Notes on the Euler Equations

These notes describe how to do a piecewise linear or piecewise parabolic method for the Euler equations.

1 Euler equation properties

The Euler equations in one dimension appear as:

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

$$\frac{\partial (\rho u)}{\partial t} + \frac{\partial (\rho u u + p)}{\partial x} = 0 \tag{2}$$

$$\frac{\partial (\rho E)}{\partial t} + \frac{\partial (\rho u E + u p)}{\partial x} = 0 \tag{3}$$

These represent conservation of mass, momentum, and energy. Here ρ is the density, u is the one-dimensional velocity, p is the pressure, and E is the total energy / mass, and can be expressed in terms of the specific internal energy and kinetic energies as:

$$E = e + \frac{1}{2} u^2 \tag{4}$$

The equations are closed with the addition of an equation of state:

$$p = \rho e (\gamma - 1) \tag{5}$$

where γ is the ratio of specific heats for the gas/fluid (for an ideal, monatomic gas, $\gamma = 5/3$).

In this form, the equations are said to be in *conservative form*. They can be written as:

$$U_t + [F(U)]_x = 0 \tag{6}$$

with

$$U = \begin{pmatrix} \rho \\ \rho u \\ \rho E \end{pmatrix} \qquad F(U) = \begin{pmatrix} \rho u \\ \rho u u + p \\ \rho u E + u p \end{pmatrix} \tag{7}$$

An alternate way to express these equations is using the *primitive variables*: ρ, u, p.

Exercise 1: Show that the Euler equations in primitive form can be written as

$$q_t + A(q) q_x = 0 \tag{8}$$

where

$$q = \begin{pmatrix} \rho \\ u \\ p \end{pmatrix} \qquad A(q) = \begin{pmatrix} u & \rho & 0 \\ 0 & u & 1/\rho \\ 0 & \gamma p & u \end{pmatrix} \tag{9}$$

The eigenvalues of A can be found via $|A - \lambda I| = 0$, where $|\ldots|$ indicates the determinant and λ are the eigenvalues.

Exercise 2: Show that the eigenvalues of A are $\lambda^{(-)} = u - c$, $\lambda^{(\circ)} = u$, $\lambda^{(+)} = u + c$ where the speed of sound is $c = \sqrt{\gamma p / \rho}$.

We'll use the symbols $\{-, \circ, +\}$ to denote the eigenvalues and their corresponding eigenvectors throughout these notes. These eigenvalues are the speeds at which information propagates for the fluid equations. Since the eigenvalues are real, this system (the Euler equations) is said to be *hyperbolic*. Additionally, since $A = A(q)$, the system is said to be *quasi-linear*. The right and left eigenvectors can be found via:

$$A\, r^{(\nu)} = \lambda^{(\nu)} r^{(\nu)}\,; \qquad l^{(\nu)}\, A = \lambda^{(\nu)} l^{(\nu)} \tag{10}$$

where $\nu = \{-, \circ, +\}$ corresponding to the three waves, and there is one right and one left eigenvector for each of the eigenvalues.

Exercise 3: Show that the right eigenvectors are:

$$r^{(-)} = \begin{pmatrix} 1 \\ -c/\rho \\ c^2 \end{pmatrix} \qquad r^{(\circ)} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \qquad r^{(+)} = \begin{pmatrix} 1 \\ c/\rho \\ c^2 \end{pmatrix} \tag{11}$$

and the left eigenvectors are:

$$l^{(-)} = \begin{pmatrix} 0 & -\frac{\rho}{2c} & \frac{1}{2c^2} \end{pmatrix} \qquad l^{(\circ)} = \begin{pmatrix} 1 & 0 & -\frac{1}{c^2} \end{pmatrix} \qquad l^{(+)} = \begin{pmatrix} 0 & \frac{\rho}{2c} & \frac{1}{2c^2} \end{pmatrix} \tag{12}$$

Note that in general, there can be an arbitrary constant in front of each eigenvector. Here they are normalized such that $l^{(i)} \cdot r^{(j)} = \delta_{ij}$.

A final form of the equations is called the *characteristic form*. Here, we wish to diagonalize the matrix A. We take the matrix R to be the matrix of right eigenvectors, $R = (r^{(-)}|r^{(\circ)}|r^{(+)})$, and L is the corresponding matrix of left eigenvectors. Note that $L\,R = I = R\,L$, and $L = R^{-1}$.

Exercise 4: Show that $\Lambda = LAR$ is a diagonal matrix with the diagonal elements simply the 3 eigenvalues we found above.

Defining $dw = Ldq$, we can write our system as:

$$w_t + \Lambda w_x = 0 \tag{13}$$

Here, the w are the characteristic variables. Note that we cannot in general integrate $dw = Ldq$ to write down the characteristic quantities. Since Λ is diagonal, this system is a set of decoupled advection-like equations. If the system were linear, then the solution to each would simply be to advect the quantity $w^{(\nu)}$ at the wave speed $\lambda^{(\nu)}$.

The way to think about this system is that there are 3 wave speeds (one for each eigenvalue) and each wave carries with it a change in the characteristic variable. Since $dq = L^{-1}dw = Rdw$, the jump in the primitive variable across each wave is proportion to the right-eigenvector associated with that wave. So, for example, since $r^{(\circ)}$ is only non-zero for the density element, this then means that only density jumps across the $\lambda^{(\circ)} = u$ wave—pressure and velocity are constant across this wave (see for example, Toro [12], Ch. 2, 3 or LeVeque [7] for a thorough discussion). Figure 1 shows the three waves emanating from an initial discontinuity.

2 Reconstruction of interface states

We will solve the Euler equations using a high-order *Godunov method*—a finite volume method whereby the fluxes through the interfaces are computed by solving the Riemann problem for our system. The finite-volume update for our system appears as:

$$U_i^{n+1} = U_i^n + \frac{\Delta t}{\Delta x}\left(F_{i-1/2}^{n+1/2} - F_{i+1/2}^{n+1/2}\right) \tag{14}$$

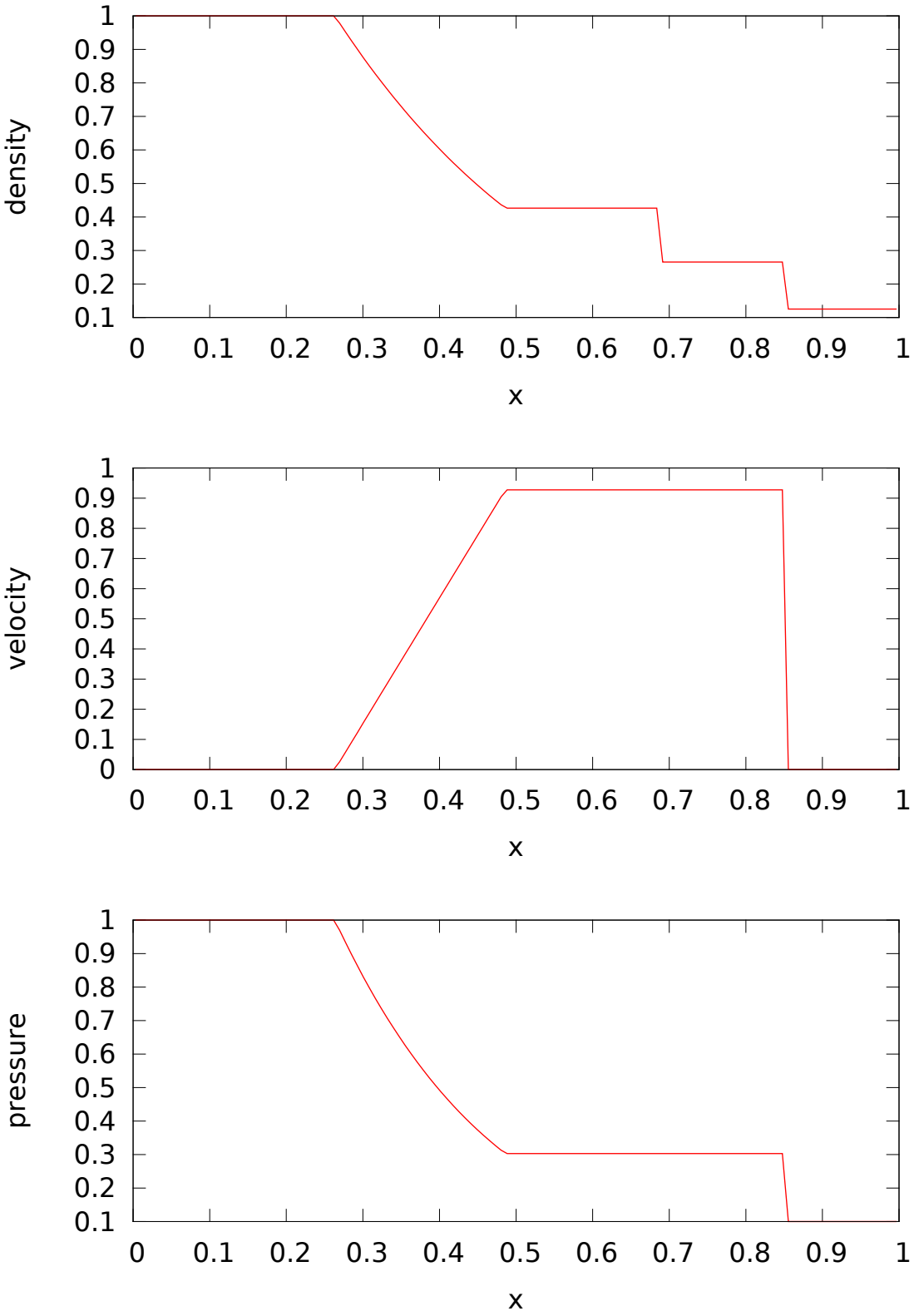

Figure 1: Evolution following from an initial discontinuity at $x = 0.5$. These particular conditions are called the *Sod problem*, and in general, a setup with two states separated by a discontinuity is called a shock-tube problem. Here we see the three waves propagating away from the initial discontinuity. The left $(u - c)$ wave is a rarefaction, the middle (u) is the contact discontinuity, and the right $(u + c)$ is a shock. Note that all 3 primitive variables jump across the left and right waves, but only the density jumps across the middle wave. This reflects the right eigenvectors.

