Chaotic Dynamics in N-body systems
Boekholt, T.C.N.

Cover Page

Universiteit Leiden

Author: Boekholt, Tjarda
Title: Chaotic dynamics in N-body systems
Issue Date: 2015-11-10

Chaotic Dynamics
in N-body Systems

Tjarda C. N. Boekholt

Chaotic Dynamics
in N-body Systems

Proefschrift

ter verkrijging van de graad van Doctor
aan de Universiteit Leiden
op gezag van de Rector Magnificus Prof. mr. C.J.J.M. Stolker,
volgens besluit van het college voor Promoties
in het openbaar te verdedigen op dinsdag 10 november 2015
klokke 13:45 uur

door

Tjarda Coenraad Nico Boekholt

geboren op 12 mei 1985
te Leidschendam

PROMOTOR: Prof. dr. S. F. Portegies Zwart

CO-PROMOTOR: Prof. dr. D. C. Heggie   University of Edinburgh

OVERIGE LEDEN: Prof. dr. S. L. W. McMillan   Drexel University
Dr. A. L. Varri   University of Edinburgh
Prof. dr. A. Doelman
Prof. dr. K. H. Kuijken
Prof. dr. H. J. A. Röttgering

Voor Pap en Mam

*"You must have chaos within you
to give birth to a dancing star."*
Friedrich Nietzsche

Omslag: ontwerp J. R. C. Boekholt

Contents

Preface

"This is how one tries to put the comet, which has caused so much fear, at the service of humour and of innocent children's play and it is clear that such things can count on a wide flow" — Leeuwarder Courant June 27, 1910

So ends the article in the Leeuwarder Courant June 27, 1910 entitled "The toy industry dedicated to Halley's Comet". And fear there was! The News of the Day for the Dutch East-Indies of June 7, 1910 reported a farmer from Eifeldorf-Geizenburg who threw her four month old baby in a pit out of fear of the destruction of the world by the arrival of Halley's Comet. The baby drowned. Also, the Algemeen Handelsblad of July 2, 1910 reported on numerous cases of suicide and insanity in Italy, and attacks of rage and suicide in France.

The close encounter of a comet with earth produces one of the most beautiful and dramatic sceneries we can observe, together with solar eclipses or the Aurora Borealis. Throughout history, the appearance of a comet was interpreted as a supernatural omen for good or bad things to come. As the examples above illustrate, the ignorance in the workings of nature and the belief in superstitions have caused harmless wonders of the universe to have horrific consequences. It is one of the tasks of the astronomer to understand the whereabouts of comets and other objects in the universe, and to explain them to the public.

Leiden Observatory
August 14 2015

"The Sun, with all those planets revolving around it and dependent on it, can still ripen a bunch of grapes as if it had nothing else in the Universe to do."
Galileo Galilei

1

Introduction

This thesis focusses on the evolution of dynamical systems in the universe. The main systems of interest are planetary systems and star clusters. Understanding the structure and evolution of these systems includes calculating the orbits of the constituent bodies. This is rather complex due to the non-linear, chaotic nature of most orbits. Therefore, we need to resort to numerical N-body techniques to approximate the orbits using sophisticated software and modern hardware.

A central theme in this thesis is the growth of a small perturbation. This could be a small numerical error that propagates through the system causing numerical noise. This could bias results from N-body simulations, questioning their reliability. The initial perturbation could also be physical and its propagation determines the stability of an orbit.

This chapter provides a general introduction to the topics mentioned above, and subsequent work in this thesis on two new N-body algorithms, testing the reliability of N-body simulations and determining the origin of chaos in the orbit of Comet Halley.

1.1   DYNAMICAL SYSTEMS

In the universe there is an endless number of examples of dynamical systems. Any system that consists of a collection of objects or so-called bodies, which move around through space and time due to their mutual forces, is a valid candidate.

Our solar system is a familiar example consisting of several types of objects. We have the sun at the center (Copernicus, 1543), the planets revolving around the sun in ellipses (Kepler, 1609), moons orbiting the planets (Galilei, 1610), and many smaller objects such as asteroids, comets (Halley, 1705) and so on (see Fig. 1.1, top image). These objects all have one thing in common, they all have mass. Since the pioneering work by Newton (1687), we know that bodies with mass attract each other. Every object feels the gravity from the other objects. As a consequence, our solar system is not static, but evolves because all the bodies are moving. If all the bodies in the system are

moving, then it is also possible for two of them to experience a close encounter, like Halley's Comet and Earth back in 1910.

A different kind of dynamical system is a star cluster (see Fig. 1.1, bottom image). To first order, such a system consists solely of stars. The number of stars varies per cluster, from a few dozen to millions. The number of stars matters for the evolution of the system as a whole. Hypothetically speaking, if we put our sun in a star cluster (modern theories actually predict this was the case when our sun was younger (Martínez-Barbosa et al., 2015; Jílková et al., 2015)) and there are only a few other stars in the cluster, then the gravitational pull from each star on the sun would form a significant contribution. If instead, we have a million other stars, each individual contribution becomes very small and so the sun would follow a smoother orbit set by the overall, background potential. This dependency on the number of stars is usually captured by the relaxation time of the cluster (Chandrasekhar, 1942), which is the time scale on which the sun would significantly alter its orbit in its birth cluster.

Only in the dense cores of massive star clusters, do close encounters between two or more stars become important again. Three-body encounters especially, form a separate category of dynamical systems. They are important for the formation of binary stars. Initially three stars will encounter each other and exchange some energy. If one of the three stars manages to steal enough energy from the other two, it will fly away out of the core of the cluster. The remaining two stars have lost energy and form a bound two-body system, e.g. a binary star (Szebehely, 1972). This binary star will interact with the other surrounding stars, effectively inserting kinetic energy into the core of the cluster. Many theoretical and numerical studies have been performed to measure these detailed energy exchanges and its feedback on the cluster (e.g. Heggie, 1975; Hut & Bahcall, 1983; McMillan & Hut, 1996; Boekholt et al., 2015). For a more general background on star cluster dynamics, the following books are highly recommended: ?, ? and ?.

Planetary systems, star clusters and three-body scatterings are just a few examples of dynamical systems in the universe. There are many more, such as galaxies consisting of billions of stars or galactic nuclei which include supermassive black holes. For all these dynamical systems there are several things we would like to know. Specifically, how many and what type of bodies are present in the system, how they are distributed throughout the system, how does the global structure evolve and what is its final fate. As an illustration, we take our own solar system again, but only consider the sun and the planets. We have
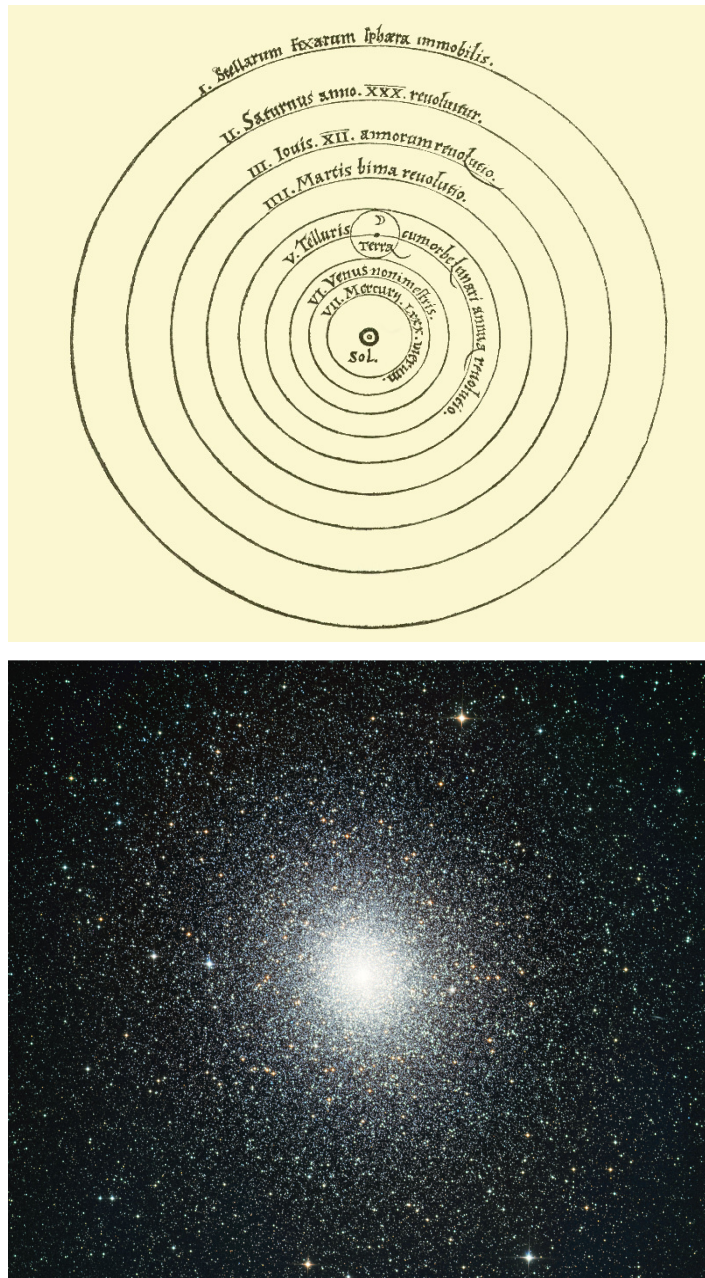
Figure 1.1: Two examples of dynamical systems: schematic overview of our solar system, including the sun, the planets and our moon (top image) and the star cluster 47Tuc, consisting of over a million stars (bottom image). Credit to Copernicus (1543) and Dieter Willasch (Astro-Cabinet).

measured their masses, positions and velocities as they are currently. One of the open problems is how these orbits evolve over billions of years. One possibility is that the solar system will always remain more or less the same, i.e. the system is stable (Ito & Tanikawa, 2002). Some studies have shown that it is also possible for orbits to change significantly, resulting in collisions between planets (Laskar & Gastineau, 2009) or an escape from the system altogether (Laskar, 2008). To investigate the origin and evolution of dynamical systems we need a mathematical model that captures the behaviour of bodies in motion.

1.2   THE N-BODY PROBLEM

Newton (1687) defined a mathematical model, called the N-body problem, which describes the following: we have a dynamical system consisting of N bodies (with $N = 1, 2, 3, ...$), each having a mass, position and a velocity. What are the positions and velocities at any time in the future or in the past? If the bodies would not influence each other, Newton's first law states that the body will continue moving on a straight trajectory with constant velocity. If the bodies do feel each other through a mutual force, they will experience an acceleration and generally follow curved trajectories. This relation is given by Newton's second law of motion

$$F = ma, \tag{1.1}$$

with F the force, m the mass of the body experiencing the force and a the acceleration. When the apple hit Newton's head, he realized that the attractive force that acts between all bodies with mass is gravity. The force of gravity is proportional to the masses of the two bodies and inversely proportional to the square of the distance between them

$$F = \frac{GMm}{r^2}. \tag{1.2}$$

Here G is the gravitational constant, M and m are the masses of the two interacting bodies and r the distance between them. To be more precise, the mass in Eq. 1.1 is the inertial mass, whereas the mass in Eq. 1.2 is the gravitational mass. Experiments have shown however, that they are equal up to at least 13 decimal places (Poisson & Will, 2014). We can therefore equate Eq. 1.1 and 1.2 and at the same time take the sum over all other bodies

$$\vec{a}_i = G \sum_{j \neq i} \frac{m_j}{r_{ij}^3} \vec{r}_{ij}. \tag{1.3}$$

Here $\vec{a}_i$ is the total acceleration experienced by particle i, due to all other particles with masses m_j at distances of $r_{ij} = |\vec{r}_j - \vec{r}_i|$. For every object we calculate its acceleration due to all the other objects. Next step is to use this acceleration to calculate the position and velocity at some time in the future. This is the tricky part.

The acceleration as a function of time is generally not a simple function, but rather complex and chaotic. Except for a few cases with $N = 2$ (Newton, 1687) and $N = 3$ (Euler, 1767; Lagrange, 1772) we can solve the orbits analytically. For the rest of the cases we need to measure the acceleration, time step by time step. This discretisation of the N-body problem introduces an error in the solutions, but makes it ideal for solving it on a computer.

1.3 N-BODY SIMULATIONS

1.3.1 Algorithms

Soon after the invention of the computer, did the first N-body simulations arise. Among the first ones to solve the N-body problem on a computer were von Hoerner (1960) and Aarseth (1963), with the aim of solving the orbits of stars under their mutual gravity.

Solving the N-body problem on a computer involves two main ingredients: an integration method and a time step criterion. The integration method determines how the new position and velocity are calculated from the current position, velocity and acceleration. The simplest integrator is the Euler method:

$$r\left(t + \Delta t\right) = r\left(t\right) + v\left(t\right)\Delta t, \qquad (1.4)$$

$$v\left(t + \Delta t\right) = v\left(t\right) + a\left(t\right)\Delta t. \qquad (1.5)$$

Here t is the current time, Δt is the time step size, r is the position and v the velocity. By iteratively performing the algorithm of applying Eq. 1.3, 1.4 and 1.5, we can sample the orbits of the bodies in time and study the evolution of dynamical systems.

The Euler algorithm is very simple and clear, but not very precise. If we regard it as a first order Taylor expansion, then the higher order terms are neglected which introduces truncation error in the results. One way to reduce this type of error is to select an appropriate time step criterion. In principle the time step size should be as small as possible, but this will increase the number of integration steps, which dramatically increases the amount of CPU time. Therefore, there is a trade off between the amount of precision and speed. A variety of time step criteria exist in the literature, two of which are

$$\Delta t = \min \left\{ \frac{r_{ij}}{v_{ij}} \right\}, \tag{1.6}$$

$$\Delta t = \min \left\{ \sqrt{\frac{r_{ij}}{a_{ij}}} \right\}, \tag{1.7}$$

The first criterion makes sure that the closest pair of bodies cannot collide within a single time step, because the time step size becomes smaller if there are close encounters. The second criterion is similar but also works for zero velocities. For more advanced time step criteria and also further details on N-body algorithms and simulations we refer the reader to Aarseth (2003) and The Art of Computational Science project by Hut and Makino[1].

Two commonly used N-body integrators are the Leapfrog and the Hermite method. The Leapfrog algorithm goes as follows

$$v(t + \frac{\Delta t}{2}) = v(t) + a(t) \frac{\Delta t}{2}, \tag{1.8}$$

$$r(t + \Delta t) = r(t) + v(t) \Delta t + \frac{1}{2} a(t) \Delta t^2, \tag{1.9}$$

$$v(t + \Delta t) = v(t + \frac{\Delta t}{2}) + a(t + \Delta t) \frac{\Delta t}{2}. \tag{1.10}$$

Compared to the Euler scheme, the new position is calculated using an extra term proportional to Δt^2. The main difference is that the velocity is updated in two steps. First, half of the time step is taken using the current acceleration, then the second half of the time step is taken using the new acceleration which is calculated at time $t+\Delta t$. The updated positions and accelerations in the second step of the velocity calculation, has the result that the energy error remains bound. This property is characteristic for symplectic methods, and is especially useful for integrations of planetary systems (**?**).

The Hermite integrator (**?**) is a fourth-order, non-symplectic method that is useful for accurate integration of collisional systems such as star clusters. The algorithm consists of a prediction step

$$r_p = r(t) + v(t) \Delta t + \frac{1}{2} a(t) \Delta t^2 + \frac{1}{6} j(t) \Delta t^3, \tag{1.11}$$

$$v_p = v(t) + a(t) \Delta t + \frac{1}{2} j(t) \Delta t^2, \tag{1.12}$$

---

[1]http://www.artcompsci.org/

and then a correction step that makes use of the initial coordinates and the predicted coordinates:

$$r(t + \Delta t) = r(t) + \frac{1}{2} \left(v(t) + v_p \right) \Delta t + \frac{1}{12} \left(a(t) - a_p \right) \Delta t^2, \qquad (1.13)$$

$$v(t + \Delta t) = v(t) + \frac{1}{2} \left(a(t) + a_p \right) \Delta t + \frac{1}{12} \left(j(t) - j_p \right) \Delta t^2. \qquad (1.14)$$

Here $j(t)$ is the jerk which is the time derivative of the acceleration, and a_p is the acceleration calculated using the predicted positions[1].

Once the software has been written it has to run on a piece of hardware. For simulations of a few particles a personal desktop is sufficient. A gain in speed can be obtained by using the fastest processors available on the market. For larger dynamical systems it becomes efficient to parallellise the computations. As was shown in the previous section, to calculate the acceleration of a single body, we need to iterate over all the other bodies, which is an operation of order N. However, we want to know the acceleration of each particle, which is another factor N, so that the total cost is of order N^2. Dividing the computational work over as many cores as possible can lead to an increase in speed. It is worth noting that there is a trade off between the time the computer spends calculating and the amount of time the computer cores are communicating and passing data. The challenge for N-body developers is to design a code which runs efficiently with as many cores as possible (Portegies Zwart et al., 2008).

Approximate methods exist which trade in some precision for better performance by relieving the constraint of having to consider all pairs of bodies. One familiar example is the tree method where clumps of bodies are replaced by a single, centre of mass body (**??**). Finally, calculations on Graphic Processing Units (GPUs), instead of Central Processing Units (CPUs), can decrease the duration of the simulations by one or two orders of magnitude. The current fastest N-body code on the planet is a tree method compatible with clusters of GPUs (**?**).

Since the N-body problem cannot be solved exactly, much effort has gone into looking for new, improved algorithms for solving the N-body problem. Different methods can be more appropriate for different problems. For example, some might aim for speed in order to be able to handle large simulations of globular clusters and galaxies, whereas others might aim for high precision for planetary and few-body systems. The quest for the ultimate N-body method that is fast and precise still continues.
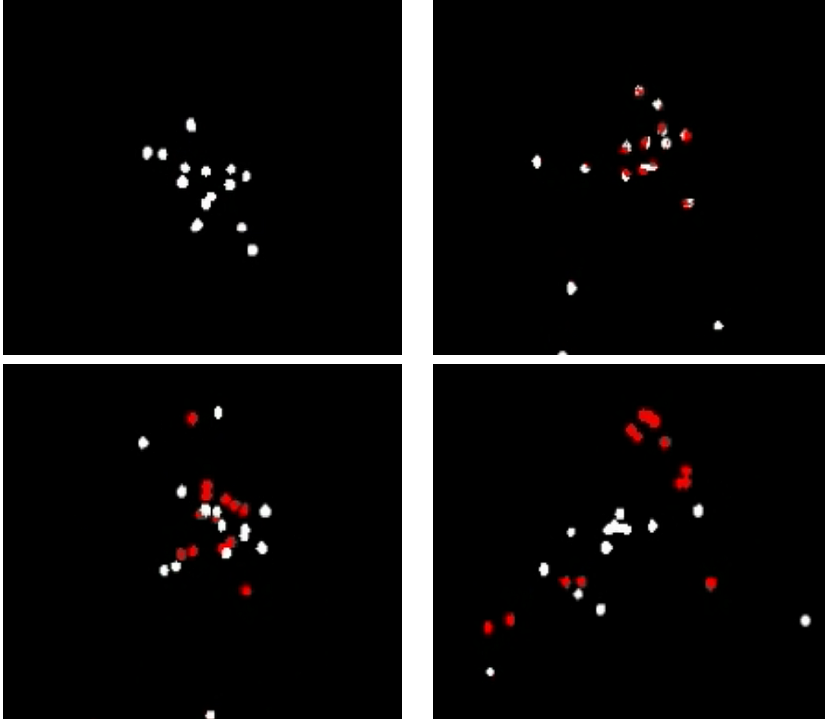
Figure 1.2: N-body simulation of a 16-body star cluster. The white stars represent the solution obtained by the Leapfrog method, and the red painted stars are obtained using the Hermite method. We show the initial condition (top row, left), and three snapshots a few crossing times apart (from left to right, and top to bottom), illustrating the divergence between neighbouring solutions.

