Chapter 2

N-body Techniques & Tools

2.1 Introduction

Our goal is to simulate galaxy mergers consisting of dark matter and gas. To achieve this, we must first create an equilibrium model for different galaxies with gaseous dark matter halos. Then, we must accurately follow the evolution of two such galaxies that are placed in a orbit around each other. Thus, there are two aspects of the modeling: *(i).* creation of an equilibrium galaxy model, and *(ii).* correctly computing the gravitational forces to describe the time-evolution of the galaxies as they undergo the violent encounter. In this chapter, we will examine the details of the various N-body techniques and tools that were used to run our merger simulation.

2.2 Collisionless Systems

A galaxy is made up of dark matter, stars and gas that move around under the force of mutual gravitational attraction. The global dynamics of this ensemble of particles is determined by the overall structure and potential of the galaxy, although external perturbations like satellite flybys can trigger long lived global features like bars, spiral arms and warps. Due to the coarse-grainedness of an actual galactic potential, a star will be deflected from its 'true' orbit in a smooth potential. A star moving with an average velocity v in a galaxy of radius R

containing N equal-mass stars will have an aggregate velocity change of the order v in one relaxation time, $t_{\rm relax}$, given by :

$$t_{\rm relax} = \frac{N}{8 \ln N} \times \frac{R}{v}. \tag{2.1}$$

In practice, we compare the relaxation time of a system with its crossing time, $t_{\rm cross} = R/v$, to determine whether the star will suffer from significant encounters in a Hubble time. For the $N = 10^{11}$ stars in a typical galaxy with $R = 20$ kpc and $v = 200$ km/s, $t_{\rm cross} = 10^8$ years, making $t_{\rm relax} = 4.9 \times 10^{16}$ years. So, a typical star will not deviate at all over the period of a Hubble time. Such a system where the timescale for close encounters is much longer than a Hubble time, is called a *collisionless system*. Stars in the galaxy constitute such a collisionless system. Even if the galaxy were to undergo a merger, it is unlikely for any star to directly encounter another star; hence the stars can be treated as collisionless particles during the course of a merger simulation. N-body models of galaxies, however, are not collisionless systems. The smallest of our galaxies is represented with $N \sim 10^5$ particles and has a relaxation time of ~ 600 Gyrs (see Table 3.1). Fortunately, the galaxies we are simulating undergo a violent merger within 1 Gyr; therefore, we only require galaxies to be stable against two-body interactions over the period of ~ 1 Gyr.

2.3 Distribution Function

The task of building an N-body model requires determining the six phase-space coordinates, i.e, (x, y, z, v_x, v_y, v_z) at some time t_0. Given such a description, it is possible to compute the state of a purely collisionless system at any later time by using Newton's Laws of Motion. This 6-D phase space description is called the distribution function (hereafter, DF), $f(\mathbf{x}, \mathbf{v}, t_0)$. The DF gives the number of particles in an infinitesimal spatial volume $\mathrm{d}^3\mathbf{x}$ around the position $\mathbf{x}$ and in the infinitesimal velocity volume of $\mathrm{d}^3\mathbf{v}$ at the instant t_0. Thus, a DF is non-negative everywhere. The time-evolution of a collisionless system is governed by

the collisionless Boltzmann equation (CBE) written in its simplest form as :

$$\frac{\mathrm{d}\, f(\mathbf{r}, \mathbf{v}, t)}{\mathrm{d}\, t} = 0. \tag{2.2}$$

In general, a DF depends on the value of the specific total energy, $E = \Phi(\mathbf{r}) + \frac{1}{2}\mathbf{v}^2$ and the specific angular momentum, $\mathbf{L} = \mathbf{r} \times \mathbf{v}$; however, it can be shown that the DF for a spherically symmetric density with an isotropic velocity distribution is dependent only on the total energy per unit mass, E, i.e., $f(\mathbf{r}, \mathbf{v}) \equiv \hat{f}(E)$ given by the Eddington inversion :

$$f(\mathcal{E}) = \frac{1}{\sqrt{8}\pi^2}\left[\int_0^E \frac{\mathrm{d}^2\rho}{\mathrm{d}\Psi^2}\frac{\mathrm{d}\Psi}{\sqrt{\mathcal{E}-\Psi}} + \frac{1}{\sqrt{\mathcal{E}}}\Big(\frac{\mathrm{d}\rho}{\mathrm{d}\Psi}\Big)_{\Psi=0}\right] \tag{2.3}$$

where, $\mathcal{E} = -E$, $\Psi = -\Phi$ and ρ is the density. For a finite size of a system, the second term is equal to zero and the DF is given by just the first term. Now with a density, ρ and a potential, Φ, we can numerically solve the above equation and derive the numerical DF. We can recover the density from the DF by using the relation :

$$\rho(\mathbf{r}) = \int f(\mathbf{r}, \mathbf{v})\mathrm{d}^3\mathbf{v}. \tag{2.4}$$

Therefore, given the DF for a system, it is possible to recover the underlying density profile and vice-versa. Likewise, the integral over $\mathrm{d}^3\mathbf{r}$ yields a velocity profile.

