N-body Algorithms and Applications to Galactic Dynamics

Steve Purchase

Doctor of Philosophy
University of Edinburgh
2000
Abstract

Typically galaxies contain $10^{11}$ stars and globular clusters contain $10^6$ stars. In order to investigate the dynamical evolution of these systems one may perform numerical simulations ($N$-body simulations); which in turn are restricted to $10^3$ to $10^7$ particles depending on the required accuracy and adopted evolutionary model. For the $N$-body direct method the computational effort scales at least as $O(N^3)$, and thus the need for efficient and accurate evolutionary models and advances in hardware is apparent. A review of the $N$-body evolutionary models is presented and, with the advent of parallel super computers, the parallelisation of these evolutionary models is investigated. In particular, we discuss the parallelisation of a direct method $N$-body code; focusing upon a new parallelisation strategy, implementing the evolutionary model with a portable parallel language (High Performance Fortran), and reducing the communication costs between processors by a suitable implementation of the Hypersystolic algorithm.

A dynamical system, such as a galaxy, may oscillate about a stable equilibrium if it is excited above that state. These modes of oscillation may persist long enough to have observable consequences, despite being weakly damped. The modes of oscillation are investigated with a range of evolutionary models. The simplest mode to detect and simulate is the “fundamental” mode or “$l = 0$” mode, which manifests itself as a radial oscillation of the entire system. To investigate this mode a King $W_0 = 1$ theoretical model is adopted. The effect of adopting a softened gravitational potential to generate the forces between stars on the fundamental mode is investigated with a direct method $N$-body evolutionary model, and these results are compared to those which use an SCF (self-consistent field) type code. The presence of small amplitude fundamental mode oscillations is detectable when the perturbation particle method is used, which would otherwise be undetectable for another $N$-body evolutionary model, due to particle noise. Furthermore, the source of heavy damping in direct method $N$-body simulations is found to be phase mixing. The experimental fundamental mode oscillations are found to match well with the theoretically predicted frequencies. A more complicated mode, the “sloshing” mode or “$l = 1$” is also studied. This mode manifests itself in the density centre shifting or sloshing about. To investigate this mode a King $W_0 = 5$ model is adopted, and the results compared to an analytical predicted frequency.
Many galaxies contain central black holes, and in the event that two such galaxies merge, then the two black holes sink towards the centre of mass of the merger remnant due to dynamical friction, and form a binary in the core. The binary black hole evolves because of close encounters, which must be integrated accurately, however the galaxy also evolves because of the collective response of the stars to changes in the potential. To suppress relaxation amongst the field stars and to satisfy the requirement that the stars be much less massive than the black holes, one requires a large number of stars, which in turn will also reduce the particle noise in the results. To this aim, the perturbation particle method is adopted to model the evolution of the two black holes in a galaxy. For comparative purposes, experiments are also performed using a simple analytic dynamical friction formula, and a direct method \(N\)-body solver. The perturbation particle method simulations give results comparable to the direct method simulations.
Acknowledgements

This work was carried out at the University of Edinburgh. The computers employed to perform the experiments included the Cray T3D (housed at the Edinburgh Parallel Computer Centre), the Waverley cluster (housed at the Edinburgh University Computing Services) and an UltraSPARC-IIi workstation (housed in the Maths department, Edinburgh University). This work was supervised by Professor D.C. Heggie.

I would like to thank Douglas Heggie for all his supervision and guidance together with the many meetings and useful discussions throughout this degree. Also I would like to extend my gratitude to John Blair-Fish for many of his useful correspondence during my first year, and to Rainer Spurzem and Sverre Aarseth for the use of their N-body codes. Finally my thanks to Alan Roy and Nikolaos Georgarakos for many useful discussions and their continuing support and friendship.
Declaration

I hereby declare that all the work presented in this thesis has been carried out by myself between October 1996 and August 2000. No part of this work has previously been submitted in support of a degree validated by a University.

Steve Purchase, 2000
Declaration

I declare that this thesis was composed by myself and that the work contained therein is my own, except where explicitly stated otherwise in the text.

(Steve Purchase)
Contents

1 Galaxies and Globular Clusters .................................................. 1
   1.1 Galaxies ............................................................................. 1
   1.2 Globular Clusters ............................................................... 2
   1.3 The origin of Globular Clusters ............................................ 4
   1.4 Early evolution of Globular Clusters ..................................... 5
   1.5 Steady evolution phase of Globular Clusters ......................... 7
       1.5.1 Mass segregation and Energy equipartition ................... 7
       1.5.2 Evaporation ............................................................... 7
   1.6 Secular evolution phase of Globular Clusters ....................... 8
   1.7 Post-collapse evolution and Disruption of Globular Clusters ... 10
   1.8 Evolutionary equations and theoretical models ..................... 10

2 N-body Solvers ........................................................................... 15
   2.1 Introduction ......................................................................... 15
       2.1.1 The N-body Problem ................................................ 15
       2.1.2 N-Body Units ............................................................ 16
   2.2 A simple Direct Method N-body solver ............................... 16
       2.2.1 Introduction ................................................................ 16
       2.2.2 The Hermite Integrator ............................................ 17
       2.2.3 Results and Conclusions ............................................ 19
   2.3 A more advanced Direct Method N-body solver .................... 20
   2.4 The Perturbation Particle Method ....................................... 22
       2.4.1 Introduction ............................................................... 22
       2.4.2 Implementing the Perturbation Particle method with the
              Hermite version of NBODY1 ........................................ 23
   2.5 Experimental Stellar Dynamics for Systems with Axial Symmetry 25
   2.6 The Particle-Mesh Method ................................................... 28
   2.7 The Tree Method .................................................................. 29
   2.8 Hybrid Methods .................................................................. 30
3 Parallelisation of the $N$-body problem

3.1 Introduction .................................................. 32
  3.1.1 The CRAY T3D Supercomputer and the Waverley Cluster  33
  3.1.2 The SIMD languages - Fortran 77 with CRAFT, and High
       Performance Fortran ..................................... 34
  3.1.3 Measuring the effect of parallelisation .................. 38

3.2 Parallelisation of a Direct $N$-body Method .................. 38
  3.2.1 General Considerations .................................. 38
  3.2.2 The Optimal Neighbour Number .......................... 42

3.3 A new strategy for parallelising NBODY6++ ...................... 47
  3.3.1 Introduction ............................................ 47
  3.3.2 An improved timestep strategy .......................... 48
  3.3.3 Results .................................................. 49

3.4 Parallel implementation of NBODY6++ with High Performance
       Fortran ..................................................... 52
  3.4.1 Introduction ............................................ 52
  3.4.2 Implementation .......................................... 52

3.5 The Hypersystolic Method of Parallelisation ................... 57
  3.5.1 Introduction ............................................ 57
  3.5.2 The Standard Systolic Method ........................... 58
  3.5.3 The Hypersystolic Algorithm ............................. 59
  3.5.4 Implementation of the Direct Method $N$-body solver with
       the Hypersystolic algorithm ............................. 61

3.6 Parallelisation of a Tree code ................................ 63

4 Galactic Oscillations ............................................. 66
  4.1 Introduction ................................................ 66
  4.2 The Fundamental Mode ....................................... 67
    4.2.1 Introduction ........................................... 67
    4.2.2 Numerical experiments ................................ 72
    4.2.3 Discussion and conclusions ............................ 80
  4.3 The Sloshing Mode ........................................... 83
    4.3.1 Introduction ........................................... 83
    4.3.2 Numerical Experiments ................................ 83
    4.3.3 Discussion and conclusions ............................ 91
  4.4 Conclusions ................................................ 91
5 Binary Black Holes in Galactic Nuclei

5.1 Introduction ........................................ 93
5.2 Numerical Experiments .............................. 98
  5.2.1 Introduction ..................................... 98
  5.2.2 Dynamical quantities associated with planar bound orbits 99
  5.2.3 Initial conditions and Evolutionary models .......... 100
  5.2.4 Experiments and Results ....................... 102
5.3 Discussion and conclusions .......................... 113
Chapter 1

Galaxies and Globular Clusters

1.1 Galaxies

Galaxies are approximately $10^{10}$ years old and contain between $10^{10}$ to $10^{12}$ stars (Binney & Tremaine 1994). For example, the Milky Way contains $10^{11}$ visible stars, as well as $10^{10} M_\odot$ of gas (where $M_\odot$ is the mass of the Sun). The gas has a minor effect on the dynamics of the galaxy; however, dense gas clouds are the birth sites of new stars and thus play an important role in the long-term evolution of a galaxy. Dark matter is a constituent component of galaxies and also plays an important role in the long-term evolution of a galaxy. In what follows, any stellar system that has low heavy element abundances, no young stars, and little rotation is referred to as a Population II system. For example, the spheroid region of the Milky Way is a Population II system. In contrast, the disk region of the Milky Way is referred to as a Population I system, and contains younger stars which move around the galactic centre in nearly circular orbits.

In the context of the Universe, the Milky Way is one galaxy in over $10^9$ galaxies. Each galaxy may be classified into one of four types according to Hubble’s classification system:

1. Elliptical galaxies - These are Population II galaxies containing little or no dust, and their contours of constant surface brightness (isophotes) are approximately concentric ellipses. Elliptical galaxies are denoted according to the eccentricity of their isophotes; specifically $E0, E1, ..., E7$ correspond to the sequence of elliptical galaxies ranging from those containing circular isophotes to those containing the most elongated isophotes. The fraction of bright galaxies that are elliptical is a function of the local density, ranging from 10% in low-density regions to 40% in dense clusters of
2. Lenticular galaxies - These are smooth, featureless galaxies with a prominent disk that contains no gas, dust, bright young stars, or spiral arms. Although similar to elliptical galaxies, lenticular galaxies obey the exponential surface brightness law characteristic of spiral galaxies. Lenticular galaxies are denoted by S0 and are the transition class between elliptical and spiral galaxies. Up to half of all galaxies in high-density regions are lenticular galaxies.

3. Spiral galaxies - These are galaxies which contain a prominent disk composed of Population I stars, gas and dust, together with spiral arms which in turn contain large numbers of currently forming stars. As with the Milky Way, spiral galaxies also contain a spheroid of Population II stars. These galaxies are denoted by a sequence of four called Sa, Sb, Sc, Sd; along this sequence (from left to right) the relative luminosity of the spheroid decreases, the relative mass of gas increases, and the spiral arms become more loosely wound. Spiral galaxies are dominant (almost 80% of all bright galaxies) in low-density regions of the Universe, but drop to less than 10% in dense regions.

4. Irregular galaxies - These galaxies are typically low-luminosity gas-rich systems which conform to no particular pattern or shape. They are denoted as Irr or Im galaxies.

1.2 Globular Clusters

Galaxies contain smaller stellar systems ranging from binary star systems to globular clusters (see below). Some aspects of the dynamics of a galaxy are also relevant to the dynamics of these smaller stellar systems. However, dark matter has a minor effect on the dynamics of these smaller stellar systems.

A globular cluster is usually described as an old star cluster found in the bulge and halo regions of a galaxy (for a review of the internal dynamics of globular clusters see Meylan & Heggie 1997). These ancient ($\sim 10^{10}$ yr) components of galaxies are Population II systems which represent a very interesting family of stellar systems in which some fundamental dynamical processes (two-body relaxation, mass segregation, evaporation and core collapse) have taken place on time scales shorter than the age of the universe. Most globular
clusters are located within 20kpc from the galactic centre; however, the furthest globular cluster within our galaxy is AM-1 at 120kpc from the galactic centre (Aaronson et al. 1984). The integrated absolute magnitude and total mass of globular clusters vary considerably, ranging from $M_v^{int} = -10.1$ and $M_{tot} = 4 \times 10^6 M_\odot$ (Meylan et al. 1994) for the giant globular cluster ω Centauri down to $M_v^{int} = -1.7$ and $M_{tot} \approx 5 \times 10^3 M_\odot$ for the Lilliputian globular cluster AM-4 (Inman & Carney 1987).

To discuss the dynamical evolution of a globular cluster one considers 3 characteristic time scales (King 1981):

- Crossing time: $t_{cr} = 2R/V$, where $R$ is the characteristic radius of the cluster and $V$ is a measure of the mean stellar velocity (Binney & Tremaine 1994, Chapter 4). This is the time scale on which a star transverses the diameter of the cluster.

- Relaxation time: $t_{rlx} = n_r t_{cr}$, where $n_r$ is the number of crossings a star makes across the cluster until its energy changes by an order of itself, and may be approximated by (Binney & Tremaine 1994, Chapter 4)

$$n_r \approx \frac{N}{8 \ln N}, \quad (1.1)$$

where $N$ is the number of stars in the cluster. Stellar encounters are important within a relaxation time, and a near-Maxwellian velocity distribution is set up. The mechanism for these changes is the cumulative effect of many mild two-body encounters. If one considers the time scale on which the cumulative mean square value of the component of the velocity which is parallel to the velocity itself becomes comparable with the mean square value of one velocity component, then the relaxation time may be defined as (Spitzer 1987)

$$t_{rlx} = \frac{0.065 \langle v^2 \rangle^{3/2}}{\rho \langle m \rangle G^2 \ln \Lambda} \quad (1.2)$$

where $\langle v^2 \rangle$ is the mass-weighted mean square velocity of the stars, $\rho$ is the mass density, $\langle m \rangle$ is the mean stellar mass, $G$ is the gravitational constant, and $\Lambda \approx 0.4 N$. The logarithmic factor is then referred to as the "Coulomb logarithm". These energy exchanges can produce major changes to the cluster structure, without disturbing the dynamical equilibrium significantly.
• Evolution time: \( t_{evol} \). This is defined as the time scale on which the the size and profile of the cluster change significantly, as a result of relaxation.

It follows from these time scale definitions that

\[
t_{cr} \ll t_{rlx} \leq t_{evol}.
\]  

Typically a galaxy has \( N \sim 10^{11} \) stars, \( R \sim 10\, \text{kpc} \), \( V \sim 200\, \text{km} \, \text{s}^{-1} \), \( t_{cr} \sim 10^8 \, \text{yr} \), and is a few hundred crossing times old (\( \sim 10^{10} \, \text{yr} \)). Stellar encounters are unimportant in galaxies since, with the approximation of Eq. (1.1), the relaxation time is of the order \( 10^9 t_{cr} \) (Binney & Tremaine 1994, p190). Thus the fundamental dynamics of galaxies is that of a collisionless system, in which the stars move under the influence of the mean potential generated by the other stars. However, in globular clusters, \( N \sim 10^5 \), \( t_{cr} \sim 10^5 \, \text{yr} \) and \( t_{rlx} \sim 10^3 t_{cr} \), so that stellar encounters may be important over the cluster lifetime of order \( 10^{10} \) years. One may refer to globular clusters as collisional systems. On the other hand, even for globular clusters, relaxation may be neglected for phenomena occurring on sufficiently short timescales.

For the remainder of this chapter globular cluster evolution is discussed; beginning with the origin of globular clusters and the initial mass function, followed by the early stellar evolution phase involving mass loss, violent relaxation and phase mixing. Thereafter the steady evolution phase, the secular evolution phase and the post-collapse evolution phase are discussed. At the end of the chapter we introduce some of the concepts and equations that are used in studying the dynamics of stellar systems.

### 1.3 The origin of Globular Clusters

The origin of globular clusters, although intricately linked to galaxy formation, differs in vital ways from galaxy formation. The standard picture for galaxies is that they form from the gravitational collapse of primordial density fluctuations. However, in contrast, the globular clusters in some low-redshift galaxies still appear to be at the stage of formation. To allow the comparison of globular cluster populations one considers the quantity \( S_N \), the specific frequency (Harris & van den Bergh 1981). This quantity represents the total number of globular clusters per unit \( M_v = -15 \) of the host galaxy luminosity, and thus

\[
S_N = N_t 10^{0.4(M_v+15)} \tag{1.4}
\]
where $N_t$ is the total number of clusters integrated over the entire globular cluster luminosity function, and $M_v$ is the absolute visual magnitude of the galaxy. Typically the specific frequency ranges from $S_N \leq 1$ for spiral and irregular galaxies, to $S_N \approx 2 - 3$ for elliptical galaxies in low-density environments, to $S_N \approx 5 - 6$ for elliptical galaxies in rich clusters. The largest specific frequency values are $S_N \approx 10 - 20$ for cD galaxies located at the centres of rich clusters; for example the galaxy M87 has $S_N = 14$. Thus the specific frequency is not only determined by the galaxy type, but also by the local galactic environment.

Many scenarios have been proposed for the formation of globular clusters (summarised by Meylan & Heggie 1997). For example, it has been hypothesised that globular clusters form during interactions or mergers of galaxies (Zepf & Ashman 1993), and there is increasing evidence to support this (Holtzman et al. 1992, 1996). This would explain why globular cluster numbers are greater in elliptical galaxies than spiral galaxies, since one assumes the former undergo more mergers than the latter. In contrast, another hypothesis suggests that originally cD galaxies had much greater $S_N$ values than at present, since it is thought that cD galaxies have grown by cannibalism and/or mergers and these mergers have reduced the $S_N$ values (van den Bergh 1984). However, none of the scenarios proposed can explain why the faintest galaxy known, the Fornax dwarf spheroidal galaxy with an absolute magnitude $M_v = -12.3$ (Harris 1991), has the highest specific frequency $S_N = 73$.

Determining the initial mass function (IMF) of the stars in a globular cluster is of fundamental importance since it plays a role in the dynamical evolution of the cluster. Unfortunately it is difficult to predict the IMF since a large number of different processes are likely to occur during star formation: for example, cloud fragmentation, fragment coalescence, mass accretion in disks and stellar wind mass loss. One attempt to predict the IMF uses a statistical approach to predict the mass distribution of stars in a cloud, but not the individual mass of a star (Elmegreen 1985). A combination of events will in theory determine the mass of the individual stars. For example, the formation of massive stars may result from combinations of the fragmentation and accretion processes in hierarchical groupings of forming stars (Larson 1982, 1992).

### 1.4 Early evolution of Globular Clusters

The early stellar evolution of a globular cluster is usually modelled by the sudden mass loss of each star at the end of its main sequence evolution (Chernoff &
The resulting mass loss of the cluster leads to an expansion of the cluster which in turn may delay the onset of relaxation processes or expose the cluster to the danger of disruption by tidal shocking. Experiments which considered a steady tide and modelled relaxation by using a Fokker-Planck code showed the mass loss of clusters with a relatively flat IMF resulted in the disruption of the cluster, and clusters with steeper IMF's would survive disruption provided that the initial concentration was high enough (Chernoff & Weinberg 1990). By using a more accurate N-body evolutionary model (see Chapter 2) and under the assumption that the timescale on which the mass lost by the cluster is not long compared to the crossing time (see section 1.2), then one finds that the destruction times predicted by Chernoff and Weinberg are underestimated by factors as large as 10 (Fukushige & Heggie 1995).

Within the early evolution phase two collisionless relaxation processes are present, specifically phase mixing and violent relaxation; these are described briefly in what follows. The evolutionary equation in a collisionless system is the Collisionless Boltzmann Equation (see section 1.8, Eq. (1.6)) which involves the fine-grained distribution function $f$, and states that the phase-space density around a star remains constant i.e. one cannot compress the phase fluid. In contrast, one may define the course-grained distribution function $\tilde{f}$, whose value at any phase point $\omega = (r, v)$ is the average value of $f$ in some specified small volume centred on $\omega$. The process which causes $\tilde{f}$ to decrease as the stars move out of phase is called phase mixing (Binney & Tremaine 1994). This process is important during the relaxation of galaxies to equilibrium configurations, and although $\tilde{f}$ is no longer necessarily constant along stellar orbits, it does ensure that $\tilde{f}$ does not increase along a stellar orbit. The course-grained phase-space density can always be diminished if the incompressible fluid of phase points is mixed with unpopulated regions of phase-space. Furthermore, the entropy $\tilde{S}$ based on the course-grained distribution function increases as $\tilde{f}$ decreases along orbits (Tremaine et al. 1986).

Violent changes in newly formed globular clusters may have resulted from large amounts of irregular mass loss. Consequently, the statistics of stellar orbits are changed on the timescale of the order of the crossing time. This encounterless relaxation phenomenon has been named violent relaxation (Lynden-Bell 1962) where the energies of the stars themselves are changed, as opposed to phase mixing, which changes the course-grained phase-space density near the phase point of each star without necessarily causing any energy change. Violent relaxation changes energy per unit mass of a star in a way that is independent
of the star's mass. Collisional relaxation in contrast tends to transfer energy from the most massive particles to the least massive particles, thus leading to equipartition of energy (see section 1.5). Violent relaxation and phase mixing subsequently lead to a smooth density distribution characterised by a steady dynamical evolution driven by relaxation due to stellar encounters.

1.5 Steady evolution phase of Globular Clusters

In this section the steady evolution (or quasi-equilibrium) phase is discussed, and we introduce a number of processes which occur over the timescale of the order of a relaxation time; including dynamical processes such as two-body relaxation, mass segregation, energy equipartition and evaporation.

1.5.1 Mass segregation and Energy equipartition

It is usually assumed that the early processes of mass loss from stellar evolution and violent relaxation do not affect the spatial distribution and dynamical evolution of stars in a way that depends on stellar mass, with the exception of the progressive loss of the more massive stars as they evolve internally. However, during the subsequent steady dynamical evolution (or quasi-equilibrium) phase, collisional relaxation is the first dynamical process to differentiate between different stellar masses. During a two-body gravitational encounter the larger mass involved tends to lose kinetic energy to the lighter mass and fall deeper into the potential well of the cluster, thus leading to an equipartition of kinetic energy. Simultaneously, one has mass segregation since the lighter stars are driven outwards. More generally there is a fairly rapid phase of evolution where mass segregation occurs in the central regions of clusters, and is more prominent amongst the heavier masses. The early phases of core collapse (Chernoff & Weinberg 1990) have been modelled dynamically by multimass King models (King et al. 1995, cf. section 1.8) to verify that the observed amount of mass segregation of globular cluster NGC6397 is in agreement with dynamical predictions.

1.5.2 Evaporation

Evaporation of globular clusters due to escaping stars has been studied a great deal theoretically, including theories of the escape rate based on relaxation
phenomena; these yield a fractional rate of escapes proportional to an inverse relaxation time (e.g. Johnstone 1993). Another theory is based on individual two-body encounters, and leads to a similar result except for a numerical factor and, more significantly, no Coulomb logarithm (Hénon 1960, Woolley & Dickens 1962, cf. Eq. (1.2)). This theory has been extended to unequal stellar masses based on a Plummer model (Hénon 1969). To investigate the escape rate without simplifying assumptions one may adopt numerical modelling, which also allows the effects of mass segregation on escape rates to be studied. Fokker-Planck models (cf. section 1.8) reveal, for isolated systems with equal stellar masses, that escape rates increase during core collapse (Spitzer & Shull 1975). This is due to both the increasing concentration of the core and the growth of anisotropy (Giersz & Heggie 1994). Furthermore, in systems with unequal stellar masses, the overall escape rate is enhanced by as much as a factor of 30 (Wielen 1974, 1975). It is relatively easy, during an impact, to transfer kinetic energy from a massive star to a low-mass star, and consequently escape rates are mass dependent. Thus, the fraction of massive stars which escape in a specified time is much less than the fraction of low-mass stars. Introducing a tide greatly increases the escape rate for systems with equal stellar masses, by about an order of magnitude for Fokker-Planck models (Spitzer & Chevalier 1973). For systems with unequal stellar masses the introduction of a tide enhances the escape rates of low mass stars, as one might expect, since mass segregation has already placed these stars at large radii. Furthermore, the escape rates of low mass stars do not depend significantly on mass (Chernoff & Weinberg 1990, Giersz & Heggie 1996). Inclusion of stellar evolution in tidally bound systems enhances escape rates of low mass stars (Chernoff & Weinberg 1990), and tidal shocking will also have an effect (Weinberg 1994a).

1.6 Secular evolution phase of Globular Clusters

In this section a very brief overview of the secular evolution phase is presented, but one should refer to Meylan & Heggie (1997) for a more detailed review, and Binney & Tremaine (1994) for a more mathematical slant. During this evolution phase the long-term effects of two-body relaxation are considered, i.e. phenomena occurring on the time scale of several relaxation times. As a result of two-body relaxation stars may escape the inner parts of
a cluster to larger radii, not necessarily escaping the cluster. This process may lead to "gravothermal instability" whereby the core temperature increases at a rate greater than the halo regions. A manifestation of this instability is "core-collapse"; where the radius of the core shrinks until it reaches a singularity (the gravothermal catastrophe). Numerical experiments have been performed to determine the time scale for core-collapse, which ranges from $0.9t_{rh}$ (in terms of the initial half-mass relaxation time (Cohn 1980)) for a tidally limited cluster with unequal masses (Chernoff & Weinberg 1990) to $17.6t_{rh}$ for a Plummer model with equal stellar masses and an anisotropic velocity distribution (Takahashi 1995).

It was once thought that evaporation was the dominant process in the secular evolution phase (Spitzer 1940), and eventually, after a certain time, the entire cluster would have evaporated. However, with the discovery of the gravothermal instability (Antonov 1962), together with the knowledge that evaporation has a time scale of the order of $100t_{rh}$ (Spitzer & Thuan 1972), one may conclude that core-collapse by gravothermal effects is the dominant process in secular evolution and not evaporation as was once thought. In fact it appears that core-collapse is a two-stage process. Initially, if different stellar masses are present, encounters in the core tend to establish an equipartition in kinetic energy (see section 1.5.1), causing the heavier stars to lose kinetic energy and sink towards the centre of the cluster whilst the lighter stars gain kinetic energy and expand their orbits, thus enhancing evaporation. After a few half mass relaxation times, the second process (i.e. gravothermal instability) intervenes and the core-collapse accelerates. Energy is lost from the centre and transferred outwards by star-star encounters, and as a consequence the core shrinks and heats up.

The process of core-collapse may not continue indefinitely, as infinite central density would be reached. The presence of binary stars is the ingredient which avoids the singular end point of the gravothermal catastrophe (when the central density goes to infinity) and consequently halts core-collapse (Hills 1975). Primordial binaries, binaries formed by three-body interactions (Heggie & Hut 1993) and binaries formed by the two-body formation mechanism (Lee & Ostriker 1986) are important because of the energy which can be imparted to single stars or other binaries during an interaction (Heggie's law). Furthermore binary stars tend to sink towards the core (the effect of mass segregation) and consequently binary-binary interactions occur relatively more frequently there than in halo regions. Binary-binary interactions are also energetically more domi-
nant than three-body interactions since the cross-sectional area for interactions is larger in the former than the latter (Meylan & Heggie 1997).

1.7 Post-collapse evolution and Disruption of Globular Clusters

Since the ages of many globular clusters exceed \(20t_r h\), some have probably undergone core collapse already. The behaviour of a star cluster after collapse is dependent on the energy flowing out of the central parts of a cluster. The source of this energy generation may again be primordial binaries (Goodman & Hut 1989), binaries formed by three-body encounters, or a massive central black hole, or mass loss from the evolution of merger products. The outpouring of energy leads to an overall expansion in the system (Hénon 1965), and if the system is tidally limited then the mass is lost nearly linearly with time (Hénon 1961).

The presence of binaries formed in three-body encounters in the core, and the emission of energy from their evolution, can cause a drop in core temperature and drive gravothermal instability in reverse; thus leading to core expansion. Thereafter the core exhibits a complicated succession of collapses and expansions called "gravothermal oscillations" (Sugimoto & Bettwieser 1983, Makino 1996). However the presence of a mass spectrum (Murphy et al. 1990) or the presence of primordial binaries can suppress gravothermal oscillations.

The lifetime of clusters is determined by several disruptive processes. These disruptive processes include both internal processes such as evaporation and mass loss associated with early cluster evolution (Chernoff & Shapiro 1987, Fukushige & Heggie 1995), and external processes such as time-dependent tidal fields and dynamical friction (Tremaine 1976).

1.8 Evolutionary equations and theoretical models

In this section we discuss one of the fundamental evolutionary equations which can be used to describe the various phases referred to in previous sections. We also present a derivation of some theoretical models for globular clusters which will be used in later chapters.

