Chapter 2

Smoothed Particle Hydrodynamics

The smoothed particle hydrodynamics (SPH) technique for simulating hydrodynamic phenomena was first described by Lucy (1977). It was originally developed for use in non-axisymmetric astrophysical problems. Since then, much effort has gone into developing the method (Gingold & Monaghan 1977, 1982; Monaghan & Gingold 1983; Monaghan 1985) and it has become widely used in the study of many astrophysical problems including accretion discs, star formation, supernovae explosions, stellar collisions, galaxy formation and galactic interactions (see Benz 1988 for a review). Recently, its use has also spread to problems outside the field of astrophysics (see Monaghan 1992).

The SPH code used for the work in this thesis was originally developed by W. Benz (Benz 1990, Benz et al. 1990). Minor modifications were also made by I. A. Bonnell. On obtaining the basic code, extensive modifications were made to it. Most importantly, the method of integration of the SPH equations was changed from using a global time step for all particles, to each particle being evolved with its own individual time step (see Section 2.2.4). Without this modification, the computer time required for the simulations presented in this thesis would have been prohibitively long. This is because, typically, there is a large variation the dynamic times of the particles in each simulation. Without individual time steps, particles with large dynamic times are evolved as often as particles with the shortest dynamic times, and are therefore evolved much more accurately than is required, wasting computer time. With individual time steps, all particles are evolved to the same accuracy, avoiding unnecessary computation for particles that can be evolved with large time steps.

This chapter is divided into two sections. In the first section, the basic principles behind the SPH method are briefly described. This is followed, in the second section, by a description of the implementation of the SPH code used for the work in this thesis. Also critical to this thesis is the addition of 'sink' particles into SPH to model accreting protostars. This is discussed in Chapter 4 as it is not part of the standard SPH formalism and it comprises a significant proportion of the work done for this thesis.

2.1 General Principles of SPH

The general principles of SPH were first derived by Lucy (1977) as a Monte-Carlo approach to solving the time evolution of hydrodynamic equations. Later, a derivation of SPH in terms of interpolation theory was provided by Gingold and Monaghan (Gingold & Monaghan 1982; Monaghan 1985, 1988), which gives a better estimate of the error scaling of the method. As SPH is best viewed as an interpolation scheme, this is the derivation that is given here. Further details can be found in the above references, as well as many other papers (e.g. Hernquist & Katz 1989; Benz 1990; Monaghan 1992).

2.1.1 Derivation of SPH

Consider a function $f(\mathbf{r})$, which is defined in a space, $\xi(\mathbf{r})$. A smoothed approximation to this function may be obtained as

$$\langle f(\mathbf{r})\rangle = \int_\xi f(\mathbf{r}')\, W(\mathbf{r}-\mathbf{r}', h)\, \mathrm{d}\mathbf{r}', \tag{2.1}$$

where W satisfies

$$\int_\xi W(\mathbf{r}, h)\, \mathrm{d}\mathbf{r} = 1. \tag{2.2}$$

The function, W is known as the kernel. Its width is parameterised by the smoothing length, h, and hence if $h \to 0$, then

$$\langle f(\mathbf{r})\rangle \to f(\mathbf{r}). \tag{2.3}$$

Typically, the kernel is a strongly peaked function at $r = 0$, so that as $h \to 0$, the kernel becomes a delta function, $\delta(\mathbf{r})$.

In the case that the kernel is strongly peaked, the function $f(\mathbf{r})$ can be expanded in a Taylor series in Equation 2.1 about $\mathbf{r}$ as a power series in h. If the kernel is an even function, the leading term in the expansion vanishes and

$$\langle f(\mathbf{r})\rangle = f(\mathbf{r}) + O(h^2). \tag{2.4}$$

In the special case that the kernel varies only as a function of r (i.e. it is spherical), then

$$\langle f(\mathbf{r})\rangle = f(\mathbf{r}) + c\,\frac{h^2}{6}\nabla^2 f(\mathbf{r}) + O(h^3), \tag{2.5}$$

where c is independent of h. Therefore, $f(\mathbf{r})$ can always be replaced by its smoothed equivalent to within the accuracy of the smoothing process itself. This also means that

$$\left\langle \frac{A(\mathbf{r})}{B(\mathbf{r})} \right\rangle = \frac{\langle A(\mathbf{r})\rangle}{\langle B(\mathbf{r})\rangle} + O(h^2). \tag{2.6}$$

Now suppose the values of f($\mathbf{r}$) are only known at a discrete number of points N, instead of over the entire domain. If the points are distributed in the domain as

$$n(\mathbf{r}) = \sum_{\mathrm{j}=1}^{N} \delta(\mathbf{r}-\mathbf{r}_{\mathrm{j}}), \tag{2.7}$$

then Equation 2.1 can be evaluated by multiplying the integrand by $n(\mathbf{r}')/\langle n(\mathbf{r}')\rangle$, using Equation 2.6, and integrating to obtain

$$\langle f(\mathbf{r})\rangle \ = \sum_{\mathrm{j}=1}^{N} \frac{f(\mathbf{r}_\mathrm{j})}{\langle n(\mathbf{r}')\rangle} \ W(\mathbf{r}-\mathbf{r}_\mathrm{j}, h). \tag{2.8}$$

To describe a fluid, a fluid element of mass m_j, is associated with each point in Equation 2.7, and hence the number density at $\mathbf{r}_\mathrm{j}$ is given by

$$\langle n(\mathbf{r}_\mathrm{j})\rangle = \frac{\rho(\mathbf{r}_\mathrm{j})}{m_\mathrm{j}}. \tag{2.9}$$