In order to make a galaxy model, one can get the particle positions by directly sampling the density profile using acceptance-rejection technique (see next section); however, obtaining the particle velocities is a bit more involved. For an isolated galaxy in equilibrium, the DF has no dependence on time, i.e., $f(\mathbf{r}, \mathbf{v}, t) \equiv f(\mathbf{r}, \mathbf{v})$. Jeans theorem states that in such a case, any solution to the CBE (Eqn 2.2) can be represented as a function of constants of motion, e.g., angular momentum. If the velocity distribution for the particles in the model is approximated by a 3-D Gaussian, specified by a mean velocity and velocity dispersion along each co-ordinate axes, then the complete velocity profile can be uniquely determined by computing the constants of motion. Most of the previous numerical work has been done under this simplifying assumption; however, such an approach is approximate and the density and velocity profiles can vary drastically from the intended ones

(Kazantzidis et al., 2004; Holley-Bockelmann et al., 2005). Therefore, we avoid that method and calculate the distribution function of the potential-density pair given by a density profile and draw position-velocity pairs directly from the DF.

2.4 Acceptance-Rejection Technique

The acceptance-rejection technique is a method of sampling a probability distribution function (hereafter, p.d.f.) by using the fact that a random generator will naturally produce an unbiased sample of random numbers; now, if this sample of random numbers is compared against the p.d.f. and the random numbers greater than the value of the p.d.f. are rejected, then the remaining sample members will naturally reflect the shape of the original p.d.f. This class of methods where a uniform random number generator is used to compute some property or generate a distribution as in this case, is called a Monte-Carlo method. Figure 2.4 shows a Monte-Carlo method to estimate the value of π. N-sets of two random numbers are drawn in the range [0,1] and each one is plotted on the figure. The probability that a particular number falls inside the quadrant is proportional to the area of the quadrant. We can directly estimate the probability of a number falling in this quadrant just by counting the number of actual such occurrences, say N_1 and dividing by the total number of tries (N, in this case). Then we can use N_1/N =Area of quadrant/Area of square, to obtain an estimate for π. As N increases, we obtain a better estimate of π since the error of a Monte-Carlo method reduces as $\mathcal{O}(1/\sqrt{N})$.

The acceptance-rejection technique is a Monte-Carlo method to create a realization of some distribution, $f(x)$, given another distribution, $g(x)$ such that $A\,g(x) \geq f(x)\,\forall x$, where $A > 1$. The acceptance-rejection technique is usually used where the form of $f(x)$ makes it difficult to invert and obtain $f^{-1}(x)$, making sampling from $f(x)$ difficult. Therefore, another more easily invertible distribution, $g(x)$, is used as an limiting value for $f(x)$, and samples are drawn from $g^{-1}(x)$ and accepted when an uniform random in $[0, 1]$ does not exceed $f(x)/A\,g(x)$. This value of x drawn from $g(x)$ is then accepted as a sample of $f(x)$. The efficiency of the algorithm depends on the value of A; for values of A closer to 1 the algorithm is more efficient since values of x sampled from $g(x)$ get rejected less frequently.

The flow-chart for the acceptance-rejection technique is shown in Fig. 2.4.

