3

Smoothed Particle Hydrodynamics

Or: How I Learned to Stop Worrying and Love the Lagrangian*

* With apologies to Drs. Strangelove and Price

As far as the laws of mathematics refer to reality, they are not certain; and as far as they are certain, they do not refer to reality.

Albert Einstein

3.1 Introduction

The study of astrophysical phenomena presents a multitude of obstacles to the potential student. In addition to the usual obstacles of understanding the physical properties of the system in question, the sheer scale of astrophysical events renders laboratory experiments impossible in the vast majority of cases. To this end, it has been necessary to assemble a new, numerical laboratory in the form of computational simulations, and conduct experiments and analyses via this medium. The growth of computing power over the past 80 years, from the Colossus of Bletchley Park's Enigma code-cracking efforts in the 1940s, through ENIAC and Los Alamos National Laboratory's modelling of thermonuclear detonations in the 1950s, up to the supercomputers of today, has in turn allowed the computational domain to become a mainstay of astrophysical experimentation.

Two principal approaches to computational simulations have evolved to enable these numerical simulations. Eulerian methods use geometric grids, either fixed or adaptive (the so-called AMR or Adaptive Mesh Refinement codes), with the fluid parameters evaluated over the grid cells. Such codes formed the basis of the revolution in Computational Fluid Dynamics (CFD) that started in the late 1960s and early 70s, and as such they remain the most widely used approach. Applications of such codes cover a huge range, from industrial aerodynamics in the automotive and aerospace sectors, to stress calculations and solid mechanics for civil engineering and architecture, to chemical reaction modelling and protein folding in biomolecular models.

Lagrangian methods on the other hand dispense with fixed points in space and instead evolve the fluid equations in a co-moving frame. A common approach is to use discrete particles that are carried with the flow – hydrodynamic (and other) properties are then evaluated at the particle positions, and are calculated from a weighted average of the values on other local particles. In this manner each particle is essentially "smoothed" over a finite volume of fixed mass, and in this way these so-called Smoothed Particle Hydrodynamics or SPH codes are naturally adaptive with density. Although SPH was originally developed by the astrophysics community, it too has found uses and applications in a much wider range of fields. In engineering it has been applied to dam breaks and atomised oil lubrication flows, while a number of physics engines in computer games use SPH as a basis. The community has grown to the point where there is now a Europe-wide network of users called SPHERIC -

the SPH European Research Interest Community[1]. This aims to share advances in code development across the user community, and to prevent the re-invention of the wheel when it comes the solution of known problems.

Each of these approaches has advantages and disadvantages with respect to the other. Generally speaking, AMR codes have a higher resolution for a given number of grid cells than an SPH code with an equal number of particles. Furthermore, they can be made to adapt to any flow parameter (although this is not always trivial!), while SPH adapts primarily with density only. On the other hand, SPH naturally handles vacuum boundary conditions, whilst large grids are required with AMR codes to prevent the flow disappearing from the edge of the computational domain. As SPH is a Lagrangian method, advection of flow properties is inherent, whereas this presents problems for AMR codes, and which usually entails an unphysical increase in entropy. In a similar manner, SPH codes can be implemented in such a manner that they are inherently conservative of mass, momentum and energy, and similarly, unless it is explicitly added in shocks, they likewise conserve entropy. Nonetheless it is emphatically **not** true to say that either SPH or grid code methods are "better" than the other, simply that the more appropriate approach should be chosen for any given problem, and indeed greater confidence in the results will ensue if the two methods concur.

Having said that, throughout this chapter I shall however consider only the SPH approach, as it is this one that I have used to generate all the results discussed henceforth. Furthermore, as all the problems I have considered have been fully three-dimensional, throughout this chapter I shall consider only the derivation and discussion of SPH in 3D. This chapter is therefore structured as follows: In Section 3.2 I shall introduce the basic concepts of the SPH method, then in Section 3.3 this is used to re-write the fluid equations in a manner that can be solved numerically. Section 3.4 discusses the dissipative processes required for the correct implementation of artificial viscosity and the introduction of entropy in shock waves, and then in Section 3.5 I discuss how the SPH formalism may be made more adaptive still by the self-consistent inclusion of variable smoothing lengths. As many astrophysical problems are strongly influenced by gravitational forces I detail how these may be implemented in Section 3.6. In Section 3.7 I briefly summarise the methods used to find the nearest neighbours, and then in Section 3.8 I consider how the code is evolved forward in time, and various time-stepping criteria. Finally in Section 3.9 I

---

[1]http://wiki.manchester.ac.uk/spheric/index.php/SPHERIC_Home_Page

briefly outline the properties of the code I have used, point the reader in the direction of some standard numerical tests used for code evaluation and consider further extensions to the method.

3.2 SPH Basics

In the following section I shall discuss the derivation of the SPH formalism from first principles, showing how a continuous field can be mapped on to (and thus approximated by) a series of discrete particles, and the errors involved in this approximation. I then show how derivatives may be calculated, and discuss ways in which the particles may be suitably smoothed to represent the field.

3.2.1 Discrete Approximations to a Continuous Field

We start from the (mathematically) trivial identity

$$f(\mathbf{r}) = \int_V f(\mathbf{r}') \, \delta(\mathbf{r} - \mathbf{r}') \, \mathrm{d}\mathbf{r}', \tag{3.1}$$

where $f(\mathbf{r})$ is any (scalar) function defined on a three-dimensional co-ordinate system $\mathbf{r}$ ranging over a volume V. Similarly, $\delta(\mathbf{r})$ is the Dirac delta function, and $\mathbf{r}'$ is a dummy variable also ranging over V.

We may generalise the delta function to a so-called smoothing kernel W with a characteristic width h (known as the smoothing length) such that

$$\lim_{h \to 0} W(\mathbf{r}, h) = \delta(\mathbf{r}), \tag{3.2}$$

subject to the normalisation

$$\int_V W(\mathbf{r}, h) \, \mathrm{d}\mathbf{r}' = 1. \tag{3.3}$$

By expanding $W(\mathbf{r} - \mathbf{r}', h)$ as a Taylor series, it can be shown that for symmetric kernels $W(\mathbf{r} - \mathbf{r}', h) = W(\mathbf{r}' - \mathbf{r}, h)$, equation 3.1 becomes

$$f(\mathbf{r}) = \int_V f(\mathbf{r}') \, W(\mathbf{r} - \mathbf{r}', h) \, \mathrm{d}\mathbf{r}' + \mathcal{O}(h^2), \tag{3.4}$$

the second order accuracy arising from the vanishing of the kernel gradient at $\mathbf{r}' = \mathbf{r}$ (see for instance Price 2005; Benz 1990; Monaghan 1992). Note that more elaborate kernels accurate to $\mathcal{O}(h^4)$ can be constructed, but these suffer from the problem that $W(\mathbf{r}, h)$ can become negative in certain ranges (Price, 2005; Monaghan, 1992), thus potentially leading to negative density evaluations in certain pathological situations.

Nonetheless for a second order, symmetric kernel, for any finite density $\rho(\mathbf{r})$ within V, equation 3.4 is exactly equivalent to

$$f(\mathbf{r}) = \int_V \frac{f(\mathbf{r}')}{\rho(\mathbf{r}')} \, W(\mathbf{r} - \mathbf{r}', h) \, \rho(\mathbf{r}')\mathrm{d}\mathbf{r}' + \mathcal{O}(h^2). \tag{3.5}$$

Discretising this continuous field on to a series of particles of (potentially variable) mass $m = \rho(\mathbf{r}')\mathrm{d}\mathbf{r}'$, the original identity equation 3.1 becomes

$$f(\mathbf{r}) \approx \sum_i \frac{m_i}{\rho_i} f(\mathbf{r}_i) \, W(\mathbf{r} - \mathbf{r}_i, h), \tag{3.6}$$

where now $f(\mathbf{r}_i)$, m_i and $\rho_i = \rho(\mathbf{r}_i)$ are the scalar value, mass and density of the ith particle, and i ranges over all particles within the smoothing kernel. Equation 3.6 therefore represents the discrete approximation to the continuous scalar field f at position $\mathbf{r}$ in the computational domain V, and is thus the basis of all SPH formalisms. Note that the position $\mathbf{r}$ at which the function f is approximated is completely general and is not restricted to particle positions, although in practice this is where the values are actually evaluated.

3.2.2 Spatial Derivatives and Vector Calculus

In order for the SPH discretisation of a field to be useful as a method of solving fluid flows, it is clear that the spatial derivatives of any given quantity must also have a suitable approximate form[2]. Here therefore, I summarise the SPH approximations for various vector calculus quantities.

3.2.2.1 Gradient of a Scalar Field

The approximation for the gradient of a scalar field can be derived by taking the spatial derivative of equation 3.1, and applying the smoothing kernel. Noting that

---

[2]Temporal derivatives are naturally also required, and these will be discussed in due course

$\nabla \equiv \partial/\partial\mathbf{r}$, we therefore see that

$$\nabla f(\mathbf{r}) \;=\; \frac{\partial}{\partial\mathbf{r}} \int_V f(\mathbf{r}') \, \delta(\mathbf{r} - \mathbf{r}') \, \mathrm{d}\mathbf{r}' \tag{3.7}$$

$$=\; \frac{\partial}{\partial\mathbf{r}} \int_V f(\mathbf{r}') \, W(\mathbf{r} - \mathbf{r}', h) \, \mathrm{d}\mathbf{r}' + \mathcal{O}(h^2), \tag{3.8}$$

in a similar manner to equation 3.4. Given that the only part to depend on $\mathbf{r}$ is the smoothing kernel W, and again introducing the density $\rho(\mathbf{r}')$ in both the numerator and the denominator we obtain

$$\nabla f(\mathbf{r}) = \int_V \frac{f(\mathbf{r}')}{\rho(\mathbf{r}')} \frac{\partial}{\partial\mathbf{r}} W(\mathbf{r} - \mathbf{r}', h)\rho(\mathbf{r}') \, \mathrm{d}\mathbf{r}' + \mathcal{O}(h^2). \tag{3.9}$$

Finally this may be discretised in the same way as before, to give

$$\nabla f(\mathbf{r}) \approx \sum_i \frac{m_i}{\rho_i} f(\mathbf{r}_i) \, \nabla W(\mathbf{r} - \mathbf{r}_i, h) \tag{3.10}$$

as an estimator for the gradient of a scalar field $f(\mathbf{r})$. Notable from the above result is that the gradient of a scalar field can be approximated by the values of the field itself along with the gradient of the kernel. Computationally this is very useful as at no point does ∇f have to be evaluated for any particle, whilst the gradient of the kernel will be known explicitly for any sensible choice of W.

3.2.2.2 Divergence of a Vector Field

Although equation 3.1 was given only for a scalar field, a similar identity may be given for a vector field $\mathbf{F}(\mathbf{R})$, namely

$$\mathbf{F}(\mathbf{r}) = \int_V \mathbf{F}(\mathbf{r}') \, \delta(\mathbf{r} - \mathbf{r}') \, \mathrm{d}\mathbf{r}', \tag{3.11}$$

Taking the divergence of this with respect to $\mathbf{r}$, and noting once again that the only term to depend on $\mathbf{r}$ is the smoothing kernel we find that the integral approximation becomes

$$\nabla \cdot \mathbf{F}(\mathbf{r}) = \int_V \mathbf{F}(\mathbf{r}') \cdot \nabla W(\mathbf{r} - \mathbf{r}', h) \mathrm{d}\mathbf{r}' + \mathcal{O}(h^2), \tag{3.12}$$

and thus as before this can be discretised to obtain the approximation

$$\nabla \cdot \mathbf{F}(\mathbf{r}) \approx \sum_i \frac{m_i}{\rho_i} \, \mathbf{F}(\mathbf{r}_i) \cdot \nabla W(\mathbf{r} - \mathbf{r}_i, h). \tag{3.13}$$

3.2.2.3   Curl of a Vector Field

By a precisely similar argument, it is possible to show that the curl of a vector $\mathbf{F}$, $\nabla \times \mathbf{F}$ can be approximated using

$$\nabla \times \mathbf{F}(\mathbf{r}) \approx \sum_i \frac{m_i}{\rho_i} \, \mathbf{F}(\mathbf{r}_i) \times \nabla W(\mathbf{r} - \mathbf{r}_i, h), \tag{3.14}$$

although this is relatively little used unless magnetohydrodynamic (MHD) effects are being taken into account.

3.2.3   Errors

The approximations given in equations 3.6, 3.10, 3.13 and 3.14 encompass both the $\mathcal{O}(h^2)$ errors of considering only the integral term, and also the errors inherent in the discretisation (which arise due to incomplete sampling of the smoothing kernel). In the former case we see that the $\mathcal{O}(h^2)$ errors are reduced by decreasing the smoothing length, while the discretisation (sampling) errors are minimised by increasing the number of particles within the smoothing kernel. Barring numerical stability issues (Read et al., 2010), this discrete approximation is therefore at its most accurate with large numbers of particles contained within a small smoothing length. However, this must be balanced against the need for computational speed and efficiency, and hence there is a compromise to be struck.

These errors are neatly illustrated by considering the approximations to a constant function $f(\mathbf{r}) \equiv 1$ and the zero function, which can be obtained by noting that with this definition of f, $\nabla f(\mathbf{r}) = 0$. The SPH approximations for one and zero therefore become

$$1 \quad \approx \quad \sum_i \frac{m_i}{\rho_i} \, W(\mathbf{r} - \mathbf{r}_i, h), \tag{3.15}$$

$$0 \quad \approx \quad \sum_i \frac{m_i}{\rho_i} \, \nabla W(\mathbf{r} - \mathbf{r}_i, h). \tag{3.16}$$

Since in neither case does the equation reduce to an identity, we see that there are

inherent errors in estimating even constant functions. Nonetheless, with suitable choices for the number of particles within the smoothing kernel and the smoothing length, these may be kept to an acceptable level. For a more detailed derivation and discussion of these errors, the reader is directed to Price (2005); Monaghan (1992); Benz (1990) and Read et al. (2010).