1.3.2 Accuracy of N-body Results

The aim of any N-body simulation is to calculate the unique, mathematical solution to the N-body problem, given a set of initial conditions. The exact solution cannot generally be obtained analytically and so we have to resort to approximate, numerical methods. How do we know that the calculated solution is accurate?

The best way to quantify the accuracy of an approximate solution is to compare it to the true solution, but since the true solution is not available we need other methods. The most often used method is to check for energy conservation. As can be derived from the equations of motion, the total energy of a dynamical system is a conserved quantity. Any simulated N-body solution should not make too large energy errors in order to remain trustworthy (**?**). On the other hand, there are an infinite number of configurations on the energy hyperspace in the

multi-dimensional phase space, so that while the total energy might be conserved exactly, the precise configuration could still be completely wrong.

Another type of test is for reversibility. We start integrating the initial configuration up to some time t, then we change the direction of time and integrate backwards to $t = 0$. Ideally, we should reproduce the initial conditions, but due to error growth in the mean time, this might not be the case. For example, in a star cluster simulation consisting of single stars, the cluster will evolve towards core collapse, where its core density becomes extremely high and where binaries are formed due to three-body interactions (**?**). If we then reverse the arrow of time, we would expect the opposite evolution, where the core density gradually decreases and binaries get ionized. This backward evolution is however very unstable, and small deviations from this track will cause the evolution to go forwards again towards higher core density and binary formation.

Convergence tests are often performed to show that the obtained N-body results are safe and sound. After all, by decreasing the time step systematically, the discretisation error will become smaller up to the point that the first few decimal places will start to converge. The required precision however, depends on the specific configuration. For very chaotic systems, the necessary time step could be so small, that round-off errors due to the many time steps and operations, will become dominant and prevent any convergence. Every result obtained from N-body simulations should be treated carefully to make sure that we do not get led astray.

1.4 CHAOTIC DYNAMICS

Central in this thesis is the growth of a small initial perturbation in a dynamical system. Consider a slight perturbation of only one coordinate of a single body in the system. If we then calculate the accelerations of all the bodies, they will be slightly perturbed as well due to the slightly perturbed distances to the perturbed body. The new positions and velocities at a time step Δt later, will inherit the perturbation as well. Subsequent integration steps will magnify the perturbation until it has become the size of the system.

This mechanism is investigated from two different perspectives. First of all, we take the small initial perturbation to be a numerical error, so that we are looking at the growth of numerical noise in N-body simulations. As described in the previous section, it is challenging to obtain true solutions to the N-body problem due to different sources

of error (see Fig. 1.2). The main reason that small integration errors
cause problems is that many dynamical systems are inherently chaotic.
Similar to the familiar butterfly effect, a small integration error can be
magnified by orders magnitude within a finite time (**???**). This limits
the predictability of the positions of the bodies after a certain amount
of time. To some degree, high-precision N-body methods can fight the
exponentially growing numerical errors.

In the second interpretation, the small initial perturbation is phys-
ical and introduced manually in the initial conditions for the N-body
experiment. By measuring the growth of the perturbation we can
determine the stability of a dynamical system. If two neighbouring
solutions remain close on the time scale of interest, then small pertur-
bations do not alter the evolution significantly and we say the system is
stable to small deviations. In this case the perturbation usually grows
linear in time. If, on the other hand, a perturbed solution is completely
different after a relatively short time, then the system is categorized
as unstable and chaotic, and the perturbations grow exponentially.

Increasing our understanding of the way perturbations grow in dy-
namical systems is crucial for designing improved N-body methods
and for determining the stability of dynamical systems and their long
term evolution. One illustrative example is the stability of planetary
systems. Hundreds of exoplanets have been discovered orbiting other
stars and in a wide variety of orbits[2]. A significant fraction of exo-
planetary systems are much more compact than our solar system. By
understanding the reason why some configurations are stable and oth-
ers unstable, we can learn about the evolution of planetary systems
and the likelihood to observe them in a certain state.

1.5   THIS THESIS

This thesis presents five studies on new N-body algorithms, the relia-
bility of N-body results, and finally the origin of chaos in dynamical
systems. Below we introduce these topics and related research ques-
tions in somewhat more detail.

Chapter 2

As mentioned in Sec. 1.3.2, the best way to quantify the accuracy of
an N-body simulation is to compare it to the true solution. We design
the new N-body code Brutus which solves the N-body problem to a
pre-defined precision. This code adaptively reduces the time step size

---

[2]http://exoplanets.org/

to control the discretisation errors, and it uses an arbitrary-precision library to control the round-off errors. By making the time step smaller and by increasing the number of used digits, we obtain solutions with an increasing precision up to convergence. These converged N-body solutions approach the true solution up to the first specified number of decimal places. Using this code we are able to quantify the accuracy of approximate solutions and for the first time obtain true solutions to general N-body configurations.

Chapter 3

We explore a conceptually novel idea for an N-body algorithm. It is based on two main arguments: the two-body problem can be solved analytically and the superposition principle holds in the calculation of the accelerations. Therefore, are we able to solve every pair of two-body problem in the system and combine them to solve the overall N-body problem? We demonstrate that this is indeed the case and perform several validation and performance tests. This code turns out to be very efficient on parallel supercomputers.

Chapter 4

The assumption in the N-body community is that even though N-body solutions inherently possess some numerical noise, the results are still valid statistically. We test this assumption for three-body scattering systems as described briefly in Sec. 1.1. We compare statistics on triple interactions and dynamically formed binary stars, obtained from an ensemble of converged solutions (obtained by Brutus) to those obtained from an ensemble of approximate solutions (obtained by Hermite). We find that for sufficient energy conservation, three-body scattering statistics are preserved under divergence of individual solutions, which is good news for the N-body user.

Chapter 5

In this Letter, we discuss in more detail the observation from the previous chapter, that an ensemble of approximate solutions preserves the global statistical distributions. We analyse the results from three-body scattering experiments and illustrate this quasi-ergodic prop-

erty of gravity, i.e. "Nagh-Hoch"[3], which exposes itself particularly in chaotic systems. This property is crucial for the reliability of N-body simulations, which inherently possesses numerical noise.

Chapter 6

Having determined in previous chapters that chaotic systems show an exponential growth of small perturbations, we now want to understand why these systems show this behaviour. Why are some systems chaotic and others regular and is it possible to go from order to chaos and vice versa? We construct a model for the growth of perturbations based on two-body Keplerian systems being perturbed by a third body. We apply the model to Halley's orbit and find its Liapounov time, i.e. the e-folding time for exponential growth, to be about 300 years. This surprisingly short time scale follows naturally from the encounter density (Halley and Jupiter are close to a 3:19 mean motion resonance) and the strength of each close encounter.

Chapter 7

The field of N-body code development and simulations has matured since the first simulations performed in the 1960s. There are however still many areas to explore or to improve. We briefly describe several open problems, which should be solved within the next decade.

---

[3]Klingon for "Ensemble of Stone", referring to the preservation of ensemble statistics under numerical errors.

2

A Precise N-body Code:
Brutus

The general consensus in the N-body community is that statistical results of an ensemble of collisional N-body simulations are accurate, even though individual simulations are not. A way to test this hypothesis is to make a direct comparison of an ensemble of solutions obtained by conventional methods with an ensemble of true solutions. In order to make this possible, we wrote an N-body code called Brutus, that uses arbitrary-precision arithmetic. In combination with the Bulirsch–Stoer method, Brutus is able to obtain converged solutions, which are true up to a specified number of digits.

In this chapter we present the structure of Brutus and illustrate the method of convergence by applying it to several small-N systems. For the first time, we can exactly determine how accurate our N-body results are, and obtain true solutions to general N-body configurations. Finally, we construct a model for the scaling of Brutus with N, for converged solutions to reach core collapse. We conclude that it scales roughly exponentially, which is effectively caused by the exponential divergence between neighbouring solutions.

2.1   INTRODUCTION

Analytical solutions to the N-body problem are known for $N = 2$, which are the familiar conic sections. Also, for several systems possessing symmetries, analytical solutions have been found, for example the equilateral triangle (Lagrange, 1772). For a more general initial configuration, solutions have to be obtained by means of numerical integration. Given an initial N-body realisation, one can calculate all mutual forces and subsequently the net acceleration of each particle.

Different integration methods exist which take the accelerations, and update the positions and velocities to a time $t + \Delta t$, with Δt the time-step size. This process is repeated until the end time is reached.

? recognised that obtaining the solution to an N-body problem by numerical integration is difficult. This is caused by exponential divergence. Consider a certain N-body problem, i.e. N point-particles, each with a given mass, position and velocity. This system evolves with time in a definite and unique way. If one goes back to the initial state and slightly perturbs only one coordinate of a single particle, the perturbed N-body problem will also have a definite and unique but different solution than the original one. When the two solutions are compared as a function of time, it is observed that differences can grow exponentially (**????**). If the initial perturbation is due to a numerical error, the calculated solution will also diverge away from the true solution.

Several authors have estimated the time-scale of this divergence (**??**), and arrived at an e-folding time-scale of the order a dynamical, crossing time. Simulation times of interest are typically much longer than a crossing time and therefore staying close to the true solution is numerically challenging.

If the result of a direct N-body simulation of for example a star cluster, has diverged away from the true solution, the result may well be meaningless (**?**). The general consensus however, is that statistically the results are representative for the true solution to the N-body problem (**???**). The underlying idea is that the statistics of an ensemble of N-body simulations are representative for the true statistics, obtained by an ensemble of true solutions, with the same set of initial conditions. We regard this the hypothesis we want to test.

One way to test this hypothesis is to directly compare statistics obtained by conventional methods, with the statistics obtained from an ensemble of true solutions (see Chapter 4). To obtain true solutions, we wrote an N-body code which can solve the N-body problem to arbitrary precision.

Such a code can be realised if the different sources of error are controlled. The error has contributions from the time discretisation of the integrator and the round-off due to the limited precision of the computer (**?**). Another possible source of error is in the initial conditions, for example the configuration of the solar system is only approximately known (Ito & Tanikawa, 2002). However, if the initial condition is a random realisation of a distribution function, this is less often a problem. Using the Bulirsch–Stoer method (**??**), the discretisation error can be controlled to stay within a specified tolerance. Using arbitrary-

precision arithmetic instead of conventional double-precision or single-precision, the round-off error can be reduced by increasing the number of digits.

We obtain converged solutions to the N-body problem by decreasing the Bulirsch–Stoer tolerance and increasing the number of digits systematically. We define a converged solution in our experiments as a solution for which the first specified number of decimal places of every phase-space coordinate in our final configuration in the N-body experiment becomes independent of the length of the mantissa and the Bulirsch–Stoer tolerance. We explain the method of convergence in Sec. 2.2, we give examples of the procedure in Sec. 2.3 and we measure the scaling of Brutus in Sec. 2.4.

2.2   METHODS

2.2.1   The Benchmark Integrator

The gravitational N-body problem aims to solve Newton's equations of motion under gravity for N stars (Newton, 1687). A popular integrator to perform this task is the fourth-order Hermite predictor-corrector scheme (**?**), using double-precision arithmetic. The experiments we discuss in Sec. 4.1 will use this integrator as a benchmark test. We adopt a shared, adaptive time-stepping scheme with the following criterion:

$$\Delta t = \eta \min \sqrt{\Delta r_{ij}/\Delta a_{ij}}. \tag{2.1}$$

Here η is the time-step parameter and Δr_{ij} and Δa_{ij} are the relative distance and acceleration for the pair of particles i and j. We implement no further constraints on the time-step size.

To test how inaccurate we are allowed to integrate while still obtaining accurate statistics (**??**) we vary the time-step parameter η, to obtain statistics from conventional simulations with different precision.

2.2.2   The Brutus N-body Code

The results obtained with the benchmark integrator will be compared to those obtained with Brutus (see Chapter 4), which uses an arbitrary-precision library [1]. With this library we can specify the number of bits, L_w, used to store the mantissa, while the exponent has a fixed word-length. The length of the mantissa can be specified and increased, with the aim of controlling the round-off error.

---

[1] We use the open-source library GMP: http://gmplib.org/

The integration of the equations of motion is realised using the Verlet-Leapfrog scheme (**?**). The time-step is shared among all particles, but varies for every step according to Eq. 2.1.

To control the discretisation error, we implemented the Bulirsch–Stoer (BS) method, which uses iterative integration and polynomial extrapolation to infinitesimal time-step size (**??**). An integration step is accepted, when two subsequent BS iterations have converged to below the BS tolerance level, ϵ.

The time-step parameter η and the BS tolerance ϵ, both influence the performance. If η is too big, convergence may not be achieved for any tolerance. If η is too small, the many integration steps will render the integration too expensive. There is an optimal value for η as a function of ϵ. We measured this relation empirically, which results in:

$$\log_{10} \eta = A \log_{10} \epsilon + B. \tag{2.2}$$

For $\epsilon < 10^{-50}$ the power law converges to $A = 0.029$ and $B = 0.45$. Extrapolating this relation to $\epsilon > 10^{-50}$ will cause the time-step size to become larger than the time scale for the closest encounter in the system. Therefore this relation saturates to a flatter power law for $\epsilon > 10^{-50}$ with $A = 0.012$ and $B = -0.40$. Compared to a fixed value for η, this relation speeds up the iterative procedure by about a factor three or more. The code is implemented as a community code in the AMUSE framework (**?**) under the name Brutus.

2.2.3   Method of Convergence

For every simulation we have to define the BS tolerance parameter ϵ and the word-length L_w. In an iterative procedure we vary both parameters systematically, each time carrying out a simulation until $t = t_{\mathrm{end}}$. We subsequently calculate the phase space distance, $\delta^2_{A,B}$, between two solutions A and B:

$$\delta^2_{A,B} = \frac{1}{6N} \sum_{i=1}^{N} \sum_{j=1}^{6} \left(q_{A,i,j} - q_{B,i,j} \right)^2 . \tag{2.3}$$

The first summation is over all particles and the second summation is over the six phase-space coordinates denoted by q (**?**). We normalise by $6N$, so that δ represents the average difference per phase-space coordinate between two solutions A and B. In our experiments we adopt Hénon units [2] (**??**), in which the typical values for the distance

---

[2]Formerly known as N-body units. Introduced by D. Heggie at MODEST14.

and velocity are of the same order. We will also use the distance in just position or just velocity space as they might behave differently.

We consider the solutions A and B to be converged when $\delta_{A,B} < 10^{-p}$ at all times during the simulation. Note that converged in this case means convergence of the total solution, contrary to convergence per integration step as in the previous section. This criterion for convergence is roughly equivalent to comparing the first p decimal places of the positions and velocities for all N stars, in two subsequent calculations A, B. In most of our experiments we adopt $p = 3$, i.e. all coordinates have to converge to about three decimal places or more. We perform a subset of simulations with $p = 15$ to investigate the effect of small errors (see Sec. 4.2.4).

Each simulation starts by specifying the initial positions and velocities of N stars in double-precision (see Sec. 4.1). The simulation is carried out with the parameter set (ϵ, L_w). We start each simulation with $\epsilon = 10^{-6}$ and $L_w = 56$ bits. This corresponds to a level of accuracy similar to what we reach with the conventional Hermite integrator. After this simulation, we increase L_w, for example to 72 bits (~ 22 decimal places), redo the simulation and calculate δ^2. We repeat this procedure until $\delta < 10^{-p}$. When this is achieved, we have obtained a solution in which the round-off error is below a specified number of decimal places for this particular value of ϵ.

We now reduce the tolerance parameter ϵ, for example by a factor of 100, and repeat the procedure of increasing L_w. This series will again lead to a converged solution, but this time it is obtained using a smaller ϵ, and is likely to be different than the previous converged solution. We continue decreasing the value of ϵ by factors of 100 and repeat the procedure, until two subsequent iterations in ϵ lead to a converged solution with a value of $\delta < 10^{-p}$. By this time we are assured of having a solution to the gravitational N-body problem, that is accurate up to at least p decimal places.

In practice, we speed up the procedure by writing the word-length as a function of BS tolerance. Consider for example a BS tolerance of 10^{-20}. To reach convergence up to this level, we need at least 20 decimal places. Adding an extra buffer of 10 digits gives a total of 30 digits, or equivalently a word-length of about 112 bits. For this example, 112 bits turns out to be a good minimum word-length. For a first estimate of the word-length, we use:

$$L_w = 4 \left| \log_{10} \epsilon \right| + 32 \text{ bits.} \tag{2.4}$$

With this relation, we will only have to specify a single parameter ϵ, which directly controls the discretisation error and indirectly controls

the round-off error. For most of the systems in our experiment the discretisation error turns out to be the dominant source of error and as a consequence ϵ has to decrease quite drastically. When ϵ decreases, L_w increases, even up to the point that there are many more digits available than really needed to control the round-off error. In the case when the discretisation error dominates, the above defined minimum word-length for a given BS tolerance will result in the converged solution. When the round-off error dominates the word-length should be varied independently.

2.3   VALIDATION AND PERFORMANCE

2.3.1   The Pythagorean Problem

To show that our method works, we adopt the Pythagorean 3-body system (**?**). Previous numerical studies have shown that this system dissolves into a binary and an escaper (**??**). After many complex, close encounters the dissolution happens at about $t = 60$ time units (**?**), or about 16 crossing times.

   We adopt the initial conditions for the Pythagorean problem and integrate up to $t = 100$. To illustrate how the method works we start with a high tolerance and short word-length, ($\epsilon = 10^{-2}$, $L_w = 40$ bits), which is less precise than double-precision. In Fig. 2.1, this calculation is compared to a simulation with ($\epsilon = 10^{-4}$, $L_w = 48$ bits), through the yellow (upper) curves in the first three panels. After the first BS integration step, δ obtains a value of the order of the BS tolerance, and continues to increase due to exponential divergence, to eventually exceed $\delta \sim 10^{-1}$, after which the errors become on the order of the typical distance and speed in the system.