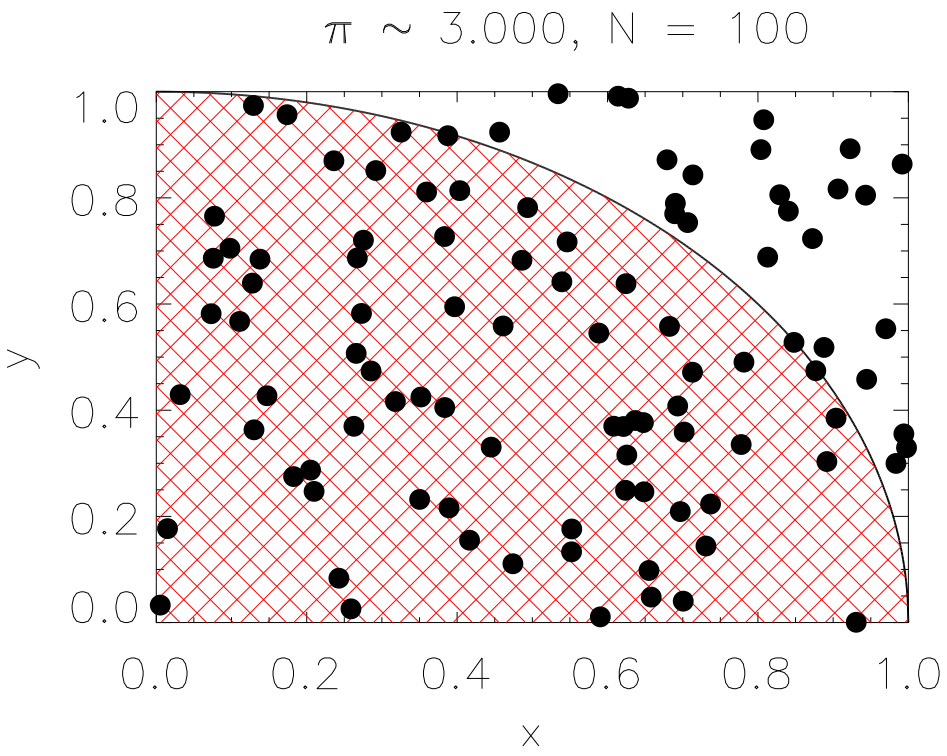


Figure 2.1. Monte-Carlo estimate of the area of the positive quadrant of an unit circle given by $x^2 + y^2 = 1$. Analytically the result is known to be $\pi/4$, and numerically it can be approximated by the ratio of the number of points that occur in the shaded region to the total number of points. Here, we show a random sample of $N = 100$ (x, y) pairs plotted as the solid circles, and we find an approximate value for π to be 3.0. As the number of tries increase, better convergence is achieved. One of the biggest advantages of a Monte-Carlo method is that the error is always $\mathcal{O}(1/\sqrt{N})$ irrespective of the dimensionality of the problem itself. The Monte-Carlo method relates to the acceptance-rejection technique in that the number of solid circles in the shaded region is larger for a smaller value of x (and a corresponding larger value of $y = \sqrt{1 - x^2}$). Thus, the actual filled circles in the shaded region for a given value of x can be used as a representative sample of the probability distribution given by the quadrant.

2.5 Simulation Software

We use the parallel, hydrodynamic code GADGET-2 (Springel et al., 2001; Springel, 2005)[1] for all the numerical simulations in this paper. GADGET-2 computes

