A Practical Introduction to Numerical Hydrodynamics

Garrelt Mellema
Stockholm Observatory, Stockholm University

1 Introduction

This exercise is an introduction to the field of numerical hydrodynamics. It will hopefully give you some insight in what is involved in such calculations. Numerical hydrodynamics is used in many parts of astrophysics. The applications we consider in this exercise are connected to stars, but this type of models could also apply to for instance galaxies, their cores, disks, and even clusters of galaxies.

1.1 The Euler equations

The behaviour of an ideal compressible gas is described by the *Euler equations*:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{v} = 0 \tag{1}$$

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot \rho \mathbf{v}\mathbf{v} + \nabla p = 0 \tag{2}$$

$$\frac{\partial e}{\partial t} + \nabla \cdot (e + p)\mathbf{v} = 0 \ , \tag{3}$$

with

$$e = \frac{1}{2}\rho v^2 + e_{\rm int} \ , \tag{4}$$

the internal energy of the gas.

Reducing these to one spatial dimension and using cartesian coordinates one can write them as

$$\frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = 0 \ , \tag{5}$$

with $\mathbf{W}$ the so called state vector (ρ, ρv, e)$^{\rm T}$, and $\mathbf{F}$ the flux vector (ρv, $\rho v^2 + p$, $(e + p)v^{\rm T}$). The elements of the state vector are also called the *conserved quantities* since the Euler equations basically say that mass, momentum and energy are conserved. The variables ρ, v, and p are often referred to as the *primitive variables*.

The Euler equations are not complete without an equation of state. We choose an ideal gas for which

$$e_{\rm int} = p/(\gamma - 1) \ , \tag{6}$$

where $\gamma = 5/3$ for a monatomic gas, and

$$p = \frac{\rho k T}{\mu m_{\rm H}} \ , \tag{7}$$

with μ the mean molecular weight.

A more general form of the Euler equations is

$$\frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{F}}{\partial \xi} = \mathbf{S} \ , \tag{8}$$

in which ξ now is an arbitrary spatial coordinate and $\mathbf{S}$ is the *source vector*. $\mathbf{S}$ can be split into two parts: geometrical source terms which arise in the case of non–cartesian coordinates, and physical source terms such as radiative heating and cooling, gravitation, etc.

We will be using the spherical coordinate r. For this the momentum equation is

$$\frac{\partial \rho v}{\partial t} + \frac{1}{r^2}\frac{\partial}{\partial r}(r^2 \rho v^2) = -\frac{\partial p}{\partial r} \ , \tag{9}$$

and similar expressions can be written down for the continuity and energy equations. To write this in the form of Eq. (8) we take

$$\mathbf{W} = r^2(\rho, \rho v, e)^{\rm T} \tag{10}$$

$$\mathbf{F} = r^2(\rho v, \rho v^2 + p, (e + p)v)^{\rm T} \tag{11}$$

$$\mathbf{S} = (0, 2rp, 0) = r^2(0, 2p/r, 0)^{\rm T} \ . \tag{12}$$

$\mathbf{W}$ is still the vector of conserved quantities vector since it is total mass, momentum and energy which are conserved, not their densities. In spherical coordinates a volume element is proportional to r^2 (really $r^2 \sin\theta$ but we are not considering

the θ coordinate here), so by multiplying the densities by the position dependent volume factor r^2, one obtains the conserved quantities.

One sees that choosing a non–cartesian coordinate leads to the appearance of a geometrical source term. There are actually several ways in which to split off the geometrical source terms, but here we choose the one shown in Eqs. (10)–(12).

1.2 Properties of the Euler equations

The Euler equations are a simpler version of the *Navier–Stokes equations*. The latter contain terms for the viscosity and thermal conductivity of the gas. In astrophysics these are normally not thought to be important.

Using the Euler equations also implies using a *fluid approximation*, i.e. the particles interact with each other sufficiently to establish a Maxwell–Boltzmann distribution. This approximation is mostly valid, but there are exceptions.

From a mathematical point of view the Euler equations are a set of non-linear, coupled, hyperbolic differential equations. Hyperbolic differential equations have two important properties:

- they allow discontinuous solutions; in physical terms this means that the flow can contain shocks or contact discontinuities.

- one can define so called *characteristics* or *characteristic speeds*. These are the eigenvalues of the problem: the solution can be written in terms of a sum of eigenvectors, three in the case of a one–dimensional problem. The three eigenvectors are also called *waves* and are physically associated with the characteristic speeds v, $v - s$, $v + s$, the velocity of the flow, and the velocity of sound added and subtracted. The physical relevance of this is that in a gas no signal can travel faster than the local sound speed, and $v - s$ and $v + s$ are the highest possible signal speed within a flow with velocity v. This also means that the characteristics delineate a domain of influence in space–time (see Fig. 1). Point Q can only influence the hashed region of space–time.