When used in Equation 2.8 this results in

$$\langle f(\mathbf{r})\rangle \ = \sum_{j=1}^{N} \frac{m_\mathrm{j}}{\rho(\mathbf{r}_\mathrm{j})} \ f(\mathbf{r}_\mathrm{j}) \ W(\mathbf{r}-\mathbf{r}_\mathrm{j}, h). \tag{2.10}$$

Each point, with its associated mass, is known as an SPH particle. Equation 2.10 is the base equation for the SPH method, as it allows us to find smoothed approximations to the physical quantities of the fluid. For example, to find an approximation to the density at any point in the fluid $f(\mathbf{r}) \equiv \rho(\mathbf{r})$ is chosen in Equation 2.10 and thus

$$\langle \rho(\mathbf{r})\rangle \ = \sum_{\mathrm{j}=1}^{N} m_\mathrm{j} \ W(\mathbf{r}-\mathbf{r}_\mathrm{j}, h). \tag{2.11}$$

This is how Smoothed Particle Hydrodynamics gets its name, as the mass of each particle can be thought of as being smoothed over the kernel region to obtain the continuous density distribution of the fluid. Note that the continuity equation is automatically satisfied by Equation 2.11 so long as particle mass kept is constant and none are lost (Benz 1990).

The smoothing formalism also provides a way to find the gradient of fluid properties. By definition

$$\langle \nabla f(\mathbf{r})\rangle \ = \int_\xi \nabla f(\mathbf{r}') \ W(\mathbf{r}-\mathbf{r}', h) \ \mathrm{d}\mathbf{r}'. \tag{2.12}$$

Integrating by parts gives

$$\langle \nabla f(\mathbf{r})\rangle \ = \int_S f(\mathbf{r}') \ W(\mathbf{r}-\mathbf{r}', h) \ \mathbf{n} \ \mathrm{d}a \ \ + \ \ \int_\xi f(\mathbf{r}') \ \nabla W(\mathbf{r}-\mathbf{r}', h) \ \mathrm{d}\mathbf{r}'. \tag{2.13}$$

The surface term can be neglected by assuming the solution space either extends so far that the function $f(\mathbf{r})$ vanishes, or the kernel function itself vanishes. In principle this surface term can be used to provide boundaries in SPH by evaluating it for particles near the boundary for which neither $f(\mathbf{r})$, nor the kernel vanish. In practice, however, this can be rather complicated. Assuming the surface term can be neglected, the integral equation can be formulated as a summation for a discrete number of points, as before, to give

$$\langle \nabla f(\mathbf{r})\rangle \ = \sum_{\mathrm{j}=1}^{N} \frac{m_\mathrm{j}}{\rho(\mathbf{r}_\mathrm{j})} \ f(\mathbf{r}_\mathrm{j}) \ \nabla W(\mathbf{r}-\mathbf{r}_\mathrm{j}, h). \tag{2.14}$$

With this and Equation 2.10, any fluid property and its local gradient can be evaluated, assuming that ∇W does not vanish. The gradients are used in Sections 2.2.2 and 2.2.4 to enable pressure and viscous forces to be calculated at any point in the fluid, and hence on the particles. The particles are then moved according to these forces to obtain the new density distribution.

It is interesting to note that Gingold and Monaghan (1982) and Monaghan (1985) have shown that the density equation (Equation 2.11) for SPH is identical to other methods such as Particle In Cell (PIC) except that PIC, and others like it, use specific kernels with a definition involving the weighting function of the particle to the nearest mesh point and the spreading function of the particles. By contrast, the kernel in SPH is arbitrary. Also, derivatives in PIC are taken using a grid, whereas SPH calculates derivatives via the derivative of the interpolation kernel.

The main advantages of SPH over other numerical methods are: it is Lagrangian and it does not require the use of grids for calculating spatial derivatives. The particles themselves essentially form a grid of moving points from which the forces on the particles are interpolated via the kernel. Hence, in contrast to grid-based methods, there is no need to rezone grid cells as the fluid evolves. Furthermore, it is not necessary to know which regions require high resolution in order to follow them correctly before the simulation starts, since the particles follow the density distribution of the fluid, automatically giving high resolution to high-density regions.

2.1.2 Variable Smoothing Lengths

The ability of SPH to naturally give higher spatial resolution to the high-density regions of a simulation is limited if the smoothing lengths of particles are kept fixed in space and time, because the smoothing lengths of particles give the scale over which forces and physical fluid properties are smoothed out. In addition, if the kernel function used is only non-zero within a finite region, particles in low-density regions with a fixed smoothing length may become so separated that they have very few, or no neighbouring particles. In this case, the fluid simulation breaks down. Thus, if the ability of SPH to vary its resolution is to be used to its greatest advantage, it is important that the smoothing lengths are varied in space and time.

There are several points to consider, however. The first is the error introduced by including a spatially-varying smoothing length. The same process used to obtain the errors inherent in the SPH method in Equations 2.4 and 2.5 can be applied to the equation

$$\langle f(\mathbf{r})\rangle = \int_{\xi} f(\mathbf{r}')\; W(\mathbf{r}-\mathbf{r}', h(\mathbf{r}'))\; \mathrm{d}\mathbf{r}', \tag{2.15}$$

where h now varies spatially. It is found by several authors (Monaghan 1985; Hernquist & Katz 1989) that the errors introduced by a spatially-varying smoothing length are again second order in h, and hence are still of the same order as the errors inherent in the SPH method.