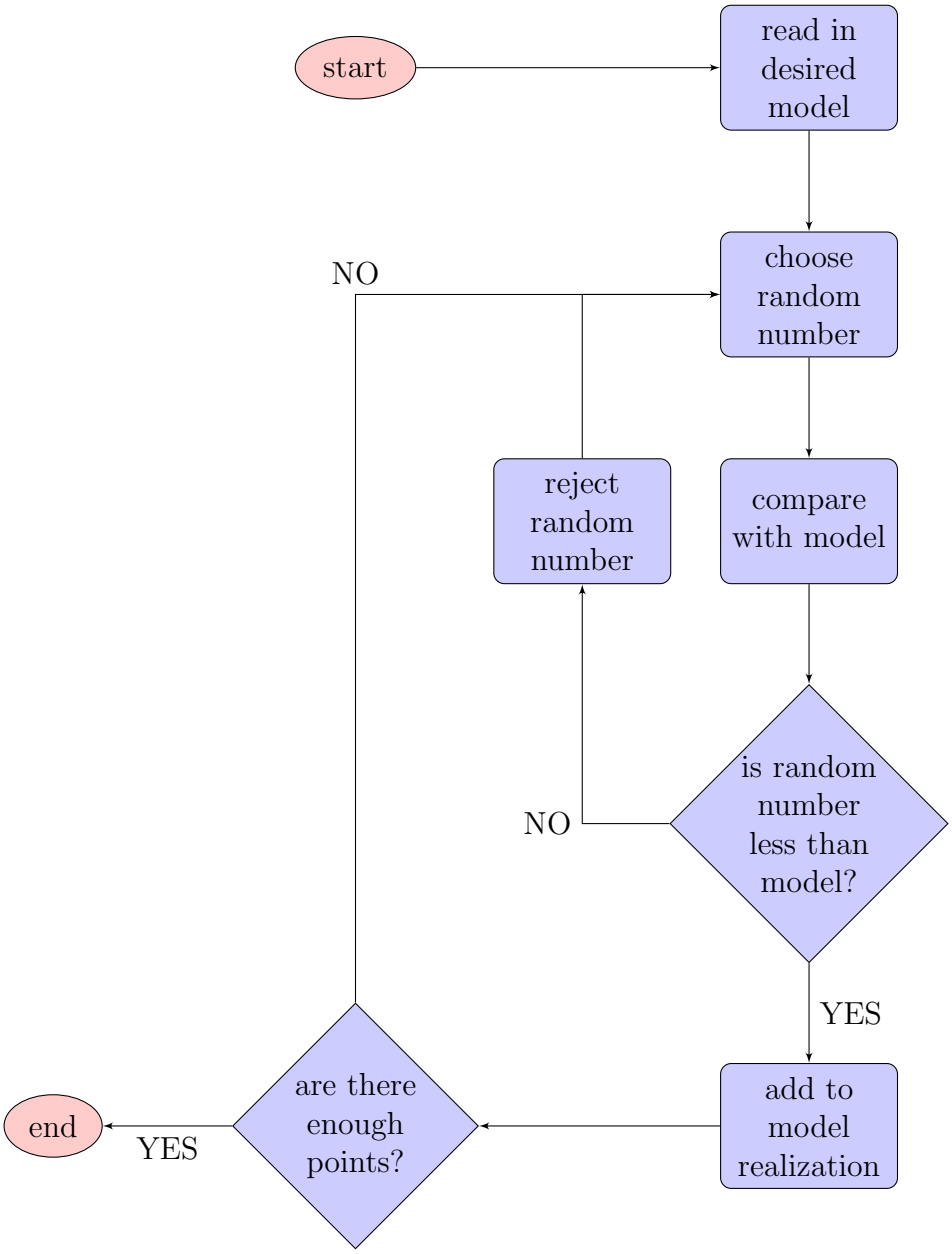


Figure 2.2. A flow-chart for the acceptance-rejection algorithm. This is a Monte-Carlo method to create a realization of some distribution, $f(x)$, given another distribution, $g(x)$ such that $A\,g(x) \geq f(x)\,\forall x$. This involves generating a uniform random number in [0,1] and a value x from the distribution $g(x)$. If the value of the random number is less than the ratio of $f(x)/A\,g(x)$, then the value of x is accepted a sample of $f(x)$.

the gravitational forces with a hierarchical tree algorithm (Barnes & Hut, 1986) while gas particles receive additional hydrodynamic acceleration as calculated using Smooth Particle Hydrodynamics (Gingold & Monaghan, 1977, hereafter SPH). The code explicitly conserves energy and entropy for the SPH particles and uses adaptive time-steps for the time-evolution. Since this is a first attempt to model the gas behavior during the merger, cooling, star formation and radiative feedback have all been neglected.

2.6 N-body Mechanics

N-body mechanics refers to simulations of dark matter and stars where a group of 'N' particles are evolved from an initially specified set of ($\overrightarrow{\mathbf{r}}$, $\overrightarrow{\mathbf{v}}$) under mutual gravitational forces. The arrangement of the 'N' particles are chosen to model a particular situation from planetary rings to the Universe as a whole.

The dominant force on the largest length scales relevant to astrophysics is Newtonian gravity. The set of initial positions and velocities can be evolved using the standard equation of motion under gravity :

$$a_i \;=\; -\sum_{j\neq i} \frac{G m_j |\overrightarrow{r_i} - \overrightarrow{r_j}|}{|\overrightarrow{r_i} - \overrightarrow{r_j}|^3}, \tag{2.5}$$

where a_i represents the net acceleration of the i^{th} particle from all the other j particles, each with mass m_j . This equation is valid for each individual particle and leads to a set of N coupled, non-linear second order differential equations that must be numerically integrated $\forall N > 2$. However, the force law, written in the above form suffers from a numerical singularity as $|\overrightarrow{\mathbf{r_i}} - \overrightarrow{\mathbf{r_j}}| \rightarrow 0$. In order to retain relative accuracy of the numerical integration, the time-steps for particles with vanishing separation become infinitesimally small, thereby reducing the overall computational efficiency of the simulation. Constant time-stepping could be used to avoid this scenario, but the accumulated errors for such close encounters then becomes large and may lead to unphysical situations such as spurious binary formation. This close encounter scenario is avoided in numerical simulations by

[1]GADGET-2 is available for download at http://www.mpa-garching.mpg.de/gadget/

modifying the force law at small separations by adding a 'softening parameter', $\epsilon > 0$, to the force equation such that the force does not diverge. This modified force law can be written as :

$$\overrightarrow{F_i} = -\sum_{j \neq i} \frac{G m_i m_j |\overrightarrow{r_i} - \overrightarrow{r_j}|}{\left(|\overrightarrow{r_i} - \overrightarrow{r_j}|^2 + \epsilon^2\right)^{3/2}}. \tag{2.6}$$

