Astrophysics with the Computer: Smoothed Particle Hydrodynamics (SPH)

Joachim Köppen Kiel/Strasbourg 2008/09

1 Astrophysics

Many problems in astrophysics deal with the fluid dynamics: the formation and evolution of galaxies, stars, and planet systems, and the evolution of the interstellar and the intergalactic gas are examples. These problems require the solution of the hydrodynamical equations, which are partial differential equations and thus demand a lot of computer time. To solve the equations in a most efficient and accurate way, a number of methods have been devised. There are the approaches based on a fixed or adaptive spatial grid, such as the finite difference methods (our exercise on One-Dimensional Hydrodynamics uses this approach), which describe the variables of the gas distribution – density, velocity, temperature – at discrete positions in space. Usually one has to specify the geometry of this spatial grid and the symmetry: the collapse of a nonrotating star can be done in one spatial dimension, assumung spherical symmetry. Treating a system with rotational symmetry, sunch as an accretion disk or a spiral galaxy would require the greater expense of two dimensions. But dealing with configuration without any symmetry, such as the collision of two black holes or two galaxies require a three dimensional approach which constitutes an even greater number of computer operations. Evidently, the number of operations determines how fine the spatial grids can be made to obtain results in reasonable times, but this limits the fine structure one can study in that system.

In the SPH approach (for an introduction, see the first chapters of Monaghan, 1992) the properties of the fluid are represented by the distribution of a large number of particles which move freely in space, without the constraint of a discrete and thus finite spatial grid. The particles follow the forces formulated in such a way, that the average properties of the particles – such as their density – behave as closely as possible the equations of hydrodynamics. Thus, the particles do not represent real gas particles, but it's their average that matches the physics. Since there is no spatial grid to limit the resolution and no imposed symmetry of the problem, this method is well suited to tackle problems that are asymmetric or whose spatial symmetry cannot be easily predetermined.

Of course, the SPH approach has its limitations: the resolution depends on the number of particles ... In this exercise we want to get a first experience with this method. Initially we shall consider the simple one-dimensional problem, the linear shock tube. Its solution is well known, and of course it can also be computed with the finite difference methods. From this example, we can learn the behaviour, the advantages and the limitations of an SPH code, and how its computational parameters have to be chosen to get reliable results. If this works well and sufficiently fast, we can quite easily extend the code into two and three spatial dimensions and study other problems.

2 How to move many particles

The core of an SPH code is the propagation of particles under the influence of some force. Suppose that we know the acceleration (in the three components $a_i(t)$) of a particle at a certain time t, we can compute the change of its velocity components during a small time step Δt:

$$v_i t + \Delta t = v_i(t) + a_i(t) \times \Delta t$$

for all components. Here we assume that during that time intervall the acceleration is constant and equal to its value at the 'old' time t. This is only an approximation, and we might construct more sophisticated and more accurate, but also more complicated methods. But let us stick to this simple approach, wich is called 'explicit', since we need only information available at the 'old' time t in order to compute the velocity at the 'new' time $t + \Delta t$.

The next step is to use the velocity to compute the new position. In principle, we could use the 'old' velocities, the 'new' ones, or some combination. If we use the 'new' velocities $v_i t + \Delta t$ and write

$$x_i t + \Delta t = x_i(t) + v_i(t + \Delta t) \times \Delta t$$

so that we now also use information from the 'new' time – this approach is called implicit – it can be shown that the combined method conserves angular momentum for the case of accelerations due to a central force. Please verify analytically that this is true, i.e. that the angular momentum (remember: $\mathbf{L} = \mathbf{r} \times \mathbf{v}$?) after one time step is equal to the one before. The fact that this is mathematically valid has the nice consequence that our code can be exspected to conserve angular momentum with the accuracy of the representation of numbers by the machine!

At time zero, we have to specify the position and the velocity of each particle, and for each component. We shall describe that below.

As a simple test case, we can compute the movement of a particle in the gravitational field of another body:

$$a_i = -\frac{GM}{r^2} \frac{x_i}{r}$$

with the mass M of the body generating the gravitational field, and $r = \sqrt{x^2 + y^2 + z^2}$. To make things simple, we choose an adimensionalised system of units: setting $GM = 1$, the speed for a circular orbit at a radius $r = 1$ is $v = 1$, and the orbital period is 2π.

3 How to compute physical quantities

At the heart of SPH is an interpolation approach, whereby any physical quantity is expressed by the average of the property over the distribution of the particles:

$$< A(\mathbf{r}) >= \sum_{i=1}^{n} m_i \frac{A_i}{\rho_i} W(\mathbf{r} - \mathbf{r}_i)$$

The sum goes over all the n particles, which have masses m_i (but we shall assume that all have the same mass $m_i = 1$), the property A_i, and at their current position there is the density ρ_i. Depending on the distance between the point of interest $\mathbf{r}$ and the particle $\mathbf{r}_i$, we apply a smoothing by a function W. This function depends on a smoothing length h; it should be normalized to unity, and converge to a δ-function in the limit of $h \to 0$.