Introducing a spatially varying smoothing length does, however, formally introduce a new term into Equation 2.14 for the gradient of fluid properties as

$$\nabla_{\mathbf{r}} W(\mathbf{u}, h(\mathbf{r}) = -\frac{1}{u}\frac{\partial W}{\partial u}\mathbf{u} + \frac{\partial W}{\partial h}\nabla_{\mathbf{r}} h(\mathbf{r}), \quad (2.16)$$

where $\mathbf{u} = \mathbf{r} - \mathbf{r}'$. The new, second term on the right-hand side of this equation is proportional to the gradient of the smoothing length, and is generally ignored. Evrard (1988) found that this term only becomes important if the smoothing length varies on a scale less than the smoothing length itself. Since all quantities in SPH are smoothed over the smoothing length scale, it can generally be neglected. Nelson and Papaloizou (1994) showed that including this additional term in the derivation and implementation of the SPH fluid equations does result in better energy conservation, especially when the smoothing lengths vary over a short distance scale. In a typical calculation though, this effect is small.

A more important consideration when including a variable smoothing length is the effect on forces between particles. It is important when calculating the gradients of physical quantities, and hence the forces on particles (see Sections 2.2.2 and 2.2.3), that the same smoothing length be used in the calculation of the force on particle i from particle j, as in the calculation of the opposite force. If this is not done, momentum is not conserved. Several methods have been proposed to achieve this symmetrisation (Gingold & Monaghan 1982; Evrard 1988; Hernquist & Katz 1989). The method used here (Evrard 1988; Benz 1990) is simply to replace h in all previous equations defined which are on a continuous space by

$$h(\mathbf{r}, \mathbf{r}') = \frac{h(\mathbf{r}) + h(\mathbf{r}')}{2}. \quad (2.17)$$

For discrete particles this is equivalent to replacing h by

$$h_{\mathrm{ij}} = \frac{h_{\mathrm{i}} + h_{\mathrm{j}}}{2}. \quad (2.18)$$

Thus, the use of variable smoothing lengths, allows the advantages of the Lagrangian nature of SPH and the lack of a grid to be used to resolve high and low-density regions to their fullest extent. The method of choosing the smoothing lengths during an SPH simulation is discussed in Section 2.2.4.

2.2 SPH Implementation

Now that the formalism behind the SPH method has been discussed, the derivation of all the equations necessary for the simulation of a self-gravitating gas is described. This is presented in four main sections. First, the kernel must be chosen. Second, the fluid equations must be derived. Third, for a self-gravitating gas, as is required in the simulation of many astrophysical problems, the implementation of a tree to calculate gravitational forces between particles is described. The use of the tree also allows the neighbours of each particle to be found easily. Finally, the integration of these equations in time using individual particle time steps, and the method of updating the smoothing lengths of particles is presented.

In the equations that follow, all quantities are smoothed SPH quantities and the angled brackets previously used to denote smoothed quantities are dropped. The replacing of any unsmoothed quantities, in a formal derivation of the following equations, by their smoothed SPH estimates is consistent with the overall accuracy of the SPH method as discussed in Section 2.1.1.

2.2.1 The Interpolating Kernel

Many different kernels have been proposed for use in SPH. The original kernels (Gingold & Monaghan 1977) were spherical and of the Gaussian form

$$W(r,h) = \frac{1}{\pi^{3/2}\ h^3} \exp\left(-v^2\right), \tag{2.19}$$

for three-dimensional calculations, where $r = |\mathbf{r}-\mathbf{r}'|$ and $v = r/h$. The main disadvantage of this kernel is that all particles in the simulation contribute to the summations, despite the fact that density, pressure and fluid forces are local quantities. Summing over all particles is an unnecessary waste of computer time.

The most common form of kernel is currently the spline kernel of Monaghan and Lattanzio (1985), and this is the one used here. It has the advantage that it is defined on compact support, and therefore only a small number of neighbouring particles must be summed over to calculate SPH quantities. The kernel is given by

$$W(r,h) = \frac{\sigma}{h^\nu} \begin{cases} 1 - \frac{3}{2}v^2 + \frac{3}{4}v^3 & \text{if } 0 \le v < 1, \\ \frac{1}{4}(2-v)^3 & \text{if } 1 \le v < 2, \\ 0 & \text{otherwise,} \end{cases} \tag{2.20}$$

where ν is the number of dimensions and σ is the normalisation with values of $2/3$, $10/(7\pi)$ and $1/\pi$ in one, two and three dimensions respectively. This thesis makes use of calculations performed in one, two and three dimensions.

2.2.2 Fluid Equations

Physical Fluid Properties

As was seen in Section 2.1.1, the density at each particle position is estimated by a weighted sum of the mass of the all the particles in the solution space. With the finite spline kernel of Equation 2.20 which is non-zero only over a radius of $2h$, however, the summation is now reduced to simply being over the particle itself and each of its neighbours as

$$\rho_{\rm i} = m_{\rm i} W(0, h_{\rm i}) + \sum_{\rm j=1}^{N_{\rm n}} m_{\rm j} W(r_{\rm ij}, h_{\rm ij}), \tag{2.21}$$

where $N_{\rm n}$ is the number of neighbours of particle i, and $r_{\rm ij} = |\mathbf{r}_{\rm i} - \mathbf{r}_{\rm j}|$.

In addition to the density of the fluid, the pressure P, must also be defined via an equation of state. Three different types of equation of state are used in this thesis, depending on the problem. The first is a purely isothermal equation of state

$$P = c_{\mathrm{s}}^2 \rho, \tag{2.22}$$

where c_{s} is the isothermal sound speed of the gas. The second is an ideal gas equation of state

$$P = (\gamma - 1) u \rho, \tag{2.23}$$

where u is the internal energy, and γ is the polytropic index of the gas. Finally, the polytropic equation of state

$$P = K \rho^{\gamma}, \tag{2.24}$$

can be used, where K depends on the entropy of the gas. When the energy conservation equations, below, are used with the ideal gas or polytropic equations of state, total energy is conserved.

