Much ado about hydrodynamics and related problems
Part I: Overview & Burger's Equation

March 28 - April 1, 2016
IFT-UNESP
São Paulo, Brazil

Luis Lehner, Perimeter Institute
Vasileios Paschalidis, Princeton
Frans Pretorius, Princeton

(Based on previous work by Inaki Olabarrieta, Matt Choptuik and Pretorius, 2003 and Choptuik, Lehner, Pretorius, 2010)

1 Basic considerations

We begin with some basic considerations form mathematical properties of generic hyperbolic systems. The subject is certainly much deeper than what can be covered in a few weeks (or even a semester). However we here include the main results directly relevant to our discussion in simple situations which can be intuitively applied to more general ones. The interested reader should consult the excellent book by Gustaffson, Kreiss and Oliger[1], arguably the fathers of numerical analysis.

All (quasi-linear) "hyperbolic" equations can be generically written, in first order form, as

$$q_{,t} + A^i(q,t) q_{,x^i} = S(q, x^a, t) \tag{1}$$

with q a vector $\in \Re^m$ describing the functions describing the solution, A^i $m \times m$ matrices, $i = 1..d$ (with d the dimensionality of space) and S particular sources or "lower order terms" (i.e. non-differentiated terms) . For instance, Maxwell, Einstein and Euler equations can all be cast this way. A system given by (1) is called "hyperbolic" if the matrices A^i can be diagonalized by appropriate transformations & all eigenvalues are real[1]. In this case, well posedness of the problem (in the absence of boundaries) can be established. The mathematical definition of well posedness (which has direct implications at the physical level) says a problem is well posed if (for simplicity we consider no boundaries are involved, though the concept can be extended for this case):

- A solution exists.
- This solution is unique.
- The solution depends continuously on the initial data. Namely the solution q_1 obtained from initial $q_1(x, t = 0)$ and the solution q_2 from initial data $q_2(x, t = 0) = q_1(x, t = 0) + \delta(x)$ (for a small data) will be "close" to each other. By close on means $|q_2(x, T) - q_1(x, T)| \leq Ae^{\kappa T}|\delta|$. (with $A, \kappa > 0$ constants which do not depend on the initial data.

Alternatively, if it can not be diagonalized or at least one eigenvalue is complex, the equations can not describe a well posed problem (ie. the problems will be ill posed generically!). Typically ill posed problems will have a bound like $|q_2(x, T) - q_1(x, T)| \leq A(\omega)e^{\kappa(\omega)T}|\delta|$. Thus, in particular, κ will grow for larger modes in the initial data.

An example in one dimension where the eigenvalues are real ($= 1$) but is not diagonalizable is given by

$$\vec{q}_t = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \vec{q}_x =: A\vec{q}_x \tag{2}$$

The behavior of solutions can be understood in terms of simple wave solutions (or by taking different Fourier modes). The solution can be written as,

$$\vec{q}(x, t) = e^{i\omega At} e^{i\omega x} \widehat{\vec{q}}(\omega, 0) = \left(i + i\,\omega \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} t \right) e^{i\omega(x+t)} \widehat{\vec{q}}(\omega, 0) \tag{3}$$

Computing the norm $|\vec{q}(x, t)|$ one sees that there is a polynomial growth in ωt. This is typically the case for weakly hyperbolic systems, while in principle a polynomial growth is not as bad, one can prove that generic lower order terms will drive the growth to be exponential.

NOTE: these are important properties regardless of whether one intends to do numerics or not. The statement of ill posedness already indicates the particular theory that gave raise to those equations *either simply does not make sense or freedom in the theory must be exploited to render the equations hyperbolic.* An example of the latter is Einstein's theory, and example of the former are many new "alternative" theories of gravity which, for instance, give rise to *third* derivatives of the field variables. As mentioned by Choptuik, the importance of true hyperbolicity has long been understood by mathematicians, physicists have only caught up to this in just a few branches, it is *all too common* to see ill-defined approaches doomed to

[1] If a single transformation can diagonalize all A^i the system is strictly hyperbolic, if the matrices A are symmetric, the system is called symmetric hyperbolic otherwise it is called strongly hyperbolic **all these types** are the only ones that *make sense* physically

fail for lack of awareness of this issue. If you understand this, you will definitively be ahead! If additionally you are interested in obtaining solutions via numerical means, you must make sure things make the most sense at the analytical level. Experience (and good arguments!) says that at the numerical level, if something can go bad... it will go bad!

Note that hyperbolicity –in particular the existence of the transformation that diagonalizes the problem – imply that the problem 1 can be re-expressed as (say in 1D)

$$Q_{,t} + D^x(q,t) Q_{,x} = T(S(q,x,t)) \tag{4}$$

for $Q \equiv T(q)$ with T the transformation that diagonalizes the problem. and so it can be seen as a series of equations like

$$Q^a_{,t} + \lambda_a \partial_x Q^a = [T(S(q,x,t))]^a \tag{5}$$

with $a = 1..m$. In what follows, unless necessary we will thus discuss things with respect to a single equation of the type (5) as hyperbolicity implies we can always reduce our system to this level.

2 Towards the Euler equations, generalities

Euler equations arise from the compressible Navier-Stokes equations by neglecting viscosity and head conduction. Mathematically one of the most interesting features admitted by solutions of these equations is the presence of *shocks*.

Shocks are mathematical idealizations of the steep gradients that can be present in smooth solutions to the full Navier-Stokes equations where rapid changes occur over *very thin* regions. Any numerical effort to describe solutions to these equations must therefore be aware of this possible scenario! In fact, "naive" or "direct" discretizations of these equations typically obtain solutions which are either very smeared out or with spurious oscillations near discontinuities. We will return to numerical techniques to address these problems later; however, we first need to understand analytical properties of these equations to understand what to do.

3 Advection equation, linearly degenerate and truly non-linear equations

Consider $q(x,t)$ a generic function we want to compute, we further assume the behavior of this function obeys the simple *hyperbolic* equation,

$$q_{,t} + A q_{,x} = 0\,, \tag{6}$$

with $q(x,t) \in \Re^m$, A a $m \times m$ matrix. The system is hyperbolic if A is diagonalizable, which allows us to view the solution in terms of propagating waves. The simplest example is the *constant coefficient one-dimensional* advection equation,

$$q_{,t} + u q_{,x} = 0\,. \tag{7}$$

The solution to this problem is given by $q(x-ut,0)$ thus any profile q has at the initial time, it is simply advected at velocity u. Of course, there are more complicated cases, for instance,

$$q_{,t} + F(q)_{,x} = 0\,. \tag{8}$$

for which a particularly simple example would be $F(q) = q^2/2$, and so $q_{,t} + qq_{,x} = 0$. While it does not look too different from equation (7), there is a lot more than meets the eye here...

3.1 Small detour, why do we even care?

Consider the function $\rho(x,t)$ describing the density of a fluid in a one-dimensional setting. The mass m in a box of extent $[x_1, x_2]$ at time t is given by

$$m = \int_{x_1}^{x_2} \rho(x,t)dx\,. \tag{9}$$

Now, if the "walls" are permeable, fluid might enter/leave the box and so m will change in time. The rate of fluid flow (flux) past any given point is $F(x,t) = \rho(x,t)u(x,t)$ so,

$$\frac{d}{dt}m = \frac{d}{dt}\int_{x_1}^{x_2} \rho(x,t)dx = \rho(x_1,t)u(x_1,t) - \rho(x_2,t)u(x_2,t) = F(x_1,t) - F(x_2,t) \tag{10}$$

Integrating the above equation in both time (in $[t_1, t_2]$) one obtains,

$$\int_{x_1}^{x_2} \rho(x,t_2)dx = \int_{x_1}^{x_2} \rho(x,t_1)dx + \int_{t_1}^{t_2} \rho(x_1,t)u(x_1,t)dt - \int_{t_1}^{t_2} \rho(x_2,t)u(x_2,t)dt \tag{11}$$

Thus, $m(t_2)$ will be given by $m(t_1)$ plus/minus the amount of fluid that entered/left the domain. What do we do with this?, suppose ρ, u are differentiable (smooth), thus

$$\rho(x,t_2) - \rho(x,t_1) = \int_{t_1}^{t_2} \partial_t\left(\rho(x,t)\right)dt \tag{12}$$

$$\rho(x_2,t)u(x_2,t) - \rho(x_1,t)u(x_1,t) = \int_{x_1}^{x_2} \partial_x\left(\rho(x,t)u(x,t)\right)dx \tag{13}$$

Replacing in equation (11), we get

$$\int_{t_1}^{t_2}\int_{x_1}^{x_2} \left[\partial_t\rho + \partial_x(\rho u)\right]dxdt = 0 \tag{14}$$

and so $\partial_t\rho + \partial_x(\rho u) = 0$ simply states conservation of mass. Recall that when obtaining this equation, we assumed ρ, u are differentiable, as we will discuss below, this assumption need not be justified.

3.2 A shocking truth, life is not so simple

Consider again the equation (8), and for simplicity take $F(q) = q^2/2$, the resulting equation is known as the Burger's equation which can also be found written as $q_{,t} + qq_{,x} = 0$. Now, this does not seems like too different from $q_{,t} + u q_{,x} = 0$ for a general function u; it turns out however, the "small" difference introduced will lead to dramatically different behavior[2]

Why is this the case? A simple way to see this is to consider the behavior of perturbations δq over a given solution q_o for a small period of time, the equation determining this behavior is straightforwardly,

$$\delta q_{,t} + q_o \delta q_{,x} = 0 \tag{15}$$

from our previous discussion, the solution will be given by $\delta q(x,t) = \delta(x - q_o t)$. Notice that if, in particular, $0 < q_o(x_2, t=0) < q_o(x_1, t=0)$ $(x_1 < x_2)$ it is trivial to see that at $t = (x_2 - x_1)/(q_o(x_1, t=0) - q_o(x_2, t=0))$ δq is multivalued, which does not make sense. What happened here is that the *characteristics* of the solution crossed. And the solution can not be determined past this point at least. Mathematically the difference between the equation $q_{,t} + q q_{,x} = 0$ and $q_{,t} + u q_{,x} = 0$ is that the characteristic speed of the former depends on the solution itself while this is not the case for the latter. Mathematicians refer to the first one as truly non-linear while the second as linearly degenerate. For higher dimensions the analog problem arises if the characteristic speed of a given mode depends on the solutions in the subspace described by the eigenvector corresponding to such speed. If so, as in the simple Burger's example, the solution can not be easily determined past some local point (points). Physically on the other hand, this does not make sense, as we expect a unique solution to exist past these problematic points, so something went wrong somewhere. Where did we go wrong? It was in our assumption of smoothness and differentiability which we used to derive the differential form of the equation which is not valid if discontinuities develop in the solution. To address this issue we approach the problem in a different light. The new approach is to consider "weak solutions", i.e. solutions to the integral form of (11),

$$\int\int \Phi(q_{,t} + \partial_x f(q)) dx dt = 0 \,, \tag{16}$$

for arbitrary (smooth and with compact support) test functions Φ, upon integration by parts (and taking the limit of the boundary –both in space and time to infinity–, one obtains,

$$\int\int (\Phi_{,t} q + f \partial_x \Phi) dx dt = \int \Phi(x,0) q(x,0) dx \,. \tag{17}$$

This approach allows for a way to deal with discontinuities in a special way (recall similar "tricks" allow to make sense of the Dirac δ "function" in a distributional sense. A detailed discussion of this theory is beyond the scope of this course. We instead describe the main aspects relevant to the numerical implementation of these type of equations.