   In the following step, we repeat the calculation with a precision of ($\epsilon = 10^{-6}$, $L_w = 56$ bits), and compare the result with the calculation using ($\epsilon = 10^{-4}$, $L_w = 48$ bits). The comparison is represented by the orange curves (second from above) in Fig. 2.1. The overall behaviour of δ is similar, but the system diverges at a later time due to a higher initial precision.

   We continue the iterative procedure until a converged solution has been obtained. In the first three panels of Fig. 2.1, it can be seen that subsequent simulations with higher precision shift the curve to lower values of δ. Superposed on the steady growth of the error are sharp spikes, where the error grows by several orders of magnitude, after which the error restores again (**?**). These spikes are dominated by errors in the velocity, as can be deduced by comparing the magnitude
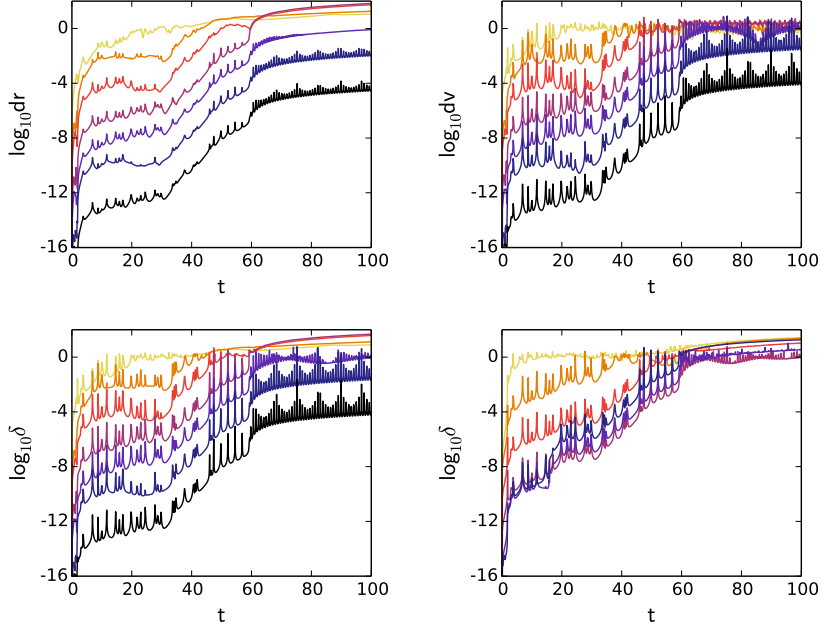
Figure 2.1: Exponential divergence in the Pythagorean problem. In the top two panels and the lower left panel, Brutus is compared with Brutus with increasing precision. The curves at the top of each panel compare a tolerance of 10^{-2} with 10^{-4}, the curve below compares 10^{-4} with 10^{-6} and so on. The word-length is a function of the tolerance as in Eq. 2.4. In the top left panel we show the distance in position-space, in the top right panel in velocity-space and in the bottom left panel in the full phase-space (all normalized by the number of stars and coordinates). The lower right panel compares the converged solution (the lowest curve in the other plots), with Hermite solutions with time-step parameters $\eta = 2^{-3}, 2^{-5}, 2^{-7}$ up to 2^{-13}.

of the spikes in position and velocity-space. Eccentric binaries which are out of phase when comparing two solutions cause large, periodic errors in the velocity. We finish the procedure when a solution is obtained for which the criterion for convergence is fulfilled, considering the magnitude of the error between the sharp spikes (bottom, black curves).

In the bottom right panel of Fig. 2.1, we compare solutions obtained by the Hermite integrator to the converged solution. The different curves belong to different time-step parameters; $\eta = 2^{-3}$, 2^{-5}, 2^{-7} up to 2^{-13}. Note that for a time-step parameter $\eta < 2^{-9}$, the curve is not shifted to lower values of δ, but even increases again. At this point round-off error becomes important, making the solution less accurate. The final close encounter in the Pythagorean problem occurs around 60 time units, after which a permanent binary and an escaper are formed. The Hermite integrator is able to accurately reproduce the evolution up to this point, but not subsequently, because δ has increased to values of order unity or higher. This can be explained by a small error in the final close encounter between all three stars, such that the direction of the escaper is slightly different.

To obtain the converged solution up to the first three decimal places, a tolerance of 10^{-14} and a word-length of 88 bits were needed. The simulation was about twice as slow compared to the Hermite simulation with $\eta = 2^{-9}$. The Hermite simulation, however, had a slightly different solution and a final, relative energy conservation of 10^{-8}, Decreasing the value of η will improve the level of energy conservation, but due to round-off error δ will not decrease.

2.3.2   The Equilateral Triangle

As a second test case, we adopt the 3-body equilateral triangle as an initial condition (Lagrange, 1772). In the exact solution this configuration remains intact, but small perturbations, such as produced by numerical errors, quickly cause the triangle to fall apart. For this problem, we also have a source of error in the initial conditions. Whereas the Pythagorean problem can be set up using integers, the initial condition for the equilateral triangle contains irrational numbers. To control the error in the initial condition, we calculate the initial coordinates with the same word-length as used for the simulation.

In the left panel of Fig. 2.2, a similar diagram is shown as for the Pythagorean problem in the lower left panel of Fig. 2.1. The starting precision is $\epsilon = 10^{-10}$ and the word-length is a function of ϵ as in Eq. 2.4. Subsequent simulations are performed with a 10 orders of
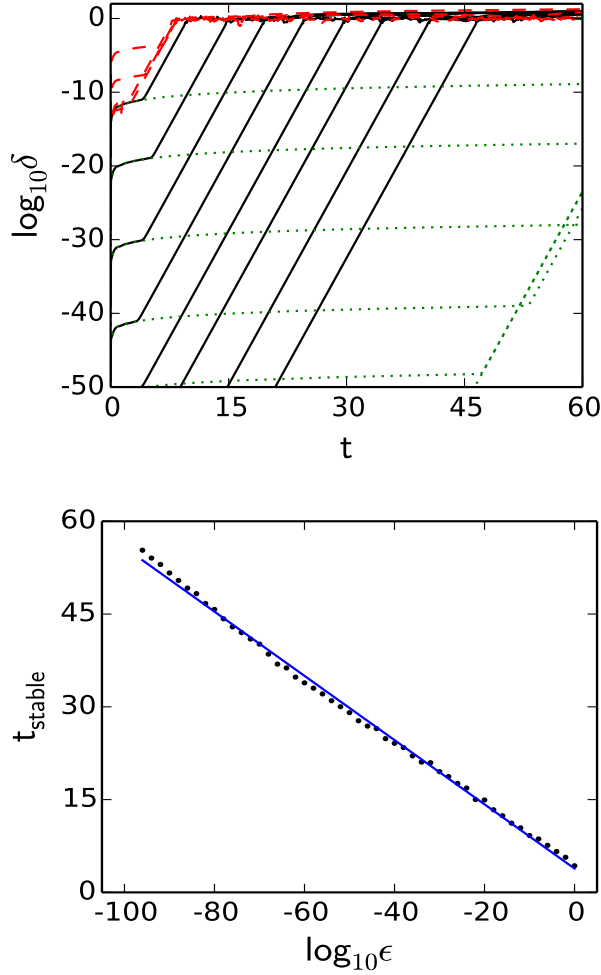
Figure 2.2: Divergence in the equilateral triangle configuration. In the top panel we show the divergence as a function of time. The solid curves compare Brutus solutions with increasing precision, where subsequent precisions are increased by 10 orders of magnitude and where the word-length is a function of tolerance as in Eq. 2.4. The dotted curves show results for similar simulations, but with a much longer, fixed word-length of 512 bits. The initial power law phase of divergence lasts longer in this case. The exponential divergence becomes dominant when the round-off error has had time to accumulate to become of the order the discretisation error. The dashed curves compare the highest precision Brutus solution with Hermite solutions with time-step parameters 10^{-1}, 10^{-2}, 10^{-3} and 10^{-4}. In the bottom panel we show for Brutus, the duration for which the triangular configuration remains intact as a function of Bulirsch–Stoer tolerance ϵ. Note that the time is in units of the period of one complete rotation of the system. The small scatter in the data is due to the discrete times at which we check the triangular configuration.

magnitude higher precision. For a short initial phase of 5 time units, the rate of divergence follows a power law. At later time, the solutions start to diverge exponentially with a characteristic rate independent of the tolerance and word-length. To investigate this transition, we redo the simulations with a large, fixed word-length of 512 bits (green dotted curves). This way, we reduce the amount of round-off error. As a consequence the rate of divergence is first dominated by the accumulation of discretisation errors and this phase lasts for a longer time, until the transition in the behaviour of the divergence, is reached, but now at ~ 45 time units. The time of the transition depends on word-length. Why the exponential divergence starts once the round-off error has kicked in, is a question that is still under investigation.

The red dashed curves in the same diagram in Fig. 2.2 give the results of the fourth-order Hermite, which are compared with the most precise Brutus simulation (with $\epsilon = 10^{-80}$, $L_w = 352$ bits). The time-step parameter $\eta = 10^{-1}$, 10^{-2}, 10^{-3} and 10^{-4} for subsequent curves. The Hermite integrations show a similar behaviour as the Brutus results, which could imply that the rate of divergence is a physical property of the configuration, rather than a property of the integrator.

In the right panel of Fig. 2.2 we show the duration for which the triangular configuration remains intact as a function of BS tolerance. For this experiment we halt the simulation when the distance between any two particles has increased or decreased by 10%, after which the triangle falls apart quickly. This diagram also illustrates the linear relation between accuracy and time in this system, which is caused by the constant number of digits being lost during every unit of time. The small scatter is due to the discrete times at which we check the triangular configuration. The solid, blue line is a fit to the data and its slope is $-0.52(3)$, which is equivalent to a loss of $1.9(1)$ digits per cycle.

2.3.3   A Plummer Distribution with N=16

As a third test we simulate the dynamical formation of the first hard binary in a small star cluster. We select a moderate number of sixteen equal mass stars and draw them randomly from a Plummer distribution (**?**). We integrate this system for about ten crossing times and apply the method of convergence. In Fig. 2.3 we present how two solutions with the same initial conditions, but different precisions, diverge as a function of time. The rate of exponential divergence, on average, starts rather constant, with a loss of $\sim 2/3$ digits per time unit. This is equivalent to an e-folding time of $t_e = 0.65$, which is consistent with
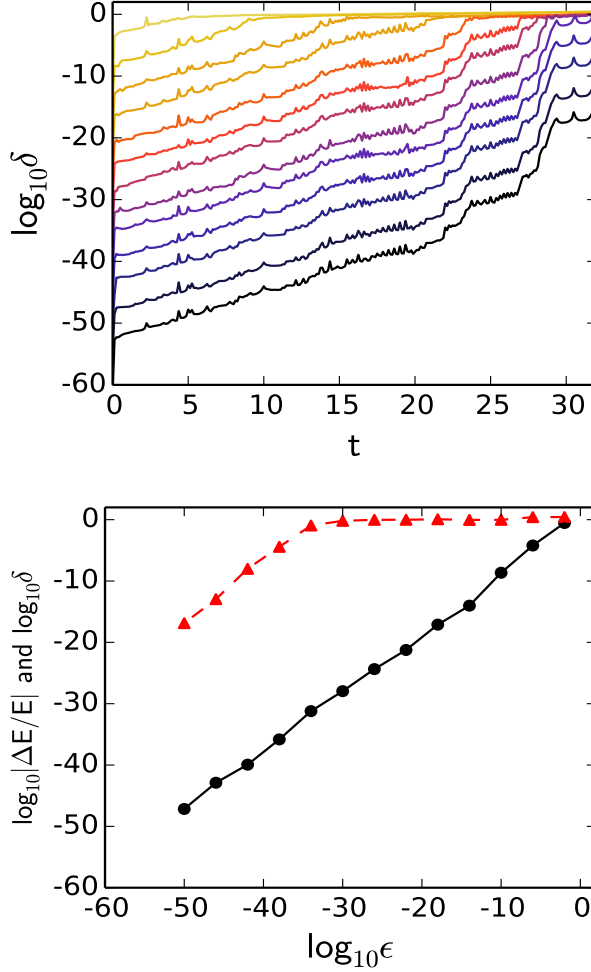
Figure 2.3: Exponential divergence in a 16-body cluster. In the top panel we illustrate the exponential divergence between Brutus simulations with increasing precision. In the bottom panel we show the final relative energy conservation (bullets, solid line) and the final normalized phase space distance between two subsequent simulations (triangles, dashed line) versus the Bulirsch–Stoer tolerance parameter ϵ. The solution starts to converge at a level of final relative energy conservation of $\sim 10^{-34}$.

the results of Goodman, Heggie and Hut (1993) (see their Fig. 8). From $t = 20$ onwards, the rate of divergence experiences systematic changes, in particular a steep rise of the error of about 10 orders of magnitude between $t = 26$ and $t = 29$. Such rises are a signature for the presence of a hard binary interacting with surrounding stars.

The right panel in Fig. 2.3 shows the energy conservation (black bullets, solid line) and the normalized phase space distance (red triangles, dashed line) versus ϵ. Energy conservation is proportional to ϵ, but the solutions only start to converge for $\epsilon < 10^{-34}$. More generally, even if conserved quantities like total energy are conserved to machine-precision or better, it is not guaranteed that the solution itself has converged.

The highest precision Brutus simulation in this example, ($\epsilon = 10^{-50}$, $L_w = 232$ bits), took about a day of wall-clock time, which is about 7000 times slower than a simulation with Hermite using $\eta = 2^{-9}$.

2.4   SCALING OF THE WALL-CLOCK TIME

The use of arbitrary-precision arithmetic dramatically increases the CPU time of N-body simulations. Also the BS method, which performs integration steps iteratively, makes an integration scheme more expensive by at least a factor two or more. To investigate for example how feasible it would be to run a converged N-body simulation for 10^3 stars through core collapse, we perform a scaling test in which we vary the number of particles and the precision, ϵ and L_w.

We randomly select positions and velocities for N equal mass stars from the virialised Plummer distribution (**?**), for $N = 2, 4, 8, ...,$ up to 1024. The BS tolerance is fixed at a level of 10^{-6} and the word-length at 64 bits. We integrate the systems for one Hénon time unit and measure the wall-clock time. In the top left panel in Fig. 2.4 we show the wall-clock time as a function of N, which fit the relation $t_{CPU} \propto N^{2.6}$.

For $N > 32$, it becomes efficient to parallellise the code. Our version implements i-parallellisation (Portegies Zwart et al., 2008) in the calculation of the accelerations. In the top right panel of Fig. 2.4, we plot the speed-up, S, against the number of cores. For $N = 1024$, we obtain a speed up of a factor 30 using 64 cores.

In the lower panels of Fig. 2.4 we present the scaling of the wall-clock time with BS tolerance and word-length. To measure the dependence on BS tolerance, we simulated a 16-body cluster for 1 Hénon time unit. We varied the BS tolerance while keeping the word-length fixed at $L_w = 1024$ bits. The relation obtained converges to $t_{CPU} \propto \epsilon^{-0.032}$.
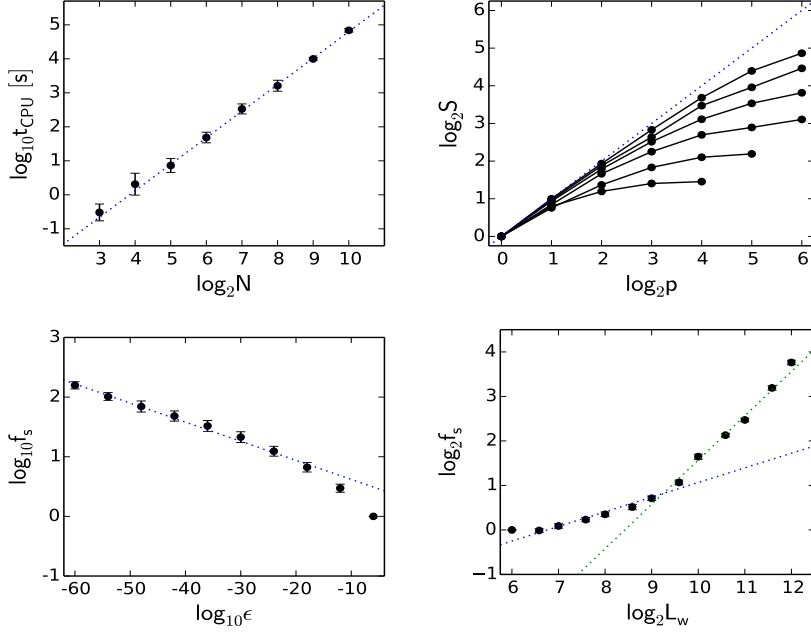
Figure 2.4: In the top left panel we show the scaling of the wall-clock time for Brutus as a function of number of stars N. The dotted curve is a fit to the data given by $t_{\mathrm{CPU}} \propto N^{2.6}$. In the top right panel we show the speed-up when the number of cores, p, is increased. The bottom, solid curve represents $N = 32$ and each curve above has an N a factor two higher than the previous curve. The dotted curve represents ideal scaling. In the bottom left panel we plot the slowdown factor as a function of the Bulirsch–Stoer tolerance ϵ, for a fixed word-length of 1024 bits. In the bottom right panel we plot the slowdown factor as a function of word-length L_w, for a fixed tolerance of 10^{-10}. The slowdown of the simulations is mainly caused by the very small Bulirsch-Stoer tolerances required.

A similar experiment was performed to measure the dependence on word-length. This time we fixed the BS tolerance at $\epsilon = 10^{-10}$ and varied the word-length. For $L_w < 1024$, the relation can be estimated as $t_{CPU} \propto L_w^{0.33}$, while for $L_w > 1024$, $t_{CPU} \propto L_w$. This transition depends on the internal workings of the arbitrary-precision library which we will not discuss here.

Using a very long word-length of 4096 bits, i.e. $\sim 10^3$ digits, results in a slowdown of a factor $f_s \sim 16$ compared to 64 bits. But for some simulations a BS tolerance smaller than 10^{-50} can easily be required to reach convergence, and this will result in a slowdown of a factor $f_s > 100$. The very small BS tolerance is often the main cause for the slowdown of the simulations, instead of the increased word-length.