The choice of the softening parameter is highly dependent on the simulation type and the exact object being simulated, but is generally taken to be on the order of the mean inter-particle separation. It has to be noted that any simulation with a finite ϵ can not be used to predict results for length scales $< \epsilon$. Thus, the spatial resolution of a gravitational simulation is set by the choice of ϵ. To make sure that this modification of the small-scale force does not have an effect on the large-scale dynamics, a gravitational kernel is defined so that the force becomes exactly Newtonian at some length scale ϵ. This is done in GADGET-2 by using a spline kernel (Monaghan & Lattanzio, 1985) $W(|x|, h = 2.8\epsilon)$ of the form :

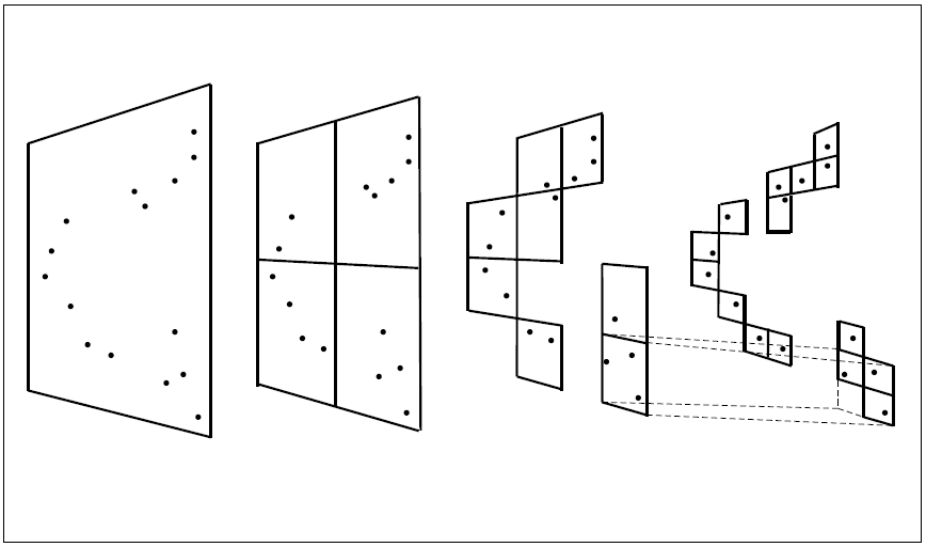
$$W(r,h) = \frac{8}{\pi h^3} \begin{cases} 1 - 6\left(\frac{r}{h}\right)^2 + 6\left(\frac{r}{h}\right)^3, & 0 \leq \frac{r}{h} \leq \frac{1}{2}, \\ 2\left(1 - \frac{r}{h}\right)^3, & \frac{1}{2} < \frac{r}{h} \leq 1, \\ 0, & \frac{r}{h} > 1. \end{cases} \tag{2.7}$$

where, r is the separation between the particles between which the force is being computed. Even with softening, directly summing the gravitational forces over N particles involves computing $\frac{1}{2}N(N-1)$ forces, which is an algorithm of $\mathcal{O}(N^2)$; the processor time per time-step becomes prohibitively large for values of N required to describe astrophysical systems (e.g., current state of the art simulations use up to 10^{10} particles, and that number is ever-increasing in keeping with Moore's Law). The long range nature of gravity makes it difficult to compute the forces accurately and efficiently. One method that is used to achieve the required spatial adaptivity groups particles in a hierarchical multipole expansion, usually referred to as a 'tree' algorithm. The specific algorithm used is the Barnes-Hut (BH) octal

tree (Barnes & Hut, 1986), which involves grouping distant particles together such that the total force for each particle can be computed by $\mathcal{O}(\log N)$ interactions. This strategy reduces the overall complexity of the computation to $\mathcal{O}(N \log N)$, and yields an acceptable total run-time for a simulation. The BH tree uses a cubic parent node that contains the entire mass distribution and continues to recursively sub-divide each node into 8 spatially equal 'daughter cells' until only one particle is contained within a node (called a 'leaf'). Each node is then associated with a multipole expansion of all the daughter nodes. The parameter that controls the accuracy of the tree algorithm is the 'cell opening angle', or the tolerance, represented by α in the following equation for a node of mass M and size l at a distance r :