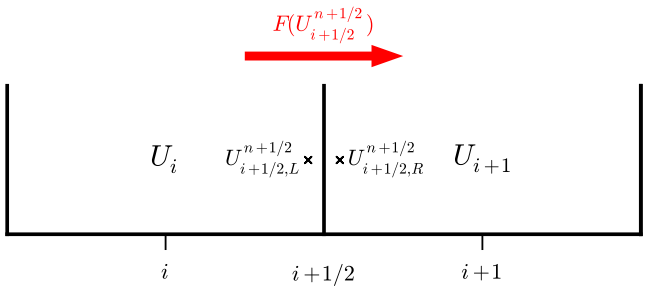

Figure 2: The left and right states at interface $i + 1/2$. The arrow indicates the flux through the interface, as computed by the Riemann solver using these states as input.

This says that each of the conserved quantities in U change only due to the flux of that quantity through the boundary of the cell.

Instead of approximating the flux itself on the interface, we find an approximation to the state on the interface, $U^{n+1/2}_{i-1/2}$ and $U^{n+1/2}_{i+1/2}$ and use this with the flux function to define the flux through the interface:

$$F^{n+1/2}_{i-1/2} = F(U^{n+1/2}_{i-1/2}) \tag{15}$$

$$F^{n+1/2}_{i+1/2} = F(U^{n+1/2}_{i+1/2}) \tag{16}$$

To find this interface state, we predict left and right states at each interface (centered in time), which are the input to the Riemann solver. The Riemann solver will then look at the characteristic wave structure and determine the fluid state on the interface, which is then used to compute the flux. This is illustrated in Figure 2. The fluxes allow us to update the state in time as:

$$U^{n+1}_i = U^n_i + \frac{\Delta t}{\Delta x}\left(F^{n+1/2}_{i-1/2} - F^{n+1/2}_{i+1/2}\right) \tag{17}$$

Finally, although we use the conserved variables for the final update, in constructing the interface states it is often easier to work with the primitive variables. These have a simpler characteristic structure. The interface states in terms of the primitive variables can be converted into the interface states of the conserved variables through a simple algebraic transformation.

Constructing these interface states requires reconstructing the cell-average data with a piecewise constant, linear, or parabolic polynomial and doing characteristic tracing to see how much of each characteristic quantity comes to the interface over $\Delta t/2$. Since we are comfortable working with the primitive variables, the reconstruction is done on those, and they are then projected into the characteristic variables (using the left- and right-eigenvectors) to determine how much of each characteristic quantity is carried by each of the 3 waves. We look at several methods below.

2.1 Piecewise constant

The simplest possible reconstruction of the data is piecewise constant. This is what was done in the original Godunov method. For the interface marked by $i + 1/2$, the left and right states on the interface are simply:

$$U_{i+1/2,L} = U_i \tag{18}$$

$$U_{i+1/2,R} = U_{i+1} \tag{19}$$

This does not take into account in any way how the state U may be changing through the cell. As a result, it is first-order accurate in space, and since no attempt was made to center it in time, it is first-order accurate in time.

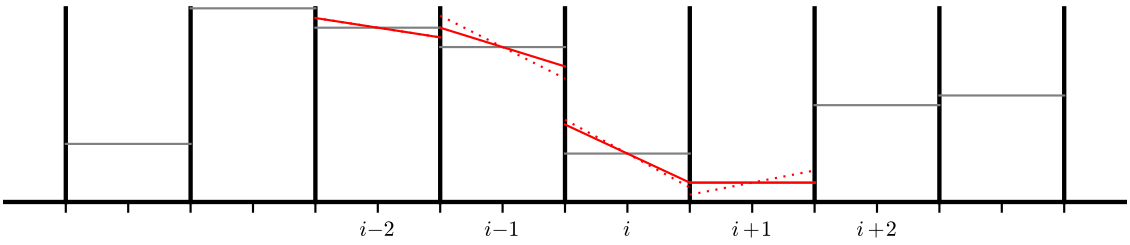

Figure 3: Piecewise linear reconstruction of the cell averages. The dotted line shows the unlimited center-difference slopes and the solid line shows the limited slopes.

2.2 Piecewise linear

For higher-order reconstruction, we first convert from the conserved variables, U, to the primitive variables, q. These have a simpler characteristic structure, making them easier to work with. Here we consider piecewise linear reconstruction—the cell average data is approximated by a line with non-zero slope within each cell. Figure 3 shows the piecewise linear reconstruction of some data.

Consider constructing the left state at the interface $i + 1/2$ (see Figure 2). Just like for the advection equation, we do a Taylor expansion through $\Delta x/2$ to bring us to the interface, and $\Delta t/2$ to bring us to the midpoint in time. Starting with q_i, the cell-centered primitive variable, expanding to the right interface (to create the left state there) gives:

$$q^{n+1/2}_{i+1/2,L} = q^n_i + \frac{\Delta x}{2}\frac{\partial q}{\partial x}\bigg|_i + \frac{\Delta t}{2}\underbrace{\frac{\partial q}{\partial t}\bigg|_i}_{=-A\partial q/\partial x} + \ldots \tag{20}$$

$$= q^n_i + \frac{\Delta x}{2}\frac{\partial q}{\partial x}\bigg|_i - \frac{\Delta t}{2}\left(A\frac{\partial q}{\partial x}\right)_i \tag{21}$$

$$= q^n_i + \frac{1}{2}\left[1 - \frac{\Delta t}{\Delta x}A_i\right]\Delta q_i \tag{22}$$

where Δq_i is the reconstructed slope of the primitive variable in that cell (similar to how we compute it for the advection equation). We note that the terms truncated in the first line are $O(\Delta x^2)$ and $O(\Delta t^2)$, so our method will be second-order accurate in space and time.

As with the advection equation, we limit the slope such that no new minima or maxima are introduced. Any of the slope limiters used for linear advection apply here as well. We represent the limited slope as $\overline{\Delta q}_i$.

We can decompose $A\Delta q$ in terms of the left and right eigenvectors and sum over all the waves that move *toward* the interface. First, we recognize that $A = R\Lambda L$ and recognizing that the '1' in Eq. 22 is the identity, $I = LR$, we rewrite this expression as:

$$q^{n+1/2}_{i+1/2,L} = q^n_i + \frac{1}{2}\left[RL - \frac{\Delta t}{\Delta x}R\Lambda L\right]_i \overline{\Delta q}_i \tag{23}$$

We see the common factor of $L\Delta q$. We now write this back in component form. Consider:

$$R\Lambda L\overline{\Delta q} = \left(\begin{array}{c:c:c} r_1^{(-)} & r_1^{(\circ)} & r_1^{(+)} \\ r_2^{(-)} & r_2^{(\circ)} & r_2^{(+)} \\ r_3^{(-)} & r_3^{(\circ)} & r_3^{(+)} \end{array}\right) \left(\begin{array}{ccc} \lambda^{(-)} & & \\ & \lambda^{(\circ)} & \\ & & \lambda^{(+)} \end{array}\right) \left(\begin{array}{ccc} l_1^{(-)} & l_2^{(-)} & l_3^{(-)} \\ \hdashline l_1^{(\circ)} & l_2^{(\circ)} & l_3^{(\circ)} \\ \hdashline l_1^{(+)} & l_2^{(+)} & l_3^{(+)} \end{array}\right) \left(\begin{array}{c} \overline{\Delta\rho} \\ \overline{\Delta u} \\ \overline{\Delta p} \end{array}\right) \tag{24}$$

Starting with $L\overline{\Delta q}$, which is a vector with each component the dot-product of a left eigenvalue with $\overline{\Delta q}$, we have

$$R\Lambda L\overline{\Delta q} = \left(\begin{array}{c|c|c} r_1^{(-)} & r_1^{(\circ)} & r_1^{(+)} \\ r_2^{(-)} & r_2^{(\circ)} & r_2^{(+)} \\ r_3^{(-)} & r_3^{(\circ)} & r_3^{(+)} \end{array}\right) \left(\begin{array}{ccc} \lambda^{(-)} & & \\ & \lambda^{(\circ)} & \\ & & \lambda^{(+)} \end{array}\right) \left(\begin{array}{c} l^{(-)} \cdot \overline{\Delta q} \\ l^{(\circ)} \cdot \overline{\Delta q} \\ l^{(+)} \cdot \overline{\Delta q} \end{array}\right) \tag{25}$$