3.2.4   Improved Approximations for Spatial Gradients

Although the approximations given in equations 3.6, 3.10, 3.13 and 3.14 are those that arise most readily from the SPH approximation, it is possible to construct other estimators for the gradient of a scalar field. For instance, by noting that for any quantity $f(\mathbf{r}) \equiv 1.f(\mathbf{r})$, we see that

$$\nabla f(\mathbf{r}.1) = 1.\nabla f(\mathbf{r}) + f(\mathbf{r})\nabla 1 \tag{3.17}$$

and therefore that

$$\nabla f(\mathbf{r}) = \nabla f(\mathbf{r}) - f(\mathbf{r})\nabla 1. \tag{3.18}$$

Clearly, since $\nabla 1 = 0$ these forms should be identical. From equation 3.16 however, we see that the SPH approximation for $\nabla 1$ is non-zero, and thus using equation 3.18 we may define another estimate for $\nabla f(\mathbf{r})$ as

$$
\begin{aligned}
\nabla f(\mathbf{r}) &= \sum_i \frac{m_i}{\rho_i} f(\mathbf{r}_i)\,\nabla W(\mathbf{r}-\mathbf{r}_i, h) - f(\mathbf{r}) \sum_i \frac{m_i}{\rho_i}\,\nabla W(\mathbf{r}-\mathbf{r}_i, h) \tag{3.19}\\
&= \sum_i m_i \frac{f(\mathbf{r}_i)-f(\mathbf{r})}{\rho_i}\,\nabla W(\mathbf{r}-\mathbf{r}_i, h). \tag{3.20}
\end{aligned}
$$

This approximation clearly has the advantage that it vanishes identically for constant functions.

A more general class of interpolants arises from considering the vector calculus identity

$$\nabla(f\rho^n) \equiv nf\rho^{n-1}\nabla\rho + \rho^n\nabla f, \tag{3.21}$$

valid for all $n \in \mathbb{R}$. This in turn leads to the following identity for ∇f

$$\nabla f \equiv \frac{1}{\rho^n}\left[\nabla(f\rho^n) - nf\rho^{n-1}\nabla\rho\right]. \tag{3.22}$$

Substituting ρ and $f\rho^n$ into equation 3.10, we obtain a general interpolant for ∇f,

such that

$$\nabla f(\mathbf{r}) = \frac{1}{\rho(\mathbf{r})^n} \sum_i m_i \left(f(\mathbf{r}_i)\rho(\mathbf{r}_i)^{n-1} - n f(\mathbf{r})\rho(\mathbf{r})^{n-1} \right) \nabla W(\mathbf{r} - \mathbf{r}_i, h). \qquad (3.23)$$

Two instances of this general case turn out to be particularly useful, namely where $n = 1$ and $n = -1$. For the former case we obtain

$$\nabla f(\mathbf{r}) = \frac{1}{\rho(\mathbf{r})} \sum_i m_i \left(f(\mathbf{r}) - f(\mathbf{r}_i) \right) \nabla W(\mathbf{r} - \mathbf{r}_i, h). \qquad (3.24)$$

This is very similar in form to that given in equation 3.20, with the exception that knowledge of the density at $\mathbf{r}$ is required *a priori*. Although no longer anti-symmetric in $f(\mathbf{r})$ and f_i, it is nonetheless exact for constant functions.

In the case where $n = -1$ we obtain

$$\nabla f(\mathbf{r}) = \rho(\mathbf{r}) \sum_i m_i \left(\frac{f(\mathbf{r})}{\rho(\mathbf{r})^2} + \frac{f(\mathbf{r}_i)}{\rho(\mathbf{r}_i)^2} \right) \nabla W(\mathbf{r} - \mathbf{r}_i, h). \qquad (3.25)$$

While this form is no longer exact for constant functions, it is commonly used as an estimator for the pressure gradient $(\nabla P)/\rho$, as it is pairwise symmetric and as such ensures conservation of momentum. This is also the form of the gradient that arises naturally from a Lagrangian formulation of the fluid equations, as I shall show in Section 3.3.2.

3.2.5 Improved Divergence Estimates

In a similar manner to the gradient, improved estimates can be made for the divergence of a vector field. By noting that $\mathbf{F}(\mathbf{r}) = 1.\mathbf{F}(\mathbf{r})$, the estimate

$$\nabla \cdot \mathbf{F}(\mathbf{r}) \approx \sum_i \frac{m_i}{\rho_i} \left(\mathbf{F}(\mathbf{r}_i) - \mathbf{F}(\mathbf{r}) \right) \cdot \nabla W(\mathbf{r} - \mathbf{r}_i, h). \qquad (3.26)$$

can be arrived at, which again becomes exact for constant functions. In a similar manner to the expansion given in equation 3.21, a general class of estimates can be arrived at by considering the identity

$$\nabla \cdot (\rho^n \mathbf{F}) = \rho^n \nabla \cdot \mathbf{F} + n\rho^{n-1} \mathbf{F} \cdot \nabla \rho, \qquad (3.27)$$

the $n = 1, -1$ cases of which are given by

$$\nabla \cdot \mathbf{F}(\mathbf{r}) \approx \frac{1}{\rho(\mathbf{r})} \sum_i m_i \left(\mathbf{F}(\mathbf{r}_i) - \mathbf{F}(\mathbf{r}) \right) \cdot \nabla W(\mathbf{r} - \mathbf{r}_i, h) \tag{3.28}$$

and

$$\nabla \cdot \mathbf{F}(\mathbf{r}) \approx \rho(\mathbf{r}) \sum_i m_i \left(\frac{\mathbf{F}(\mathbf{r}_i)}{\rho(\mathbf{r}_i)^2} + \frac{\mathbf{F}(\mathbf{r})}{\rho(\mathbf{r})^2} \right) \cdot \nabla W(\mathbf{r} - \mathbf{r}_i, h) \tag{3.29}$$

respectively. Once again these estimates have the advantages of being exact for constant functions in the former case and pairwise symmetric in $(\nabla \cdot \mathbf{F})/\rho$ in the latter case.

3.2.6   Smoothing Kernels

From the above it is clear that the choice of smoothing kernel is an important one. It must by definition obey the criteria set out in equations 3.2 and 3.3 in that it must tend to a δ-function as $h \to 0$ and it must be normalised so the area under the curve is unity. For the purposes of calculating the gradients of quantities it is also clear that it should have a continuous and well defined first derivative, and from a symmetry argument it should be spherically symmetric, and thus depend only on $r = |\mathbf{r} - \mathbf{r}'|$ and h.

One of the first choices for the smoothing kernel was the Gaussian function, such that

$$W(r, h) = \frac{1}{h^3 \pi^{3/2}} e^{-x^2}, \tag{3.30}$$

where $x = r/h$. However, this has the drawback that $W > 0$ for all r, and thus all particles within the computational domain contribute. The computational cost of such a kernel therefore scales as $\mathcal{O}(N^2)$, where N is the number of particles in the simulation. Given that (for purely hydrodynamical quantities) long range forces are negligible, it makes sense to restrict the kernel to those with compact support, i.e. make them subject to the condition that $W(r, h) = 0$ where $r/h > k$ for some constant k. This means that the computational cost scales as $\mathcal{O}(N N_{\text{neigh}})$, where N_{neigh} is the average number of particles within a sphere of radius $r = kh$ about any one particle.

For this reason, cubic spline kernels are often used (see Monaghan & Lattanzio

1985 for instance), where the kernel is defined as

$$W(r,h) = \frac{1}{\pi h^3} \begin{cases} 1 - \frac{3}{2}x^2 + \frac{3}{4}x^3 & 0 \le x \le 1; \\ \frac{1}{4}(2-x)^3 & 1 \le x \le 2; \\ 0 & x \ge 2, \end{cases} \tag{3.31}$$

where $x = r/h$ as in equation 3.30. Here only particles within $2h$ of the central particle contribute to the smoothing kernel, which is spherically symmetric and smoothly differentiable for all r. Although many other kernels are possible (see Monaghan 1992; Fulk & Quinn 1996; Price 2005; Read et al. 2010 for example) this is a commonly used kernel, and is the one present in the code I have used throughout.

Note from the above the gradient of the kernel is well defined for all values of x, such that

$$\nabla W(r,h) = \frac{\partial}{\partial r}W(r,h) \tag{3.32}$$

$$= \frac{1}{\pi h^4} \begin{cases} \frac{9}{4}x^2 - 3x & 0 \le x \le 1 \\ -\frac{3}{4}(2-x)^2 & 1 \le x \le 2 \\ 0 & x \ge 2 \end{cases}. \tag{3.33}$$

Finally it is worth noting that in general the form of the kernel makes little overall difference to the computational speed of the code. This is because most codes tabulate the values of both the kernel and its gradient rather than compute them directly, and thus the form of the kernel may be as simple or as complex as required, even (theoretically at least) to the extent of being non-analytic functions.

3.3   Fluid Equations

Given that the SPH formalism has now been put on a sound mathematical footing, in this section I shall use it to obtain approximations to the equations governing fluid motion, such that they can be used to construct a viable numerical algorithm for solving fluid flows. For the dual purposes of brevity and simplicity I shall here consider only the case of an inviscid compressible flow in the absence of body forces, although the inclusion of both gravity and (artificial) viscosity will be discussed in due course. First however, it is useful to summarise the principal equations of

motion in their standard conservative form.

The continuity (conservation of mass) equation is given by

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \tag{3.34}$$

where as normal, ρ is the density, t is time and $\mathbf{v}$ is velocity.

The Euler equation gives the equations of motion in the case of an inviscid fluid, and encapsulates the conservation of momentum. In the absence of external (body) forces it becomes

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v}) + \nabla P = \mathbf{0}, \tag{3.35}$$

where P is the fluid pressure and $\otimes$ represents the outer or tensor product[3]. For compressible flows it is also necessary to take into account the energy equation, and as such the conservation of energy is embodied in the following equation;

$$\frac{\partial u}{\partial t} + \nabla \cdot [(u + P)\mathbf{v}] = 0, \tag{3.36}$$

where u is the specific internal energy and $v = |\mathbf{v}|$ is the magnitude of the velocity vector. Finally it is worth noting that these five equations (there are 3 components to the momentum equation) contain six unknowns (ρ, the three components of velocity v_x, v_y and v_z, P and u). Therefore in order to solve the system we require a further constraint; an equation of state is required. All the analysis I shall present henceforth uses the ideal gas equation of state, where

$$P = \kappa(s)\rho^\gamma, \tag{3.37}$$
$$= (\gamma - 1)u\rho, \tag{3.38}$$

where γ is the adiabatic index (the ratio of specific heats), which throughout has been set to 5/3, and $\kappa(s)$ is the adiabat, itself a function of the specific entropy s. In the case of isentropic flows, s and (thus κ) remains constant.

I shall now discuss the SPH formulation of each of the continuity, momentum and energy equations in turn. Note that again for the purposes of brevity I assume that the smoothing length is held constant (i.e. $\dot{h} = 0$, where the dot denotes the derivative with respect to time), and is equal for all particles. Individual, variable smoothing lengths will be discussed in due course. Furthermore, I assume through-

---

[3]The outer product of two vectors may be summarised as $\mathbf{A} \otimes \mathbf{B} = \mathbf{A}\mathbf{B}^T = A_i B_j$ (in indicial notation).

out that the mass of each particle is held constant, such that $m_i = const$, and again that all particles are of equal mass. Although it is possible to have individually varying particle masses, the code I use does not have this feature, and therefore I have not included a discussion of it here. Finally note that from here onwards, all the approximations are evaluated at specific particle positions, as this is how the SPH algorithm is implemented within particle-based codes.

3.3.1   Conservation of Mass

Using equation 3.6, we see that in the case of the density, the SPH approximation becomes very simple, namely that at particle j the density ρ_j becomes

$$
\begin{aligned}
\rho_j &= \sum_i m_i W(\mathbf{r}_j - \mathbf{r}_i, h), \\
&= \sum_i m_i W_{ji},
\end{aligned}
\tag{3.39}
$$

where we write $W_{ji} = W(\mathbf{r}_j - \mathbf{r}_i, h)$, and where by symmetry, $W_{ji} = W_{ij}$. Note that here and henceforth, as the SPH formalism is a discrete approximation to the underlying continuous medium, we assume equality between the estimator on the RHS and the SPH quantity on the LHS.

Taking the full time derivative of equation 3.39 we obtain

$$
\frac{\mathrm{d}\rho_j}{\mathrm{d}t} = \sum_i m_i \left[\frac{\partial W_{ji}}{\partial \mathbf{r}_j} \cdot \frac{\mathrm{d}\mathbf{r}_j}{\mathrm{d}t} + \frac{\partial W_{ji}}{\partial \mathbf{r}_i} \cdot \frac{\mathrm{d}\mathbf{r}_i}{\mathrm{d}t} + \frac{\partial W_{ji}}{\partial h} \frac{\mathrm{d}h}{\mathrm{d}t} \right],
\tag{3.40}
$$

and noting that

$$
\frac{\mathrm{d}\mathbf{r}_j}{\mathrm{d}t} = \mathbf{v}_j, \quad \frac{\mathrm{d}\mathbf{r}_i}{\mathrm{d}t} = \mathbf{v}_i, \quad \frac{\mathrm{d}h}{\mathrm{d}t} = 0,
$$

we find that the time derivative of density becomes

$$
\begin{aligned}
\frac{\mathrm{d}\rho_j}{\mathrm{d}t} &= \sum_i m_i \left(\mathbf{v}_j \cdot \nabla_j W_{ji} + \mathbf{v}_i \cdot \nabla_i W_{ji} \right) \\
&= \sum_i m_i \, \mathbf{v}_{ji} \cdot \nabla_j W_{ji}
\end{aligned}
\tag{3.41}
$$

where we use $\mathbf{v}_{ji} = \mathbf{v}_j - \mathbf{v}_i$, and where we note that the gradient of the kernel is antisymmetric, i.e. that

$$
\nabla_i W_{ji} = -\nabla_j W_{ji}.
\tag{3.42}
$$

From equation 3.28, we note that the RHS of equation 3.41 is simply an estimator of $-\rho_j \nabla_j \cdot \mathbf{v}_j$. Hence equation 3.41 becomes

$$\frac{\mathrm{d}\rho_j}{\mathrm{d}t} = -\rho_j \nabla_j \cdot \mathbf{v}_j, \tag{3.43}$$

which is simply a reformulation of the continuity equation equation 3.34 using the Lagrangian time derivative

$$\frac{\mathrm{d}}{\mathrm{d}t} = \frac{\partial}{\partial t} + (\mathbf{v} \cdot \nabla), \tag{3.44}$$

in which the second term accounts for the advection of flow properties through the fluid. Therefore we see that the SPH estimate for density equation 3.39 is *automatically* conservative of mass (as long as equation 3.28 is used as an estimate for the divergence of velocity).