$$\frac{GM}{r^2}\left(\frac{l}{r}\right)^2 \leq \alpha\,|\mathbf{a}|, \tag{2.8}$$

where $|\mathbf{a}|$ is the total acceleration obtained in the previous time-step. While computing the gravitational force, each node is opened and compared with the set parameter, α, to see if the multipole expansion is accurate enough (this happens if the node is small and distant from the particle for which the force is being computed). If so, the multipole expansion is used, the 'tree walk' is terminated; if not, the daughter nodes are opened recursively until the desired accuracy is reached. As such, the accuracy can be increased to match the exact force given by Eqn. (2.6), albeit at a higher computational cost. The cell-opening criteria used in GADGET-2 (refer to Eqn. 2.8) compares an estimate of the total force with the truncation error for each particle-node interaction and limits the absolute error. It has the advantage that the cell opening adaptively adjusts with the acceleration. Such a cell opening criteria can be subject to unbounded errors in the case that a particle is sufficiently close to a node (Salmon & Warren, 1994); to prevent this from occurring an additional criteria is imposed such that a node is always opened if a particle is located inside geometric boundaries 10% larger than the node.



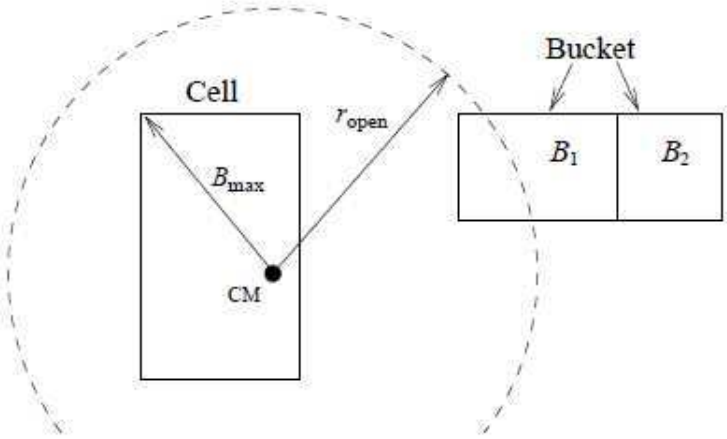


Figure 2.3. *Top.* A schematic for the Barnes-Hut method of tree construction that is used in GADGET-2. Each cube (node) is recursively divided into 8 smaller cubes until all the smallest cubes contain exactly one particle. *Bottom.* An illustration of the opening angle for the tree code. If θ is small as per Eqn 2.8, then the multiple expansion is used. Otherwise the node is opened and similar comparisons are carried out with the daughter cells. Figure is taken from (Wadsley et al., 2004).

2.7 Smooth Particle Hydrodynamics

Gas is difficult to simulate accurately, due to the additional hydrodynamic evolution equations that the gas must follow. In numerical simulations, gas is usually represented as a fluid that must satisfy the continuity equation in addition Newtonian gravity. Conventional grid-based numerical techniques have extreme memory requirements for 3-D simulations of very large scale regions. This is because each point in space is a part of the mesh which serves to compute spatial derivatives for quantities like velocity, pressure etc. This means that even empty regions in space must contain grid points, which requires enormous memory if sufficient resolution is to be maintained throughout the simulation box.

Smooth Particle Hydrodynamics is a particle method (Lucy, 1977; Gingold & Monaghan, 1977; Monaghan, 1992) that overcomes the memory constraints by representing the fluid with a finite number of particles that interact hydro-dynamically with each other – characteristic of gas in astrophysical situations. Each particle experiences gravitational forces as well as gas pressure from the particles that are nearby. Ideally, if the particle interacts with every other particle then a complete analytic solution is obtained. In practice, SPH schemes use an interpolating function, a kernel similar to the one used in evaluating softened gravitational forces[1], that has a finite value for small separations between two interacting particles and is zero when the separation is too large compared to a hydrodynamic length scale. This characteristic length scale within which particles interact is called the smoothing length, h_i. The smoothing length is adaptive in most SPH simulations and adjusts to encompass a fixed number of neighbors. Since the particle motion matches the fluid flow, an adaptive smoothing length naturally ensures higher resolution in denser locations.