Using the above results, we can construct the following model to estimate the wall-clock time for integrating 1 Hénon time unit with $L_w < 1024$ bits:

$$t_{CPU} = \left(\frac{N}{512}\right)^{2.6} \left(\frac{\epsilon}{10^{-6}}\right)^{-0.032} \left(\frac{L_w}{64}\right)^{0.33} 10^4 \, [s]. \qquad (2.5)$$

Integrating $N = 1024$ with standard precision, ($\epsilon = 10^{-6}$, $L_w = 64$ bits), up to core collapse at ~ 300 time units, and taking into account a speed up of a factor 30 due to parallellisation, we estimate a total wall-clock time of a week. Increasing the precision to ($\epsilon = 10^{-20}$, $L_w = 112$ bits), will take about a month. A precision of ($\epsilon = 10^{-50}$, $L_w = 232$ bits) will take roughly a year. To estimate how much precision is needed, we will assume that the rate of exponential divergence before the formation of the first hard binary is approximately constant. In the left panel of Fig. 2.3, the initial slopes correspond to a loss of $\sim 2/3$ digits per time unit. We construct the following approximate model for the initial BS tolerance needed to end up with a converged solution:

$$\log_{10} \epsilon = \log_{10} \delta_{\text{final}} - R_{\text{div}} t_{\text{cc}}. \qquad (2.6)$$

Here ϵ is the BS tolerance parameter, δ_{final} is the final precision of all the coordinates in the system, R_{div} is the approximately constant rate of divergence, e.g. the number of accurate digits lost per unit of time, and t_{cc} is the core collapse time. We set the final precision to 10^{-6}, i.e. convergence to the first 6 decimal places, and we set the core collapse time to ~ 300 as before. If we adopt $R_{\text{div}} = 2/3$, we estimate that we need an $\epsilon \sim 10^{-206}$. This would take about 10^5 years to finish. It would be more practical to simulate a 256-body cluster. If we set the

core collapse to 100 time units we estimate $\epsilon \sim 10^{-73}$, which would take about a month on a cluster of 64 Intel(R) Xeon(R) E5530 cores.

For direct N-body codes, the time for integrating up to core collapse usually scales as $\mathcal{O}(N^3)$. Using the analysis above, we estimate that the time for converged core collapse simulations scales approximately exponentially. This is effectively caused by the exponential divergence.

3

A Parallel Efficient
N-body Code: Sakura

We develop a Keplerian-based Hamiltonian splitting for solving the
gravitational N-body problem. This splitting allows us to approxi-
mate the solution of a general N-body problem by a composition of
multiple, independently evolved 2-body problems. While the Hamil-
tonian splitting is exact, we show that the composition of independent
2-body problems results in a non-symplectic non-time-symmetric first-
order map. A time-symmetric second-order map is then constructed
by composing this basic first-order map with its self-adjoint. The re-
sulting method is precise for each individual 2-body solution and pro-
duces quick and accurate results for near-Keplerian N-body systems,
like planetary systems or a cluster of stars that orbit a supermassive
black hole. The method is also suitable for integration of N-body
systems with intrinsic hierarchies, like a star cluster with primordial
binaries. The superposition of Kepler solutions for each pair of par-
ticles makes the method excellently suited for parallel computing; we
achieve $\gtrsim 64\%$ efficiency for only eight particles per core, but close to
perfect scaling for 16384 particles on a 128 core distributed-memory
computer. We present several implementations in Sakura, one of which
is publicly available via the AMUSE framework.

3.1 INTRODUCTION

Since the pioneering work of von Hoerner (1960), Aarseth (1963) and
? N-body simulations have been an essential tool for the theoretical
understanding of self-gravitating astrophysical systems. Such systems
often show a large dynamic range of time scales. Thus, instead of a

fixed or adaptive global time step, most of the N-body codes adopt
individual or block time step algorithms in order to advance the par-
ticles in time (**??**Aarseth, 2003). In addition, different approaches to
calculate the acceleration of each particle, such as using grids (**?**) or
a hierarchical tree data structure (**?**), are commonly employed to de-
crease the computational cost of the simulations. These approaches
allow the use of a larger number of particles, despite only giving an
approximation to the true acceleration of each particle. Therefore,
these codes should not inadvertently be used in simulations of colli-
sional systems such as planetary systems, dense star clusters or the
inner parts of galactic nuclei.

In collisional systems the individual interactions between particles
play an important role in the dynamical evolution of the system as
a whole. For example, the formation of hard binaries in star cluster
core collapse (**?**) requires very precise integration methods to correctly
evolve close encounters between particles. This precision is only pos-
sible if we use more accurate, direct brute-force methods, to calculate
the accelerations due to each pair of particles in the system. The main
difficulty here is that with the formation of the first hard binary in the
system, the simulation as a whole experiences a slow-down in perfor-
mance due to the necessity to decrease the time-step size in order to
accurately integrate such compact sub-systems.

Currently, the most effective and common approach to overcome
such obstacles seems to be a combination of the block time step al-
gorithm (**?**), Ahmad-Cohen neighbour scheme (**?**) and some sort of
2-body regularization (**????**) in order to handle very compact sub-
systems efficiently. This is the approach used in modern Hermite in-
tegrators for collisional stellar systems (**?**Aarseth, 2003).

In this chapter, we develop a new Keplerian-based Hamiltonian
splitting for the gravitational N-body problem. This splitting allows
us to approximate the solution of a general N-body problem by a
composition of independently evolved two-body problems. While the
Hamiltonian splitting is exact, we show in section 3.2 that the compo-
sition of independent two-body problems results in a non-symplectic
non-time-symmetric first-order map. A time-symmetric second-order
map is then constructed by composing this basic first-order map with
its self-adjoint. The advantages of this Keplerian-based integrator are:
i) a guarantee that every pair of particles is always integrated pre-
cisely; ii) the method does not suffer from slow-down in performance
when tight binaries are present in the simulation, and iii) the method
allows for good parallel efficiency (**?**).

3.2   METHOD

3.2.1   Hamiltonian Splitting

We begin the derivation of our scheme for the numerical integration of a gravitational N-body system by considering its Hamiltonian,

$$H = H_T + H_U \,. \tag{3.1}$$

Here,

$$H_T \equiv \sum_{i=1}^{N} H_{T_i} \,, \qquad H_{T_i} \equiv \frac{1}{2} m_i v_i^2 \,, \tag{3.2}$$

and

$$H_U \equiv \frac{1}{2} \sum_{i=1}^{N} \sum_{j \neq i}^{N} H_{U_{ij}} \,, \qquad H_{U_{ij}} \equiv -\frac{m_i m_j}{r_{ij}} \,, \tag{3.3}$$

are the kinetic and potential energies of the system, respectively; m_i and $v_i = |\mathbf{v}_i|$ are the mass and velocity of the i-th particle and $r_{ij} = |\mathbf{r}_{ij}| = |\mathbf{r}_i - \mathbf{r}_j|$ is the relative distance between particles i and j.

The time evolution of a Hamiltonian system is formally given by the operator[1] $e^{\tau \widehat{H}}$, which can be approximated by a composition of individually solvable operators $e^{\tau \widehat{H}_A}$ and $e^{\tau \widehat{H}_B}$ in cases when the Hamiltonian can be split as $\widehat{H} = \widehat{H}_A + \widehat{H}_B$. The simplest example of Hamiltonian splitting is the case when $\widehat{H}_A = \widehat{H}_T$ and $\widehat{H}_B = \widehat{H}_U$, for which we can generate the time-symmetric second-order Drift-Kick-Drift (DKD) variant of the Leapfrog integrator: $e^{\tau \widehat{H}} \approx e^{\frac{\tau}{2} \widehat{H}_T} e^{\tau \widehat{H}_U} e^{\frac{\tau}{2} \widehat{H}_T}$. This Hamiltonian splitting is not the only possibility and many other ways of subdividing the system have been tried (**?????**).

In the present chapter we introduce a way to split the Hamiltonian of an N-body system, which is based on two main arguments: i) the validity of the superposition principle[2], and ii) the existence of an analytical solution for the 2-body problem. Therefore, a natural way to approximate the time evolution of an N-body system is by using a composition of 2-body problems to solve a more general N-body problem. While this approach may seem computationally expensive,

---

[1] Hamiltonian associated operators are denoted by a $\widehat{}$ symbol.

[2] Recall that the gravitational potential and acceleration at the position of a given particle consists of a superposition of 2-body contributions due to the interaction with every other particle in the system.

our aim here is to present a theoretical formulation of the method. Possible optimizations, such as applying the Kepler-solver only to a few close pairs in the simulation, or to make use of Newton's third law during the force loop, are left for future implementations.

We first rewrite the potential energy term in Eq. 3.3 as follows:

$$
\begin{aligned}
H_U &= \frac{1}{2} \sum_{i=1}^{N} \sum_{j \neq i}^{N} H_{U_{ij}} \\
&= \frac{1}{2} \sum_{i=1}^{N} \sum_{j \neq i}^{N} -\frac{m_i m_j}{r_{ij}} \\
&= \frac{1}{2} \sum_{i=1}^{N} \sum_{j \neq i}^{N} -\mu_{ij} \frac{(m_i + m_j)}{r_{ij}} \\
&= \frac{1}{2} \sum_{i=1}^{N} \sum_{j \neq i}^{N} \mu_{ij} \left\{ \left[\frac{1}{2} v_{ij}^2 - \frac{(m_i + m_j)}{r_{ij}} \right] - \frac{1}{2} v_{ij}^2 \right\} \\
&= \frac{1}{2} \sum_{i=1}^{N} \sum_{j \neq i}^{N} \left(H_{K_{ij}} - H_{T_{ij}} \right) \\
&= \frac{1}{2} \sum_{i=1}^{N} \sum_{j \neq i}^{N} H_{W_{ij}} \equiv H_W \, .
\end{aligned}
\tag{3.4}
$$

Here,

$$
H_{K_{ij}} \equiv \mu_{ij} \left[\frac{1}{2} v_{ij}^2 - \frac{(m_i + m_j)}{r_{ij}} \right]
\tag{3.5}
$$

is the 2-body Keplerian Hamiltonian and

$$
H_{T_{ij}} \equiv \frac{1}{2} \mu_{ij} v_{ij}^2,
\tag{3.6}
$$

where $\mu_{ij} = m_i m_j / (m_i + m_j)$ is the reduced mass of the $i - j$ pair. The original N-body Hamiltonian in Eq. 3.1 can now be rewritten as follows:

$$
H = H_T + H_W = \sum_{i=1}^{N} H_{T_i} + \frac{1}{2} \sum_{i=1}^{N} \sum_{j \neq i}^{N} H_{W_{ij}} \, ,
\tag{3.7}
$$

with

$$
H_{W_{ij}} = H_{K_{ij}} - H_{T_{ij}} \equiv H_{U_{ij}} \, .
\tag{3.8}
$$

We note that Eq. 3.7, as is clear from the equivalence in Eq. 3.8, can always be reduced by simplification into Eq. 3.1, which implies that, in principle, our Keplerian-based Hamiltonian splitting does not change the dynamics of the system.

3.2.2 Equations of Motion

According to the general theory of geometric integrators (**?**) we can construct a time-symmetric second-order method by composing a (possible non-time-symmetric) first order method, $\phi(\tau)$, with its self-adjoint, $\phi^\dagger(\tau)$. Moreover, the composition $\Psi(\tau) = \phi(\frac{\tau}{2}) \circ \phi^\dagger(\frac{\tau}{2})$ is symplectic if both $\phi(\tau)$ and $\phi^\dagger(\tau)$ are symplectic methods.

In our Keplerian-based Hamiltonian splitting, time evolution operators can be constructed by taking into account that

$$e^{\tau \widehat{H}_T} = \prod_{i=1}^{N} e^{\tau \widehat{H}_{T_i}} , \tag{3.9}$$

$$e^{\tau \widehat{H}_U} = \prod_{i=1}^{N} e^{\tau \frac{1}{2} \widehat{H}_{U_i}} = \prod_{i=1}^{N} e^{\tau \frac{1}{2} \sum_{j \neq i}^{N} \widehat{H}_{U_{ij}}} . \tag{3.10}$$

and, by Eq. 3.4,

$$e^{\tau \widehat{H}_W} = \prod_{i=1}^{N} e^{\tau \frac{1}{2} \widehat{H}_{W_i}} = \prod_{i=1}^{N} e^{\tau \frac{1}{2} \sum_{j \neq i}^{N} \widehat{H}_{W_{ij}}} , \tag{3.11}$$

where the last term on the right hand side in eqs. 3.10 and 3.11 is a simple substitution of the definition of operators $\widehat{H}_{U_i}$ and $\widehat{H}_{W_i}$, i.e., $\widehat{H}_{U_i} = \sum_{j \neq i}^{N} \widehat{H}_{U_{ij}}$ and similar for $\widehat{H}_{W_i}$, and the presence of the factor $1/2$ follows from the fact that we have to take into account each $i - j$ pair only once. In eqs. 3.9, 3.10 and 3.11 the individual operators $e^{\tau \widehat{H}_{T_i}}$, $e^{\tau \widehat{H}_{U_i}}$ and $e^{\tau \widehat{H}_{W_i}}$ act on the $6N$ dimensional array $(\mathbf{r}_i, \mathbf{v}_i)$. Here the "one-subscript" operators individually commutate since they can only act on the corresponding coordinates with subscript i. Therefore, the order in which the product of operators is executed in each of eqs. 3.9, 3.10 and 3.11 is unimportant. In order to proceed with the derivation we present these operators in a more explicit form as follows:

$$e^{\tau \widehat{H}_{T_i}} : \quad \begin{pmatrix} \mathbf{r}_i \\ \mathbf{v}_i \end{pmatrix} \leftarrow \begin{pmatrix} \mathbf{r}_i \\ \mathbf{v}_i \end{pmatrix} + \tau \begin{pmatrix} \mathbf{v}_i \\ \mathbf{0} \end{pmatrix} , \tag{3.12}$$

$$e^{\tau \widehat{H}_{U_i}} : \quad \begin{pmatrix} \mathbf{r}_i \\ \mathbf{v}_i \end{pmatrix} \leftarrow \begin{pmatrix} \mathbf{r}_i \\ \mathbf{v}_i \end{pmatrix} + \tau \begin{pmatrix} \mathbf{0} \\ \mathbf{a}_i \end{pmatrix} , \qquad (3.13)$$

$$e^{\tau \widehat{H}_{W_i}} : \quad \begin{pmatrix} \mathbf{r}_i \\ \mathbf{v}_i \end{pmatrix} \leftarrow \begin{pmatrix} \mathbf{r}_i \\ \mathbf{v}_i \end{pmatrix} + \begin{pmatrix} \delta \mathbf{r}_i \\ \delta \mathbf{v}_i \end{pmatrix} , \qquad (3.14)$$

where $\mathbf{a}_i$ is the acceleration and $(\delta \mathbf{r}_i, \delta \mathbf{v}_i)$ are the increments in absolute coordinates and will be specified later on in Eq. 3.23.

In a similar way, individual "two-subscript" operators are explicitly written as follows:

$$e^{\tau \widehat{H}_{U_{ij}}} : \quad \begin{pmatrix} \mathbf{r}_{ij} \\ \mathbf{v}_{ij} \end{pmatrix} \leftarrow \begin{pmatrix} \mathbf{r}_{ij} \\ \mathbf{v}_{ij} \end{pmatrix} + \tau \begin{pmatrix} \mathbf{0} \\ \mathbf{a}_{ij} \end{pmatrix} , \qquad (3.15)$$

$$e^{\tau \widehat{H}_{W_{ij}}} : \quad \begin{pmatrix} \mathbf{r}_{ij} \\ \mathbf{v}_{ij} \end{pmatrix} \leftarrow \begin{pmatrix} \mathbf{r}_{ij} \\ \mathbf{v}_{ij} \end{pmatrix} + \begin{pmatrix} \delta \mathbf{r}_{ij} \\ \delta \mathbf{v}_{ij} \end{pmatrix} , \qquad (3.16)$$

$$e^{\tau \left(-\widehat{H}_{T_{ij}} \right)} : \quad \begin{pmatrix} \mathbf{r}_{ij} \\ \mathbf{v}_{ij} \end{pmatrix} \leftarrow \begin{pmatrix} \mathbf{r}_{ij} \\ \mathbf{v}_{ij} \end{pmatrix} - \tau \begin{pmatrix} \mathbf{v}_{ij} \\ \mathbf{0} \end{pmatrix} , \qquad (3.17)$$

$$e^{\tau \widehat{H}_{K_{ij}}} : \quad \mathbf{r}_{ij}, \mathbf{v}_{ij} \leftarrow \texttt{kepler\_solver}(\tau, m_{ij}, \mathbf{r}_{ij}, \mathbf{v}_{ij}) , \qquad (3.18)$$

where $m_{ij} = m_i + m_j$, $\mathbf{a}_{ij} = -m_{ij} \mathbf{r}_{ij} / r_{ij}^3$ is the relative 2-body acceleration. The increments in relative coordinates, $(\delta \mathbf{r}_{ij}, \delta \mathbf{v}_{ij})$, are obtained independently for each $i - j$ pair from the application of one of the first-order maps:

$$e^{\tau \left(\widehat{H}_{K_{ij}} - \widehat{H}_{T_{ij}} \right)} \approx e^{\tau \left(-\widehat{H}_{T_{ij}} \right)} e^{\tau \widehat{H}_{K_{ij}}} , \qquad (3.19a)$$

$$e^{\tau \left(\widehat{H}_{K_{ij}} - \widehat{H}_{T_{ij}} \right)} \approx e^{\tau \widehat{H}_{K_{ij}}} e^{\tau \left(-\widehat{H}_{T_{ij}} \right)} . \qquad (3.19b)$$

Eqs. 3.12 to 3.17 are first-order approximations to the respective operators in these equations. It will be clear below that this low-order approximation is enough for our purposes since, ultimately, the order of the full time evolution operator in Eq. 3.25 will be determined by the composition of those operators. In this sense, if a high-order approximation of the method presented here is needed, we argue that this should be obtained not by extending eqs. 3.12 to 3.17 to higher order, but rather, by making a high-order composition of these operators in a similar way as in symplectic integrators (**??**), where a second-order

map is constructed as a composition of first-order operators, and so on.

We notice here that, contrary to the "one-subscript" operators, the "two-subscript" operators act on the $6N(N-1)/2$ dimensional array $(\mathbf{r}_{ij}, \mathbf{v}_{ij})$. Therefore, it remains to be shown how to relate "one-subscript" and "two-subscript" operators in a consistent way. From Eq. 3.10 and the definition of $H_{U_{ij}}$, it is easy to see that the equivalence,

$$\prod_{\substack{j \neq i}}^{N} e^{\tau \widehat{H}_{U_{ij}}} \equiv e^{\tau \sum_{j \neq i}^{N} \widehat{H}_{U_{ij}}} = e^{\tau \widehat{H}_{U_i}} \,, \tag{3.20}$$

is valid for every N because the operators $e^{\tau \widehat{H}_{U_{ij}}}$ commutate. On the other hand, from Eq. 3.11, an equivalence similar to Eq. 3.20 relating $\widehat{H}_W$-type operators is only possible for $N = 2$. For $N > 2$ the operators $e^{\tau \widehat{H}_{W_{ij}}}$ do not commutate. However, we can write a similar equation approximately as

$$\prod_{\substack{j \neq i}}^{N} e^{\tau \widehat{H}_{W_{ij}} + \mathcal{O}(\tau^2)} \approx e^{\tau \sum_{j \neq i}^{N} \widehat{H}_{W_{ij}}} = e^{\tau \widehat{H}_{W_i}} \,, \tag{3.21}$$

where the error $\mathcal{O}(\tau^2)$ is not guaranteed to be Hamiltonian due to the fact that we treat each $i - j$ pair independently. As a consequence *the symplecticity of the present method is lost.*

Apart from the loss of symplecticity, as mentioned above, a time-symmetric second-order method for our Keplerian-based Hamiltonian splitting can still be constructed by using a composition of self-adjoint first-order methods (see **?**).