3.3.2   Conservation of Momentum

Although there are various ways of deriving the equations of motion consistently with the SPH framework, a particularly appealing one is to use the Lagrangian formalism. As long as the discrete Lagrangian functional preserves the fundamental symmetries of the underlying continuous one, this confers the inherent advantages that the resulting SPH equations of motion will automatically fulfil the requisite conservation laws (through Noether's Theorem) and also that the only approximations made are in the discretisation of the Lagrangian itself.

3.3.2.1   Linear Momentum

Defined as the total kinetic energy of the system minus the total internal energy (for purely hydrodynamical flows), the Lagrangian functional $\mathcal{L}$ for the fluid is

$$\mathcal{L}(\mathbf{r}, \mathbf{v}) = \int_V \frac{1}{2}\rho\mathbf{v} \cdot \mathbf{v} - \rho u \; \mathrm{d}\mathbf{r}, \tag{3.45}$$

where as before, u is the specific internal energy. For later simplicity, we note that through the equation of state (equation 3.38) the specific internal energy is a function of density and pressure $u = u(\rho, P)$, which in turn are functions of position. This gives $u = u(\mathbf{r})$. Now if we again make the discretisation $m_i = \rho\mathrm{d}\mathbf{r}$, the SPH

estimate of the Lagrangian becomes

$$\mathcal{L}(\mathbf{r}, \mathbf{v}) = \sum_i m_i \left(\frac{1}{2} \mathbf{v}_i \cdot \mathbf{v}_i - u_i(\mathbf{r}_i) \right), \tag{3.46}$$

where i ranges over all particles.

The equations of motion for particle j are obtained from the Lagrangian through the Euler-Lagrange equations, as follows;

$$\frac{\mathrm{d}}{\mathrm{d}t} \left(\frac{\partial \mathcal{L}}{\partial \mathbf{v}_j} \right) - \frac{\partial \mathcal{L}}{\partial \mathbf{r}_j} = \mathbf{0}. \tag{3.47}$$

By considering each of the terms in this equation it is therefore possible to obtain an SPH approximation to the equations of motion that remains fully conservative. If we therefore consider the derivative of the Lagrangian with respect to the velocity at particle j, we find

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{v}_j} &= \frac{\partial}{\partial \mathbf{v}_j} \sum_i m_i \left(\frac{1}{2} \mathbf{v}_i \cdot \mathbf{v}_i - u_i(\mathbf{r}_i) \right), \\ &= m_j \mathbf{v}_j, \end{aligned} \tag{3.48}$$

noting that since the velocities are independent the differential is zero unless $i = j$.

Considering now the second term in the Euler-Lagrange equation 3.47 we find that

$$\frac{\partial \mathcal{L}}{\partial \mathbf{r}_j} = -\sum_i \left[\frac{\partial u_i}{\partial P_i} \frac{\partial P_i}{\partial \mathbf{r}_j} + \frac{\partial u_i}{\partial \rho_i} \frac{\partial \rho_i}{\partial \mathbf{r}_j} \right], \tag{3.49}$$

where we have used equation 3.38 to obtain the full derivative of the internal energy. In the isentropic (dissipationless) case we see that $\kappa(s)$ is constant, and thus the pressure is a function of density only, leading to

$$\frac{\partial \mathcal{L}}{\partial \mathbf{r}_j} = -\sum_i \left[\frac{\partial u_i}{\partial P_i} \frac{\mathrm{d} P_i}{\mathrm{d} \rho_i} + \frac{\partial u_i}{\partial \rho_i} \right] \frac{\partial \rho_i}{\partial \mathbf{r}_j}. \tag{3.50}$$

From the equation of state equation 3.38 we find that

$$\frac{\partial u_i}{\partial P_i} \frac{\mathrm{d} P_i}{\mathrm{d} \rho_i} + \frac{\partial u_i}{\partial \rho_i} = \frac{P_i}{\rho_i^2}, \tag{3.51}$$

and thus the derivative of the Lagrangian with respect to the position of particle j

becomes

$$\frac{\partial \mathcal{L}}{\partial \mathbf{r}_j} = -\sum_i \frac{P_i}{\rho_i^2} \frac{\partial \rho_i}{\partial \mathbf{r}_j}. \qquad (3.52)$$

Using equation 3.39 we find that

$$\frac{\partial \rho_i}{\partial \mathbf{r}_j} = \sum_k m_i \frac{\partial W_{ik}}{\partial \mathbf{r}_j} \qquad (3.53)$$

$$= \sum_k m_i \frac{\partial W_{ik}}{\partial r_{ik}} \frac{\partial r_{ik}}{\partial \mathbf{r}_j}, \qquad (3.54)$$

where we take $r_{ik} = |\mathbf{r}_{ik}|$, and use the fact that the kernel is spherically symmetric. By direct differentiation,

$$\frac{\partial r_{ik}}{\partial \mathbf{r}_j} = (\delta_{ij} - \delta_{kj})\hat{\mathbf{r}}_{ik}, \qquad (3.55)$$

with $\hat{\mathbf{r}}_{ik} = \mathbf{r}_{ik}/r_{ik}$ the unit vector in the direction of $\mathbf{r}_{ik}$. Substituting this back into equation 3.54 we find that

$$\frac{\partial \rho_i}{\partial \mathbf{r}_j} = \sum_k m_k \frac{\partial W_{ik}}{\partial r_{ik}} (\delta_{ij} - \delta_{kj})\hat{\mathbf{r}}_{ik} \qquad (3.56)$$

$$= \sum_k m_k \nabla_j W_{ik} \ (\delta_{ij} - \delta_{kj}), \qquad (3.57)$$

where in the second case we have used the fact that $\partial/\partial \mathbf{r}_j \equiv \nabla_j$.

With reference to equation 3.52 we find therefore that

$$\frac{\partial \mathcal{L}}{\partial \mathbf{r}_j} = -\sum_i m_i \frac{P_i}{\rho_i^2} \sum_k m_k \nabla_j W_{ik} (\delta_{ij} - \delta_{kj}) \qquad (3.58)$$

$$= -m_j \frac{P_j}{\rho_j^2} \sum_k m_k \nabla_j W_{jk} - \sum_i m_i m_j \frac{P_i}{\rho_i^2} \nabla_j W_{ij} \qquad (3.59)$$

$$= -m_j \sum_i m_i \left(\frac{P_j}{\rho_j^2} + \frac{P_i}{\rho_i^2} \right) \nabla_j W_{ji}, \qquad (3.60)$$

where we have changed the summation index in the first term to i and used the fact that the gradient of the kernel is antisymmetric, i.e. that $\nabla_j W_{kj} = -\nabla_j W_{jk}$. Finally, by substituting equations 3.48 and 3.60 into equation 3.47 and dividing through by the common factor m_j, we find that the SPH equations of motion become

$$\frac{d\mathbf{v}_j}{dt} = -\sum_i m_i \left(\frac{P_j}{\rho_j^2} + \frac{P_i}{\rho_i^2} \right) \nabla_j W_{ji}. \qquad (3.61)$$

Since this equation is pairwise symmetric in i, j, it is clear that the pressure force on particle j due to particle i is equal and opposite (due to the antisymmetry of the kernel gradient) to the force on particle i from particle j. In this manner, it is clear that this formulation of the equation of motion conserves linear momentum by construction.

3.3.2.2 Angular Momentum

To check that angular momentum $\mathbf{L} = \mathbf{r} \times m\mathbf{v}$ is conserved, we note that its derivative with respect to time should be zero. By using equation 3.61 we see that the time derivative of the angular momentum of particle j is given by

$$\frac{\mathrm{d}\mathbf{L}_j}{\mathrm{d}t} = m_j \mathbf{v}_j \times \mathbf{v}_j + m_j \mathbf{r}_j \times \frac{\mathrm{d}\mathbf{v}_j}{\mathrm{d}t} \tag{3.62}$$

$$= -m_j \sum_i m_i \left(\frac{P_j}{\rho_j^2} + \frac{P_i}{\rho_i^2} \right) \mathbf{r}_j \times \nabla_i W_{ij}, \tag{3.63}$$

since by definition $\mathbf{v}_j \times \mathbf{v}_j = \mathbf{0}$. The total time derivative of the angular momentum is therefore given by the sum over all particles j, such that

$$\frac{\mathrm{d}\mathbf{L}}{\mathrm{d}t} = -\sum_j \sum_i m_j m_i \left(\frac{P_j}{\rho_j^2} + \frac{P_i}{\rho_i^2} \right) \mathbf{r}_j \times \nabla_i W_{ij}. \tag{3.64}$$

Hence we see that by reversing the summation indices the entire sum is *anti*symmetric in i and j, i.e.

$$\frac{\mathrm{d}\mathbf{L}}{\mathrm{d}t} = -\sum_j \sum_i m_j m_i \left(\frac{P_j}{\rho_j^2} + \frac{P_i}{\rho_i^2} \right) \mathbf{r}_j \times \nabla_i W_{ij}, \tag{3.65}$$

$$= \sum_i \sum_j m_i m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right) \mathbf{r}_i \times \nabla_j W_{ji}, \tag{3.66}$$

which can only be the case where the total sum is zero. Hence the angular momentum is constant with time, and thus angular momentum is explicitly conserved.

3.3.3 Conservation of Energy

In the case of a purely hydrodynamical flow, the total energy $E = \rho u + \rho v^2/2$ is given by the sum of the kinetic and internal energies, such that the SPH estimator

becomes

$$E = \sum_i m_i \left(\frac{1}{2} \mathbf{v}_i \cdot \mathbf{v}_i + u_i \right). \tag{3.67}$$

Clearly, where energy is conserved the time derivative of the total energy should be zero. Taking the time derivative therefore, we find that

$$\frac{\mathrm{d}E}{\mathrm{d}t} = \sum_i m_i \left(\mathbf{v}_i \cdot \frac{\mathrm{d}\mathbf{v}_i}{\mathrm{d}t} + \frac{\partial u_i}{\partial P_i} \frac{\mathrm{d}P_i}{\mathrm{d}t} + \frac{\partial u_i}{\partial \rho_i} \frac{\mathrm{d}\rho_i}{\mathrm{d}t} \right), \tag{3.68}$$

$$= \sum_i m_i \left(\mathbf{v}_i \cdot \frac{\mathrm{d}\mathbf{v}_i}{\mathrm{d}t} + \frac{P_i}{\rho_i^2} \frac{\mathrm{d}\rho_i}{\mathrm{d}t} \right), \tag{3.69}$$

where we have again used the fact that in the dissipationless case $P = P(\rho)$ and we can therefore amalgamate the latter two terms of the RHS of equation 3.68 using equation 3.51. Using also the equation of motion derived above in equation 3.61 and $\mathrm{d}\rho/\mathrm{d}t$ from the continuity equation 3.41 we therefore find that

$$\frac{\mathrm{d}E}{\mathrm{d}t} = -\sum_i m_i \mathbf{v}_i \cdot \sum_j m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right) \nabla_j W_{ji}$$

$$+ \sum_i m_i \frac{P_i}{\rho_i^2} \sum_j m_j (\mathbf{v}_i - \mathbf{v}_j) \cdot \nabla_j W_{ji} \tag{3.70}$$

$$= \sum_i \sum_j m_i m_j \left(\frac{P_j}{\rho_j^2} \mathbf{v}_i + \frac{P_i}{\rho_i^2} \mathbf{v}_j \right) \cdot \nabla_j W_{ji}, \tag{3.71}$$

where we have again used the fact that the kernel is antisymmetric to obtain equation 3.71. Now using the same argument we used to show that angular momentum is conserved, we note that equation 3.71 is antisymmetric under a reversal of i and j, and thus must be equal to zero. Hence we find that

$$\frac{\mathrm{d}E}{\mathrm{d}t} = 0, \tag{3.72}$$

and therefore that the total energy is also explicitly conserved.

A corollary of this is that the time derivative of the internal energy is given by the second term on the RHS of equation 3.69, such that

$$\frac{\mathrm{d}u_j}{\mathrm{d}t} = \frac{P_j}{\rho_j^2} \frac{\mathrm{d}\rho_j}{\mathrm{d}t}, \tag{3.73}$$

$$= \frac{P_j}{\rho_j^2} \sum_i m_i (\mathbf{v}_j - \mathbf{v}_i) \cdot \nabla_i W_{ji}, \tag{3.74}$$

and indeed this is how the internal energy is evolved within SPH codes.

It is worth noting that the formulation of SPH outlined above is therefore explicitly conservative of mass, momentum (in both forms) and energy. Hence, while there are inevitably errors inherent in the SPH discretisation of a continuous medium, these are the *only* errors that appear, at least in the case of a dissipationless hydrodynamical flow.