3.3 Riemann problem and general considerations

A Riemann problem is defined by a conservation law type equation with piecewise constant data having a single discontinuity. Let's go back to Burger's equation as an example and consider initial data given by:

$$q(x, t=0) = \begin{cases} q_l & \text{if } x < 0 \\ q_r & \text{if } x > 0 \end{cases} \tag{18}$$

[2] The word "dramatic" is often a hyperbola used by physicists to stress an important point. It is often an exaggeration but, we can assure you... in this case it is definitively not!

For $q_l > q_r$ it is easy to see that the "left" (l) state will run into the 'right' (r) state. The solution is thus multivalued along a line defined by $x = st \equiv (q_l + q_r)/2$. The velocity of the shock s is given by the Rankine-Hugoniot jump conditions, which for one-dimensional problems is simply $s = (f(q_l) - f(q_r))/(q_l - q_r)$. The unique solution is given by,

$$q(x, t = 0) = \begin{cases} q_l & \text{if } x < st \\ q_r & \text{if } x > st \end{cases} \tag{19}$$

For $q_l < q_r$ the characteristics diverge and several weak solutions exist. The requirement that the entropy across a discontinuity increases help single out a unique solution, which is known as a rarefraction wave given by,

$$q(x, t) = \begin{cases} q_l & \text{if } x < q_l t \\ x/t & \text{if } q_l t < x < q_r t \\ q_r & \text{if } q_r t < x \end{cases} \tag{20}$$

In principle the approach we took to obtain solutions can be generalized to arbitrary dimensions, however it is expensive and cumbersome. Since we are after an approximate solution we can take a less costly approach –still based in the previous discussion– which still provides the correct solution as the discretization length is taken to 0 in a controlled manner.

4 Discretization

4.1 Finite Volumes

We are interested in truly non-linear problems which, as discussed, give rise to shocks (discontinuities on the variables describing the state of the fluid) even when the initial data is smooth. This implies that naïve discretizations based on the continuity of the functions (like some of the finite difference methods used on Project 1) are doomed to fail. There are different approaches we could take to solve this system. Here we will take a finite volume approach, meaning that we will assume that we have a mesh of grid points that define a cell structure on our spacetime (see Figure 1). In the presence of discontinuities the only way to make sense of our system of equations is to consider averages over a finite volume of the spacetime. Therefore to find the discretization we take the average of equation (44) over a spacetime cell $\mathcal{C}_i^{n+1/2}$:

$$\frac{1}{V_{\mathcal{C}_i^{n+1/2}}} \int_{\mathcal{C}_i^{n+1/2}} \frac{\partial \mathbf{q}}{\partial t} + \frac{1}{V_{\mathcal{C}_i^{n+1/2}}} \int_{\mathcal{C}_i^{n+1/2}} \frac{\partial \mathbf{f}}{\partial x} = \frac{1}{V_{\mathcal{C}_i^{n+1/2}}} \int_{\mathcal{C}_i^{n+1/2}} \boldsymbol{\psi}, \tag{21}$$

where $\mathcal{C}_i^{n+1/2}$ is the region of spacetime defined by $(t^n, t^{n+1}) \times (x_{i-1/2}, x_{i+1/2})$, and $V_{\mathcal{C}_i^{n+1/2}} = \Delta t \Delta x$ is its volume. The resulting equation can be written as:

$$\frac{1}{\Delta t \Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{t^n}^{t^{n+1}} \frac{\partial \mathbf{q}}{\partial t} dx dt + \frac{1}{\Delta t \Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{t^n}^{t^{n+1}} \frac{\partial \mathbf{f}}{\partial x} dx dt = \frac{1}{\Delta t \Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{t^n}^{t^{n+1}} \boldsymbol{\psi} dx dt. \tag{22}$$

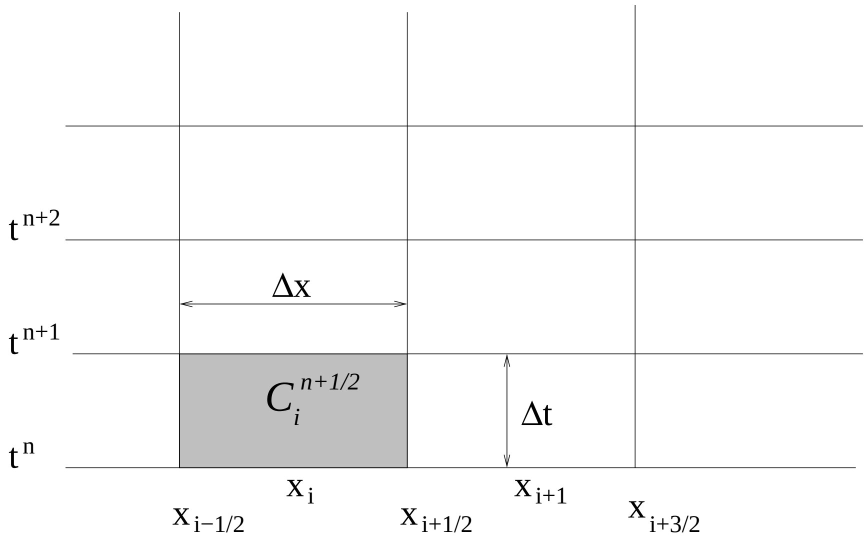


Figure 1: Cell structure of the spacetime for a finite volume discretization in one dimension. The spacetime cells $C_i^{n+1/2}$ are centred at positions $(t^{n+1/2}, x_i)$ and their volumes are $\Delta t \Delta x$.

We can partially integrate the different terms of the equation using Gauss' theorem:

$$\frac{\bar{\mathbf{q}}_i^{n+1} - \bar{\mathbf{q}}_i^n}{\Delta t} + \frac{\mathbf{F}_{i+1/2}^{n+1/2} - \mathbf{F}_{i-1/2}^{n+1/2}}{\Delta x} = \widehat{\boldsymbol{\psi}}_i^{n+1/2}. \tag{23}$$

Here we have used the following definitions: the spatial averages of the conservative variables,

$$\bar{\mathbf{q}}_i^n \equiv \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{q}(t^n, x) dx, \tag{24}$$

the temporal averages for the fluxes, also referred as the numerical fluxes,

$$\mathbf{F}_{i+1/2}^{n+1/2} \equiv \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} \mathbf{f}\left(\mathbf{q}(t, x_{i+1/2})\right) dt, \tag{25}$$

and the total averages over the spacetime cell for the sources,

$$\widehat{\boldsymbol{\psi}}_i^{n+1/2} \equiv \frac{1}{\Delta x \Delta t} \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{t^n}^{t^{n+1}} \boldsymbol{\psi}(t, x) dx dt. \tag{26}$$

The idea now is to use equation (23) to calculate $\{\bar{\mathbf{q}}_i^{n+1}\}$ assuming we know the values $\{\bar{\mathbf{q}}_i^n\}$. However the calculation of the numerical fluxes $\{\mathbf{F}_{i+1/2}^{n+1/2}\}$ is not as straightforward as one may think—these fluxes are averages in time, so in order to explicitly calculate them we need to already know the solution. More importantly, the values for the fluid quantities on the left side of the cell boundary $\{\bar{\mathbf{q}}_i^n\}$ and on the right side $\{\bar{\mathbf{q}}_{i+1}^n\}$ won't agree in general. The values have discontinuities and *a priori* is not clear which values to use in order to compute the numerical fluxes. One way to solve these problems is to use an idea due to Godunov that involves solving a Riemann problem at every cell boundary in order to calculate $\{\mathbf{F}_{i+1/2}^{n+1/2}\}$. For more information about Godunov methods see LeVeque [6]. In the following section we explain one such method.

4.2 Roe Solver

The solution of the full Riemann problem at every cell boundary is usually not very efficient. In most cases the overall time step to update the variables to the future time will involve some kind of iterative process, and thus exactly solving the Riemann problem at each iteration will not imply that the overall process will be solved more rapidly or accurately. The Roe solver is a solver that uses modified Riemann problems in order to compute the numerical fluxes. For a more extensive explanation of this and other approximate Riemann solvers see LeVeque [6]. The main idea is to linearize the fluxes in equation (44) as functions of $\mathbf{q}$, also assuming that the sources vanish:

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{f}}{\partial \mathbf{q}} \frac{\partial \mathbf{q}}{\partial x} = 0. \tag{27}$$

Considering $\partial \mathbf{f}/\partial \mathbf{q}$ to have constant coefficients linearizes the above equation, and a solution can be obtained by diagonalizing the Jacobian matrix (see appendix for the solution of the scalar linear equation and its generalization to a system of linear equations). The numerical flux can then be written as a function of the solution to this problem. Here we write the resulting numerical fluxes directly as:

$$\mathbf{F}^{\text{Roe}}_{i+1/2} = \frac{1}{2}\left[\mathbf{f}\left(\tilde{\mathbf{p}}^R_{i+1/2}\right) + \mathbf{f}\left(\tilde{\mathbf{p}}^L_{i+1/2}\right) - \sum_\alpha |\lambda_\alpha| \omega_\alpha \mathbf{r}_\alpha\right]. \tag{28}$$