Any function, $f(r)$ can be recast in terms of the kernel as :

$$f_{\rm interp}(r_i) = \sum_{j \neq i} m_j \frac{f(r_j)}{\rho(r_j)} W(\mathbf{r_i} - \mathbf{r_j}; h), \tag{2.9}$$

where the usual integral has been replaced by a more convenient summation. Replacing $f(r)$ with $\rho(r)$, one can immediately see that the density of an SPH particle

[1]In GADGET-2, the kernels for SPH and gravity are identical

is given by :

$$\rho_i = \sum_{j \neq i} m_j W(\mathbf{r_i} - \mathbf{r_j}; h_i) \tag{2.10}$$

The derivative of any function, $f(r)$, written in terms of the interpolant becomes:

$$\nabla f_{\rm interp}(r_i) = \sum_{j \neq i} m_j \frac{f(r_j)}{\rho(r_j)} \nabla W(\mathbf{r_i} - \mathbf{r_j}; h_i) \tag{2.11}$$

Such a formulation ensures that derivatives of numerically ill-behaved quantities can be easily computed without having to resort to special formulations to deal with numerical discontinuities. The derivative of any function then becomes the convolution of the value of the function and the pre-computed analytic derivative of the kernel, an incredible simplification in terms of the numerical overhead. This advantage, however, becomes a problem when it comes to resolving actual physical discontinuities – shocks. SPH techniques will intrinsically smooth out a shock and broaden it over 2-3 smoothing lengths; though observations indicate that the shock front is very sharp. We will discuss the methods used in SPH simulations to realistically capture shocks in a later section.

The conservation of momentum locally requires Newton's third law be satisfied for each force calculation between two particles. To achieve this, the kernel and its derivatives are symmetrized (Hernquist & Katz, 1989):

$$W_{ij} = \frac{1}{2}\left[W_i(\mathbf{r}; h_i) + W_j(\mathbf{r}; h_j)\right]; \mathbf{r} = \mathbf{r_i} - \mathbf{r_j} \tag{2.12}$$

$$\nabla W_{ij} = \frac{1}{2}\left[\nabla_i W_i(\mathbf{r_i}; h_i) + \nabla_j W_j(\mathbf{r_j}; h_j)\right], \tag{2.13}$$

where i and j refer to two different particles between which hydrodynamic forces are being computed, W_i and W_j are the values of the kernel obtained using the respective smoothing lengths, h_i and h_j respectively. The equation of motion in this discretized representation of the gas then becomes :

$$\frac{{\rm d}\mathbf{v}_i}{{\rm d}t} = -\sum_{j=1}^{N_{\rm ngb}} m_j \left[f_i \frac{P_i}{\rho_i^2} \nabla_i W_{ij}(h_i) + f_j \frac{P_j}{\rho_j^2} \nabla_i W_{ij}(h_j) \right], \tag{2.14}$$

where $f_i = \left(1 + \frac{h_i}{3\rho_i}\frac{\partial \rho_i}{\partial h_i}\right)^{-1}$ and P_i is the pressure of the $i^{\rm th}$ gas particle. We explicitly include the fact that the hydrodynamic sum is carried out only over the $N_{\rm ngb}$ neighbors. If the gas were to be isentropic throughout the course of a simulation, then the above equations are enough to describe the state of the gas at all times. Unfortunately, actual astrophysical situations involve shocks quite frequently which changes the entropy of the system. To handle this additional term, SPH methods use a viscous force and 'hide' all the micro-physics of shock generation and dissipation into a term known as artificial viscosity (Monaghan & Gingold, 1983; Monaghan, 1992; Balsara, 1995). This viscous force causes an additional acceleration term to be present in the equation of motion for the gas particles :

$$\left.\frac{{\rm d}\mathbf{v}_i}{{\rm d}t}\right|_{\rm visc} = -\sum_{j=1}^{N_{\rm ngb}} m_j \Pi_{ij} \nabla_i W_{ij}\,, \tag{2.15}$$

where Π_{ij} is a viscous term that takes on a non-zero value only for a convergent flow ($v \cdot r < 0$). This viscous force results in an irreversible transformation of gas kinetic energy into thermal energy with a corresponding rate of entropy generation given by :

$$\frac{{\rm d}S_i}{{\rm d}t} = \frac{1}{2}\frac{\gamma - 1}{\rho_i^{\gamma-1}} \sum_{j=1}^{N_{\rm ngb}} m_j \Pi_{ij} \mathbf{v}_{ij} \cdot \nabla_i W_{ij}\,, \tag{2.16}$$

where S_i is the entropy for the i^{th} gas particle, γ is the polytropic index, v_{ij} is the relative velocity between the i^{th} and the j^{th} gas particle and ρ_i is the density of the i^{th} particle. In GADGET-2, Π_{ij} is computed from the formula:

$$\Pi_{ij} = -\frac{\alpha}{2}\frac{\left[c_i + c_j - 3w_{ij}\right] w_{ij}}{\rho_{ij}}, \tag{2.17}$$