3.4 Dissipative Effects

So far we have assumed the fluid flow to be barotropic (i.e. $P = P(\rho)$), and polytropic, with the polytropic index set equal to the adiabatic index γ, the ratio of specific heats. This in turn means that the flow is isentropic, and therefore completely dissipationless. While this is an adequate approximation for many incompressible, inviscid and unshocked compressible flows, it presents serious problems when it comes to modelling transonic and supersonic flow regimes, as the conversion of mechanical (kinetic) energy into heat (internal) energy is not correctly captured. The problem occurs because at a shock front, flow properties such as the velocity, pressure, density and entropy change very rapidly, on the order of the mean free path of the gas particles. On large scales therefore these changes appear discontinuous, and flow solvers that do not resolve the mean free path (which is all of them) break down due to the apparently singular flow gradients.

There are two principal workarounds that allow numerical codes to solve shocked flows. One is to use a Riemann solver in a Gudonov-type code (see for instance Inutsuka 2002; Cha & Whitworth 2003), but I shall not go into any detail here as this is not the approach used in the code I have used. The alternative approach, used in the majority of SPH codes, is to broaden the shock across a small number of smoothing lengths. This ensures that the flow gradients do *not* become infinite, and gives the correct asymptotic behaviour away from the shock. This latter method is implemented by including an *artificial* dissipative term in the momentum and energy equations that is triggered only in the presence of shocks, and it is this method that I shall consider here.

3.4.1   Standard Artificial Viscosity Prescription

Due to the fact that by construction, shock capturing through a viscous process is an artificial one, there is considerable latitude in the way in which such an artificial viscosity may be implemented. This being said, it must obey the following general rules (von Neumann & Richtmyer, 1950; Rosswog, 2009):

- The flow equations should contain no discontinuities;

- The shock front should be of the order of a few times the smoothing length;

- The artificial viscosity should reduce to zero away from the shock front;

- The Rankine-Hugoniot equations should hold over length scales larger than that over which the shock is smoothed, i.e.

$$\rho_0 v_0 \;=\; \rho_1 v_1, \tag{3.75}$$

$$P_0 + \frac{\rho_0 v_0^2}{2} \;=\; P_1 + \frac{\rho_1 v_1^2}{2,} \tag{3.76}$$

$$\frac{P_0}{\rho_0} + u_0 + \frac{v_0^2}{2} \;=\; \frac{P_1}{\rho_1} + u_1 + \frac{v_1^2}{2}, \tag{3.77}$$

where the subscripts 0 and 1 refer to pre- and post-shock regions respectively.

- The overall conservation of momentum and energy should not be adversely affected, while the entropy should rise from the pre- to post-shock regions.

By considering the SPH approximation to the momentum equation 3.61 where the force is based on pairwise addition of terms of the form P/ρ^2, on dimensional grounds it seems sensible to consider an artificial viscosity term Π of the form

$$\Pi \propto \frac{v^2}{\rho} \tag{3.78}$$

for some suitable velocity scale v. von Neumann & Richtmyer (1950) suggested a viscous term dependent on the squared velocity divergence (which gives an indication of the local expansion or contraction of the fluid), which translates into SPH form as

$$(\Pi_{ij})_{\mathrm{NR}} = \frac{\beta_{\mathrm{SPH}} \, h^2 |\nabla \cdot \mathbf{v}_{ij}|^2}{\bar{\rho}_{ij}}, \tag{3.79}$$

where h represents a characteristic length scale (in SPH this is equivalent to the smoothing length), $\bar{\rho}_{ij}$ is the average density of particles i and j and β_{SPH} is a

constant term of order unity. Noting that to first order

$$|\nabla \cdot \mathbf{v}_{ij}| \quad = \quad \frac{|\mathbf{v}_{ij}|}{|\mathbf{r}_{ij}|} \tag{3.80}$$

$$\approx \quad \frac{|\mathbf{v}_{ij} \cdot \mathbf{r}_{ij}|}{|\mathbf{r}_{ij}|^2 + \epsilon h^2} \tag{3.81}$$

where as previously $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ and where we have added the extra (small) term in the denominator to prevent it becoming singular, this von Neumann-Richtmyer term becomes

$$(\Pi_{ij})_{\mathrm{NR}} = \frac{\beta_{\mathrm{SPH}} \, \mu_{ij}^2}{\bar{\rho}_{ij}} \tag{3.82}$$

where

$$\mu_{ij} = \frac{h\mathbf{v}_{ij} \cdot \mathbf{r}_{ij}}{|\mathbf{r}_{ij}|^2 + \epsilon h^2}. \tag{3.83}$$

By considering the bulk and shear viscosities of a generalised fluid it is possible to obtain a second form of the artificial viscosity, and indeed this has been known for some time (Landshoff, 1930; Landau & Lifshitz, 1959). This form is linear in the velocity divergence and uses the average sound speed[4] $\bar{c}_{\mathrm{s},ij}$ as a second, characteristic velocity component, giving the overall form

$$(\Pi_{ij})_{\mathrm{b}} = -\frac{\alpha_{\mathrm{SPH}} \, \bar{c}_{\mathrm{s},ij} \, \mu_{ij}}{\bar{\rho}_{ij}}, \tag{3.84}$$

where μ_{ij} is as before, and α_{SPH} is a second constant of order unity. Note that the negative on the RHS arises from the requirements that the viscous force component must be non-negative (i.e. $\Pi_{ij} > 0$) and that it should be present only for convergent flows, where $\mathbf{v}_{ij} \cdot \mathbf{r}_{ij} < 0$. In fact these criteria also hold for the von Neumann-Richtmyer form of the viscosity, and therefore in both cases the viscosity is set to zero in expanding flow conditions.

These two forms have different and complementary numerical effects. At low Mach numbers ($\mathcal{M} \lesssim 5$) the linear form performs very well in shock tube tests (Monaghan, 1985), whereas for stronger shocks it fails to prevent inter-particle penetration (Lattanzio et al., 1985). This is an unphysical phenomenon in which the two streams pass through each other at the shock front, leading to the possibility of two particles occupying the same position with differing velocities – a multi-valued velocity field. This possibility can be prevented by using the quadratic form of

---

[4]Where as usual, the sound speed is defined as $c_{\mathrm{s}}^2 = \mathrm{d}P/\mathrm{d}\rho$.

von Neumann & Richtmyer as it provides a stronger viscosity for high Mach number, although conversely, on its own this decays too rapidly at low Mach numbers and fails to damp out the unphysical post-shock oscillations or "ringing" that occurs. The standard solution is therefore to use the sum of the two terms (Monaghan, 1989), resulting in a "standard" SPH viscous term of the form

$$\Pi_{ij} = \begin{cases} \dfrac{-\alpha_{\mathrm{SPH}}\, c_{\mathrm{s},ij}\mu_{ij} + \beta_{\mathrm{SPH}}\, \mu_{ij}^2}{\rho_{ij}} & \mathbf{v}_{ij} \cdot \mathbf{r}_{ij} < 0, \\ 0 & \text{otherwise.} \end{cases} \tag{3.85}$$

Various numerical tests have showed that in general the constant values $\alpha_{\mathrm{SPH}} = 1$, $\beta_{\mathrm{SPH}} = 2$ and $\epsilon = 0.01$ in equation 3.83 give good results without significantly affecting non-shocked flows. However, throughout the simulations discussed in the later chapters of this thesis we have used values of $\alpha_{\mathrm{SPH}} = 0.1$ and $\beta_{\mathrm{SPH}} = 0.2$, which have been found to be adequate to accurately resolve (weak) shocks, while at the same time minimising the artificial heating which would have biased our simulation results – details can be found in Lodato & Rice (2004).

This general form of the viscosity can then be incorporated into the momentum equation to give the following form;

$$\frac{\mathrm{d}\mathbf{v}_j}{\mathrm{d}t} = -\sum_i m_i \left(\frac{P_j}{\rho_j^2} + \frac{P_i}{\rho_i^2} + \Pi_{ji} \right) \nabla_j W_{ji}. \tag{3.86}$$

Given that the artificial viscosity term is also pairwise symmetric in i, j (since both $\mathbf{r}_{ij}$ and $\mathbf{v}_{ij}$ are anti-symmetric in i and j) it is clear that this form of the equation of motion also conserves momentum exactly. Likewise it is clear that angular momentum is conserved, and furthermore, by a similar argument to that presented in Section 3.3.3 it is possible to show that in order to preserve energy conservation, the energy equation must be modified to include an extra dissipative term such that

$$\frac{\mathrm{d}u_j}{\mathrm{d}t} = \frac{P_j}{\rho_j^2} \sum_i m_i \mathbf{v}_{ji} \cdot \nabla_i W_{ji} + \sum_i m_i \Pi_{ji} \mathbf{v}_{ji} \cdot \nabla_i W_{ji}. \tag{3.87}$$

In this manner it is therefore possible to include a dissipative term such that shocks can be accurately captured, albeit broadened across a few smoothing lengths. Since mass, momentum and energy are still explicitly conserved across the shock the Rankine-Hugoniot equations are automatically satisfied at distances greater than a few smoothing lengths from the shock. Furthermore, since (in theory at least) the

artificial viscosity is zero away from shocks, all the other initial criteria are satisfied also. However, there are various improvements that can be implemented, and these will now be briefly discussed.

3.4.2 More Advanced Viscosities

The thorn in the side of all viscosity prescriptions is the requirement that in the absence of shocks or other natural dissipative processes the artificial viscosity should reduce to zero, thereby requiring some means to discriminate between shocks and other flow features. Compounding the problem is the fact that careful consideration of the artificial viscosity given above (see for instance Lodato & Price 2010) shows that it provides both a bulk and a shear viscosity, while to resolve shocks only the bulk component is required. Any artificial viscosity in the form of equation 3.85 therefore necessarily introduces an unrequired shear viscosity, which can be problematic in situations where shear flows are important (such as discs), leading to spurious energy and angular momentum transport. Furthermore, since the shear force across any given particle varies with its smoothing length, it is clear that this shear component is resolution dependent. Generally speaking however this effect can be reduced by sensible choices for α_{SPH} and β_{SPH} (Lodato & Rice, 2004) – this further explains the low values of α_{SPH} and β_{SPH} mentioned earlier.

3.4.2.1 The Balsara Switch

An attempt to reduce the induced viscosity in shear flows was presented by Balsara (1995), in which the standard artificial viscosity term Π_{ij} is diminished by the factor $f_{ij} = |f_i + f_j|/2$, where

$$f_i = \frac{|\nabla \cdot \mathbf{v}_i|}{|\nabla \cdot \mathbf{v}_i| + |\nabla \times \mathbf{v}_i| + 0.0001 c_{\mathrm{s},i}/h}. \tag{3.88}$$

The inclusion of the vorticity (the curl of the flow field) allows this form of the viscosity to perform better in shearing and obliquely shocked flows (see for instance Steinmetz 1996), while remaining unaffected in the case of normal shocks. In a similar manner to the "standard" artificial viscosity term, this form also includes a small term $0.0001 c_{\mathrm{s},i}/h$ to prevent the viscosity from becoming singular.

3.4.2.2   The Morris & Monaghan Switch

Although the Balsara switch represents a considerable improvement over the standard form of artificial viscosity, problems still arise in the case of shocks in shearing flows, such as those found in accretion discs. For this reason, Morris & Monaghan (1997) introduced the idea of a time-variant viscosity such that Π_{ij} remains unchanged from the standard form, but where $\alpha_{\mathrm{SPH}} = \alpha(t)$, and where $\beta_{\mathrm{SPH}} = 2\alpha_{\mathrm{SPH}}$. The value of α is then evolved for each particle according to the following equation;

$$\frac{\mathrm{d}\alpha}{\mathrm{d}t} = -\frac{\alpha - \alpha_{\min}}{\tau} + S_{\mathbf{v}}. \tag{3.89}$$

Here $\alpha_{\min} \sim 0.1$ is some minimum value, justified by the requirement that *some* level of artificial viscosity is required to maintain particle order[5], $\tau \sim 0.1 - 0.2\ h/c_{\mathrm{s}}$ is a decay timescale (chosen so that the viscosity decays away over a few smoothing lengths) and $S_{\mathbf{v}} = \max(-\nabla \cdot \mathbf{v}, 0)$ is a source term, activated whenever the flow becomes convergent. Although this form of the source term is still non-zero for pure shear flows, this is counter-balanced to some extent by the decay term, and has been found to work well in many tests of the artificial viscosity (Dolag et al., 2005). Further variations on this theme have been effected, including incorporating the Balsara switch into the Π_{ij} term, and capping the maximum value to which α can rise by using a source term of the form $S_{\mathbf{v}} = \max((\alpha_{\max} - \alpha)\nabla \cdot \mathbf{v}, 0)$ (Rosswog et al., 2000). For a good general overview of the relative merits of a variety of artificial viscosity methods, see for instance Lombardi et al. (1999); Rosswog (2009); Cullen & Dehnen (2010, in prep.).

3.4.3   A Note on Entropy

All of the above methods have essentially been aiming to capture the same phenomenon, namely the increase in entropy found across a shock front, while simultaneously ensuring isentropic flow elsewhere. Furthermore all share the common feature that flow evolution proceeds via integration of the energy equation. However, an approach espoused by Springel & Hernquist (2002) is to consider evolving the entropy directly, thereby ensuring that the entropy can only *increase*.

In this manner, we recall that in terms of density ρ and specific entropy s, the

---

[5]Note that only very low levels of viscosity are required for this purpose – $\alpha_{\min} \sim 0.01$ should suffice (Cullen & Dehnen, 2010, in prep.).

equation of state is given by

$$P_i = \kappa_i(s_i)\rho_i^{\gamma}. \tag{3.90}$$

for some entropic function $\kappa(s)$. Similarly, the internal energy u_i may be obtained from ρ and s via

$$u_i = \frac{\kappa_i(s_i)}{\gamma - 1}\rho_i^{\gamma-1}. \tag{3.91}$$

In the case of isentropic flow, we have $\kappa(s) = const$, and thus by definition

$$\frac{\mathrm{d}\kappa_i}{\mathrm{d}t} = 0. \tag{3.92}$$

However, in the case where artificial viscosity is included, the time derivative of the entropic function becomes

$$\frac{\mathrm{d}\kappa_i}{\mathrm{d}t} = \frac{1}{2}\frac{\gamma - 1}{\rho_i^{\gamma-1}}\sum_j m_j \Pi_{ij}\mathbf{v}_{ij} \cdot \nabla_i W_{ij}. \tag{3.93}$$

By noting that

$$\nabla_i W_{ij} = |\nabla_i W_{ij}|\hat{\mathbf{r}}_{ij}, \tag{3.94}$$

and also that Π_{ij} is only non-zero for $\mathbf{v}_{ij} \cdot \mathbf{r}_{ij} < 0$, it is clear that the term on the RHS of equation 3.93 is strictly non-negative, and thus that entropy can only increase throughout the flow. Using this method of evolving the flow properties it is therefore possible to explicitly ensure that the entropy of any particle increases monotonically with time.

