

Relativistic Hydrocode

Joseph Corrado
Andrew MacFadyen
Chris Tiede

December 13, 2018

1 Reconstruction Techniques

1.1 First Order in Space

At each interface the U_j^n manifests a jump which generates left and right states of the local riemann problem. The original idea by Godunov was to use a piecewise constant reconstruction of the values of the conserved variables throughout each of the cells in order to solve the riemann problem at each interface. This method works to first order in space, as it takes no information about the derivatives of the function into account. This reconstruction is given by:

$$U(x, 0) = \begin{cases} U_j^n, & x < x_{j+1/2} \\ U_{j+1}^n, & x > x_{j+1/2} \end{cases}$$

When we use this reconstruction method, we only need one guard cell on either side because we are simply taking the left and right hand conserved states on either side of every boundary to compute the numerical flux across each boundary.

1.2 Total variation Diminishing Methods

The monotonicity of a solution is closely related to the appearance of oscillations. Given a numerical method, where evolution of the conserved variables forward in time is a function of the current conserved variables, a method is called monotone if:

$$\frac{\partial f}{\partial U_j^n} \geq 0, \quad \forall U_j^n$$

If we require that our method be monotone, this is equivalent to all of the coefficients of the linear combination used to propagate the solution forward in time need to be positive semi-definite. If we apply this requirement, we obtain the result that the solution will satisfy:

$$\max_j \{U_j^{n+1}\} \leq \max_j \{U_j^n\}, \quad \min_j \{U_j^{n+1}\} \geq \min_j \{U_j^n\}$$

If we have these properties for our solution, then our method will not introduce spurious new extrema in the solution as time evolves. In this sense, the monotonicity of a numerical scheme is closely related to the potential appearance of spurious oscillations in the solution across a discontinuity. In order to attenuate these oscillations, we can introduce artificial velocity. In order to measure the oscillations appearing in the solution, we define the total variation as:

$$TV(U^n) = \sum_{j=-\infty}^{\infty} |U_j^n - U_{j-1}^n|$$

Requiring that the scheme does not produce spurious oscillations is equivalent to requiring that the total variation does not increase in time, or that the total variation is bounded by the total variation of the initial data. Given this definition of variation, a numerical method is total variation diminishing (TVD) if our solution satisfies:

$$TV(U^{n+1}) \leq TV(U^n)$$

for every timestep. TVD methods differ from artificial viscosity by controlling the appearance of oscillations in the solution without explicitly introducing viscous terms in the numerical scheme, and by reducing the oscillations of the methods by the method itself.

One way to characterize schemes by the ability to not introduce or increase oscillations in the solution, we consider methods such that:

$$U_j^n \geq U_{j+1}^n, \quad \forall j$$

Methods that satisfy this constraint are said to be monotonicity preserving. This results in preventing the appearance of oscillations since monotonicity-preserving methods do not permit new local extrema as the value of local minima cannot decrease and the value of local maxima cannot increase. A result due to Harten states that any monotone numerical method is TVD, and any TVD method is monotonicity preserving. However, this comes at the cost that TVD methods do not extend beyond second-order accuracy, and they reduce to first-order at local extrema. If we want to construct higher-order methods, we must drop the total variation condition. In practice, the TVD property can be imposed through a slope limiter method. A theorem encapsulating the restrictions on TVD methods was given by Godunov: a linear and monotonicity-preserving scheme is at most first-order accurate.

Modern HRSC methods all possess the properties:

1. at least second order accuracy on smooth parts of the solution
2. sharp resolution of discontinuities without large smearing
3. absence of spurious oscillations in the solution
4. convergence to the true solution as the grid is refined
5. no use of artificial viscosity terms

1.3 Second Order in Space

As mentioned above, slope limiter methods can be used to improve the representation of the solution within the cell while imposing a TVD property on the method. In order to increase the order of convergence, we can perform a piecewise linear reconstruction inside each cell:

$$U_j^n(x) = U_j^n + \sigma_j^n(x - x_j), \quad x_{j-1/2} \leq x \leq x_{j+1/2}$$