$$w_{ij} = \begin{cases} \mathbf{v}_{ij} \cdot \mathbf{r}_{ij}/|\mathbf{r}_{ij}| & \text{if } \mathbf{v}_{ij} \cdot \mathbf{r}_{ij} < 0, \\ 0 & \text{otherwise,} \end{cases} \tag{2.18}$$

where c_i is the sound speed for the i^{th} gas particle, r_{ij} is the relative separation between the i^{th} and j^{th} particle, and α is the artificial viscosity parameter (hereafter, AVP). This ensures that the gas particles receive this additional force only when they approach each other and that the viscous force does not diverge at small

separations, causing the particles to fly apart. Such a formulation of Π_{ij} does not diverge and halts the particles but does not allow them to receive an additional acceleration once they are stationary. The SPH time-step is then calculated with :

$$\Delta t_{\rm hydro} = \frac{C_{\rm Courant}\, h_i}{\max_j (c_i + c_j - 3w_{ij})} \tag{2.19}$$

where $C_{\rm Courant}$ is a parameter less than 1.0 and the denominator is the maximum found over all the neighbors j for the i^{th} particle. The Courant-Friedrichs-Lewy condition (Courant et al., 1928, 1967), commonly shortened as the Courant condition, is a limitation for ensuring convergence while solving time-dependent differential equations. This condition ensures that the time-step taken in a numerical simulation *must* be smaller than the time taken by a characteristic signal to cross a characteristic length. For SPH simulations, the relevant quantities are the sound speed, c_i and the smoothing length, h_i for the i^{th} gas particle. $C_{\rm Courant}$ is taken to be 0.15 in all our simulations.

2.8 Time Integration

Gravity naturally leads to large density contrasts in all collapsed objects resulting in a corresponding range in timescales ($\propto 1/\sqrt{\rho}$). If all the particles were to share the same time-step, low density regions would have artificially small time-steps. Adaptive and individual time-steps are, thus, a very essential requirement for all numerical simulations. The most commonly used time integration scheme used is the 'leapfrog' – a second-order method. The method gets the name because the velocity and position are evaluated at half time-steps, with the velocity and position 'leaping' over each other. Mathematically, for a system with an uniform time-step $\mathrm{d}t$, position x_i, velocity v_i and acceleration a_i at time t_i, the leapfrog method advances the particle with the following algorithm :

$$\begin{aligned} x_i &= x_{i-1} + v_{i-\mathrm{d}t/2} \times \mathrm{d}t/2, \\ a_i &= f(x_i), \\ v_{i+\mathrm{d}t/2} &= v_{i-\mathrm{d}t/2} + a_i \times \mathrm{d}t. \end{aligned} \tag{2.20}$$

where, $f(x_i)$ is some function that gives the net acceleration based on the position of the particle. Thus, the velocities are defined at $t_{i-1/2}$, $t_{i+1/2}$, $t_{i+3/2}\ldots$ while the positions are defined at t_i, t_{i+1}, $t_{i+2}\ldots$.

The leapfrog method is symplectic, meaning that it is time-reversible and ensures that the total energy oscillates near the true total energy of the system. For small enough errors in the integrator, all stable orbits in a real astrophysical system will also be stable in the numerical simulation (Quinn et al., 1997).

In GADGET-2 the time-step, $\Delta t_{\rm grav}$, for an individual particle is determined by the formula:

$$\Delta t_{\rm grav} = \min\left(\Delta t_{\rm max}, \left(\frac{2\,\eta\,\epsilon}{|\mathbf{a}|}\right)^{1/2}\right), \tag{2.21}$$

where $\Delta t_{\rm max}$ defines the maximum time-step permitted for a particle, η is the numerical accuracy parameter, ϵ is the gravitational softening parameter and $|\mathbf{a}|$ is the acceleration of the particle. The time-step for an SPH particle also has to satisfy the Courant condition and is taken as the minimum of $\Delta t_{\rm grav}$ computed above and $\Delta t_{\rm hydro}$ computed in Eqn. 2.19. In practice, GADGET-2 subdivides individual time-steps in powers of two of a global time-step. Particles are allowed to move into a smaller time-step at every iteration but only allowed to attain a larger time-step at every second step if that leads to synchronization with the higher time-step hierarchy.

2.9 Overview

In this chapter, we discussed various N-body methods: Eddington inversion for computing the DF; acceptance-rejection technique; tree codes to compute softened gravitational forces in an efficient manner; SPH methods to capture hydrodynamics; and the time-stepping strategies to accurately evolve astrophysical systems. In the following chapters we will see the application of all of these techniques to create equilibrium galaxies and run merger simulations.