Some explanation of the different terms that appear in the above equation are in order. First, $(\tilde{\mathbf{p}}^R,\ \tilde{\mathbf{p}}^L)$, known as right and left *reconstructed variables*, are the values of the primitive variables at the boundary, $x_{i+1/2}$ calculated via some specific reconstruction (interpolation) scheme. Special care is taken in calculating the reconstructed variables in order to reduce spurious oscillations close to discontinuities. Here we use a *slope limiter* interpolation to compute the reconstructed values (see Martí and Mueller [2] and LeVeque [6] for alternate reconstruction algorithms):

$$\begin{aligned} \tilde{\mathbf{p}}^L_{i+1/2} &= \bar{\mathbf{p}}_i + \boldsymbol{\sigma}_i \left(x_{i+1/2} - x_i\right), \qquad (29) \\ \tilde{\mathbf{p}}^R_{i+1/2} &= \bar{\mathbf{p}}_{i+1} + \boldsymbol{\sigma}_{i+1} \left(x_{i+1/2} - x_{i+1}\right), \qquad (30) \end{aligned}$$

where $\boldsymbol{\sigma}_i$ is given by

$$\boldsymbol{\sigma}_i = \text{minmod}\left(\mathbf{s}_{i-1/2}, \mathbf{s}_{i+1/2}\right). \tag{31}$$

Here:

$$\mathbf{s}_{i+1/2} = \frac{\bar{\mathbf{p}}_{i+1} - \bar{\mathbf{p}}_i}{x_{i+1} - x_i}, \tag{32}$$

and the minmod function is defined by

$$\text{minmod}(a,b) = \begin{cases} 0 & \text{if} & & ab < 0 \\ a & \text{if} & |a| < |b| & \text{and} \ \ ab > 0 \\ b & \text{if} & |a| > |b| & \text{and} \ \ ab > 0. \end{cases} \tag{33}$$

In equation (28) we also use the characteristic structure of the Riemann problem at the $x_{i+1/2}$ interface (λ_α, ω_α, $\mathbf{r}_\alpha$). Given the Jacobian matrix:

$$\mathbf{A}|_{i+1/2} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{q}}\right|_{\mathbf{q}=1/2\,(\tilde{\mathbf{q}}^L_{i+1/2}+\tilde{\mathbf{q}}^R_{i+1/2})}, \tag{34}$$

λ_α are the eigenvalues[3] of $\mathbf{A}$, $\mathbf{r}_\alpha$ are the right eigenvectors associated with the eigenvalues λ_α and ω_α are the jumps in the characteristic variables defined by

$$\tilde{\mathbf{q}}^R_{i+1/2} - \tilde{\mathbf{q}}^L_{i+1/2} = \sum_\alpha \omega_\alpha \boldsymbol{r}_\alpha . \tag{35}$$

Here $(\tilde{\mathbf{q}}^R, \tilde{\mathbf{q}}^L)$ are the values of the conservative variables calculated from the reconstructed primitive variables $(\tilde{\mathbf{p}}^R, \tilde{\mathbf{p}}^L)$. Reconstruction of the primitive variables followed by transformation to conservative variables generally yields more stable results than direct reconstruction of the conservative variables.

The final part of the Roe solver involves the update of $\{\bar{\mathbf{q}}^n_i\}$. The fact that we use an approximate Riemann solver to calculate the numerical flux makes (28), when evaluated using the spatial averages at time t^n, a first order approximation to the real numerical flux defined by (25). This is usually the case also when calculating the numerical sources (26). In order to make the time evolution second order (in the temporal discretization scale), we use a second order Runge-Kutta method to advance the solution in time:

$$\begin{aligned} \bar{\mathbf{q}}^{n+1/2} &= \bar{\mathbf{q}}^n + \frac{\Delta t}{2} L(\bar{\mathbf{q}}^n) && (36)\\ \bar{\mathbf{q}}^{n+1} &= \bar{\mathbf{q}}^n + \Delta t L(\bar{\mathbf{q}}^{n+1/2}) \,. && (37) \end{aligned}$$

Here L is defined by:

$$L(\bar{\mathbf{q}}^n) = -\frac{\mathbf{F}^{Roe}_{i+1/2}\left(\bar{\mathbf{q}}^n\right) - \mathbf{F}^{Roe}_{i-1/2}\left(\bar{\mathbf{q}}^n\right)}{\Delta x} + \widehat{\boldsymbol{\psi}}_i\left(\bar{\mathbf{q}}^n\right). \tag{38}$$

Summarizing the approach described above involves:

- Expressing the system of equation in "conservation" form, i.e. $q^i_{,t} + \partial_j F^{ij} = 0$.
- Solve a series of (approximate) Riemann problems at cell interface.
- Calculating the characteristic structure of the system (for solving the Riemann problem).

Armed with the considerations and techniques described so far, let's turn to a problem of relevance to us.

[3] Here α (and later on also β) labels the equation number, in the fluid case since we have two equations it takes values on $\{1, 2\}$.

Conversely at every half and full step in our update procedure we need to calculate the primitive variables after the conservative variables have been evolved. It is not difficult to invert the equations that define the conservative variables in order to get the primitive ones (see NC [5]):

$$\begin{aligned} P &= -2\beta\tau + \sqrt{4\beta^2\tau^2 + (\Gamma - 1)\left(\tau^2 - S^2\right)} && (60) \\ \rho &= P/(\Gamma - 1) && (61) \\ v &= \frac{S}{\tau + P} && (62) \end{aligned}$$

where $\beta = (2 - \Gamma)/4$.

Floor

Due to numerical errors (truncation error, roundoff error) the quantities that describe the fluid can sometimes take unphysical values (i.e. negative pressures, negative densities, speeds larger than one, etc,...) (see NC [5]). Effects of such errors become particularly important in "evacuated" regions, where densities are low and velocities can be very large. In order to circumvent problems associated with these errors, we force certain values to be above some threshold, that we call a *floor*. For the ultrarelativistic fluid it is convenient to floor the conservative variable τ in the following way:

$$\tau = \max\{\tau, \mathit{floor} + |S|\}, \tag{63}$$

where $|S|$ is the absolute value of S and *floor* is a small value, typically several orders of magnitude (usually 13 or 14 orders of magnitude) smaller than typical values of τ. Generally, a flooring procedure of this sort will not have an important dynamical effect (although this is something that needs to be verified empirically), and ameliorates the problems described above. We recommend application of this algorithm every time the conservative variables are updated or calculated from reconstructed primitive variables at the cell boundaries.

7 Burger's Equation

Burger's equations is an example of a non-linear scalar equation that produces shocks, even from smooth initial data (see LeVeque [6]). One form of the equation is

$$\dot{q} + q\frac{\partial q}{\partial x} = 0. \tag{64}$$

which is easy to cast into conservative form

$$\dot{q} + \left(\frac{1}{2}q^2\right)' = 0. \tag{65}$$

Using the notation introduced previously, we have

$$\mathbf{q} = q, \qquad \mathbf{f} = \frac{1}{2}q^2, \qquad \boldsymbol{\psi} = 0. \tag{66}$$

We solve this equation with a finite volume discretization and a Roe solver, as outlined in the previous section. The finite volume discretization of the equation is

$$\frac{\bar{q}_i^{n+1} - \bar{q}_i^n}{\Delta t} + \frac{F_{i+1/2}^{n+1/2} - F_{i-1/2}^{n+1/2}}{\Delta x} = 0\,, \tag{67}$$

where $\bar{q}_i^n$ is the spatial average defined by equation (24) and $F_{i+1/2}^{n+1/2}$ is the numerical flux. We now focus on a description of the characteristic structure of the equation that will allow us to compute the Roe flux. Since (64) is a scalar equation the Jacobian matrix $\mathbf{A}$ is also a scalar

$$\mathbf{A} = q, \tag{68}$$

the eigenvalue is the same scalar, $\lambda = q$, and we can take the right eigenvector to be 1. Finally, the jump ω is just the difference of q across the cell boundary,

$$\omega = q^R - q^L\,. \tag{69}$$

Thus, we can write the Roe numerical flux as:

$$F_{i+1/2}^{\mathrm{Roe}} = \frac{1}{2}\left[f\left(q^L\right) + f\left(q^R\right) - |\lambda|\, r\, \omega\right]_{i+1/2}, \tag{70}$$

where $[\cdots]_{i+1/2}$ means that the quantities within the bracket are evaluated at $x_{i+1/2}$. We then solve (67) using the following time-stepping procedure:

```
1) Calculate the Roe numerical fluxes at the cell boundaries.
2) Update the variables to the half time step using equation (47) with
   Delta t=Delta t/2.
3) Use the quantities at the half time step to compute the Roe numerical fluxes.
4) Do a full step to update to the future time step using the numerical fluxes
   calculated in 3).
```

The previous pseudo code describes the use of the Roe solver within a second order Runge-Kutta time stepping scheme. The overall method should be second order if no shocks are developed, except in the vicinity of extrema of the dynamical variable, where the slope limiting interpolation will generally degrade the solution to first order.

The calculation of the numerical fluxes involve the following steps:

```
1) Calculate the left and right reconstructed variables at the cell boundary.
2) Calculate the characteristic structure: eigenvalues, right eigenvectors
   and jumps in the characteristic variables.
4) Calculate the physical fluxes F for the left and right reconstructed
   variables.
3) Calculate the Roe numerical flux using equation (50).
```

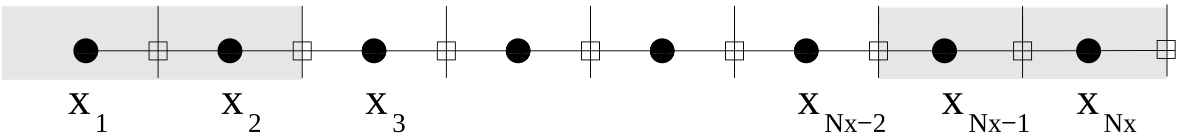


Figure 2: Spatial cell structure for a grid with N_x cells and two ghost cells ($N_g = 2$) per boundary. The solid circles lie at the spatial locations of the grid cell centres, and coincide with the grid points generated by the RNPL code. Shaded areas represent ghost cells where dynamical variables are updated according to the boundary conditions we impose. The squares and vertical lines denote the cell boundaries, and are the locations at which the reconstructed variables and numerical fluxes are calculated.

Boundary Conditions and Cell Structure

In order to impose boundary conditions we make use of *ghost cells.* These cells are *not* updated using the equations of motion, but rather are set according to the specific boundary conditions that we wish to impose. The boundary conditions that we impose are a first order approximation to outgoing boundary conditions (often called *outflow conditions* in the fluid literature). We implement these conditions simply by setting the ghost cell values to the value in the last regular cell:

$$\begin{aligned} q_1 &= q_3 && (71)\\ q_2 &= q_3 && (72)\\ q_{N_x-1} &= q_{N_x-2} && (73)\\ q_{N_x} &= q_{N_x-2} && (74)\end{aligned}$$

Here N_x is the number of cells in the entire grid (including ghost cells).

Note that in order to update the interior points, i.e. the x_i with $i = N_g + 1, ..., N_x - N_g$ we need to calculate the numerical fluxes at positions $x_{i+1/2}$ with $i = N_g, ..., N_x - N_g$.