Momentum Equation

The momentum conservation equation for an inviscid fluid in the absence of gravity is given, in Eulerian form, by

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} = -\frac{\nabla P}{\rho}. \tag{2.25}$$

By multiplying both sides of this equation by the kernel and integrating over the solution space, an integral equation can be found. The left-hand side of this integral equation simply gives the left-hand side of the above equation, except now all the velocities are the smoothed SPH velocities. The right-hand side of the equation can be reformulated by integrating it by parts. The surface term vanishes as either the pressure or the kernel are assumed to vanish themselves. If this is not so, boundary conditions must be added to the SPH simulation. The remaining integral can be reformulated by comparing it to a direct expansion of the right-hand side of the original integral equation obtained from Equation 2.25 to eventually obtain

$$\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = -\nabla_{\mathrm{r}} \left(\frac{P}{\rho} \right) - \left(\frac{P}{\rho^2} \right) \nabla_{\mathrm{r}} \rho. \tag{2.26}$$

For a full derivation of the above, the reader is referred to Benz (1990).

Taking Equation 2.26, and using the same method as in Section 2.1.1 to transform it to a summation over a discrete number of points, the final SPH momentum equation is

$$\frac{\mathrm{d}\mathbf{v}_{\mathrm{i}}}{\mathrm{d}t} = -\sum_{\mathrm{j}=1}^{N_{\mathrm{n}}} m_{\mathrm{j}} \left(\frac{P_{\mathrm{i}}}{\rho_{\mathrm{i}}^2} + \frac{P_{\mathrm{j}}}{\rho_{\mathrm{j}}^2} \right) \nabla_{\mathrm{i}} W(r_{\mathrm{ij}}, h_{\mathrm{ij}}) \tag{2.27}$$

where ∇_{i} means take the gradient with respect to the coordinates of particle i. This equation conserves total momentum. Note that angular momentum is conserved locally as well, since all forces are directed along a line joining the centres of the particles.

Artificial Viscosity

The equations thus far have been derived for an inviscid fluid. There is no mechanism which allows the conversion of kinetic energy into heat due to viscosity, and for this reason the simulation of shocks is not possible with the equations as they stand. In a real fluid, the inherent viscosity allows for this dissipation of heat, and the energy in shocks is always able to be converted heat no matter how small this viscosity is, as the gradient of the variables across the shock adjusts to allow for it. In most astrophysical problems the molecular viscosity is very small, and therefore dissipation should ideally only occur in shocks.

In SPH, this is achieved by introducing standard artificial viscosity terms. Two different types of viscosity are used which both translate into artificial viscous pressures between particles. They take the forms

$$P_\alpha = \Pi_\alpha \rho^2 = -\alpha \rho l c_{\mathrm{s}} \nabla \cdot \mathbf{v}, \tag{2.28}$$

and

$$P_\beta = \Pi_\beta \rho^2 = \beta \rho l^2 \left(\nabla \cdot \mathbf{v}\right)^2, \tag{2.29}$$

where α and β are free parameters controlling the strength of the viscosity and l is a typical length scale over which the shock is spread. Equation 2.28 is a bulk viscosity and its primarily purpose is to eliminate subsonic velocity oscillations, especially after a shock. Equation 2.29 is a second-order, von Neumann-Richtmyer-type viscosity which is important primarily in the supersonic regime (since it depends on the square of the divergence) to eliminate particle interpenetration in high-Mach-number shocks. Typical values for the free parameters are $\alpha \approx 1$ and $\beta \approx 2$, though in simple accretion disc simulations, where high Mach number shocks are not expected, it is common to include only the α-viscosity with a value ranging between 0.1 and 1.0. See Chapter 3 for testing of the behaviour shocks under different strengths of the viscosity.

The forms of artificial viscosity used in the SPH code are included in the momentum Equation 2.27 as

$$\frac{\mathrm{d}\mathbf{v}_{\mathrm{i}}}{\mathrm{d}t} = -\sum_{\mathrm{j}} m_{\mathrm{j}} \left(\frac{P_{\mathrm{i}}}{\rho_{\mathrm{i}}^2} + \frac{P_{\mathrm{j}}}{\rho_{\mathrm{j}}^2} + \Pi_{\mathrm{ij}} \right) \nabla_{\mathrm{i}} W(r_{\mathrm{ij}}, h_{\mathrm{ij}}). \tag{2.30}$$

We use the formulations for Equations 2.28 and 2.29 first used by Monaghan and Gingold (1983) (see also Benz 1990; Monaghan 1992). This is the most common form of artificial viscosity used in SPH and has been shown to reproduce shocks well (Monaghan & Gingold 1983). Chapter 3 of this thesis is devoted to testing the SPH code, especially how well shocks are modelled by this artificial viscosity when the parameters α and β are varied. The artificial viscosity is given by

$$\Pi_{\mathrm{ij}} = \begin{cases} (-\alpha c_{\mathrm{s}} \mu_{\mathrm{ij}} + \beta \mu_{\mathrm{ij}}^2)/\rho_{\mathrm{ij}} & \mathbf{v}_{\mathrm{ij}} \cdot \mathbf{r}_{\mathrm{ij}} \le 0, \\ 0 & \mathbf{v}_{\mathrm{ij}} \cdot \mathbf{r}_{\mathrm{ij}} > 0, \end{cases} \tag{2.31}$$

where $\rho_{\mathrm{ij}} = (\rho_{\mathrm{i}} + \rho_{\mathrm{j}})/2$ and

$$\mu_{\mathrm{ij}} = \frac{h\ \mathbf{v}_{\mathrm{ij}} \cdot \mathbf{r}_{\mathrm{ij}}}{\mathbf{r}_{\mathrm{ij}}^2 + \eta^2}, \tag{2.32}$$

where $\mathbf{v}_{\rm ij} = \mathbf{v}_{\rm i} - \mathbf{v}_{\rm j}$. The quantity $\eta^2 = 0.01h^2$, is included to avoid divergence for small separations between neighbouring particles.