The Collisional Boltzmann Equation describes the general evolution of a
stellar system, and is
\[
\frac{\partial f}{\partial t} + v \cdot \frac{\partial f}{\partial \mathbf{r}} - \frac{\partial \phi}{\partial \mathbf{r}} \cdot \frac{\partial f}{\partial \mathbf{v}} = \left( \frac{\partial f}{\partial t} \right)_{enc}
\]  
where \( f = f(w, t) \) is the time-dependent distribution function (henceforth denoted by DF), \( w \) represents the 6-dimensional phase space coordinates, and \( \phi(r, t) \) is the gravitational potential. The term \( \left( \frac{\partial f}{\partial t} \right)_{enc} \) stands for the Boltzmann “collision” integral which describes the two-body relaxation effects of the interactions that the test star has with the other stars (involving \( f \), derivatives of \( f \) and integrals over \( f \)), and may be approximated by the Fokker-Planck collision term (Binney & Tremaine 1994, Chapter 8). When the relaxation time is large compared to the age of the system one can ignore the \( \left( \frac{\partial f}{\partial t} \right)_{enc} \) term and one gets the Collisionless Boltzmann Equation (henceforth known as the CBE)
\[
\frac{\partial f}{\partial t} + v \cdot \frac{\partial f}{\partial \mathbf{r}} - \frac{\partial \phi}{\partial \mathbf{r}} \cdot \frac{\partial f}{\partial \mathbf{v}} = 0.
\]  
A derivation can be found in Binney & Tremaine (1994), Chapter 4. In words, the flow of stellar phase points through phase space is incompressible, so that the phase-space density around a star remains constant, and thus \( \frac{\partial f}{\partial t} = 0 \). Furthermore, because \( t_{cr} \) is so much the shortest time scale of interest, after a few orbital times we often assume that the stars are mixed into a nearly stationary distribution (see sections 1.4 and 1.5) and one may take \( \frac{\partial f}{\partial t} \) to be zero. The resulting equation for a collisionless system in equilibrium is
\[
\frac{v}{\mathbf{r}} \cdot \frac{\partial f}{\partial \mathbf{r}} - \frac{\partial \phi}{\partial \mathbf{r}} \cdot \frac{\partial f}{\partial \mathbf{v}} = 0.
\]  
By Jeans’s Theorem (Jeans 1915) a fairly general solution of this equation for a spherical system is
\[
f(r, v) = f(E, J)
\]  
where \( J \) is the angular momentum vector and \( E \) is the energy of a particle, given by
\[
E = \frac{v^2}{2} + \phi(r),
\]  
where \( v = |v| \) and \( r = |r| \). Together \( J \) and \( E \) represent the four isolating integrals. The position-space density in this context is the integral of the DF over the velocity coordinates of the phase space upto escape velocity, and is given by
\[
\rho(r) = \int_0^{v_e} f(r, v) d^3v
\]
where \( v_e \) is the escape velocity of a star at any point and is given by

\[
v_e = \sqrt{-2\phi(r)}
\]  

(assuming that the system is isolated and \( \phi \to 0 \) as \( r \to \infty \)). Since \( \rho \) is a function of the potential \( \phi \), one can now derive \( \phi \) by solving Poisson’s equation,

\[
\nabla^2 \phi = 4\pi G\rho.
\]  

Many different models may be constructed from Eq.’s (1.10) and (1.12), but in this section and for the experiments performed in this thesis, we shall describe two theoretical models which have an isotropic velocity distribution and a DF which depends only on the energy \( E \).

The Plummer model is used frequently by theorists for its analytical convenience, but now is not used in fitting observed clusters. To construct the DF for the Plummer model consider the general polytropic DF (Binney & Tremaine 1994, Chapter 4),

\[
f(E) = \begin{cases} 
A(-E)^{n-3/2} , & E < 0 \\
0 , & E \geq 0
\end{cases}
\]  

(1.13)

The density from Eq. (1.10) is

\[
\rho(r) = c_n(-\phi(r))^n
\]  

(1.14)

where

\[
c_n = \frac{A(2\pi)^{3/2}(n - \frac{3}{2})!}{n!}
\]  

(1.15)

and \( n > \frac{1}{2} \) for \( c_n \) to be finite. Poisson’s equation in spherical coordinates can be rewritten as the Lane-Emden equation,

\[
\frac{1}{s^2} \frac{d}{ds} \left( s^2 \frac{d\psi}{ds} \right) = -\psi^n
\]  

(1.16)

where \( \psi = \phi(r)/\phi_0 \), \( \phi_0 = \phi(0) \), \( s = r/a \), and \( a = \left(4\pi G(-\phi_0)^{n-1}c_n\right)^{-1/2} \), with boundary conditions \( \psi = 1 \) and \( \frac{d\psi}{ds} = 0 \) at \( s = 0 \).

Consider the case when the polytropic index is \( n = 5 \). It can easily be shown that \( \psi = (1 + \frac{s^2}{3})^{-\frac{1}{5}} \) satisfies the Lane-Emden equation, and thus the density function is given by

\[
\rho = c_5(-\phi_0)^5 \left(1 + \frac{s^2}{3}\right)^{-5/2}
\]  

(1.17)
which is known as Plummer's Law. Specifically the Plummer DF is

\[ f(r, v) = A \left( -\phi(r) - \frac{v^2}{2} \right)^{7/2} \]

when \( E < 0 \), where \( A \) is a constant and

\[ \phi(r) = -\frac{MG}{(r^2 + b^2)^{1/2}}. \]

Here \( M \) is the total mass of the system, \( G \) is the gravitational constant and \( b \) is the scale length of the model. In N-body units one scales the system so that \( M = 1, G = 1 \), \( E = -\frac{1}{4} \), and then \( b = \frac{3\pi}{16} \). It follows from Eq. (1.12) that

\[ \rho(r) = \frac{3Mb^2}{4\pi(r^2 + b^2)^{5/2}}. \]

The second of the two theoretical models considered is the King model (King 1966), which presents a more realistic model of a cluster. To derive the DF for a King model one allows the DF at the center of the cluster, where relaxation is fastest, to be

\[ f(0, v) = k(e^{-j^2v^2} - e^{-j^2\nu^2}), \ v < \nu_e \]

where \( k \) and \( j^2 \) are constants, and there is a cut-off in the velocities determined by \( \nu_e \); this velocity DF is approximately Maxwellian. One takes the DF to be zero for \( E > 0 \) (here the density is taken to vanish at a certain finite radius), since stars with positive energies are assumed to be removed by tidal forces, and the DF at the center can be rewritten in terms of energy as

\[ f(0, v) = k e^{2j^2\phi_0} (e^{-2j^2E} - 1). \]

According to Jeans' Theorem, the DF must be the same function of energy at all points. Thus,

\[ f(r, v) = k e^{2j^2\phi_0} (e^{-2j^2(\phi + v^2/2)} - 1). \]

The density at any point can now be found by Eq. (1.10) and is thus given by

\[ \rho(W) = \frac{4\pi k e^{-W_0}}{3j^3} \left( -\sqrt{W} \left( W + \frac{3}{2} \right) + \frac{3}{4} \sqrt{\pi} e^W \text{erf}(\sqrt{W}) \right) \]

where the King potential is

\[ W = -2j^2\phi, \]
and the error function is defined as
\[
\text{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt.
\] (1.26)

The King potential function is a solution of Poisson’s equation, and thus in spherical coordinates satisfies
\[
\frac{d^2 W}{dR_k^2} + \frac{2}{R_k} \frac{dW}{dR_k} = -8\pi G j^2 r_c^2 \rho(W)
\] (1.27)
where the dimensionless radius
\[
R_k = r/r_c.
\] (1.28)

The substitution of a power series for \( W \) in terms of \( R_k \) into Eq. (1.27) fixes the central value of \( \nabla^2 W \) to be \(-9\) if the scale factor \( r_c \) is to be very close to the “core radius” (King 1958). That is to say, \( 8\pi G j^2 r_c^2 \rho_0 = 9 \), where \( \rho_0 \) is the central density. Thus, Eq. (1.27) can be rewritten as
\[
\frac{d^2 W}{dR_k^2} + \frac{2}{R_k} \frac{dW}{dR_k} = -\frac{9 \rho}{\rho_0}
\] (1.29)
subject to the boundary conditions \( W = W_0 \) and \( \frac{dW}{dR_k} = 0 \) at \( R_k = 0 \), and may be solved numerically as follows:

1. Since \( \rho(W) \) is available for any value of \( W \), the differential equation (1.29) is transformed so as to make \( R_k \) the dependent variable and \( W \) the independent variable. Thus, Eq. (1.29) becomes
\[
-X \frac{d^2 X}{dW^2} + \frac{3}{2} \left( \frac{dX}{dW} \right)^2 = -\frac{9 \rho(W)}{4\rho_0} \left( \frac{dX}{dW} \right)^3
\] (1.30)
where \( X = R_k^2 \).

2. This new differential equation is solved numerically at \( m_k \) discrete points of \( W \) (ranging from \( W = W_0 \) to \( W = 0 \)) by a 4th order Runge-Kutta integrator (henceforth known as RK4, see Press et al. 1996, Chapter 16).\(^1\) In order to solve Eq. (1.30) using RK4 one needs to know the central values of the first and second derivatives of \( X \) with respect to \( W \), and these are found to be
\[
\lim_{X \to 0} \frac{dX}{dW} = -\frac{2}{3},
\] (1.31)
\[
\lim_{X \to 0} \frac{d^2 X}{dW^2} = \frac{2}{5} \left( 1 + \frac{4\pi ke^{-W_0 W_0^{3/2}}}{3j^3 \rho_0} \right).
\] (1.32)

\(^1\) To obtain values of \( W \) between the \( m_k \) discrete points cubic spline interpolation is adopted (see Press et al. 1996, Chapter 3).
Chapter 2

$N$-body Solvers

2.1 Introduction

In this chapter the $N$-body problem is considered in the context of globular clusters and galaxies. Many ways of solving this problem numerically have been developed, and are discussed in this chapter in the form of different $N$-body solvers.

2.1.1 The $N$-body Problem

The dynamical behaviour of a stellar system is determined by Newton's laws of motion and Newton's law of gravity. For a stellar system of $N$ stars, a star's motion is determined by the force exerted on it by the other $(N-1)$ stars, as prescribed by Newton's second law. Thus, for star $i$, its motion is determined by solving

$$\frac{d^2r_i}{dt^2} = -G \sum_{j=1, j\neq i}^{N} \frac{m_j (r_i - r_j)}{|r_i - r_j|^3},$$

where $r_i$ is the position vector of the $i^{th}$ particle and $m_i$ the corresponding mass.

The problem of determining the behaviour of $N$ point masses when they are moving under their mutual gravitational forces is called the gravitational $N$-body problem. The $N$-body problem has proved to be a formidable challenge to the numerical analyst, and many methods have been adopted in order to provide a solution to this problem, some of which are now discussed after suitable $N$-body units have been defined.
2.1.2 N-Body Units

For the experiments performed and subsequently described in this thesis N-body units have been adopted; where the phase-space coordinates have been scaled such that the total energy of the system is \( E = -\frac{1}{4} \), the total mass of the system is \( M = 1 \), and the gravitational constant is \( G = 1 \) (Heggie & Mathieu 1986).

2.2 A simple Direct Method N-body solver

2.2.1 Introduction

The direct method solves the N-body problem by computing the right side of Eq. (2.1) directly, and integrating the second order differential equation using a suitable integrator. The amount of work necessary for computing pairwise interactions in a system of \( N \) particles for the direct method is \( O(N^2) \).

Relaxation may be neglected for phenomena occurring on sufficiently short timescales or when \( N \) is sufficiently large. In particular, close encounters are then unimportant and it is possible to evolve the phase space trajectories of the particles by considering a “softened” gravitational potential (first introduced by Aarseth 1963). Thus, for a star \( i \), its motion is determined by a “softened” version of Newton's law, i.e.

\[
\frac{d^2r_i}{dt^2} = -G \sum_{j=1, j \neq i}^{N} \frac{m_j(r_i - r_j)}{(\epsilon^2 + |r_i - r_j|^2)^{3/2}},
\]

where the parameter \( \epsilon \) is often called the softening parameter. One purpose of this softening parameter is to avoid extremely small time steps (as would otherwise be required for close encounters). For example, if \( \epsilon = 0 \) then the force between two point masses becomes extremely large at small separations, and consequently, the velocities of these point masses may change rapidly, and extremely small time steps must be taken if the motion is to be followed accurately. One way of interpreting the softening parameter is that the particles are now not to be thought of as stars, but rather as bunches of stars that are located in the same small volume of space and have very similar velocities. Incidentally this softened potential is the same as the potential of a Plummer distribution (see Chapter 1.8) where the softening parameter is related to the half-mass radius by \( R_h \approx 1.3\epsilon \). For a homogenous spherical system of radius \( R \),
one might take the softening parameter to be of order the average inter-particle distance and thus

\[ \epsilon = \frac{R}{N^{1/3}}. \]  

(2.3)

Alternatively, from considerations based on a homogeneous system in virial equilibrium, where a close encounter, \( r_{cd} = \frac{4R}{N} \), between two particles leads to a 90 degree deflection, the softening parameter might be taken to be

\[ \epsilon = \frac{4R}{N}. \]  

(2.4)

A general discussion of these considerations is given by several authors (White 1976, Gerhard 1981, and Governato et al. 1991). Numerical simulations are restricted to \( 10^3 \) to \( 10^7 \) particles depending on the required accuracy and adopted evolutionary model. However, in real galaxies a typical number of stars is \( 10^{11} \).

Thus, without softening, the artificial relaxation arising in restricted \( N \)-body simulations is much stronger (by several orders of magnitude) than the relaxation in the real \( N \)-body systems. Introducing the softening parameter in the restricted \( N \)-body system weakens relaxation; furthermore, the relaxation rates of two-body relaxation in softened potentials has been investigated by Theis (1997).

### 2.2.2 The Hermite Integrator

The \( N \)-body code described here is denoted by NBODY1. This relatively simple code solves Eq. (2.2) by employing a suitable integrator. Older versions of NBODY1 adopted Aarseth’s 4th order force polynomial method or equivalently the four-step 4th order Adams-Bashforth-Moulton integrator (henceforth known as ABMI) to evolve the particles forward in time. However, for the relevant experiments performed and detailed in this thesis, this integrator has been improved by replacing it with the one-step 4th order Hermite integrator (henceforth known as HI). A comparison of the HI scheme and ABMI scheme has been discussed by Makino & Aarseth (1992), and it has been found that the HI scheme allows time steps which are on average twice as large as the ABMI scheme for the same accuracy, together with a reduction in storage requirements.

In the HI scheme each particle has its own time \( t_i \) (of last update), timestep \( \delta t_i \), position \( \mathbf{r}_i \), velocity \( \mathbf{v}_i \), acceleration \( \mathbf{a}_i \) and acceleration time derivative \( \dot{\mathbf{a}}_i \). The HI scheme is described algorithmically as follows:
1. Find next particle, particle \(i\), to integrate forward in time from time \(t_i\) to time \(t = t_i + \delta t_i\).

2. Predict positions and velocities of all particles up to first derivative in acceleration (known as the particle prediction loop):

\[
\delta t_j = t - t_j, \\
S_{p,j} = \frac{\delta t_j^3}{6} \dddot{a}_j + \frac{\delta t_j^2}{2} \ddot{a}_j + \delta t_j \dot{a}_j + \ddot{a}_j, \\
\ddot{v}_{p,j} = \frac{\delta t_j^2}{2} \dddot{a}_j + \delta t_j \ddot{a}_j + \dot{v}_j
\]

where \(j = 1, ..., N\).

3. Calculate the acceleration and its time derivative for particle \(i\) at time \(t\) (known as the force computation loop):

\[
a_{1,i} = G \sum_{j=1}^{N} \frac{m_j r_{ij}}{R_{ij}^3}, \\
\ddot{a}_{1,i} = G \sum_{j=1}^{N} \left( \frac{m_j v_{ij}}{R_{ij}^3} - 3 \frac{m_j \dot{r}_{ij} (\dot{r}_{ij} \cdot \dot{r}_{ij})}{R_{ij}^5} \right)
\]

where \(R_{ij}^2 = |r_{p,j} - r_{p,i}|^2 + \epsilon^2\), \(\dot{r}_{ij} = r_{p,j} - r_{p,i}\) and \(v_{ij} = v_{p,j} - v_{p,i}\).

4. Calculate second and third time derivative for particle \(i\) at time \(t_i\):

\[
a_{0,i}^{(2)} = -\frac{6(a_{0,i} - a_{1,i}) - \delta t_i (4 \ddot{a}_{0,i} + 2 \dddot{a}_{1,i})}{\delta t_i^2}, \\
\dddot{a}_{0,i} = \frac{12(a_{0,i} - a_{1,i}) + 6 \delta t_i (\dddot{a}_{0,i} + \dddot{a}_{1,i})}{\delta t_i^3}
\]

where the subscript 0 denotes an evaluation at time \(t_i\) and the subscript 1 denotes an evaluation at the new time \(t\).

5. Add corrections to the new position and velocity of particle \(i\):

\[
\ddot{r}_{i}(t) = r_{p,i} + \frac{\delta t_i^4}{24} a_{0,i}^{(2)} + \frac{\delta t_i^5}{120} a_{0,i}^{(3)}; \\
\dddot{v}_{i}(t) = v_{p,i} + \frac{\delta t_i^3}{6} a_{0,i}^{(2)} + \frac{\delta t_i^4}{24} a_{0,i}^{(3)}.
\]

6. Calculate new timestep for particle \(i\), based on Aarseth’s criterion (Aarseth 1985):

\[
\delta t_i = \sqrt{\frac{|a_{0,i}| |a_{0,i}^{(2)}| + |a_{0,i}^{(3)}|^2}{\eta (|\dot{a}_{0,i}| |a_{0,i}^{(2)}| + |a_{0,i}^{(3)}|^2)^2}}
\]

18
where \( \eta \) is the accuracy parameter (see Chapter 2.3, Eq. (2.16)).

7. If termination time is not exceeded or an output is not required then goto 1.

2.2.3 Results and Conclusions

In order to test the implementation of the HI scheme, NBODY1 was executed on a 167MHz 64 bit UltraSparc processor for 1024 particles, with \( \epsilon = 0.1 \) and for 6 crossing times (see Chapter 1.2). For comparative purposes, the ABMI version of NBODY1 was executed similarly.

<table>
<thead>
<tr>
<th>Table 2.1: Executing NBODY1 with the ABMI scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \eta )</td>
</tr>
<tr>
<td>----------</td>
</tr>
<tr>
<td>0.01</td>
</tr>
<tr>
<td>0.02</td>
</tr>
</tbody>
</table>

Table 2.2: Executing NBODY1 with the HI scheme

<table>
<thead>
<tr>
<th>( \eta )</th>
<th>Number of Integration Steps</th>
<th>Relative Energy Error</th>
<th>Average Timestep per particle (( t_a ))</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>969851</td>
<td>1.48 \times 10^{-6}</td>
<td>0.00633</td>
</tr>
<tr>
<td>0.02</td>
<td>679312</td>
<td>8.69 \times 10^{-6}</td>
<td>0.00904</td>
</tr>
<tr>
<td>0.04</td>
<td>473913</td>
<td>3.40 \times 10^{-5}</td>
<td>0.01296</td>
</tr>
<tr>
<td>0.06</td>
<td>383236</td>
<td>2.24 \times 10^{-5}</td>
<td>0.01603</td>
</tr>
<tr>
<td>0.08</td>
<td>328982</td>
<td>1.10 \times 10^{-4}</td>
<td>0.01868</td>
</tr>
</tbody>
</table>

Makino and Aarseth found that the energy error of the HI scheme was about 1/10th that of the ABMI (Adams-Bashford-Moulton Integrator) for the same number of timesteps. This appears to be confirmed by the simulations performed to compare the HI scheme with the ABMI scheme (Tables 2.1 and 2.2).

In terms of work, the cost of executing the ABMI scheme and HI scheme for a single timestep is approximately 45\( N \) and 65\( N \) flops respectively. In conclusion the HI scheme is a more efficient scheme than the ABMI scheme. More precisely, for the same level of accuracy, the HI scheme executes approximately 40% faster than the ABMI scheme. Furthermore, the HI scheme is much simpler to implement and requires less computer memory than the ABMI scheme.

It is not entirely clear what the order of the truncation error of the HI scheme is. One knows that the integration is correct up to \( O(\delta t^5) \) in position and \( O(\delta t^4) \) in velocity, where \( \delta t \) is a typical timestep. Therefore one would
expect the relative energy error (which is proportional to the truncation error associated with the HI scheme) to be proportional to $t_a^\alpha$, where $t_a$ is the average timestep for a typical particle and $4 < \alpha < 5$. By performing a least squares approximation to the relevant data in Table 2.2 for $\eta = 0.01, 0.02, 0.04$ one finds that

$$\epsilon_E \approx 6539.2 t_a^{0.37},$$

where $\epsilon_E$ is the relative error in the energy. As expected the truncation error is $O(\delta t^\alpha)$ where $4 < \alpha < 5$.

### 2.3 A more advanced Direct Method $N$-body solver

A more sophisticated code than NBODY1 has been written (Aarseth 1985) to perform $N$-body simulations. As described in Aarseth’s paper this code incorporates:

- high-order force polynomials
- individual time steps
- a neighbour scheme
- close encounter treatments

and is denoted by NBODY5. The force on each particle is represented by a fourth order polynomial, and the solutions (the position and velocity of each star) are obtained by stepwise integration. (This integrator has been changed since 1985 to the Hermite integrator, see Chapters 2.2 and 3.2, and subsequently this version of the code is known as NBODY6).

Individual time steps permit each particle to take the largest time step consistent with accuracy, and consequently the system of particles is evolved in the minimum number of integration steps. Aarseth uses the following time step criterion,

$$\delta t_i = \sqrt{\frac{\eta |a_i| (|a_i^{(2)}| + |\dot{a}_i|^2)}{|\dot{a}_i| (|a_i^{(3)}| + |a_i^{(2)}|^2)}}$$

(2.16)

where $\eta$ is a dimensionless accuracy parameter (typically $\eta = 0.02$), and $a_i$ is the $i^{th}$ particle’s acceleration. This time step criterion was adopted after some
experimentation and ensures that all known acceleration derivatives play a role in determining the time step.

The neighbour scheme, known as the Ahmad-Cohen scheme (Ahmad & Cohen 1973), henceforth AC, has been introduced to reduce the computational cost of calculating the total force on a particle. The total force acting on a particle is represented by a sum of 2 terms. Thus, only the local interactions are obtained by summation at each step (irregular force contributions), whereas the more distant contributions are added by prediction, using a polynomial that is updated less frequently (regular force contributions).

Experiments with three-body systems have shown that such a system is very sensitive to very small changes in initial conditions, resulting in a near-exponential growth of deviations between neighbouring trajectories in the system’s 18-dimensional global phase space (Dejonghe & Hut 1986). In general, $N$-body simulations are particularly sensitive to truncation errors, which propagate exponentially even if the subsequent integration is exact (Heggie 1988, Goodman et al. 1993). Furthermore, numerical errors are particularly acute when dealing with close encounters such as close binaries, since the denominator of Eq. (2.1) can become very small and thus subject to large rounding error when represented computationally. Aarseth(1985) takes special care when dealing with close encounters in order to reduce numerical errors. To have an accurate and economical treatment of close encounters one transforms the nearly singular differential Eq. (2.1) to a new regular differential equation. In particular, close encounters are treated in relative KS two-body regularised coordinates (Kustaanheimo & Stiefel 1965). This involves transforming the set of relative coordinates of the two bodies to a set of four regularised coordinates $u_j$, together with a fictitious time $\tau$, obeying the fundamental relations

$$u_1^2 + u_2^2 + u_3^2 + u_4^2 = R,$$

$$dt = Rd\tau,$$  \hspace{1cm} (2.17)

where $R$ is the separation of the two mass points. A KS transformation removes the singularity (when $R \to 0$) from Eq. (2.1) and thus makes the numerical problem regular. Alternative regularisations of the two-body problem have been investigated, one of which considers Kepler and perturbed Kepler motions (Burdet 1967).

Aarseth also discusses three-body regularisation by introducing two simultaneous KS transformations (Aarseth & Zare 1974). In a more recent paper a chain regularisation method is described for the few body problem (Mikkola
& Aarseth 1990), whereby a chain of inter-particle vectors is chosen in such a way that the critical interactions requiring regularisation are included in the chain. The equations of motion for the chain vectors are subsequently regularised using KS variables and a time transformation. A global regularisation of the gravitational $N$-body problem has been achieved whereby all $N(N-1)/2$ interactions are treated simultaneously (Heggie 1974). So far Heggie's method has mainly been applied to binary-binary scattering. Furthermore, in Mikkola & Aarseth's chain regularisation method an increase in degrees of freedom is allowed, as shown by Heggie (1974, appendix), and thus all possible branching and looping chains may be used. As a consequence Heggie's global method may be recovered.

2.4 The Perturbation Particle Method

2.4.1 Introduction

Conventional $N$-body simulations are not well suited to the study of perturbed stellar systems because the perturbations under study have to compete with non-negligible particle noise (caused by Monte Carlo sampling of the total distribution function). This problem may be overcome if one uses all available particles (perturbation particles) to represent the difference between the perturbed system and the underlying equilibrium system.

Thus in principle each perturbation particle can have negligible mass (provided the perturbation is small enough), and consequently the particle noise is much reduced. This perturbation particle method (henceforth known as the PPM) may be of great use for studies of fine structure in galaxies, such as spirals and rings in disc galaxies, shells in elliptic galaxies and the effects of black holes in galactic nuclei.

The PPM was originally devised by G.Rybicki in 1972, and has been reformulated more recently in a paper by Leeuwin et al. (1993). They show how the PPM can be employed with any standard direct method $N$-body code.

As a result of the CBE (Eq. (1.6)) one has $\frac{df}{dt} = 0$, and thus

$$f(w(t), t) = f(w(0), 0),$$

(2.18)

where $w$ is the 6-dimensional phase space coordinates. That is to say, $f$ is constant along phase-space trajectories. Furthermore, if $d^6w$ is an element of phase-space volume then the quantity $f d^6w$ is also conserved.
Now let $f_0$ be an equilibrium solution of the CBE, and let $f = f_0 + f_1$, where $f_1$ is a perturbation in $f_0$. It will prove useful to normalise $f$ and $f_0$ such that their integral over all phase space is unity. Furthermore, it follows that

$$f_1(w(t), t) = f(w(0), 0) - f_0(w(t), t), \quad (2.19)$$

and so $f_1$ is readily available once the phase-space trajectories are known. To determine these trajectories one requires the perturbed potential

$$\phi_1(\mathbf{r}, t) = \phi(\mathbf{r}, t) - \phi_0(\mathbf{r}, t) = -GM_0 \int \frac{f_1(w', t)}{|\mathbf{r} - \mathbf{r}'|} d^6w', \quad (2.20)$$

where $M_0$ is the mass of the unperturbed system and $G$ is the gravitational constant. Evaluating this equation by Monte Carlo sampling yields

$$\phi_1(\mathbf{r}, t) = -GM_0 \frac{1}{N} \sum_{i=1}^{N} \frac{f_1(w_i, t)}{f_s(w_i, t)|\mathbf{r} - \mathbf{r}_i|}, \quad (2.21)$$

where the points $w_i$ are randomly chosen with sampling probability density $f_s(w_i, t)$. However, since we shall assume that $f_s d^6w$ is conserved, then so is $f_s$. Hence, the PPM boils down to solving

$$\phi_1(\mathbf{r}, t) = -G \sum_{i=1}^{N} \frac{m_i(t)}{|\mathbf{r} - \mathbf{r}_i|}, \quad (2.22)$$

where

$$m_i(t) = \frac{M_0}{N} \frac{f(w_i(0), 0) - f_0(w_i(t), t)}{f_s(w_i(0), 0)}. \quad (2.23)$$

### 2.4.2 Implementing the Perturbation Particle method with the Hermite version of NBODY1

Implementing the PPM with the Hermite version of NBODY1 (see Chapter 2.2) is in principle relatively simple. As an example consider perturbing a Plummer model (see Chapter 1.8).\(^1\) Thus the unperturbed DF is

$$f_0(w_i(t), t) = A_0 \left( -\phi_0(r_i) - \frac{r_i^2}{2} \right)^{7/2} \quad (2.24)$$

where

$$\phi_0(r_i) = \frac{-GM_0}{\sqrt{r_i^2 + b^2}}. \quad (2.25)$$

The constant $A_0$ is fixed such that the distribution function is normalised to unity. The PPM implementation is described algorithmically as follows:

\(^1\)If a King model is perturbed, and since $m_k$ discrete values of the potential are known (see Chapter 1.8), then cubic spline interpolation is recommended to compute any values of the potential between the $m_k$ discrete values (see Press et al. 1996, Chapter 3).
1. Initialisation:

(a) At $t = 0$ pick some convenient probability density $f_s$ (typically $f_s = f_0$) and distribute perturbation particles and allocate appropriate velocities according to this probability $DF$.