Finite Difference

Due to the simplicity of equation (64), it is also straightforward to solve using a finite difference approximation that uses an *upwind stencil* (an upwind stencil is one which uses information only a specific characteristic direction, relative to the point at which the approximation is applied). Interestingly, however, if we discretize Burger's equation in the form (64) using such a technique

$$q_i^{n+1} = q_i^n - \frac{\Delta t}{\Delta x} q_i^{n+1/2} \left(q_i^{n+1/2} - q_{i-1}^{n+1/2} \right), \tag{75}$$

we find that the shock speeds obtained are erroneous, even in the continuum limit. In this instance, the problem can be solved by discretizing the conservative form of the equation, (65):

$$q_i^{n+1} = q_i^n - \frac{\Delta t}{\Delta x} \left[\frac{1}{2} \left(q_i^{n+1/2} \right)^2 - \frac{1}{2} \left(q_{i-1}^{n+1/2} \right)^2 \right]. \tag{76}$$

High Resolution Shock Capturing (HRSC) Scheme

March 28 - April 1, 2016

Reference book: *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction* by E.F. Toro

Based on work by Yuk Tung Liu, 2010

1 Equations of hydrodynamics in conservative form

Recall eqns of hydrodynamics (adiabatic flow, no gravity):

$$\begin{aligned} \partial_t \rho + \boldsymbol{\nabla}\cdot(\rho \boldsymbol{u}) &= 0 && (1)\\ \partial_t(\rho u_i) + \sum_j \partial_j \Pi_{ij} &= 0 && (2)\\ \partial_t E + \boldsymbol{\nabla}\cdot[(E+P)\boldsymbol{u}] &= 0, && (3) \end{aligned}$$

where

$$\begin{aligned} \Pi_{ij} &= \rho u_i u_j + P\delta_{ij} && (4)\\ E &= \rho\epsilon + \frac{1}{2}\rho u^2. && (5) \end{aligned}$$

γ-law equation of state:

$$P = (\gamma-1)\rho\epsilon. \tag{6}$$

These hydro equations are written in conservative form:

$$\partial_t U + \boldsymbol{\nabla}\cdot\boldsymbol{F}(U) = S(U), \tag{7}$$

where

$$U = \begin{pmatrix} \rho \\ \rho u_x \\ \rho u_y \\ \rho u_z \\ E \end{pmatrix} \quad , \quad S = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \tag{8}$$

$$F_x = \left(\begin{array}{c} \rho u_x \\ \rho u_x^2 + P \\ \rho u_x u_y \\ \rho u_x u_z \\ (E+P)u_x \end{array} \right) \quad , \quad F_y = \left(\begin{array}{c} \rho u_y \\ \rho u_x u_y \\ \rho u_y^2 + P \\ \rho u_y u_z \\ (E+P)u_y \end{array} \right) \quad , \quad F_z = \left(\begin{array}{c} \rho u_z \\ \rho u_x u_z \\ \rho u_y u_z \\ \rho u_z^2 + P \\ (E+P)u_z \end{array} \right) . \tag{9}$$

HRSC methods are designed to evolve the system of equations written in conservative form. In the following I will only discuss the equations in 1D.

1.1 Equations in one dimension

1.1.1 Plane 1D flow

Consider $\rho = \rho(x)$, $P = P(x)$ and $\boldsymbol{u} = u(x)\hat{\boldsymbol{x}}$. Hydrodynamics equations become:

$$\partial_t U + \partial_x F(U) = 0, \tag{10}$$

where

$$U = \left(\begin{array}{c} \rho \\ \rho u \\ E \end{array} \right) \quad , \quad F = \left(\begin{array}{c} \rho u \\ \rho u^2 + P \\ (E+P)u \end{array} \right) . \tag{11}$$

The variables ρ, ρu and E are called the *conservative variables*, whereas ρ, u and P are called the *primitive variables*.

2 Riemann Problem

The Riemann problem plays an important role in the HRSC technique.

Definition: The Riemann problem for 1D equations is the initial value problem (IVP) for the conservation laws

$$\partial_t U + \partial_x F(U) = 0 \tag{12}$$

with initial conditions (IC)

$$U(x,0) = \left\{ \begin{array}{ll} U_L & \text{if } x < 0 \\ U_R & \text{if } x > 0 \end{array} \right. . \tag{13}$$

The domain of interest in the x-t plane are points (x,t) with $-\infty < x < \infty$ and $t > 0$.

2.1 Linear advection equation

$$\text{PDE: } \partial_t u + \lambda \partial_x u = 0, \tag{14}$$

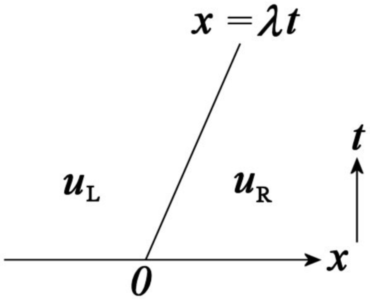
$$\text{IC: } u(x,0) = \left\{ \begin{array}{ll} u_L & \text{if } x < 0 \\ u_R & \text{if } x > 0 \end{array} \right. , \tag{15}$$

where λ is a constant.

The general solution of (14) is $u(x,t) = f(x - \lambda t)$, where f is an arbitary function. Hence the solution to the Riemann problem is

$$u(x,t) = \begin{cases} u_L & \text{if } x < \lambda t \\ u_R & \text{if } x > \lambda t \end{cases} . \tag{16}$$

This solution is depicted in the following diagram in the x-t plane:



2.2 Linear hyperbolic systems

$$\text{PDE: } \partial_t U + A \partial_x U = 0, \tag{17}$$

$$\text{IC: } U(x,0) = \begin{cases} U_L & \text{if } x < 0 \\ U_R & \text{if } x > 0 \end{cases} , \tag{18}$$

where

$$U = \begin{pmatrix} U_1 \\ . \\ . \\ . \\ U_m \end{pmatrix}, \tag{19}$$

and A is an $m \times m$ matrix with constant coefficients and has m *real* eigenvalues $\lambda_1, \cdots, \lambda_m$. Let M be the $m \times m$ matrix that diagonalizes A: $M^{-1}AM = \Lambda$ where Λ is the diagonal matrix

$$\Lambda = \begin{pmatrix} \lambda_1 & & & & \\ & . & & & \\ & & . & & \\ & & & . & \\ & & & & \lambda_m \end{pmatrix}. \tag{20}$$

Consider

$$U = MW \tag{21}$$

with

$$W = \begin{pmatrix} W_1 \\ . \\ . \\ . \\ W_m \end{pmatrix}, \tag{22}$$

Then (17) becomes

$$\begin{aligned} M\partial_t W + AM\partial_x W &= 0 \\ \Rightarrow \partial_t W + M^{-1}AM\partial_x W &= 0 \\ \Rightarrow \partial_t W + \Lambda\partial_x W &= 0 \qquad (23) \end{aligned}$$

or

$$\partial_t W_i + \lambda_i \partial_x W_i = 0. \qquad (24)$$

The solution of the above PDE is $W_i(x,t) = W_i(x - \lambda_i t)$. Hence

$$U_i(x,t) = \sum_{j=1}^{m} M_{ij} W_j(x - \lambda_j t) \ . \qquad (25)$$

The functions $W_j(x)$ are determined by $U(x,0)$ as

$$W_j(x) = (M^{-1}U(x,0))_j = \begin{cases} W_j^L & \text{if } x < 0 \\ \\ W_j^R & \text{if } x > 0 \end{cases} . \qquad (26)$$

where

$$W_j^L = (M^{-1}U_L)_j = \sum_{k=1}^{m} (M^{-1})_{ik} (U_L)_k, \qquad (27)$$

$$W_j^R = (M^{-1}U_R)_j = \sum_{k=1}^{m} (M^{-1})_{ik} (U_R)_k. \qquad (28)$$

Introduce a set of vectors $K^{(j)}$

$$K^{(j)} = \begin{pmatrix} K_1^{(j)} \\ . \\ . \\ . \\ K_m^{(j)} \end{pmatrix} \qquad (29)$$

such that

$$K_i^{(j)} = M_{ij}. \qquad (30)$$

The matrix M can be expressed as

$$M = \begin{bmatrix} K_1^{(1)} & K_1^{(2)} & & K_1^{(m)} \\ . & . & & . \\ . & . & \cdots & . \\ . & . & & . \\ K_m^{(1)} & K_m^{(2)} & & K_m^{(m)} \end{bmatrix}. \qquad (31)$$

According to linear algebra theory, $K^{(j)}$ is an eigenvector of A with eigenvalue λ_j:

$$AK^{(j)} = \lambda_j K^{(j)}. \qquad (32)$$

It follows from (25) that U can be expressed as a sum over the eigenvectors:

$$U(x,t) = \sum_{j=1}^{m} W_j(x - \lambda_j t) K^{(j)}. \tag{33}$$

Sort the eigenvalues so that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$. Then

$$U(x,t) = \begin{cases} U_R & \text{if } x > \lambda_1 t \\ U^{(1)} & \text{if } \lambda_2 t < x < \lambda_1 t \\ U^{(2)} & \text{if } \lambda_3 t < x < \lambda_2 t \\ . & . \\ . & . \\ . & . \\ U^{(m-1)} & \text{if } \lambda_m t < x < \lambda_{m-1} t \\ U_L & \text{if } x < \lambda_m t \end{cases}, \tag{34}$$

where

$$U^{(k)} = \sum_{j=1}^{k} W_j^L K^{(j)} + \sum_{j=k+1}^{m} W_j^R K^{(j)}, \quad (k = 1, 2, \cdots, m-1) \tag{35}$$

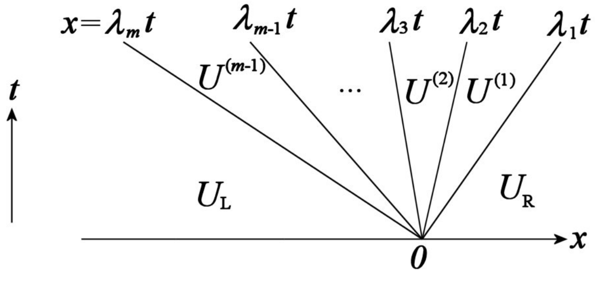
or

$$\begin{aligned} U^{(1)} &= U_R + \alpha_1 K^{(1)} && (36) \\ U^{(k)} &= U^{(k-1)} + \alpha_k K^{(k)} \quad (k = 2, 3, \cdots, m-1) && (37) \\ U_L &= U^{(m-1)} + \alpha_m K^{(m)}, && (38) \end{aligned}$$

where

$$\alpha_k = W_k^L - W_k^R \quad (k = 1, 2, \cdots, m). \tag{39}$$

The solution of the Riemann problem consists of $m+1$ states separated by m lines $x = \lambda_k t$ $(k = 1, 2, \cdots, m)$ as depicted below:



2.3 Riemann problem for 1D fluid dynamics

$$\text{PDE: } \partial_t U + \partial_x F = 0, \tag{40}$$

$$\text{IC: } U(x,0) = \begin{cases} U_L & \text{if } x < 0 \\ U_R & \text{if } x > 0 \end{cases}. \tag{41}$$

Here

$$U = \begin{pmatrix} U_1 \\ U_2 \\ U_3 \end{pmatrix} = \begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix} , \tag{42}$$

$$F = \begin{pmatrix} \rho u \\ \rho u^2 + P \\ (E+P)u \end{pmatrix} = \begin{pmatrix} U_2 \\ \frac{(3-\gamma)U_2^2}{2U_1} + (\gamma-1)U_3 \\ \frac{\gamma U_2 U_3}{U_1} - \frac{(\gamma-1)U_2^3}{2U_1^2} \end{pmatrix} . \tag{43}$$

As mentioned in the previous subsection, the PDE can be written as

$$\partial_t U + A(U)\partial_x U = 0 \tag{44}$$

with the Jacobian matrix given by

$$\begin{aligned} A(U) &= \frac{\partial F}{\partial U} \\ &= \begin{bmatrix} 0 & 1 & 0 \\ -\frac{1}{2}(\gamma-3)\left(\frac{U_2}{U_1}\right)^2 & (3-\gamma)\frac{U_2}{U_1} & \gamma-1 \\ -\gamma\frac{U_2U_3}{U_1^2} + (\gamma-1)\left(\frac{U_2}{U_1}\right)^3 & \gamma\frac{U_3}{U_1} - \frac{3}{2}(\gamma-1)\left(\frac{U_2}{U_1}\right)^2 & \gamma\frac{U_2}{U_1} \end{bmatrix} \qquad (45) \\ &= \begin{bmatrix} 0 & 1 & 0 \\ \frac{1}{2}(\gamma-3)u^2 & (3-\gamma)u & \gamma-1 \\ u\left[\frac{1}{2}(\gamma-1)u^2 - H\right] & H-(\gamma-1)u^2 & \gamma u \end{bmatrix} . \qquad (46) \end{aligned}$$

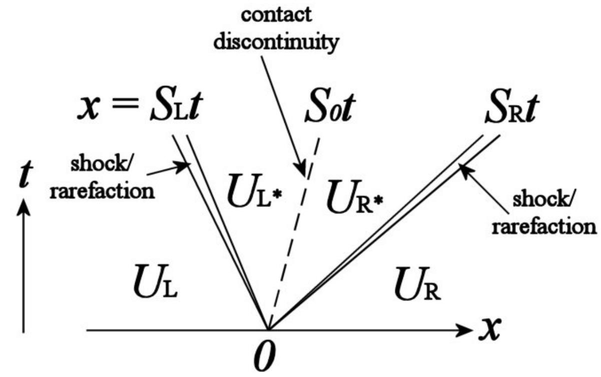
The eigenvalues of $A(U)$ are

$$\lambda_1 = u + c \ \ , \ \ \lambda_2 = u \ \ , \ \ \lambda_3 = u - c. \tag{47}$$

Unlike the previous subsection, the coefficients of the matrix $A(U)$ are *not* constants. The technique in Section 2.2 does not apply here. The system of PDEs is nonlinear.

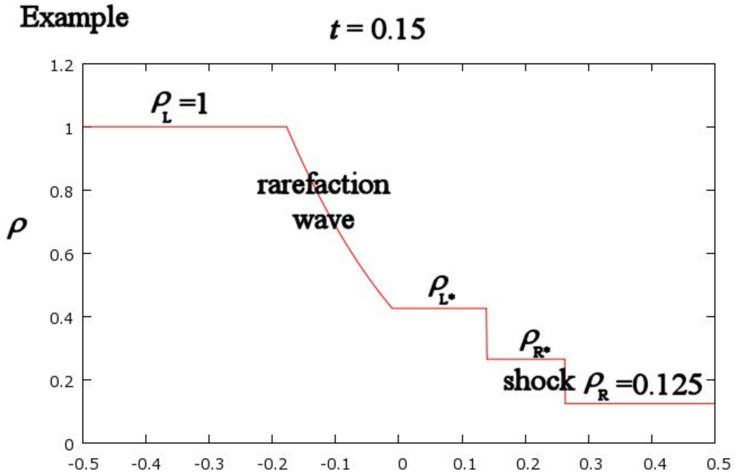
There is no exact closed-form solution to the Riemann problem for these equations. However, it is possible to devise an iterative numerical schemes to compute the solution to any desired degree of accuracy (see chapter 4 of Toro's book).

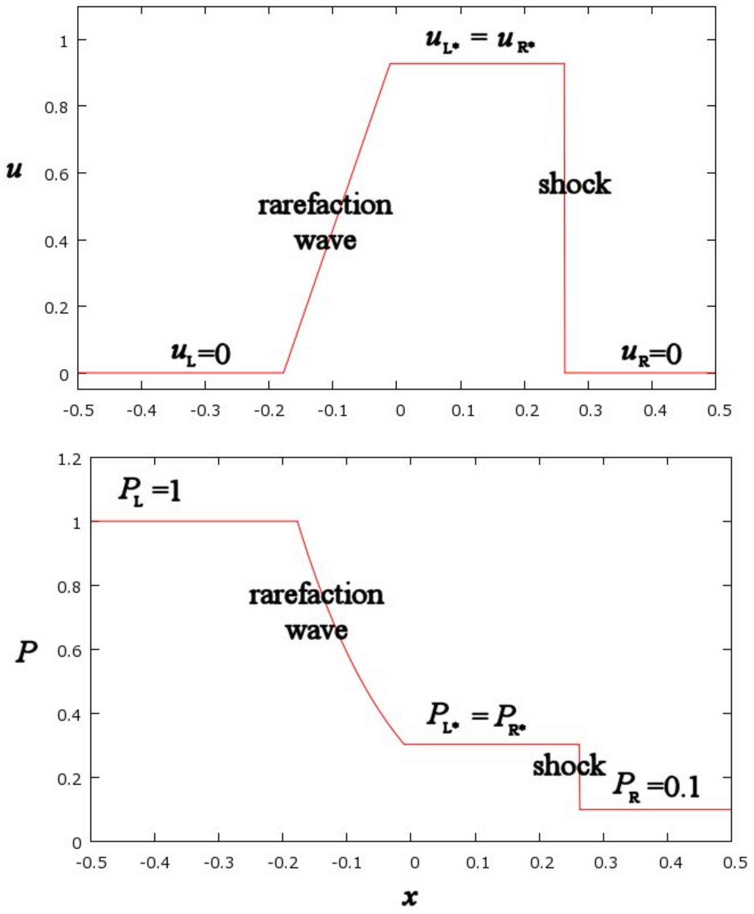
Here we summarize the general property of the solution as follows.



- Four constant states U_L, U_{L*}, U_{R*} and U_R (see the graph above).
- A contact discontinuity between U_{L*} and U_{R*} across which the velocity and pressure remain constant.
- U_L and U_{L*} are joined by either a shock or rarefaction wave; U_R and U_{R*} are joined by either a shock or rarefaction wave.
- Values of S_L, S_0 and S_R are not, in general, the characteristic speeds given by the eigenvalues of $A(U)$.

Example





3 High Resolution Shock Capturing Schemes

HRSC schemes are numerical schemes for solving a system of PDE written in conservative form

$$\partial_t U + \boldsymbol{\nabla} \cdot \boldsymbol{F}(U) = S(U). \tag{48}$$

Here we will only discuss the 1D equations and numerical schemes accurate to first order in space and time.

3.1 Discretisation

Computational domain: $x_L \leq x \leq x_R$.

Discretise the interval $[x_L, x_R]$ into N equal cells. Each cell has length

$$\Delta x = \frac{x_R - x_L}{N}. \tag{49}$$

The location of cell centers are

$$x_i = x_L + \left(i - \frac{1}{2}\right)\Delta x \quad , \quad i = 1, 2, \cdots, N. \tag{50}$$

The cell interfaces are located at

$$x_{i+\frac{1}{2}} = x_L + i\Delta x \quad , \quad i = 0, 2, \cdots, N. \tag{51}$$

3.2 Conservative methods

Consider a system of PDE written in the form

$$\partial_t U + \partial_x F(U) = 0. \tag{52}$$

In the finite volume approach, the cell-averaged value of U is evolved. Denote

$$U_i^n = \frac{1}{\Delta x}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} U(x, t_n)dx. \tag{53}$$

Given the data U_i^n ($i = 1, 2, \cdots, N$), we want to calculate U_i^{n+1}, the cell-averaged U at $t = t_{n+1}$ for $i = 1, 2, \cdots, N$. Denote the time-averaged flux at the cell interface as

$$f_{i+\frac{1}{2}} = \frac{1}{\Delta t_n}\int_{t_n}^{t_{n+1}} F(U(x_{i+\frac{1}{2}}, t))dt, \tag{54}$$

where $\Delta t_n = t_{n+1} - t_n$. Integrate (52) in space from $x_{i-\frac{1}{2}}$ to $x_{i+\frac{1}{2}}$ and in time from t_n to t_{n+1}:

$$\int_{t_n}^{t_{n+1}} dt \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} dx(\partial_t U + \partial_x F) = 0. \tag{55}$$

Using the definitions in (53) and (54) gives the finite volume equation

$$U_i^{n+1} = U_i^n - \frac{\Delta t_n}{\Delta x}(f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}}). \tag{56}$$

Note that equation (56) is exact. No approximation has been made so far.

A conservative scheme for evolving equation (52) is a numerical method based on eqn. (56) with $f_{i+\frac{1}{2}}$ taking the form

$$f_{i+\frac{1}{2}} = f_{i+\frac{1}{2}}(U_{i-k_L}^n, \cdots, U_{i+k_R}^n), \tag{57}$$

where k_L and k_R are two non-negative integers. This is called the numerical flux, an approximation to the exact flux in (54).