In order to construct $\phi(\tau)$ and $\phi^\dagger(\tau)$ we first need to specify the increments $\delta \mathbf{r}_i$ and $\delta \mathbf{v}_i$ in Eq. 3.14. Since in the present method we take advantage of a Kepler-solver to evolve each pair of particles independently, the relative increments $(\delta \mathbf{r}_{ij}, \delta \mathbf{v}_{ij})$ can be easily calculated for each interaction after application of one of the maps in eqs. 3.19a or 3.19b. Here, what we seek is an approximate relation between the increments in relative coordinates $(\delta \mathbf{r}_{ij}, \delta \mathbf{v}_{ij})$ and those in absolute coordinates $(\delta \mathbf{r}_i, \delta \mathbf{v}_i)$, in order to construct the full integrator. By noting that increments associated with operators $\widehat{H}_{U_{ij}}$ and $\widehat{H}_{U_i}$ are related by

$$\tau \begin{pmatrix} \mathbf{0} \\ \mathbf{a}_i \end{pmatrix} = \frac{1}{m_i} \sum_{j \neq i}^{N} \mu_{ij} \tau \begin{pmatrix} \mathbf{0} \\ \mathbf{a}_{ij} \end{pmatrix} \,, \tag{3.22}$$

a way to specify $(\delta \mathbf{r}_i, \delta \mathbf{v}_i)$ consists of exploring the equivalence between H_U and H_W, as first presented in Eq. 3.4. In addition, if we take into account the discussion above regarding to eqs. 3.10, 3.11, 3.20 and 3.21, a relation between relative and absolute increments can be defined in analogy to Eq. 3.22 as follows:

$$\begin{pmatrix} \delta \mathbf{r}_i \\ \delta \mathbf{v}_i \end{pmatrix} = \frac{1}{m_i} \sum_{j \neq i}^{N} \mu_{ij} \begin{pmatrix} \delta \mathbf{r}_{ij} \\ \delta \mathbf{v}_{ij} \end{pmatrix} + \mathcal{O}(\tau^2), \qquad (3.23)$$

which constitutes a first-order approximation as explained above (see Eq. 3.21). While we were not able to provide a more formal derivation to Eq. 3.23, we will show below (see explanation about Eq. 3.28) that when we calculate the relative increments from an ordinary Leapfrog map rather than the Kepler-solver in Eq. 3.18, then Eq. 3.23 reduces to Eq. 3.22.

We can now define a time-symmetric second-order map for our Keplerian-based Hamiltonian splitting as follows:

$$\begin{aligned} \Psi(\tau) & \equiv & \phi(\frac{\tau}{2}) \circ \phi^\dagger(\frac{\tau}{2}), \\ & \equiv & e^{\frac{\tau}{2} \widehat{H}_T} e^{\frac{\tau}{2} \widehat{H}_W} \circ e^{\frac{\tau}{2} \widehat{H}_W} e^{\frac{\tau}{2} \widehat{H}_T}, \end{aligned} \qquad (3.24)$$

where the increments $(\delta \mathbf{r}_{ij}, \delta \mathbf{v}_{ij})$ which appear in the $e^{\frac{\tau}{2} \widehat{H}_W}$ operator on the left side of $\circ$ are independently obtained after application of Eq. 3.19a for each $i - j$ pair, while those which appear on the right side of $\circ$ are independently obtained after application of the (self-adjoint) method in Eq. 3.19b for each $i - j$ pair. Eq. 3.24 can be further simplified by merging operators on both sides of $\circ$, giving,

$$\Psi(\tau) \equiv e^{\frac{\tau}{2} \widehat{H}_T} e^{\tau \widehat{H}_W} e^{\frac{\tau}{2} \widehat{H}_T}, \qquad (3.25)$$

in which case the increments $(\delta \mathbf{r}_{ij}, \delta \mathbf{v}_{ij})$ appearing in the $e^{\tau \widehat{H}_W}$ operator should be independently obtained after application of a time-symmetric second-order map for each $i - j$ pair,

$$e^{\tau \left(\widehat{H}_{K_{ij}} - \widehat{H}_{T_{ij}} \right)} \approx e^{\frac{\tau}{2} \left(-\widehat{H}_{T_{ij}} \right)} e^{\tau \widehat{H}_{K_{ij}}} e^{\frac{\tau}{2} \left(-\widehat{H}_{T_{ij}} \right)}. \qquad (3.26)$$

The equations of motion that result from the full map in Eq. 3.25 can be written in the following discrete form:

$$\mathbf{r}_i^{1/2} = \mathbf{r}_i^0 + \frac{\tau}{2}\mathbf{v}_i^0 \,, \tag{3.27a}$$

$$\tilde{\mathbf{r}}_i = \mathbf{r}_i^{1/2} + \frac{1}{m_i}\sum_{j \neq i}^{N} \mu_{ij}\delta\mathbf{r}_{ij} \,, \tag{3.27b}$$

$$\mathbf{v}_i^1 = \mathbf{v}_i^0 + \frac{1}{m_i}\sum_{j \neq i}^{N} \mu_{ij}\delta\mathbf{v}_{ij} \,, \tag{3.27c}$$

$$\mathbf{r}_i^1 = \tilde{\mathbf{r}}_i + \frac{\tau}{2}\mathbf{v}_i^1 \,, \tag{3.27d}$$

where $\mathbf{r}_i^1 = \mathbf{r}_i(t+\tau)$, $\mathbf{r}_i^0 = \mathbf{r}_i(t)$ and similar for $\mathbf{v}_i$, and the increments $(\delta\mathbf{r}_{ij}, \delta\mathbf{v}_{ij})$ are calculated independently as explained above.

As it can be seen, eqs. 3.27 are remarkably similar to the Leapfrog method. It remains to be shown that these equations effectively reduce to the Leapfrog equations when we substitute the 2-body Kepler-solver to a simple DKD-type integrator. In this case, the map in Eq. 3.26 becomes:

$$\mathbf{r}_{ij} \leftarrow \mathbf{r}_{ij} - \frac{\tau}{2}\mathbf{v}_{ij} \,, \tag{3.28a}$$

$$\mathbf{r}_{ij} \leftarrow \mathbf{r}_{ij} + \frac{\tau}{2}\mathbf{v}_{ij} \,, \tag{3.28b}$$

$$\mathbf{v}_{ij} \leftarrow \mathbf{v}_{ij} + \tau\,\mathbf{a}_{ij} \,, \tag{3.28c}$$

$$\mathbf{r}_{ij} \leftarrow \mathbf{r}_{ij} + \frac{\tau}{2}\mathbf{v}_{ij} \,, \tag{3.28d}$$

$$\mathbf{r}_{ij} \leftarrow \mathbf{r}_{ij} - \frac{\tau}{2}\mathbf{v}_{ij} \,, \tag{3.28e}$$

which results in $\delta\mathbf{v}_{ij} = \tau\mathbf{a}_{ij}$ and $\delta\mathbf{r}_{ij} = \mathbf{0}$ and, in view of eqs. 3.22 and 3.23, completes the demonstration. It should be noted that in this particular case, the error in Eq. 3.23 disappears because $\delta\mathbf{r}_{ij} = \mathbf{0}$ and Eq. 3.21 reduces to Eq. 3.20, restoring the symplecticity of the method. Note also that this is true only if we use a DKD-type integrator as a 2-body solver. For a KDK-type 2-body solver the symplecticity of the method is not restored because the order in which $(\mathbf{r}_{ij}, \mathbf{v}_{ij})$ is evolved in eqs. 3.28 changes and $\delta\mathbf{r}_{ij} \neq \mathbf{0}$. In other words, using a simple DKD-type integrator as a 2-body solver in the scheme above results in a very expensive implementation of a traditional Leapfrog method.

On the other hand, with the Kepler-solver function as a 2-body solver, a non-Hamiltonian error is made due to the non-commutativity

of the $e^{\tau \widehat{H}_{W_{ij}}}$ operators and the fact that each $i-j$ pair is treated independently, leading to the loss of symplecticity of the resulting method. Because our Keplerian-based integrator is constructed as a composition of self-adjoint first-order maps, it still preserves time-reversibility and second-order convergence (error $\mathcal{O}(\tau^3)$).

The advantage of using the Kepler-solver instead, comes from the fact that it is guaranteed that all pairwise interactions are always integrated precisely, which, in practical N-body simulations, is a much stronger requirement than the symplecticity of the Hamiltonian flow.

3.2.3   Implementation

The method described in the previous section has been implemented in a new code called Sakura, which is available in Astrophysical MUlti-purpose Software Environment (AMUSE[3], **?**). In order to clarify the implementation, Listing 3.1 shows a Python[4] code for the main loop calculation which evolves the particle's coordinates according to the map in Eq. 3.25 or, equivalently, eqs. 3.27. The Kepler-solver function at line 47 implements a universal variable Kepler-solver closely following **?**. Note that the memory and CPU requirements of this code scales as $O(N)$ and $O(N^2)$, respectively.

Listing 3.1: `Python` code for the main loop in Sakura integrator

```
1  """The functions below implement the main
2  steps of Sakura integrator.
3
4  The required parameters are the following:
5
6  :param tau: the time-step size.
7  :param n: the number of particles.
8  :param m: array with particles' masses.
9  :param r: 3D array with particles' positions.
10 :param v: 3D array with particles' velocities.
11 """
12
13 def do_step(tau, n, m, r, v):
14   r, v = evolve_HT(tau/2, n, m, r, v)
15   r, v = evolve_HW(tau, n, m, r, v)
16   r, v = evolve_HT(tau/2, n, m, r, v)
17   return r, v
18
19 def evolve_HT(tau, n, m, r, v):
```

---

[3]www.amusecode.org

[4]The actual implementation has been done in C/C++ for efficiency purposes.

```
20   for i in range(n):
21     for k in range(3):
22       r[i][k] += v[i][k] * tau
23   return r, v
24
25 def evolve_HW(tau, n, m, r, v):
26   # Allocate/initialize 3D arrays to store
27   # increments in position/velocity due to
28   # 2-body interactions.
29   dmr = numpy.zeros((n, 3))
30   dmv = numpy.zeros((n, 3))
31
32   # For each i-j pair, this corresponds to
33   # the Eq. 26 in the main text.
34   for i in range(n):
35     for j in range(n):
36       if i != j:
37         mij = m[i] + m[j]
38         mu = m[i] * m[j] / mij
39         for k in range(3):
40           rr0[k] = r[i][k] - r[j][k]
41           vv0[k] = v[i][k] - v[j][k]
42         ###
43         for k in range(3):
44           r0[k] = rr0[k] - vv0[k] * tau / 2
45           v0[k] = vv0[k]
46         #
47         r1, v1 = kepler_solver(tau, mij, r0, v0)
48         #
49         for k in range(3):
50           rr1[k] = r1[k] - v1[k] * tau / 2
51           vv1[k] = v1[k]
52         ###
53         for k in range(3):
54           dmr[i][k] += mu * (rr1[k] - rr0[k])
55           dmv[i][k] += mu * (vv1[k] - vv0[k])
56
57   # This corresponds to eqs. 27b and 27c
58   # in the main text.
59   for i in range(n):
60     for k in range(3):
61       r[i][k] += dmr[i][k] / m[i]
62       v[i][k] += dmv[i][k] / m[i]
63   return r, v
```

3.3   VALIDATION AND PERFORMANCE

In order to verify that Sakura performs well on collisional N-body systems, we present some tests for N ranging from a few to a thousand. We compare the results of Sakura to those obtained using a modified version of the Leapfrog integrator and a standard 4-th order Hermite integrator, available in the AMUSE framework. The modification in the Leapfrog integrator consists of the introduction of a routine to allow the use of adaptive time-steps. In this case the time-symmetry of the Leapfrog method is still preserved because we adopted the recipe for time-symmetrisation as suggested in **?**. A comparison of the computational costs and scalings with N is also presented. We emphasize that the base time-step size in each of the tests of Sakura is kept constant during the simulation, whilst in Leapfrog and Hermite integrations a shared adaptive time-step scheme has been adopted. The time-step criterion used within Leapfrog integrations is the time-symmetrized version of $\tau \sim \min((r_{ij}/a_{ij})^{1/2})$, whilst in Hermite code the standard Aarseth-criterion is used. For other details about these codes we refer the reader to the AMUSE documentation[3]. The value of the constant time-step size in Sakura is chosen in such a way that the same number of integration steps is taken as in the case of the Hermite integrations. Similarly, the time-step parameter in Leapfrog integrations is chosen to give approximately the same number of steps as in Hermite integrations. Note that, by construction, Sakura does not admit any softening parameter. Therefore, we also use zero softening in the other methods.

3.3.1   Small-N Systems

We start by presenting some numerical tests for well known simple small-N systems including the figure-eight system ($N = 3$; **?**), the Pythagorean system ($N = 3$; **?**) and the sun with planets[5] ($N = 10$; Ito & Tanikawa (2002)). We do not show results for a single binary system ($N = 2$), since in this case Sakura reduces to an ordinary Kepler-solver which gives a solution for the binary orbit accurate to machine precision. The simulation time spans 100 N-body units (**?**) in the case of the first two systems and 10^3 yr in the case of the solar system.

In Fig. 3.1, we present the relative energy error as a function of the average time-step size (left panels) and CPU time vs relative

---

[5]We include Pluto in our simulations of the solar system since we use the initial conditions as given in Ito & Tanikawa (2002).

energy error (right panels) for the figure-eight system (top panels), Pythagorean system (middle panels) and sun with planets (bottom panels), for the Leapfrog, 4-th order Hermite and Sakura. We note that for the figure-eight system the 4-th order Hermite usually performs better than Leapfrog and Sakura for a level of energy conservation $\lesssim 10^{-6}$. We attribute this to the fact that in this system the intrinsic time-step size of the particles does not change considerably during the orbital evolution and then, for smaller τ, the 4-th order convergence rate of the Hermite integrator outperforms Leapfrog and Sakura, which are of 2-nd order. We notice that in this case, where all three particles democratically interact among themselves, Sakura is not expected to be the most suitable method of integration due to the non-commutativity of 2-body interactions. Nevertheless, as we see in Fig. 3.1 (top panels), its performance is comparable to that of the Leapfrog integrator. For the Pythagorean system, which contains several close encounters between particles during its orbital evolution, all three integration methods are somewhat comparable, despite Sakura using constant time-steps and the other two methods using adaptive time-steps. For the solar-system, in which the orbital evolution of the planets is almost Keplerian, Sakura delivers about four orders of magnitude better energy conservation than Leapfrog, being also more precise than Hermite integration for time-steps $\gtrsim 10^{-3}$, while consuming the least amount of CPU time.

For those kind of systems, an integration step using Sakura is usually more expensive than an integration step using Hermite or Leapfrog by a factor $2 - 4$. Also, since all these codes scales as $O(N^2)$, these figures are expected to remain unchanged when the number of particles increases. However, due to the fact that Sakura can handle compact binaries and/or resolve close encounters even with constant τ, less time-steps are required for a given level of energy conservation implying that in these cases Sakura might outperform Hermite and Leapfrog integrations. In order to confirm this, we also include a test with a specially constructed initial condition which consists of a hierarchical binary system ($N = 4$) with two tight binaries orbiting around each other in a circular orbit with semi-major axis $a_{outer} = 1$ (N-body units). The particles in each tight binary are themselves in a circular orbit with semi-major axis a_{inner}. We have selected a semi-major axis ratio in the range $a_{outer}/a_{inner} = 10 - 1000$, and performed a simulation for these systems for a time span of one P_{outer}, i.e., the largest orbital period in the system (which is the same for all semi-major axis ratios). In Fig. 3.2 we present the relative energy error as a function of
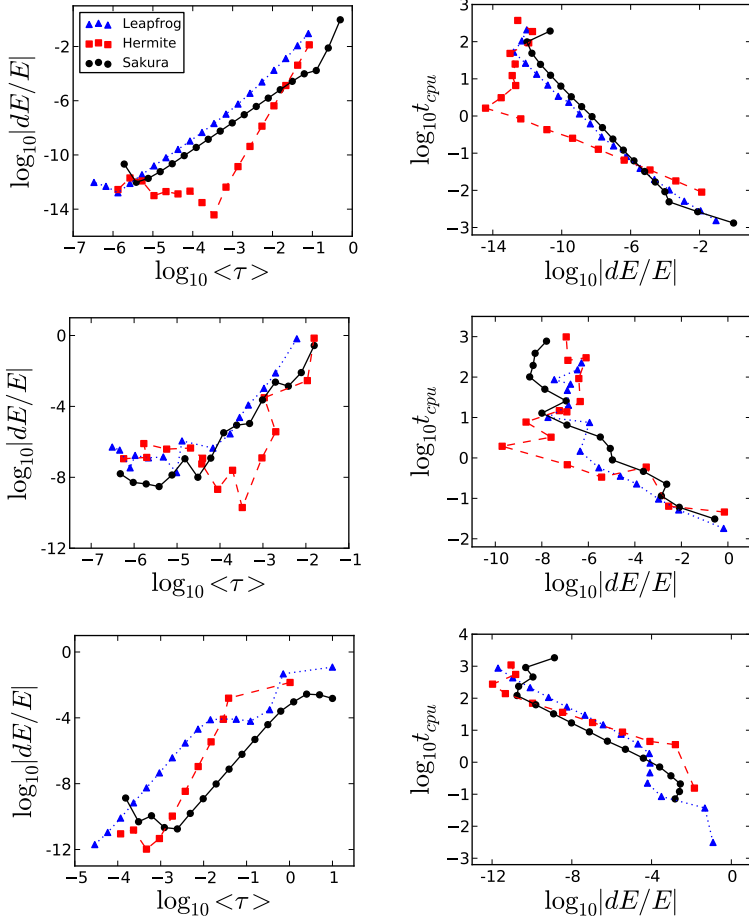
Figure 3.1: Relative energy error as a function of the average time-step size (left panels) and CPU time (in seconds) vs relative energy error (right panels) for the Leapfrog integrator (triangles), 4-th order Hermite (squares) and Sakura (bullets) for three different systems: figure-eight system (top panels), Pythagorean system (middle panels) and sun with planets (bottom panels). $\langle \tau \rangle$ is given in N-body units and stands for the average value of the shared adaptive time-step size in Hermite integrations.

the time-step size (left panels) and CPU time vs relative energy error (right panels) for the 4-th order Hermite, Leapfrog and Sakura.