(b) Initialise distribution functions $f(w_i(0),0)$ and $f_0(w_i(0),0)$ for all perturbation particles $i = 1, ..., N$.

(c) Initialise masses of perturbation particles by computing Eq. (2.23).

(d) Compute acceleration and acceleration derivative on particle $i$ due to the equilibrium model (denoted by $a_i^{eq}$ and $\dot{a}_i^{eq}$ respectively):

$$a_i^{eq} = \frac{-GM_0r_i}{(r_i^2 + b^2)^{3/2}}, \quad \dot{a}_i^{eq} = GM_0\left( -\frac{v_i}{(r_i^2 + b^2)^{3/2}} + 3\frac{(v_i \cdot r_i)r_i}{(r_i^2 + b^2)^{5/2}} \right).$$

(e) Compute acceleration on particle $i$ due to the other perturbation particles and consequently the total acceleration (denoted by $a_i^{pp}$ and $a_i$ respectively):

$$a_i^{pp} = \sum_{j=1, j \neq i}^N \frac{m_j r_{ij}}{R_{ij}^3},$$

$$a_i = a_i^{eq} + a_i^{pp}.$$  

where $R_{ij}^2 = |r_j - r_i|^2 + \epsilon^2$, $r_{ij} = r_j - r_i$ and and $\epsilon$ is the softening parameter.

(f) Compute mass time derivative:

$$\dot{m}_i(t) = -\frac{M_0}{N} \frac{f_0(w_i(t),t)}{f_s(w_i(0),0)}$$

where

$$\dot{f}_0(w_i(t),t) = \frac{7}{2} A_0 \left( -\phi_0(r_i) - \frac{v_i^2}{2} \right)^{5/2} \left( -\nabla \phi_0(r_i) - \dot{a}_i \right) \cdot \dot{v}_i.$$  

(g) Compute acceleration derivative on particle $i$ due to the other perturbation particles and consequently the total acceleration derivative (denoted by $\dot{a}_i^{pp}$ and $\dot{a}_i$ respectively):

$$\dot{a}_i^{pp} = \sum_{j=1, j \neq i}^N \left( \frac{m_j v_{ij}}{R_{ij}^3} - 3 \frac{m_j r_{ij}(v_{ij} \cdot r_{ij})}{R_{ij}^5} + \dot{m}_j \frac{r_{ij}}{R_{ij}^3} \right),$$

$$\dot{a}_i = \dot{a}_i^{eq} + \dot{a}_i^{pp}.$$
where $u_{ij} = u_j - u_i$.

(h) Compute the initial timestep for each perturbation particle based on the formula

$$\delta t_i = 0.01 \frac{|a_i|}{|\dot{a}_i|} \quad \text{(2.34)}$$

2. Update phase-space trajectories:

(a) Find next perturbation particle to update, and denote its index by $i$.

(b) Predict all perturbation particles’ position, velocity and acceleration to second order in acceleration derivative.\(^2\)

(c) Compute masses, $m_j$, and mass derivatives, $\dot{m}_j$, for all perturbation particles ($j = 1, ..., N$) using predicted positions, velocities and accelerations, where the masses are determined by Eq. (2.23) and the mass derivative by Eq. (2.30).

(d) Compute acceleration and acceleration derivative on particle $i$ due to the equilibrium model by Eq. (2.26) and Eq. (2.27) respectively.

(e) Compute acceleration and acceleration derivative on particle $i$ due to the other perturbation particles by Eq. (2.28) and Eq. (2.32) respectively.

(f) Compute total acceleration and acceleration derivative on particle $i$ by Eq. (2.29) and Eq. (2.33) respectively.

(g) Perform Hermite corrector on the position and velocity of particle $i$ (to include third derivative of acceleration).

(h) Compute new timestep for particle $i$ based on Eq. (2.16)

3. If termination time is not exceeded then goto 2 else STOP.

2.5 Experimental Stellar Dynamics for Systems with Axial Symmetry

A method for simulating the dynamical evolution of a star cluster with axial symmetry has been devised (van Albada & van Gorkom 1976) which solves Poisson’s equation by expanding density and potential into Legendre polynomials

\(^2\)Has to be second order to preserve order of integrator.
as follows,

$$\rho(r, \theta) = \sum_{l=0}^{\infty} a_l(r) P_l(\cos \theta),$$  \hspace{1cm} (2.35)$$

$$\phi(r, \theta) = \sum_{l=0}^{\infty} b_l(r) P_l(\cos \theta),$$  \hspace{1cm} (2.36)$$

where

$$a_l = \frac{2l + 1}{2} \int_{-1}^{1} \rho(r, \theta) P_l(\cos \theta) d(\cos \theta).$$  \hspace{1cm} (2.37)$$

Substitution of these expansions into Poisson’s equation yields the differential equation

$$\frac{d^2 b_l}{dr^2} + \frac{2 \frac{db_l}{dr}}{r} + \frac{l(l + 1)}{r^2} b_l = 4\pi G a_l.$$  \hspace{1cm} (2.38)$$

This has a solution that satisfies the boundary conditions \(\phi \to 0\) as \(r \to \infty\) and \(\phi \to \text{constant}\) as \(r \to 0\), and is given by

$$b_l = -\frac{4\pi G}{2l + 1} \left( r^l \int_{0}^{\infty} s^{1-l} a_l(s) ds + \frac{1}{r^{l+1}} \int_{0}^{r} s^{2+l} a_l(s) ds \right),$$  \hspace{1cm} (2.39)$$

If one assumes symmetry about the z-axis then one can introduce a fixed radial grid into position space as illustrated in Figure 2.1. Each spherical shell has “thickness” \(\delta r = \frac{r_{\max}}{n_s}\), where \(n_s\) is the number of shells, and for \(r > r_{\max}\) one assumes the density is zero.

To compute the coefficients \(a_l(r)\) for a shell one adopts an averaging strategy such that

$$\bar{a}_l(r) = \frac{4\pi}{V_f} \int_{r}^{r+\delta r} a_l(r') r'^2 dr',$$  \hspace{1cm} (2.40)$$

where

$$V_f = \frac{4\pi}{3} ((r + \delta r)^3 - r^3).$$  \hspace{1cm} (2.41)$$

Therefore if \(\bar{a}_l(r_{i+\frac{1}{2}})\) denotes the average coefficient \(a_l\) belonging to the shell between \(r_i\) and \(r_{i+1}\) then

$$\bar{a}_l(r_{i+\frac{1}{2}}) = \frac{2l + 1}{V_f} \sum_{j \{ |r_i| < |r_j| < r_{i+1} \}} m_j P_l(\cos \theta_j),$$  \hspace{1cm} (2.42)$$
where $m_j$ and $r_j$ correspond to the mass and position vector of the $j^{th}$ particle respectively, and in total one has $N$ particles. At the upper boundary of the $i^{th}$ shell one has

$$\phi_i(r_i, \theta_j) = \frac{-4\pi GP_i(\cos \theta_j)}{2l + 1}(F_1(r_i) + G_1(r_i)), \quad (2.43)$$

where

$$F_1(r_i) = r_i \int_{r_i}^{r_{i+1}} s^{1-l} a_i(s) ds, \quad (2.44)$$

$$G_1(r_i) = \frac{1}{r_{i+1}} \int_{0}^{r_i} s^{2+l} a_i(s) ds. \quad (2.45)$$

Furthermore, $F_1(r_i)$ satisfies the recurrence relation

$$F_i(r_{i+1}) = r_{i+1} \int_{r_i}^{r_{i+1}} s^{1-l} a_i(s) ds + \frac{r_i^{l-1}}{r_i} F_i(r_i). \quad (2.46)$$

Thus by approximating the integral by $\bar{a}_i(r_{i-\frac{1}{2}})(r_i - r_{i-1})\left(\frac{r_{i-1} + r_i}{2}\right)^{1-l}$ one can rewrite Eq. (2.46) as

$$F_i(r_{i-1}) = \left(\frac{r_{i-1}}{r_i}\right)^l \left[\left(\frac{2r_i}{r_i + r_{i-1}}\right)^l \left(\frac{r_i^2 - r_{i-1}^2}{2}\right) \bar{a}_i(r_{i-\frac{1}{2}}) + F_i(r_i)\right], \quad (2.47)$$

27
with \( F_i(r_{\text{max}}) = 0 \). A similar recurrence relation can be derived for \( G_i \) and is found to be

\[
G_i(r_{i+1}) = \left( \frac{r_i}{r_{i+1}} \right)^{l+1} \left[ \left( \frac{r_{i+1} + r_i}{2r_i} \right)^{l+1} \left( \frac{r_{i+1}^2 - r_i^2}{2} \right) \tilde{a}_i(r_{i+\frac{1}{2}}) + G_i(r_i) \right],
\]

where \( G_i(0) = 0 \).

The values of \( F_n(r) \) and \( G_n(r) \) for any radius \( r \) can be approximated by linear interpolation between the shell boundaries, taking particular care at the centre of the system where the only non-zero \( F \) value is when \( l = 0 \).

The forces may be obtained by differentiation of \( \phi \), and thus the force at radius \( r \) and polar angle \( \theta \) is

\[
E(r, \theta) = -\frac{\partial \phi}{\partial r}(r, \theta) \hat{r} - \frac{1}{r} \frac{\partial \phi}{\partial \theta}(r, \theta) \hat{\theta}.
\]

The partial derivatives are approximated by

\[
\frac{\partial \phi}{\partial r}(r, \theta) = -4\pi G \sum_{l=0}^{\infty} \frac{P_l(\cos \theta)}{2l + 1} \left( \frac{dF_l(r)}{dr} + \frac{dG_l(r)}{dr} \right),
\]

where

\[
\frac{dF_l(r)}{dr} = \frac{l}{r} F_l(r) - ra_l(r),
\]

\[
\frac{dG_l(r)}{dr} = -\frac{1 + l}{r} G_l(r) + ra_l(r),
\]

and

\[
\frac{\partial \phi}{\partial \theta}(r, \theta) = -4\pi G \sum_{l=0}^{\infty} \frac{1}{2l + 1} \frac{d(P_l(\cos \theta))}{d\theta} \left( F_l(r) + G_l(r) \right).
\]

For the experiments considered henceforth the potential expansion is truncated after \( l = 3 \) (increasing \( l \) further has a negligible effect on the results). The particles are evolved by a leap-frog method with a fixed time-step.

### 2.6 The Particle-Mesh Method

For completeness, in the remainder of this chapter we now describe some other common methods for solving gravitational \( N \)-body problems. None have been adopted in the numerical experiments described in this thesis, but they have been used in the literature to which we shall refer.
The Particle-Mesh (PM) method solves Poisson's equation on a rectilinear spatial grid. Specifically the space occupied by the model is divided into $J$ cells. In the simplest implementation the potential at the centre of any cell is

$$\phi_{\eta} = \sum_{\xi=1}^{J} G_{\eta\xi} M_{\xi}$$  \hspace{1cm} (2.54)$$

where $M_{\xi}$ is the mass of particles in the $\xi^{th}$ cell and $G_{\eta\xi}$ is the potential at the centre of the $\eta^{th}$ cell which is generated by a unit mass at the centre of the $\xi^{th}$ cell. Thus

$$G_{\eta\xi} = -\frac{G}{\sqrt{\epsilon^2 + |x_{\eta} - x_{\xi}|^2}}$$  \hspace{1cm} (2.55)$$

where $\epsilon$ is the softening parameter (see Chapter 2.2.1), and $x_{\eta}$ and $x_{\xi}$ are the centres of the $\eta^{th}$ and $\xi^{th}$ cells respectively.

Since the centres of the cells are fixed then the $(J - 1)^2$ quantities $G_{\eta\xi}$ are evaluated at the start of the simulation, and one must compute Eq. (2.54) anew at each timestep. By applying the discrete Fourier convolution theorem and taking fast Fourier transforms (see Binney & Tremaine 1994, Chapter 2.8) one may reduce the cost of computing Eq. (2.54) directly. In comparison with the direct method, the PM method reduces the statistical uncertainty in the results since a large number of particles is allowed ($N \sim 100000$). A few studies of clusters have been carried out by PM methods; for example, Hockney & Brownrigg (1974), Miller (1978), but nowadays it is most commonly used for galactic and cosmological simulations.

### 2.7 The Tree Method

A quite different method of directly calculating the force on $N$ bodies, in which the computational effort only grows as $N \log N$, and which models local interactions very accurately, was described by Barnes & Hut (1986). This hierarchical force-calculation algorithm uses a tree-structured subdivision of space into cubic cells, each of which is recursively divided into eight subcells whenever more than one particle is found to occupy the same cell. This tree is constructed anew at every time step (if we assume all particles share the same timestep).

The system is initialised with an empty cubical cell large enough to contain the whole system. One by one, we load particles into this "root" cell. If any two particles fall into the same cell (parent cell), we divide that cell into eight cubical
subcells (daughter cells). Each divided cell is represented by a data structure that holds some information about the distribution of particles that it contains (mass and centre of mass (COM) position) as well as pointers to daughter cells. Once all the particles have been loaded into the tree, the particle-bearing cells are grouped together into larger cubical cells, which are grouped together into still larger parent cells, and so on down to the root cell.

To calculate the force on particle \( p \) one must:

1. Start at the root cell of the tree. Let \( l \) be the length of the cell currently being processed and \( D \) the distance from the cell’s COM to \( p \)

2. If \( l/D < \theta \), where \( \theta \) is a fixed accuracy parameter \( \sim 1 \), then include the interaction between this cell and \( p \) in the total being accumulated (that is, add this cell to the cell interaction list for particle \( p \)), else resolve the current cell into its eight subcells and recursively examine each one in turn. The larger \( \theta \) is the smaller the interaction list becomes and the poorer the accuracy.

Barnes & Hut used a second order time-centred leap frog integrator to test the tree algorithm. In a more recent paper (Aarseth & McMillan 1993) an improved tree code is discussed which incorporates:

1. a fourth order polynomial method (Aarseth 1985)

2. block timesteps (allowing full vectorisation of the code, and larger time steps per particle than a fixed time step algorithm), whereby particles are constrained to have a time step of the form \( 2^n \delta t_0 \), where \( \delta t_0 \) is a minimum time step and \( n \geq 0 \)

3. KS regularisation for strongly interacting bodies

2.8 Hybrid Methods

Performing \( N \)-body simulations requires a certain level of accuracy coupled with a minimal computational cost. One must strike a balance between these two aspects, and this has stimulated the development of a number of hybrid evolutionary models.

The PM method, although very fast, limits spatial resolution to the grid spacing. To improve the accuracy one can compute the short-range forces directly whilst still employing the PM method to compute the long-range forces;
the so called Particle-Particle-Particle-Mesh method, or P$^3$M method (Ferrel & Bertschinger 1994). Another hybrid method is the Tree-Particle-Mesh (TPM) method, where the long-range forces are computed by the PM method, and the short-range forces are handled by a Tree method (Xu 1995). Furthermore it has been found that the TPM results are at least as accurate as the P$^3$M method, whilst taking significantly less computing time if efficiently parallelised (Bode et al. 1999).
Chapter 3

Parallelisation of the $N$-body problem

3.1 Introduction

In order to model large globular clusters realistically one requires a large number of stars ($N \sim 10^6$). Unfortunately with respect to $N$-body simulations as $N$ increases then so does the computational cost. For example, the direct method for solving the $N$-body problem has a computational cost of $O(N^2)$ for the force computations alone. In addition, relaxation effects develop on a time scale of order $N$ crossing times, and so the computational effort scales at least as $O(N^3)$ (see Makino & Hut 1999 for a detailed analysis). In order to allow $N$ to be increased as high as is economically viable one must exploit computer architecture as effectively and efficiently as possible. There are exciting developments in special-purpose hardware designed and built by a group at Tokyo University (Sugimoto et al. 1990); in particular, one class (GRAPE-1 and GRAPE-3) of these devices compute the forces with relatively low accuracy (Makino et al. 1990) and are not suitable for problems dominated by close two-body encounters, whilst the other class (GRAPE-2, GRAPE-4 and GRAPE-6) are ideal for all kinds of problems in star cluster dynamics (Ito et al. 1993, Makino 1999). Alternatively with the birth of parallel super computers one may be able to reduce the computational execution time by a factor of $n_p$, the number of processors; assuming the parallelisation of the code incurs no communication costs during the execution phase.

The rest of this chapter is dedicated to the latter of these strategies, and begins in section 3.1.1 with a description of a specific parallel machine, the CRAY T3D together with a description of a serial machine, part of the Waverley cluster at Edinburgh University. This is followed with a summary of the parallel lan-
gauages adopted for the experiments described and performed in this chapter. This section is concluded with a description of how the parallelisation effect is measured. In section 3.2 the parallelisation of the direct \( N \)-body method (see Chapter 2.2 and 2.3) is reviewed, and the choice of the optimal neighbour number is investigated. Thereafter, in section 3.3, a new strategy to parallelise the direct \( N \)-body method is proposed and tested with numerical simulations. In section 3.4, the advantages and disadvantages of adopting a portable parallel language (High Performance Fortran) is discussed, and investigated further with the aid of numerical experiments. A major aim in the parallelisation of a code is to minimise the inter-processor communication cost. To this end, the inter-processor communication costs may be reduced significantly by a suitable implementation of the hypersystolic method, and this is discussed in section 3.5. To conclude this chapter the parallelisation of a tree code is reviewed in section 3.6.

3.1.1 The CRAY T3D Supercomputer and the Waverley Cluster

Many of the experiments described in this chapter have been performed on the CRAY T3D parallel supercomputer (which was based at the Edinburgh Parallel Computing Centre (EPCC)). The T3D array comprised 256 nodes each with two processing elements (PEs). Each PE consisted of a DEC Alpha 21064 processor running at 150 MHz, supporting 64-bit integer and 64-bit IEEE floating point operations and delivered peak performance per node of 150 64-bit Mflops per second. The peak performance of the 512-PEs was 76.8 Gflop per second. The DEC Alpha 21064 included an 8-Kbyte data cache and an 8-Kbyte instruction cache. Each processor had 64 Mbytes of RAM, giving an aggregate memory of 32 Gbytes. The CRAY T3D was a MIMD (multiple instruction, multiple data) machine which supported SIMD (single instruction, multiple data) parallel languages. The appropriate loops in a code written in a SIMD language are parallelised provided that the data accessed within these loops have been distributed accordingly over the processors. Furthermore, the serial sections of code are executed on all processors simultaneously.

For comparison a number of serial program executions have been performed on the Waverley cluster (based at Edinburgh University Computing Services (EUCS)), specifically Waverley-P3. The P3 is an Ultra-2 platform, runs the Solaris 2.6 operating system and consists of two 296 MHz processing units,
together with 512 Mb of RAM.

3.1.2 The SIMD languages - Fortran 77 with CRAFT, and High Performance Fortran

The CRAY T3D allowed codes to be written in FORTRAN 77 and parallelised by an appropriate implementation of the Cray purpose built CRAFT parallel directives (Henty 1995). To parallelise certain loops one requires appropriate data distributions across the processors. To illustrate some of CRAFT's parallel directives we shall consider distributing the array $X(12, 12)$ across 4 processors.

- The SHARED directive - this directive distributes data across the processors and allows both fine grained and course grained distributions of data. The most fine grained distribution directive takes the form

```
CDIR$ SHARED X(:block(1),:)
```

Here processor 1 has data elements $X(1,:), X(5,:), X(9,:)$, processor 2 has data elements $X(2,:), X(6,:), X(10,:)$ and so on, where the "." in the second component of $X$ implies that all elements 1 to 12 are included. The course grained distribution takes the form

```
CDIR$ SHARED X(: block(3),:)
```

So processor 1 has data elements $X(1:3,:), X(4:6,:), X(7:9,:)$, and so on. More generally the course grained distribution has the syntax

```
CDIR$ SHARED X(:block(N/n_p),:)
```

- The DO SHARED directive - this directive allows DO loops to be parallelised. For example, consider the following data dependent loop

```
DO J=1,12
    sum=sum+X(J,1)
END DO
```

This is parallelised as follows

```
CDIR$ SHARED X(:block(1),:)
```
sum=0.0

CDIR$ MASTER
sumglobal=0.0

CDIR$ ENDMASTER

CDIR$ DO SHARED(J) on X(J, 1)
   DO J=1,12
      sum=sum+X(J, 1)
   END DO

CDIR$ ATOMIC UPDATE
   sumglobal=sumglobal+sum
   sum=sumglobal

   where the statements enclosed by the MASTER and ENDMASTER directives are performed solely on a single processor, namely the master processor. The loop over \( J \) is distributed over the processors determined by the fine grained distribution of \( X \), and the partial sums accumulated on each processor are subsequently added together (the two penultimate lines in the above code) to form the final total \( sum_{global} \) on the MASTER processor. This is then copied to all the other processors in the final line of the above code.

Another SIMD language supported on the CRAY T3D is High Performance Fortran (HPF) which unlike CRAFT is a portable parallel language. Also in contrast to CRAFT, which uses the Fortran 77 program language, HPF uses features present in Fortran 77 and Fortran 90 (Metcalf & Reid 1990) together with additional features. The following summarises the main HPF directives and features

- The PROCESSORS directive: the arrangement of so called "abstract" processors may be specified by the PROCESSORS directive, which assigns a name to the arrangement and dimensions. For example, if the following directive is present in a sample HPF program, the compiler would assume that there are 8 physical processors, and constructs a \( 4 \times 2 \) array of abstract processors which it names \( procs \).

  \[
  \text{!HPF$ PROCESSORS procs(4,2)}
  \]
• The DISTRIBUTE directive: this is used to map array indices onto abstract processors. For example, if we have a $2 \times 2$ array of abstract processors and a two dimensional array $X(1000,1000)$, then the following directives generate a block distribution along both axis:

```
!HPF$ PROCESSORS procs(2,2)
!HPF$ DISTRIBUTE (BLOCK,BLOCK) onto procs :: X
```

That is to say, $\text{procs}(1,1)$ has elements $X(1:500,1:500)$, $\text{procs}(1,2)$ has $X(1:500,501:1000)$, $\text{procs}(2,1)$ has $X(501:1000,1:500)$, and $\text{procs}(2,2)$ has $X(501:1000,501:1000)$.

It is also possible to apply a block distribution along one axis only. For example, the following directives will distribute $X$ along the rows:

```
!HPF$ PROCESSORS procs(4,1)
!HPF$ DISTRIBUTE (BLOCK,*) onto procs :: X
```

The "*" in the distribution array informs the compiler to ignore that dimension whilst distributing the array; in fact, for this example "*" is equivalent to replacing it by "BLOCK". That is to say, $\text{procs}(1,1)$ has elements $X(1:250,1:1000)$, $\text{procs}(2,1)$ has $X(251:500,1:1000)$, $\text{procs}(3,1)$ has $X(501:750,1:1000)$, and $\text{procs}(4,1)$ has $X(751:1000,1:1000)$.

• The FORALL construct: this is similar to the DO loop except that the FORALL creates a parallel construct which guarantees identical results whether applied in serial or parallel. This is achieved by computing the right-hand side of all the assignment statements within a FORALL construct before assigning the left-hand side; in contrast, the DO loop statements are executed in strict sequential order. For example, consider the serial execution of the following DO loop

```
DO I=2,4
   A(I)=A(I-1)
END DO
```

and FORALL statement

```
FORALL (I=2:4) A(I)=A(I-1)
```
If initially \( A = (3, 2, 0, 1) \) then executing the DO loop generates \( A = (3, 3, 3, 3) \); whilst executing the FORALL construct computes \( A = (3, 3, 2, 0) \).

- The INDEPENDENT directive: this provides the compiler with additional information about the execution of a DO loop or a FORALL construct. It is a promise by the user to the compiler that the result of the loop statements will be the same whether executed in serial or in parallel. That is to say, it allows the compiler to parallelise loops in situations where it is not obvious that the result of one pass through the loop is independent of the result of any previous pass (see above example for the FORALL construct, which is not an INDEPENDENT loop). The restrictions on the use of the INDEPENDENT directive state that if a data object is written to in one iteration, then it cannot be written to or read from in another iteration.

- The NEW clause: the NEW clause in an INDEPENDENT directive modifies the meaning of the directive by restricting the scope of the variables used. That is to say, it is a means by which “temporary” variables can be made “private” to a loop, and thus allowed to by-pass some of the restrictions placed on the reading and writing of data objects within an INDEPENDENT loop. For example, if the following loop was executed in parallel, then the variable \( T \) must be made private to each processor. The following directive ensures that the correct results are generated:

\[
!\text{HPF}\$ \text{INDEPENDENT, NEW(T)}
\]

\[
\text{DO J = 1,N }
\]

\[
T = A(J) \times B(J)
\]

\[
A(J) = \text{SQRT(T)}
\]

\[
\text{END DO}
\]

- The REDUCTION clause: this specifies a list of accumulator variable names (separated with commas) and allows the compiler to parallelise loops involving data dependencies. For example, the following directive and loop may be safely parallelised with the aid of the reduction clause:

\[
!\text{HPF}\$ \text{INDEPENDENT, REDUCTION(T)}
\]

\[
\text{DO J = 1,N }
\]
\begin{align*}
T &= T + A(J) \\
\text{END DO}
\end{align*}

It should be noted that it is not correct to place accumulators in NEW clauses because their values must be accumulated across processors.

The above descriptions are only a flavour of the HPF parallel features; detailed descriptions and more advanced topics can be found in Ewing et al. (1994).

### 3.1.3 Measuring the effect of parallelisation

Once a code has been suitably parallelised it is important to ascertain how effective the parallelisation is. In other words, as the processor number \( n_p \) is increased then one desires to know how effective the reduction in execution time is. To measure this effectiveness consider the quantity

\[ E_f = \frac{t_s}{n_p t_p} \times 100\%, \]

where \( E_f \) is known as the efficiency, \( t_p \) and \( t_s \) are the wall-clock times (in seconds) taken to execute the code on \( n_p \) processors and on a single processor respectively. If a code is perfectly parallelised, which is to say that all the computational work required to execute the code has been evenly distributed across the processors and no communication between the processors is necessary, then the efficiency is \( E_f = 100\% \); however, in contrast, the efficiency will decrease if the inter-processor communications increase and/or if a load imbalance between the processors occur.

### 3.2 Parallelisation of a Direct \( N \)-body Method

#### 3.2.1 General Considerations

The simple direct method \( N \)-body solver (described in Chapter 2.2 and denoted by NBODY1) is reasonably easy to parallelise when written in a SIMD language and run on a MIMD machine; specifically NBODY1 has been written in Fortran 77 and CRAFT parallel directives adopted (Heggie & Blair-Fish 1994). The phase-space coordinates were evenly distributed across the processors and the two main loops (see below) in the integrator routine were easily parallelised. The first of these loops, as described in Chapter 2.2.2, is the particle prediction...
loop (part 2 of the algorithm) and was parallelised implicitly without any communication between the processors. The second loop is the force computation loop (part 3 of the algorithm) where the force on a given particle is computed. The force computation loop is a data dependent loop and was explicitly parallelised as described in section 3.1.2. Furthermore since one requires the force on particle $i$ due to the other $(n - 1)$ particles then one first has to ensure that particle $i$'s predicted position is resident on each processor. One can then compute the partial force on particle $i$ due to the other particles resident on a particular processor (as governed by the initial distribution of phase-space coordinates). To obtain the total force on particle $i$ one simply adds together all the partial forces on each processor, and the result is stored on the master processor. Once the force on a given particle is known then its phase-space coordinates are corrected and copied to all processors.