Next we see that multiplying this vector by Λ simply puts the eigenvalue with its respective eigenvector in the resulting column vector:

$$R\Lambda L\overline{\Delta q} = \left(\begin{array}{c|c|c} r_1^{(-)} & r_1^{(\circ)} & r_1^{(+)} \\ r_2^{(-)} & r_2^{(\circ)} & r_2^{(+)} \\ r_3^{(-)} & r_3^{(\circ)} & r_3^{(+)} \end{array}\right) \left(\begin{array}{c} \lambda^{(-)}\, l^{(-)} \cdot \overline{\Delta q} \\ \lambda^{(\circ)}\, l^{(\circ)} \cdot \overline{\Delta q} \\ \lambda^{(+)}\, l^{(+)} \cdot \overline{\Delta q} \end{array}\right) \tag{26}$$

Finally, the last multiply results in a column vector:

$$R\Lambda L\overline{\Delta q} = \left(\begin{array}{c} r_1^{(-)}\, \lambda^{(-)}\, l^{(-)} \cdot \overline{\Delta q} + r_1^{(\circ)}\, \lambda^{(\circ)}\, l^{(\circ)} \cdot \overline{\Delta q} + r_1^{(+)}\, \lambda^{(+)}\, l^{(+)} \cdot \overline{\Delta q} \\ r_2^{(-)}\, \lambda^{(-)}\, l^{(-)} \cdot \overline{\Delta q} + r_2^{(\circ)}\, \lambda^{(\circ)}\, l^{(\circ)} \cdot \overline{\Delta q} + r_2^{(+)}\, \lambda^{(+)}\, l^{(+)} \cdot \overline{\Delta q} \\ r_3^{(-)}\, \lambda^{(-)}\, l^{(-)} \cdot \overline{\Delta q} + r_3^{(\circ)}\, \lambda^{(\circ)}\, l^{(\circ)} \cdot \overline{\Delta q} + r_3^{(+)}\, \lambda^{(+)}\, l^{(+)} \cdot \overline{\Delta q} \end{array}\right) \tag{27}$$

We can rewrite this compactly as:

$$\sum_\nu \lambda^{(\nu)} (l^{(\nu)} \cdot \overline{\Delta q}) r^{(\nu)} \tag{28}$$

where we use ν to indicate which wave we are summing over. A similar expansion is used for $RL\overline{\Delta q}$. The resulting vector for the left state is:

$$q_{i+1/2,L}^{n+1/2} = q_i^n + \frac{1}{2} \sum_{\nu;\lambda^{(\nu)} \geq 0} \left[1 - \frac{\Delta t}{\Delta x} \lambda_i^{(\nu)}\right] (l_i^{(\nu)} \cdot \overline{\Delta q}_i) r_i^{(\nu)} \tag{29}$$

Note that we make a slight change here, and only include a term in the sum if its wave is moving toward the interface ($\lambda^{(\nu)} \geq 0$).

Starting with the data in the $i+1$ zone and expanding to the left, we can find the right state on the $i+1/2$ interface:

$$q_{i+1/2,R}^{n+1/2} = q_{i+1}^n - \frac{1}{2} \sum_{\nu;\lambda^{(\nu)} \leq 0} \left[1 + \frac{\Delta t}{\Delta x} \lambda_{i+1}^{(\nu)}\right] (l_{i+1}^{(\nu)} \cdot \overline{\Delta q}_{i+1}) r_{i+1}^{(\nu)} \tag{30}$$

A good discussion of this is in Miller & Colella [8] (Eq. 85). This expression is saying that each wave carries a jump in $r^{(\nu)}$ and only those jumps moving toward the interface contribute to our interface state. This restriction of only summing up the waves moving toward the interface is sometimes called *characteristic tracing*. This decomposition in terms of the eigenvectors and eigenvalues is commonly called a *characteristic projection*. In terms of an operator, P, it can be expressed as:

$$P\chi = \sum_\nu (l^{(\nu)} . \chi) r^{(\nu)} \tag{31}$$

Exercise 5: Show that $Pq = q$, using the eigenvectors corresponding to the primitive variable form of the Euler equations.

In the literature, sometimes a '$>$' or '$<$' subscript on P is used to indicate the characteristic tracing.

We could stop here, but Colella & Glaz [4] (p. 278) argue that the act of decomposing A in terms of the left and right eigenvectors is a linearization of the quasi-linear system, and we should minimize the

size of the quantities that are subjected to this characteristic projection. To accomplish this, they suggest subtracting off a *reference state*. Saltzman (Eq. 8) further argues that since only jumps in the solution are used in constructing the interface state, and that the characteristic decomposition simply adds up all these jumps, we can subtract off the reference state and project the result. In other words, we can write:

$$q^{n+1/2}_{i+1/2,L} - q_{\rm ref} = q^n_i - q_{\rm ref} + \frac{1}{2}\left[1 - \frac{\Delta t}{\Delta x} A_i\right] \overline{\Delta q}_i \tag{32}$$

Then we subject the RHS to the characteristic projection—this tells us how much of the quantity $q^{n+1/2}_{i+1/2,L} - q_{\rm ref}$ reaches the interface. Colella & Glaz (p. 278) and Colella (Eq. 2.11) suggest

$$q_{\rm ref} = \tilde{q}_{i,L} \equiv q_i + \frac{1}{2}\left[1 - \frac{\Delta t}{\Delta x} \max(\lambda^{(+)}_i, 0)\right] \overline{\Delta q}_i \tag{33}$$

where $\lambda^{(+)}$ is the fastest eigenvalue, and thus will see the largest portion of the linear profiles. Physically, this reference state represents the jump carried by the fastest wave moving toward the interface. Then,

$$q^{n+1/2}_{i+1/2,L} - \tilde{q}_{i,L} = \frac{1}{2}\frac{\Delta t}{\Delta x}\left[\max(\lambda^{(+)}_i, 0) - A_i\right] \overline{\Delta q}_i \tag{34}$$

and projecting this RHS (see Colella & Glaz Eq. 43; Miller & Colella Eq. 87), and isolating the interface state, we have

$$q^{n+1/2}_{i+1/2,L} = \tilde{q}_{i,L} + \frac{1}{2}\frac{\Delta t}{\Delta x} \sum_{\nu;\lambda^{(\nu)}\geq 0} l^{(\nu)}_i \cdot \left[\max(\lambda^{(+)}_i, 0) - A_i\right] \overline{\Delta q}_i \, r^{(\nu)}_i \tag{35}$$

$$= \tilde{q}_{i,L} + \frac{1}{2}\frac{\Delta t}{\Delta x} \sum_{\nu;\lambda^{(\nu)}\geq 0} \left[\max(\lambda^{(+)}_i, 0) - \lambda^{(\nu)}_i\right] (l^{(\nu)}_i \cdot \overline{\Delta q}_i) \, r^{(\nu)}_i \tag{36}$$

This is equivalent to the expression in Saltzman [10] (p. 161, first column, second-to-last equation) and Colella [3] (p. 191, the group of expressions at the end). The corresponding state to the right of this interface is:

$$q^{n+1/2}_{i+1/2,R} = \tilde{q}_{i+1,R} + \frac{1}{2}\frac{\Delta t}{\Delta x} \sum_{\nu;\lambda^{(\nu)}\leq 0} \left[\min(\lambda^{(-)}_{i+1}, 0) - \lambda^{(\nu)}_{i+1}\right] (l^{(\nu)}_{i+1} \cdot \overline{\Delta q}_{i+1}) \, r^{(\nu)}_{i+1} \tag{37}$$

where now the reference state captures the flow from the $i+1$ zone moving to the *left* to this interface (hence the appearance of $\lambda^{(-)}$, the leftmost eigenvalue):

$$\tilde{q}_{i+1,R} = q_{i+1} - \frac{1}{2}\left[1 + \frac{\Delta t}{\Delta x} \min(\lambda^{(-)}_{i+1}, 0)\right] \overline{\Delta q}_{i+1} \tag{38}$$

Side note: the data in zone i will be used to construct the right state at $i - 1/2$ (the left interface) and the left state at $i + 1/2$ (the right interface) (see Figure 4). For this reason, codes usually compute the eigenvectors/eigenvalues for that zone and then compute $q^{n+1/2}_{i-1/2,R}$ together with $q^{n+1/2}_{i+1/2,L}$ in a loop over the zone centers.