3.5   Variable Smoothing Lengths

Up to now, it has been assumed that the smoothing length h is held constant with time, and is moreover equal for all particles. In regions where the density (and thus the number of neighbours) is roughly constant, this maintains a constant (small) sampling error within the SPH smoothing kernel. This requirement of constant smoothing length is quite restrictive however, as it prevents the code adapting effectively to regions of higher or lower than average density (Steinmetz & Mueller, 1993). By allowing the smoothing length to vary both temporally and spatially, sampling errors can be minimised across regions of varying density, as either the number of neighbours or the mass within a smoothing kernel (and thus the resolution) may

be maintained. There are various simple ways of allowing variable effective smoothing lengths that have been introduced, for instance Benz (1990) suggested using a symmetrised smoothing length $h_{ij} = (h_i + h_j)/2$, such that the kernel becomes

$$W_{ij} = W\left(\mathbf{r}_{ij}, \frac{h_i + h_j}{2}\right). \tag{3.95}$$

An alternative method has been suggested by Hernquist & Katz (1989), in which the average kernel value is used rather than the average smoothing length, such that

$$W_{ij} = \frac{W(\mathbf{r}_{ij}, h_i) + W(\mathbf{r}_{ij}, h_j)}{2}. \tag{3.96}$$

With variable smoothing lengths it then becomes necessary to determine the value of h for each particle. A standard method of doing this is to link the smoothing length to the local density, such that

$$\rho_i h_i^3 = const. \tag{3.97}$$

Since this constant clearly has units of mass, it is frequently linked to the particle mass, giving the following prediction for the particle smoothing length;

$$h_i = \eta \left(\frac{m_i}{\rho_i}\right)^{1/3}, \tag{3.98}$$

where the coupling constant is generally in the range $1.2 < \eta < 1.5$ (Rosswog, 2009). By construction this method maintains a constant mass within the smoothing kernel. As each of the above formalisms remains pairwise symmetric, momentum remains fully conserved, and increased spatial resolution is achieved at relatively low computational cost. The latter method (using the averaged kernel value as in equation 3.96) has additional advantages in it is less problematic across shocks, and it couples better with tree methods for calculating self-gravity (Steinmetz & Mueller, 1993). Nonetheless, in both cases errors appear in either the entropy or energy equation, such that either

$$\frac{\mathrm{d}E}{\mathrm{d}t} \text{ or } \frac{\mathrm{d}\kappa(s)}{\mathrm{d}t} \sim \frac{\partial W}{\partial h}\frac{\partial h}{\partial t} \neq 0, \tag{3.99}$$

(Hernquist, 1993), and the relevant quantity is therefore not explicitly conserved.

It is however possible to construct SPH estimates that self-consistently account

for the variation in smoothing length, and therefore ensure exact energy conservation. In this case, the estimator for density equation 3.39 becomes

$$\rho_j = \sum_i m_i W(\mathbf{r}_{ji}, h_j),\tag{3.100}$$

noting that the smoothing length used in the kernel is that associated with particle j only, and thereby remains constant throughout the summation. By taking the (Lagrangian) time derivative, we obtain

$$\frac{\mathrm{d}\rho_j}{\mathrm{d}t} = \sum_i m_i \left(\mathbf{v}_{ji} \cdot \nabla_i W_{ji}(h_j) + \frac{\partial W_{ji}}{\partial h_j} \frac{\mathrm{d}h_j}{\mathrm{d}t} \right),\tag{3.101}$$

noting the extra terms compared to equation 3.41, and where now we set $W_{ji} = W(\mathbf{r}_{ji}, h_j)$. Noting that

$$\frac{\mathrm{d}h_j}{\mathrm{d}t} = \frac{\mathrm{d}h_j}{\mathrm{d}\rho_j} \frac{\mathrm{d}\rho_j}{\mathrm{d}t},\tag{3.102}$$

and using equation 3.98 we see that

$$\frac{\mathrm{d}h_j}{\mathrm{d}\rho_j} = -\frac{h_j}{3\rho_j}.\tag{3.103}$$

Substituting this into equation 3.101 and gathering like terms, we find that the time derivative of the density becomes

$$\frac{\mathrm{d}\rho_j}{\mathrm{d}t} = \frac{1}{\Omega_j} \sum_i m_i \mathbf{v}_{ji} \cdot \nabla_i W_{ij}(h_j)\tag{3.104}$$

where

$$\Omega_j \;=\; 1 - \frac{\mathrm{d}h_j}{\mathrm{d}\rho_j} \sum_i m_i \frac{\partial W_{ji}(h_j)}{\partial h_j},\tag{3.105}$$

$$\;=\; 1 + \frac{h_j}{3\rho_j} \sum_i m_i \frac{\partial W_{ji}(h_j)}{\partial h_j},\tag{3.106}$$

and where $\partial W_{ji}/\partial h_j$ is known from the choice of kernel. Although it can be calculated directly from the kernel, in the case of the cubic spline kernel given in equation 3.31 it is generally evaluated by noting that

$$\frac{\partial W}{\partial h_j} = -x\nabla W - \frac{3}{h}W,\tag{3.107}$$

where W and ∇W are given by equations 3.31 and 3.33 respectively.

Similarly, there is a correction factor to the momentum equation to allow for the spatial variation in smoothing lengths. Recall from equation 3.52 that in order to calculate the spatial variation of the Lagrangian, we need to know the spatial derivative of the density. Allowing now for variable smoothing lengths and using equation 3.57, we therefore find that

$$\frac{\partial \rho_j}{\partial \mathbf{r}_i} = \sum_k m_k \left(\nabla_j W_{ji}(h_j) \left[\delta_{ji} - \delta_{jk} \right] + \frac{\partial W_{jk}(h_j)}{\partial h_j} \frac{\mathrm{d}h_j}{\mathrm{d}\rho_j} \frac{\partial \rho_j}{\partial \mathbf{r}_i} \right). \tag{3.108}$$

By gathering like terms, we find that the correction factor for the spatial derivative of the density is same as that for the temporal one, namely that

$$\frac{\partial \rho_j}{\partial \mathbf{r}_i} = \frac{1}{\Omega_j} \sum_k m_k \nabla_i W_{jk}(h_j) \left[\delta_{ji} - \delta_{jk} \right], \tag{3.109}$$

with the factor Ω_j defined as before in equation 3.106.

Following the same derivation as in Section 3.3.2, it is then easy to show that the acceleration due to hydrodynamic forces with spatially varying smoothing lengths is given by

$$\frac{\mathrm{d}\mathbf{v}_j}{\mathrm{d}t} = -\sum_i m_i \left(\frac{P_j}{\Omega_j \rho_j^2} \nabla_i W_{ji}(h_j) + \frac{P_i}{\Omega_i \rho_i^2} \nabla_j W_{ji}(h_i) \right). \tag{3.110}$$

Finally, from equation 3.73, we see that the evolution of the internal energy in the presence of variable smoothing lengths becomes

$$\frac{\mathrm{d}u_j}{\mathrm{d}t} = \frac{P_j}{\Omega_j \rho_j^2} \sum_i m_i \mathbf{v}_{ji} \cdot \nabla W_{ji}(h_j). \tag{3.111}$$

By an analogous process to that described in Section 3.3.3, it is possible to show that this equation for the evolution of the internal energy is also explicitly conservative of the total energy of the system, E. The three equations 3.100, 3.110 and 3.111 along with the relationship between the density and the smoothing length equation 3.98 therefore form a fully consistent, fully conservative SPH formalism with spatially varying smoothing lengths.

A problem exists however, in that in order to obtain the density, one needs to know the smoothing length (equation 3.100) and to obtain the smoothing length one needs to know the density (equation 3.98). In order to resolve this, this pair of equa-

tions can be solved iteratively by the Newton-Raphson method (Price & Monaghan, 2007). By rewriting equation 3.98, we can combine these two equations to reduce the problem to that of finding the root h_j of the equation $\zeta(h_j) = 0$, where

$$\zeta(h_j) = m_j \left(\frac{\eta}{h_j} \right)^3 - \sum_i m_i W(\mathbf{r}_{ji}, h_j). \tag{3.112}$$

Here the first term represents the density obtained from assuming a fixed mass within the smoothing kernel, while the second term is the standard SPH estimate for the density. From some initial estimate of the root h_j, the Newton-Raphson method gives a better estimate as being

$$h_{j,\text{new}} = h_j - \frac{\zeta(h_j)}{\zeta'(h_j)}, \tag{3.113}$$

where the prime denotes differentiation with respect to h. By using equation 3.106 we see that

$$\zeta'(h_j) = -\frac{3\rho_j \Omega_j}{h_j}, \tag{3.114}$$

and thus the updated value $h_{j,\text{new}}$ is given by

$$h_{j,\text{new}} = h_j \left(1 + \frac{\zeta(h_j)}{3\rho_j \Omega_j} \right). \tag{3.115}$$

This may be repeated until $|h_{j,\text{new}} - h_j|/h_j < \epsilon$ for some small value of ϵ, frequently set to 10^{-3}. Then in turn, a self consistent value of the density is then obtained from equation 3.98. As there is generally relatively little change in h_j and ρ_j between timesteps, the estimator for h_j is taken as the value from the previous timestep, and convergence usually occurs within a small number of iterations (Price & Monaghan, 2007). In pathological cases where the Newton-Raphson method does not converge, other, universally convergent but slower methods such as the bisection method may be used instead.

Although the inclusion of variable smoothing lengths through this method does inevitably increase the computational cost of the code, this is relatively small, and the conservation properties are recovered to within machine (and integrator) tolerance. Other tricks, such as predicting the change in the smoothing length using equation 3.102 can reduce the computational cost still further (see for instance, Price & Monaghan 2007).

3.6   Including Gravity

As many astrophysical situations are driven at some level by gravitational forces, it is important to be able to include this consistently within the SPH framework, and in such a manner that the inherent conservation properties of the algorithm are not compromised. While much work has been put into N-body simulations of discrete particles, within the SPH formalism we are aiming to model the gravitational force over a continuum, and thus it should be smoothed (or *softened* in SPH parlance) in a similar manner to that in which the discrete particle mass is smoothed into the density of a fluid continuum. In this section we therefore consider how this can be done in a consistent manner, and one in which as before momentum and energy are explicitly conserved.

3.6.1   Gravity in the Lagrangian

In an extension to the Lagrangian for the hydrodynamic equations of motion, it is possible to incorporate the effects of gravity by considering a Lagrangian of the form

$$\mathcal{L}(\mathbf{r}, \mathbf{v}) = \sum_i m_i \left(\frac{1}{2} \mathbf{v}_i \cdot \mathbf{v}_i - u_i(\mathbf{r}_i) \right) - \Psi, \tag{3.116}$$

where Ψ is an as yet undefined measure of the total gravitational potential energy of the system. By comparison with equation 3.46 this is clearly just the hydrodynamic Lagrangian with an additional term

$$\mathcal{L}_{\mathrm{grav}} = -\Psi \tag{3.117}$$

which describes the effects of gravity.

Now, as with the density, at position $\mathbf{r}_i$ we can obtain the local gravitational potential Φ_i, via a sum over all particles such that

$$\Phi_i = G \sum_j m_j \phi(\mathbf{r}_i - \mathbf{r}_j, \varepsilon_i), \tag{3.118}$$

where $\phi(\mathbf{r}_i - \mathbf{r}_j) = \phi(\mathbf{r}_{ij})$ is known as the (gravitational) softening kernel, G is the universal gravitational constant and where ε_i is the softening length associated with particle i. The softening kernel at this stage is fairly general, but it must have the following properties:

- $\phi(r, h) < 0$ for all r, h, as the local potential Φ must be strictly negative definite;

- $\nabla\phi(0, h) = 0$, such that the gravitational force exerted by any particle on itself is zero;

- $\lim\limits_{r/h \to \infty} \phi(r, h) = -\dfrac{1}{r}$, i.e. the softening should reduce to zero at large inter-particle distances, and the Newtonian potential should be recovered.

Generally speaking, and throughout this thesis, it is assumed that the softening length is exactly equal to the smoothing length for all particles, i.e. $\varepsilon_i = h_i$ for all i. In a similar manner it is generally taken that the force should only be softened when $r < 2h$, so that force softening and density smoothing occur over exactly the same region.