For the $a_{\text{outer}}/a_{\text{inner}} = 10$ case (top panels in Fig. 3.2), Sakura delivers the same level of energy conservation as Leapfrog, although being more time consuming, whilst 4-th order Hermite has better energy conservation due to its higher order convergence for time-step sizes $\lesssim 10^{-2}$. However, for tighter interacting binaries (middle and bottom panels in Fig. 3.2), Sakura shows increasingly better performance with the compactness of the interacting binaries. In particular, for a level of energy conservation of 10^{-6}, typically adopted in collisional N-body simulations, Sakura is more than a order of magnitude faster than Hermite for the tightest binary configuration, $a_{\text{outer}}/a_{\text{inner}} = 1000$, while having a similar speed as Leapfrog. Also for the tightest binary configuration, Sakura is the most precise integration method for a range in time-steps of 6 orders of magnitude. On the other hand, for this latter system, the 4-th order Hermite results only start converging to good energy conservation when using time-steps $\lesssim 10^{-5.5}$, which in some circumstances might be impractical in computational terms, when systems of this kind are present in a large-scale simulation.

3.3.2 Large-N Systems

To test how Sakura behaves with a more general N-body problem, we use as initial condition a 128-body Plummer sphere containing a black-hole in its center. We assume equal mass for the stars and construct the system in virial equilibrium but for different black-hole to star mass ratios, $q \equiv M_{\text{bh}}/M_{\text{star}}$, ranging from $q = 1$ (no black-hole) to $q = 10^{12}$. We performed simulations for each of these initial conditions for 1 N-body time unit. Once again, the performance of Sakura is compared with that of the Leapfrog and standard 4-th order Hermite integrators. The results are shown in Fig. 3.3 which presents the relative energy error as a function of the mass ratio for time-step sizes $\langle \tau \rangle = 10^{-3}, 10^{-4}, 10^{-5}$ (top, middle and bottom lines), and Fig. 3.4 which present the CPU time vs relative energy error for different mass ratios: $q = 10^3$ (top left), $q = 10^6$ (top right), $q = 10^9$ (bottom left) and $q = 10^{12}$ (bottom right).

In Fig. 3.3 we see that the relative energy error for all three methods initially increases with the mass ratio till the point when $q \sim 10^2$. For larger mass ratios, the behaviour of Sakura clearly differs from the other two methods. While in Leapfrog and Hermite integrators the energy error stabilizes at a certain level, in Sakura we observe a very interesting trend in which its energy error decreases with increasing
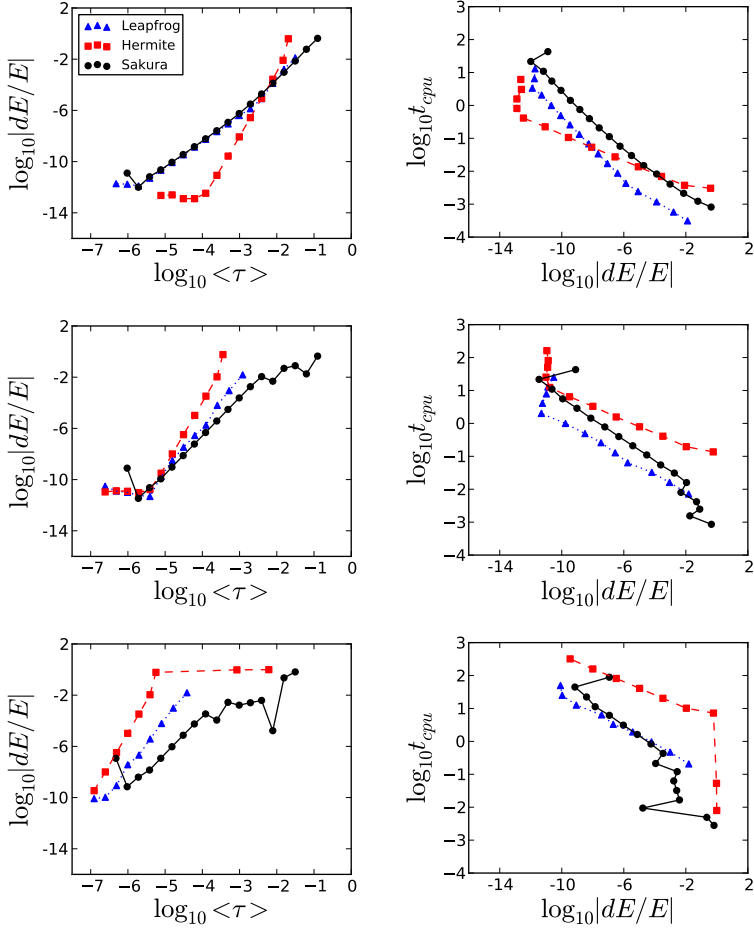
Figure 3.2: The same as Fig. 3.1 but for the hierarchical binary system for the following semi-major axis ratios: $a_{\mathrm{outer}}/a_{\mathrm{inner}} = 10$ (top panels), $a_{\mathrm{outer}}/a_{\mathrm{inner}} = 100$ (middle panels), $a_{outer}/a_{inner} = 1000$ (bottom panels).
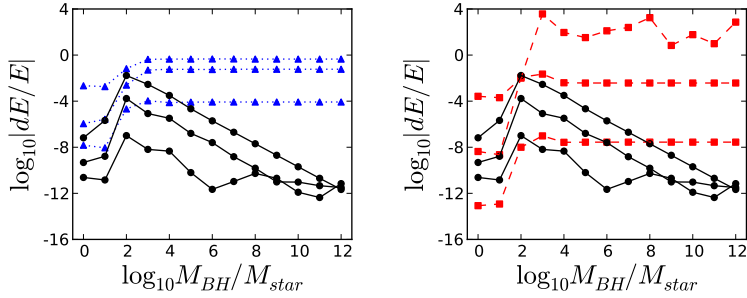
Figure 3.3: For a Plummer sphere with a central black hole, the panels show a comparison of the relative energy error as a function of the black-hole to stellar mass ratio for time-step sizes $\langle \tau \rangle = 10^{-3}, 10^{-4}, 10^{-5}$ (top, middle and bottom lines). The left panel present the results for Leapfrog (triangles) and Sakura (bullets) and the right panel present the results for 4-th order Hermite (squares) and Sakura (bullets).
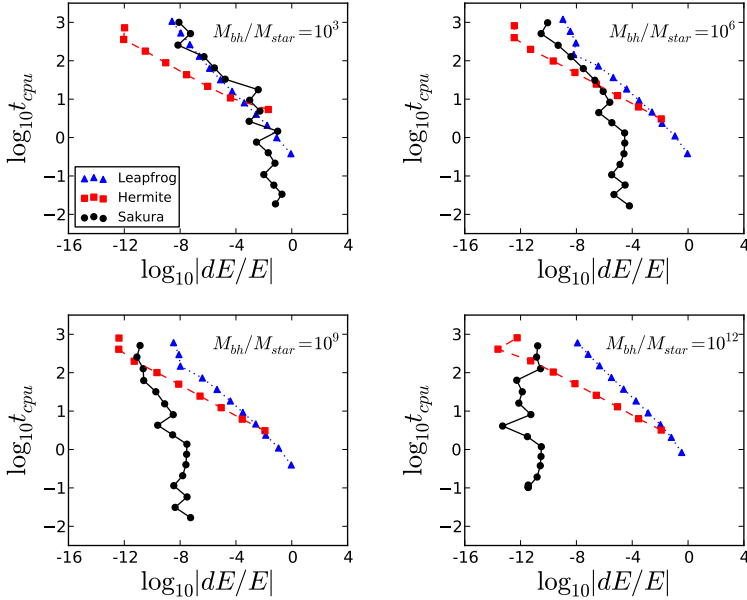
Figure 3.4: For the same system as in Fig. 3.3, the panels show the CPU time (in seconds) vs relative energy error for the following mass ratios: $q \equiv M_{\mathrm{bh}}/M_{\mathrm{star}} = 10^3$ (top left), $q = 10^6$ (top right), $q = 10^9$ (bottom left) and $q = 10^{12}$ (bottom right).

mass ratio. In other words, Sakura becomes more precise and therefore more efficient when the mass ratio grows, as can be seen in Fig. 3.4 for mass ratios (top left), $q = 10^6$ (top right), $q = 10^9$ (bottom left) and $q = 10^{12}$ (bottom right). An explanation of why these methods behave this way is as follows.

When no dominant massive particle is present in the system ($q \sim 1 - 10$), after only 1 N-body time unit the system has not evolved for enough time to form a close binary (which is an outcome of strong few-body interactions, see e.g. **?**). Therefore, in these circumstances most of the particles interact weakly among themselves and all the methods are able to integrate the orbital evolution of stars with relatively good energy conservation. Around a mass ratio $q \sim 10 - 10^3$ the massive particle quickly forms a binary system with a close neighbour, which eventually experiences several interactions with close perturbers, thus deteriorating the precision of the integration in all three methods. For mass ratios $q \gtrsim 10^3$ the orbital motion of stars becomes predominantly Keplerian. In this regime, the orbits in the system become mostly regular, and close encounters between stars become gradually less important. Therefore, the energy error is expected to converge to the truncation error associated to each of these methods. In Leapfrog and Hermite integrators, by decreasing the time-step size the energy conservation is thus improved but it remains approximately at the same level of conservation regardless the mass ratio (for $q \gtrsim 10^3$). On the other hand, Sakura departs from a constant level of energy conservation observed in the other two integrators, and becomes increasingly more precise with the mass ratio. This happens because in Sakura, the truncation error comes from two different sources: i) the error due to the Kepler-solver, which is essentially at machine precision, and ii) the error associated to the non-commutativity of 2-body interactions in close multiple-body encounters. With this knowledge, it is easy to intuitively understand why Sakura becomes more precise with the increase of the mass ratio: simply because the error associated to the non-commutativity of 2-body interactions becomes less important and, thus the overall error of the integrator converges to that of the Kepler-solver.

For $\langle \tau \rangle \sim 10^{-4}$, which corresponds to the middle lines (for each integrator) in Fig. 3.3, Sakura is ~ 5 (~ 6) orders of magnitude more precise than Hermite (Leapfrog), for a mass ratio $q = 10^6$. Also, as is shown in Fig. 3.4, Sakura's performance is similar to Leapfrog, for a mass ratio $q = 10^3$, and becomes gradually more efficient than Hermite and Leapfrog, when the mass ratio increases. This happens due to a change in slope of Sakura's curves in panels showing the CPU

time vs relative energy error when the mass ratio goes from $q = 10^3$ to $q = 10^{12}$ in Fig. 3.4, which means that for mass ratios $q \gtrsim 10^{12}$, Sakura can give very accurate results ($dE/E \sim 10^{-10} - 10^{-12}$) even when using relatively large time-steps, thus saving a big amount of computational time compared to Leapfrog and Hermite integrators.

As an additional general N-body test we performed a simulation of a 1024-body system through core collapse using Sakura with several time-step sizes $\tau = 10^0, 10^{-1}, 10^{-2}, 10^{-4}$, and the Leapfrog and standard 4-th order Hermite code using shared adaptive time-steps. For the parameter of precision we choose $\eta = 2^{-5} \approx 0.03$ in order to have a level of energy conservation of about 10^{-4} by the moment of core collapse in Hermite integration. In this particular test, we have used a parallel version of Sakura (see section 3.4) running on a 4-core Intel Xeon CPU @2.40 GHz. For the Leapfrog and Hermite codes (which are also parallelised) we setup the number of MPI processes to 4. In Fig. 3.5 we present the time evolution of the core radius using these codes. We see from this figure that for a sufficiently small time-step size ($\tau \sim 10^{-4}$, lowest black curve in Fig. 3.5) Sakura is able to evolve the system through core collapse. As expected from the exponential orbital instability (**?**), the results from Sakura slightly differ from Hermite and Leapfrog calculations. Apart from that, the core radius evolution obtained using Sakura follows remarkably well the results from the other two integrators.

In Sakura, the appearance of close binaries does not represent a computational challenge. Therefore, in this simulation no slow down in performance is observed, as is the case in most other N-body codes that also try to correctly evolve such compact sub-systems. As a consequence, the most expensive simulation using Sakura (bottom black line in Fig. 3.5) was completed in about three days of CPU time. The Leapfrog integration took about a week of processing time, whereas the Hermite simulation, after more than a month of CPU time (on the same machine), had not been completed, due to the dynamical formation of very close binaries and consequent decrease of the adaptive time-step size.

Although Sakura integrates all pairwise interactions exactly, the presence of close perturbers for a particular $i - j$ pair represents the main source of error during the integration. The reason for that originates from our assumption that each pair of particles can be treated as an independent 2-body problem during a time-step τ. If τ is larger than the time scale of interaction between the $i - j$ pair and its perturber, the perturbation will be delayed by τ, leading to spurious integration of a tight multi-component sub-system in an N-body sim-

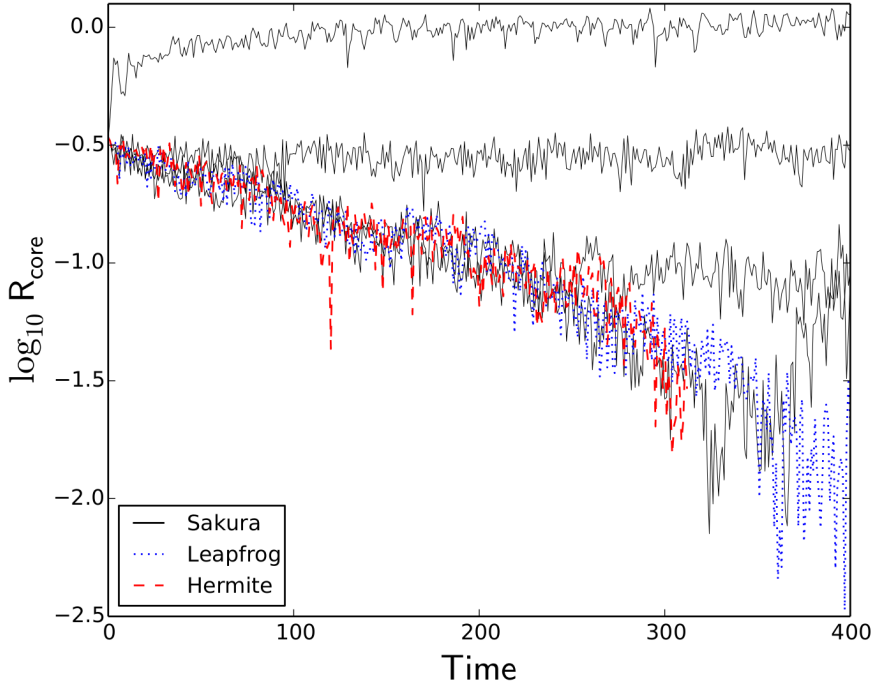Figure 3.5: Core radius vs simulation time for a 1024-body Plummer sphere. We compare Sakura using different time-step sizes (solid lines, $\tau = 10^0, 10^{-1}, 10^{-2}, 10^{-4}$ from top to bottom) to Leapfrog (dotted line) and standard 4-th order Hermite (dashed line), using shared adaptive time-steps with a parameter of precision $\eta \approx 0.03$. All the quantities are presented in N-body units.

ulation. This is a consequence of the non-commutativity of 2-body interactions. In Fig. 3.5, the use of relatively large time-steps reveals this issue: although the system as a whole stays bound, strong few-body interactions in the cluster core are not correctly integrated and as a consequence the core radius expands. However, by using smaller τ the numerical issues due to strong perturbations on the $i - j$ pair is diminished and as a consequence Sakura evolves the multi-component sub-systems that may form dynamically during the simulation more precisely. In those calculations, the level of energy conservation at the moment of core collapse stayed within $dE/E \lesssim 10^{-4}$ for Hermite, and $dE/E \lesssim 10^{-2}$ for Leapfrog and Sakura (for the bottom black line in Fig. 3.5), even though Sakura used a constant time-step.

The possibility to include a variable time-step scheme in Sakura might improve its results and is currently under investigation. The fact that Sakura evolves each pair of particles exactly, implies that the time-step criterion does not need to be so restrictive as in the case of traditional integration schemes. For example, if we consider the case of a hierarchical triple system in which the orbital period of the inner binary is a certain factor shorter than the time-scale of interaction between the binary and the outer perturber, we have observed in our tests (not reported here) that choosing a time-step size comparable to the longest time-scale still preserves the binary orbital evolution. In traditional codes, this would not be possible and the inner binary would end up being artificially disrupted if the time-step size has not been decreased to a fraction of its orbital period. Therefore, for Sakura we suspect that a time-step criterion based on the closest perturber distance to a given pair being evolved seems to be a more appropriate choice than an Aarseth-like time-step criterion. We will further discuss this issue on section 3.5.

3.4 PARALLELIZATION

We have implemented three different versions of Sakura: i) a single GPU implementation using OpenCL; ii) a distributed memory parallel implementation using MPI, and iii) a serial implementation in C/C++ (used in all the tests presented above, with exception of the one in Fig. 3.5, for which the MPI version was used). The parallelisation schemes adopted for distributed memory and GPU versions are quite similar as those adopted for conventional N-body codes on those platforms (see Portegies Zwart et al. 2008 and **?**, respectively). At the current stage of development our GPU implementation is not yet very efficient due to many branch conditions present in the Kepler-solver.
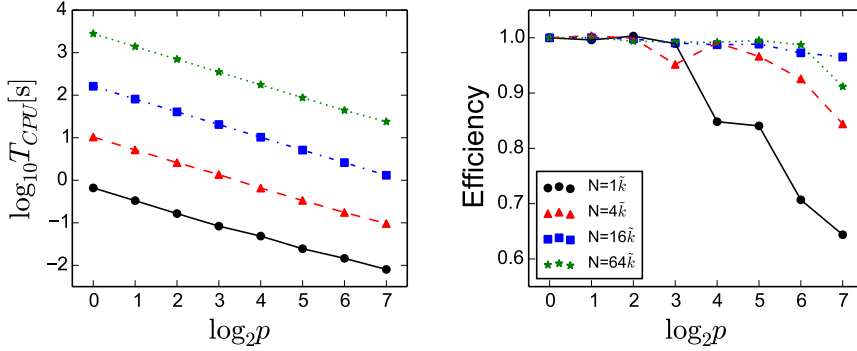
Figure 3.6: For the MPI version of Sakura the plots show the strong scaling (left panel) and the parallel efficiency (right panel) for four different problem sizes: $N = 1\tilde{k}$ (solid lines), $N = 4\tilde{k}$ (dashed lines), $N = 16\tilde{k}$ (dot-dashed lines) and $N = 64\tilde{k}$ (dotted lines). Here, $\tilde{k}$ stands for 1024 and p is the number of processor cores used for the run.