2.3 Piecewise parabolic

The piecewise parabolic method uses a parabolic reconstruction in each cell. This is more accurate than the linear reconstruction. Figure 5 shows the reconstructed parabolic profiles within a few cells. Since the original PPM paper [6], there have been many discussions of the method, with many variations. Here we focus on the presentation by Miller & Colella [8], since that is the most straightforward. Note: even

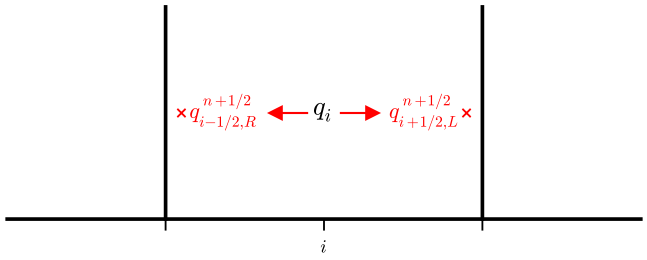

Figure 4: The two interface states that are constructed using q_i as the starting point.

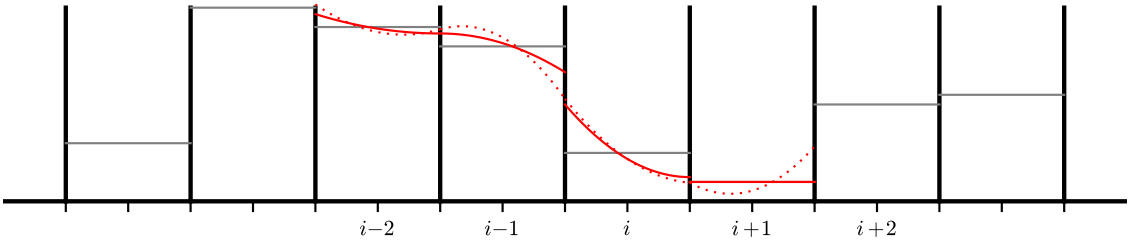

Figure 5: Piecewise parabolic reconstruction of the cell averages. The dotted line shows the unlimited parabolas—note how they touch at each interface, since the interface values come from the same interpolant initially. The solid line shows the limited parabolas.

though a parabolic profile could be third-order accurate, the temporal discretization and prediction in this method is still only second-order.

Miller & Colella give an excellent description of how to take the results for piecewise linear reconstruction and generalize it to the case of PPM [6] (see Eqs. 88-90). Starting with Eq. 32, we can write this (after the characteristic projection) as

$$q^{n+1/2}_{i+1/2,L} = \tilde{q}_+ - \sum_{\nu;\lambda^{(\nu)}\geq 0} l_i^{(\nu)} \cdot \left\{ \tilde{q}_+ - \left[q_i^n + \frac{1}{2}\left(1 - \frac{\Delta t}{\Delta x}\lambda_i^{(\nu)}\right)\overline{\Delta q}_i \right] \right\} r_i^{(\nu)} \tag{39}$$

Miller & Colella rewrite the portion inside the $[\ldots]$ recognizing that (similar to M&C Eq. 88, but for the $i+1/2, L$ interface):

$$q_i^n + \frac{1}{2}\left(1 - \frac{\Delta t}{\Delta x}\lambda_i^{(\nu)}\right)\overline{\Delta q}_i \approx \frac{1}{\lambda \Delta t}\int_{x_{i+1/2}-\lambda\Delta t}^{x_{i+1/2}} q(x)dx \tag{40}$$

where $q(x)$ is the reconstructed functional form of q in the zone.

> *Exercise 6: Show that this is exactly true for a linear reconstruction of $q(x)$, i.e., $q(x) = q_i + (\partial q/\partial x)(x - x_i)$.*

The integral on the right represents the average of q that can reach the right interface of the cell i over timestep Δt, moving at the wavespeed λ. This suggests that we can replace the linear reconstruction of q with a parabolic one, and keep our expressions for the interface states.

In particular, we define

$$\mathcal{I}_+^{(\nu)}(q_i) = \frac{1}{\sigma^{(\nu)}\Delta x}\int_{x_{i+1/2}-\sigma^{(\nu)}\Delta x}^{x_{i+1/2}} q(x)dx \tag{41}$$

with $\sigma^{(\nu)} = |\lambda^{(\nu)}|\Delta t/\Delta x$ (see Almgren et al. Eq. 31) (see Figure 6). Then

$$q^{n+1/2}_{i+1/2,L} = \tilde{q}_+ - \sum_{\nu;\lambda^{(\nu)}\geq 0} l_i^{(\nu)} \cdot \left(\tilde{q}_+ - \mathcal{I}_+^{(\nu)}(q_i) \right) r_i^{(\nu)} \tag{42}$$

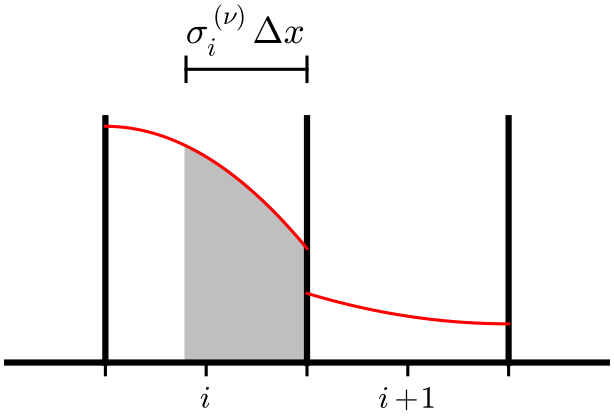

Figure 6: Integration under the parabolic profile. For each of the waves, σ is the fraction of the cell that they cross in a timestep, and $\sigma\Delta x = \lambda\Delta t$ is the distance they can travel. Here we are integrating under the parabola to the right interface of cell i to define $\mathcal{I}_+$ (this is indicated by the shaded region). The $\mathcal{I}_+$ carried by this wave will be added to those carried by the other waves to form the left state at interface $i + 1/2$.

Miller & Colella choose the reference state as

$$\tilde{q}_+ = \begin{cases} \mathcal{I}_+^{(+)}(q_i) & \text{if } u + c > 0 \\ q_i & \text{otherwise} \end{cases} \tag{43}$$

where the superscript $(+)$ on $\mathcal{I}$ indicates that the fastest eigenvalue ($\lambda^{(+)} = u + c$) is used. This is similar in spirit to Eq. 33. Note: in the original PPM paper, if the wave is not approaching the interface, instead of using the cell-average, q_i, they use the limit of the quadratic interpolant. In contrast to the above, the Castro paper [1] just uses q_i for the reference state regardless of whether the wave is moving toward or away from the interface. Note that if the system were linear, then the choice of reference state would not matter.

To finish the reconstruction, we need to know the parabolic form of $q(x)$. Here, we do the reconstruction from the original PPM paper:

$$q(x) = q_- + \xi(x)\left(\Delta q + q_6(1 - \xi(x))\right) \tag{44}$$

with $\Delta q = q_+ - q_-$, and q_-, q_+ the values of the polynomial on the left and right edges, respectively, of the current cell, and

$$q_6 \equiv 6\left[q_i - \frac{1}{2}(q_- + q_+)\right] \tag{45}$$

and

$$\xi(x) = \frac{x - x_{i-1/2}}{\Delta x} \tag{46}$$

To complete the description, we need to determine the parameters of the parabola. The values of q_- and q_+ are computed and limited as described in the original PPM paper. With this definition, we can do the integral $\mathcal{I}_+$:

$$\mathcal{I}_+^{(\nu)}(q_i) = q_{+,i} - \frac{\sigma_i^{(\nu)}}{2}\left[\Delta q_i - q_{6,i}\left(1 - \frac{2}{3}\sigma_i^{(\nu)}\right)\right] \tag{47}$$

Figure 6 illustrates the process of integrating under the parabolic profile.

Exercise 7: Show that $q(x)$ is a conservative interpolant. That is

$$\frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_{i+1/2}} q(x)dx = q_i \tag{48}$$

You can also see that the average over the left half of the zone is $q_i - \frac{1}{4}\Delta q$ and the average over the right half of the zone is $q_i + \frac{1}{4}\Delta q$. This means that there are equal areas between the integral and zone average on the left and right sides of the zone. This can be seen by looking at Figure 5.

Aside: Note that this characteristic projection of $\tilde{q}_+ - \mathcal{I}_+^{(\nu)}$ is discussed in the original PPM paper in the paragraph following Eq. 3.5. They do not keep things in this form however, and instead explicitly multiply out the $l \cdot [\ldots] r$ terms to arrive at Eq. 3.6. For example, starting with Eq. 42, we can write the left velocity state as (leaving off the i subscripts on the vectors):

$$u_{i+1/2,L}^{n+1/2} = \tilde{u}_+ - \sum_\nu l^{(\nu)} \cdot (\tilde{q}_+ - \mathcal{I}_+^{(\nu)}(q)) \underbrace{r^{(\nu)}}_{\substack{\text{only the} \\ u \text{ 'slot'}}} \tag{49}$$

(where, as above, the $\sim$ indicates the reference state). Here the r eigenvector on the end is representative—we only pick the row corresponding to u in the q vector (in our case, the second row).