Physically, shocks can be thought to occur because the particles suddenly have to adjust to a new situation. There is therefore a close relation between the characteristics and the shocks: if for example an explosion occurs at point A, its effect will spread with the characteristic speeds $v - s$ and $v + s$, or in other words $v - s$ and $v + s$ are the shock speeds.

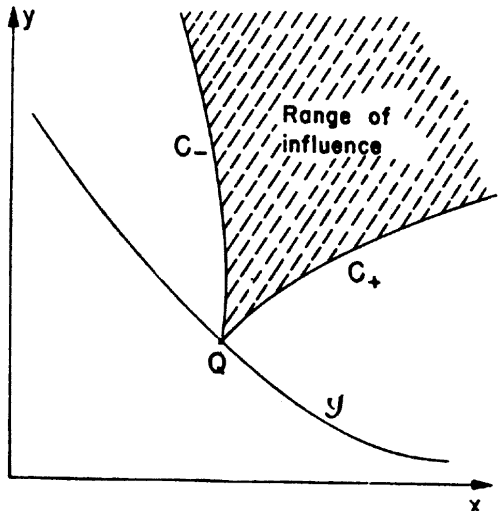

Figure 1: The domain of influence in the (x, y) space

1.3 Numerical approach

Solving the Euler equations analytically is only possible in a few simple cases. For more general solutions one has to use a numerical approach. The first step is to discretise space and time:

$$\mathbf{W}(x,t) \rightarrow \mathbf{W}_i^n = \mathbf{W}(x_i, t_n) \; , \tag{13}$$

with $x_i = x_0 + i\Delta x$, $t_n = t_0 + n\Delta t$. Note that Δx and Δt can in principle be time and postion dependent.

Solving the Euler equations using normal methods for differential equations is not a good idea. One reason is that the solutions can contain shocks, i.e. very steep gradients, which are not handled well by these methods. Another reason is that the solutions should be conservative, no mass or energy should appear or disappear, and this is generally not guaranteed by these methods. Therefore special methods have been developed to deal with the Euler equations.

Gauss divergence theorem says that for a differential equation

$$\frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = 0 \; , \tag{14}$$

the following relation holds for closed contour C in space–time (x, t),

$$\oint_C (\mathbf{W}\mathrm{d}x + \mathbf{F}\mathrm{d}t) = 0 \, . \tag{15}$$

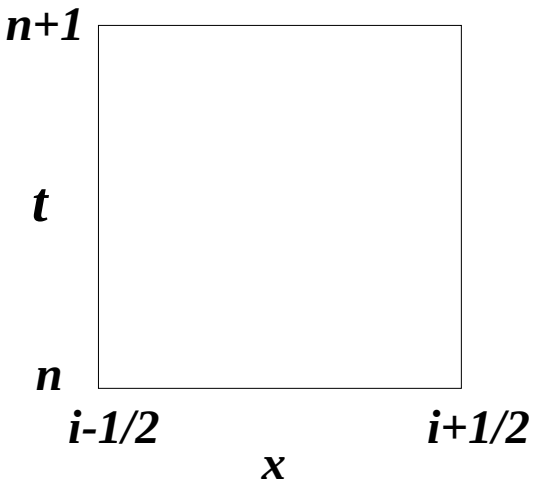

Figure 2: One grid cell in the (x, t) space

If one defines a contour in (x, t) space using one grid cell and one time step one obtains the expression

$$\mathbf{W}_i^{n+1} = \mathbf{W}_i^n + \frac{\Delta t}{\Delta x}\left(\mathbf{F}_{i-1/2} - \mathbf{F}_{i+1/2}\right) , \tag{16}$$

or in other words, $\frac{\Delta t}{\Delta x}\left(\mathbf{F}_{i-1/2} - \mathbf{F}_{i+1/2}\right)$ is the amount $\mathbf{W}_i$ changes in one time step Δt. $\mathbf{F}_{i-1/2}$ can be seen as the flux coming from the lefthand neighbour, and $\mathbf{F}_{i+1/2}$ as the flux coming from the righthand neighbour. Since one with this formalism is just moving mass, momentum and energy from one cell to another, the scheme is *conservative*. The trick is now to come up with a smart way to calculate the $\mathbf{F}_{i-1/2}$, the *interface fluxes*. Two criteria for the method are that shocks should be handled well and that the equations should be solved second order accurately in both space and time.