Here we mainly present some performance results using the MPI version of Sakura for tests using up to 128 CPU cores. The test simulations consist of a Plummer sphere with N equal mass particles being integrated for 1 N-body time unit. We use four different number of particles $N = 1\tilde{k}$, $4\tilde{k}$, $16\tilde{k}$, $64\tilde{k}$ ($\tilde{k}$ stands for 1024) and in each case we measure the total wall clock time needed to complete the simulation with different number of cores. In Fig. 3.6 we present, for four different problem sizes, the performance measurements in the form of the strong scaling ($T_{CPU}(p)$ vs p) and the parallel efficiency:

$$\text{Efficiency} \equiv \frac{T_{CPU}(p)}{p T_{CPU}(1)} , \tag{3.29}$$

where $T_{CPU}(p)$ is the CPU time measured when using p processor cores.

As is evident from the Fig. 3.6, Sakura exhibits an almost perfect strong scaling (top panel) and a remarkably good parallel efficiency (bottom panel). For the worst case scenario presented here ($N = 1\tilde{k}$, using 128 CPU cores), Sakura achieves a parallel efficiency as good as 64%, even though the workload in this case is as small as 8 particles per core. In addition, the strong scaling plot shows that, even in this worst case scenario, the CPU time could still be decreased by using a higher number p of processor cores. For $N > 4\tilde{k}$, the parallel efficiency of Sakura stays very close to 100%.

3.5   SUMMARY AND DISCUSSION

We have described a Keplerian-based Hamiltonian splitting for gravitational N-body simulations and its implementation in a new code called Sakura. In this method a general N-body problem can be solved as a composition of multiple, independent, 2-body problems. The integration scheme is constructed on the assumption that, during a small time interval τ, each pair of particles in the system can be treated as an independent 2-body problem. With this splitting an analytical Kepler-solver can be used to accurately, and independently, evolve each 2-body interaction in the system, thus making the code especially suitable for simulations in which compact primordial binaries or close dynamically formed binaries are present. Hierarchies in which one of the components is a compact binary and systems with a central dominant mass are also examples of physical systems in which Sakura performs well when compared to traditional codes.

Because Sakura can easily handle arbitrarily compact binaries in an N-body simulation, the code is able to evolve a star-cluster through core-collapse without much difficulty. In particular, since Sakura can do this even with the use of constant time-steps, the simulation does not suffer from any slow down in performance as is the case in other non-regularized N-body codes. As an example, in the 1024-bodies core-collapse simulation presented in section 3.3.2, Sakura was able to complete the run in about 3 days of CPU time on a 4-core machine. The same system being integrated with a 4-th order Hermite integrator took more than one month of CPU time on the same machine, due to a severe slow down in performance after the formation of the first hard binary in the system.

There are, however, some circumstances in which Sakura may not be the most suitable code to perform an N-body simulation. For example, for systems in which multiple bodies democratically interact among themselves, Sakura may perform almost as badly as a simple Leapfrog integrator, as demonstrated in the integrations of a figure-eight system in section 3.3.1. This happens because of our underlying assumption that the N-body problem can be decomposed in multiple, independent, 2-body problems. Such decomposition in fact constitutes the main source of error when a given $i-j$ pair is being integrated with a time-step τ which is larger than the time-scale of the perturbation due to a close neighbour. In many cases this issue may be surpassed by decreasing the constant time-step size used in the simulation. However, the cause of the problem lies on the non-commutativity of 2-body interactions when multiple bodies are involved in a democratic close

encounter. While it is not easy to solve this issue without breaking our Keplerian splitting approach, the introduction of an adaptive time-step scheme in Sakura might alleviate these numerical difficulties and is currently under investigation.

According to some of our tests (not reported in the present paper), a time-step criterion based on the strength of the perturbation on a given $i - j$ pair seems to work relatively well compared to a constant τ. However, this improvement is only significant when close multiple-body encounters take place. On the other hand, one could in principle choose $\tau \sim \min(r_{ij}/v_{ij})$, $\tau \sim \min((r_{ij}/a_{ij})^{1/2})$ or use a traditional Aarseth-like time-step criterion, but we advocate that this may not be the optimal choice because these criteria also include the contribution of the $i-j$ pair itself, which in principle contributes to a severe decrease in time-steps if a close binary is present in the system. In Sakura, these severely short time-steps are not necessary, because the use of a Keplerian treatment for each pair of particles automatically regularizes every 2-body interaction in the system. It is only when multiple-body encounters happens that the time-step should adapt itself to properly resolve the approximation of a perturber. Therefore, we stress here our preference for a perturbation-based time-step criterion rather than an Aarseth-like criterion for use in Sakura. Whether or not such perturbation-based criterion is the best choice for Sakura is a matter that will be addressed elsewhere.

Another point we want to emphasize here is the behaviour of Sakura when integrating a system with a central massive black-hole. As shown in Fig. 3.3, the level of energy conservation in Leapfrog and 4-th order Hermite integrations remains approximately constant with the increase of the black-hole to stellar mass ratio. For Sakura, we found that it performs much better than previous approaches, becoming gradually more precise with the increase of the mass ratio. In particular, for the case of a mass ratio $q = 10^6$ Sakura can give $\gtrsim 5$ orders of magnitude better energy conservation than Hermite integrator, being at the same time up to 4 orders of magnitude faster when the mass ratio increases to $q \gtrsim 10^9$. The fact that Sakura can be, at the same time, fast and accurate in this regime, makes this code highly suitable for nearly Keplerian systems where a massive particle dominates the evolution of surrounding particles, such as in planetary systems and galactic nuclei with super-massive black-holes.

Lastly, Sakura has proven to be quite easy to parallelise for distributed memory systems using MPI. The GPU implementation, even though theoretically easy, is still not totally efficient due to the presence of many branching conditions in the Kepler-solver. In algorithmic

terms, the bulk of computation in Sakura occurs inside a double loop, similar to the one used to calculate the acceleration of particles in conventional N-body codes. Therefore, we were able to immediately employ existent parallelisation schemes in Sakura without much effort. We argue that the fact that our GPU implementation is not yet very efficient is not a problem due to the parallelisation scheme itself, but rather due to the poor/inefficient support for branch conditions in current GPUs. A restructure in our Kepler-solver in order to eliminate (or minimize) these branch conditions may address this issue, and possibly speed up even more the MPI version on CPUs, which has already shown a remarkable parallel efficiency, with close to 100 percent efficiency for $16\tilde{k}$ particles on 128 cores, and 64 percent efficiency when using only eight particles per core.

4

On the Reliability
of N-body Simulations

As mentioned in Chapter 2, the general consensus in the N-body community is that statistical results of an ensemble of collisional N-body simulations are accurate, even though individual simulations are not. In order to test this assumption, we developed the new N-body code Brutus that solves the N-body problem to a pre-defined precision.

Using this new, brute force N-body code, we test the reliability of N-body simulations by a controlled numerical experiment. In this experiment we perform a series of resonant 3-body simulations, where the term resonant implies a phase or multiple phases during the interaction where the stars are more or less equidistant (Hut & Bahcall, 1983). These phases are intermingled by ejections, where a binary and single star are clearly separated. We perform the simulations with conventional double-precision, and with arbitrary-precision to reach the converged solution. In Sec. 4.1 we explain the experiment in more detail, and in Sec. 4.2 we compare the solutions individually to investigate the distribution of the errors. We also compare the global statistical distributions using two-sample Kolmogorov–Smirnov tests (**??**).

In summary, we find that on average at least half of the conventional simulations diverge from the converged solution, such that the two solutions are microscopically incomparable. For the solutions which have not diverged significantly, we observe that if the integrator has a bias in energy and angular momentum, this propagates to a bias in the statistical properties of the binaries. In the case when the conventional solution has diverged onto an entirely different trajectory in phase-space, we find that the errors are centred around zero and symmetric; the error due to divergence is unbiased, as long as the time-

step parameter, $\eta \leq 2^{-5}$ and when simulations which violate energy conservation by more than 10% are excluded. For resonant 3-body interactions, we conclude that the statistical results of an ensemble of conventional solutions are indeed accurate.

4.1 PRECISION OF STATISTICAL RESULTS: EXPERIMENTAL SETUP

In the previous section we demonstrated that it is possible to obtain a converged solution for a particular initial condition. We have also shown that a solution obtained by Hermite diverges from the converged solution, even up to the point that the microscopic solution given by Hermite is beyond recognition. We now perform a statistical study, to examine the hypothesis that double-precision N-body simulations produce statistically indistinguishable results, from those obtained from an ensemble of converged solutions with the same set of initial conditions. Because it is computationally expensive to reach convergence, we start investigating the hypothesis above by exploring the accuracy of 3-body statistics.

The $N = 3$ experiment is inspired by the Pythagorean problem, where after a complex 3-body interaction, a binary and an escaper are formed. As a variation to this, we define four different sets of initial conditions as follows:

1. Plummer distribution equal mass

2. Plummer distribution with masses 1:2:4

3. Plummer distribution equal mass with zero velocities

4. Plummer distribution with masses 1:2:4 and zero velocities.

The positions and velocities of the three stars are selected randomly from a virialised Plummer distribution (**??**). For the cold collapse systems, we set the velocities to zero. Then we rescale the positions and velocities to virialise the systems if the initial velocities are non-zero, or we set the total energy equal to $E = -0.25$ if the system starts out cold. We adopt standard Hénon units (**??**) throughout.

In the case of the cold initial conditions, the systems start democratically, i.e. the minimal distance between each pair of particles is greater than N^{-1}. We reject initial conditions in which this criterion is not satisfied. This is to prevent initial realisations where two stars which are very near, fall to each other radially causing very long

wall-clock times for the integration. When starting with a democratic configuration, there will also be an initial close triple encounter (**?**), which is hard to integrate accurately and is therefore a good test. A total of 10000 random realisations are generated for each set of initial conditions and can be found in the accompanying data files.

We stop the simulations when the system is dissolved into a permanent binary and an escaper. The criteria used to detect an escaper are the following:

1. escaper has a positive energy, $E > 0$,

2. is a certain distance away from the center of mass, $r > 2\,r_{\mathrm{virial}}$,

3. is moving away from the center of mass, $r \cdot v > 0$,

The energy of the escaper is calculated in the barycentric frame of the three particles and r_{virial} is the virial radius of the system, which is of the order unity in Hénon units.

There may be situations in which a star is ejected without actually escaping from the binary. After a long excursion the star turns around and once again engages the binary in a 3-body resonance (Hut & Bahcall, 1983). Because these systems need to be integrated for a longer time, they also require higher precision to reach convergence, which takes a long time to integrate (see also **?**). To deal with this issue, we perform the simulations iteratively by increasing the final integration time t_{end}. Starting with $t_{\mathrm{end}} = 50$ Hénon time units, we evolve every system and detect those that are dissolved. Then we increase t_{end} to 100, 150, 200 etc., but only for those systems which have not yet dissolved. A complete ensemble of solutions is obtained up to $t_{end} \sim 500$, or equivalently ~ 180 crossing times where the crossing time has a value of $2\sqrt{2}$ in Hénon units (**??**). Systems which take a longer time to integrate are not taken into account in this research. The fraction of long-lived systems is however a statistic we measure. We gathered the final, converged configurations in the accompanying data files.

Each initial realisation is run with the Hermite code, using standard double-precision, and with Brutus, using arbitrary-precision until a converged solution is obtained. At the end of each simulation, we investigate the nature of the binary and the escaper. In addition to the BS tolerance, word-length, CPU time and dissolution time, we record the mass, speed and escape direction of the escaping single star, and the semimajor axis, binding energy and eccentricity of the binary. In this way, we obtain statistics for $N = 3$ generated by a conventional N-body solver and by Brutus.

4.2   RESULTS

Before we perform a detailed comparison between results obtained by
Hermite and Brutus, we first compare the Brutus results with ana-
lytical distributions from the literature in order to relate to previous
studies. We compare Hermite and Brutus on a global level by perform-
ing two-sample Kolmogorov–Smirnov tests (**??**) to see whether global
distributions are statistically indistinguishable. We also compare the
distribution of lifetimes of triples to see whether precision influences
the stability and we measure the typical CPU time and BS tolerance
needed to obtain a converged solution. After this, we compare Her-
mite and Brutus per individual system, with the aim of investigating
the nature of the differences of every individual outcome. Finally, we
define categories which classify a conventional simulation as a preser-
vation or exchange, depending on whether the identity of the escaping
star is consistent between Hermite and Brutus.

4.2.1   Brutus versus Analytical Distributions

In Fig. 4.1, the distributions obtained by converged solutions are given
for the following quantities: velocity and kinetic energy of the escaper
in the barycentric reference frame, and semimajor axis, binding energy
and eccentricity of the binary. We start by looking at the eccentricity
distributions (bottom panel in Fig. 4.1). These distributions can be
estimated analytically by assuming that the probability of a certain
configuration is proportional to the associated volume in phase space
(**??**) or by considering an equilibrium distribution of binary stars in
a cluster (Heggie, 1975). The resulting thermal distribution in the
three-dimensional case is given by

$$f(e) = 2e, \qquad\qquad (4.1)$$

and in the two-dimensional case by

$$f(e) = \frac{e}{\sqrt{1 - e^2}}, \qquad\qquad (4.2)$$

The 3-body cold collapse problem is essentially a two-dimensional
problem. We compare the empirical and theoretical distributions by
means of the K–S test (see also next section). It turns out that the
distributions in eccentricity are statistically distinguishable. By in-
spection by eye we observe that in the virialised case, there are slight
deviations at high eccentricities. In the case of the equal-mass, cold
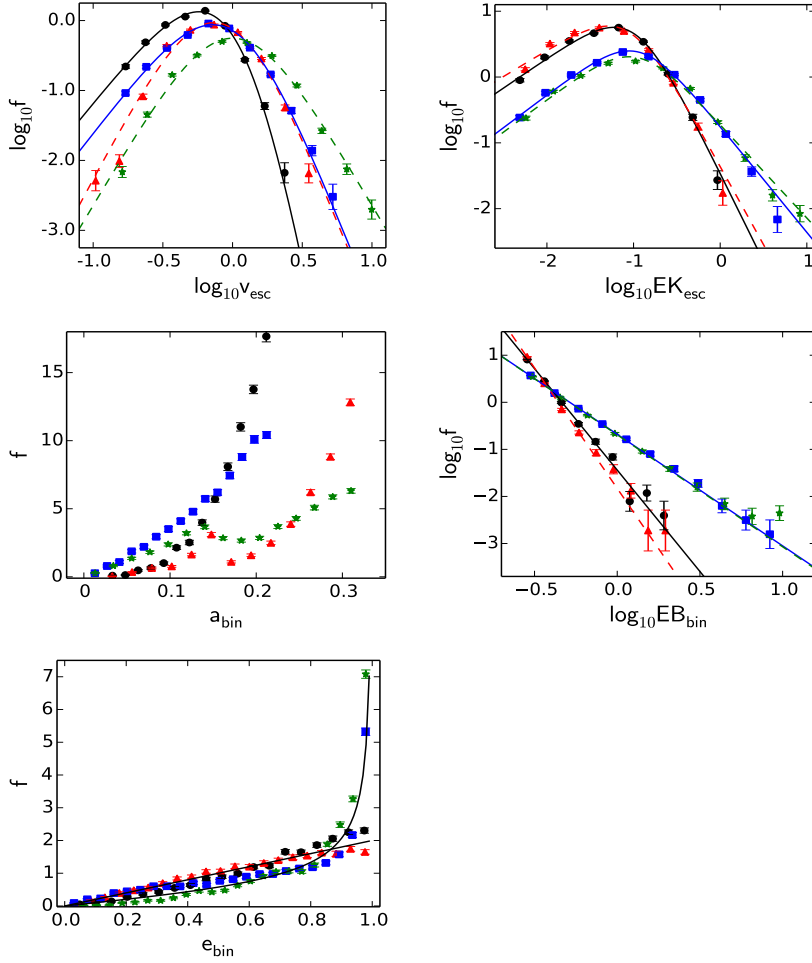
Figure 4.1: Comparison of Brutus results and analytical distributions. Distributions are given for the escaper speed (top left) and kinetic energy (top right), binary semimajor axis (middle left), binding energy (middle right) and binary eccentricity (bottom). The results from the Brutus simulations are represented by the data points, for each of the four sets of initial conditions: Plummer equal mass (bullets), Plummer with different masses (triangles), cold Plummer equal mass (squares) and cold Plummer with different masses (stars).. Note that we use standard Hénon units (**??**). Analytical models from the literature are fitted to the empirical distributions represented by the curves. For the eccentricities we plot the thermal distributions.

Velocity	α		β	
Plummer equal mass	2.5 $\pm$ 0.09		6.7 $\pm$	1.02
Plummer mass ratio	3.8	0.16	4.4	0.43
Cold Plummer equal mass	2.6	0.19	3.8	0.28
Cold Plummer mass ratio	3.4	0.45	3.4	0.19
Kinetic energy				
Plummer equal mass	0.9	0.02	1.8	0.04
Plummer mass ratio	0.8	0.02	1.6	0.04
Cold Plummer equal mass	0.99	0.02	1.3	0.03
Cold Plummer mass ratio	0.98	0.03	1.2	0.02
Binding energy				
Plummer equal mass	4.31	0.13		
Plummer mass ratio	5.12	0.32		
Cold Plummer equal mass	2.37	0.11		
Cold Plummer mass ratio	2.38	0.12		

Table 4.1: Fitted power law indices for the velocity and kinetic energy distributions of the escaping stars and for the binding energy distribution of the binary stars. Note that we use equal intervals in logarithmic space.

systems, there are more low eccentricity binaries compared to the theoretical prediction. They coincide at an eccentricity of about 0.7, after which they deviate again. For the cold systems with unequal masses, this behaviour is the other way around. The analytical predictions are able to capture the empirical distributions only in a qualitative manner.

The velocity distribution of the single escaping star can be estimated analytically in a similar way as was done for the eccentricities. The resulting distribution is predicted to be a double power law given by (**??**):

$$f(v) \propto \frac{v^{\alpha}}{(1 + \gamma v^2)^{\beta}}. \tag{4.3}$$

We fit this model to the data (see Fig. 4.1, first panel) and obtain values for α and β which are given in Table 4.1. The power law indices vary with mass ratio and total angular momentum. To remove the dependence on mass ratio, we plot the kinetic energy of the escaper (see Fig. 4.1, top right panel). Again, we fit a double power law of a similar form as Eq. 4.3, and the power law indices are given in Table 4.1. Both the escaper velocity and kinetic energy are consistent with a double power law distribution.

The binary semimajor axis and binding energy are related quantities. We fit the binding energy distribution (see Fig. 4.1, middle right panel) to a power law (Heggie, 1975; **?**; **?**):

$$f(E_{\mathrm{B}}) \propto E_{\mathrm{B}}^{-\alpha}. \tag{4.4}$$

The fitted power law indices are given in Table 4.1. The empirical distributions are consistent with a power law, although somewhat steeper than predicted (Heggie, 1975; **?**; **?**). The slopes do tend to vary somewhat as a function of angular momentum (**??**).