Noting that the gravitational potential energy is just the mass times the gravitational potential, since the latter is defined over pairs of particles, by definition the *total* gravitational potential energy of the system is given by the sum over all *pairs* of particles, such that

$$\Psi \;=\; G \sum_i m_i \sum_{j \leq i} m_j \phi_{ij}(h_i) \tag{3.119}$$

$$\;=\; \frac{G}{2} \sum_i \sum_j m_i m_j \phi_{ij}(h_i) \tag{3.120}$$

Note that equation 3.119 sums over all pairs of particles, including the so-called self-interaction terms where $i = j$, and thus explains the factor of a half in equation 3.120. From this definition we therefore find that

$$\Psi = \frac{1}{2} \sum_i m_i \Phi_i, \tag{3.121}$$

and thus that the full Lagrangian in the presence of gravity becomes

$$\mathcal{L}(\mathbf{r}, \mathbf{v}) = \sum_i m_i \left(\frac{1}{2} \mathbf{v}_i \cdot \mathbf{v}_i - u_i(\mathbf{r}_i) - \frac{1}{2} \Phi_i \right). \tag{3.122}$$

By considering only the gravitational term in the Lagrangian, we can as before use the Euler-Lagrange equations 3.47 to obtain the acceleration due to gravity,

which becomes

$$m_j \frac{\mathrm{d}v_j}{\mathrm{d}t} = \frac{\partial \mathcal{L}_{\text{grav}}}{\partial \mathbf{r}_j}. \tag{3.123}$$

Using equations 3.117 and 3.120 we therefore find that the spatial derivative of the gravitational Lagrangian becomes

$$\frac{\partial \mathcal{L}_{\text{grav}}}{\partial \mathbf{r}_j} = -\frac{G}{2} \sum_i \sum_k m_i m_k \frac{\partial \phi_{ik}(h_i)}{\partial \mathbf{r}_j} \tag{3.124}$$

$$= -\frac{G}{2} \sum_i \sum_k m_i m_k \left(\nabla_j \phi_{ik}(h_i) + \frac{\partial \phi_{ik}(h_i)}{\partial h_i} \frac{\partial h_i}{\partial \mathbf{r}_j} \right). \tag{3.125}$$

Here we see that in the case of fixed smoothing (and therefore softening) lengths, $\partial h_i / \partial \mathbf{r}_k = 0$, and thus we only require the first term to determine the effects of gravity. The second term is therefore a correction term to allow for spatial variation in h.

As before, using the method of equations 3.54 to 3.57 the spatial gradient becomes

$$\left. \frac{\partial \mathcal{L}_{\text{grav}}}{\partial \mathbf{r}_j} \right|_h = -\frac{G}{2} \sum_i \sum_k m_i m_k \, \nabla_j \phi_{ik}(h_i) \left[\delta_{ij} - \delta_{kj} \right], \tag{3.126}$$

$$= -\frac{G}{2} m_j \sum_k m_k \nabla_j \phi_{jk}(h_j) + \frac{G}{2} \sum_i m_i m_j \nabla_j \phi_{ij}(h_i). \tag{3.127}$$

Now by changing the summation index of the first term to i, and noting again that the kernel is antisymmetric we obtain

$$\left. \frac{\partial \mathcal{L}_{\text{grav}}}{\partial \mathbf{r}_j} \right|_h = -\frac{G}{2} m_j \sum_i m_i \left(\nabla_j \phi_{ji}(h_j) + \nabla_j \phi_{ji}(h_i) \right), \tag{3.128}$$

which therefore encapsulates the effects of gravity in the case of constant smoothing lengths.

If we now consider the second term in equation 3.125 and self-consistently correct for spatial variation in the smoothing length, we find that

$$\left. \frac{\partial \mathcal{L}_{\text{grav}}}{\partial \mathbf{r}_j} \right|_{\text{corr}} = -\frac{G}{2} \sum_i \sum_k m_i m_k \frac{\partial \phi_{ik}(h_i)}{\partial h_i} \frac{\mathrm{d}h_i}{\mathrm{d}\rho_i} \frac{\partial \rho_i}{\partial \mathbf{r}_j}. \tag{3.129}$$

By substituting equation 3.109 into the above we find that

$$
\left.\frac{\partial \mathcal{L}_{\text{grav}}}{\partial \mathbf{r}_j}\right|_{\text{corr}} = -\frac{G}{2}\sum_i\sum_k m_i m_k \frac{\partial \phi_{ik}(h_i)}{\partial h_i}\frac{\mathrm{d}h_i}{\mathrm{d}\rho_i}\frac{1}{\Omega_i}\sum_l m_l \nabla_j W_{il}(h_l)[\delta_{ij}-\delta_{lj}] \quad (3.130)
$$

$$
\begin{aligned}
&= -\frac{G}{2}m_j\sum_k m_k \frac{\partial \phi_{jk}}{\partial h_j}\frac{\mathrm{d}h_j}{\mathrm{d}\rho_j}\frac{1}{\Omega_j}\sum_l m_l \nabla_j W_{jl}(h_l) \\
&\quad + \frac{G}{2}\sum_i\sum_k \frac{\partial \phi_{ik}(h_i)}{\partial h_i}\frac{1}{\Omega_i}m_j \nabla_j W_{ij}(h_i).
\end{aligned}
\qquad (3.131)
$$

Now by changing the summation index of the second sum in the first term of equation 3.131 to i, defining a new quantity ξ_p such that

$$
\xi_p = \frac{\mathrm{d}h_p}{\mathrm{d}\rho_p}\sum_q m_q \frac{\partial \phi_{pq}(h_p)}{\partial h_p}, \qquad (3.132)
$$

and using the antisymmetry property of the gradient of the smoothing kernel, we see that the correction term reduces to

$$
\left.\frac{\partial \mathcal{L}_{\text{grav}}}{\partial \mathbf{r}_j}\right|_{\text{corr}} = -\frac{G}{2}m_j\sum_i m_i \left(\frac{\xi_j}{\Omega_j}\nabla_j W_{ji}(h_j) + \frac{\xi_i}{\Omega_i}\nabla_j W_{ji}(h_i)\right). \qquad (3.133)
$$

Finally, using equation 3.123 and by incorporating the effects of gravity into the equations of motion for a hydrodynamic flow with artificial viscosity (while self-consistently allowing for variable smoothing lengths) we find that the full equations of motion become

$$
\begin{aligned}
\frac{\mathrm{d}\mathbf{v}_j}{\mathrm{d}t} = &-\sum_i m_i \left(\frac{P_j}{\Omega_j \rho_j^2}\nabla_j W_{ji}(h_j) + \frac{P_i}{\Omega_i \rho_i^2}\nabla_j W_{ji}(h_i) + \Pi_{ji}\frac{\nabla_j W_{ji}(h_j) + \nabla_j W_{ji}(h_i)}{2}\right) \\
&-\frac{G}{2}\sum_i m_i \left(\nabla_j \phi_{ji}(h_j) + \nabla_j \phi_{ji}(h_j)\right) \\
&-\frac{G}{2}\sum_i m_i \left(\frac{\xi_j}{\Omega_j}\nabla_j W_{ji}(h_j) + \frac{\xi_i}{\Omega_i}\nabla_j W_{ji}(h_i)\right),
\end{aligned}
\qquad (3.134)
$$

with Ω_i and ξ_i defined as per equations 3.106 and 3.132 respectively[6]. As in the case for the pure hydrodynamic flow, the use of a Lagrangian in deriving these equations ensures the explicit conservation of both linear and angular momentum, which is also clear from the pairwise symmetry present in all terms in the above equation.

---

[6]Note that for consistency, the artificial viscosity term Π_{ji} uses the *average* value of the smoothing lengths $h_{ji} = (h_j + h_i)/2$ in its definition of μ_{ji} (equation 3.83).

3.6.2   Evolution of the Gravitational Potential

Clearly as particles move about within a gravitational potential, their potential energy (given in SPH terms by $m_j \Phi_j$) will also vary. Although the potential (and thus the potential energy) is obtained at any point by the sum over particles using equation 3.118, the time evolution of the potential energy is required to maintain energy conservation. Hence in a similar manner to Section 3.3.3 we must consider the total energy of the system, which including the gravitational potential energy becomes

$$E = \sum_j m_j \left(\frac{1}{2} \mathbf{v}_j \cdot \mathbf{v}_j + u_j + \frac{1}{2} \Phi_j \right). \tag{3.135}$$

As before, to ensure energy conservation we require that the time derivative of the total energy is zero, i.e. that

$$\sum_j m_j \left(\mathbf{v}_j \cdot \frac{\mathrm{d}\mathbf{v}_j}{\mathrm{d}t} + \frac{\mathrm{d}u_j}{\mathrm{d}t} + \frac{1}{2} \frac{\mathrm{d}\Phi_j}{\mathrm{d}t} \right) = 0. \tag{3.136}$$

By considering equation 3.118, we see that

$$\frac{\mathrm{d}\Phi_j}{\mathrm{d}t} = \frac{G}{2} \sum_i m_i \left(\nabla_j \phi_{ji}(h_j) \cdot \frac{\mathrm{d}\mathbf{r}_j}{\mathrm{d}t} + \nabla_i \phi_{ji}(h_j) \cdot \frac{\mathrm{d}\mathbf{r}_i}{\mathrm{d}t} + \frac{\partial \phi_{ji}}{\partial h_j} \frac{\mathrm{d}h_j}{\mathrm{d}\rho_j} \frac{\mathrm{d}\rho_j}{\mathrm{d}t} \right). \tag{3.137}$$

Recalling the definition of ξ_j from equation 3.132, and using equation 3.104 for the definition of $\mathrm{d}\rho_j/\mathrm{d}t$ with variable smoothing lengths, we obtain

$$\frac{\mathrm{d}\Phi_j}{\mathrm{d}t} = \frac{G}{2} \sum_i m_i \mathbf{v}_{ji} \cdot \left(\nabla_j \phi_{ji}(h_j) + \frac{\xi_j}{\Omega_j} \nabla_j W_{ji}(h_j) \right), \tag{3.138}$$

From Sections 3.3.3 and 3.5 we know that in the energy balance (equation 3.135), over all particles the hydrodynamic terms in the equations of motion (equation 3.110) exactly counteract the temporal rate of change of the internal energy,

$$\frac{dE_{\mathrm{hydro}}}{\mathrm{d}t} = \sum_j m_j \left(\mathbf{v_j} \cdot \frac{\mathrm{d}\mathbf{v}_j}{\mathrm{d}t} \bigg|_{\mathrm{hydro}} + \frac{\mathrm{d}u_j}{\mathrm{d}t} \right) = 0, \tag{3.139}$$

and thus pure hydrodynamic flows are exactly conservative of energy. With the inclusion of gravity we therefore only need to show that over the whole system the gravitational terms in the equations of motion balance the time derivative of the

gravitational potential, i.e. that

$$\frac{dE_{\mathrm{grav}}}{\mathrm{d}t} = \sum_j m_j \left(\mathbf{v_j} \cdot \left.\frac{\mathrm{d}\mathbf{v}_j}{\mathrm{d}t}\right|_{\mathrm{grav}} + \frac{1}{2}\frac{\mathrm{d}\Phi_j}{\mathrm{d}t} \right) = 0 \tag{3.140}$$

in order to maintain exact conservation of energy in self-gravitating systems.

From equations 3.134 and 3.138 this gravitational energy balance becomes

$$\begin{aligned}
\frac{dE_{\mathrm{grav}}}{\mathrm{d}t} &= -\frac{G}{2}\sum_j\sum_i m_j m_i \mathbf{v}_j \cdot \Big(\nabla_j \phi_{ji}(h_j) + \nabla_j \phi_{ji}(h_i) + \\
&\qquad\qquad\qquad\frac{\xi_j}{\Omega_j}\nabla_j W_{ji}(h_j) + \frac{\xi_i}{\Omega_i}\nabla_j W_{ji}(h_i) \Big) \\
&\quad + \frac{G}{2}\sum_j\sum_i m_j m_i\, \mathbf{v}_{ji} \cdot \Big(\nabla_j \phi_{ji}(h_i) + \frac{\xi_j}{\Omega_j}\nabla_j W_{ji}(h_j) \Big)
\end{aligned} \tag{3.141}$$

Cancelling like terms, this reduces to

$$\begin{aligned}
\frac{dE_{\mathrm{grav}}}{\mathrm{d}t} &= -\frac{G}{2}\sum_j\sum_i m_j m_i \Big(\mathbf{v}_j \cdot \nabla_j \phi_{ji}(h_j) + \mathbf{v}_i \cdot \nabla_j \phi_{ji}(h_i) \\
&\qquad\qquad + \frac{\xi_i}{\Omega_i}\mathbf{v}_j \cdot \nabla_j W_{ji}(h_i) + \frac{\xi_j}{\Omega_j}\nabla_j W_{ji}(h_j) \Big),
\end{aligned} \tag{3.142}$$

and finally, noticing that the gradients of both the smoothing and the softening kernels are antisymmetric under a reversal of the summation indices i and j, we obtain the desired result that

$$\frac{dE_{\mathrm{grav}}}{\mathrm{d}t} = 0. \tag{3.143}$$

Therefore, we see that gravity can be included into SPH in such a manner that the algorithm remains explicitly conservative of energy.

3.6.3   Gravitational Potentials and the Softening Kernel

Finally for this section, we need to consider the form of the gravitational softening kernel ϕ, and its relation to the smoothing kernel W. Recall that Poisson's equation links the gravitational potential $\Phi(\mathbf{r})$ to the density $\rho(\mathbf{r})$ at position $\mathbf{r}$, such that

$$\nabla^2 \Phi(\mathbf{r}) = 4\pi G\rho(\mathbf{r}). \tag{3.144}$$

Given that we implicitly assume each particle to be spherically symmetric, by using spherical polar co-ordinates and substituting equations 3.39 and 3.118 into equation 3.144 we find that (for a generalised radial co-ordinate r)

$$W(r, h) = \frac{1}{4\pi r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial \phi(r, h)}{\partial r} \right), \tag{3.145}$$

where we have neglected the spatial variation of h[7].

We can now integrate this, to link the derivative of the softening kernel $\partial\phi/\partial r$ (also known as the force kernel) to the smoothing kernel, such that

$$\frac{\partial \phi}{\partial r} = \frac{4\pi}{r^2} \int^r r'^2 W(r') \mathrm{d}r' + \frac{C_1}{r^2}, \tag{3.146}$$

with the integration constant C_1 subject to the condition that for $r \geq 2h$ we recover the standard Newtonian inverse square law, which using our definitions becomes $\partial\phi/\partial r = 1/r^2$. In a similar manner we can integrate this a step further (by parts), to give the full softening kernel, such that

$$\phi = 4\pi \left[-\frac{1}{r} \int^r r'^2 W(r') \mathrm{d}r' + \int^r r' W(r') \mathrm{d}r' \right] + \frac{C_1}{r^2} + \frac{C_2}{r}, \tag{3.147}$$

where the second integration constant allows the correct asymptotic behaviour (i.e. $\phi \to 0$ as $r \to \infty$) to be established.

