.

Basic equations

The special relativistic hydrodynamics equation describe the conservation of energy and momentum of a fluid:

$$\partial_\nu T^{\mu\nu} = 0, \tag{1}$$

where we have used Einstein's convention of sum over repeated indices. We consider a perfect fluid, i.e., we neglect viscosity and heat transport inside of the fluid, so that the stress energy tensor reads

$$T^{\mu\nu} = (\rho + p)\, u^\mu\, u^\nu + p\, \eta^{\mu\nu}\,, \tag{2}$$

where ρ is the energy density of the fluid measured in a comoving frame, p is the pressure, u^μ the fluid four-velocity and $\eta = \mathrm{diag}(-1, 1, 1, 1)$ is the metric of Minkowsky spacetime. We remind that the four-velocity can be written as

$$u^\mu = W(1, v^i)\,, \tag{3}$$

Where

$$W = (1 - \eta_{ij} v^i v^j)^{-1/2} \tag{4}$$

is the Lorentz factor and v^i the spatial velocity.

At the energy scales relevant for neutron star mergers we can assume the baryon number to be conserved. We introduce the number density current $J^\mu = n u^\mu$. The conservation of the total number of baryons reads

$$\partial_\mu J^\mu = 0\,. \tag{5}$$

In analogy with the classical case it is convenient to split the energy density in a term due to the rest-mass contribution and a term due to the thermal energy

$$\rho = n\, m_b\, (c^2 + \epsilon)\,, \tag{6}$$

where we have introduced a mass scale m_b and temporarily restored the speed of light c $(= 1)$ for clarity. ϵ is the specific internal energy. We also introduce the rest-mass density (not to be confused with the energy density ρ):

$$\rho_0 = n\, m_b\,. \tag{7}$$

The equation of state we will consider for relativistic hydrodynamics is that of an ideal gas

$$p = m_b\,(\gamma - 1)\, n\, \epsilon = (\gamma - 1)\, \rho_0\, \epsilon\,, \tag{8}$$

where $1 < \gamma \leq 2$ is the so-called adiabatic index.

In the case of smooth relativistic flows of a perfect fluid it is possible to show the law of entropy conservation holds

$$\partial_\mu (n\, s\, u^\mu) = 0\,, \tag{9}$$

where s is the entropy per baryon. Under the assumption of isentropic flow, the specific internal energy of the fluid ϵ is uniquely determined by the number density n. Consequently, a barotropic equation of state can be used to model smooth flows:

$$p_{\rm b}(\rho_0) := p\,(\rho_0, \epsilon(\rho_0))\ . \tag{10}$$

However, when shocks are present Eq. (9) breaks down and it is necessary to solve Eq. (1) and (5) that still hold, at least in the integral sense, even for shock solutions.

Homework

1. This repository contains a python-3 library, `PyHRSC1D.py`, implementing a variant of the classical Kurganov-Tadmor central scheme, one of the most commonly employed schemes in current numerical relativity codes. There is also a jupyter notebook `WorkBook.ipynb` that shows how to use `PyHRSC1D` to simulate a classical shock problem in non-relativistic hydrodynamics. You should be able to run this code using a recent version of jupyter with a python-3 kernel. I recommend the use of the Anaconda python distribution, since it comes with everything needed to run the examples: `https://www.anaconda.com/download`. Note how the equations of hydrodynamics are implemented. Compare the numerical solution with the exact solution available in the file `data/newt_sod_t0p2.npz`. Try to vary the numerical resolution between $N = 50$ and 1000 grid points.

2. There is vast literature available on how to numerically solve hyperbolic equations in conservation form, i.e., written as

$$\partial_t F^0(u) + \partial_i F^i(u) = S(u)\,, \tag{11}$$

where u is a set of variables completely describing the state of the fluid (for instance velocity, density, and specific internal energy), F^0 are the conservative variables (e.g., the linear momentum), and F^i and S are the fluxes and sources, respectively. Take Eqs. (5) and (1) and cast them in the form of Eq. (11). What are the conservative variables, the fluxes, and the source terms for special relativistic hydrodynamics?