Putting in the eigenvectors and writing out the sum, we have:

$$\begin{aligned} u_{i+1/2,L}^{n+1/2} = \tilde{u}_+ &- \begin{pmatrix} 0 & -\frac{\rho}{2c} & \frac{1}{2c^2} \end{pmatrix} \begin{pmatrix} \tilde{\rho}_+ - \mathcal{I}_+^{(-)}(\rho) \\ \tilde{u}_+ - \mathcal{I}_+^{(-)}(u) \\ \tilde{p}_+ - \mathcal{I}_+^{(-)}(p) \end{pmatrix} \begin{pmatrix} 1 \\ -c/\rho \\ c^2 \end{pmatrix} \\ &- \begin{pmatrix} 1 & 0 & -\frac{1}{c^2} \end{pmatrix} \begin{pmatrix} \tilde{\rho}_+ - \mathcal{I}_+^{(\circ)}(\rho) \\ \tilde{u}_+ - \mathcal{I}_+^{(\circ)}(u) \\ \tilde{p}_+ - \mathcal{I}_+^{(\circ)}(p) \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \\ &- \begin{pmatrix} 0 & \frac{\rho}{2c} & \frac{1}{2c^2} \end{pmatrix} \begin{pmatrix} \tilde{\rho}_+ - \mathcal{I}_+^{(+)}(\rho) \\ \tilde{u}_+ - \mathcal{I}_+^{(+)}(u) \\ \tilde{p}_+ - \mathcal{I}_+^{(+)}(p) \end{pmatrix} \begin{pmatrix} 1 \\ c/\rho \\ c^2 \end{pmatrix} \end{aligned} \tag{50}$$

Here again we show the entire right eigenvector for illustration, but only the element that comes into play is drawn in black. This shows that the second term is 0—the contact wave does not carry a jump in velocity. Multiplying out $l^{(\nu)} \cdot (\tilde{q}_+ - \mathcal{I}_+^{(\nu)})$ we have:

$$u_{i+1/2,L}^{n+1/2} = \tilde{u}_+ - \frac{1}{2}\left[(\tilde{u}_+ - \mathcal{I}_+^{(-)}(u)) - \frac{\tilde{p}_+ - \mathcal{I}_+^{(-)}(p)}{C}\right] - \frac{1}{2}\left[(\tilde{u}_+ - \mathcal{I}_+^{(+)}(u)) + \frac{\tilde{p}_+ - \mathcal{I}_+^{(+)}(p)}{C}\right] \tag{51}$$

where C is the Lagrangian sound speed ($C = \sqrt{\gamma p \rho}$). Defining

$$\beta^+ = -\frac{1}{2C}\left[(\tilde{u}_+ - \mathcal{I}_+^{(+)}(u)) + \frac{\tilde{p}_+ - \mathcal{I}_+^{(+)}(p)}{C}\right] \tag{52}$$

$$\beta^- = +\frac{1}{2C}\left[(\tilde{u}_+ - \mathcal{I}_+^{(-)}(u)) - \frac{\tilde{p}_+ - \mathcal{I}_+^{(-)}(p)}{C}\right] \tag{53}$$

we can write our left state as:

$$u_{i+1/2,L}^{n+1/2} = \tilde{u}_+ + C(\beta^+ - \beta^-) \tag{54}$$

This is Eqs. 3.6 and 3.7 in the PPM paper. Note that in their construction appears to use the reference state in defining the Lagrangian sound speed (in their β expressions is written

as $\tilde{C}$). This may follow from the comment before Eq. 3.6, "modified slightly for the present application". Similarly, the expressions for ρ_L and p_L can be written out.

Similar expressions can be derived for the right state at the left interface of the zone ($q^{n+1/2}_{i-1/2,R}$). Here, the integral under the parabolic reconstruction is done over the region of each wave that can reach the left interface over our timestep:

$$\mathcal{I}^{(\nu)}_-(q) = \frac{1}{\sigma^{(\nu)}\Delta x}\int_{x_{i-1/2}}^{x_{i-1/2}+\sigma^{(\nu)}\Delta x} q(x)dx \tag{55}$$

The right state at $i-1/2$ using zone i data is:

$$q^{n+1/2}_{i-1/2,R} = \tilde{q}_- - \sum_{\nu;\lambda_\nu \le 0} l^{(\nu)}_i \cdot \left(\tilde{q}_- - \mathcal{I}^{(\nu)}_-(q_i)\right) r^{(\nu)}_i \tag{56}$$

where the reference state is now:

$$\tilde{q}_- = \begin{cases} \mathcal{I}^{(-)}_-(q_i) & \text{if } u-c<0 \\ q_i & \text{otherwise} \end{cases} \tag{57}$$

where the $(-)$ superscript on $\mathcal{I}$ indicates that the most negative eigenvalue ($\lambda^- = u-c$) is used. The integral $\mathcal{I}^{(\nu)}_-(q)$ can be computed analytically by substituting in the parabolic interpolant, giving:

$$\mathcal{I}^{(\nu)}_-(q_i) = q_{-,i} + \frac{\sigma^{(\nu)}_i}{2}\left[\Delta q_i + q_{6,i}\left(1-\frac{2}{3}\sigma^{(\nu)}_i\right)\right] \tag{58}$$

This is equivalent to Eq. 31b in the Castro paper.

3 The Riemann problem

Once the interface states are created, the Riemann solver is called. This returns the solution at the interface:

$$q^{n+1/2}_{i+1/2} = \mathcal{R}(q^{n+1/2}_{i+1/2,L}, q^{n+1/2}_{i+1/2,R}) \tag{59}$$

Solving the Riemann problem for the Euler equations can be a complex operation, but the general ideas are straightforward. Here we review the basic outline of operations, and refer to Toro [12] for full details on a variety of methods for solving the Riemann problem.

The Riemann problem consists of a left and right state separated by an interface. For the Euler equations, there are three eigenvalues, which are the speeds at which information propagates. Each of these correspond to a wave that will move out from the interface with time, and each wave will carry with it a jump in the characteristic variables. The figure below shows the three waves moving out from the interface, separating space into 4 regions, marked: L, L^*, R^*, and R. We typically work in terms of primitive variables. The states in the L and R regions are simply the left and right input states—the waves have not had time to reach here, so they are unmodified.

We are interested in the state at the interface. To determine this, we need to determine which region we are in. That requires an estimation of the wave speeds. Since these are nonlinear waves, we cannot in general just use the eigenvalues (although some approximate solvers do). Different Riemann solvers will have different approximations for finding the speeds of the left, center, and right wave. Note the figure shows only one possible configuration for the waves—they can all be on one side of the interface (for all supersonic waves), or the contact (the middle wave) can be on either side.

Once the wave speeds are known, we look at the sign of the speeds to determine which of the 4 regions is on the interface. In the 'star' region, only ρ jumps across the middle (contact) wave, the pressure and

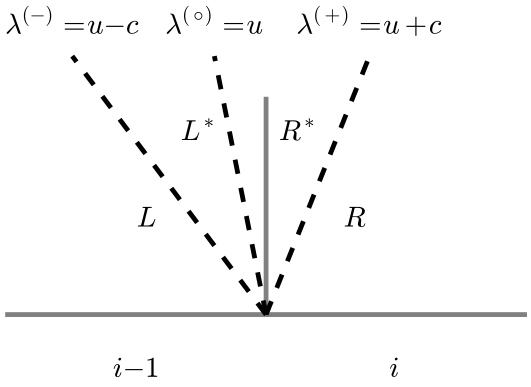

Figure 7: The wave structure and 4 distinct regions for the Riemann problem. Time can be thought of as the vertical axis here, so we see the waves moving outward from the interface.

velocity are constant across that wave (see $r^{(\circ)}$). We determine the state in the star region ($\rho_l^*, \rho_r^*, u^*, p^*$) by using the jump conditions for the Euler equations. In general, these differ depending on whether the waves are shocks or rarefactions. In practice, approximate Riemann solvers often assume one or the other (for example, the two-shock Riemann solver used in [4]). With the wave speeds and the states known in each region, we can evaluate the state on the interface, $q_{i+1/2}^{n+1/2}$.

Recall that a rarefaction involves diverging flow—it spreads out with time. Special consideration needs to be taken if the rarefaction wave spans the interface (a *transonic rarefaction*). In this case, most Riemann solvers interpolate between the left or right state and the appropriate star state.

Then the fluxes are computed from this state as:

$$F_{i+1/2}^{n+1/2} = \begin{pmatrix} \rho_{i+1/2}^{n+1/2} u_{i+1/2}^{n+1/2} \\ \rho_{i+1/2}^{n+1/2} (u_{i+1/2}^{n+1/2})^2 + p_{i+1/2}^{n+1/2} \\ u_{i+1/2}^{n+1/2} p_{i+1/2}^{n+1/2} / (\gamma - 1) + \frac{1}{2} \rho_{i+1/2}^{n+1/2} (u_{i+1/2}^{n+1/2})^3 + u_{i+1/2}^{n+1/2} p_{i+1/2}^{n+1/2} \end{pmatrix} \tag{60}$$

Note that instead of returning an approximate state at the interface, some Riemann solvers (e.g. the HLL(C) solvers) instead approximate the fluxes directly.