3.2.1 Method of Godunov

Let $\tilde{U}(x,t)$ with $t > t_n$ be the solution of the Riemann problem

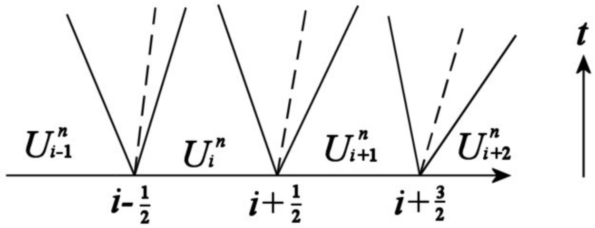
$$\text{PDE: } \partial_t U + \partial_x F(U) = 0 \tag{58}$$

$$\text{IC: } U(x,t_n) = \begin{cases} U_i^n & \text{if } x < x_{i+\frac{1}{2}} \\ U_{i+1}^n & \text{if } x > x_{i+\frac{1}{2}} \end{cases} . \tag{59}$$

Godunov's method is to compute the numerical flux according to

$$f_{i+\frac{1}{2}} = \frac{1}{\Delta t_n} \int_{t_n}^{t_{n+1}} F(\tilde{U}(x_{i+\frac{1}{2}},t))dt. \tag{60}$$

This method is first-order accurate.



Hence to compute the flux $f_{i+\frac{1}{2}}$, we need to solve the local Riemann problem at each cell interface. In general, there are multiple waves emerging from each cell interface (see the diagram above). In eqn. (60), there is an implicit assumption that the value of the local Riemann problem solution along the intercell boundary is constant. This means that the fastest wave at a given time travels for less than one cell length Δx. This places an restriction on the timestep Δt_n. Denote by $S^n_{\max}$ the maximum wave speed throughout the domain at time $t = t_n$. The maximum timestep is given by

$$\Delta t_n^{\max} = \frac{\Delta x}{S^n_{\max}}. \tag{61}$$

One may set

$$\Delta t_n = C_F \Delta t_n^{\max}, \tag{62}$$

where $0 < C_F < 1$ is called the Courant factor.

Example:

Consider the linear advection equation

$$\partial_t u + \partial_x F(u) = 0 \tag{63}$$

with $F(u) = \lambda u$, where λ is a constant. The solution of the Riemann problem for (63) with the IC

$$u(x,t_n) = \begin{cases} u_i^n & \text{if } x < x_{i+\frac{1}{2}} \\ u_{i+1}^n & \text{if } x > x_{i+\frac{1}{2}} \end{cases} \tag{64}$$

is

$$\tilde{u}(x_{i+\frac{1}{2}}, t) = \begin{cases} u_i^n & \text{if } \lambda > 0 \\ u_{i+1}^n & \text{if } \lambda < 0 \end{cases} . \tag{65}$$

Hence

$$f_{i+\frac{1}{2}} = \begin{cases} \lambda u_i^n & \text{if } \lambda > 0 \\ \lambda u_{i+1}^n & \text{if } \lambda < 0 \end{cases} \tag{66}$$

and eqn. (56) becomes

$$u_i^{n+1} = \begin{cases} u_i^n - \frac{\lambda \Delta t_n}{\Delta x}(u_i^n - u_{i-1}^n) & \text{if } \lambda > 0 \\ \\ u_i^n - \frac{\lambda \Delta t_n}{\Delta x}(u_{i+1}^n - u_i^n) & \text{if } \lambda < 0 \end{cases} , \tag{67}$$

which is known as the first-order upwind scheme.

3.2.2 HLL Riemann solver

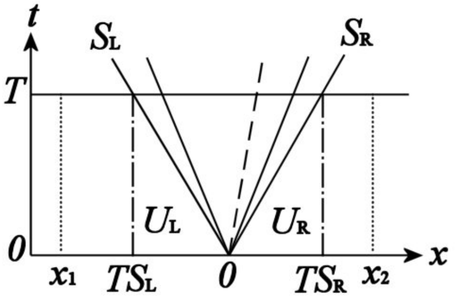
A Riemann problem is not always easy to solve and may require intense computation. An alternative is to devise a scheme to solve it approximately. Here we introduce one of the approximate Riemann solvers called the HLL (named after Harten, Lax and van Leer) Riemann solver.

Consider the Riemann problem

$$\text{PDE: } \partial_t U + \partial_x F(U) = 0 \tag{68}$$

$$\text{IC: } U(x, 0) = \begin{cases} U_L & \text{if } x < 0 \\ U_R & \text{if } x > 0 \end{cases} . \tag{69}$$

Consider the spacetime region $[x1, x2] \times [0, T]$ with $x_1 < TS_L$ and $x_2 > TS_R$ (see the diagram below), where S_L and S_R are the fastest signal velocities perturbing the initial states U_L and S_R, respectively.



Assume that $S_R > 0$ and $S_L < 0$. Integrate eqn. (68) over x from x_1 to x_2 and over t from 0 to T

$$\int_0^T dt \int_{x_1}^{x_2} dx[\partial_t U + \partial_x F(U)] = 0 \tag{70}$$

yields

$$\int_{x_1}^{x_2} U(x, T)dx = \int_{x_1}^{x_2} U(x, 0)dx + \int_0^T F(U(x_1, t))dt - \int_0^T F(U(x_2, t))dt. \tag{71}$$

Evaulate each term using the IC and the conditions $x_1 < TS_L$ and $x_2 > TS_R$.
LHS:

$$\begin{aligned}\int_{x_1}^{x_2} U(x,T)dx &= \int_{x_1}^{TS_L} U(x,T)dx + \int_{TS_L}^{TS_R} U(x,T)dx + \int_{TS_R}^{x_2} U(x,T)dx \\ &= (TS_L - x_1)U_L + \int_{TS_L}^{TS_R} U(x,T)dx + (x_2 - TS_R)U_R. \quad (72)\end{aligned}$$

First term on RHS:

$$\int_{x_1}^{x_2} U(0,x)dx = x_2 U_R - x_1 U_L. \quad (73)$$

Second term on RHS:

$$\int_0^T F(U(x_1,t))dt = TF(U_L) = TF_L. \quad (74)$$

Third term on RHS:

$$\int_0^T F(U(x_2,t))dt = TF(U_R) = TF_R. \quad (75)$$

Here

$$F_L = F(U_L) \quad , \quad F_R = F(U_R). \quad (76)$$

Combining all the terms yields

$$\int_{TS_L}^{TS_R} U(x,T)dx = T(S_R U_R - S_L U_L + F_L - F_R) \quad (77)$$

or

$$\frac{1}{T(S_R - S_L)} \int_{TS_L}^{TS_R} U(x,T)dx = \frac{S_R U_R - S_L U_L + F_L - F_R}{S_R - S_L}. \quad (78)$$

This means that the integral average of the exact solution of the Riemann problem between the slowest and fastest signals at time T is a known constant given by the right-hand-side of the above equation. Denote this average by U^{hll}:

$$U^{hll} = \frac{S_R U_R - S_L U_L + F_L - F_R}{S_R - S_L}. \quad (79)$$

Consider the integral of eqn. (68) over x from TS_L to 0^- and over t from 0 to T:

$$\int_0^T dt \int_{TS_L}^{0^-} dx[\partial_t U + \partial_x F(U)] = 0, \quad (80)$$

which gives

$$\int_{TS_L}^{0^-} U(x,T)dx = \int_{TS_L}^{0^-} U(x,0)dx - \int_0^T F(U(0^-,t))dt + \int_0^T F(U(TS_L,t))dt$$

$$= \quad -TS_LU_L + T(F_L - F_{0L}), \tag{81}$$

where

$$F_{0L} = \frac{1}{T}\int_0^T F(U(0^-,t))dt. \tag{82}$$

Similarly, integrating eqn. (68) over x from 0^+ to TS_R and over t from 0 to T gives

$$\int_{0^+}^{TS_R} U(x,T)dx = TS_RU_R + T(F_{0R} - F_R), \tag{83}$$

where

$$F_{0R} = \frac{1}{T}\int_0^T F(U(0^+,t))dt. \tag{84}$$

Adding eqns. (81) and (83) yields

$$\int_{TS_L}^{TS_R} U(x,T)dt = T(S_RU_R - S_LU_L + F_L - F_R + F_{0R} - F_{0L}). \tag{85}$$

Combining this equation with eqn. (77) gives

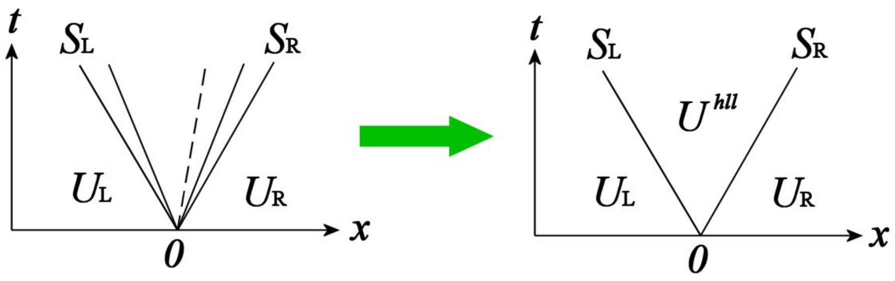
$$F_{0L} = F_{0R} \equiv F_0. \tag{86}$$

From eqns. (81) and (83) one obtains

$$\begin{aligned} F_0 &= F_L - S_LU_L - \frac{1}{T}\int_{TS_L}^{0} U(x,T)dt & (87) \\ &= F_R - S_RU_R + \frac{1}{T}\int_0^{TS_R} U(x,T)dt. & (88) \end{aligned}$$

No approximation has been made so far. The HLL approximate Riemann solver approximates the Riemann solution $\tilde{U}(x,t)$ by

$$\tilde{U}(x,t) \approx \begin{cases} U_L & \text{if } x < S_Lt \\ U^{hll} & \text{if } S_Lt < x < S_Rt \\ U_R & \text{if } x > S_Rt \end{cases} . \tag{89}$$



Note that this approximation consists of just three constant states separated by two fastest waves of speeds S_L and S_R. All intermediate states separated by intermediate waves are lumped into the single state U^{hll}. The corresponding flux F_0 is given by (87) or (88) as

$$F_0^{hll} = F_L + S_L(U^{hll} - U_L) = F_R + S_R(U^{hll} - U_R). \tag{90}$$

Use of (79) in (90) gives the HLL flux

$$F_0^{hll} = \frac{S_R F_L - S_L F_R + S_L S_R (U_R - U_L)}{S_R - S_L}. \tag{91}$$

Note that this formula is derived by assuming that $S_L < 0$ and $S_R > 0$. Obviously, if $S_L < 0$, $F_0 = F_L$; and if $S_R < 0$, $F_0 = F_R$. Hence

$$F_0^{hll} = \begin{cases} F_L & \text{if } S_L \geq 0 \\ \frac{S_R F_L - S_L F_R + S_L S_R (U_R - U_L)}{S_R - S_L} & \text{if } S_L < 0 < S_R \\ F_R & \text{if } S_R \leq 0 \end{cases}. \tag{92}$$