Once such a way is chosen, it can used for updating the state vector for consecutive timesteps, that way following the evolution of $\mathbf{W}$ with time. Choosing the time steps should be done carefully, see section 1.5.2.

For this we use a method called the *Roe solver*, based on the work of Roe (1981). A description of the ideas behind it (together with a lot of useful background on numerical hydrodynamics) can be found in his review paper from 1986 (ask for copies from me).

1.4 Roe solver routine

Since programming this solver is a lot of work and requires even more knowledge of the mathematics behind the equations than I described above, I provide subroutines which contains the Roe solver, one in FORTRAN, the other in C++. Both can be found at the following web address
`http://www.astro.su.se/~garrelt/Num_hydro`.

The FORTRAN version can be found in the file `roesol.f`. This subroutine takes as input

- `dr` (real*8) — the cell size
- `dt` (real*8)— the time step
- `gamma` (real*8)— the adiabatic index γ
- `vol` (real*8 array `(0:meshmax+1)`) — the volume factor of each cell (r^2 in the spherical case)
- `state` (real*8 array `(0:meshmax+1,3)`) — the state vector $(\rho, \rho v, e)$. Note that unlike the definition of $\mathbf{W}$ in Eq. 10, this state vector should *not* contain the volume terms.
- `flux` (real*8 array `(meshmax+1,3)`) — the interface fluxes for $i - 1/2$, see Eq. 11.
- `meshr` (integer*4) — size of the grid; should be smaller than `meshmax`.
- `ictl` (integer*4) — control variable ($\neq 0$ in case of errors).

The grid is assumed to be `meshr` cells, with `meshr` given in as an argument to the the subroutine. `meshr` should be smaller than the maximum size, `meshmax`, a parameter of the subroutine (currently set to 2000). The points `0` and `meshr+1` are the boundary points.

There is also a C++ version of the routine, which can be found in the file `roesol.cpp`. This version also needs the incldue files `Euler.h` and `EulerMath.h`.

1.5 Making a hydrodynamics code

To make a full program several more things are needed:

- Initial conditions

- Boundary conditions
- Time step calculation
- Source term integration

Items 3 and 4 are general, 1 and 2 depend on the specific problem you want to solve. Of course you will also need in– and output routines.

1.5.1 Choosing the time step

The maximum time step one can take is set by the assumption that one can consider each cell only to interact with its direct neighbours: the domain of influence of cell i should not reach cells $i-2$ and $i+2$! Since the maximum communication speed is $v-s$ or $v+s$, this gives a condition on the time step. This is known as the *Courant–Friedrichs–Lewy (CFL) condition*. If your time step exceeds this value, the method will become unstable. Since this maximum time step depends on v and s, this means that the time step is not fixed, but depends on the conditions on the grid, and should be re–calculated every time step. It is normally wise to take a fraction between 1 and 0.5 of this maximum time step.

1.5.2 Source term integration

The source terms fall outside the Roe solver formalism. It is in fact possible to integrate them into the Roe solver but we will not do that here. Instead we will use an approach called *operator splitting*. It is possible to show that an equation

$$\frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{F}}{\partial \xi} - \mathbf{S} = 0 \ , \tag{17}$$

can be numerically solved (to second order accuracy) by solving for each time step the two separate equations

$$\frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{F}}{\partial \xi} = 0 \tag{18}$$

$$\frac{\partial \mathbf{W}}{\partial t} - \mathbf{S} = 0 \ , \tag{19}$$

after each other. The first equation is dealt with by the Roe solver routine, the second one you will have to write an integration routine for. Since it is an ordinary differential equation, standard methods can be used. Do not use a very fancy

method however, realise that both the Roe solver and the splitting are only second order accurate, so it is not necessary to be more accurate than that when solving Eq. (19).

One can show that a more accurate (but perhaps unnecessary) approach to the splitting is to integrate Eq. 19 over $\frac{1}{2}\Delta t$, then Eq. 18 over Δt, then Eq. 19 over $\frac{1}{2}\Delta t$ again. This is known as *Strang splitting*.

1.5.3 Boundary conditions

The boundary conditions are usually dealt with using one or more *ghost cells*, in our Fortran case these are cells `0` and `meshmax+1`. Before the integration step the value for **W** is set in a pre–determined way, depending on the problem. Possible choices are:

- **Inflow:** The ghost cells are filled with the material flowing onto the grid.
- **Outflow:** The ghost cells are filled so that material can flow off the grid without influencing the material on the grid (normally achieved by setting **W** equal to the value of the last proper grid cell).
- **Reflection:** The ghost cells are filled so that it seems that the material runs into a wall (normally achieved by setting **W** equal to the value of the last proper grid cell but changing the sign of the velocity).