4 Conservative update

Once we have the fluxes, the conservative update is done as

$$U_i^{n+1} = U_i^n + \frac{\Delta t}{\Delta x} \left(F_{i-1/2}^{n+1/2} - F_{i+1/2}^{n+1/2} \right) \tag{61}$$

The timestep, Δt is determined by the time it takes for the fastest wave to cross a single zone:

$$\Delta t < \max_i \frac{\Delta x}{|u_i| + c} \tag{62}$$

5 Multidimensional problems

The multidimensional case is very similar to the multidimensional advection problem. Our system of equations is now:

$$U_t + [F^{(x)}(U)]_x + [F^{(y)}(U)]_y = 0 \tag{63}$$

with

$$U = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix} \qquad F^{(x)}(U) = \begin{pmatrix} \rho u \\ \rho uu + p \\ \rho vu \\ \rho uE + up \end{pmatrix} \qquad F^{(y)}(U) = \begin{pmatrix} \rho v \\ \rho vu \\ \rho vv + p \\ \rho vE + vp \end{pmatrix} \tag{64}$$

For a directionally-unsplit discretization, we predict the cell-centered quantities to the edges by Taylor expanding the conservative state, U, in space and time. Now, when replacing the time derivative ($\partial U/\partial t$) with the divergence of the fluxes, we gain a transverse flux derivative term. For example, predicting to the upper x edge of zone i, j, we have:

$$\begin{aligned} U^{n+1/2}_{i+1/2,j,L} &= U^n_{i,j} + \frac{\Delta x}{2}\frac{\partial U}{\partial x} + \frac{\Delta t}{2}\frac{\partial U}{\partial t} + \dots & (65)\\ &= U^n_{i,j} + \frac{\Delta x}{2}\frac{\partial U}{\partial x} - \frac{\Delta t}{2}\frac{\partial F^{(x)}}{\partial x} - \frac{\Delta t}{2}\frac{\partial F^{(y)}}{\partial y} & (66)\\ &= U^n_{i,j} + \frac{1}{2}\left[1 - \frac{\Delta t}{\Delta x}A^{(x)}(U)\right]\Delta U - \frac{\Delta t}{2}\frac{\partial F^{(y)}}{\partial y} & (67) \end{aligned}$$

where $A^{(x)}(U) \equiv \partial F^{(x)}/\partial U$. We decompose this into a *normal state* and a *transverse flux difference*. Adopting the notation from Colella (1990), we use $\hat{U}$ to denote the normal state:

$$\begin{aligned} \hat{U}^{n+1/2}_{i+1/2,j,L} &\equiv U^n_{i,j} + \frac{1}{2}\left[1 - \frac{\Delta t}{\Delta x}A^{(x)}(U)\right]\Delta U & (68)\\ U^{n+1/2}_{i+1/2,j,L} &= \hat{U}^{n+1/2}_{i+1/2,j,L} - \frac{\Delta t}{2}\frac{\partial F^{(y)}}{\partial y} & (69) \end{aligned}$$

The primitive variable form for this system is

$$q_t + A^{(x)}(q)q_x + A^{(y)}(q)q_y = 0 \tag{70}$$

where

$$q = \begin{pmatrix} \rho \\ u \\ v \\ p \end{pmatrix} \qquad A^{(x)}(q) = \begin{pmatrix} u & \rho & 0 & 0 \\ 0 & u & 0 & 1/\rho \\ 0 & 0 & u & 0 \\ 0 & \gamma p & 0 & u \end{pmatrix} \qquad A^{(y)}(q) = \begin{pmatrix} v & 0 & \rho & 0 \\ 0 & v & 0 & 0 \\ 0 & 0 & v & 1/\rho \\ 0 & 0 & \gamma p & v \end{pmatrix} \tag{71}$$

There are now 4 eigenvalues. For $A^{(x)}(q)$, they are $u - c$, u, u, $u + c$. If we just look at the system for the x evolution, we see that the transverse velocity (in this case, v) just advects with velocity u, corresponding to the additional eigenvalue.

Exercise 8: Derive the form of $A^{(x)}(q)$ and $A^{(y)}(q)$ and find their left and right eigenvectors.

We note here that $\hat{U}^{n+1/2}_{i+1/2,j,L}$ is essentially one-dimensional, since only the x-fluxes are involved (through $A^{(x)}(U)$). This means that we can compute this term using the one-dimensional techniques developed in § 2. In particular, Colella (1990) suggest that we switch to primitive variables and compute this as:

$$\hat{U}^{n+1/2}_{i+1/2,j,L} = U(\hat{q}^{n+1/2}_{i+1/2,j,L}) \tag{72}$$

Similarly, we consider the system projected along the y-direction to define the normal states on the y-edges, again using the one-dimensional reconstruction on the primitive variables from § 2:

$$\hat{U}^{n+1/2}_{i,j+1/2,L} = U(\hat{q}^{n+1/2}_{i,j+1/2,L}) \tag{73}$$

To compute the full interface state (Eq. 69), we need to include the transverse term. Colella (1990) gives two different procedures for evaluating the transverse fluxes. The first is to simply use the cell-centered $U_{i,j}$ (Colella 1990, Eq. 2.13); the second is to use the reconstructed normal states (the $\hat{U}$'s) (Eq. 2.15). In both cases, we need to solve a *transverse Riemann problem* to find the true state on the transverse interface. This latter approach is what we prefer. In particular, for computing the full x-interface left state, $U^{n+1/2}_{i+1/2,j,L}$, we need the transverse (y) states, which we define as

$$U^T_{i,j+1/2} = \mathcal{R}(\hat{U}^{n+1/2}_{i,j+1/2,L}, \hat{U}^{n+1/2}_{i,j+1/2,R}) \tag{74}$$

$$U^T_{i,j-1/2} = \mathcal{R}(\hat{U}^{n+1/2}_{i,j-1/2,L}, \hat{U}^{n+1/2}_{i,j-1/2,R}) \tag{75}$$

Taken together, the full interface state is now:

$$U^{n+1/2}_{i+1/2,j,L} = U(\hat{q}^{n+1/2}_{i+1/2,j,L}) - \frac{\Delta t}{2}\frac{F^{(y)}(U^T_{i,j+1/2}) - F^{(y)}(U^T_{i,j-1/2})}{\Delta y} \tag{76}$$

The right state at the $i+1/2$ interface can be similarly computed (starting with the data in zone $i+1, j$ and expanding to the left) as:

$$U^{n+1/2}_{i+1/2,j,R} = U(\hat{q}^{n+1/2}_{i+1/2,j,R}) - \frac{\Delta t}{2}\frac{F^{(y)}(U^T_{i+1,j+1/2}) - F^{(y)}(U^T_{i+1,j-1/2})}{\Delta y} \tag{77}$$

Note the indices on the transverse states—they are now to the right of the interface (since we are dealing with the right state).

We then find the x-interface state by solving the Riemann problem normal to our interface:

$$U^{n+1/2}_{i+1/2,j} = \mathcal{R}(U^{n+1/2}_{i+1/2,j,L}, U^{n+1/2}_{i+1/2,j,R}) \tag{78}$$

Therefore, construction of the interface states now requires two Riemann solves: a transverse and normal one. The fluxes are then evaluated as:

$$F^{(x),n+1/2}_{i+1/2,j} = F^{(x)}(U^{n+1/2}_{i+1/2,j}) \tag{79}$$

Note, for multi-dimensional problems, in the Riemann solver, the transverse velocities are simply selected based on the speed of the contact, giving either the left or right state.

The final conservative update is done as:

$$U^{n+1}_{i,j} = U^n_{i,j} + \frac{\Delta t}{\Delta x}\left(F^{(x),n+1/2}_{i-1/2,j} - F^{(x),n+1/2}_{i+1/2,j}\right) + \frac{\Delta t}{\Delta y}\left(F^{(y),n+1/2}_{i,j-1/2} - F^{(y),n+1/2}_{i,j+1/2}\right) \tag{80}$$

6 Boundary conditions

Boundary conditions are implemented through ghost cells. The following are the most commonly used boundary conditions. For the expressions below, we use the subscript lo to denote the spatial index of the first valid zone in the domain (just inside the left boundary).

- *Outflow*: the idea here is that the flow should gracefully leave the domain. The simplest form is to simply give all variables a zero-gradient:

$$\begin{pmatrix} \rho_{\mathrm{lo}-1,j} \\ (\rho u)_{\mathrm{lo}-1,j} \\ (\rho v)_{\mathrm{lo}-1,j} \\ (\rho E)_{\mathrm{lo}-1,j} \end{pmatrix} = \begin{pmatrix} \rho_{\mathrm{lo},j} \\ (\rho u)_{\mathrm{lo},j} \\ (\rho v)_{\mathrm{lo},j} \\ (\rho E)_{\mathrm{lo},j} \end{pmatrix} \tag{81}$$

  Note that these boundaries are not perfect. At the boundary, one (or more) of the waves from the Riemann problem can still enter the domain. Only for supersonic flow, do all waves point outward.