Define two quantities $c_{\max}$ and $c_{\min}$ as

$$c_{\max} = \max(0, S_R) \quad , \quad c_{\min} = \min(0, -S_L). \tag{93}$$

Equation (92) can be written as

$$F_0^{hll} = \frac{c_{\max} F_L + c_{\min} F_R - c_{\max} c_{\min} (U_R - U_L)}{c_{\max} + c_{\min}}. \tag{94}$$

The HLL Riemann solver provides a prescription for the numerical flux $f_{i+\frac{1}{2}}$. Denote

$$F_R = F(U_{i+1}^n) \quad , \quad F_L = F(U_i^n). \tag{95}$$

Then

$$f_{i+\frac{1}{2}}^{hll} = \frac{c_{\max} F_L + c_{\min} F_R - c_{\max} c_{\min} (U_R - U_L)}{c_{\max} + c_{\min}}. \tag{96}$$

Estimate of S_R and S_L

The HLL flux requires the values of S_R and S_L for the Riemann problem at each cell interface. One of the most popular approaches for estimating bounds for S_L and S_R is based on the eigenvalues of the Jacobian matrix $A(U) = \partial F / \partial U$. Denote $\lambda_{\max}(U)$ and $\lambda_{\min}(U)$ as the maximum and minimum eigenvalues of $A(U)$. At the cell interface $i + \frac{1}{2}$, one may set

$$S_R = \max[\lambda_{\max}(U_i^n), \lambda_{\max}(U_{i+1}^n)] \quad , \quad S_L = \min[\lambda_{\min}(U_i^n), \lambda_{\min}(U_{i+1}^n)]. \tag{97}$$

For example, in the 1D fluid dynamics equation, $\lambda_{\max} = u + c$ and $\lambda_{\min} = u - c$. Hence one may set

$$S_R = \max(u_i^n + c_i^n, u_{i+1}^n + c_{i+1}^n) \quad , \quad S_L = \min(u_i^n - c_i^n, u_{i+1}^n - c_{i+1}^n). \tag{98}$$

3.2.3 PDEs with source term

So far the focus has been on the homogeneous equation $\partial_t U + \partial_x F = 0$. Now consider a PDE with a source term

$$\partial_t U + \partial_x F(U) = S(U). \tag{99}$$

One approach is simply to add the source term to the solution of the homogeneous equation:

$$U_i^{n+1} = U_i^n - \frac{\Delta t_n}{\Delta x}(f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}}) + \Delta t_n S(U_i^n), \tag{100}$$

where the flux $f_{i+\frac{1}{2}}$ is computed by the solution of the (approximate) Riemann solver.

Since the Riemann solution does not take into account the presence of the source term, this simple approach might lead to unphysical results due to the mismatch in the treatment of flux and source terms. However, the problem is largely reduced when sufficient resolution is used in many cases. Other techniques of handling the source term can be found in Chapter 15 of Toro's book and the references therein.

3.2.4 Boundary conditions

Boundary conditions are needed for U_1^{n+1} (or $f_{\frac{1}{2}}$) and U_N^{n+1} (or $f_{N+\frac{1}{2}}$). There are various ways of imposing boundary conditions depending on the specific problems one wants to solve. Here we only mention three of them. In the following, Y denotes U^{n+1} or f, and Y_b denotes U^{n+1} or f at the boundary points, i.e. Y_b can be U_1^{n+1}, U_N^{n+1}, $f_{\frac{1}{2}}$ or $f_{N+\frac{1}{2}}$.

Prescribed BC: The boundary points of U^{n+1} (or f) are determined by a specified function $B(t)$, i.e.

$$Y_b = B(t_{n+1}). \tag{101}$$

The nature of $B(t)$ depends on the specific problem, e.g. pumping of material into the domain from the boundaries in a specified way.

Copy BC: Value of Y_b is copied from the nearest point, which corresponds to the BC that $\partial_x Y$ vanishes at the boundary. This BC is also called the transmissive BC. It is a numerical attempt to produce boundaries that allow the passage of waves without any effect on them.

Extrapolation: Value of Y_b are extrapolated from the nearby points, which is a generalization of the copy BC.

Reflective BC: Consider the boundary on the left and suppose it physically consists of a fixed, reflective impermeable wall. The reflective BCs for the fluid variables ρ, u and P are

$$u_1^{n+1} = -u_2^{n+1} \quad , \quad \rho_1^{n+1} = \rho_2^{n+1} \quad , \quad P_1^{n+1} = P_2^{n+1}. \tag{102}$$

Basic Ingredients of higher-order Finite Volume methods

1) Derive a conservation-law formalism for the equations of motion of the conserved variables U: $\partial_t U + \partial_x F(U) = S(U)$ (1)

2) Reconstruct the Primitive (P) variables to obtain the left and right states across cell interfaces: $P_L = P_{i+1/2-\varepsilon}, \; P_R = P_{i+1/2+\varepsilon}$

3) Solve the Riemann problem (approximately) to obtain the numerical flux at the cell interface: $F_{i+1/2}$

4) Compute the source and the numerical fluxes to get: $S_i = S(U_i^n)$ $(F_{i+1/2} - F_{i-1/2}) / \Delta x$, i.e., compute the RHS of (1)

5) Update the conservative by solving the ODE: $\partial_t U = RHS$

6) Invert the conservative variables to recover the primitive variables and apply boundary conditions

Reconstruction methods

- Piecewise linear reconstruction

$$P_L = P_i + \nabla P_i \Delta x / 2$$

$$P_R = P_{i+1} - \nabla P_{i+1} \Delta x / 2$$

where ∇P_i is called the slope limited gradient.

- Slope limiters are important to avoid the spurious oscillations that would otherwise occur with high order spatial discretization schemes due to shocks

- There exist multiple slope limiters here we will present two

Reconstruction methods

- Minmod slope limiter (Roe 1986)

$$\nabla P_i = \frac{1}{\Delta x} minmod\left(P_{i+1} - P_i, P_i - P_{i-1}\right)$$

where

$$minmod\left(a, b\right) = \begin{matrix} 0\,,\ if & ab < 0 \\ a\,,\ if\ |a| < |b|, & ab > 0 \\ b\,,\ if\ |a| > |b|, & ab > 0 \end{matrix}$$

Reconstruction methods

- Monotonized central (MC) slope limiter (Van Leer 1977)

$$\nabla P_i = \frac{1}{\Delta x} MC\left(P_{i+1} - P_i, P_i - P_{i-1}\right)$$

where

$$MC(a,b) = \begin{matrix} 0 & , if\ ab < 0 \\ sign(a) min\left(2|a|, 2|b|, |a+b|/2\right) & , otherwise \end{matrix}$$

- Piecewise linear reconstruction with MC/minmod limiters are 2nd-order accurate for smooth flows and become 1st-order accurate at shocks and extrema of P.

- Higher-order reconstruction: PPM (Colella & Woodward 1984), WENO(5) (Jiang & Shu 1986)

4 Recipe for evolving the 1D hydrodynamics equations

In this section, a recipe is given for solving the 1D hydrodynamics equations using the HRSC scheme with the HLL Riemann solver. The numerical techniques introduced in the previous section are applied to this specific system of PDEs.

4.1 Plane symmetry

Recall the equations of hydrodynamics in 1D with plane symmetry are

$$\partial_t U + \partial_x F(U) = 0 \tag{103}$$

with the conservative variables U and fluxes F given by

$$U = \begin{pmatrix} U_1 \\ U_2 \\ U_3 \end{pmatrix} = \begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix} , \tag{104}$$

$$F = \begin{pmatrix} \rho u \\ \rho u^2 + P \\ (E+P)u \end{pmatrix} . \tag{105}$$

Since many formulae are expressed in terms of the primitive variables ρ, u and P, it is convenient to compute them from the conservative variables U_1, U_2 and U_3:

$$\rho = U_1 \quad , \quad u = U_2/U_1 \quad , \quad P = (\gamma - 1)\left(U_3 - \frac{U_2^2}{2U_1} \right) . \tag{106}$$

Recipe for computing U_i^{n+1} given U_i^n

1. Determine the timestep Δt_n by

$$\Delta t_n = C_F \frac{\Delta x}{\max(|u_j^n + c_j^n|, |u_j^n - c_j^n|)} , \tag{107}$$

where $0 < C_F < 1$ is the Courant factor and the maximum are taken over all cells $j = 1, 2, \cdots N$. The sound speed c_j^n is $c_j^n = \sqrt{\gamma P_j^n / \rho_j^n}$.

2. For each cell $i = 1, 2, \cdots , N-1$:

   (a) Compute the fluxes $F_L = F(U_i^n)$ and $F_R = F(U_{i+1}^n)$.

   (b) Estimate the signal speeds S_L and S_R according to

$$S_R = \max(u_i^n + c_i^n, u_{i+1}^n + c_{i+1}^n) \tag{108}$$

$$S_L = \min(u_i^n - c_i^n, u_{i+1}^n - c_{i+1}^n). \tag{109}$$

   and then compute $c_{\max}$ and $c_{\min}$ by

$$c_{\max} = \max(0, S_R) \quad , \quad c_{\min} = \min(0, -S_L). \tag{110}$$

(c) Calculate the HLL flux

$$f^{hll}_{i+\frac{1}{2}} = \frac{c_{\max}F_L + c_{\min}F_R - c_{\max}c_{\min}(U_R - U_L)}{c_{\max} + c_{\min}}. \tag{111}$$

3. For each cell $i = 2, 3, \cdots, N-1$, compute U_i^{n+1} according to

$$U_i^{n+1} = U_i^n - \frac{\Delta t_n}{\Delta x}(f^{hll}_{i+\frac{1}{2}} - f^{hll}_{i-\frac{1}{2}}). \tag{112}$$

4. Apply BC on U_1^{n+1} and U_N^{n+1}.

Recipe for computing U_i^{n+1} given U_i^n

1. Determine the timestep Δt_n by

$$\Delta t_n = C_F \frac{\Delta r}{\max(|u_j^n + c_j^n|, |u_j^n - c_j^n|)}, \tag{113}$$

where $0 < C_F < 1$ is the Courant factor and the maximum are taken over all cells $j = 1, 2, \cdots N$. The sound speed c_j^n is $c_j^n = \sqrt{\gamma P_j^n / \rho_j^n}$.