The slope limiter method I used was a generalized minmod method. In this reconstruction, the slope σ is given by the minmod function evaluated on the adjacent states of the conserved variables. The minmod function is defined as:

$$\text{minmod}(x, y, z) = \frac{1}{4} |\text{sign}(x) + \text{sign}(y)| (\text{sign}(x) + \text{sign}(z)) \min(|x|, |y|, |z|)$$

Using this method, we need to use two cells on either side of every boundary, and so we need two guard cells on either side of the grid for this reconstruction technique. This reconstruction is given by:

$$U(x, 0) = \begin{cases} U_j^+, & x < x_{j+1/2} \\ U_{j+1}^-, & x > x_{j+1/2} \end{cases}$$

Where $U_j^+ = U_j(x_j + 1/2)$ and $U_j^- = U_j(x_{j-1/2})$. Where we are interpreting integer values of the subscript to be the bin centers, and half integer values of the subscripts refer to the boundaries between cells. The reconstructed states are then given by:

$$c_{i+1/2}^L = c_i + 0.5 \minmod(\theta(c_i - c_{i-1}), 0.5(c_{i+1} - c_{i-1}), \theta(c_{i+1} - c_i))$$

$$c_{i+1/2}^R = c_{i+1} - 0.5 \minmod(\theta(c_{i+1} - c_i), 0.5(c_{i+2} - c_i), \theta(c_{i+2} - c_{i+1}))$$

This generalized minmod function is a single parameter version of slope limiter, and has the property that when $\theta = 1$, it becomes more diffusive normal minmod limiter, and when $\theta = 2$, it becomes the monotonized central-difference limiter. In my code, I used $\theta = 1.5$. The minmod limiter always takes the least steep slope amongs those provided by the backward and forward finite-difference, while the MC limiter compares three different slopes and takes the least steep amongst the three.

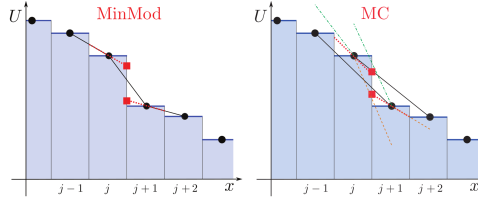


Figure 1: Minmod and MC Slope Limiters

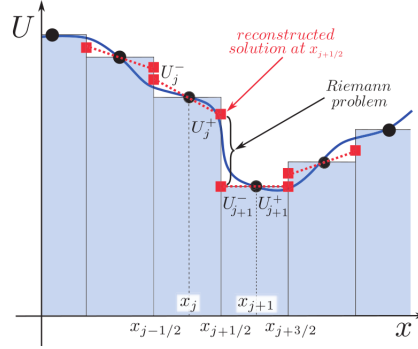


Figure 2: Schematic Representation of Boundary-Extrapolated Values

2 Physical Fluxes

We can compute the physical fluxes from the expressions for the fluxes as a function of the conserved variables and primitive variables. The physical fluxes are given from the euler equations by:

$$F = \begin{bmatrix} \rho v \\ \rho v^2 + p \\ (E + p)v \end{bmatrix}$$

After retrieving the primitives from the conserved variables, we can compute the physical fluxes for each of the physical cells.

3 Numerical Flux

Computing the exact solution to the riemann problem at the boundary of each of the cells is computationally expensive because it involves iterations over all of the cells for every timestep. In order to avoid doing this to be more computationally efficient, modern high resolution shock capturing schemes employ approximate riemann solvers to numerically approximate the solutions to the riemann problem at each of the boundaries. In my project, I used one of the simplest numerical flux functions, the

4 Ghost cells

In order to reconstruct the states to the left and right of the first and last interface, I needed to use imaginary ghost cells in which I fill the values of the conserved quantities as specified by the boundary conditions each iteration. I then could use these values of the ghost cells in order to compute the left and right states during reconstruction. For first order in space, I only needed one ghost cell, whereas for second order in space, when I did the TVD minmod reconstruction, I needed two ghost cells on either side.

5 Retrieving the Primitives from the Conserved Variables

In order to compute the physical and numerical fluxes, one needs to retrieve the primitive variables from the conserved variables at least once per timestep. For the nonrelativistic case, the prescription for computing the primitive variables comes directly from a manipulation of the euler equations.