- *Reflect*: this is appropriate at a solid wall or symmetry plane. All variables are reflected across the boundary, with the normal velocity given the opposite sign. At the x-boundary, the first ghost cell is:

$$\begin{pmatrix} \rho_{\mathrm{lo}-1,j} \\ (\rho u)_{\mathrm{lo}-1,j} \\ (\rho v)_{\mathrm{lo}-1,j} \\ (\rho E)_{\mathrm{lo}-1,j} \end{pmatrix} = \begin{pmatrix} \rho_{\mathrm{lo},j} \\ -(\rho u)_{\mathrm{lo},j} \\ (\rho v)_{\mathrm{lo},j} \\ (\rho E)_{\mathrm{lo},j} \end{pmatrix} \tag{82}$$

  The next is:

$$\begin{pmatrix} \rho_{\mathrm{lo}-2,j} \\ (\rho u)_{\mathrm{lo}-2,j} \\ (\rho v)_{\mathrm{lo}-2,j} \\ (\rho E)_{\mathrm{lo}-2,j} \end{pmatrix} = \begin{pmatrix} \rho_{\mathrm{lo}+1,j} \\ -(\rho u)_{\mathrm{lo}+1,j} \\ (\rho v)_{\mathrm{lo}+1,j} \\ (\rho E)_{\mathrm{lo}+1,j} \end{pmatrix} \tag{83}$$

  and so on ...

- *Inflow*: inflow boundary conditions specify the state directly on the boundary. Technically, this state is on the boundary itself, not the cell-center. This can be accounted for by modifying the stencils used in the reconstruction near inflow boundaries.

- *Hydrostatic*: a hydrostatic boundary can be used at the base of an atmosphere to provide the pressure support necessary to hold up the atmosphere against gravity while still letting acoustic waves pass through. An example of this is described in [13].

7 Additional stuff

- *Flattening*: shocks are self-steepening (this is how we detect them in the Riemann solver—we look for converging characteristics). This can cause trouble with the methods here, because the shocks may become too steep.

  Flattening is a procedure to add additional dissipation at shocks, to ensure that they are smeared out over $\sim$ 2 zones. The flattening procedure is a multi-dimensional operation that looks at the pressure and velocity profiles and returns a coefficient, χ in $[0, 1]$ that multiplies the limited slopes. The convention most sources use is that $\chi = 1$ means no flattening (the slopes are unaltered), while $\chi = 0$ means complete flattening—the slopes are zeroed, dropping us to a first-order method.

  We use the flattening method described in Saltzman (1994).

  Note that the flattening algorithm increases the stencil size of piecewise-linear and piecewise-parabolic reconstruction to 4 ghost cells on each side. This is because the flattening procedure itself looks at the pressure 2 zones away, and we need to construct the flattening coefficient in both the first ghost cell (since we need the interface values there) and the second ghost cell (since the flattening procedure looks at the coefficients in its immediate upwinded neighbor).

- *Artificial viscosity*: Colella and Woodward argue discuss that behind slow-moving shocks these methods can have oscillations. The fix they propose is to use some artificial viscosity—this is additional dissipation that kicks in at shocks. (They argue that flattening alone is not enough).

  We use a multidimensional analog of their artificial viscosity ([6], Eq. 4.5) which modifies the fluxes. By design, it only kicks in for converging flows, such that you would find around a shock.

- *Contact steepening*: in contrast to shocks, contact waves do not steepen (they are associated with the middle characteristic wave, and the velocity does not change across that, meaning there cannot be any convergence). The original PPM paper advocates a contact steepening method to artificially steepen contact waves. While it shows good results in 1-d, it can be problematic in multi-dimensions.

Overall, the community seems split over whether this term should be used. Many people advocate that if you reach a situation where you think contact steepening may be necessary, it is more likely that the issue is that you do not have enough resolution.

- *Species*: for multifluid flows, the Euler equations are augments with continuity equations for each of the (chemical or nuclear) species:

$$\frac{\partial(\rho X_k)}{\partial t} + \frac{\partial(\rho X_k u)}{\partial x} = 0 \tag{84}$$

here, X_k are the mass fractions and obey $\sum_k X_k = 1$. Using the continuity equation, we can write this as an advection equation:

$$\frac{\partial X_k}{\partial t} + u\frac{\partial X_k}{\partial x} = 0 \tag{85}$$

When we now consider our primitive variables: $q = (\rho, u, p, X_k)$, we find

$$A(q) = \begin{pmatrix} u & \rho & 0 & 0 \\ 0 & u & 1/\rho & 0 \\ 0 & \gamma p & u & 0 \\ 0 & 0 & 0 & u \end{pmatrix} \tag{86}$$

There are now 4 eigenvalues, with the new one also being simply u. This says that the species simply advect with the flow. The right eigenvectors are now:

$$r^{(1)} = \begin{pmatrix} 1 \\ -c/\rho \\ c^2 \\ 0 \end{pmatrix} \qquad r^{(2)} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad r^{(3)} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \qquad r^{(4)} = \begin{pmatrix} 1 \\ c/\rho \\ c^2 \\ 0 \end{pmatrix} \tag{87}$$

corresponding to $\lambda^{(1)} = u - c$, $\lambda^{(2)} = u$, $\lambda^{(3)} = u$, and $\lambda^{(4)} = u + c$. We see that for the species, the only non-zero element is for one of the u eigenvectors. This means that X_k only jumps over this middle wave. In the Riemann solver then, there is no 'star' state for the species, it just jumps across the contact wave.

To add species into the solver, you simply need to reconstruct X_k as described above, find the interface values using this new $A(q)$ and associated eigenvectors, solve the Riemann problem, with X_k on the interface being simply the left or right state depending on the sign of the contact wave speed, and do the conservative update for ρX_k using the species flux.

One issue that can arise with species is that even if $\sum_k X_k = 1$ initially, after the update, that may no longer be true. There are a variety of ways to handle this:

  - You can update the species, (ρX_k) to the new time and then define the density to be $\rho = \sum_k (\rho X_k)$—this means that you are not relying on the value of the density from the mass continuity equation itself.
  - You can force the interface states of X_k to sum to 1. Because the limiting is non-linear, this is where problems can arise. If the interface values of X_k are forced to sum to 1 (by renormalizing), then the updated cell-centered value of X_k will as well. This is the approach discussed in [9].
  - You can design the limiting procedure to preserve the summation property. This approach is sometimes taken in the combustion field. For piecewise linear reconstruction, this can be obtained by computing the limited slopes of all the species, and taking the most restrictive slope and applying this same slope to all the species.

- *Source terms*: adding source terms is straightforward. For a system described by

$$U_t + [F^{(x)}(U)]_x + [F^{(y)}(U)]_y = H \tag{88}$$

we predict to the edges in the same fashion as described above, but now when we replace $\partial U/\partial t$ with the divergence of the fluxes, we also pick up the source term. This appears as:

$$\begin{aligned} U^{n+1/2}_{i+1/2,j,L} &= U^n_{i,j} + \frac{\Delta x}{2}\frac{\partial U}{\partial x} + \frac{\Delta t}{2}\frac{\partial U}{\partial t} + \ldots & (89)\\ &= U^n_{i,j} + \frac{\Delta x}{2}\frac{\partial U}{\partial x} - \frac{\Delta t}{2}\frac{\partial F^{(x)}}{\partial x} - \frac{\Delta t}{2}\frac{\partial F^{(y)}}{\partial y} + \frac{\Delta t}{2}H_{i,j} & (90)\\ &= U^n_{i,j} + \frac{1}{2}\left[1 - \frac{\Delta t}{\Delta x}A^{(x)}(U)\right]\Delta U - \frac{\Delta t}{2}\frac{\partial F^{(y)}}{\partial y} + \frac{\Delta t}{2}H_{i,j} & (91) \end{aligned}$$

We can compute things as above, but simply add the source term to the $\hat{U}$'s and carry it through.

Note that the source here is cell-centered. This expansion is second-order accurate. Nevertheless, some sources (e.g. the PPM paper) use edge-centered values for the gravitational source. Note also that in the original PPM paper, they include the source terms in the characteristic projection (see [6], Eq. 3.7, but note that there the full acceleration over the timestep is used while most sources have a factor of 1/2).

- *General equation of state*: The above methods were formulated with a constant gamma equation of state. A general equation of state (such as degenerate electrons) requires a more complex method. The classic prescription for extending this methodology is presented by Colella and Glaz [4]. They construct a thermodynamic index,

$$\gamma_e = \frac{p}{\rho e} + 1 \tag{92}$$

and derive an evolution equation for γ_e (C&G, Eq. 26). This evolution equation is used to predict γ_e to interfaces, and these interface values of γ_e are used in the Riemann solver presented there to find the fluxes through the interface. A different adiabatic index (they call Γ, others call γ_c) appears in the definition of the sound speed. They argue that this can be brought to interfaces in a piecewise constant fashion while still making the overall method second order, since γ_c does not explicitly appear in the fluxes (see the discussion at the top of page 277).

Extending this to an unsplit formulation is not obvious, since the evolution equation for γ_e is not a conservation law, and it is not obvious what sort of transverse term needs to be incorporated.

Alternately, the Castro paper [1] relies on an idea from an unpublished manuscript by Colella, Glaz, and Ferguson that predicts ρe to edges in addition to ρ, u, and p. Since ρe comes from a conservation-like equation, predicting it to the interface in the unsplit formulation is straightforward. This over-specifies the thermodynamics, but eliminates the need for γ_e.