In order to reduce the computational execution time of the Aarseth $N$-body code NBODY6 (as described in Chapter 2.3 and, with some differences, Aarseth 1985) a parallel version has been developed and implemented on the CRAY T3D (Spurzem 1996). This is known as NBODY6++, and is written in Fortran 77 with CRAFT parallel directives. In order to parallelise this code the individual time step scheme (ITS) employed originally by Aarseth has been changed to a hierarchical or block time step scheme, in which the time steps of all particles are commensurate, such that at any given time an entire group of particles are due for a parallel force calculation (irregular and/or regular). This is achieved by allowing the time steps to double in size, halve in size or remain unchanged. At a specified time there will be a group of particles (typically containing $N_{gr}$ particles), referred to as a "block", requiring a force update, and it is these particles within the group that are distributed across the processors. Ideally each processor would be allocated an equal amount of particles from the group to be updated. However, in practise this is not the case as the particle data is distributed a priori across the processors at the start of program execution without any further redistributions of particle data. As a consequence it becomes uncertain how good the load balance will be. After each timestep typically $k \times N_{gr}/n_p$ pieces of data need to be communicated to all the other processors, so that all processors have up-to-date positions and velocities of particles together with force, force derivatives and current neighbour lists for each particle. (The Ahmad Cohen neighbour scheme, as

---

1Subsequently it has been made more portable through the use of the standard message-passing interface MPI.
described in Chapter 2.3, is employed with an optimal neighbour number $N_n$ which on average remains fixed as the particles are evolved with time.) After an irregular force update $k = 19$, and after a regular force update $k = 41 + l_{\text{max}}$, where $l_{\text{max}}$ is the maximum neighbour number. The poor efficiencies Spurzem found when executing NBODY6++ (Spurzem 1996, Table 2) for large processor numbers may be attributed to the following:

- if $\text{mod}(n_p, N_{\text{gr},i}) \neq 0$, where $n_p$ denotes the number of processors and $N_{\text{gr},i}$ is the number of particles in a block to be advanced at the $i^{th}$ time step, then a number of processors will be redundant during parallel updating of the block;

- even if $\text{mod}(n_p, N_{\text{gr},i}) = 0$, then efficiency may be reduced if the particles within the current block are not evenly distributed across the processors;

- communication costs between processors due to the fragmentation of data across the processors.

The first two of these problems are addressed in section 3.3. The last of these problems can be improved if, instead of relying on implicit communication between processors as invoked by the CRAFT parallel directives, one optimises the communication by using hand-written calls of shmem routines; although shmem is not portable its close-to-hardware approach allows a significant increase in efficiency (see http://www2.cray.com/cgi-bin/nph-dweb/dynaweb/all/2518_3.0/ Generic.BookView). This has been achieved by Spurzem & Baumgardt (1998) and implemented on the CRAY T3E. They compared their results to an implementation of NBODY6++ on special purpose computer boards HARP2 (in Heidelberg) or HARP3 (in Cambridge), which originate from the GRAPE group in Tokyo (see section 3.1). At present the CRAY T3E is approximately 10 times faster than HARP, which suffers from a limited memory for neighbour lists and only the regular force calculation is parallelised; however, the HARP is accessible and has a low cost. A different way of achieving improved efficiency by reducing communication costs is to adopt the hypersystolic method of parallelisation, and this is discussed in section 3.5. The execution times of NBODY6++ also suffer since

- block time-steps are used and thus some steps are too short;

- the optimal neighbour number may not be truly optimal.
The first of these problems is an unavoidable sacrifice which enables the parallelisation of NBODY6++. However, the latter is investigated further in the following.

Historically NBODY6++ has evolved from NBODY6 which in turn made use of the one-step 4\textsuperscript{th} order Hermite integrator (HI, see Chapter 2.2.2) as opposed to NBODY5 and its predecessors which made use of the four-step 4\textsuperscript{th} order Adams-Bashforth-Moulton integrator (ABMI). A comparison of the HI scheme and ABMI scheme has been discussed (Makino & Aarseth 1992) and summarised along with some new data in Chapter 2.2.3. Furthermore, Makino & Aarseth have compared in total four methods employed to provide a solution to the \( N \)-body problem; the individual time step scheme (henceforth, ITS) and the Ahmad-Cohen scheme (henceforth, ACS; see Chapter 2.3), both making use of the ABMI scheme, and their equivalent HI schemes, HITS and HACS. As mentioned previously Makino & Aarseth found that the integration error in the total energy of HITS (and HACS) is about \( 1/10 \)th that of ITS (and ACS) for the same number of time steps. For ACS the gain in computational speed, as compared to ITS, is \( (N/3.8)^{1/4} \) for a scalar computer, and the optimal number of neighbours for ACS based on a homogeneous particle distribution is predicted to be (Makino & Hut 1988)

\[
N_n = \left( \frac{\sqrt{n_n}}{n_d} \right)^{3/4} N \tag{3.2}
\]

where \( \eta_n \) and \( \eta_d \) are the accuracy parameters for calculating the irregular and regular time steps respectively (cf. Eq. (2.16)). This is the optimal neighbour number adopted by Spurzem (1996).

An estimate of the processing time \( C_t \), of executing the ACS per small time step (under the assumption that block time-steps are not used and ignoring parallelisation) is given by

\[
C_t = \frac{1}{\gamma} (k_0 N + k_1 N_n + k_2) + k_0 N_n + k_3 \tag{3.3}
\]

where the \( k_i \)'s are defined as follows

- \( k_0 \) is the c.p.t (central processor time) per pairwise force calculation together with a particle's predicted position and velocity calculation;
- \( k_1 \) is the c.p.t to update a particle's neighbour list;
- \( k_2 \) is the c.p.t per regular particle update that is not included in \( k_0 \) and \( k_1 \);
• $k_3$ is the c.p.t per irregular particle update that is not included in $k_0$; and

• $\gamma$ is the average ratio of regular to irregular timesteps, and is estimated by Makino and Hut to be proportional to $N_n^{1/3}$ for a homogeneous particle distribution.

The estimate $C_t$ adopted here differs from both Makino and Hut’s estimate and the estimate made by Spurzem (1996). Makino and Hut split $k_0$ into two constants, one associated with the force update and the other associated with the computation of the particle’s predicted phase-space coordinates. Spurzem’s estimate incorporates constants relating to the speedup (obtained by parallelisation) and suggested that the $k_0$’s associated with the regular and irregular updates may differ on vector computers according to these different vector lengths. The estimate $C_t$ given by Eq. (3.3) is simplified since it does not take into account the fact that neighbour lists may be merged (if there is more than one particle to be updated at a given time) in order to predict all neighbours’ position and velocity prior to an irregular force update. Also, modelling the neighbour list update by $k_1 N_n$ may be inaccurate, and indeed Spurzem found empirically that $k_1 \propto N^{1/5} N_n^{-3/4}$ based on a Plummer particle distribution. However, this simplified estimate of $C_t$ may be sufficient to predict the optimal neighbour number for an inhomogeneous particle distribution.

### 3.2.2 The Optimal Neighbour Number

In order to predict the optimal neighbour number for an initial Plummer distribution, scalar simulations have been performed on the Waverley-P3 processor (see section 3.1.1) so as to ascertain the dependence of $\gamma$ on $N$ and $N_n$. These simulations varied $N$ from 1000 to 8000 and $N_n$ from 20 to 300, and lasted typically for a duration of a few crossing times (see Chapter 1.2). The optimal neighbour number for an homogeneous particle distribution (Eq. (3.2)) depends on the accuracy parameters $\eta_n$ and $\eta_d$, and the total number of particles $N$. For an inhomogeneous particle distribution the optimal neighbour number may vary from particle to particle according to a particle’s location in a cluster; however, in the following one assumes that the optimal neighbour number remains fixed and is independent of phase-space coordinates. The scope of variability of the accuracy parameters is restricted to $\eta_n \leq \eta_d$, since the irregular force is then updated more accurately than the regular force, and it has been found that, as a consequence, the energy error is better behaved (see Figures 3.1(a) and
Figure 3.1: Plot of relative energy error (REE) for $N = 2000$ and with (a) $\eta_n = \eta_d = 0.04, 0.02, 0.01$ (red, blue and green lines respectively), (b) $\eta_n = 0.01$ and $\eta_d = 0.04, 0.02, 0.01$ (red, blue and green lines respectively) and (c) $\eta_d = 0.01$ and $\eta_n = 0.04, 0.02, 0.01$ (red, blue and green lines respectively) against time (in units of $t_{cr}$) (b)). However, if $\eta_n > \eta_d$ then the energy error becomes unpredictable (see Figure 3.1(c)), and furthermore the energy error paradoxically decreases as $\eta_n$ is increased whilst $\eta_d$ is held fixed. For the experiments performed below we take $\eta_n = \eta_d = 0.02$. From Figure 3.2, which plots the execution time against $N_n$, the optimal neighbour number may be estimated (see Table 3.1).
Figure 3.2: Plot of execution time (seconds) against neighbour number $N_n$ for $N = 1000, 2000, 4000, 8000$ (top left, top right, bottom left and bottom right plots respectively).
We may also estimate the optimal neighbour number from a slightly more theoretical basis. First we estimate $\gamma$ by carrying out a least squares fit to the data from the experiments performed (see Figure 3.3), and it is found that

$$\gamma = 0.8184N^{0.251}N_n^{0.146}$$

which fits the data to within 5%. (Incidentally this matches well to the findings of Spurzem & Baumgardt (1998) that $\gamma \propto N^{1/4}$ for the same initial conditions; in contrast, however, they also find empirically that $\gamma \propto N_n^{1/20}$.) Empirically, from the present experiments one finds that $k_0N > 10(k_1N_n + k_2)$ (for all $N$ and $N_n$ considered) and so, if we neglect the terms in $k_1$ and $k_2$ in Eq. (3.3),

Figure 3.3: (a) Plot of log($\gamma$) against log($N_n$) for $N = 1000, 2000, 4000, 8000$ (blue, red, pink and light blue lines respectively). The green line is a least squares fit of the empirical data for $N = 8000$ yielding $\gamma \propto N_n^{0.146}$. (b) Plot of log($\gamma$) against log($N$). The blue line is the least squares fit to the empirical data for $N = 1000, 2000, 4000, 8000$, and the green line is $\gamma \propto N^{0.251}$. 
differentiating Eq. (3.5) with respect to \( N \) and equating to zero yields an approximation to the optimal neighbour number given by

\[
N_n = 0.222N^{0.6536}.
\]

As shown in Table 3.1 this formula for the optimal neighbour number underestimates the optimal neighbour number obtained from the empirical data in Figure 3.2, possibly due to an over-simplification in the estimate of \( C_t \). However, one should note from Table 3.1 that adopting the predicted optimal neighbour number (Eq. (3.6)) reduces the execution time from that based on a homogeneous model (\( N_n^{\text{homog}} \), see Table 3.1 for definition); furthermore, though the execution time may be reduced further by choosing an optimal neighbour number based on empirical data (\( N_n^{\text{inhomog}} \), see Table 3.1 for definition), at best this leads to only an 11% reduction in execution time as compared to \( t_{\text{homog}} \). Spurzem & Baumgardt (1998) found that the optimal neighbour number can be as low as 50 for large particle numbers (\( N = 50000 \)), which is in partial agreement with the above results in the sense that Makino & Hut's estimate of the optimal neighbour is too large. Increasing the neighbour number leads to a decrease in both the number of irregular and regular time steps; firstly the

Figure 3.4: Plot of relative energy error (REE) for \( N = 1000, 2000, 4000 \) (green, blue and red lines respectively) against \( N_n \). The vertical lines represent the optimal neighbour number based on the empirical data in Figure 3.2 (light blue line) and Eq. (3.2) (pink line).
Figure 3.3: (a) Plot of log(γ) against log(Nn) for N = 1000, 2000, 4000, 8000 (blue, red, pink and light blue lines respectively). The green line is a least squares fit of the empirical data for N = 8000 yielding $\gamma \propto N_n^{0.146}$. (b) Plot of log(γ) against log(N). The blue line is the least squares fit to the empirical data for N = 1000, 2000, 4000, 8000, and the green line is $\gamma \propto N^{0.251}$.

We may also estimate the optimal neighbour number from a slightly more theoretical basis. First we estimate $\gamma$ by carrying out a least squares fit to the data from the experiments performed (see Figure 3.3), and it is found that

$$
\gamma = 0.8184N^{0.251}N_n^{0.146}
$$

(3.4)

which fits the data to within 5%. (Incidentally this matches well to the findings of Spurzem & Baumgardt (1998) that $\gamma \propto N^{1/4}$ for the same initial conditions; in contrast, however, they also find empirically that $\gamma \propto N_n^{1/20}$. ) Empirically, from the present experiments one finds that $k_0N > 10(k_1N_n + k_2)$ (for all $N$ and $N_n$ considered) and so, if we neglect the terms in $k_1$ and $k_2$ in Eq. (3.3),
we may approximate

\[ C_t \approx \frac{k_0 N}{\gamma} + k_0 N_n + k_3. \] (3.5)

Differentiating Eq. (3.5) with respect to \( N_n \) and equating to zero yields an approximation to the optimal neighbour number given by

\[ N_n = 0.222N^{0.6536}. \] (3.6)

As shown in Table 3.1 this formula for the optimal neighbour number underestimates the optimal neighbour number obtained from the empirical data in Figure 3.2, possibly due to an over-simplification in the estimate of \( C_t \). However, one should note from Table 3.1 that adopting the predicted optimal neighbour number (Eq. (3.6)) reduces the execution time from that based on a homogeneous model \( (N_n^{\text{homog}}, \text{see Table 3.1 for definition}) \); furthermore, though the execution time may be reduced further by choosing an optimal neighbour number based on empirical data \( (N_n^{\text{inhomog}}, \text{see Table 3.1 for definition}) \), at best this leads to only an 11% reduction in execution time as compared to \( t_{\text{homog}} \). Spurzem & Baumgardt (1998) found that the optimal neighbour number can be as low as 50 for large particle numbers \( (N = 50000) \), which is in partial agreement with the above results in the sense that Makino & Hut's estimate of the optimal neighbour is too large. Increasing the neighbour number leads to a decrease in both the number of irregular and regular time steps; firstly the
irregular force becomes smoother since there are more particles in the neigh-
bouring sphere and thus the irregular time step may be increased, and secondly
the regular force becomes smoother since it loses particles which contribute
the highest fluctuating force to the regular force and thus the regular time step
may be increased. Collectively a decrease in the number of irregular and regular
time steps causes the relative energy error of the system to increase (see Figure
3.4). Thus, the energy error may be reduced by adopting the optimal neighbour
number based on the empirical data of an inhomogeneous model as opposed to
basing the optimal neighbour number on Eq. (3.2) (since $N_{n}^{inhomag} < N_{n}^{homag}$);
in fact, for $N = 4000$ the energy error may be reduced by 50% by adopting the
empirical optimal neighbour number $N_{n}^{inhomag}$ as opposed to $N_{n}^{homag}$.

Table 3.1: Comparison of the optimal neighbour neighbour number - where $N_{n}^{homag}$
is the optimal neighbour number based on Eq. (3.2) (a homogeneous distribution),
$N_{n}^{inhomag}$ is based on the empirical data of Figures 3.2 (an inhomogeneous distri-
bution), $N_{n}^{pred}$ is based on Eq. (3.6), $t_{homog}$ is the execution time when employing
$N_{n}^{homag}$, $t_{inhomog}$ is the execution time when employing $N_{n}^{inhomag}$, $t_{pred}$ is the execution
time when employing $N_{n}^{pred}$, $t_{crit}$ is the N-body model termination time (in units
of the crossing time) and the "Gain" is the percentage decrease of the execution time
from $t_{homog}$ when using $N_{n}^{inhomag}$.

<table>
<thead>
<tr>
<th>$N$</th>
<th>$N_{n}^{homag}$</th>
<th>$N_{n}^{inhomag}$</th>
<th>$N_{n}^{pred}$</th>
<th>$t_{homog}$</th>
<th>$t_{inhomog}$</th>
<th>$t_{pred}$</th>
<th>$t_{crit}$</th>
<th>Gain (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>63</td>
<td>30</td>
<td>20</td>
<td>70</td>
<td>66</td>
<td>67</td>
<td>2</td>
<td>5.7</td>
</tr>
<tr>
<td>2000</td>
<td>106</td>
<td>50</td>
<td>32</td>
<td>231</td>
<td>209</td>
<td>220</td>
<td>2</td>
<td>9.5</td>
</tr>
<tr>
<td>4000</td>
<td>178</td>
<td>70</td>
<td>50</td>
<td>876</td>
<td>790</td>
<td>810</td>
<td>2</td>
<td>9.8</td>
</tr>
<tr>
<td>8000</td>
<td>299</td>
<td>130</td>
<td>79</td>
<td>1750</td>
<td>1559</td>
<td>1600</td>
<td>1</td>
<td>10.9</td>
</tr>
</tbody>
</table>

3.3 A new strategy for parallelising NBODY6++

3.3.1 Introduction

As stated previously the parallelisation of NBODY6++ suffers poor efficien-
cies as one increases processor numbers. In particular if $\text{mod}(n_{p}, N_{gr,i}) \neq 0$,
where $n_{p}$ is the number of processors and $N_{gr,i}$ is the number of particles in
a block to be updated at the $i^{th}$ time level, then a number of processors will
be redundant at some point during the parallel execution phase. Furthermore,
if $\text{mod}(n_{p}, N_{gr,i}) = 0$ then the efficiency may be poor since the particles in
the block may not be evenly distributed across the processors. So as to avoid
these two problems, and thus improve efficiency, one may adopt the following
proposal.
3.3.2 An improved timestep strategy

Instead of block time steps as used by Spurzem (1996), suppose one adopts individual time steps. The basic strategy is then to find the next particle (particle $i$) to evolve which resides on a given processor, and to "bring down" a particle on each of the other processors to the same time level as particle $i$ (a similar strategy has been discussed by Sweatman (1994)). So all one has to do is to evolve all $n_p$ particles simultaneously, and since each particle resides on a different processor then this update can be done in parallel. Of course the particles one chooses to "bring down" are of paramount importance to the success of this strategy, and this "choice" is detailed more precisely in the following algorithm, denoted by the "Optimal parallel algorithm (OPA)":

1. Find next particle on each processor due for advancement. So for $n_p$ processors one finds $n_p$ particles, denoted by $\{\text{particles}_j : j = 1, \ldots, n_p\}$.

2. Find a "back-up" particle on each processor such that this particle will only require an irregular force update (and not a regular force update), and out of these particles considered on a particular processor choose the one which is next to be updated. These $n_p$ particles are denoted by $\{\text{BUparticles}_j : j = 1, \ldots, n_p\}$ and may indeed be the same as $\{\text{particles}_j\}$ at least for some $j$.

3. Find the next particle to be updated out of $\{\text{particles}_j : j = 1, \ldots, n_p\}$, and let this be referred to as particle $i$ on processor $p$.

4. If the $i^{th}$ particle requires a regular force update then look at all the processors (except processor $p$) and find the next particle to be updated on each of these processors provided that such a particle invokes a regular force update (a backup list of particles only requiring a regular force update is not computed in step 2 since the total number of regular force updates is much less than the total number of irregular force updates, and thus it is computationally cheaper to compute the list of particles requiring a regular force update at this step). All of these $n_p$ particles are now stored in an array $\text{nxtlst}(1 : n_p)$.

Otherwise if the $i^{th}$ particle only requires an irregular force update then the particles one chooses to bring down to the $i^{th}$ particle's update time level must only invoke an irregular force update (and not a regular force update). In other words, one chooses the particles $\{i\}$ and $\{\text{BUparticles}_j : \ldots\}$.
\( j = 1, \ldots, n_p, j \neq i \). All of these \( n_p \) particles are now stored in an array \( \text{nxtlst}(1 : n_p) \).

5. The next block of \( n_p \) particles specified in the array \( \text{nxtlst}(1 : n_p) \) will either require solely an irregular force update or an irregular force update followed by a regular force update. The \( n_p \) particles in the block are updated accordingly.

6. Compute new timestep for each of the \( n_p \) particles in the current block.

7. If termination time or output time has not been exceeded then goto 1.

As a consequence of the above strategy one ensures that \( \text{mod}(n_p, N_{gr,i}) = 0 \) since \( N_{gr,i} = n_p \), and the particles within a block are evenly distributed across the processors. However, there is an inefficiency due to updating particles before one has to.

### 3.3.3 Results

As can be observed from Tables 3.2 and 3.3, in terms of the efficiencies (see Eq. (3.1)) associated with the irregular and regular force updates, the OPA is much improved over the “block-timestep” approach to parallelising NBODY6++ (see Table 3.4 which shows efficiencies for both approaches based on the quantity \( t_{irr} \), which is defined below). The headings of the columns of Tables 3.2 and 3.3 are:

- \( N \) - particle number
- \( P_{\text{setup}} \) - the number of virtual processors encoded into the program (Table 3.3 only). This is necessary so that one can measure the speed-up when the real processor number is increased from 1 to \( n_p \).
- \( n_p \) - the number of processors
- \( t_{irr} \) - the real time (in seconds) taken to execute all irregular force updates
- \( t_{reg} \) - the real time (in seconds) taken to execute all regular force updates
- \( t_{mos} \) - the real time (in seconds) of the “one to all” communications after all irregular and/or regular force updates
- \( t_{\text{block}} \) - the total real time (in seconds) of finding the next block of particles to update
- \textit{istep} - the total number of irregular particle updates
- \textit{rstep} - the total number of regular particle updates
- \textit{t_{crit}} - termination time (in units of the crossing time).

Table 3.2: Timing NBODY6++ using the “block-timestep” approach to parallelise the code. The parallelised code was implemented with Fortran 77 and CRAFT on the CRAY T3D. The accuracy parameters were $\eta_n = 0.02$, $\eta_d = 0.04$

<table>
<thead>
<tr>
<th>(N)</th>
<th>(n_p)</th>
<th>(t_{\text{irr}})</th>
<th>(t_{\text{reg}})</th>
<th>(t_{\text{mov}})</th>
<th>(t_{\text{block}})</th>
<th>(\text{istep})</th>
<th>(\text{rstep})</th>
<th>(t_{\text{crit}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>1014</td>
<td>1</td>
<td>114.9</td>
<td>81.8</td>
<td>5.6</td>
<td>1.1</td>
<td>237265</td>
<td>19438</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>7.1</td>
<td>3.9</td>
<td>9.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>4.9</td>
<td>2.4</td>
<td>8.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2038</td>
<td>1</td>
<td>482.3</td>
<td>307.1</td>
<td>16.3</td>
<td>3.7</td>
<td>577865</td>
<td>37766</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>24.8</td>
<td>13.4</td>
<td>29.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>16.4</td>
<td>8.1</td>
<td>27.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4086</td>
<td>1</td>
<td>180.2</td>
<td>91.9</td>
<td>3.0</td>
<td>1.5</td>
<td>144847</td>
<td>5191</td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>8.1</td>
<td>4.5</td>
<td>4.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>5.2</td>
<td>2.8</td>
<td>4.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>3.5</td>
<td>1.8</td>
<td>5.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.3: Timing NBODY6++ using the “OPA” approach to parallelise the code. The parallelised code was implemented with Fortran 77 and CRAFT on the CRAY T3D. The accuracy parameters were $\eta_n = 0.02$, $\eta_d = 0.04$

<table>
<thead>
<tr>
<th>(N)</th>
<th>(P_{\text{setup}})</th>
<th>(n_p)</th>
<th>(t_{\text{irr}})</th>
<th>(t_{\text{reg}})</th>
<th>(t_{\text{mov}})</th>
<th>(t_{\text{block}})</th>
<th>(\text{istep})</th>
<th>(\text{rstep})</th>
<th>(t_{\text{crit}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>1014</td>
<td>32</td>
<td>1</td>
<td>151.7</td>
<td>70.9</td>
<td>6.9</td>
<td>7.7</td>
<td>328129</td>
<td>16768</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>6.5</td>
<td>2.5</td>
<td>10.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>1</td>
<td>219.1</td>
<td>82.8</td>
<td>9.2</td>
<td>6.3</td>
<td>461210</td>
<td>19776</td>
<td></td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>4.8</td>
<td>1.5</td>
<td>12.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2038</td>
<td>32</td>
<td>1</td>
<td>490.1</td>
<td>304.0</td>
<td>14.7</td>
<td>29.9</td>
<td>669047</td>
<td>31200</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>18.9</td>
<td>10.0</td>
<td>22.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>1</td>
<td>626.2</td>
<td>341.1</td>
<td>18.6</td>
<td>21.9</td>
<td>886198</td>
<td>35200</td>
<td></td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>12.4</td>
<td>5.6</td>
<td>25.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4086</td>
<td>64</td>
<td>1</td>
<td>198.3</td>
<td>109.9</td>
<td>3.8</td>
<td>9.1</td>
<td>195531</td>
<td>5120</td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>3.7</td>
<td>1.9</td>
<td>5.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>1</td>
<td>259.8</td>
<td>126.1</td>
<td>5.1</td>
<td>7.0</td>
<td>270264</td>
<td>5888</td>
<td></td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>2.5</td>
<td>1.0</td>
<td>7.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The OPA does suffer from a slight increase in the time associated with communicating all fragmented data to all the other processors \((t_{\text{mov}})\) for sufficiently large processor number, due to an increase in the number of irregular and regular timesteps. However, this may not be a major setback for the OPA since:
• Work has been done by Spurzem & Baumgardt (1998) to reduce the communication costs by using hand-written calls of shmem routines (see http://www2.cray.com/cgi-bin/nph-dweb/dynaweb/all/2518.3.0/@Generic_BookView).

• Improvements to parallel machine architecture will reduce these communication costs. For example on the CRAY T3E these communication costs should be reduced by a factor of three.

It follows that, if these "one-to-all" communication costs are reduced, then in principle the OPA will perform much better than the current "block-timestep" approach. For example, for \( N = 4086 \) and \( n_p = 128 \) the dominant timings of NBODY6++ will come from \( t_{irr} \) and \( t_{reg} \), and then the OPA yields a 34% decrease in execution time.

Table 3.4: Comparison of efficiencies (for irregular force update). Efficiencies based on regular force updates are similar. The two methods under comparison are the "block-timestep" and OPA strategies for the parallelisation of NBODY6++.

<table>
<thead>
<tr>
<th>( N )</th>
<th>( n_p )</th>
<th>Efficiency of the block timestep method</th>
<th>Efficiency of the OPA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1014</td>
<td>32</td>
<td>51</td>
<td>73</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>37</td>
<td>71</td>
</tr>
<tr>
<td>2038</td>
<td>32</td>
<td>61</td>
<td>81</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>46</td>
<td>79</td>
</tr>
<tr>
<td>4086</td>
<td>64</td>
<td>54</td>
<td>84</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>40</td>
<td>81</td>
</tr>
</tbody>
</table>

Unfortunately, as can be seen from Table 3.3, there is a problem with the OPA, and that is the time spent finding the next block of particles to update (\( t_{block} \)). In theory, the section of code associated with finding the next block of particles when using the OPA can be parallelised (and indeed is parallelised, but rather poorly, to judge by the above results). When this section of code is parallelised in isolation (i.e. it is treated as a "toy" code without using NBODY6++) then it yields an efficiency of 70% (for \( N = 1014, n_p = 32 \)). When embedded in the NBODY6++ code the time \( t_{block} \) associated with the OPA in scalar mode exceeds the time \( t_{block} \) associated with the "block-timestep" method; however, since the former is parallelised then the time \( t_{block} \) associated with the OPA should reduce significantly as \( n_p \) is increased. Unfortunately it turns out that this is not the case. Any attempt, at present, at parallelising
this section of code embedded in the OPA version of NBODY6++ runs into cache memory problems, and consequently $t_{\text{block}}$ cannot be reduced when $n_p$ is increased. However, on a machine which supports a secondary cache memory per processor (unlike the CRAY T3D which only had a primary cache) one may overcome this problem. Such a machine is the CRAY T3E which has a secondary data cache memory of 96K.