More concretely, we shall use the commonly used weighting function (or kernel) from page 554 of Monaghan (1992):

$$W(r) = \frac{\sigma}{h^k} \begin{cases} 1 - \frac{3}{2}q^2 + \frac{3}{4}q^4 & 0 \leq \mathrm{q} \leq 1 \\ \frac{1}{4}(2-q)^3 & 1 \leq \mathrm{q} \leq 2 \\ 0 & 2 \leq \mathrm{q} \end{cases}$$

where $q = r/h$. $\sigma = 2/3, 10/7\pi, 1/\pi$ is a normalization constant for k = 1, 2, and 3 dimensions, respectively.

For example, the mass density of the gas is simply:

$$< \rho(\mathbf{r}) >= \sum_{i=1}^{n} W(\mathbf{r} - \mathbf{r}_i)$$

To compute other quantities, including the acceleration, we shall need the gas density at the position of particle k

$$\rho_k = \sum_{i=1}^{n} W(\mathbf{r}_k - \mathbf{r}_i)$$

4 How to compute the accelerations

In the isothermal shock tube the gas is driven only by the gas pressure. As explained in Section 3.1 of Monaghan (1992), the j-component of the acceleration of the particle k can be formulated as (his Eq. 3.3)

$$a_j(k) = -\sum_{i=1}^{n} (\frac{P_k}{\rho_k^2} + \frac{P_i}{\rho_i^2}) \frac{\partial}{\partial x_j} W(\mathbf{r}_k - \mathbf{r}_i)$$

(See also e.g. Kitsionas & Whitworth 2007). The gradient of the smoothing kernel is

$$\frac{\partial}{\partial x_j} W(r) = \frac{dW(r)}{dr} \frac{\partial r}{\partial x_j} = \frac{dW(r)}{dr} \frac{x_j}{r}$$

Since W is given analytically, we can also compute the derivative analytical and write a routine for that function. Here we also see that the acceleration is due to a central force, hence our method for the propagation of the particles will conserve angular momentum.

A isothermal ideal gas of number density n and temperature T has the pressure

$$P = \frac{3}{2} n k_B T$$

and the mass density

$$\rho = mn$$

with the mean molecular mass m. The ratio

$$\frac{P}{\rho} = \frac{k_B T}{m} = c^2$$

is the square of the isothermal sound speed c. This gives for the expression in the accelerations

$$\frac{P_k}{\rho_k^2} = \frac{c^2}{\rho_i}$$

5 Initial conditions

In the shock tube we have the following initial configuration: one half of the volume is filled with gas of high density, the other contains gas at a lower density. To represent this, we may place a correspondingly higher number of particles in the section which contains the denser gas. In each section, the particles are evenly distributed, and their initial velocity shall be zero.

6 Suggestions on how to proceed

The best way is to start with a simple code to compute the orbit of a single particle in the gravitational field of another massive body. This program should nicely work. One expects that with a constant time step of 0.001 (how many points to make one orbit?) you should get good results. But do check the conservation of angular momentum of the code!

As a next step, one builds the SPH code in one dimension: implement the various functions and routines, make sure that the weighting function has the correct form and behaviour. It might be good to create the initial configuration of the particles and then compute the gas density from the formula, to see how the smoothing works. Then comes the part which computes the accelerations. This essential part contains loops over all the particles and it will also be called at every time step, so be careful not to waste time in recomputing the same things ... Also be careful with the indices of the various data vectors. It is very easy to overlook a very small error or simple typo, and spend weeks of trying to find out why the results are so weird or nonesense ...!

The main features of the shock tube should be obtained: that the dense gas flows into the empty section by forming a shock. You can compare with the results of a finite difference code, e.g. someone in the group chose that exercise, or check with my hydrodynamics applet. The quality of the results depend on the various computational parameters: number of particles, length of the time step, the value of the smoothing length h. Find out which parameter influences the results in what way, and how to choose good parameters.

SPH is a bit problematic with borders and boundaries. Here, we do not take any special measures to improve this ... so the particles will drift out of the computational region at each end. Don't worry about that.

If everything works nicely in one dimension, why not go into two or three dimensions? The formulae are easy to extend into more dimensions. Obviously, one will need more particles to cover the object, and one has to find good ways to plot the results. We can also add other forces, such as self-gravity, and try to find some interesting examples to study.

7 Literature

Monaghan J.J., 1992, Ann.Rev.Astron.Astrophys. 30, 543

Kitsionas S., Whitworth A.P., 2007, Mon.Not.R.Astron.Soc. 378, 507

The book "Numerical Recipes" is available in the MS2 library.

There is a test version of a Java applet - but still without any explanations – available at
`http://astro.u-strasbg.fr/~koppen/sph/`

An Applet on the computation of the gravitational two-body problem is available at
`http://astro.u-strasbg.fr/~koppen/body/TwoBody.html`

A Java applet for one dimensional hydrodynamics (finite difference method for the shock tube) is available at
`http://astro.u-strasbg.fr/~koppen/hydro/Hydro.html`