With the addition of ρe, our system becomes:

$$q = \begin{pmatrix} \rho \\ u \\ p \\ \rho e \end{pmatrix} \qquad A = \begin{pmatrix} u & \rho & 0 & 0 \\ 0 & u & 1/\rho & 0 \\ 0 & \rho c^2 & u & 0 \\ 0 & \rho h & 0 & u \end{pmatrix} \tag{93}$$

where $h = e + p/\rho$ is the specific enthalpy. The eigenvalues of this system are:

$$\lambda^{(1)} = u - c \qquad \lambda^{(2)} = u \qquad \lambda^{(3)} = u \qquad \lambda^{(4)} = u + c \tag{94}$$

and the eigenvectors are:

$$r^{(1)} = \begin{pmatrix} 1 \\ -c/\rho \\ c^2 \\ h \end{pmatrix} \qquad r^{(2)} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad r^{(3)} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \qquad r^{(4)} = \begin{pmatrix} 1 \\ c/\rho \\ c^2 \\ h \end{pmatrix} \tag{95}$$

and

$$\begin{aligned} l^{(1)} &= \begin{pmatrix} 0 & -\frac{\rho}{2c} & \frac{1}{2c^2} & 0 \end{pmatrix} \\ l^{(2)} &= \begin{pmatrix} 1 & 0 & -\frac{1}{c^2} & 0 \end{pmatrix} \\ l^{(3)} &= \begin{pmatrix} 0 & 0 & -\frac{h}{c^2} & 1 \end{pmatrix} \\ l^{(4)} &= \begin{pmatrix} 0 & \frac{\rho}{2c} & \frac{1}{2c^2} & 0 \end{pmatrix} \end{aligned} \tag{96}$$

Remember that the state variables in the q vector are mixed into the other states by $l \cdot q$. Since all $l^{(\nu)}$'s have 0 in the ρe 'slot' (the last position) except for $l^{(3)}$, and the corresponding $r^{(3)}$ is only non-zero in the ρe slot, this means that ρe is not mixed into the other state variables. This is as expected, since ρe is not needed in the system.

Also recall that the jump carried by the wave ν is proportional to $r^{(\nu)}$—since $r^{(1)}$, $r^{(3)}$, and $r^{(4)}$ have non-zero ρe elements, this means that ρe jumps across these three waves.

Working through the sum for the (ρe) state, and using a $\sim$ to denote the reference states, we arrive at:

$$\begin{aligned} (\rho e)^{n+1/2}_{i+1/2,L} = \widetilde{(\rho e)} \quad &- \quad \frac{1}{2}\left[-\frac{\rho}{c}\left(\tilde{u} - \mathcal{I}^{(1)}_{+}(u)\right) + \frac{1}{c^2}\left(\tilde{p} - \mathcal{I}^{(1)}_{+}(p)\right)\right] h \\ &- \quad \left[-\frac{h}{c^2}\left(\tilde{p} - \mathcal{I}^{(3)}_{+}(p)\right) + \left(\widetilde{(\rho e)} - \mathcal{I}^{(3)}_{+}(\rho e)\right)\right] \\ &- \quad \frac{1}{2}\left[\frac{\rho}{c}\left(\tilde{u} - \mathcal{I}^{(4)}_{+}(u)\right) + \frac{1}{c^2}\left(\tilde{p} - \mathcal{I}^{(4)}_{+}(p)\right)\right] h \end{aligned} \tag{97}$$

This is the expression that is found in the Castro code.

All of these methods are designed to avoid EOS calls where possible, since general equations of state can be expensive.

- *Axisymmetry*: it is common to so 2-d axisymmetric models—here the r and z coordinates from a cylindrical geometry are modeled. This appears Cartesian, except there is a volume factor implicit in the divergence that must be accounted for. Our system in cylindrical coordinates is:

$$\frac{\partial U}{\partial t} + \frac{1}{r}\frac{\partial r F^{(r)}}{\partial r} + \frac{\partial F^{(z)}}{\partial z} = 0 \tag{98}$$

Expanding out the r derivative, we can write this as:

$$\frac{\partial U}{\partial t} + \frac{\partial F^{(r)}}{\partial r} + \frac{\partial F^{(z)}}{\partial z} = -\frac{F^{(r)}}{r} \tag{99}$$

This latter form is used when predicting the interface states, with the volume source that appears on the right treated as a source term to the interface states (as described above). Once the fluxes are computed, the final update uses the conservative form of the system, with the volume factors appearing now in the definition of the divergence.

- *Defining temperature*: although not needed for the pure Euler equations, it is sometimes desirable to define the temperature for source terms (like reactions) or complex equations of state. The temperature can typically be found from the equation of state given the internal energy:

$$e = E - \frac{1}{2}u^2 \tag{100}$$

$$T = T(e, \rho) \tag{101}$$

Trouble can arise when you are in a region of flow where the kinetic energy dominates (high Mach number flow). In this case, the e defined via subtraction can become negative due to truncation error in the evolution of u compared to E. In this instance, one must either impose a floor value for e or find an alternate method of deriving it.

In [2], an alternate formulation of the Euler equations is proposed. Both the total energy equation *and* the internal energy equation are evolved in each zone. When the flow is dominated by kinetic energy, then the internal energy from the internal energy evolution equation is used. The cost of this is conservation—the internal energy is not a conserved quantity, and switching to it introduces conservation of energy errors.

- *Limiting on characteristic variables*: some authors (see for example, [11] Eqs. 37, 38) advocate limiting on the characteristic variables rather than the primitive variables. The characteristic slopes for the quantity carried by the wave ν can be found from the primitive variables as:

$$\Delta w^{(\nu)} = l^{(\nu)} \cdot \Delta q \tag{102}$$

any limiting would then be done to $\Delta w^{(\nu)}$ and the limited primitive variables would be recovered as:

$$\overline{\Delta q} = \sum_\nu \overline{\Delta w}^{(\nu)} r^{(\nu)} \tag{103}$$

(here we use an overline to indicate limiting).

This is attractive because it is more in the spirit of the linear advection equation and the formalism that was developed there. A potential downside is that when you limit on the characteristic variables and convert back to the primitive, the primitive variables may now fall outside of valid physical ranges (for example, negative density).

- *New PPM limiters*: Recent work [5] has formulated improved limiters for PPM that do not clip the profiles at extrema. This only changes the limiting process in defining q_+ and q_i, and does not affect the subsequent parts of the algorithm.

- *3-d unsplit*: The extension of the unsplit methodology to 3-d is described by Saltzman [10]. The basic idea is the same as in 2-d, except now additional transverse Riemann solve are needed to fully couple in the corners.

References

[1] A. S. Almgren, V. E. Beckner, J. B. Bell, M. S. Day, L. H. Howell, C. C. Joggerst, M. J. Lijewski, A. Nonaka, M. Singer, and M. Zingale. CASTRO: A New Compressible Astrophysical Solver. I. Hydrodynamics and Self-gravity. *Astrophys J*, 715:1221–1238, June 2010.

[2] G. L. Bryan, M. L. Norman, J. M. Stone, R. Cen, and J. P. Ostriker. A piecewise parabolic method for cosmological hydrodynamics. *Computer Physics Communications*, 89:149–168, August 1995.

[3] P. Colella. Multidimensional upwind methods for hyperbolic conservation laws. *Journal of Computational Physics*, 87:171–200, March 1990.

[4] P. Colella and H. M. Glaz. Efficient solution algorithms for the Riemann problem for real gases. *Journal of Computational Physics*, 59:264–289, June 1985.

[5] P. Colella and M. D. Sekora. A limiter for PPM that preserves accuracy at smooth extrema. *Journal of Computational Physics*, 227:7069–7076, July 2008.

[6] P. Colella and P. R. Woodward. The Piecewise Parabolic Method (PPM) for Gas-Dynamical Simulations. *Journal of Computational Physics*, 54:174–201, September 1984.

[7] Randall J. LeVeque. *Finite-Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2002.

[8] G. H. Miller and P. Colella. A Conservative Three-Dimensional Eulerian Method for Coupled Solid-Fluid Shock Capturing. *Journal of Computational Physics*, 183:26–82, November 2002.

[9] T. Plewa and E. Müller. The consistent multi-fluid advection method. *Astron Astrophys*, 342:179–191, February 1999.

[10] J. Saltzman. An Unsplit 3D Upwind Method for Hyperbolic Conservation Laws. *Journal of Computational Physics*, 115:153–168, November 1994.

[11] J. M. Stone, T. A. Gardiner, P. Teuben, J. F. Hawley, and J. B. Simon. Athena: A New Code for Astrophysical MHD. *Astrophys J Suppl S*, 178:137–177, September 2008.

[12] E. F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer, 1997.

[13] M. Zingale, L. J. Dursi, J. ZuHone, A. C. Calder, B. Fryxell, T. Plewa, J. W. Truran, A. Caceres, K. Olson, P. M. Ricker, K. Riley, R. Rosner, A. Siegel, F. X. Timmes, and N. Vladimirova. Mapping Initial Hydrostatic Models in Godunov Codes. *Astrophys J Suppl S*, 143:539–565, December 2002.