### 3.4 Parallel implementation of NBODY6++ with High Performance Fortran

#### 3.4.1 Introduction

The parallel code NBODY6++, with the usual block time step procedure, has been implemented on the CRAY T3D in Fortran 77 with CRAFT parallel directives (Spurzem 1996), and was described in section 3.2. The motivation to rewrite NBODY6++ in HPF is that HPF is a portable parallel language whereas CRAFT runs solely on the CRAY parallel machine. There is a price to pay for the portability of HPF, which is that an HPF code will not scale as well as a CRAFT code on a CRAY parallel machine. One of course expects this since CRAFT is a purpose built parallel language for the CRAY parallel machine, and as a consequence has been optimised in such a way that the efficiency has been maximised and communication costs minimised.

#### 3.4.2 Implementation

As already mentioned in section 3.2 the implementation of NBODY6++ on the CRAY T3D (which is a MIMD computer) worked by executing the serial sections of code on all processors simultaneously, and executing the parallel sections of code in parallel across the processors according to the appropriate data distribution. In contrast to using Fortran77 with CRAFT parallel directives, the code was written by the author in HPF (which is also a SIMD parallel language). In this code the loop in the main integrator of NBODY6++ over all particles in a group (the particles at a specified time for which a force calculation is simultaneously necessary) was split into two parts. One contains the neighbour force computation. The second one is restricted to that subset of the particles of the group which are due for a regular (non-neighbour) force calculation as well. Each loop was executed in parallel, one after the other.
In this code, the communication costs occur immediately after each of these loops has been completed, when the distributed variables are copied to the undistributed variables. This was necessary since the serial sections of code required up-to-date values for variables such as the position vectors of the particles. To elaborate further, consider one component of a position vector which has been updated in one of the parallel loops: this new value needs to be copied to all copies of that undistributed particle’s position vector, a copy of which resides on each processor. In other words, every update of a distributed variable results in a “one to all” processor communication. It is these “one to all” communication costs which tend to dominate execution times as the processor number increases, as shown by Spurzem for his CRAFT version of NBODY6++. As a consequence, for NBODY6++ to be efficiently implemented with HPF one requires that the “one to all” communication costs (a processor communicating to all other processors) of HPF be (at worst) comparable to those of CRAFT.

To illustrate the difference in the “one to all” communication costs of HPF and CRAFT, consider the following loops embedded in “toy” HPF and CRAFT codes. In both cases the variable $X$ has dimension $(SIZE,L)$, where $SIZE$ is defined in the following, and $X$ is distributed block-wise in the second dimension across the processors. The cost of the “one to all” communications can be estimated by considering the timings (in seconds) of two constructs: firstly, the real time $t_A$ of executing the following

```fortran
DO I=1,L
  DO K=1,SIZE
    A(K)=X(K,I)
  END DO
END DO
```

and secondly, the real time, $t_B$, of executing the following “colon” structure (only available with HPF)

```fortran
DO I=1,L
  A(1:SIZE)=X(1:SIZE,I)
END DO
```

where $A$ has dimension $(SIZE)$ and is not distributed, and thus a copy of $A$ resides on each processor. The two loops above are equivalent with respect to the final outcome, and it is the way in which the assignments are made from the right-side to the left-side variables which differ between the two loops; the first
loop communicates data from $X$ to $A$ one element at a time, but in contrast the second loop communicates data from $X$ to $A$ a block (of length $SIZE$) at a time.

As can be observed from Tables 3.5 and 3.6, HPF performs poorly in its “one to all” communication costs as compared to CRAFT. This is especially evident when $SIZE = 2$ and we compare $t_B$ for HPF with $t_A$ for CRAFT (see Table 3.5). Furthermore HPF’s “one to all” communication costs are at best approximately 37 times slower than CRAFT’s “one to all” communication costs for this particular value of $SIZE$.

Table 3.5: Comparison of the “one to all” communication costs with $L = 2000$, $SIZE = 2$ between HPF and CRAFT

<table>
<thead>
<tr>
<th>$n_p$</th>
<th>HPF ($t_A^{(H)}$, $t_B^{(H)}$)</th>
<th>CRAFT ($t_A^{(C)}$)</th>
<th>$t_B^{(H)}/t_A^{(C)}$</th>
<th>$t_B^{(H)}/t_A^{(C)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0032, 0.70</td>
<td>0.00291</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.11, 0.78</td>
<td>0.00297</td>
<td>37</td>
<td>262</td>
</tr>
<tr>
<td>4</td>
<td>0.14, 0.82</td>
<td>0.0031</td>
<td>45</td>
<td>264</td>
</tr>
<tr>
<td>8</td>
<td>0.16, 0.94</td>
<td>0.0040</td>
<td>40</td>
<td>235</td>
</tr>
<tr>
<td>16</td>
<td>0.19, 1.08</td>
<td>0.0037</td>
<td>51</td>
<td>292</td>
</tr>
</tbody>
</table>

Table 3.6: Comparison of the “one to all” communication costs with $L = 2000$, $SIZE = 32$ between HPF and CRAFT

<table>
<thead>
<tr>
<th>$n_p$</th>
<th>HPF ($t_A^{(H)}$, $t_B^{(H)}$)</th>
<th>CRAFT ($t_A^{(C)}$)</th>
<th>$t_B^{(H)}/t_A^{(C)}$</th>
<th>$t_B^{(H)}/t_A^{(C)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.51, 0.72</td>
<td>0.035</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1.95, 0.80</td>
<td>0.050</td>
<td>39</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>2.54, 0.85</td>
<td>0.059</td>
<td>43</td>
<td>14</td>
</tr>
<tr>
<td>8</td>
<td>2.97, 0.99</td>
<td>0.072</td>
<td>41</td>
<td>14</td>
</tr>
<tr>
<td>16</td>
<td>3.39, 1.16</td>
<td>0.067</td>
<td>51</td>
<td>17</td>
</tr>
</tbody>
</table>

These incredibly poor HPF “one to all” communication costs can be improved (for larger “$SIZE$”, see Table 3.6) by employing the “colon” structure, whereby blocks of data are transmitted to all processors, as compared to transmitting data as individual elements. But still HPF is, at best, approximately 14 times slower (for $SIZE = 32$) than CRAFT. There is a minimum value for the $SIZE$ of data being transmitted, whilst using HPF, at which the colon structure becomes more efficient than transmitting data elements individually. Experimentally $SIZE$ must be greater than 15 in order for the
colon structure to be the more efficient. Furthermore, the experiments performed above with the simple “toy” HPF and CRAFT codes have been executed on the Cray T3E at Edinburgh University by Pringle (1998). The HPF code was rewritten in HPF-CRAFT (HPF program language with CRAFT parallel directives, see http://www.epcc.ed.ac.uk/t3d/documents/techreports/EPCC-TR98-02/index.html) and $t^{(h)}_A$ is approximately the same as $t^{(C)}_A$ as one might expect. However, if one adopts the “colon” structure and $SIZE > 110$ then for $n_p = 8$ the HPF-CRAFT code executes faster than the F77-CRAFT code.

Returning to NBODY6++, for each particle that is updated, a certain number ($SIZE$) of updated data (see section 3.2) must be copied to all the other processors in a “one to all” communication. This number varies from 19 per particle (after an irregular force update) to 119 per particle (after a regular update). As expected these “one to all” communication costs are reduced if one employs the “colon” structure, as opposed to the DO loop structure, for these communications in the HPF version of NBODY6++ implemented on the T3D. Furthermore, in order to minimise the “one to all” communication costs one may adopt a hybrid “colon” approach, as follows.

Consider the “one to all” communication costs after an irregular force update. For each particle in the group, 19 pieces of data need to be copied to all processors. After each irregular force update, $19 \times N_{gr,i}$ pieces of data are copied to all processors. The distributed variable associated with the irregular force update is called $X_{KSR}$ and has dimensions $(64, NMAX)$, where $NMAX$ is the maximum number of particles (including centres of mass of regularised pairs), and is distributed block-wise in the second dimension across the processors. It is the first 19 elements of the first dimension of $X_{KSR}$ which need to be copied to all processors, and this is done for each particle in the group (which corresponds to indices in the second dimension of $X_{KSR}$). In the hybrid “colon” approach one transmits the data to other processors according to the following algorithm:

1. Consider each processor in turn and compute how many particles of the group reside on that processor. Denote this number by $Num_p$ where $p = 1, 2, ..., n_p$.

2. If $Num_p \leq \theta$ then for each particle $i$ on that processor which is also in the group, copy the first 19 elements to an undistributed variable $A1$ i.e. $A1(1:19) = X_{KSR}(1:19, i)$. The data contained in $A1$ is then copied to
the relevant undistributed variables (for example the phase-space coordinates). This is done separately for each of the \( N_{um_p} \) particles.

3. If \( N_{um_p} > \theta \) then copy the entire block of particles on that processor to an undistributed variable \( A2 \). In other words, \( A2(1:19,1:N/n_p) = XKSR(1:19,l:u) \) where \( l \) and \( u \) correspond to the lower and upper indices respectively of the block of particles on that particular processor. The data contained in \( A2 \) is then copied to the relevant undistributed variables.

The rationale behind this approach is that if there is little data to move from a given processor then it would be wasteful to move the entire block of data on that processor, despite the advantage of using the colon construct. In contrast, if there is much data on a given processor which needs to be communicated to all the other processors then it becomes more economical to copy the entire block of data residing on that processor since the inefficiency of copying superfluous data is compensated by the use of the colon construct, which is more efficient when \( N_{um_p} \) is large enough.

Now we consider the efficiency of this hybrid scheme compared with simpler schemes. Initially NBODY6++ was directly translated into HPF from its original CRAFT form. The timings of the CRAFT NBODY6++ code and HPF NBODY6++ code can be observed in Tables 3.7 and 3.8.

### Table 3.7: Timing of NBODY6++ implemented with CRAFT, for \( t_{crit} = 0.1 \) and \( N = 1000 \)

<table>
<thead>
<tr>
<th>( n_p )</th>
<th>( t_{int} )</th>
<th>( t_{irr} )</th>
<th>( t_{reg} )</th>
<th>( t_{mov,irr} )</th>
<th>( t_{mov,reg} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>110</td>
<td>63</td>
<td>41</td>
<td>1.75</td>
<td>0.64</td>
</tr>
<tr>
<td>2</td>
<td>60</td>
<td>33</td>
<td>21</td>
<td>1.8</td>
<td>0.81</td>
</tr>
<tr>
<td>4</td>
<td>37</td>
<td>17.3</td>
<td>10.8</td>
<td>1.9</td>
<td>0.9</td>
</tr>
<tr>
<td>8</td>
<td>22</td>
<td>9.7</td>
<td>5.8</td>
<td>1.98</td>
<td>0.98</td>
</tr>
<tr>
<td>16</td>
<td>16.2</td>
<td>5.7</td>
<td>3.17</td>
<td>1.98</td>
<td>1.01</td>
</tr>
<tr>
<td>32</td>
<td>13.1</td>
<td>3.5</td>
<td>1.88</td>
<td>2.11</td>
<td>1.04</td>
</tr>
</tbody>
</table>

### Table 3.8: Timing of NBODY6++ implemented with HPF, for \( t_{crit} = 0.1 \) and \( N = 1000 \)

<table>
<thead>
<tr>
<th>( n_p )</th>
<th>( t_{int} )</th>
<th>( t_{irr} )</th>
<th>( t_{reg} )</th>
<th>( t_{mov,irr} )</th>
<th>( t_{mov,reg} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>131</td>
<td>55</td>
<td>47</td>
<td>17.2</td>
<td>6.4</td>
</tr>
<tr>
<td>2</td>
<td>164</td>
<td>28</td>
<td>21.4</td>
<td>82</td>
<td>28</td>
</tr>
<tr>
<td>4</td>
<td>183</td>
<td>15</td>
<td>11</td>
<td>114</td>
<td>38</td>
</tr>
<tr>
<td>8</td>
<td>215</td>
<td>8.5</td>
<td>6.3</td>
<td>147</td>
<td>49</td>
</tr>
<tr>
<td>16</td>
<td>248</td>
<td>4.9</td>
<td>3.6</td>
<td>177</td>
<td>58</td>
</tr>
<tr>
<td>32</td>
<td>284</td>
<td>2.97</td>
<td>1.99</td>
<td>206</td>
<td>68</td>
</tr>
</tbody>
</table>

56
The headings of the columns in Tables 3.7 and 3.8 are described in section 3.3.3 except:

- $t_{\text{mov,irr}}$ - the real time (in seconds) of the “one to all” communications after an irregular force update
- $t_{\text{mov,reg}}$ - the real time (in seconds) of the “one to all” communications after a regular force update.

As can be observed from these tables HPF performs very poorly in terms of “one to all” communications ($t_{\text{mov,irr}}$ and $t_{\text{mov,reg}}$) as compared to CRAFT. One can reduce the communication costs incurred by HPF by employing the hybrid “colon” algorithm mentioned earlier. Considering just the “one to all” communications after the irregular force updates, by an appropriate choice of $\theta$ these communication costs can be reduced by a factor of between 5 and 6 (see Table 3.9).

Table 3.9: Timing the “one to all” communications after the irregular force update of the NBODY6++ HPF code, for $t_{\text{crit}} = 0.1$ and $N = 1000$

<table>
<thead>
<tr>
<th>$n_p$</th>
<th>$t_{\text{mov,irr}}$</th>
<th>$\theta$</th>
<th>$n_p$</th>
<th>$t_{\text{mov,irr}}$</th>
<th>$\theta$</th>
<th>$n_p$</th>
<th>$t_{\text{mov,irr}}$</th>
<th>$\theta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>25.9</td>
<td>1</td>
<td>4</td>
<td>26.9</td>
<td>1</td>
<td>8</td>
<td>40</td>
<td>1</td>
</tr>
<tr>
<td>16.8</td>
<td>19.2</td>
<td>8</td>
<td>28</td>
<td>8</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15.1</td>
<td>19.6</td>
<td>16</td>
<td>29</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Unfortunately this reduction in communication costs is still not enough to make HPF a viable option. For example, for 8 processors $t_{\text{mov,irr}} = 29$ seconds for HPF, whilst for CRAFT $t_{\text{mov,irr}} = 1.98$ seconds. So for 8 processors the “one to all” communication costs for HPF are 14 times slower than CRAFT’s communication costs. Furthermore, as observed in Tables 3.8 and 3.9, the “one to all” communication costs still dominate the HPF timings for four or more processors ($N = 1000$). In conclusion HPF is inefficient in the parallelisation of NBODY6++ and proves to be no competition to CRAFT, at least on the T3D and for the “SIZE” of data communicated to the other processors.

3.5 The Hypersystolic Method of Parallelisation

3.5.1 Introduction

As discussed in sections 3.2 to 3.4 it is of paramount importance to minimise communication costs between processors, and although the hypersystolic
method does not make communication costs negligible, it does significantly reduce them. This new class of parallel algorithm (Lippert et al. 1995) can reduce the conventional communicational complexity of $O(n^2)$ to $O(n^{3/2})$, where $n$ is typically both the number of particles and the number of processors. The general computational problem considered by Lippert et al. is the evaluation of

$$y_i = f(x_i, x_j)$$  \hspace{1cm} (3.7)

for $i = 1, 2, ..., n$, with an input array $x = (x_1, x_2, ..., x_n)$ and a resulting array $y = (y_1, y_2, ..., y_n)$. Clearly the $N$-body problem (Eq. (2.1)) is a member of this general computational problem provided that a fixed time step criterion is employed.

### 3.5.2 The Standard Systolic Method

Consider the case when the $i^{th}$ processor computes $y_i$, and $x$ is distributed over the $n$ processors such that $x_i$ resides on processor $i$. A copy $\hat{x}$ is made of $x$, where $x$ is the fixed array and $\hat{x}$ is the moving array which communicates data elements $x_j$ to the other $(n-1)$ processors in $n-1$ clock steps (see Figure 3.5). Both the computational and communicational complexity for the standard systolic method is $O(n^2)$. 

Figure 3.5: Topology of the standard systolic method (Lippert et al. 1995)
3.5.3 The Hypersystolic Algorithm

As with the standard systolic method there is only one moving array $\hat{x}$, but in contrast the shifted arrays are stored. Consider the matrix $C$ which explicitly enumerates the data elements of the moving array $\hat{x}$, for clock steps $t = 0, 1, \ldots, n - 1$, as delivered successively in the standard systolic method:

$$C = \begin{bmatrix}
1 & 2 & 3 & \cdots & n - 1 & n \\
2 & 3 & 4 & \cdots & n & 1 \\
\vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\
n - 1 & n & \cdots & n - 4 & n - 3 & n - 2 \\
n & 1 & \cdots & n - 3 & n - 2 & n - 1
\end{bmatrix}$$

where the rows correspond to the clock steps and columns correspond to the processors. For the standard systolic method there are $n - 1$ shifts of stride $a_t = 1$ for $t = 1, 2, \ldots, n - 1$.

The topological features of the underlying communication structure of a parallel computer may allow shifts of different strides. Henceforth, let $r$ denote the number of different strides supported on a given system, and let the set of shifts be $A_k = (a_1, a_2, \ldots, a_k)$ where $k$ denotes the number of shifts. For example, consider $n = 6$, $r = 2$, $k = 2$ and $A_2 = (1, 2)$. The resulting matrix is

$$C = \begin{bmatrix}
1 & 2 & 3 & 4 & 5 & 6 \\
2 & 3 & 4 & 5 & 6 & 1 \\
4 & 5 & 6 & 1 & 2 & 3
\end{bmatrix}$$

After 2 clock steps it is possible to have computed a fixed array given by

$$y = (f_{1,4}, f_{3,5}, f_{4,6}, f_{5,1}, f_{6,2}, f_{1,3}),$$

where $f_j^i = \sum_j f(x_j, x_j)$ and the $i^{th}$ element of $y$ resides on the $i^{th}$ processor. Now let $\hat{y}_t$ be two separate moving arrays defined by

$$\hat{y}_1 = (f_{1,4}^2, f_{2,5}^3, f_{3,6}^4, f_{4,1}^5, f_{5,2}^6, f_{1,3}),$$

$$\hat{y}_2 = (f_{1,4}^3, f_{2,5}^5, f_{3,6}^6, f_{4,1}^1, f_{5,2}^2, f_{1,3}^3).$$

The array $\hat{y}_2$ is then inversely shifted by a stride $a_2 = 2$ and added to $\hat{y}_1$. Thus,

$$\hat{y}_1 = (f_{1,4}^2, f_{2,5}^3, f_{3,6}^4, f_{4,1}^5, f_{5,2}^6, f_{1,3}).$$

The array $\hat{y}_1$ is now inversely shifted by a stride $a_1 = 1$ and added to $y$ to yield the final resulting vector

$$y = (f_{1,4,5,6}, f_{1,3,4,5,6}, f_{1,2,4,5,6}, f_{1,2,3,5,6}, f_{1,2,3,4,6}, f_{1,2,3,4,5}).$$
This example illustrates the hypersystolic algorithm which only required 4 communication shifts as opposed to 5 shifts for the standard systolic method. The general hypersystolic algorithm is

1. For a given array \( x \) of length \( n \), \( k \) copies are generated by shifting the original array \( x \) \( k \) times by strides \( a_1, \ldots, a_k \) and storing the resulting arrays as \( \hat{x}_t \) for \( t = 1, 2, \ldots, k \).

2. The required results are successively computed and added to \( y \) and the \( k \) moving arrays \( \hat{y}_t \) (avoiding repetitions).

3. Finally, applying the inverse shift sequence, the arrays \( \hat{y}_t \) are shifted back and corresponding entries summed up to build the elements of the final array \( y \).

One now has the problem of determining \( A_k \) such that

- all pairings of data elements occur
- the number of shifts \( k \) is minimised
- \( r \) is minimised

A lower bound on the minimal possible \( k \) has been derived in Lippert et al. Since the possible number of combinations between elements of a given column of \( C \) is \( k+1C_2 \) then it follows that

\[
n^{(k+1)C_2} \geq \frac{n(n-1)}{2}.
\]

Therefore,

\[
k \geq -\frac{1}{2} + \sqrt{n - \frac{3}{4}}.
\]  

Hence, the computational complexity for the hypersystolic algorithm must be at least \( O(n^{3/2}) \). Furthermore, Lippert et al. shows that a regular base defined by

\[
A_k = (1, 1, \ldots, 1, K, K, \ldots K)
\]

is a nearly optimal choice and suitable for large processor number, where there are \( K \) of the 1's and \( (K - 1) \) of the \( K \)'s in this sequence and \( K = \sqrt{n/2} \).\(^2\)

\(^2\) \( K \) only makes sense for integer values and is ensured if \( n/2 \) is a square number. However, if \( n/2 \) is not a square number then \( K \) is chosen to be a suitable neighbouring integer value of \( \sqrt{n/2} \).
It follows that the number of shifts for a regular base is \( k = 2K - 1 \), and the communicational complexity is \( 2n(2\sqrt{n}/2 - 1) \).

With some experimentation Lippert et al. have derived some extremal (shortest) bases. For example, they found for \( n = 32 \) that \( A_6 = (1, 1, 1, 4, 4, 8) \) was an extremal base. Of course implementing an extremal base may not be feasible if the underlying topological communication structure of a parallel machine does not support strides of a certain length.

### 3.5.4 Implementation of the Direct Method N-body solver with the Hypersystolic algorithm

Clearly the simplest of N-body codes, namely NBODY1 (as described in Chapter 2.2), can be implemented with the hypersystolic algorithm if one adopts the same time-step for all particles and if the number of particles equals the number of processors. In this case suppose that the phase-space coordinates, forces and the 1st force derivatives are distributed across the processors. At time \( t \), therefore, processor \( i \) has the phase-space coordinates of particle \( i \) and the total force acting on particle \( i \) and its 1st derivative. Each processor computes its part of the particle prediction loop and thus the phase-space coordinates for particle \( i \) residing on processor \( i \) are predicted to the new time \( t + \delta t \). The hypersystolic algorithm is easily implemented as described above and consequently the total force (and its 1st derivative) on all particles due to all the other particles is computed and stored on that particle’s processor. Each processor simultaneously corrects its particle’s predicted phase-space coordinates, and so the time-stepping continues and the system evolves.

Unfortunately it is unlikely that \( N = n_p \). However, if \( N = m_q n_p \) for some integer \( m_q \) then the hypersystolic algorithm can still be implemented. For example consider \( N = 12, n_p = 6 \) and consequently \( m_q = 2 \). Instead of 1 fixed array \( \bar{x} \) there are \( m_q = 2 \) fixed arrays

\[
\bar{x}^1 = (1, 2, 3, 4, 5, 6), \\
\bar{x}^2 = (7, 8, 9, 10, 11, 12),
\]

which are distributed over the processors. Similar to the previous example, but with \( k = 4 \) and \( A_4 = (a_1^1, a_2^1, a_1^2, a_2^2) = (1, 1, 2, 2) \), these shifts act on both fixed arrays \( \bar{x}^1 \) and \( \bar{x}^2 \) according to the superscript value of \( a_t \). Furthermore, there are \( k = 4 \) resulting moving arrays \( \bar{y}_t^s \) where \( t = 1, 2, ..., k \), and \( m_q = 2 \) fixed resulting arrays \( y^s \) where the \( s \) denotes that the calculations are based on \( \bar{x}^s \) (or...
its moving equivalent). It follows that after 4 clock steps one has the matrix

\[
C = \begin{bmatrix}
1 & 2 & 3 & 4 & 5 & 6 \\
7 & 8 & 9 & 10 & 11 & 12 \\
2 & 3 & 4 & 5 & 6 & 1 \\
8 & 9 & 10 & 11 & 12 & 7 \\
4 & 5 & 6 & 1 & 2 & 3 \\
10 & 11 & 12 & 7 & 8 & 9 \\
\end{bmatrix}
\]

where the 1st two rows correspond to clock step 0 and rows 3 to 6 correspond to clock steps 1 to 4 respectively. It follows that after 4 clock steps we can compute

\[
y_1 = (f_1^{1}, f_2^{1}, f_3^{1}, f_4^{1}, f_5^{1}, f_6^{1}, f_7^{2}, f_8^{2}, f_9^{2}, f_{10}^{2}, f_{11}^{2}, f_{12}^{2}),
\]
\[
y_2 = (f_1^{7}, f_2^{7}, f_3^{7}, f_4^{7}, f_5^{7}, f_6^{7}, f_7^{8}, f_8^{8}, f_9^{8}, f_{10}^{8}, f_{11}^{8}, f_{12}^{8}),
\]
\[
y_3 = (f_1^{3}, f_2^{3}, f_3^{3}, f_4^{3}, f_5^{3}, f_6^{3}, f_1^{9}, f_2^{9}, f_3^{9}, f_4^{9}, f_5^{9}, f_6^{9}),
\]
\[
y_4 = (f_1^{11}, f_2^{11}, f_3^{11}, f_4^{11}, f_5^{11}, f_6^{11}, f_1^{12}, f_2^{12}, f_3^{12}, f_4^{12}, f_5^{12}, f_6^{12}).
\]

In constructing these arrays repeated terms have been avoided. The arrays \( y_3 \) and \( y_4 \) are now inversely shifted by a stride \( a_3 = a_4 = 2 \) and added to \( y_1 \) and \( y_2 \) respectively yielding

\[
y_1 = (f_1^{1}, f_2^{1}, f_3^{1}, f_4^{1}, f_5^{1}, f_6^{1}, f_7^{2}, f_8^{2}, f_9^{2}, f_{10}^{2}, f_{11}^{2}, f_{12}^{2}),
\]
\[
y_2 = (f_1^{7}, f_2^{7}, f_3^{7}, f_4^{7}, f_5^{7}, f_6^{7}, f_7^{8}, f_8^{8}, f_9^{8}, f_{10}^{8}, f_{11}^{8}, f_{12}^{8}).
\]

These two arrays are inversely shifted by a stride \( a_1 = a_2 = 1 \) and added to \( y_1 \) and \( y_2 \) to give the two final resulting arrays

\[
y_1 = (y_1, y_2, y_3, y_4, y_5, y_6),
\]
\[
y_2 = (y_7, y_8, y_9, y_{10}, y_{11}, y_{12}).
\]

Hence, in reference to the N-body problem, the total force on each particle has now been computed. In total, for this example, the hypersystolic method incurs 8 communication shifts as opposed to the 10 communication shifts with the standard systolic method.

In general terms:

- For a given array \( x \) of length \( N = m_q n_p \) this array is partitioned into \( m_q \) arrays \( x^s \) each containing \( n_p \) elements and where \( s = 1, 2, \ldots, m_q \). Each
element of $x^s$ is resident on a different processor. In total $k/m_q$ copies are generated by shifting the original arrays $x^s$, $k/m_q$ times by a stride $a_t^s$ and storing the resulting moving arrays $\tilde{x}_t^s$ for $t = 1, 2, ..., k$.

- The results $y_i$ are successively computed and added to the $k$ resulting moving arrays $\tilde{y}_t^s$ (avoiding repetitions in the summing).

- Applying the inverse shift sequence, the arrays $\tilde{y}_t^s$ are shifted back and corresponding entries summed up to build the elements of the final array $y^s$.

It is unclear how an $N$-body solver which employs the Ahmad-Cohen neighbour scheme (see Chapter 2.3) could be implemented with the hypersystolic algorithm since the algorithm is set up to compute the total force on a given particle and not a partial force contributed by the particle's neighbours. As a consequence, no attempt at implementing NBODY6++ with the hypersystolic algorithm is discussed.

### 3.6 Parallelisation of a Tree code

In recent years a number of ways of developing parallel versions of a tree code have been developed (Salmon 1991, Olson & Dorband 1994, Olson & Packer 1996, Antonuccio-Delogu et al. 1998, Yahagi Mori & Yoshii 1999, Vittor & Carpintero 2000, Lia & Carraro 2000). These are based on several different architectures and applications. Here we describe one other, to illustrate the various issues that must be faced.