Energy Conservation and Entropy Changes

For energy conservation, without artificial viscosity, the effects of compression and rarefaction on the internal energy of the gas must be taken into account. This is not required if an isothermal equation of state is used, as the internal energy of the gas is constant due to the loss of heat from the system. If a polytropic equation of state (Equation 2.24) is being used, the specific internal energy of the gas is given explicitly by

$$u = \frac{K}{\gamma - 1}\,\rho^{\gamma-1}. \tag{2.33}$$

However, for the ideal gas equation of state (Equation 2.23) these effects on the internal energy of the gas must be calculated. The first law of thermodynamics describes adiabatic energy conservation for an inviscid, ideal gas as

$$\frac{\partial u}{\partial t} + (\mathbf{v}\cdot\nabla)u = -\frac{P}{\rho}\,\nabla\cdot\mathbf{v}. \tag{2.34}$$

Following the same method as was used to obtain the SPH momentum equation, the SPH energy equation is (Benz 1990)

$$\frac{\mathrm{d}u_{\rm i}}{\mathrm{d}t} = \frac{P_{\rm i}}{\rho_{\rm i}^2}\sum_{\rm j=1}^{N_{\rm n}} m_{\rm j}\mathbf{v}_{\rm ij}\cdot\nabla_{\rm i}W(r_{\rm ij},h_{\rm ij}) \tag{2.35}$$

where the right-hand side is simply $P_{\rm i}/\rho_{\rm i}$ times the divergence of the fluid $\nabla\cdot\mathbf{v}_{\rm i}$. This is the SPH representation of the $P\mathrm{d}V/\mathrm{d}t$ term. Total energy can be shown to be conserved by this equation in the absence of artificial viscosity (Benz 1990).

The inclusion of artificial viscosity gives heating of the gas due to shocks. For the polytropic equation of state (Equation 2.24), the internal energy of the gas changes because the entropy (and hence K) of the gas increases. The change in K due to the artificial viscosity is given by

$$\frac{\mathrm{d}K_{\rm i}}{\mathrm{d}t} = \frac{\gamma-1}{2\rho_{\rm i}^{\gamma-1}}\sum_{\rm j=1}^{N_{\rm n}} m_{\rm j}\Pi_{\rm ij}\mathbf{v}_{\rm ij}\cdot\nabla_{\rm i}W(r_{\rm ij},h_{\rm ij}). \tag{2.36}$$

For the ideal gas equation of state (Equation 2.23), total energy conservation requires a further term to be added to Equation 2.35. The final equation combining Equation 2.35 and this new term is

$$\frac{\mathrm{d}u_{\rm i}}{\mathrm{d}t} = \frac{P_{\rm i}}{\rho_{\rm i}^2}\sum_{\rm j=1}^{N_{\rm n}} m_{\rm j}\mathbf{v}_{\rm ij}\cdot\nabla_{\rm i}W(r_{\rm ij},h_{\rm ij}) \quad + \quad \frac{1}{2}\sum_{\rm j=1}^{N_{\rm n}} m_{\rm j}\Pi_{\rm ij}\mathbf{v}_{\rm ij}\cdot\nabla_{\rm i}W(r_{\rm ij},h_{\rm ij}). \tag{2.37}$$

This equation ensures total conservation of energy for the gas.

2.2.3 Self-Gravity and the Neighbours Lists

For the simulation of self-gravitating systems, a method of calculating the gravitational forces between all pairs of particles is required. The method described by Benz (1990) is used to implement self-gravity. The gravitational forces between SPH particles are found by interpreting the kernel as the density distribution of the particles, and thus the particles are spherically symmetric.

Starting with the Poisson equation

$$\Delta\phi = \frac{1}{r^2}\frac{\mathrm{d}}{\mathrm{d}r}\left(r^2\frac{\mathrm{d}\phi}{\mathrm{d}r}\right) = 4\pi G\rho, \tag{2.38}$$

the usual solution for the forces exerted on particle i by particle j can be obtained as

$$-\left(\nabla\phi_{\mathrm{i}}\right)_{\mathrm{j}} = -G\,\frac{M(r_{\mathrm{ij}})}{r_{\mathrm{ij}}^2}\,\frac{\mathbf{r}_{\mathrm{ij}}}{r_{\mathrm{ij}}}, \tag{2.39}$$

where $\mathbf{r}_{\mathrm{ij}} = \mathbf{r}_{\mathrm{i}} - \mathbf{r}_{\mathrm{j}}$, and

$$M(r_{\mathrm{ij}}) = 4\pi\int_0^{r_{\mathrm{ij}}} u^2\rho(u)\mathrm{d}u = 4\pi\int_0^{r_{\mathrm{ij}}} u^2 W(u, h_{\mathrm{ij}})\mathrm{d}u. \tag{2.40}$$

This definition of the forces between particles introduces a new constraint on the kernel that

$$\int_0^{\infty} r^2 W(r,h)\mathrm{d}r < \infty, \tag{2.41}$$

to avoid an infinite mass. The total gravitational force felt on particle i is then given by a summation over all other particles as

$$-\nabla\phi_{\mathrm{i}} = -G\sum_{\mathrm{j}=1}^{N}\frac{M(r_{\mathrm{ij}})}{r_{\mathrm{ij}}^2}\,\frac{\mathbf{r}_{\mathrm{ij}}}{r_{\mathrm{ij}}}. \tag{2.42}$$

This term is added to the momentum equation (Equation 2.30) to model a self-gravitating gas.