With this in mind, for the cubic spline kernel defined in equation 3.31 the force kernel $\partial\phi/\partial r$ becomes

$$\frac{\partial \phi(r, h)}{\partial r} = \begin{cases} \frac{1}{h^2} \left(\frac{4}{3}x - \frac{6}{5}x^3 + \frac{1}{2}x^4 \right) & 0 \leq x \leq 1, \\ \frac{1}{h^2} \left(\frac{8}{3}x - 3x^2 + \frac{6}{5}x^3 - \frac{1}{6}x^4 - \frac{1}{15x^2} \right) & 1 \leq x \leq 2, \\ \frac{1}{r^2} & x \geq 2, \end{cases} \tag{3.148}$$

where $x = r/h$ and where the integration constants have been absorbed to ensure piecewise continuity. Finally, we therefore find the full softening kernel consistent

---

[7]This is because the smoothing length essentially acts as a normalising constant in both the smoothing and the softening kernels, and for any given particle is held constant within Poisson's equation. Thus its spatial variation is immaterial here.

with the cubic spline smoothing kernel to be

$$
\phi(r,h) = \begin{cases}
\dfrac{1}{h}\left(\dfrac{2}{3}x^2 - \dfrac{3}{10}x^3 + \dfrac{1}{10}x^5 - \dfrac{7}{5}\right) & 0 \leq x \leq 1, \\[2ex]
\dfrac{1}{h}\left(\dfrac{4}{3}x^2 - x^3 + \dfrac{3}{10}x^4 - \dfrac{1}{30}x^5 - \dfrac{8}{5} + \dfrac{1}{15x}\right) & 1 \leq x \leq 2, \\[2ex]
-\dfrac{1}{r} & x \geq 2.
\end{cases}
\tag{3.149}
$$

Using this definition of the softening kernel along with the cubic spline smoothing kernel equation 3.31, the equations of motion equation 3.134 and equation 3.138 for the evolution of the gravitational potential therefore allows gravity to be included in a manner that it is fully conservative, and is such that Poisson's equation is satisfied throughout.

3.7   Finding the Nearest Neighbours

Various methods exist for finding the nearest neighbours (i.e. those particles within the smoothing kernel of any given particle), with the simplest being a direct search over all particles. This becomes very expensive in the limit of large numbers of particles N however, as the computational cost scales as $\mathcal{O}(N^2)$. Other methods such as using an overlaid grid or a linked list of particle positions have been used (Hockney & Eastwood, 1981; Monaghan, 1985; Murray, 1996; Deegan, 2009). One of the more efficient methods however is to use a hierarchical tree structure, an approach that grew out the requirements of N-body codes to distinguish distant particles (where the gravitational forces could be evaluated via multipole expansions) from local particles (where direct N-body calculation of the forces was still required). These in general reduce the cost of neighbour-finding from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log N)$ (Barnes & Hut, 1986; Hernquist, 1987; Hernquist & Katz, 1989), although reductions to $\mathcal{O}(N)$ have been achieved (Dehnen, 2002, 2000).

Trees are essentially data structures which decompose the computational domain into a series of discrete volumes, the sum over which contains all the particles. The smallest volumes generally contain only a single particle, but this is not strictly necessary, and indeed may inhibit the efficiency of the code (Dehnen, 2009, private communication). By construction, particles which are near each other in space are near each other within the tree structure, and thus by looping over a relatively small part of the tree, the nearest neighbours may be found efficiently. There are various

different algorithms that perform this decomposition described in the literature, none of which are trivial, so I shall not attempt to go into any depth here. For further details, see for instance Press et al. (2007); Dehnen (2002); Steinmetz & Mueller (1993); Barnes & Hut (1986); Bentley (1975) and references therein.

A distinct advantage of using trees for neighbour finding within an SPH code is that they couple readily with pre-existing methods for evaluating the gravitational force between large numbers of particles. Rather than a direct summation (which is of $\mathcal{O}(N^2)$) over all particles to find the gravitational force at a specific location, particles at large distances can be effectively treated as a single body, and multipole expansions used to approximate the force. This approach has found success in various N-body codes as a means of reducing the computation time to $\mathcal{O}(N \log N)$ or lower (Barnes & Hut, 1986; Hernquist, 1987; Hernquist & Katz, 1989). Use of a tree algorithm therefore allows the process of neighbour finding to be coupled to that of finding the gravitational forces acting on a particle, with an attendant saving of computational expense.

3.8 Integration and Timestepping

So far we have obtained equations to evolve the density, the three components of velocity under the influence of pressure, gravitational and (artificial) viscous forces, the internal energy and the gravitational potential. Finally therefore it is time to consider *how* these equations are actually evolved, and to discuss the issues of temporal integration and time-stepping.

Generally speaking there are two principal methods used to perform the time evolution, and indeed the code I have used throughout includes the option to use either. They are the so-called Leapfrog integrator (also known as the kick-drift or Störmer-Verlet integrator) and the Runge-Kutta-Fehlberg method, and I shall now briefly consider both of these.

3.8.1 The Leapfrog Integrator

The leapfrog integrator is a second-order integrator, so-called because the position and the velocity are advanced half a timestep out of phase, with each update of position or velocity using the value of the velocity or position evaluated at the previous half timestep. In this manner, the positions and velocities "leap-frog" over

each other at every half timestep, giving rise to the name. The leapfrog method is widely used in N-body codes, since in the case where the acceleration is independent of the velocity, i.e. where $\mathbf{a} = \mathbf{a}(\mathbf{r})$ only, it is particularly simple to implement. In its "pure" form it is a time-reversible, symplectic integrator, which by definition is explicitly conservative of both energy and angular momentum (see for instance Springel (2005) and references therein).

In essence then, if the position, velocity and acceleration at time t_i are given by $\mathbf{r}_i$, $\mathbf{v}_i$ and $\mathbf{a}_i$ respectively, with a timestep δt the standard form of the leapfrog integrator gives the positions and velocities as

$$\begin{aligned} \mathbf{r}_{i+1} &= \mathbf{r}_i + \mathbf{v}_{i-1/2}\,\delta t, \\ \mathbf{v}_{i+1/2} &= \mathbf{v}_{i-1/2} + \mathbf{a}_i\,\delta t. \end{aligned} \tag{3.150}$$

Here it is clear that the positions and velocities are evaluated at half timesteps with respect to each other, and "leap-frog" over each other as they are evolved. In this form it is also clear that the integrator should be perfectly time-reversible.

A form that is often more readily applied is the equivalent definition at integer timesteps, which becomes

$$\begin{aligned} \mathbf{r}_{i+1} &= \mathbf{r}_i + \delta t \left(\mathbf{v}_i + \frac{\delta t}{2}\,\mathbf{a}_i \right), \\ \mathbf{v}_{i+1} &= \mathbf{v}_i + \frac{\delta t}{2} \left(\mathbf{a}_i + \mathbf{a}_{i+1} \right). \end{aligned} \tag{3.151}$$

Although it is now less obvious, these equations are still fully time reversible. Note further that the form of the increments on the RHS of each of the above equations is equivalent to an estimate of the relevant quantity at the following half timestep, noting in particular that

$$\mathbf{v}_i + \frac{\delta t}{2}\,\mathbf{a}_i = \mathbf{v}_{i+1/2}, \tag{3.152}$$

to first order.

Note however, that in both cases problems arise if the acceleration depends on the velocity, since from equation 3.151 we see that to calculate $\mathbf{v}_{i+1}$ we already need to know the acceleration $\mathbf{a}_{i+1}$, and the scheme becomes implicit. Since in SPH simulations both the pressure force and the artificial viscous force depend on the local velocity, it is clear that modifications are required before this integrator may be used. The standard way this correction is implemented (see for instance Springel et al. 2001; Wetzstein et al. 2009) is as follows:

- Firstly, predict the positions at time $t_{i+1/2}$ in a manner analogous to equation 3.152 via

$$\mathbf{r}_{i+1/2} = \mathbf{r}_i + \frac{\delta t}{2}\,\mathbf{v}_i. \tag{3.153}$$

- Secondly, use equation 3.152 to obtain the velocity at time $t_{i+1/2}$, and extrapolate other values (such as density, internal energy and gravitational potential) at the half timestep also. Hence calculate the acceleration at the half timestep, $\mathbf{a}_{i+1/2}$.

- Calculate the velocity at time t_{i+1} using

$$\mathbf{v}_{i+1} = \mathbf{v}_i + \delta t\,\mathbf{a}_{i+1/2}. \tag{3.154}$$

- Now update the positions to timestep t_{i+1} using

$$\mathbf{r}_{i+1} = \mathbf{r}_i + \frac{\delta t}{2}\left(\mathbf{v}_i + \mathbf{v}_{i+1}\right). \tag{3.155}$$

The process is now repeated as required.

Although the strict symmetry between the integration of positions and velocities has been lost by the inclusion of these predictor steps, this method still remains time-reversible. Furthermore, it is also now possible to include adaptive timestepping, which would have been problematic before precisely *because* of the symmetry between the integrations (Wetzstein et al., 2009). Generally speaking however, maintaining time-reversibility with adaptive timestepping is difficult (Springel et al., 2001; Quinn et al., 1997), though not impossible.

3.8.2   The Runge-Kutta-Fehlberg Integrator

Runge-Kutta methods for integrating systems of differential equations are well known, tried and trusted methods, which use multiple estimates of the derivative across a given timestep to arrive at accurate, generally high order estimates for the evolved quantities. Most common is the fourth order Runge-Kutta method, often simply abbreviated to RK4, which has been known and used for over a century (Kutta, 1901). Moreover, Fehlberg (1968, 1969) obtained a modified Runge-Kutta integrator (now known as a Runge-Kutta-Fehlberg integrator) which embedded a order $n+1$ method within an order n method. This allows the two methods to be compared to give an error estimate, and thus for the error to be controlled to some

given tolerance. The most common of these methods embeds a fifth order estimate within a fourth order scheme, and is therefore known as an RK45 integrator.

However, compared to the leapfrog integrator, which requires only one evaluation of the acceleration per timestep, the RK4 scheme requires four, and the RK45 method requires six. Therefore these methods, although correct to much higher order than the leapfrog, add significantly to the computation time required. (Note also that they are not necessarily more *accurate* either, as they are not explicitly conservative in the way that the leapfrog method is. See for instance Springel (2005); Wetzstein et al. (2009); Rosswog (2009) for a comparison of these integrators as applied to a simple Keplerian orbit evolved over many dynamical times.) The solution is to go to a lower order RKF method, where the implicit error control is still present but the number of derivative evaluations is reduced. A common choice for many SPH codes including VINE (Wetzstein et al., 2009; Nelson et al., 2009) and the one used in the code I have used, is the RK12 integrator developed by Fehlberg (1969) which proceeds as follows.

For a given variable $\mathbf{z}$, the evolution from $\mathbf{z}_i$ at time t_i to $\mathbf{z}_{i+1}$ at time $t_{i+1} = t_i + \delta t$ is given by

$$\mathbf{z}_{i+1} = \mathbf{z}_i + \left(\frac{1}{256} k_0 + \frac{255}{256} k_1 \right) \delta t, \tag{3.156}$$

where the values of k_0 and k_1 are provided by evaluating $\dot{\mathbf{z}}$ at various points, such that

$$
\begin{aligned}
k_0 &= \dot{\mathbf{z}}(t_i, \mathbf{z}_i), \\
k_1 &= \dot{\mathbf{z}}(t_i + \frac{\delta t}{2}, \mathbf{z}_i + \frac{\delta t}{2} k_0)
\end{aligned}
\tag{3.157}
$$

and where the dot as usual denotes differentiation with respect to time. Expansion via Taylor series shows that this is accurate to first order, with the choice of coefficients in equation 3.156 producing a leading order truncation error τ_{trunc} such that

$$\tau_{\text{trunc}} = -\frac{1}{512} \delta t^2 \ddot{\mathbf{z}}. \tag{3.158}$$

Using the values for k_0 and k_1 defined above, we can compute a further estimate $\mathbf{z}_{i+1}^*$ for $\mathbf{z}$ at time t_{i+1}, such that

$$\mathbf{z}_{i+1}^* = \mathbf{z}_i + \frac{\delta t}{2} \left(\frac{1}{512} k_0 + \frac{255}{256} k_1 + \frac{1}{512} k_2 \right), \tag{3.159}$$

with the additional value k_2 defined such that

$$
\begin{aligned}
k_2 &= \dot{\mathbf{z}}(t_i + \delta t, \, \mathbf{z}_i + \left(\frac{1}{256}k_0 + \frac{255}{256}k_1 \right) \delta t), \\
&= \dot{\mathbf{z}}(t_{i+1}, \mathbf{z}_{i+1}).
\end{aligned}
\tag{3.160}
$$

Again, by considering Taylor series expansions, this value $\mathbf{z}_{i+1}^*$ can be shown to be a second order estimate. One of the more appealing tricks of this method is that here k_2 is simply k_0 evaluated for the *next* timestep, and thus per timestep, only two derivative evaluations are required.

We now therefore have both a first and a second order estimate for the value of $\mathbf{z}$ at time t_{i+1}, with a known truncation error for the first order method. This can therefore be used for error control, to ensure that the timestep used is appropriate (see for instance, Press et al. 2007). However, in order for this error control to be valid, the *first* order scheme must be used for the evolution. To mitigate this, by construction this first order scheme has very small second order errors (equation 3.158), and so is effectively a quasi-second order integrator.

3.8.3   Timestepping Criteria

For either integrator, it is crucial that the timestep size is chosen correctly, both to ensure the accuracy of the evolution and to ensure numerical stability. In this section I shall briefly discuss the principal timestepping criteria in general use, and one specific to the code I have used.