2. For each cell $i = 1, 2, \cdots, N-1$:

(a) Compute the fluxes $F_L = F(U_i^n)$ and $F_R = F(U_{i+1}^n)$.

(b) Estimate the signal speeds S_L and S_R according to

$$S_R = \max(u_i^n + c_i^n, u_{i+1}^n + c_{i+1}^n) \tag{114}$$
$$S_L = \min(u_i^n - c_i^n, u_{i+1}^n - c_{i+1}^n). \tag{115}$$

and then compute $c_{\max}$ and $c_{\min}$ by

$$c_{\max} = \max(0, S_R) \quad , \quad c_{\min} = \min(0, -S_L). \tag{116}$$

(c) Calculate the HLL flux

$$f^{hll}_{i+\frac{1}{2}} = \frac{c_{\max}F_L + c_{\min}F_R - c_{\max}c_{\min}(U_R - U_L)}{c_{\max} + c_{\min}}. \tag{117}$$

3. Set $f^{hll}_{\frac{1}{2}} = 0$.

4. For each cell $i = 1, 2, \cdots, N-1$, compute U_i^{n+1} according to

$$U_i^{n+1} = U_i^n - \frac{\Delta t_n}{\Delta r}(f^{hll}_{i+\frac{1}{2}} - f^{hll}_{i-\frac{1}{2}}) + \Delta t_n S(U_i^n). \tag{118}$$

5. Apply BC on U_N^{n+1}.

4.2 Numerical Tests

Here we illustrate the performance of the first-order HLL scheme on the 1D hydrodynamics equation with plane symmetry and spherical symmetry. In all the tests shown below, the adiabatic index of the gas is set to $\gamma = 1.4$. Copy boundary conditions are used in all simulations.

Although the numerical scheme evolves the conservative variables U, we are actually interested in the primitive variables X:

$$X = \begin{pmatrix} \rho \\ u \\ P \end{pmatrix}. \tag{119}$$

The initial conditions, as well as the final results, are expressed in terms of X instead of U.

4.2.1 Plane symmetry tests

The two tests shown below solves the Riemann problem with the IC

$$X = \begin{cases} X_L & \text{if } x > x_0 \\ X_R & \text{if } x < x_0 \end{cases}, \tag{120}$$

where x_0 is a constant parameter.

Numerical solutions are computed by the first-order HLL scheme described in section 4.1. Exact solutions are computed by the numerical code given in the end of Chapter 4 of Toro's book.

Test 1

Computation domain: $[0, 1]$

Initial conditions:

$\rho_L = 1$, $u_L = 0.75$, $P_L = 1$;

$\rho_R = 0.125$, $u_R = 0$, $P_R = 0.1$.

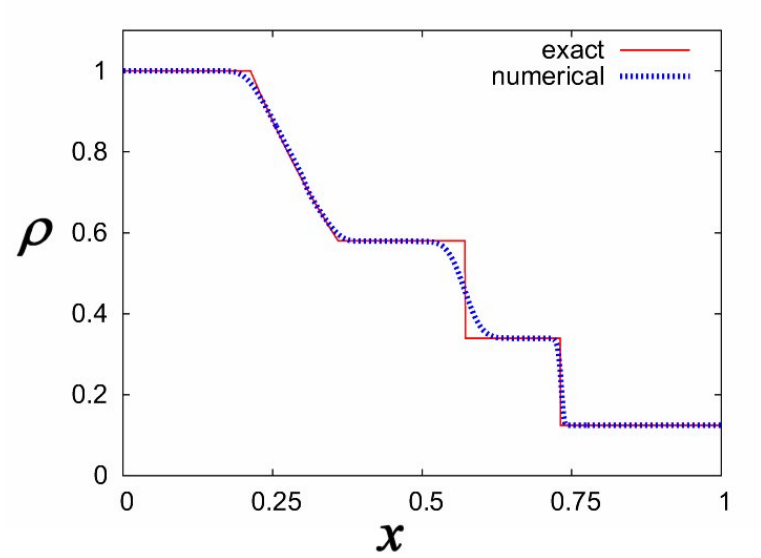
Position of discontinuity: $x_0 = 0.3$

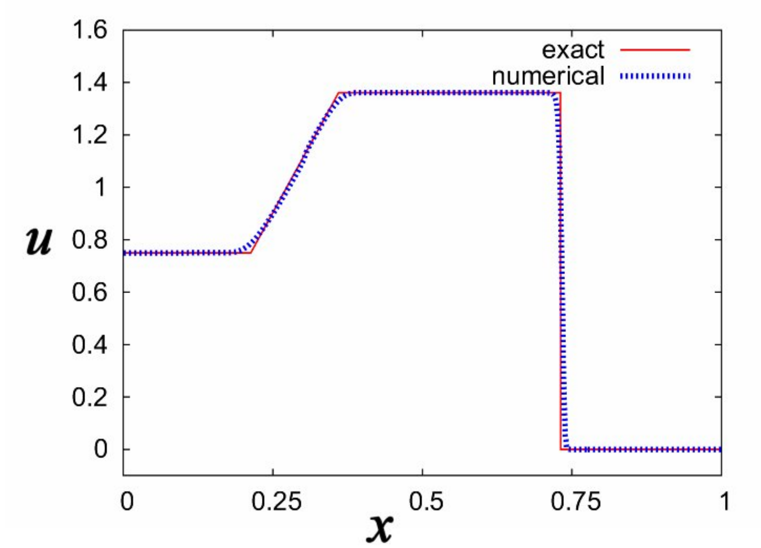
Courant factor: $C_F = 0.5$

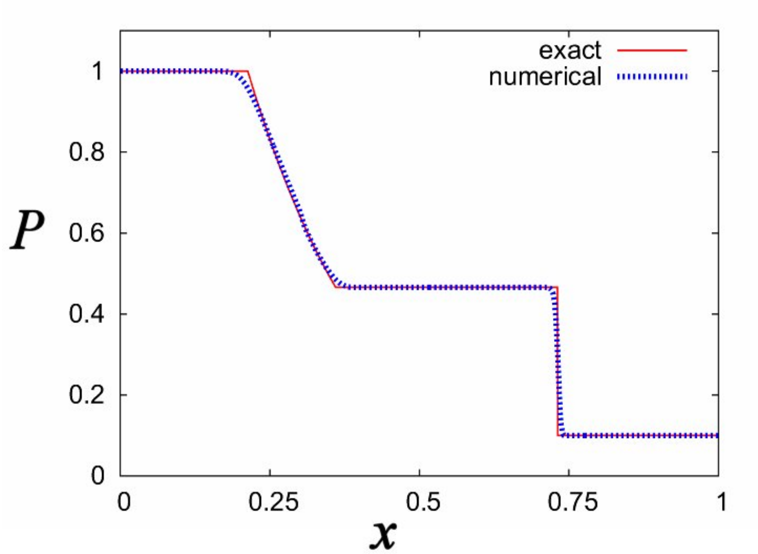
Number of cells: $N = 400$

Simulation time: $t_{\text{final}} = 0.2$

Results: The solution has a right shock wave, a right traveling contact wave and a left rarefaction wave. The graphs below show the comparison between numerical and exact solutions.


exact
numerical
ρ
1
0.8
0.6
0.4
0.2
0
0
0.25
0.5
0.75
1
x


exact
numerical
u
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
0.25
0.5
0.75
1
x



Test 2

Computation domain: $[0, 1.2]$

Initial conditions:

$\rho_L = 1$, $u_L = 0$, $P_L = 1000$;

$\rho_R = 1$, $u_R = 0$, $P_R = 0.01$.

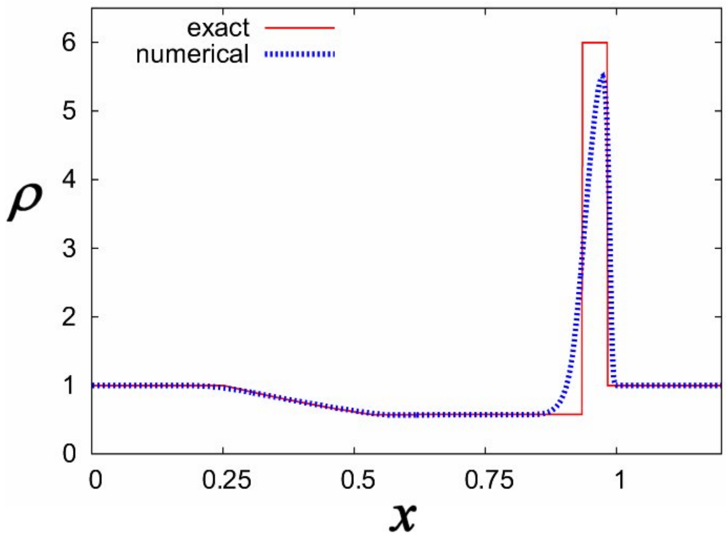
Position of discontinuity: $x_0 = 0.7$

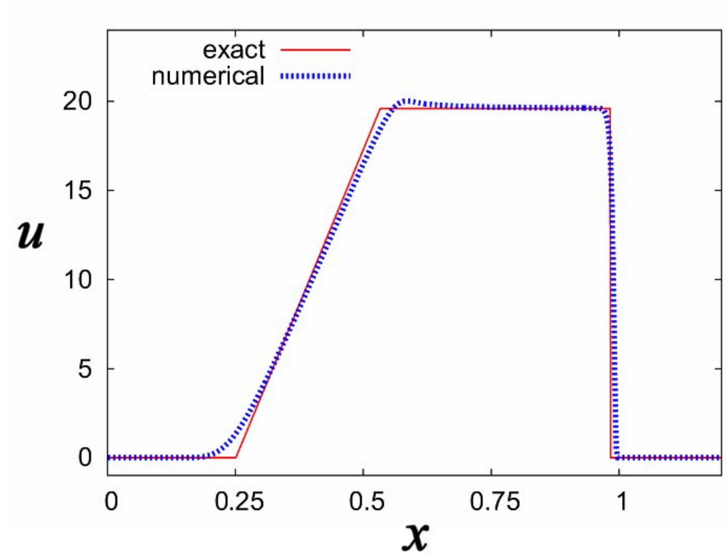
Courant factor: $C_F = 0.5$

Number of cells: $N = 400$

Simulation time: $t_{\text{final}} = 0.012$

Results: This is a severe test. The solution consists of a strong shock wave of Mach numerical 198, a contact surface and a left rarefaction wave. The graphs below show the comparison between numerical and exact solutions.


exact
numerical
ρ
6
5
4
3
2
1
0
0
0.25
0.5
0.75
1
x


exact
numerical
u
20
15
10
5
0
0
0.25
0.5
0.75
1
x