A simple implementation of Equation 2.42 is to sum over all particles for each particle, calculating all the forces. However, the number of force computations then scales as $N^2/2$ and becomes impractical for any realistic number of particles. Instead, a hierarchical-tree method can be used. The force computations required to calculate gravity using a tree scale as $N\log N$. In addition, the lists of neighbours for each particle, required for the local fluid calculations, can be found simultaneously.

The details of the implementation of the hierarchical tree method used by this SPH code can be found in the appendix of Benz et al. (1990). The tree code was originally written by Press (1986). Only an overview of the method is given here.

The tree is, in fact, an inverted tree. The lowest level of the tree is formed from all the particles in the calculation. The next level in the tree consists of nodes, which are formed from a particle and its mutual nearest neighbour. This process continues all the way up the tree, until only one node remains. The total number of nodes is $2N - 1$. When a node is formed from two particles, two nodes, or a particle and a node, its location is the centre of mass of the pair, and its quadrupole moment is computed. It is also given a radius, defined by ensuring all of its constituent nodes are contained within this radius of the centre of mass of the node. A conservative, but simple way of defining this radius is

$$R_{\rm n} = {\rm MAX}\left(\frac{M_{\rm i}}{M_{\rm n}} r_{\rm ij} + R_{\rm j},\ \frac{M_{\rm j}}{M_{\rm n}} r_{\rm ij} + R_{\rm i}\right), \tag{2.43}$$

where $M_{\rm i}$ and $M_{\rm j}$ are the masses of the two sub-nodes, $M_{\rm n} = M_{\rm i} + M_{\rm j}$, and $R_{\rm i}$ and $R_{\rm j}$ are the radii of the two sub-nodes. Each node is also assigned a smoothing length, defined as

$$h_{\rm n} = {\rm MAX}\left(h_{\rm i},\ h_{\rm j}\right). \tag{2.44}$$

When computing the gravitational forces on a particle, a distant node does not need to be expanded to consider the forces from each of its components. Rather, the force from this node is computed only from the position of the node and its quadrupole correction. A node is only expanded to consider the individual forces from its component parts if

$$\frac{R_{\rm n}}{r_{\rm in}} < \theta, \tag{2.45}$$

where i is the particle, n is a node, and θ is some opening angle. In the limit that $\theta \rightarrow 0$, all nodes are opened, and the tree reverts to the simple $N^2/2$ summation for the gravity forces on particles. However, as θ is increased, forces from nodes progressively closer to the particle are calculated from the centre of mass and quadrupole correction. The value of θ controls the accuracy of the force calculation. The code used here has $\theta = 0.5$. Several tests were performed by collapsing clouds (as in Section 3.4) with different values of θ and comparing the results with a $\theta = 0$ calculation. The value of $\theta = 0.5$ gives negligible errors and a much decreased calculation time over the direct summation. Only with $\theta > 0.7$ do the calculations show large differences from the exact calculation.

As well as expanding nodes to calculate gravitational forces directly, nodes must also be expanded to allow their constituent particles to be added to a neighbours list. Constituent particles of a node may be neighbours of particle i if

$$R_{\rm n} + h_{\rm i} + h_{\rm n} < r_{\rm in}. \tag{2.46}$$

All nodes which satisfy this equation must be opened and searched for neighbours of particle i. A list of neighbours of the particle is stored to enable local SPH fluid quantities to be computed. Also, all gravitational forces between neighbours are calculated directly.

The use of the kernel to define the density distribution of the particles means that gravitational forces between particles which are neighbours are smoothed (see Equations 2.39 and 2.40) and the problem of a diverging gravitational force between particles with a small separations is avoided. However, this leads to non-conservation of energy when the code is used with a variable smoothing length, as the potential energy of the a particle is being changed when the smoothing length is changed. This enters the calculation of the potential energy of the system as a change in the zero-point of the gravitational energy of the system. As given by Benz (1990), this change is given by

$$\mathrm{d}H = -G \sum_{\mathrm{i<j}} m_{\mathrm{i}} m_{\mathrm{j}} \frac{\partial \phi}{\partial h_{\mathrm{ij}}} \mathrm{d}h_{\mathrm{ij}} \tag{2.47}$$

and can be added to the calculation of the potential energy of the system during the calculation. The need for this term can be avoided by having a fixed gravitational smoothing length as used by Hernquist and Katz (1989) and Evrard (1988). However, this removes part of the advantage of introducing variable smoothing lengths in the first place. Linking the gravitational smoothing to the smoothing of the mass of the particles is clearly more consistent, and with a fixed gravitational smoothing length it is possible for one type of smoothing length to be very much larger than the other - the desirability of which is unclear.