3.8.3.1   CFL Criterion

By far the most general timestep criterion for gas-dynamical systems is the so-called *Courant-Friedrichs-Lewy* or CFL condition, given in it simplest form by

$$
\delta t_{\mathrm{CFL}} \leq \frac{\delta x}{c},
\tag{3.161}
$$

where δx is a characteristic length scale, and c is a characteristic speed (Anderson, 1995). For SPH simulations, these are both well defined; the smoothing length h provides the characteristic length, and sound speed c_{s} gives the characteristic speed

of the medium. The CFL condition for particle i then becomes

$$\delta t_{\mathrm{CFL}} \leq \frac{h}{c_{\mathrm{s}}}. \tag{3.162}$$

This has a ready physical interpretation in that it prevents spatial information transfer through the code at a rate greater than the local sound speed. In the presence of artificial viscosity this requires a slight alteration, and as such Bate et al. (1995) recommend using the following criterion;

$$\delta t_{\mathrm{CFL}} = \frac{0.3h}{c_{\mathrm{s}} + h|\nabla \cdot \mathbf{v}| + 1.2(\alpha_{\mathrm{SPH}}c_{\mathrm{s}} + \beta_{\mathrm{SPH}}h|\nabla \cdot \mathbf{v}|)}, \tag{3.163}$$

where the factors of 0.3 in the numerator and 1.2 in the denominator are empirically determined. The α_{SPH} and β_{SPH} terms are those used to determine the strength of the artificial viscosity (see Section 3.4), and it should be noted that the final term in the denominator is only included in the case where $|\nabla \cdot \mathbf{v}| < 0$. The extra $h|\nabla \cdot \mathbf{v}|$ term in the denominator accounts for the expansion or contraction of the flow, and thus explicitly allows for compressibility effects. There are variations on this theme (see for instance Deegan 2009; Monaghan 1992; Monaghan 1989) but the definition given above is the one present in the code I have used.

3.8.3.2 Force Condition

A further commonly used timestep condition is that based on the acceleration of the particle, known as the force condition. This is simple in form, and is given by

$$\delta t_{\mathrm{F}} = f_{\mathrm{F}} \sqrt{\frac{h}{|\mathbf{a}|}}, \tag{3.164}$$

where as before $\mathbf{a}$ is the particle acceleration, and $f_{\mathrm{F}} < 1$ is a tuning constant. Values for f_{F} vary from code to code but are generally in the range 0.25 - 0.5 (Wetzstein et al., 2009; Bate et al., 1995; Monaghan, 1989). The code I have used employs $f_{\mathrm{F}} = 0.3$.

3.8.3.3 Integrator Limits

Dependent on the choice of integrator, other timestep criteria may be required. In particular, if using the RKF method the timestep criterion associated with the error

correction must be incorporated. Using the method outlined above, this corresponds to a timestep of

$$\delta t_{\mathrm{RKF}} = \delta t_{\mathrm{old}} \sqrt{\frac{512\epsilon}{|\mathbf{z}_{RK2} - \mathbf{z}_{RK1}|}}, \tag{3.165}$$

where ϵ is the desired error tolerance (usually of the order of 10^{-4} - 10^{-5}) and the $\mathbf{z}_{RK1}, \mathbf{z}_{RK2}$ are the predictions for any quantity $\mathbf{z}$ from the first and second order methods within the integrator respectively. The δt_{old} term is simply the increment used for the previous step.

3.8.3.4 Generalised Timestep Criteria

A general class of additional timestep criteria may be obtained by dimensional analysis, in that for any time-varying quantity z we may define a characteristic timescale on which it varies as

$$\tau_z = \frac{z}{\dot{z}}, \tag{3.166}$$

where as usual $\dot{z}$ is the time derivative of z. To ensure that this timescale is properly resolved, we can therefore define a timestep condition such that

$$\delta t_z = f_z \frac{z}{\dot{z}}, \tag{3.167}$$

where $f_z < 1$ is a tuning factor. Although seldom required in general, a timestep criterion of this form was implemented into the code when looking at the effects of strongly varying cooling times in Chapter 5, and will be discussed in more detail there.

3.8.4 Setting the Timestep

There are therefore a variety of possible timestep choices, and thus to ensure that they are all satisfied, the timestep for each particle used is the minimum of all possibilities, i.e.

$$\delta t_i = \min(\delta t_{\mathrm{CFL},i}, \delta t_{\mathrm{F},i}, \delta t_{\mathrm{RKF},i}, \delta t_z). \tag{3.168}$$

Where there are only relatively small changes in the characteristic timescale, a standard choice is to use a global timestep δt_{glob}, which is set by the minimum of the timesteps δt_i for the individual particles, such that

$$\delta t_{\mathrm{glob}} = \min_i (\delta t_i). \tag{3.169}$$

This has the advantage that all particles are evolved in lockstep, and thus there is no 'information lag' due to particles being on separate timesteps.

On the other hand, individual particle timestepping has the advantage of being much faster and thus more computationally efficient wherever there are large ranges in the timescales of the problem being investigated. It can however introduce instabilities into the integrator (Wetzstein et al., 2009), and can also lead to the phenomenon of low density particles on long timesteps drifting into regions of high density evolving on much shorted timesteps, leading to spurious entropy generation (Pearce, 2010, private communication). This latter effect is particularly noticeable in tests of Sedov blasts (see for instance Tasker et al. 2008), in which small entropy-driven bubbles lead to granularity in the post-shock region. This is a relatively uncommon phenomenon however, and occurs principally in the case of strongly shocked systems.

Integrator stability may be maximised (particularly for the leapfrog scheme) by using timesteps that are integer multiples of each other, and generally speaking, for a maximum timestep T, sub-timesteps will be given by $2^{-n}T$. The particle timesteps are therefore rounded *down* to the nearest relevant power of two in this case. This is the case in the code I have used for all the simulations presented in this thesis, which uses individual particle timesteps and is only weakly shocked throughout.

3.9   Summary

In this chapter I have derived the SPH algorithm from first principles, and then built it up in a series of steps to solve for pure hydrodynamical isentropic flows, dissipational flows, and finally dissipational flows under the influence of gravitational forces. Additionally I have shown that it is possible to self-consistently allow for spatially variable smoothing lengths, which allows the algorithm to be highly adaptive with the local fluid density, but to maintain exact conservation of mass, linear and angular momentum and energy, to within the integrator tolerance. In the case of isentropic flows, entropy is also conserved by construction. Furthermore I have also briefly detailed various methods of finding the nearest neighbours, and two means of evolving the fluid flow forward in time.

Since the problems I shall be considering in later chapters require only that dissipational flow in the presence of gravity to be modelled, this is all that I have covered here. However this is by no means the limit for the SPH for-

malism. Much effort has been put into including additional physics such as radiative transfer (Nayakshin et al. (2009); Petkova & Springel (2009); Forgan et al. (2009); Bisbas et al. (2009); Gritschneder et al. (2009); Pawlik & Schaye (2008) and Altay et al. (2008) to name but few of the recent efforts) and magnetic fields/MHD (see for instance Price (2010); Dolag & Stasyszyn (2009); Rosswog & Price (2007); Price & Monaghan (2005, 2004) and Price & Monaghan 2004), and this will no doubt continue as computing power steadily increases.

As with any numerical scheme however, SPH remains an approximation to reality, and as such reality checks are required in the form of standard tests. These act as calibration routines, to ensure the the results of any simulations are physically realistic, and can be relied upon. Many such tests exist, and there are far too many to do justice to here, but see for instance the astro code wiki[8], which has a number of cross-comparison tests with other codes, specifically aimed at disc-like models. As the code I use is a derivative of the one discussed in Price (2005) the discussion of numerical tests found here is particularly appropriate. A further suite of standard tests including Sod shocks and Sedov blasts among others, used for both code verification and comparison, is given in (Tasker et al., 2008).

---

[8]http://www-theorie.physik.unizh.ch/astrosim/code/doku.php?id=home:home

References

Altay, G., Croft, R. A. C., & Pelupessy, I.: 2008, *MNRAS* **386**, 1931

Anderson, J. D.: 1995, *Computational Fluid Dynamics: The Basics With Applications*, McGraw-Hill

Balsara, D. S.: 1995, *Journal of Computational Physics* **121**, 357

Barnes, J. & Hut, P.: 1986, *Nature* **324**, 446

Bate, M. R., Bonnell, I. A., & Price, N. M.: 1995, *MNRAS* **277**, 362

Bentley, J. L.: 1975, *Commun. ACM* **18(9)**, 509

Benz, W.: 1990, in J. R. Buchler (ed.), *Numerical Modelling of Nonlinear Stellar Pulsations Problems and Prospects*, p. 269

Bisbas, T. G., Wünsch, R., Whitworth, A. P., & Hubber, D. A.: 2009, *A&A* **497**, 649

Cha, S. & Whitworth, A. P.: 2003, *MNRAS* **340**, 73

Cullen, L. & Dehnen, W.: 2010, *arXiv:1006.1524*

Deegan, P.: 2009, *Ph.D. thesis*, Univ. Leicester, (2009)

Dehnen, W.: 2000, *ApJ* **536**, L39

Dehnen, W.: 2002, *Journal of Computational Physics* **179**, 27

Dehnen, W.: 2009, Private Communication

Dolag, K. & Stasyszyn, F.: 2009, *MNRAS* **398**, 1678

Dolag, K., Vazza, F., Brunetti, G., & Tormen, G.: 2005, *MNRAS* **364**, 753

Fehlberg, E.: 1968, *NASA Technical Report,* NASA-TR-R-287

Fehlberg, E.: 1969, *NASA Technical Report,* NASA-TR-R-315

Forgan, D., Rice, K., Stamatellos, D., & Whitworth, A.: 2009, *MNRAS* **394**, 882

Fulk, D. A. & Quinn, D. W.: 1996, *Journal of Computational Physics* **126(1)**, 165

Gritschneder, M., Naab, T., Burkert, A., Walch, S., Heitsch, F., & Wetzstein, M.:
    2009, *MNRAS* **393**, 21

Hernquist, L.: 1987, *ApJS* **64**, 715

Hernquist, L.: 1993, *ApJ* **404**, 717

Hernquist, L. & Katz, N.: 1989, *ApJS* **70**, 419

Hockney, R. W. & Eastwood, J. W.: 1981, *Computer Simulation Using Particles*

Inutsuka, S.: 2002, *Journal of Computational Physics* **179**, 238

Kutta, W.: 1901, *Zeitschrift fr. Mathematische Physik* **46**, 435

Landau, L. D. & Lifshitz, E. M.: 1959, *Fluid mechanics*

Landshoff, R.: 1930, *Los Alamos Laboratory Report*

Lattanzio, J. C., Monaghan, J. J., Pongracic, H., & Schwarz, M. P.: 1985, *MNRAS*
    215, 125

Lodato, G. & Price, D. J.: 2010, *MNRAS* **405**, 1212

Lodato, G. & Rice, W. K. M.: 2004, *MNRAS* **351**, 630

Lombardi, J. C., Sills, A., Rasio, F. A., & Shapiro, S. L.: 1999, *Journal of Compu-
    tational Physics* **152**, 687

Monaghan, J.: 1989, *Journal of Computational Physics* **82(1)**, 1

Monaghan, J. J.: 1985, *Computer Physics Reports* **3(2)**, 71

Monaghan, J. J.: 1992, *ARA&A* **30**, 543

Monaghan, J. J. & Lattanzio, J. C.: 1985, *A&A* **149**, 135

Morris, J. P. & Monaghan, J. J.: 1997, *Journal of Computational Physics* **136(1)**, 41

Murray, J. R.: 1996, *MNRAS* **279**, 402

Nayakshin, S., Cha, S., & Hobbs, A.: 2009, *MNRAS* **397**, 1314

Nelson, A. F., Wetzstein, M., & Naab, T.: 2009, *ApJS* **184**, 326

Pawlik, A. H. & Schaye, J.: 2008, *MNRAS* **389**, 651

Pearce, F. R.: 2010, Private Communication

Petkova, M. & Springel, V.: 2009, *MNRAS* **396**, 1383

Press, W., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P.: 2007, *Numerical Recipes – the Art of Scientific Computing (3rd Edition)*, Cambridge University Press

Price, D.: 2005, *arXiv:0507.472*

Price, D. J.: 2010, *MNRAS* **401**, 1475

Price, D. J. & Monaghan, J. J.: 2004, *MNRAS* **348**, 139

Price, D. J. & Monaghan, J. J.: 2005, *MNRAS* **364**, 384

Price, D. J. & Monaghan, J. J.: 2007, *MNRAS* **374**, 1347

Quinn, T., Katz, N., Stadel, J., & Lake, G.: 1997, *arXiv:astro-ph/9710043*

Read, J. I., Hayfield, T., & Agertz, O.: 2010, *MNRAS* **405**, 1513

Rosswog, S.: 2009, *New Astronomy Review* **53**, 78

Rosswog, S., Davies, M. B., Thielemann, F., & Piran, T.: 2000, *A&A* **360**, 171

Rosswog, S. & Price, D.: 2007, *MNRAS* **379**, 915

Springel, V.: 2005, *MNRAS* **364**, 1105

Springel, V. & Hernquist, L.: 2002, *MNRAS* **333**, 649

Springel, V., Yoshida, N., & White, S. D. M.: 2001, *New Astronomy* **6**, 79

Steinmetz, M.: 1996, *MNRAS* **278**, 1005

Steinmetz, M. & Mueller, E.: 1993, *A&A* **268**, 391

Tasker, E. J., Brunino, R., Mitchell, N. L., Michielsen, D., Hopton, S., Pearce, F. R., Bryan, G. L., & Theuns, T.: 2008, *MNRAS* **390**, 1267

von Neumann, J. & Richtmyer, R. D.: 1950, *Journal of Applied Physics* **21**, 232

Wetzstein, M., Nelson, A. F., Naab, T., & Burkert, A.: 2009, *ApJS* **184**, 298