A **Work and Data SHaring Parallel Tree** $N$-body code (denoted by WDSH-PTc) has been developed and implemented on a CRAY T3D parallel machine with CRAFT parallel directives (Becciani et al. 1997). Three phases in the code structure have been identified as requiring efficient parallelisation:

- Tree formation and calculation of cell properties: parallelisation is achieved by sharing the do loop structures among the processors (by means of CRAFT parallel directives), whereby the data involved in the loop is distributed across the processors accordingly

- Tree inspection (tree-walk), force evaluation and update of position of bodies (the FC (force computation) phase): this part of the code consumes more than 80% of the total computation time in the serial code and thus
needs to be efficiently parallelised. The particles’ position vectors are
distributed across the processors, and so for a fixed time step algorithm, all
the particles are updated simultaneously across the processors by applying
the tree algorithm to each particle

* Dynamic load balance (DLB): a load imbalance arises when a small num-
ber of processors, most probably during the tree-walk, spend more time
processing than all other processors. Consequently the code will run at
the speed of the slowest processors. The DLB routines help to avoid such
load imbalances by performing a load redistribution among the proces-
sors. This is achieved by forcing each processor to execute the FC phase
for a fixed number of particles in the local memory, denoted by $N_{LM}$ and
defined as

$$N_{LM} = \frac{N}{n_p} \times \psi$$

where $\psi$ is a constant ranging from 0 to 1 and $n_p$ is the number of pro-
cessors. The FC phase for all the remaining particles $N_{free}$, where

$$N_{free} = n_p \times \left(\frac{N}{n_p} \times (1 - \psi)\right)$$

is executed by all the processors that have finished the FC phase for
$N_{LM}$ particles. Hence, a value of $\psi$ less than one gives an automatic
DLB mechanism to the system. Furthermore, if $\psi$ is equal to zero then
the system has the maximum load balance and the maximum degree of
autonomy in the FC phase.

Much experimentation has been done on the WDSH-PTc to find the optimal
distribution of:

1. tree data (containing cell information: COM, daughter cells, etc.)
2. particle data (particle position, particle velocity, etc.)

Firstly, one possible approach considered for tree data distribution is based
on the so-called “Locally Essential Tree”, henceforth known as LET (Dubinski
1996), where each processor builds a “local tree” for the particles assigned to it.
The force acting on each particle is then computed after all the processors have
assembled together the pointers to all cells on other processors which contribute
to the force on their particles. Unfortunately, this approach suffers from the
fact that memory requirements grow very quickly with the number of particles.
As a consequence, the approach adopted for the WDSH-PTc is to find the optimal distribution of tree cells, whereby a single tree is shared by all processors. After experimentation it was found that a fine grained distribution (CDIR$ SHARED POS_CELL(:BLOCK(1,:))) (see section 3.1.2) proves to be optimal since this ensures that each processor has the same number of cells. In other words, the cells are numbered progressively starting from the root cell, and this fine grained distribution ensures that all the processors spend, on average, an equal amount of time in tree data access. Secondly, a course grained distribution (CDIR$ SHARED POS_CELL(:BLOCK(N/n_p,:))) (see section 3.1.2) of particle data yields the best performance, since the closest particles are located on the same processors. (The particles are initially labelled in such a way that bodies near in space are labelled with nearby integers, and if necessary during execution the particles may be re-labelled and re-distributed so as to ensure that the closest particles are located on the same processor or nearby processors.)

Performance on the CRAY T3D of the WDSH-PTc is comparable to that of the LET code, but the memory requirement of the WDSH-PTc is lower than that of LET codes. As future work, a WDSH-PTc version will be developed by Becciani et al., using HPF-CRAFT (see previous section 3.4.2 and http://www.epcc.ed.ac.uk/t3d/documents/techreports/EPCC-TR98-02/index.html) and the shmem library (see http://www2.cray.com/cgi-bin/nph-dweb/dynaweb/all/2518.3.0/@Generic_BookView) on the CRAY T3E system. This new version will include an enhanced grouping strategy and periodic boundary conditions, and will allow execution of large simulations with very high performance.
Chapter 4

Galactic Oscillations

4.1 Introduction

A dynamical system such as a galaxy oscillates about a stable equilibrium if it is excited above that state, and although these modes of oscillation may be weakly damped, they may persist long enough to have observable consequences (Weinberg 1994b). These modes of oscillation are defined by expanding the density $\rho$ and potential $\phi$ in terms of spherical harmonics. For example, one might expand the potential as

$$\phi(r, \theta, \psi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} F(r, l, m) Y_l^m(\theta, \psi),$$

(4.1)

where $F$ is some function modelling $\phi$'s dependence on the radial spherical coordinate $r$, and $Y_l^m$ is the spherical harmonic defined by

$$Y_l^m(\theta, \psi) = \sqrt{\frac{(2l+1)(l-|m|)!}{4\pi(l+|m|)!}} P_l^{|m|}(\cos \theta) e^{im\psi} \times \left\{ \begin{array}{ll} (-1)^m & \text{for } m \geq 0, \\ 1 & \text{for } m < 0. \end{array} \right. \quad (4.2)$$

The variables $\theta$ and $\psi$ are the two angular coordinates in a spherical coordinate system, $P_l^{|m|}(\cos \theta)$ is the associated Legendre function (Binney & Tremaine 1994, p655), $l = 0, 1, 2, \ldots$ and $m = -l, -l + 1, \ldots, l - 1, l$. For the remainder of this chapter the presence of these modes in stellar systems is investigated. In particular, the "$l = 0$" and "$l = 1$" modes are discussed in sections 4.2 and 4.3 respectively.
4.2 The Fundamental Mode

4.2.1 Introduction

The fundamental mode, known also as the breathing mode or the "$l = 0$" mode in terms of spherical harmonics, manifests itself as a radial oscillation of the entire system. To investigate this mode a number of different quantities may be considered. Firstly, consider a roughly spherically symmetric system of $N$ equal mass points of mass $m$ with positions $r_i$ relative to the system’s centre of mass frame. A characteristic linear dimension $R$ of the system may be defined as (Chandrasekhar 1942)

$$R = -\frac{GM^2}{2U},$$

(4.3)

where $M$ is the total mass of the system and $U$ is the system’s potential energy. Under the particles’ mutual gravitational attractions, the scalar form of the virial theorem is

$$\frac{1}{2} \frac{d^2 I}{dt^2} = 2T + U = 2E - U,$$

(4.4)

where $I = \sum_{i=1}^{N} m_i |r_i|^2$ is the moment of inertia, $T$ is the system's kinetic energy and $E = T + U$ is the system’s total energy. Furthermore, the kinetic energy $T$ defines the rms velocity $V$

$$V = \sqrt{\frac{2T}{M}}.$$  

(4.5)

The virial equilibrium values for $V$ and $R$ are

$$V_0 = \sqrt{\frac{2|E|}{M}}, \quad R_0 = \frac{GM^2}{4|E|},$$

(4.6)

(4.7)

which in turn can be used to define the crossing time (see Chapter 1.2)

$$t_{cr} = \frac{2R_0}{V_0} = \frac{GM^{5/2}}{(2|E|)^{3/2}}.$$  

(4.8)

Furthermore the virial ratio is defined as

$$q(t) = \frac{T}{|U|}.$$  

(4.9)

and the structure parameter (David & Theuns 1989) as

$$\lambda(t) = \frac{S}{R}.$$  

(4.10)
where $S$ is another linear measure of the system given by

$$ S = \sqrt{\frac{I}{M}}. \tag{4.11} $$

As an example of the application of these formulae, consider a spherically symmetric system with a Gaussian distribution of density

$$ \rho(r) = \frac{N m}{a^3 4\pi^3/2} e^{-r^2/a^2}, \tag{4.12} $$

where $a = a(t)$ is a measure of the radius of the system. By the scalar virial theorem one is led to the following differential equation (Chandrasekhar & Elbert 1972)

$$ \frac{3}{4} N m \frac{d^2 a^2}{dt^2} = 2E + \frac{GN^2 m^2}{\sqrt{2\pi} a}. \tag{4.13} $$

By a suitable change of variable Chandrasekhar & Elbert solved Eq. (4.13) (under the assumption that the evolution proceeded homologously) and for $E > 0$ found that the system tends to disperse to infinity, whilst for $E < 0$, Eq. (4.13) yields a periodic solution and the system exhibited periodic radial oscillations of a finite amplitude. This is an approximation to the $l = 0$ mode.

This fundamental mode can also be predicted analytically for more general models (David & Theuns 1989) by rewriting the scalar form of the virial theorem (see Eq. (4.4)) as

$$ \frac{M d^2 S^2}{2 dt^2} = 2E + \frac{GM^2 \lambda}{2S}. \tag{4.14} $$

This differential equation is easily solved under the assumption that $\lambda$ remains constant and equal to some value $\lambda_0$, and the parametric solutions are

$$ t = \frac{T_p}{2\pi} (\xi + \xi_0 + A \sin(\xi - \eta)), \tag{4.15} $$

$$ S = S_0 (1 + A \cos(\xi - \eta)), \tag{4.16} $$

where

$$ S_0 = \frac{\lambda_0 GM^2}{4|E|}, \tag{4.17} $$

$$ T_p^2 = \frac{8\pi^2 S_0^3}{\lambda_0 GM}, \tag{4.18} $$

and $\xi_0$, $\eta$ and $A$ are constants determined by the initial conditions (in fact one of these is redundant, and we can set $\xi = 0$ at $t = 0$ without loss of generality).
The characteristic length $S$ oscillates around the value $S_0$ with amplitude $AS_0$ and period $T_p$.

David & Theuns state, that for an $N$-body system, if $N$ is sufficiently large and if the system is set into motion with $q \approx 0.5$ and with an isotropic velocity distribution which is cutoff well below the escape velocity, then any change in the structure (as reflected by the structure parameter $\lambda$) occurs fairly gradually, and becomes apparent on a time-scale of the order of the relaxation time $t_{\text{rlx}}$ (as defined in Chapter 1.2). However, if the virial ratio is significantly different from $q = 0.5$ then the system will undergo violent relaxation (see Chapter 1.4), and as a consequence a core-halo structure and possibly a number of escapers are produced on a time-scale of the order $T_p$, as given by Eq. (4.18). Therefore, since the radial oscillations are predicted under the assumption that $\lambda$ remains constant, one expects these oscillations to be weakened after a relaxation time $t_{\text{rlx}}$ (in a quiet system) or after a time of the order $T_p$ (in a violent system). The damping of radial oscillations due to core-halo formation can be understood dynamically as a result of the energy transfer from this fundamental mode to individual particles (Landau damping, see Binney & Tremaine 1994, p685) and phase mixing (Binney & Tremaine 1994, Chapter 4.7.2). Furthermore, radial oscillations which last longer than one period $T_p$ can thus be expected to occur only with relatively small amplitudes and in systems with a relaxation time of at least a few times $T_p$.

To test the analytically predicted virial radial oscillations David & Theuns performed numerical simulations based on two different $N$-body codes. The first is a Poisson solver (PS) code, originally developed to calculate the self-gravity in SPH calculations (Monaghan & Varnas 1988), and a predicted Euler method is employed to perform the time-stepping. This PS code provides a relatively quasi-collisionless model. The second is Aarseth’s standard NBODY2 code (Aarseth 1985), where the forces are softened with a softening parameter $4R/N$ (see Eq. (2.4)). The initial conditions were characterised by the initial virial ratio $q(0)$, and the $N$ particles were given random positions within a sphere of radius 1 and random velocities within a spherical shell between 10 and 11 units in velocity space. All dynamical quantities were scaled such that $M = N, G = 1$ and $E = -N^2/4$. Independent of these units the analytically predicted period of the radial oscillations from Eq.’s (4.8), (4.17) and (4.18) is

$$T_p = \pi \lambda_0 t_{\text{cr}}.$$  \hspace{1cm} (4.19)

David & Theuns performed only one experiment for a quiet system ($q(0) =$
0.5), which used the PS code with \( N = 2000 \). Although the experiment only lasted for 10 crossing times the experimental period matched the analytically predicted period well. For a moderate collapse situation \( (q(0) = 0.4) \) and with \( N = 2000 \), the PS code produced oscillations in \( S \) and \( R \) lasting for more than 20 crossing times, whilst with the NBODY2 code these corresponding oscillations were strongly damped after 8-10 crossing times (due to relaxation and core-halo formation). Furthermore it was found that the structure parameter \( \lambda \) exhibited a greater rate of change with the NBODY2 code as opposed to the PS code, since relaxation is stronger in the former, and consequently core-halo formation is more prominent and escapers more frequent than with the latter. This was reinforced further still when David & Theuns performed experiments for a violent collapse \( (q(0) = 0.2) \).

The fundamental mode has also been detected in \( N \)-body numerical simulations based on the standard \( n = 3 \) polytropic model and a conical model (Miller & Smith 1994). The \( N \)-body code was a particle-mesh code (see Chapter 2.6) with a cubic lattice, and the particle positions and velocities were advanced by a time-centred leapfrog integrator. Each simulation had 400384 particles, with 64 active grid points along each coordinate direction and each simulation lasted from 50 to 80 crossing times. Miller & Smith found that the oscillations in kinetic energy had a period of 3.32 crossing times (the fundamental mode) for initial conditions based on the conical model. These oscillations persisted beyond 80 crossing times with little damping (about 5% at the end of 80 crossing times). Plots of the Lagrangian radii containing \( 5/8 \) and \( 1/16 \) of the system’s total mass suggested the existence of two radial oscillations. The outer Lagrangian radii had a period close to the period of the oscillations in kinetic energy, whilst the inner Lagrangian radii had a period of 1.52 crossing times. The latter of these radial oscillations is referred to by these authors as the second radial mode, and they refer to both modes as normal modes.

Wachlin & Muzzio (1997) made the same assumption of self-similarity as David & Theuns, but linearised about equilibrium. Eq. (4.14) may be rewritten as

\[
\frac{1}{2} \frac{d^2 I}{dt^2} = 2E + \frac{C}{I^{1/2}},
\]

where

\[
C = \frac{GM^{5/2}\lambda}{2}.
\]

Consider perturbing the system slightly in such a way that the moment of inertia
is perturbed to $I = I_0 + \delta I$, where $I_0$ is the moment of inertia that the system has when it is in the equilibrium state. It follows from Eq. (4.20) that

$$2E + \frac{C}{I_0^{1/2}} = 0$$

(4.22)

and consequently

$$\frac{1}{2} \frac{d^2 \delta I}{dt^2} = 2E + \frac{C}{(I_0 + \delta I)^{1/2}}.$$  

(4.23)

Under the assumption that $\delta I \ll I_0$ this differential equation becomes approximately the harmonic oscillator equation

$$\frac{d^2 \delta I}{dt^2} + \frac{C}{I_0^{3/2}} \delta I = 0.$$  

(4.24)

Hence the solution consists of harmonic oscillations of angular frequency

$$\omega = \sqrt{\frac{C}{I_0^{3/2}}} = \sqrt{\frac{GM^2}{2R_0 I_0}},$$

(4.25)

where $R_0$ is the virial radius as defined by Eq. (4.7), and we have used Eq.'s (4.10) and (4.11). Note that $\omega = 2\pi/T_\nu$ (Eq. (4.18)), by Eq.'s (4.10) and (4.11).

Wachlin & Muzzio performed a number of numerical experiments based on a $n = 3$ polytropic model, where the Lane-Emden equation (Eq. (1.16)) is solved numerically to obtain the potential of the system. The initial perturbation resembles the poissonian noise present in an $N$-body system of 10000 particles, which should excite the fundamental mode. The perturbation particle method (see Chapter 2.4) was adopted, except that instead of perturbation particles, since the problem is radial, Wachlin & Muzzio used perturbation shells. The equations of motion were integrated by a Runge-Kutta algorithm (Fehlberg 1968) which incorporates a variable time-step. All the experiments lasted for 128 crossing times and the number of shells adopted varied from 1000 to 4000. Table 4.1 summarises the results of their experiments, where the frequency of oscillations of the kinetic energy (computed from the velocities of the shells) were estimated by Fourier analysis (see Press et al. 1996, Chapter 12.1). The theoretical predicted frequency (reciprocal period) for these radial oscillations is

$$f = \frac{\omega}{2\pi}.$$  

(4.26)

For the units chosen by Wachlin & Muzzio the theoretical frequency is $f = 0.21t_{cr}^{-1}$, which is reasonably close to the frequencies found in their numerical experiments.
Table 4.1: The frequency of oscillations of the kinetic energy.

<table>
<thead>
<tr>
<th>Number of shells</th>
<th>frequency (per crossing time)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>0.234</td>
</tr>
<tr>
<td>2000</td>
<td>0.195</td>
</tr>
<tr>
<td>4000</td>
<td>0.203</td>
</tr>
</tbody>
</table>

Softening of the gravitational potential (see Chapter 2.2.1) to improve the collisionless nature of the system can affect the global dynamics of the system (Gerber 1996). For example, Gerber performed numerical experiments based on the fundamental mode oscillations of the conical model described by the aforementioned Miller & Smith. To test the effect of softening on the frequency of these oscillations a Particle Mesh (PM) code was adopted (see Chapter 2.6). Gerber found that as the softening length is increased, the amplitude of oscillations of the kinetic energy increased and the frequency decreased.

Many other authors have investigated the fundamental mode including Sweatman (1993), Sridhar (1987,1989), Sridhar et al. (1989), Mathur (1990) and Palmer et al. (1989).

4.2.2 Numerical experiments

To investigate the fundamental mode of oscillations further a King model (see Chapter 1.8) was adopted with a central concentration of $W_0 = 1$. The system was initially "cooled" down in such a way that the virial ratio $q < 0.5$, and consequently the fundamental mode was excited. All dynamical quantities were scaled to $N$-body units (see Chapter 2.1.2); however for the illustrations presented in this chapter the time is expressed in units of the crossing time (Eq. (4.8)). The experiments were performed on the Waverley-P3 machine (see Chapter 3.1.1) or a workstation with a 400MHz UltraSPARC-IIi processor. The total energy of the system is conserved to better than 0.0001% for all the experiments performed.

Table 4.2: Summary of experiments performed, where $t_{crit}$ is the termination time (in units of the crossing time).

<table>
<thead>
<tr>
<th>Experiment</th>
<th>$q$</th>
<th>$N$</th>
<th>$\epsilon$</th>
<th>$t_{crit}$</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.4</td>
<td>500,2000,8000</td>
<td>$4R/N$</td>
<td>20</td>
<td>NBODY1</td>
</tr>
<tr>
<td>B</td>
<td>0.4</td>
<td>500,2000,8000</td>
<td>$R/N^{1/3}$</td>
<td>20</td>
<td>NBODY1</td>
</tr>
<tr>
<td>C</td>
<td>0.4</td>
<td>8000</td>
<td>0.1,...,0.0005</td>
<td>20</td>
<td>NBODY1</td>
</tr>
<tr>
<td>D</td>
<td>0.4</td>
<td>80000</td>
<td></td>
<td>110</td>
<td>SCF</td>
</tr>
<tr>
<td>E</td>
<td>0.499</td>
<td>8000</td>
<td>0.1</td>
<td>40</td>
<td>PPM</td>
</tr>
<tr>
<td>F</td>
<td>0.4</td>
<td>8000</td>
<td></td>
<td>20</td>
<td>NSG</td>
</tr>
</tbody>
</table>

72
The particular aim of these calculations is to study the effects of softening (see Chapter 2.2.1) on the frequency and amplitude of the fundamental mode, and the detection of this mode in quiet systems ($q \approx 0.5$). To this aim the following evolutionary models (as described in Chapter 2) have been adopted:

- **NBODY1** - This is a direct method $N$-body solver (see Chapter 2.2.1) which uses a softened gravitational potential and solves Eq. (2.2) directly. The softening parameter $\epsilon$ is varied according to Eq.'s (2.3) and (2.4), and the effect on the $l = 0$ mode is observed.

- **PPM** - The Perturbation Particle Method (see Chapter 2.4) is implemented with NBODY1 which in principle should reduce the effects of particle noise, making it particularly appropriate for the investigation of oscillations in quiet systems.

- **SCF** - This is a field code designed for systems with axial symmetry, and expands the potential and density in terms of Legendre polynomials (see Chapter 2.5). This code models an approximately collisionless system over the timescales considered.

- **NSG** - This is a “no self-gravity” code based on NBODY1, but the forces are computed from a fixed potential.

![Figure 4.1: Plot of $T$ against time (in units of $t_{cr}$), with $\epsilon = 4R/N$. The particle number is varied, $N = 500, 2000, 8000$ corresponding to the red, blue and green lines respectively: Experiment A.](image)

In total six separate experiments have been performed to investigate the fundamental mode of oscillation, and these are summarised in Table 4.2.
Figure 4.2: Plot of $S$ (green line) and the theoretical $S_{th}$ (blue dotted line) against time (in units of $t_{cr}$), with $\epsilon = 4R/N$: The particle number is varied, $N = 500, 2000, 8000$ corresponding to plots (a), (b) and (c) respectively: Experiment A.

The results of a moderate collapse situation corresponding to $q = 0.4$ are shown in Figures 4.1 to 4.6. Specifically Figure 4.1 plots the total kinetic energy of the system against time for $N = 500, 2000, 8000$ corresponding to experiment A. Clearly the fundamental mode is present and becomes increasingly less prominent as $N$ is decreased. The linear measure of the system $S$, Eq. (4.11), is plotted in Figures 4.2(a), (b) and (c) for various $N$ together with the
Figure 4.3: Plot of (a) $T$ against time (in units of $t_{cr}$) with $N = 500, 2000, 8000$ corresponding to the red, blue and green lines respectively, and (b) Lagrangian radii against time (in units of $t_{cr}$) for $N = 8000$, and the curves correspond to the mass fractions stated in the text. Both plots with softening $\epsilon = R/N^{1/3}$: Experiment B.

Theoretical linear measure, denoted by $S_{th}$, based on Eq.'s (4.15) to (4.18). In order to derive the linear measure $S_{th}$ one needs to determine the constants $\xi_0$, $A$ and $\eta$. One chooses $\xi = 0$ at $t = 0$ which leads to

$$\xi_0 = A \sin \eta,$$

$$A = \sqrt{a^2 + b^2},$$

$$\eta = \tan^{-1}(b/a),$$

where

$$a = \frac{S|_{t=0}}{S_0} - 1,$$

$$b = \frac{T_p S|_{t=0}}{2\pi S_0^2} \frac{dS}{dt}|_{t=0}.$$
Figure 4.4: Plot of $S$ (green line) and the theoretical $S_{th}$ (blue dotted line) against time (in units of $t_{cr}$), with $\epsilon = R/N^{1/3}$. The particle number is varied, $N = 500, 2000, 8000$ corresponding to plots (a), (b) and (c) respectively: Experiment B.

The theoretical linear measure $S_{th}$ should be determined by the initial conditions based on a galaxy represented by a realistic number of stars. The values $\xi_0$, $A$ and $\eta$ were determined from the initial conditions for $N = 40000$ with $q = 0.4$ (increasing $N$ further does not lead to a significant change in $S_{th}$). The experimental variation of $S$ is well matched (with respect to frequency) to $S_{th}$ for $N = 500$ (Figure 4.2(a)), but only for a single period $T_p$ (Eq. (4.19)). As
Figure 4.5: Plot of (a) $T$ and (b) $S$, against time (in units of $t_{cr}$), for $N = 8000$ and $\epsilon = 0.1, 0.05, 0.025, 0.0005$ corresponding to the green, blue, red and pink lines respectively: Experiment C. 

$N$ is increased to 2000 (Figure 4.2(b)) the “trough” of the second oscillation is almost aligned with the corresponding “trough” in $S_{th}$. With $N = 8000$ (Figure 4.2(c)) the experimental and theoretical linear measures are closely aligned up to the fifth trough.

In contrast to experiment A which adopted a softening parameter of $\epsilon = 4R/N$, experiment B adopts the larger softening radius $\epsilon = R/N^{1/3}$. Figure 4.3(a) shows the manifestation of the fundamental mode for experiment B in the form of oscillations in the system’s total kinetic energy. The global nature of the oscillations is seen in the behaviour of the Lagrangian radii containing 20, 40, 60 and 80 percent of the system’s total mass (Figure 4.3(b)). Figure 4.4 compares the experimental $S$ with the theoretical $S_{th}$ for various $N$. For $N = 500$ (Figure 4.4(a)) the experimental linear measure $S$ matches well, with
respect to frequency, to the theoretical linear measure up to the second trough in the oscillation; thereafter the oscillation is heavily damped. As \( N \) is increased to 2000 (Figure 4.4(b)) the peaks in \( S \) are closely aligned to those in \( S_{th} \) at least up to peak 4. Increasing \( N \) to 8000 (Figure 4.4(c)) yields a very good match of \( S \) to \( S_{th} \), to beyond 6 periods of the oscillation.

The effects of softening on the fundamental mode of oscillations are investigated further by fixing the particle number \( N = 8000 \) and allowing the softening
parameter to vary (experiment C, see Figure 4.5). As the softening parameter is decreased the amplitude of the initial peak in the kinetic energy (Figure 4.5(a)) is increased and the mean kinetic energy about which the system oscillates is greater. The effect of softening on the initial peak/trough is also evident in the system’s linear measure $S$ (Figure 4.5(b)).

Since the fundamental mode manifests itself as spherical radial oscillations then one may employ the evolutionary model designed for systems with axial symmetry, specifically the aforementioned SCF code. The results for this approximately collisionless system, experiment D, can be seen in Figures 4.6(a) to (c). The experiment lasts beyond 100 crossing times and the period of the experimental linear measure $S$ closely resembles that of its theoretical counterpart (Figure 4.6(a)). This mode is also apparent in the kinetic energy (Figure 4.7): Plot of (a) $T_{pp}$ (green line) and $I_{pp}$ (blue line) for $N = 8000$ and $\epsilon = 0.1$, (b) $I_{pp}$ against time (in units of $t_{cr}$) for $N = 8000$ and $\epsilon = 0.1, 0.05, 0.025$ corresponding to the green, blue and red lines respectively: Experiment E.
4.6(b)) and Lagrangian radii containing 20, 40, 60 and 80 percent of the system’s total mass (Figure 4.6(c)).

To investigate the “l=0” mode for a quiet start one is led to experiment E. The results of employing the PPM evolutionary model can be observed in Figures 4.7(a) and (b). The total kinetic energy and moment of inertia of the perturbation particles is defined by

\[
T_{pp} = \sum_{i=1}^{N} \frac{m_i}{2} v_i^2, \quad (4.32)
\]

\[
I_{pp} = \sum_{i=1}^{N} m_i r_i^2, \quad (4.33)
\]

where \(m_i\) is the mass of a perturbation particle (which may be negative) based on Eq. (2.23). The experimental frequency determined from either of the quantities \(T_{pp}\) or \(I_{pp}\) (Figure 4.7(a)) is \(0.282t^{-1}\) which compares reasonably well to the theoretical frequency of \(0.32t^{-1}\) based on Eq. (4.26), which in turn makes use of the approximation Eq. (4.25). Figure 4.7(b) shows the effects of softening on the oscillations of \(I_{pp}\); if the softening parameter is too small then the oscillations begin to break down.

### 4.2.3 Discussion and conclusions

The effects of relaxation are evident in experiment A, where the softening is \(4R/N\). For \(N = 500\) one would expect to see a number of escapers on the timescale of approximately \(10.1t_{cr}\) (based on Eq. (1.1)); this is clearly the cause of the increase trend of \(S\) in Figure 4.2(a). As \(N\) is increased the rate of increase in \(S\) is reduced (Figures 4.2(a) and (b)), since relaxation is weakened (and therefore the rate of escape is reduced).

The theoretical oscillations \(S_{th}\) are predicted under the assumption that the structure parameter \(\lambda\) remains constant, and thus the smaller the rate of change of \(\lambda\) with respect to time then the more prominent the oscillations will be in the experimental linear measure of the system \(S\). This is clearly observed in Figure 4.8(a), which shows the structure parameter varying with time for various \(N\), and should be studied in conjunction with Figure 4.2. The larger \(N\), and consequently the weaker the relaxation, the smaller the rate of change of \(\lambda\).