2.2.4 Time Integration of the SPH Equations

When the SPH code was originally obtained, time integration of the SPH equations was performed using the method of Benz (Benz 1984) based on a second-order, Runge-Kutta-Fehlberg integrator with an adaptive time step. All particles were constrained to move with the same time step. Using this method, the time step required to integrate a particle with a specified accuracy by the Runge-Kutta-Fehlberg integrator is set by constraining the changes, between two consecutive time steps, of several physical quantities of the particle to be less than some certain tolerance. The quantities constrained are the particle's velocities, accelerations, internal energy, and smoothing length. The next time step is then given by

$$\delta t_{\mathrm{RKF}} = \sqrt{\frac{512\ \delta t_{\mathrm{old}}\ \lambda}{|Q_{\mathrm{new}} - Q_{\mathrm{old}}|}}, \tag{2.48}$$

where Q is each physical quantity in turn, and λ is the given tolerance. The minimum of the time steps given by using each of a particle's physical quantities in turn, gives the time step required to integrate the particle to the required accuracy. Then the global time step for the code is taken to be the minimum of this value for all particles.

This method of integration works well if the time steps required by the particles do not differ too much over the entire simulation. However, for a calculation in which very high-density regions are formed, for example, the collapse of a molecular cloud to form a protostar, the particles in the high-density regions have very small separations, and high velocities and densities, and thus require time steps many orders of magnitude less than particles on the outside of the cloud which still have to collapse. In this case, much computer time is wasted to integrate particles in the outer parts of the cloud with far more accuracy than is required.

Individual Time Steps

To avoid this excessive use of computer time, and speed up such simulations, it is necessary to introduce individual time steps. This was the first major modification made to the original SPH code. Each particle is integrated with its own time step, and hence there is no waste of computational resources by integrating particles with more accuracy than is required.

In practice, individual time steps are calculated for each particle, and then these are binned into multiples of two of the minimum possible time step for a particle (see Figure 2.1). This is desirable for several reasons. First, binning the time steps in to multiples of two means that, typically, many particles are evolved at once, instead of each particle being evolved at a different time. As well as all particles using the same time step being evolved together, particles on other time steps may also be evolved at the same time, if they have smaller time steps or if the current time is a multiple of their time step. This is demonstrated in Figure 2.1. Computations can then be minimised, as equal and opposite forces between particles that are being evaluated together only have to be calculated once.

Second, this procedure allows particles to be given an integer representation of their time step. This avoids the problems of rounding error due to repeated additions of small times on to a large time. If a particle has a small time step, then after many thousands, perhaps millions of additions of the value of that time step on to the current time, rounding error can be significant. In the worst case, a particle could have a time step of around 10^{-6} of the greatest time step allowed. In this case, the rounding error in the addition of the time step on to the current time can exceed the time step itself, and the time to which that particle has been integrated never increases! The use of integer representation of time steps allows a time step differentiation of 2^{20} for standard FORTRAN integers, and there is no rounding error, as the current time is calculated by adding integers. The current time, kept as an integer number of times the smallest possible time step, is zeroed when it reaches the maximum integer allowed by the integer representation. The time for this to occur is known as the synchronisation time, and is also equal to the maximum time step allowed for any particle. At the synchronisation time, all particles in the simulation are updated, and it is a natural time to save the simulation's state to disk. When the current integer time is zeroed, the time since the beginning of the simulation is increased by one synchronisation time (a floating point number).

When changing the SPH code to individual time steps, the Runge-Kutta-Fehlberg condition for setting the time steps of the particles is no longer enough to ensure correct integration of the system. As well as the time step $\delta t_{\rm RKF}$ that is calculated for each particle, two more time steps must be calculated for each particle.

The first is the Courant condition. The time step given by the Courant condition is found by considering a characteristic length scale divided by a characteristic velocity. This takes the form

$$\delta t_{\rm cr} = \frac{0.3\ h}{c_{\rm s} + h|\nabla \cdot \mathbf{v}| + 1.2(\alpha c_{\rm s} + \beta h|\nabla \cdot \mathbf{v}|)}, \tag{2.49}$$

where α and β are the two artificial viscosity parameters, and the last term in the denominator is only included if $\nabla \cdot \mathbf{v} < 0$. The divergence $\nabla \cdot \mathbf{v}$ can be determined from

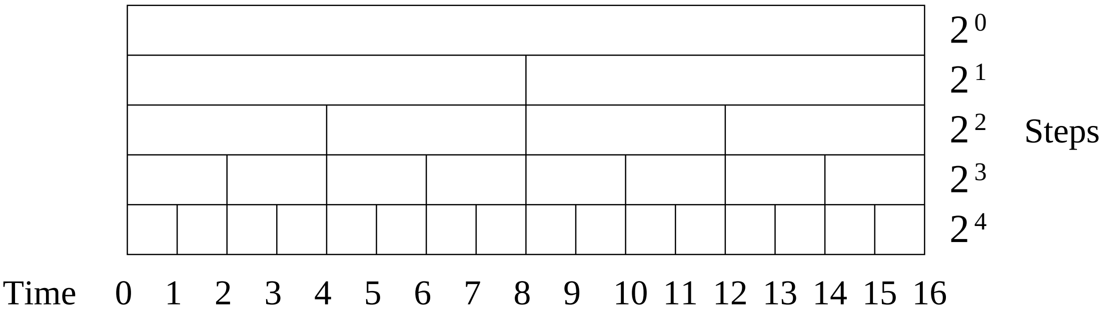

Figure 2.1: An example of the time steps of particles in an individual-time-steps code. Here, there are five levels of time step allowed. The largest time step is the top level, and also corresponds to the synchronisation time at which all particles are evolved together. The smallest time step is a factor of 2^4 smaller than the largest time step. Clearly, any particle can reduce its time step at any time, as it is always being evolved at the same time as particles with shorter time steps. However, for a particle to increase its time step, it must be being evaluated at the same time as a particle which has a larger time step, otherwise it is not synchronised at the next synchronisation time.

Equation 2.35.