5.1 Nonrelativistic Case

The equations for one dimensional hydrodynamics are given by:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = 0$$

Where the conserved variables are:

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho v \\ E \end{bmatrix}$$

and the fluxes are given by:

$$\mathbf{F} = \begin{bmatrix} \rho v \\ \rho v^2 + P \\ (E + P)v \end{bmatrix}$$

The first primitive ρ is given by the first component of the conserved vector. The second primitive, v , is given by the second component of the conserved vector, divided by the first component of the conserved vector. The third primitive, the pressure can be obtained from the conserved variables once we introduce an equation of state in order to close the system of equations. If we assume an ideal gas equation of state:

$$p(\rho, e) = (\Gamma - 1)\rho e$$

Using the expression for the total energy density:

$$E = \rho e + \frac{1}{2}\rho v^2$$

We can use the equation of state to eliminate the specific internal energy (e). Doing this yields the expression:

$$p = \left(E - \frac{1}{2}\rho v^2 \right) (\Gamma - 1)$$

which can be obtained by using the components of the conserved vector.

5.2 Relativistic Case

In the relativistic case, the conserved variables become heavily coupled via the lorentz factor as we consider velocities a substantial fraction of the speed of light. For the relativistic case, the conserved variables are given by:

$$U = \begin{bmatrix} D \\ S_j \\ \tau \end{bmatrix} = \begin{bmatrix} \rho W \\ \rho h W^2 v_j \\ D(hW - 1) - p \end{bmatrix}$$

However, trying to retrieve the primitive variables from those expressions for the conserved variables cannot be done analytically. So, we need to use a root-finding algorithm in order to retrieve the pressure every timestep. Using the pressure and the conserved variables, we can then retrieve the rest of the primitive variables.

In my code, I implemented a newton-raphson root-finding algorithm to solve for the optimal value of the pressure for each cell using the equation of state and an implicit expression for pressure of the state. The implicit expression for the pressure is of the form:

$$p - \bar{p}[\rho(\mathbf{U}, p), e(\mathbf{U}, p)] = 0$$

Where \bar{p} is given by the equation of state. My expressions in order to calculate the value of the equation of state, evaluated at my current guess for the pressure, and the current values of the conserved variables are given by:

$$v^* = \frac{S_j}{\tau + p + D}$$

$$W^* = \frac{1}{\sqrt{1 - (v^*)^2}}$$

$$\rho^* = \frac{D}{W^*}$$

$$\epsilon^* = \frac{\tau + D(1 - W^*) + (1 - (W^*)^2)p}{DW^*}$$

The implicit expression for the pressure that I used in my code is:

$$(\Gamma - 1)\rho^*\epsilon^* - p = 0$$

For the newton-raphson algorithm, I need to supply the derivative of the function for which I am trying to find the root of. In Numerical Hydrodynamics in Special Relativity, by Marti and Muller, it is stated that the derivative of the implicit function with respect to \bar{p} can be approximated using the expression:

$$\frac{df}{dp} = (v^*)^2(cs^*)^2 - 1$$

Where the sound speed here is the relativistic one, and is evaluated using our current guess for the pressure and the current value of the conserved variables. This approximation has the property that it tends to the exact derivative as the solution is approached.

One issue that I encountered with this was that sometimes I would get nonphysical results from the root-finding algorithm. In order to correct this, I checked for whenever there was a velocity greater than 1, or a pressure that was less than the theoretical minimum ($P_{min} = |S - \tau - D|$, as stated in Solving 1-D SRH Equations Using Different Numerical Method and Results - Donmez), and corrected it by replacing the value with the minimum value for the pressure when the pressure went below, or a value for the velocity slightly below 1 whenever the velocity went above 1. However, I found that the problem was largely averted by retrieving the primitives before reconstructing the left and right conservative states, and then reconstructing the primitives directly. This method helped significantly for avoiding negative pressures, as opposed to reconstructing the conserved variables close to the boundary, and then retrieving the primitive variables from the reconstructed conserved variables.