1.6 Testing the code

When developing the code, it is a good idea to run some simple tests, such as for example first testing a cartesian version (no source terms and volume factor equal to 1) on a pressure jump. These kind of tests are known as *shock tube tests*. A popular one is *Sod's shock tube test*, described for instance by Stone & Norman (1992), Sect. 5.4 (the tests in their Sects. 5.1–2 may also be useful).

2 Interacting winds

A) Construct a numerical hydrodynamics code to follow the interaction between a fast stellar wind and its slower precursor. These kinds of interactions take place when stars change from low effective temperature ('red') to high effective temperature ('blue'), for instance in the post–AGB phase (for low to intermediate mass

stars), and when massive stars change from red supergiants to blue supergiants (as happened with the precursor of SN1987A).

Use the following parameters: $\dot{M} = 10^{-7}\ M_\odot$ yr^{-1}, $v = 1000$ km s^{-1} for the fast wind, and $\dot{M} = 10^{-5}\ M_\odot$ yr^{-1}, $v = 10$ km s^{-1} for the slow wind. You may assume that all the gas is ionized by the UV radiation from blue star, and that it initially has the typical temperature of an ionized gas, 10^4 K. Start the interaction between the two winds at an initial radius of 10^{14} m. Choose the outer grid radius such that you can follow the interaction for 500 years or more.

The inner boundary condition should be an inflow condition, the outer boundary an outflow condition. The choice of the position of the inner boundary is up to you.

B) The interaction leads to a so called *three shock pattern*: an outer shock travelling into the slow wind, an inner shock travelling into the fast wind, and a contact discontinuity separating the two; these three boundaries mean that there are four regions in the flow. Identify the three discontinuities in the results of simulation and describe the structure of the four regions of the flow.

C) Compare the results of the simulation with analytical expectations. A full self–similar model was published by Chevalier & Imamura (1983), whereas Kahn (1983) used a simpler 'thin shell' model (which assumes efficient cooling, see Sect. 3). How good is the match between the two analytical models and your numerical model? Use these quantities to compare: the (pattern) speed of the outer shock, temperature of the shocked fast wind, position of the three discontinuities. In Chevalier & Imamura (1983) you will have to use Fig. 5 to derive the value of b_1 and b_2 for the parameters chosen here.

D) Study the effects of resolution. Use as few as 50 grid cells, and then increase the resolution by using more grid cells, up to 10 or 20 times as much (if CPU time allows). What are the differences?

E) Study the effect of reducing the order of the solution. In the Roe solver routine set the parameters `sbpar1` and `sbpar2` to zero. This will make the code only first order accurate. Compare the result of the simulation with your previous simulations. Has reducing the order of the solution a similar effect as reducing the resolution, or in other words can one compensate for the lower order solution by running at a higher resolution?

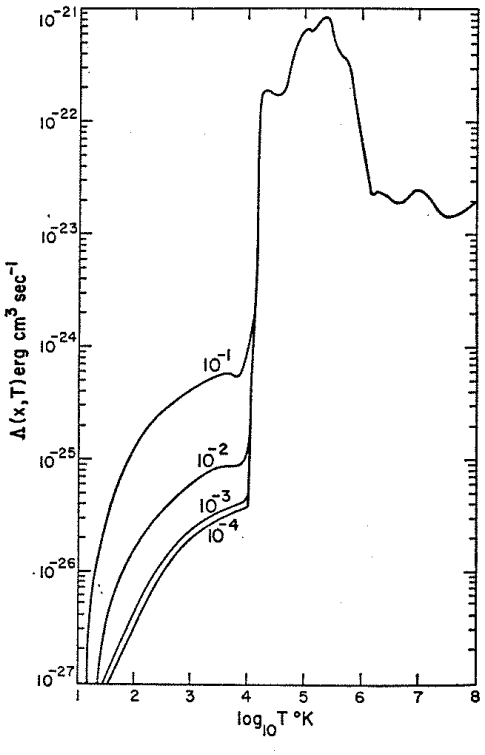

Figure 3: The coronal cooling curve from Dalgarno & McCray (1972)

3 Radiative cooling

Astrophysical gases tend to radiate when heated. This radiation is an energy leak which may carry away substantial amounts of energy. A simulation which does *not* include the radiative cooling (such as the one in the previous section) is often referred to as an *adiabatic* simulation.