Choosing a larger softening parameter \(\epsilon = R/N^{1/3}\) (experiment B) suppresses relaxation further for the same particle number as compared to experiment A.
Figure 4.8: Plot of $\lambda$ for (a) $\epsilon = 4R/N$ and $N = 500, 2000, 8000$ (red, blue and green lines respectively) (b) $\epsilon = R/N^{1/3}$ and $N = 500, 2000, 8000$ (red, blue and green lines respectively), and (c) $N = 8000$ and $\epsilon = 0.1, 0.05, 0.025, 0.0005$ (green, blue, red and pink lines respectively) against time (in units of $t_{cr}$).

As a result, the experimental oscillations in $S$ (Figure 4.4) are more prominent than those observed in experiment A. As expected, this choice of a larger softening will result in a smaller rate of change in $\lambda$ (Figure 4.8(b) compared with Figure 4.8(a)) since relaxation has been weakened. The effect of softening, for fixed $N$, on the structure parameter is observed in Figure 4.8(c); as $\epsilon$ is increased (thus improving the collisionless behaviour of the system), $\lambda$ remains closer to
its initial constant value of $\lambda_0$, and the experimental radial oscillations become more prominent (Figure 4.5). Increasing the softening parameter for fixed $N$ weakens the gravitational field of the system, and as a consequence the initial collapse of the system is less pronounced, and the initial peak in the kinetic energy decreases (Figure 4.5(a)). Furthermore, the kinetic energy will oscillate about a greater value as the softening is decreased since the gravitational field is strengthened.

The radial oscillations may be damped by a number of different processes including phase mixing (see Chapter 1.4), Landau damping (Binney & Tremaine 1994, Appendix 5A) and relaxation. The dominant process causing the damping of radial oscillations in experiments A to E is phase mixing, which is present even in systems without self-gravity. This is demonstrated in experiment F, which evolves the system without self-gravity. The heavy damping of $S$ is observed in Figure 4.9. This damping is presumably also present in a self-gravitating system (Figure 4.9, Experiment A). A greater period and amplitude in the oscillation of $S$ with the self-consistent model in comparison to the fixed potential model (NSG) is also observed. Sweatman (1993) also found this in some experiments with the Plummer model. Landau damping (which manifests itself as the collective effects of the system's self-gravity) and relaxation may also, but to a lesser extent, damp the radial oscillations. For an isotropic Plummer model Palmer et al. (1989) found that the radial oscillations died away by Landau damping.

The numerical experiments based on the SCF evolutionary model compare
well to the theoretically predicted frequencies for the fundamental mode. Calculation of a power spectrum (see Press et al. 1996, Chapter 12.1) shows that the frequency of the experimental \( S \) is \( 0.31t_{cr}^{-1} \) (based on the empirical data observed in Figure 4.6) which is within 4% of the theoretical frequency of \( 0.32t_{cr}^{-1} \) based on Eq. (4.19).

4.3 The Sloshing Mode

4.3.1 Introduction

The "sloshing" mode or "l=1" mode manifests itself as a shift in the system's density centre (Casertano & Hut 1985),

\[
\tau_{\text{dens},j} = \frac{\sum_{i=1}^{N} \tau_{i} \rho_{j}^{(i)}}{\sum_{i=1}^{N} \rho_{j}^{(i)}}
\]  

(4.34)

where \( \rho_{j}^{(i)} \) is the density estimator of order \( j \) around the \( i^{th} \) particle, which slowly revolves or oscillates around the initial centre. Studying modes of oscillation in \( N \)-body simulations may be difficult since one must discriminate the behaviour of the mode from the effects of the numerical method. As observed in the previous section, the \( l=0 \) mode is affected by softening and relaxation, and it is clear that, if the amplitude of the oscillation is small, then this mode may be undetectable due to particle noise. For these reasons it is important to also study modes theoretically.

To this aim, a semi-theoretical method for investigating the dispersion relation for spherical stellar systems and for locating weakly damped modes has been developed by Weinberg (1994b) and applied to King models of varying concentration. In particular, Weinberg investigated the \( l = m = 1 \) mode and found that these are very weakly damped with periods \( P = 78.5, 23.1, 18.7, 12.3 \) (in units of the crossing time) corresponding to the King models \( W_0 = 3, 5, 6, 7 \) respectively. The predictions were partially verified by Weinberg, who performed an \( N \)-body direct method simulation, but only for a very short execution time (3 crossing times), and he cautioned that the mode is sensitive to features such as softening and relaxation.

4.3.2 Numerical Experiments

In an attempt to verify the predicted period of the \( l = 1 \) mode, as derived by Weinberg (1994b), a number of numerical simulations have been performed on
Figure 4.10: The initial conditions are constructed by beginning with a King $W_0 = 5$ model in virial equilibrium and shifting a number of particles with $z > 0$ to $z < 0$ (within a radius $r_d$ of the centre of the cluster); this shifts the density centre and the centre of mass (COM). To fix the COM at the origin a number of particles at large radii are shifted from $z < 0$ to $z > 0$. The resulting initial system is shown above. The number of particles in each region is denoted by $N^-$, $N^+$, respectively.

The King $W_0 = 5$ model. To excite the $l = 1$ mode, a King $W_0 = 5$ model is modified as depicted in Figure 4.10. If the $l = 1$ mode has been excited then the density centre of the system will oscillate along the $z$-axis with a certain period. Unfortunately the computation of the density centre can be computationally expensive for sufficiently large particle number $N$, and so it is preferable to measure other quantities in which the $l = 1$ mode should manifest itself. For example, $N^-_l - N^+_l$ should exhibit the $l = 1$ oscillation, where $N^-_l$ and $N^+_l$ are the total numbers of particles with $r$ less than some fixed radius (relative to the fixed origin of the coordinate system) and such that the $z$-component of the position vectors are negative and positive respectively.

To evolve the system a number of different $N$-body evolutionary models have been adopted:

- G3NBODY1 - This is a direct method $N$-body solver (see Chapter 2.2.1) which uses a softened gravitational potential and solves Eq. (2.2). To maximise particle number this code is executed on the GRAPE-3 (see section 3.1) at Edinburgh University.

- SCF - This is a field code designed for systems with axial symmetry, and
expands the potential and density in terms of Legendre polynomials (see Chapter 2.5). This code simulates an approximately collisionless model over the timescales considered.

- NSG - This is a "no self-gravity" code based on NBODY1, and the forces are computed from a fixed potential.

Table 4.2: Summary of experiments performed, where \( r_d \) is the radius illustrated in Figure 4.10, \( r_{\text{king}} = 4.35 \) is the King radius, \( r_c = 0.27 \) is the core radius and \( t_{\text{crit}} \) is the termination time (in units of the crossing time).

<table>
<thead>
<tr>
<th>Experiment</th>
<th>( N )</th>
<th>( r_d )</th>
<th>( N^- )</th>
<th>( N^+ )</th>
<th>( t_{\text{crit}} )</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>40000</td>
<td>0.5</td>
<td>6009</td>
<td>4066</td>
<td>100</td>
<td>G3NBODY1</td>
</tr>
<tr>
<td>A2</td>
<td>40000</td>
<td>0.5</td>
<td>6009</td>
<td>4066</td>
<td>100</td>
<td>SCF</td>
</tr>
<tr>
<td>A3</td>
<td>320000</td>
<td>0.5</td>
<td>48438</td>
<td>32609</td>
<td>100</td>
<td>SCF</td>
</tr>
<tr>
<td>B1</td>
<td>80000</td>
<td>( r_{\text{king}} )</td>
<td>45090</td>
<td>34910</td>
<td>70</td>
<td>SCF</td>
</tr>
<tr>
<td>B2</td>
<td>80000</td>
<td>( r_{\text{king}} )</td>
<td>45090</td>
<td>34910</td>
<td>43</td>
<td>NSG</td>
</tr>
<tr>
<td>C1</td>
<td>160000</td>
<td>( r_c )</td>
<td>6653</td>
<td>4443</td>
<td>70</td>
<td>SCF</td>
</tr>
<tr>
<td>C2</td>
<td>500000</td>
<td>( r_c )</td>
<td>20748</td>
<td>13848</td>
<td>70</td>
<td>SCF</td>
</tr>
</tbody>
</table>

A number of different experiments have been performed to investigate the \( l = 1 \) mode, and these are summarised in Table 4.2 (\( N \) needs to be larger than in the previous \( l = 0 \) mode experiments since the core of the cluster needs to be modelled as realistically as possible). Initially the investigation began by using \( N = 40000 \) particles and a direct method as the evolutionary model, which was implemented (by D.C. Heggie) on the GRAPE-3 (experiment A1). In the initial conditions the density centre was shifted by moving particles from \( z > 0 \) to \( z < 0 \) within a fixed radius \( r_d = 0.5 \) of the origin. This modified system, in which the initial shift in the \( z \)-component of the density centre was \(-0.06\), was evolved for 90 crossing times. The \( z \)-component of the density centre moved quickly back to the origin where its subsequent motion became indistinguishable, due to particle noise, from that of the \( y \)-component of the density centre (see Figure 4.11(a)). To reduce the effects of particle noise on the results it would be necessary to increase the particle number \( N \) significantly.

Unfortunately the maximum number of particles supported on the GRAPE-3 is \( N = 65K \) (without severe degradation of its efficiency); thus the SCF code was then adopted, as this allowed a greater number of particles to be used for the simulations. As a verification, the same initial conditions for the GRAPE-3 experiment A1 was adopted for the SCF simulation (experiment A2), and the
results were found to be comparable (see Figure 4.11(b)). The amplitude of the oscillations in the SCF experiments may be slightly larger than those of the GRAPE experiments. This could be attributed to halo formation.

![Graph](image)

**Figure 4.11:** Plot of (a) the y-component (green line) and z-component (blue line) of the density centre against time (in units of $t_{cr}$) for experiment A1 (GRAPE, $N = 40000$), and (b) $N_f^{-}$ ($r < 0.5$) against time (in units of $t_{cr}$) for experiments A1 (GRAPE, $N = 40000$, green line) and A2 (SCF, $N = 40000$, blue line).

Increasing the particle number from $N = 40000$ to $N = 320000$ (experiment A3) will reduce the effects of particle noise on the results. To detect the $l = 1$ mode the quantity $\phi_1 (r = 2)$ was recorded and is plotted in Figure 4.12(a). The quantity $\phi_1 (r = 2)$ is defined by Eq. (2.43) with $l = 1$, where we set the
Figure 4.12: Plot of (a) $\phi_1(r = 2)$ against time (in units of $t_{cr}$) for experiments A2 (SCF, $N = 40000$, blue line) and A3 (SCF, $N = 320000$, green line), and (b) $N_i^-$ ($r < 0.5$) against time (in units of $t_{cr}$) for experiment A3 (SCF, $N = 320000$).

Factor $P_1$ to the value 1, and $r_i = 2$. The presence of two peaks in this quantity before 20 crossing times may be seen in Figure 4.12(a). After 20 crossing times the quantity $\phi_1(r = 2)$ exhibits no interesting behaviour, except for a general increase from $t_{cr} \approx 20$ to $t_{cr} \approx 70$ and a decrease from $t_{cr} \approx 70$ to $t_{cr} \approx 100$ for the $N = 40000$ experiment; this is due to a slow oscillatory behaviour of the $z$-component of the system's centre of mass (COM). This component of the COM differs from the $x$ and $y$-components because of a peculiar feature of the SCF code (see Chapter 2.5), which conserves the time-derivative of the
Figure 4.13: Plot of (a) $\phi_1(r = 2)$ against time (in units of $t_{cr}$) for experiment B1 (SCF, $N = 80000$), and (b) $N_1^- - N_1^+$ ($r < 0.5$) against time (in units of $t_{cr}$) for experiments B1 (SCF, $N = 80000$, green line) and B2 (NSG, $N = 80000$, blue line).

z-component of the COM much better than that of the other two components. In Figure 4.12(b) $N_1^-$ ($r < 0.5$) is plotted against time for experiment A3, and once again there is a rapid loss of particles from $z < 0$, $r < 0.5$ but no clear indication of the $l = 1$ oscillation.

In experiments B1 and B2 the initial conditions are generated by moving some particles from $z > 0$ to $z < 0$ within a fixed radius $r_d = r_{king}$ of the origin (where $r_{king} = 4.35$ is the King radius). Once again there are two peaks in $\phi_1(r = 2)$ prior to 20 crossing times for the SCF experiment (see Figure
Figure 4.14: Plot of (a) $\phi_1(r = 2)$ against time (in units of $t_{cr}$) for experiments C1 (SCF, $N = 160000$, green line) and C2 (SCF, $N = 500000$, blue line), and (b) $N_I^- - N_I^+ (r < r_c)$ against time (in units of $t_{cr}$) for experiment C2 (SCF, $N = 500000$, line).

4.13(a)), and these are also clearly present in $N_I^- - N_I^+ (r < 0.5)$ (see Figure 4.13(b)). Beyond this time the behaviour of these quantities is noisy and unremarkable. Evolving the perturbed system without self-gravity one again observes two peaks in the quantity $N_I^- - N_I^+ (r < 0.5)$ prior to 20 crossing times (see Figure 4.13(b), experiment B2); after this time the oscillation is heavily damped probably due to phase mixing. The initial trough in the NSG experiment is much deeper than the corresponding SCF experiment (see Figure

89
Figure 4.15: Plot of $N_I^− - N_I^+$ ($r < 0.5$) against time (in units of $t_{cr}$) for (a) experiment C1 (SCF, $N = 160000$), and (b) experiment C2 (SCF, $N = 500000$).

4.13(b)) and also the subsequent peaks and troughs in the NSG experiment appear sooner than the corresponding peaks in the SCF experiment.

Altering the initial conditions further (experiments C1 and C2), the initial density centre is shifted by moving particles from $z > 0$ to $z < 0$ within a fixed radius $r_d = r_c$ of the origin (where $r_c = 0.27$ is the core radius). The quantity $\phi_1(r = 2)$ exhibits some fluctuations (see Figure 4.14(a)) but a periodic oscillation is not clearly observable. This is also the case for the quantity $N_I^− - N_I^+$ ($r < 0.5$) for $N = 160000$ and $500000$ (see Figure 4.15). A similar quantity $N_I^− - N_I^+$ ($r < r_c$) was also recorded (see Figure 4.14(b)); this exhibits a rapid
movement of particles out of the region restricted by $z < 0, r < r_c$ followed only by noisy fluctuations around zero.

### 4.3.3 Discussion and conclusions

In conclusion, the above experiments are inconclusive in verifying the predicted frequency of the $l = 1$ mode made by Weinberg (1994b). The $l = 1$ mode may at present be undetectable due to

1. the initial conditions not exciting this particular $l = 1$ mode in such a way that it is the dominant mode of oscillation;

2. the amplitude of the $l = 1$ mode is so small that a much larger number of particles need to be used to reduce the effects of particle noise on the results.

Furthermore, analysis of the data using power spectrums did not reveal the $l = 1$ oscillation, but perhaps, although not attempted by the author here, autocorrelations might (see Sweatman 1993 for more details on autocorrelations). However, Figure 4.13(b) suggests that the behaviour of these oscillations is significantly affected by self-gravity.

### 4.4 Conclusions

As observed from the experiments performed, a dynamical system, such as a galaxy, may oscillate about a stable equilibrium if it is excited above that state. These modes of oscillation have been investigated with a range of evolutionary models.

The simplest mode to detect and simulate was the "fundamental" mode, which manifested itself as a radial oscillation of the entire system. A King $W_0 = 1$ theoretical model was adopted to investigate this mode. The effect of adopting a softened gravitational potential to generate the forces between stars on the fundamental mode was investigated with a direct method $N$-body evolutionary model, and these results were compared to those which used an SCF (self-consistent field) type code. Similar to the results of Gerber (1996), as the softening length is increased then there is an increase in the period of the oscillation. The presence of small amplitude fundamental mode oscillations is detectable when the perturbation particle method was used, which would
otherwise be undetectable for other $N$-body evolutionary models (due to particle noise). Furthermore, the source of heavy damping in direct method $N$-body simulations was found to be phase mixing. The experimental fundamental mode oscillations were found to match well with the theoretically predicted frequencies especially for the SCF experiments.

A more complicated mode, the "sloshing" mode or "$l = 1$" was also studied. This mode manifests itself in the density centre shifting or sloshing about. To investigate this mode a King $W_0 = 5$ model was adopted, and the results compared to an analytical predicted frequency (Weinberg 1994b). No evidence of the period Weinberg predicted was found in the experiments performed, although if they were present then they may have been masked by the oscillations observed.
Chapter 5

Binary Black Holes in Galactic Nuclei

5.1 Introduction

Many galaxies contain central black holes (Rees 1990, Magorrian et al. 1998), and in the event that two such galaxies merge, then the two black holes sink towards the centre of mass of the merger remnant due to dynamical friction and form a binary in the core. Observations (e.g. Wright et al. 1990) and $N$-body simulations (e.g. Okumura et al. 1991) have confirmed that some elliptical galaxies have evolved from mergers (Zepf 1997). In particular, Makino & Ebisuzaki (1996) performed numerical $N$-body simulations of the merging of two King $W_0 = 7$ models with central black holes; the results may explain the structure and size of the cores in bright elliptical galaxies. Elliptical galaxies account for 10% of galaxies in low-density regions to 40% in dense clusters of galaxies (Dressler 1980). Observational detection of binary black holes in galactic nuclei may manifest itself as a precession of radio jets emerging from the black holes; Owen et al. (1985) have detected a pair of twin radio jets originating in the apparent double nucleus of the radio galaxy 3C 75 which is associated with the central galaxy in Abell 400 (Figure 5.1).

Several stages have been identified by Begelman et al. (1980) relating to the dynamical evolution and formation of binary black holes in galactic nuclei, which is summarised as follows. Initially dynamical friction is extremely efficient and the cores merge and undergo violent relaxation (see Chapter 1.4) on the timescale of the order of a crossing time (see Chapter 1.2). Under the influence of dynamical friction the black holes sink towards the centre of the merger remnant and become bound to one another when the black hole separation becomes $r_b \sim (M_{BH}/M_e)^{1/3} r_c$, where $M_{BH}$ is the mass of the larger black hole,
Figure 5.1: Contour plot of 3C 75. This high resolution image at 6cm (4852 MHz) was cleaned and restored with a circular Gaussian beam with a FWHM (Full Width Half Maximum) of 1".4. The two point sources (corresponding to the two black disks at coordinates (02h55m03s,05°49.3') and (02h55m03s,05°49.7') respectively in the above plot) coincide with the two nuclei of the central galaxy in Abell 400. Each of the nuclei has a twin jet structure emanating from it (Owen et al. 1985).

$M_c$ is the total mass of the core (excluding the black holes) and $r_c$ is the core radius. Thereafter the black holes continue to spiral in towards one another and the binary becomes harder. The binary effectively expels low-angular-momentum stars which interact with it when the black hole separation $R_{BH}$ shrinks to

$$r_{lc} = \left(\frac{M_{BH}}{M_c}\right)^{3/4} r_c. \quad (5.1)$$

At this point loss cone depletion (the depletion of low-angular-momentum stars that can interact with the binary) affects the hardening rate and merger time of
the binary black holes. With sufficient repopulation of low-angular-momentum stars, diffusing into the loss cone in the two-body relaxation timescale of the core, the binary becomes more tightly bound, and it becomes hard at a separation $r_h \sim r_h^2/r_c^2$. Once the binary has become sufficiently tightly bound then gravitational radiation shrinks the orbit further, and the evolution proceeds rapidly until coalescence. According to Begelman et al. the lifetime of a central black hole binary is much longer than the Hubble time ($\sim 10^{10}$ yr), assuming the binary orbit is circular and assuming that the binary does not wander during loss cone depletion.

Fukushige et al. (1992) modelled the early evolution of a black hole binary undergoing dynamical friction by Chandrasekhar’s simple analytic formula (Chandrasekhar 1943, Binney & Tremaine 1994), and performed numerical integrations of the equal mass black hole orbits using a fourth-order Runge Kutta integrator (Press et al. 1996, Chapter 16). The force on a black hole is computed by adding the force from the other black hole to the dynamical friction force; the gravitational force from the core (consisting of a uniform distribution of field stars) is ignored since it is assumed that the black holes are sufficiently close to one another, and the effects of self-gravity amongst the stars are not considered. The initial conditions of the black holes were such that the eccentricity is $e = 0.7$, the semi-major axis of the initial orbits is $a = 0.29$ and the mass of each black hole is $M_{BH} = 1$ (with $G = 1$, and the units of length and mass were 100 pc and $10^8 M_\odot$ respectively). The numerical experiments were terminated once the pericentric distance reduced to 100 times the Schwarzschild radius of the black holes. Fukushige et al. found that the time taken by the black hole binary to merge is $\sim 10^7$ yr, significantly less than the Hubble time. Furthermore, the minimal orbital speed occurs at apocentre when dynamical friction is most effective, resulting in a highly eccentric binary orbit.

Makino et al. (1993) modelled the evolution of two black holes (equal mass) in a merger remnant by adopting a direct method N-body solver implemented, in part, on the GRAPE-3 (see Chapter 3.1). The force between a field star and a field star was softened with a softening length of $\epsilon = 0.025$, and all other force interactions were unsoftened. The force on each field star from the other field stars was computed on the GRAPE-3 and its motion was integrated using a 2-step predictor-corrector scheme; the force contributions from the black holes on a field star were integrated with greater accuracy using a 4th-order Hermite integrator (see Chapter 2.2, 2.3, 3.2) on the host computer. The force on a black hole from the field stars was computed on the GRAPE-3, with the exception
of close neighbours and the other black hole, and all these contributions were computed and integrated on the host computer with greater accuracy using the Hermite integrator. The field stars were initially distributed according to a Plummer model (see Chapter 1.8), and the black holes initially placed in symmetric positions with respect to the centre at a radius of 0.7 (all quantities having been scaled to N-body units, see Chapter 2.1.2) together with initial velocities tangential to the circle upon which the black holes lay and opposite in direction to one another. In total $N = 16384$ field stars were used to model the galaxy of mass $M = 1$. Makino et al. performed a variety of experiments allowing $M_{BH} = 0.01, 0.02, 0.04, 0.08, 0.16$, initial speed of the black holes $v_0 = 0.5, 0.125, 0.03125$ and terminated the experiment prior to 30 crossing times.

Makino et al. found that the eccentricity of the binary remained roughly constant once the binary had formed, and depended strongly on the initial speed, $v_0$, of a black hole, and weakly on the black hole mass. In fact for $M_{BH} = 0.04$ and $v_0 = 0.5$ then $e = 0.23$, whilst in contrast for $M_{BH} = 0.04$ and $v_0 = 0.03125$ then $e = 0.996$. The dependence of the eccentricity on $v_0$ has been attributed to core rotation. To elaborate, if $v_0$ is large enough then the core starts to rotate due to a back reaction from the black holes, and this in turn reduces the angular momentum transfer rates between the binary black holes in the core and the surrounding field stars; it is this which Makino et al. suggest restricts the eccentricity from becoming very large. Also, the black hole binary continued to become harder beyond Begelman’s loss cone radius (Eq. (5.1)) to at least $0.1r_{lc}$, and that the binding energy per unit mass becomes $\geq 10$ times larger than the kinetic energy per unit mass of field stars. They found that the evolution becomes dominated by gravitational radiation after about $10^9$ yr. Thus, Makino et al. concluded that the binary black holes merge on a timescale shorter than the Hubble time scale.

Makino (1997) performed similar experiments to Makino et al. (1993) but on the GRAPE-4. The maximum particle number was increased to 262144 and two King $W_0 = 7$ models with central black holes were merged. The galaxies were initially on parabolic orbits with periastron distance equal to 1 and the initial separation was 10 (N-body units). The black hole masses were equal to 0.03125 and the system was integrated for 21 crossing times. Makino found that the evolution timescale of the binary black hole was independent of the particle number $N$ until the semi-major axis reached a certain critical value; as a result of the loss cone effect. This critical semi-major axis value was smaller than the one implied by Begelman et al. (1980), and the lifetime of a binary black
hole was found to be much longer than the Hubble time. This increase in the merger time scale may be due to a number of possible mechanisms, including the wandering of the binary away from the dense central part of the nucleus.

The binary black hole evolves because of close encounters, which must be integrated accurately, but the galaxy also evolves because of the collective response of the stars to changes in the potential. To suppress relaxation amongst the field stars and to satisfy the requirement that the stars be much less massive than the black holes, one requires a large number of stars. To this end, Quinlan & Hernquist (1997) computed the force on a field star from the two black holes by direct summation with softening (using NBODY1, see Chapter 2.2), and the force on a field star from the other field stars was computed through an expansion of the potential with coefficients $A_{ntm}$ that were updated with time self-consistently. In other words, the star/star interactions are updated using the SCF program of Hernquist & Ostriker (1992). The force on a black hole from the other black hole was computed directly without softening, and the force from the other stars was also computed directly but this force was softened (using NBODY1). This hybrid code combines the accuracy of a direct summation code together with the speed of a SCF code; and thus the number of stars could be reasonably large. The initial conditions were based on a Plummer model (see Chapter 1.8) and a Jaffe model, where the density distribution is

$$\rho(r) = \frac{3 - \gamma}{4\pi} \frac{Md}{r^\gamma (r + d)^{4-\gamma}}, \quad (5.2)$$

with $\gamma = 2$; here $d$ is a length scale close to the effective radius (the radius of the isophote containing half the total luminosity of the system). The masses of the stars had a mass spectrum decreasing towards the centre as $m \sim r_p^\lambda$ (but with the restriction that $m_{max} \leq 1000m_{min}$), where $r_p$ is the initial pericentre of the star and $\lambda$ is an exponent usually in the range $0.5 - 1.0$; this improved the statistical resolution towards the centre (i.e. near the binary black hole). The black holes had initial masses $M_{BH} = 0.01$ (all dynamical quantities having been scaled to $N$-body units) and initial phase-space coordinates $(r, v)$ and $(-r, -v)$ respectively, where $|r| = 0.5$ and $|v|$ was chosen so that the black holes were initially on circular orbits in the galactic potential.

Quinlan & Hernquist performed numerical experiments on the Jaffe model for $N = 10^5$ field stars, with a mass exponent $\lambda = 0.75$ and a softening parameter $\epsilon = 4/N$ (see Eq. (2.4)). The simulation lasted for 28 crossing times and the eccentricity at the end was $\leq 0.1$. The hardening slowed down noticeably after
1/a reached about 500 since the central density had reduced significantly (due to many ejected stars); thereafter the hardening continued due to the black hole binary wandering and interacting with stars. Quinlan & Hernquist found that the hardening rate of the binary decreased as $N$ is increased, and explained this as follows. As $N$ increases the fluctuations driving the wandering of the black holes decrease in amplitude. Thus the black holes are effectively trapped within the loss cone where the stellar density, and the hardening rate, are both relatively small. The distance to which the binary wanders does not continue to decrease indefinitely as $N$ is increased, however, since the restoring force which keeps the centre of mass of the binary at the centre weakens as the central density decreases; the cut-off point appeared to be $N = 10^5$ stars from empirical data. Quinlan and Hernquist also found (as did Makino et al. (1993)) that the eccentricity depended strongly on the initial orbital speeds of the black holes; they found that the eccentricity decreased as the initial orbital speeds of the black holes was increased. Also it was easier for a binary to form with a large eccentricity in an isotropic model without a density cusp (Plummer) than in an isotropic model with a density cusp (Jaffe), which is in good agreement with an earlier result that a density cusp lowers the eccentricity (Polnarev & Rees 1994).