3. The numerical solver evolves F^0, but fluxes and other quantities of interest are given as a function of u. For this reason u needs to be recovered from F^0 using a numerical inversion procedure. One way to proceed is to assume knowledge of the pressure $\tilde{p}$. Then it is possible

to compute all primitive quantities from F^0 analytically. In particular, it is possible to recover ρ_0 and ϵ given a trial value for the pressure $\tilde{p}$. Then, with the guesses for ρ_0 and ϵ ($\rho_0(\tilde{p})$ and $\epsilon(\tilde{p})$), we can compute a new pressure using the equation of state:

$$p^* = p\left(\rho_0(\tilde{p}), \epsilon(\tilde{p})\right) . \tag{12}$$

If $p^* = \tilde{p}$ this means that we have found the right pressure and $\rho_0(\tilde{p})$ and $\epsilon(\tilde{p})$ are the right values for the primitives. Otherwise, we need to make another guess. A systematic method to proceed is to use a standard root finding algorithm to find the zero of

$$p\left(\rho_0(\tilde{p}), \epsilon(\tilde{p})\right) - \tilde{p} = 0 . \tag{13}$$

Implement a function that recovers the primitive variables u given the conservatives F^0. To this aim use the `brentq` routine of `scipy.optimize` to bracket and refine the solution of Eq. (13).
(Hint: use the fact that $v < 1$ to derive an upper bound on the pressure.)

4. `WorkBook.ipynb` contains a derivation of the equation describing the propagation of sound waves in a classical fluid. Extend it to the relativistic case. How do the fundamental thermodynamic identities for a classical fluid change in the relativistic case? What is the expression for the sound speed for the equation of state (8)?
(Hint: linearize the hydrodynamics equations around a static background and note that $\delta W = 0$ at linear order.)

5. The numerical solver needs to know the speed at which disturbances propagate in the fluid measured in the lab frame: the characteristic velocities. In a perfect fluid information propagates with the flow, at the velocity of the fluid v, and with the sound waves. What are the three characteristic velocities assuming slab geometry?
(Hint: generalize the non-relativistic case by using the proper formula for the addition of velocities.)
(Shortcut: using the speed of light as upper bound on the characteristic speeds, that is take -1 and 1 as the characteristic velocities, is also possible. This will yield a less accurate, but still stable numerical scheme.)

6. Implement a new python class `RelEuler` in the python notebook with the relativistic version of the `Euler` classes. You will need to implement functions to compute F^0, F^i, and S given u, a function to compute the characteristic propagation speeds, and a function to compute u given F^0 (calling the stand alone routine implemented in point 3).

7. Solve the relativistic version of the Sod problem given in `WorkBook.ipynb`. Compare the numerical solution at the time $t = 0.4$ with the exact solution that can be found in the file `data/rel_sod_t0p4.npz`.

8. Solve a shock problem where the initial data is given by the following two states:

$$(n_L, v_L, \epsilon_L) = (0.001, 0, 1500), \quad (n_R, v_R, \epsilon_R) = (0.001, 0, 0.015). \tag{14}$$

Assume adiabatic index $\gamma = 5/3$ and $m_b = 1$. Compare with the exact solution given at time $t = 0.4$ included in the file `data/rel_blast_t0p4.npz`. What is happening? Why is this

problem hard?
(Hint: you might need to tune some of the numerical settings of the `KurganovTadmor` solver to be able to run this test successfully.)
(Context: this is problem #2 of the review by Martí and Müller.)

References

Font, Jose A. (2008). "Numerical Hydrodynamics and Magnetohydrodynamics in General Relativity". In: *Living Rev. Rel.* 11, p. 7. DOI: `10.12942/lrr-2008-7`.

Gourgoulhon, Eric (2006). "An Introduction to relativistic hydrodynamics". In: *EAS Publ. Ser.* 21, pp. 43–79. DOI: `10.1051/eas:2006106`. arXiv: `gr-qc/0603009 [gr-qc]`.

Kurganov, A. and E. Tadmor (2000). "New High-Resolution Central Schemes for Nonlinear Conservation Laws and Convection–Diffusion Equations". In: *J. Comput. Phys.* 160.1, pp. 241–282. DOI: `10.1006/jcph.2000.6459`.

Martí, J. M. and E. Müller (2003). "Numerical Hydrodynamics in Special Relativity". In: *Living Reviews in Relativity* 6, 7, p. 7. DOI: `10.12942/lrr-2003-7`.