Using only the Runge-Kutta-Fehlberg and Courant conditions, the individual-time-steps code was tested against the original, single-time-step code to check the two codes gave the same results. The test case used was the collapse of an initially spherical, isothermal, $1/r$-centrally-condensed gas cloud (see Section 3.4). It was found that the individual-time-steps code did not react as fast to rapid changes in the forces on particles during the bounce of the calculation as the original code. To correctly follow the calculation, another condition for setting the particle time steps is required.

Monaghan (1992) gives a force condition in addition to the Courant condition to set the time steps in his SPH code. The time step of a particle using this condition is

$$\delta t_{\rm for} = 0.3\sqrt{\frac{h}{|\mathbf{F}|}}, \tag{2.50}$$

where $\mathbf{F}$ is the net acceleration on the particle. When this is added to the above two conditions to set the time steps, the discrepancy between simulations performed with the single-time-step and individual-time-steps codes is eliminated.

Thus, the time step for each particle is calculated as the minimum of the above three time steps, and then rounded down to equal a multiple of 2 times the minimum possible time step the SPH code can take. Care must be taken when increasing the time step of a particle. A particle can at any point in the simulation move to a smaller time step (Figure 2.1), however, it can only move to a larger time step if

$$({\rm d}t_{\rm syn} - t) \;\; {\rm MOD} \;\; {\rm d}t = 0 \tag{2.51}$$

where t is the current time, $\mathrm{d}t$ is the new (larger) time step, and $\mathrm{d}t_{\rm syn}$ is the synchronisation time (all quantities are the integer representations of the times). If this constraint is not used, a particle may not be synchronised at the synchronisation time (Figure 2.1). Also, no particle is allowed to increase its time step by more than a factor of two between two consecutive steps, but a particle can reduce its time step by as many factors of two as is required.

Evolution of Particle Smoothing Lengths

The smoothing lengths of particles in an SPH code with variable particle smoothing lengths should be evolved with time to keep the number of neighbours over which the SPH equations are calculated approximately constant. There is a trade off to be made between having too few neighbours, and too many. If particles have too few neighbours the statistical errors in the calculations of smoothed quantities are high. In the worse case, when particles have no neighbours, there are no pressure and viscous forces between the particles, and the whole fluid simulation breaks down. On the other hand, if the particles have a large number of neighbours, then computer time is being wasted. The summations over the lists of neighbours to calculate SPH quantities and forces take longer than necessary. Also, because SPH only resolves structure on the scale of a smoothing length, a simulation with many neighbours for each particle requires more particles in total to give the same resolution as a simulation that requires fewer neighbours for each particle.

Generally, a minimum of ≈ 50 neighbours in three dimensions is required to accurately calculate SPH quantities. This number can be reduced in fewer dimensions. This SPH code tries to maintain the number of neighbours for each particle to between 5 - 10, 20 - 35, and 30 - 70 for one, two and three dimensions respectively.

In a dynamic simulation the smoothing lengths must constantly change to keep the number of neighbours roughly constant for each particle. The scaling law to find the new smoothing length h, required to keep the same number of neighbours as a particle had originally is

$$h = h_0 \left(\frac{\rho_0}{\rho}\right)^{(1/\nu)}, \tag{2.52}$$

where ρ_0 and h_0 are the original densities and smoothing lengths of the particle respectively, and ν is the number of dimensions. The problem with this scaling law is that the new smoothing length is needed to determine ρ (Equation 2.21).

Following Benz (1990), however, the derivative of Equation 2.52 can be taken to obtain

$$\frac{\mathrm{d}h}{\mathrm{d}t} = -\frac{1}{3}\frac{h}{\rho}\frac{\mathrm{d}\rho}{\mathrm{d}t}. \tag{2.53}$$

Then, using the continuity equation the time derivative of the density can be replaced to give

$$\frac{\mathrm{d}h_{\mathrm{i}}}{\mathrm{d}t} = \frac{1}{3}h\nabla\cdot\mathbf{v}_{\mathrm{i}}. \tag{2.54}$$

This equation for the change in the smoothing length is then integrated in time along with all the other quantities. Generally, it maintains the number of neighbours for each particle very well. In the event that this derivative form for changing the smoothing length does begin to fail, and the number of neighbours starts to approach the above upper and lower bounds, the derivative of the smoothing length is modified from Equation 2.54 to

$$\frac{\mathrm{d}h_{\mathrm{i}}}{\mathrm{d}t} = \frac{\exp(x)(h\nabla \cdot \mathbf{v}_{\mathrm{i}})/3 \ \pm \ \exp(-x)2h_{\mathrm{i}}}{\exp(x) + exp(-x)}, \tag{2.55}$$

where + is chosen if the the smoothing length is to increase, and − is chosen if the smoothing length is to decrease. The exponent is

$$x = \begin{cases} (N_{\mathrm{upper}} - N_{\mathrm{n}})/(0.3\ n_{\mathrm{range}}) & \text{if } (N_{\mathrm{upper}} - N_{\mathrm{n}}) < n_{\mathrm{range}}, \\ (N_{\mathrm{n}} - N_{\mathrm{lower}})/(0.3\ n_{\mathrm{range}}) & \text{if } (N_{\mathrm{n}} - N_{\mathrm{lower}}) < n_{\mathrm{range}}, \\ \infty & \text{otherwise,} \end{cases} \tag{2.56}$$

where N_{upper} and N_{lower} are the upper and lower bounds on the number of neighbours a particle should have (given above), and n_{range} is 1, 5, or 12 in one, two or three dimensions, respectively. This ensures the number of neighbours of a particle is quickly brought back into the correct range, if Equation 2.54 begins to fail.