5.2 Numerical Experiments

5.2.1 Introduction

In the research described here, the dynamical evolution of two black holes in a galaxy was re-investigated with the aid of numerical simulations. The dynamical effect which the merging of two galaxies had on the resulting merger remnant was not modelled, and for analytical convenience a Plummer model (see Chapter 1.8) was adopted to represent the initial structure of the merged galaxy. For comparative purposes, experiments were performed using both a simple analytic dynamical friction formula and a direct method $N$-body solver. As mentioned in the previous section, it is important to supress relaxation and to minimise the amount of noise in the results. To this end, in some simulations the Perturbation Particle Method (henceforth PPM, see Chapter 2.4) was adopted to evolve the galaxy self-consistently. The PPM is ideally suited to evolve systems which have undergone a small perturbation from equilibrium. However, it is unclear how large a perturbation the system can experience and
5.2.2 Dynamical quantities associated with planar bound orbits

In order to study the dynamical evolution of two black holes it was useful to measure certain quantities associated with one black hole relative to the other black hole; as depicted in Figure 5.2. Let $R_1$ and $R_2$ be the position vectors of black hole 1 and black hole 2 respectively. Similarly let $V_1$ and $V_2$ be the velocity vectors of black hole 1 and black hole 2 respectively. The separation vectors for the position and velocity of the two black holes are respectively $R_{BH} = R_1 - R_2$ and $V_{BH} = V_1 - V_2$. The eccentricity vector (Goldstein 1950) is defined to be

\[
\mathbf{e} = -\frac{R_{BH}}{R_{BH}} + \frac{V_{BH} \times (R_{BH} \times V_{BH})}{\mu}, \tag{5.3}
\]
where $R_{BH} = |R_{BH}|$, $\mu$ is $G$ times the sum of the black hole masses, and the eccentricity $e = |e|$. The binding energy of the binary black hole is

$$E_b = \frac{1}{2} V_{BH}^2 - \frac{\mu}{R_{BH}}$$

and the semi-major axis is

$$a = -\frac{\mu}{2E_b}.$$  \hspace{1cm} (5.5)

**5.2.3 Initial conditions and Evolutionary models**

The initial phase-space coordinates of the black holes were $(r, v)$ and $(-r, -v)$ respectively, where $|r| = 0.5$ and $|v|$ was typically chosen so that the black holes were initially on circular orbits in the galactic potential and thus $|v|$ equals

$$v_c = \sqrt{\frac{d\phi}{dr}},$$

where $\phi$ is the Plummer potential (see Chapter 1.8). All dynamical quantities have been scaled to N-body units (see Chapter 2.1.2) such that the total mass and energy of the system (excluding the black holes) are $M = 1$ and $E = -1/4$ respectively. In the results described in this chapter, however, the simulation time $t$ (based on the aforementioned units) was expressed in units of the crossing time or dynamical time $t_{cr}$ (where $t_{cr}$ is defined in Eq. (4.8)). The masses of the black holes were equal and typically took the value $M_{BH} = 0.02$ (corresponding to a large black hole with a mass of a few percent of the bulge mass of a galaxy).

As mentioned briefly in section 5.2.1, three evolutionary models have been adopted to investigate the evolution of a binary black hole:

- **CDF** - Chandrasekhar’s Dynamical Friction formula (Chandrasekhar 1943, Binney & Tremaine 1994) is a simple analytic formula used to model the dynamical friction undergone by the black holes as they orbit in the galactic potential. The equation of motion of a black hole with position vector $R_1$ is

$$\frac{d^2 R_1}{dt^2} = \frac{G M_{BH} (R_2 - R_1)}{|R_2 - R_1|^3} - \frac{G M R_1}{(R_1^2 + b^2)^{3/2}} + a_{DF},$$ \hspace{1cm} (5.7)

where the first term on the right hand side is the acceleration contribution from the other black hole with position vector $R_2$, the second term is the acceleration contribution from the equilibrium model (the Plummer
model) and the third term is the dynamical friction due to the field stars. The dynamical friction term is based on Chandrasekhar’s formula and expressed as

\[ a_{DF} = -16\pi^2 \ln(\Lambda) G^2 m_f (M_{BH} + m_f) J_0 V_1 f(v_f) v_1^2 dv_f V_1, \]  

(5.8)

where \( m_f \) is the mass of a field star, \( V_1 \) \( (\leq \sqrt{-2\phi}) \) is the speed of the black hole with position vector \( R_1 \) and

\[ \Lambda = \frac{\kappa b_{\text{max}} v_0^2}{G(M_{BH} + m_f)}. \]  

(5.9)

Here, \( b_{\text{max}} \) and \( v_0 \) are respectively the maximum impact parameter and a field star’s typical velocity, and \( \kappa \) is some constant which is yet to be determined. For the Plummer model the values chosen were \( b_{\text{max}} = 1 \) and \( v_0^2 = 1/2 \). To evaluate the integral in Eq. (5.8) one requires the Plummer distribution function, which is given by

\[ f(r, v) = A \left( -\phi(r) - \frac{v^2}{2} \right)^{7/2}, \]  

(5.10)

where the constant \( A \) was chosen such that the integral of \( f \) over all phase-space is equal to the number of field stars \( N \). Thus,

\[ A = \frac{24\sqrt{2} N b^2}{7\pi^3}. \]  

(5.11)

The integral in Eq. (5.8) is simple, and has the following analytic form

\[ \int_0^{V_1} f(v_f) v_1^2 dv_f = 2\sqrt{2} A (-\phi)^5 \left( \frac{\beta_9}{10} + \frac{\beta_7}{80} + \frac{7\beta^5}{480} \right) + \frac{7\beta^3}{384} + \frac{7\beta}{256} + \frac{7\gamma}{256}, \]  

(5.12)

where

\[ \alpha = \frac{V_1}{\sqrt{-2\phi}}; \]
\[ \beta = \frac{\sqrt{-2\phi - V_1^2}}{\sqrt{-2\phi}}; \]
\[ \gamma = \sin^{-1} \alpha. \]

A fourth-order Runge-Kutta method (Press et al. 1996, Chapter 16) was used to integrate Eq. (5.7).
• NBODY1 - This is a direct method N-body solver (see Chapter 2.2.1). The program evolved the black holes and the stars according to the following equations of motion

\[ \frac{d^2 R_i}{dt^2} = \frac{GM_{BH}(R_j - R_i)}{|R_j - R_i|^3} + G \sum_j \frac{m_j(r_j - R_i)}{(\epsilon_{BS}^2 + |r_j - R_i|^2)^{3/2}}, \]

\[ \frac{d^2 r_j}{dt^2} = G \sum_k \frac{M_{BH}(R_k - r_j)}{(\epsilon_{BS}^2 + |R_k - r_j|^2)^{3/2}} + G \sum_j \frac{m_j(r_j - r_i)}{(\epsilon_{SS}^2 + |r_j - r_i|^2)^{3/2}}, \]

where \( R \) and \( r \) are the position vectors of a black hole and a star respectively, \( m_j \) is the mass of a field star, \( \epsilon_{SS} = 4/N \) is the softening parameter (see Chapter 2.2.1, Eq. (2.4)) for star/star interactions, and \( \epsilon_{BS} \) is the softening parameter for the black hole/star interactions; in what follows, various values were adopted for \( \epsilon_{BS} \). The force between the two black holes is unsoftened, and since the time scales considered were relatively short no special regularisation (see Chapter 2.3) for close encounters was undertaken.

• PPM - The Perturbation Particle Method (see Chapter 2.4) was implemented with NBODY1. The program evolved the black holes and the perturbation particles according to the following equations of motion

\[ \frac{d^2 R_i}{dt^2} = \frac{GM_{BH}(R_j - R_i)}{|R_j - R_i|^3} + G \sum_j \frac{m_j(r_j - R_i)}{(\epsilon_{BS}^2 + |r_j - R_i|^2)^{3/2}} - \frac{GMR_i}{(R_i^2 + b^2)^{3/2}}, \]

\[ \frac{d^2 r_j}{dt^2} = G \sum_k \frac{M_{BH}(R_k - r_j)}{(\epsilon_{BS}^2 + |R_k - r_j|^2)^{3/2}} + G \sum_j \frac{m_j(r_j - r_i)}{(\epsilon_{SS}^2 + |r_j - r_i|^2)^{3/2}} - \frac{GM_{r_j}}{(r_j^2 + b^2)^{3/2}}, \]

where \( m_j \) is the mass of a perturbation particle (Eq. (2.23)), \( \epsilon_{BS} \) is the softening parameter for perturbation particle/black hole interactions and was varied, and \( \epsilon_{SS} = 0.1 \) is the softening parameter for interactions between stars.

### 5.2.4 Experiments and Results

In total thirteen experiments were performed to investigate the evolution of two black holes in a galaxy, and these are summarised in Table 5.1. These experiments were performed on the Waverley-P3 machine (see Chapter 3.1.1) or a workstation with a 400MHz UltraSPARC-IIi processor. The total energy
of the system (including the black holes) was conserved to better than 1% for the NBODY1 experiments \((N = 8000)\) and better than 6% for the PPM experiments \((N = 8000)\).

Table 5.1: Summary of experiments performed. The column heading “Plane” indicates which plane the initial orbits of the black holes were in, and \(t_{\text{crit}}\) is the execution time (expressed in units of the crossing time).

<table>
<thead>
<tr>
<th>Experiment</th>
<th>(N)</th>
<th>(M_{\text{BH}})</th>
<th>(\epsilon_{\text{BS}})</th>
<th>Plane</th>
<th>(t_{\text{crit}})</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>(10^{11})</td>
<td>0.01</td>
<td></td>
<td>(xy)</td>
<td>33</td>
<td>CDF</td>
</tr>
<tr>
<td>B</td>
<td>(10^{11})</td>
<td>0.02</td>
<td></td>
<td>(xy)</td>
<td>17</td>
<td>CDF</td>
</tr>
<tr>
<td>C</td>
<td>(10^{11})</td>
<td>0.04</td>
<td></td>
<td>(xy)</td>
<td>9</td>
<td>CDF</td>
</tr>
<tr>
<td>D</td>
<td>500</td>
<td>0.02</td>
<td>0.008</td>
<td>(yz, xz, xy)</td>
<td>18</td>
<td>NBODY1</td>
</tr>
<tr>
<td>E</td>
<td>2000</td>
<td>0.02</td>
<td>0.002</td>
<td>(yz, xz, xy)</td>
<td>18</td>
<td>NBODY1</td>
</tr>
<tr>
<td>F</td>
<td>8000</td>
<td>0.02</td>
<td>0.0005</td>
<td>(yz, xz, xy)</td>
<td>18</td>
<td>NBODY1</td>
</tr>
<tr>
<td>G</td>
<td>500</td>
<td>0.02</td>
<td>0.004</td>
<td>(yz, xz, xy)</td>
<td>11</td>
<td>PPM</td>
</tr>
<tr>
<td>H</td>
<td>2000</td>
<td>0.02</td>
<td>0.004</td>
<td>(yz, xz, xy)</td>
<td>11</td>
<td>PPM</td>
</tr>
<tr>
<td>I</td>
<td>8000</td>
<td>0.02</td>
<td>0.004</td>
<td>(yz, xz, xy)</td>
<td>11</td>
<td>PPM</td>
</tr>
<tr>
<td>J</td>
<td>8000</td>
<td>0.02</td>
<td>0.001</td>
<td>(yz)</td>
<td>11</td>
<td>NBODY1</td>
</tr>
<tr>
<td>K</td>
<td>8000</td>
<td>0.02</td>
<td>0.004</td>
<td>(yz)</td>
<td>11</td>
<td>NBODY1</td>
</tr>
<tr>
<td>L</td>
<td>8000</td>
<td>0.02</td>
<td>0.001</td>
<td>(yz)</td>
<td>11</td>
<td>PPM</td>
</tr>
<tr>
<td>M</td>
<td>8000</td>
<td>0.02</td>
<td>0.016</td>
<td>(yz)</td>
<td>11</td>
<td>PPM</td>
</tr>
</tbody>
</table>

Figure 5.3: Plot of the position of one black hole relative to the other black hole in the \(xy\)-plane. The masses of the equal mass black holes are \(M_{\text{BH}} = 0.01, 0.02, 0.04\) corresponding to the green, blue and red lines respectively: Experiments A,B and C.

Modelling the dynamical friction by a simple analytic formula based on a modified form of Chandrasekhar’s dynamical friction formula (see Eq. (5.8)) was investigated in experiments A, B and C. The coefficient \(\kappa\) within the Coulomb algorithm was estimated to be \(\kappa \approx 0.145\); this was chosen to ensure
Figure 5.4: Plot of the reciprocal of the semi-major axis $1/a$ against time (in units of $t_{cr}$). The masses of the equal mass black holes are $M_{BH} = 0.01, 0.02, 0.04$ corresponding to the green, blue and red lines respectively: Experiments A, B and C.

that the black holes become bound at a time comparable to the time predicted by the direct method $N$-body solver (see experiment F).

Table 5.2: Comparison of some predicted dynamical quantities by Begelman et al. (1980) with the results of experiments A, B and C. The quantities $r_b$ and $t_b$ are respectively the black hole separation and time (in units of $t_{cr}$) when the black holes became bound. The superscripts $B$ and $E$ denote the predictions of Begelman et al. and the CDF experimental results respectively.

<table>
<thead>
<tr>
<th>$M_{BH}$</th>
<th>$r_b^B$</th>
<th>$r_b^E$</th>
<th>$t_b^E$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>0.166</td>
<td>0.156</td>
<td>6.95</td>
</tr>
<tr>
<td>0.02</td>
<td>0.21</td>
<td>0.213</td>
<td>3.76</td>
</tr>
<tr>
<td>0.04</td>
<td>0.26</td>
<td>0.248</td>
<td>2.13</td>
</tr>
</tbody>
</table>

In this simple approximation, dynamical friction is approximately propor-
Figure 5.5: Plot of (a) the total angular momentum (AM) of the black holes against time (in units of $t_{\text{cr}}$), and (b) the total kinetic energy (KE) of the black holes against time (in units of $t_{\text{cr}}$). The masses of the equal mass black holes are $M_{BH} = 0.01, 0.02, 0.04$ corresponding to the green, blue and red lines respectively: Experiments A, B and C.

tional to the mass of a black hole. As a consequence, the black holes spiraled towards the centre of the galaxy more rapidly as $M_{BH}$ was increased (see Figure 5.3). The black holes became bound to one another when $1/a$ turned positive, and this occurred at times (in units of the crossing time) 6.95, 3.759, 2.13 for $M_{BH} = 0.01, 0.02, 0.04$ respectively (see Figure 5.4(a)). This matches well to the prediction of Begelman et al. (1980) of the black hole separation when the black holes first become bound (see Table 5.2). The hardening rate of the binary was constant after an early transient (see Figure 5.4(b)) since loss cone depletion was ignored. As the binary continued to harden the black holes lost angular momentum (see Figure 5.5(a)), and the total kinetic energy of the black holes exhibited fluctuations of increasing amplitude (see Figure 5.5(b)), since
Figure 5.6: Plot of (a) the eccentricity $e$ against time (in units of $t_{cr}$), and (b) the natural logarithm of the apocentric distance (AP, top 3 lines) and the natural logarithm of the pericentric distance (PER, bottom 3 lines) against time (in units of $t_{cr}$) (see Figure 5.2 for an illustration of the apocentric and pericentric distances). The masses of the equal mass black holes are $M_{BH} = 0.01, 0.02, 0.04$ corresponding to the green, blue and red lines respectively (and pericentric distance pink, light blue and yellow respectively): Experiments A, B and C.

The orbital speed of a black hole at apocentre decreased at late times, whilst at pericentre the orbital speed increased. The simulation was terminated when the black hole separation was 100 times the Schwarzschild radius (about $10^{-5}$ in the units described in Chapter 5.2.3); at this separation gravitational radiation would begin to dominate and the black holes would merge very quickly.

The orbits of the black holes are hyperbolic prior to becoming bound, and thereafter the orbits change from highly eccentric orbits to circular orbits in less than a crossing time (see Figure 5.6(a)). Just prior to coalescence the orbits are highly eccentric; this is a result of the velocity dependence in Chandrasekhar's
Figure 5.7: Plot of the reciprocal of the semi-major axis $1/a$ against time (in units of $t_{cr}$). The plots (a), (b) and (c) correspond to $N = 500, 2000, 8000$ stars employed to model the galaxy. The 3 different colours green, blue and red correspond to 3 different initial planar orbits for the black holes, respectively the $yz, xz, xy$ planes: Experiments D, E and F. The pink line represents the results when the dynamical friction is modelled by Chandrasekhar’s dynamical friction formula: Experiment B.
Figure 5.8: Plot of the natural logarithm of the pericentric distance (PER) of the binary against time (in units of $t_{cr}$). The plots (a), (b) and (c) correspond to $N = 500, 2000, 8000$ stars employed to model the galaxy. The 3 different colours green, blue and red correspond to 3 different initial planar orbits for the black holes, respectively the $yz, xz, xy$ planes: Experiments D, E and F.
Figure 5.9: Plot of (a) the fraction of stars, within a radius of 0.5 relative to the density centre of the stars, which are still present compared to the initial number of stars in this region, and (b) the distance of the centre of mass of the black holes relative to the density centre of the stars; both against time (in units of $t_\text{gr}$). The total number of stars used to model the galaxy was $N = 500, 2000, 8000$ corresponding to the green, blue and red lines respectively: Experiments D, E and F ($xz$-plane).

Dynamical friction formula, since dynamical friction is strongest at apocentre (corresponding to the minimal orbital speed) and weakest at pericentre (corresponding to the maximal orbital speed), which in turn leads to a higher rate of decrease in the pericentric distance as compared to the apocentric distance (see Figure 5.6(b)).

In conditions appropriate to galaxies like M87 ($r_c \sim 100\text{pc}, v_c \sim 300\text{kms}^{-1}$,
Figure 5.10: Plot of the reciprocal of the semi-major axis $1/a$ against time (in units of $t_{\text{crit}}$). The plots (a), (b) and (c) correspond to $N = 500, 2000, 8000$ stars employed to model the galaxy. The 3 different colours green, blue and red correspond to 3 different initial planar orbits for the black holes, respectively the $yz, xz, xy$ planes: Experiments D, E and F. The pink line represents the results when the dynamical friction is modelled by Chandrasekhar's dynamical friction formula: Experiment B.
the results imply that binary black holes of equal masses $M_{BH} = 0.01, 0.02, 0.04$ should merge in a time $t_c \sim 6.7 \times 10^7 \text{yr}, 3.4 \times 10^7 \text{yr}, 1.8 \times 10^7 \text{yr}$ respectively. This is almost certainly an under-estimation of the evolution timescale of a black hole binary, however, since the longest phase of the evolution has been ignored (the loss cone depletion and repopulation).

A more accurate and realistic simulation of the evolution of binary black holes was performed in experiments D, E and F (with the same initial field star population for all three initial black hole planar orbits) with the aid of a direct method N-body solver (NBODY1). Although there was a restriction on the number of stars adopted to model the galaxy, this N-body evolutionary model did not ignore the self-consistent response of the galaxy. The effects of loss cone depletion were also not ignored but may have been inaccurate since

- the black hole/star interaction was softened
- the number of stars used in the simulation was much smaller than in a typical galaxy, and although the star/star interactions were softened, it is unclear how this affects the loss cone depletion and repopulation timescale

Nevertheless, despite these uncertainties, the early evolution phase was investigated, with particular emphasis on the effect of particle noise on the results. Increasing the particle number from $N = 500$ to 8000 reduced the particle noise, as is evident in Figures 5.7, 5.8 and 5.12. For example, this reduction is apparent in the reciprocal of the semi-major axis $1/a$ (Figure 5.7), which increases at approximately a constant rate until $1/a = 250$ (for $N = 2000, 8000$); at this time the simulation was terminated.

There is evidence of increasing loss cone depletion in the form of the fraction of stars remaining in the central region of the system (see Figure 5.9(a), $N = 2000, 8000$). Also perhaps some of the fluctuations observed in Figure 5.9 can be explained in terms of noise, and consequently you would expect the amplitude with $N = 500$ to be approximately 16 times bigger than that with $N = 2000$; however, the amplitude with $N = 2000$ is comparable to the amplitude with $N = 8000$. (It should be noted that the results illustrated in Figure 5.9 are based on experiments where the initial black hole orbits are in the $xz$-plane; this gives the lowest hardening rate for the $N = 500$ case (see Figure 5.7(a)). Consideration of the $yz$ and $xy$-planes do not alter the results of Figure 5.9 significantly.) Some support for the dynamical effect of loss cone depletion is provided by the fact that the hardening rate of the black hole binary is
larger when using Chandrasekhar's dynamical friction formula (experiment B) than when the system was evolved by a direct method code (see Figure 5.7). Another possible reason for this, however, is that Chandrasekhar's dynamical friction formula assumes that the centre of mass (COM) of the binary is fixed at the density centre of the stars and does not wander; a consequence of this is that the binary hardens near the point of highest star density. In contrast, the direct method evolutionary code allowed the COM of the binary to wander away from the density centre of the cluster (see Figure 5.9(b)). The amplitude of this wandering may be governed by two possible factors:

1. as the particle number $N$ is increased, the perturbations that the COM of the binary endures are reduced, and thus the COM wanders less from the density centre of the galaxy

2. if the central density reduces (due to loss cone depletion) then the restoring force which keeps the COM at the centre weakens (Quinlan & Hernquist 1997).

As observed in Figure 5.9(b) the COM of the black hole binary wandered significantly less far from the density centre of the stars as $N$ was increased. This may, in part, explain why the hardening rate for $N = 500$ (Figure 5.7(a), $xz$-plane) is less than the hardening rate for $N = 8000$ (Figure 5.7(c), $xz$-plane). Furthermore, as discussed below, the hardening rate is sensitive to the length of the softening parameter associated with a black hole/star interaction. Thus the length of the softening parameter associated with a black hole/star interaction for $N = 500$ may simply be too large to permit an effective energy transfer between a black hole and a star, and consequently the hardening rate of the binary is too small.

By extrapolating the pericentric distance of the binary (see Figure 5.8) for $N = 8000$, and assuming that loss cone depletion does not further reduce the hardening rate of the binary, then a minimal evolution timescale for a binary black hole (until coalescence) may be estimated. Fitting a first degree polynomial (by a least squares fit) to the empirical data in Figure 5.8(c) ($N = 8000$ and for the time greater than 8 crossing times) yields

$$\ln p \approx -0.188t - 2.9324,$$

where $p$ is the pericentric distance and $t$ is the time (in units of the crossing time). Assuming the binary would merge soon after the pericentric distance
had reduced to 100 times the Schwarzschild radius then the minimal evolution timescale of a binary black hole may be estimated as \( t_e \approx 45.6t_{cr} \approx 9.12 \times 10^7 \text{yr} \) (based on the core parameters of M87). This is approximately 2.7 times the evolution timescale predicted by the CDF experiments.

Some experiments have been performed on the evolution of binary black holes using the PPM (experiments G, H and I, with the same initial field star population for all three initial black hole planar orbits). Up to the time when the black holes became bound to one another \( (1/a > 0) \) the PPM experiments produced results comparable to the NBODY1 experiments (see Figures 5.10 and 5.11). There is no evidence, up to the maximum particle number considered \( (N = 8000) \), to suggest that the PPM reduced the particle noise in the results (see Figures 5.10 to 5.13). Furthermore, the hardening rate of the binary (once the black holes were bound) was significantly less for the PPM experiments than the NBODY1 experiments. This is presumably due to the fact that the length of softening chosen for the perturbation particle/black hole interactions \( (\epsilon_{BS} = 0.004) \) in the PPM experiments was larger than the softening length adopted in the NBODY1 experiments \( (\epsilon_{BS} = 0.0005) \); this will have affected the energy transfer between the black hole and the interacting star or perturbation particle, which in turn will have affected the hardening rate of the binary. Further investigation with both NBODY1 (Figure 5.14(a)) and the PPM (Figure 5.14(b)) separately, showed that the hardening rate decreased as \( \epsilon_{BS} \) was increased in both cases. Choosing the same softening parameter \( \epsilon_{BS} \) for both sets of experiments (NBODY1 and PPM) yielded comparable results for the hardening rate of the binary black hole (see Figure 5.14(c)).

5.3 Discussion and conclusions

Many galaxies contain central black holes, and in the event that two such galaxies merge, then the two black holes sink towards the centre of mass of the merger remnant due to dynamical friction, and form a binary in the core. The binary black hole evolves because of close encounters, which must be integrated accurately; however the galaxy also evolves because of the collective response of the stars to changes in the potential. To suppress relaxation amongst the field stars and to satisfy the requirement that the stars be much less massive than the black holes, a large number of stars is required.

To this aim, the perturbation particle method was adopted to model the evolution of the two black holes in a galaxy, but only for the first \( 2 \times 10^7 \text{yr} \) of
their initial evolution (based on the core parameters of M87). For comparative purposes, experiments were also performed using a simple analytic dynamical friction formula (Chandrasekhar's), and a direct method N-body solver.

In summary, Chandrasekhar's dynamical friction formula, although easy to implement, underestimates the evolution timescale of the black hole binary due to a number of factors including the absence of loss cone depletion. These results are similar to those of Fukushige et al. (1992). The direct method N-body code, which attempts to model loss cone depletion, increased the evolution timescale by a factor of 2.7 as compared to the CDF experiments. Conclusions from this are not as reliable as those of Makino (1997) since the particle number adopted to model the galaxy is smaller here and the loss cone depletion phase was only investigated for a short time.

The PPM experiments offered comparable results to the NBODY1 method up to about 10 crossing times with a suitable choice of softening for the black hole/star (or perturbation particle) interactions. There are, for the particle numbers considered here, no indications that the PPM was a better method (in the sense that the noise in the results is reduced) in evolving binary black holes in galactic nuclei than the NBODY1 method. However, increasing the particle number further may reveal benefits of using the PPM. A couple of possibilities which would allow an increase in \( N \) is an implementation of the PPM with the Ahmad Cohen neighbour scheme (see Chapter 2.3) and an efficient paralleli-sation of the resulting code. Also the black hole/black hole interactions can be more efficiently dealt with by a suitable regularisation (see Chapter 2.3). Understanding the effect that \( \epsilon_{BS} \) has on the hardening rate of the binary is crucial if one is to understand the evolution of a black hole binary. On the other hand, no consideration has been paid to the softening associated with star/star interactions; loss-cone depletion and repopulation are sensitive to this.
Figure 5.11: Plot of the reciprocal of the semi-major axis $1/a$ against time (in units of $t_{cr}$). The plots (a), (b) and (c) correspond to $N = 500, 2000, 8000$ perturbation particles employed to model the galaxy. The 3 different colours green, blue and red correspond to 3 different initial planar orbits for the black holes, respectively the $yz, xz, xy$ planes: Experiments G, H and I. The pink line represents the results when the dynamical friction is modelled by Chandrasekhar’s dynamical friction formula: Experiment B.
Figure 5.12: Plot of the eccentricity $e$ of the binary against time (in units of $t_{cr}$). The plots (a), (b) and (c) correspond to $N = 500, 2000, 8000$ stars employed to model the galaxy. The 3 different colours green, blue and red correspond to 3 different initial planar orbits for the black holes, respectively the $yz, xz, xy$ planes: Experiments D, E and F. The pink line represents the results when the dynamical friction is modelled by Chandrasekhar’s dynamical friction formula: Experiment B.
Figure 5.13: Plot of the eccentricity $e$ of the binary against time (in units of $t_{cr}$). The plots (a), (b) and (c) correspond to $N = 500, 2000, 8000$ perturbation particles employed to model the galaxy. The 3 different colours green, blue and red correspond to 3 different initial planar orbits for the black holes, respectively the $yz, xz, xy$ planes: Experiments G, H and I. The pink line represents the results when the dynamical friction is modelled by Chandrasekhar’s dynamical friction formula: Experiment B.
Figure 5.14: Plot of the reciprocal of the semi-major axis against time (in units of $t_{cr}$); the softening parameter ($\epsilon_{BH}$) for black hole/star (or perturbation particle) interactions is varied (a) $\epsilon_{BH} = 0.0005, 0.001, 0.004$ corresponding to the green, blue and red lines respectively (Experiment F, J and K), (b) $\epsilon_{BH} = 0.001, 0.004, 0.016$ corresponding to the blue, green and red lines respectively (Experiment L, I and M), and (c) $\epsilon_{BH} = 0.001$ (Experiment J, green line; Experiment L, blue line) and $\epsilon_{BH} = 0.004$ (Experiment K, red line; Experiment I, pink line).
Bibliography


120


123


[106] Salmon J.K., 1991, PhD'T, 7


[121] Sweatman W., 1994, JCompPhys, 111, 1, p110

[139] Zepf S.E., 1997, Nat, 390, 377