The file `corocool.tab` (in anonymous ftp) contains what is called a *coronal cooling curve*. The file contains two columns, the first one lists the log of the temperature, the second one the log of the cooling Λ in units of erg cm^3 s^{-1}. This particular cooling curve is taken from Dalgarno & McCray (1972), reproduced here as Fig. 3. It was calculated by determining the equilibrium ionization state at every temperature and from that the amount of emission. Below 10^4 K the cooling depends strongly on the ionization fraction of the gas, the one in the table is the curve labelled 10^{-4} in the paper. The cooling rate of a gas can be approximated by $n^2\Lambda(T)$, where n is the number density of particles.

Numerically, the cooling is a source term for the energy density and can be included in a similar way as the geometric source terms, i.e. in a separate calculation every time step. However, there is the danger that the cooling will reduce the temperature/internal energy density below zero, which is unphysical. The reason for this is the very non-linear behaviour of the cooling function. There are two ways to control this:

- Limit the time step not only with the CFL condition, but also with the cooling time. Limit the change of internal energy to something like 10% per time step. This may lead to very short time steps, perhaps too short in be practical (experiment with this).
- Solve the equation implicitly, i.e. find the new energy density or temperature and use this to find a new and better estimate for the change in energy density. Implicit methods are described in any book on numerical methods, for example in *Numerical Recipes*.

The best solution for situations in which heating is solely by shocks (as we have here) is to try to resolve the so-called cooling region. In a shock the temperature is first raised to the post-shock value and then as it moves away from the shock it cools. If this *cooling region* is contained in one cell of your calculation you are not resolving this process and the solution will be some average temperature and density which could have little to do with the real average density and temperature in the cooling region. In this case you will also have to use extremely short time steps or an implicit solution. If however the cell size is small enough to follow the post-shock cooling you will not only have a more accurate representation of the post-shock region, but also much less problems with solving for the cooling, since the change in internal energy density will be small per time step. For safety reasons one can still limit the time step with the cooling time, but usually the cooling time will be of order of or less than the CFL time.
A) Solve the problem from Section 2 with radiative cooling included. Still assume that all material is ionized. What differences are there with the non-cooling solution? Why is the effect larger for the outer shock than for the inner shock? Is it possible to resolve the cooling region? What compression (density jump) do you find for the swept up shell?

If you are unsuccessful in getting the code to run with cooling, you can try lowering the overall density. This will reduce the cooling rate. You could for instance try $\dot{M} = 10^{-8}\ M_\odot$ yr^{-1} and $\dot{M} = 10^{-6}\ M_\odot$ yr^{-1} for the fast and slow wind respectively. A 10 times lower density gives a 100 times lower cooling. If

you reduce the density too much the result of the calculation will be the similar to the one without cooling.

B) Follow the interaction between a moderately fast wind and an outer slow wind. Use the following parameters: $\dot{M} = 10^{-7}\ M_\odot$ yr^{-1}, $v = 120$ km s^{-1} for the fast wind, and $\dot{M} = 10^{-6}\ M_\odot$ yr^{-1}, $v = 10$ km s^{-1} for the slow wind. First do a simulation without cooling and then add the cooling. Compare the two results and also compare them to the differences between the non-cooling and cooling simulation for the 1000 km s^{-1} wind. Why is the behaviour of the inner shock very different?

C) A weakly cooling bubble is often called *energy conserving* and a strongly cooling bubble *momentum conserving*. Lamers & Cassinelli (1999) state in Sect. 12.3 of their book that one can observationally check whether a bubble is momentum or energy conserving by considering the following ratios:

ϵ=(total kinetic energy in shell)/(total wind energy provided)

π=(momentum in the shell)/(momentum imparted by the wind)

If $\epsilon \approx 1$ the shell is energy conserving, whereas when $\pi \approx 1$ it is momentum conserving.

Check whether the shells in your models are indeed *energy conserving* and *momentum conserving* according to these definitions, by calculating ϵ and π for both cases.

References

[1] Chevalier R. A., Imamura J. N. 1983, ApJ 270, 554

[2] Dalgarno A., McCray R. 1972, ARA&A 10, 375

[3] Kahn F. D. 1983, in: IAU Symposium 103, Planetary Nebulae, D. R. Flower (ed.). Reidel, Dordrecht. p. 305

[4] Lamers, H. J. G. L. M., Cassinelli, J. P., 1999, Introduction to stellar winds. Cambridge Univ. Press.

[5] Roe P. L., 1981, J. Comp. Phys. 43, 357

[6] Roe P. L., 1986, Ann. Rev. Fluid Mech. 18, 337

[7] Stone J. M., Norman M. L., 1992, ApjS 80, 753