The empirical distributions obtained by Brutus are in qualitative agreement with the analytical estimates present in the literature (Heggie, 1975; **?**; **?**). Slight variations are present due to the dependence on total angular momentum, a limited statistical sampling and assumptions made in the derivation of the analytical distributions. Nevertheless, a similar qualitative agreement has been obtained between the analytical distributions discussed above and empirical distributions from an ensemble of conventional numerical solutions, e.g. not converged (**?**, chapters 7–8 and references therein). The question remains to what extend conventional and converged solutions agree quantitatively.
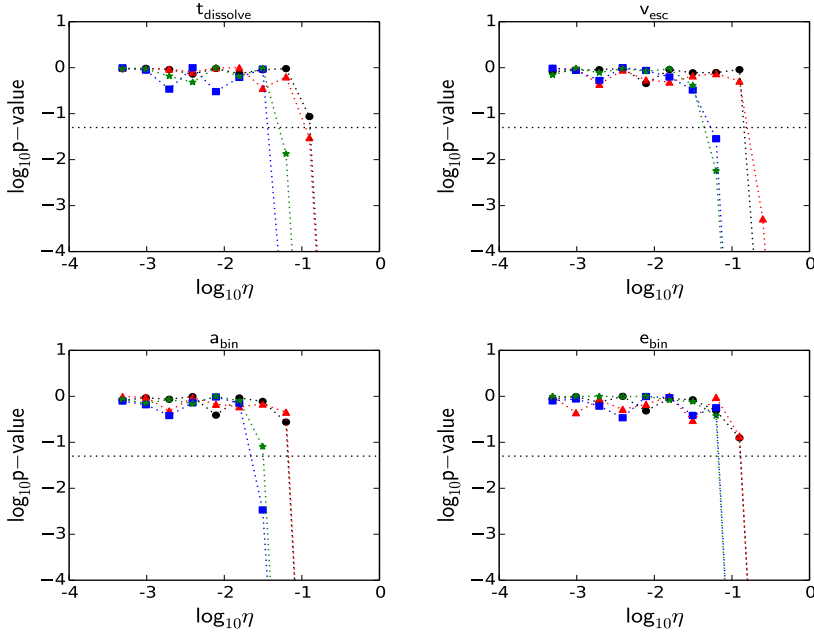
Figure 4.2: Two-sample K–S tests on distributions obtained by Hermite and Brutus. We compare distributions of dissolution time (top left), escaper speed (top right), binary semimajor axis (bottom left) and binary eccentricity (bottom right). Two-sample K–S tests are performed and the p–value is plotted versus Hermite time-step parameter η. The dashed line represents the 5% significance level. For $\eta < 2^{-5}$, the distributions are not significantly different. (The different curves represent the different data sets similar as in Fig. 4.1)

4.2.2   Brutus versus Hermite: Global Comparison

A quantitative way to compare global distributions is by performing two-sample Kolmogorov–Smirnov tests (K–S tests) (**??**). The K–S test gives the likelihood that two samples are drawn from the same distribution, quantified by the value called p. When the p-value is below five percent, the distributions are considered to be significantly different.

In Fig. 4.2 we plot the p-value obtained by comparing the Brutus distribution with the Hermite distribution versus time-step parameter η used for Hermite. In the panel showing the data for the binary semimajor axis, the distributions of the cold systems become significantly different for $\eta > 2^{-6}$. The distributions from the initially virialised systems start to differ for $\eta > 2^{-4}$. The cold systems are harder to model accurately, because of the close encounters that occur shortly after
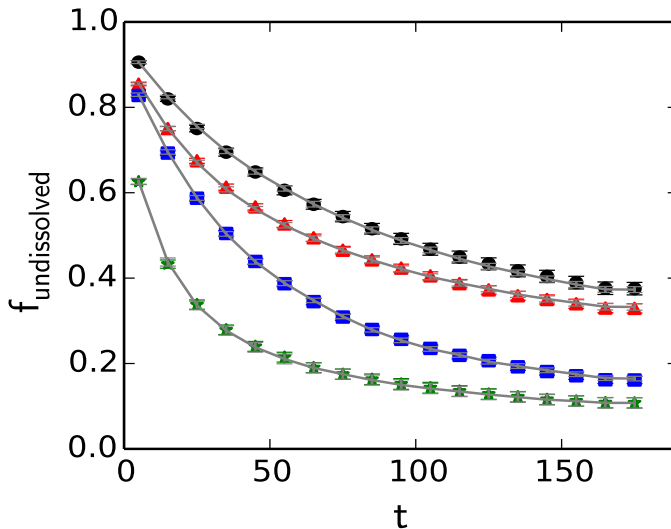
Figure 4.3: Lifetime of triple systems. We plot the fraction of triple systems that have not dissolved yet into a permanent binary and escaping single star configuration, as a function of simulation time (in units of crossing time). The curves through the data points represent the interpolated Hermite results with a time-step parameter $\eta = 2^{-5}$. (The different curves represent the different data sets similar as in Fig. 4.1)

the start. The reason the distributions start to become significantly different at large time-steps is because at these large time-steps most simulations violate energy conservation by $|\Delta E/E| > 0.1$. When this occurs, solutions might reach regions in $6N$-dimensional phase-space, which theoretically are forbidden. The distribution then becomes biased by these outlier solutions.

4.2.3 Lifetime of Triple Systems

In Fig. 4.3, we present the fraction of triple systems which are undissolved, i.e. still interacting, as a function of time. The results by Brutus are represented by the data points: equal-mass Plummer (black bullets), Plummer with different masses (red triangles), equal-mass cold Plummer (blue squares) and cold Plummer with different masses (green stars). The results by Hermite for a time-step parameter $\eta = 2^{-5}$ are represented by the curves appearing to go through the data points.

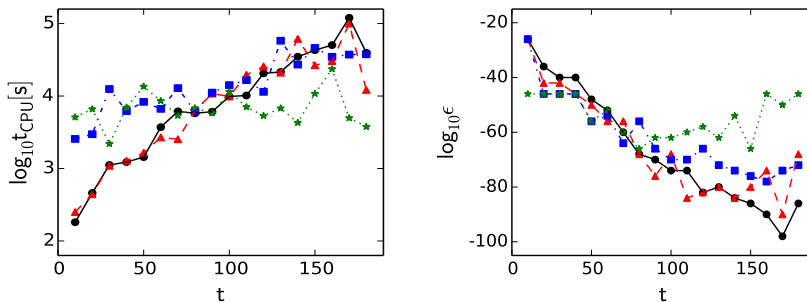Figure 4.4: CPU time and precision as a function of time for Brutus. On the left, we plot the CPU time of the simulation which took the longest, as a function of dissolution time. On the right, we plot the Bulirsch–Stoer tolerance of the simulation which needed the highest precision, as a function of dissolution time. The different curves represent the four sets of initial conditions as in the previous plots.

The initially cold systems dissolve faster than the initially virialised systems. This is somewhat expected due to the close triple encounter resulting from the initial cold collapse: the rate of energy exchange can be very high for these encounters (**?**). After ~ 180 crossing times, about 40% of the systems which started with an equal-mass Plummer initial configuration, are undissolved, compared to about 10% for the cold Plummer with different masses. Systems which include stars with different masses dissolve faster than their equal mass counterparts. Energy equipartition tends to cause the lightest particle to quickly reach the escape velocity.

In Fig. 4.3, the grey curves through the data points represent the interpolated Hermite results. Even though Hermite and Brutus use different algorithms and precisions to solve the equations of motion, we find that the lifetime of an unstable triple is statistically indistinguishable between converged Brutus and non-converged Hermite solutions (but see also Sec. 4.3.3).

In Fig. 4.4, we plot the maximum CPU time and minimum BS tolerance, both as a function of dissolution time. This is shown for the Brutus simulations, for the four different initial conditions. The longer it takes for a system to dissolve, the longer the CPU time and the higher the precision needed to reach a converged solution. To reach ~ 180 crossing times, there are systems which require a BS tolerance of the order 10^{-100}, with the final converged run taking of the order a few days. The average CPU time as a function of time is about an order of magnitude smaller than the maximum CPU time. The average BS

tolerance ranges from $\sim 10^{-20}$ to 10^{-30}. For systems which dissolve within 100 crossing times, Brutus is on average about a factor 120 slower than Hermite.

We were able to obtain a complete ensemble of systems dissolving within ~ 180 crossing times. Simulations which take longer than this are not taken into account in this experiment. The fraction of long-lived systems as obtained by Hermite and Brutus are consistent. For our purpose of comparing results from conventional integrators with the converged solution, integrating up to ~ 180 crossing times is sufficient, in the sense that there is enough time for conventional solutions to diverge from the true solution (see Sec. 4.2.4). Including the long-lived triple systems may however influence the statistical distributions and biases on the long term.

4.2.4 Brutus versus Hermite: Individual Comparison

For the individual comparison, we take a certain initial realisation and compare the solutions of Hermite and Brutus. In Fig. 4.5 we show scatter plots of the Hermite solution (with time-step parameter $\eta = 2^{-5}$) versus the converged Brutus solution for the equal-mass Plummer data set.

Data points on the diagonal represent accurate solutions, whereas the scatter around it represents inaccurate Hermite solutions. The diagonal is present in each panel and extends throughout the range of possible outcomes. The width of the diagonal is very narrow. When the normalized phase-space distance between the Hermite and Brutus solution $\delta < 10^{-1}$, then the coordinates are accurate enough to produce derived quantities accurate to at least one decimal place and Hermite and Brutus will give similar results. Once $\delta > 10^{-1}$, the solution has diverged to a different trajectory in phase-space leading to a different outcome. This outcome could in principle be any of the possible outcomes as can be derived from the amount of scatter in the Hermite solutions at a fixed Brutus solution.

In the scatter plot of the dissolution time, we observe that for small times ($t < 10$), Hermite and Brutus agree on the solution in the sense that the data points lie on the diagonal. Systems which dissolve after a short time don't have sufficient time to accumulate enough error to diverge to another trajectory in phase-space. Once however this level of divergence is reached, the scatter immediately covers the entire, available outcome space. This randomisation is also observed in the other panels.
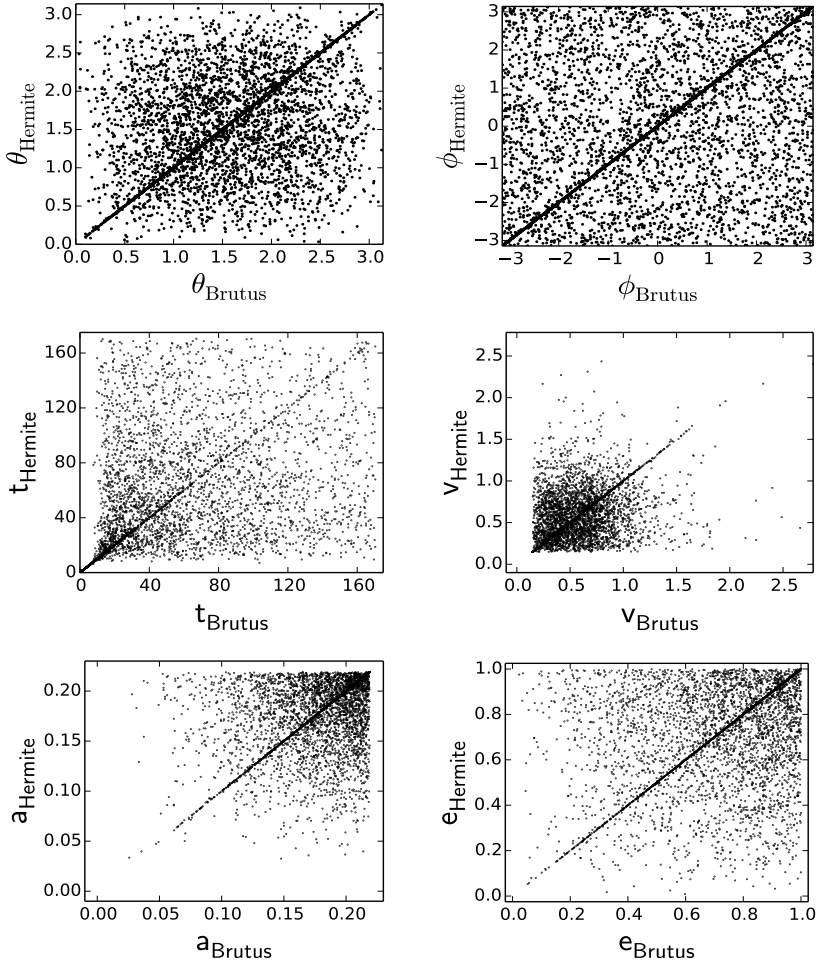
Figure 4.5: Direct comparison of Brutus and Hermite results per individual simulation. The results are shown only for the $N = 3$ equal mass Plummer data set and for a Hermite time-step parameter $\eta = 2^{-5}$. Each dot in a panel represents a different initial realisation. The value on the ordinate is the value obtained using Hermite and the value on the abscissa the value obtained by Brutus. We compare the direction of the escaper: polar angle (top left) and azimuthal angle (top right), (with respect to the plane of the binary and pericentre direction), dissolution time (middle left), escaper velocity (middle right), binary semimajor axis (bottom left) and binary eccentricity (bottom right). The diagonal represents accurate Hermite solutions. The scatter around it represents solutions where Hermite and Brutus have diverged.

Figure 4.6: The fraction of accurate Hermite simulations as a function of Hermite time-step parameter η. The different curves represent the different data sets: equal mass Plummer (bullets), Plummer with different masses (triangles), equal mass cold Plummer (squares) and cold Plummer with different masses (stars). As η decreases, the accurate fraction increases. However, for $\eta < 2^{-7}$, the fraction starts to saturate, more so for the cold data sets. At this point the effect of round-off error becomes important.

The Fraction of Accurate Solutions

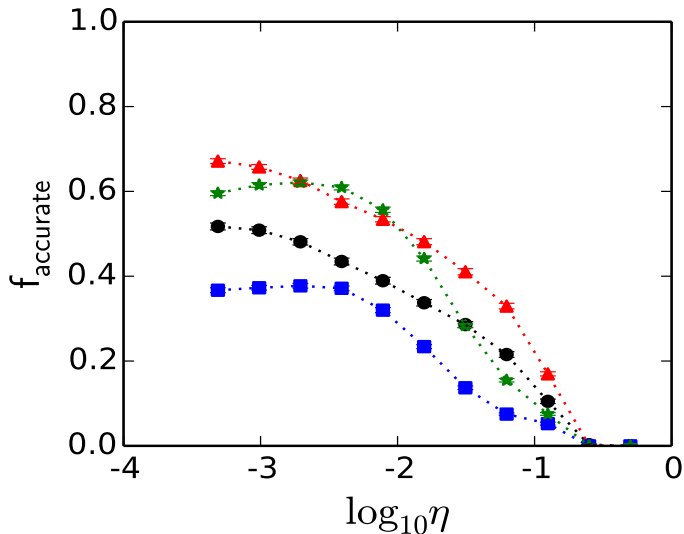In Fig. 4.6 we estimate the fraction of data points on the diagonal as a function of the Hermite time-step parameter, η. We only include the data points for which the normalized phase-space distance $\delta < 10^{-1}$. For the largest time-step parameters used ($\eta > 10^{-1}$) the fraction on the diagonal, or the accurate fraction, varies from zero to about 0.2. By reducing the time-step parameter, the accurate fraction increases until it saturates at about 0.4 to 0.7 depending on the initial conditions. Even though by reducing η, the discretisation error decreases, the number of integration steps increases, which then increases the round-off error. For the data sets with zero angular momentum, the maximum accurate fraction is obtained for $\eta \sim 2^{-9}$. For the initially virialised systems this seems to occur between $\eta \sim 10^{-3} - 10^{-4}$, although the actual saturation point is not visible yet. This dependence on angular momentum is due to the initial cold collapse and subsequent close encounters, which increases the round-off error.
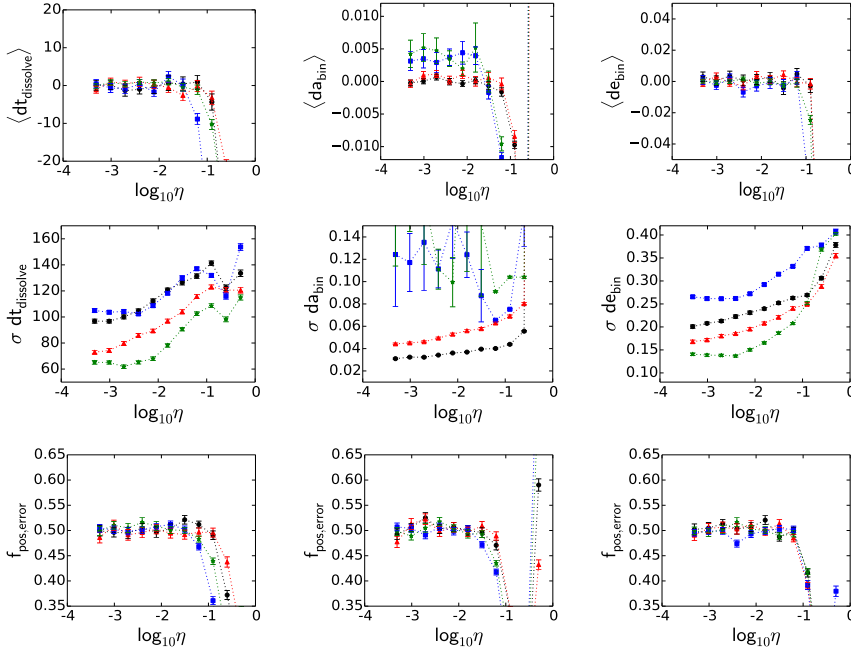
Figure 4.7: Statistics on the error distribution of Hermite results. We present the average error (top row), the standard deviation of the error distribution (middle row) and the fraction of errors which are positive (bottom row). The errors are given for the dissolution time (left column), binary semimajor axis (middle column) and eccentricity (right column). The different curves represent the different data sets similar as in Fig. 4.6.

The Error Distribution

In Fig. 4.7 we present statistics on the distribution of the errors, i.e. $S_{Hermite} - S_{Brutus}$, with S a statistic. For the dissolution time and the eccentricity, the average error converges to zero for $\eta < 10^{-1}$. For larger time-steps, simulations which grossly violate energy conservation ($|\Delta E/E| > 0.1$) cause biases in the average error. For the binary semimajor axis however, the data representing the cold collapse simulations also seem to be systematically biased for small time-steps, in the sense that Hermite makes fewer tight binaries.

The width of the error distributions converge to a non-zero value. This can be understood because with decreasing time-step, round-off errors will become more important so that the standard deviation of the errors will never reach zero. For the dissolution time, the width of the error distribution for the smallest time-step parameter adopted, varies from 60 to 100 crossing times. For the eccentricities the width

is on average ~ 0.2. For the semimajor axis the width approaches ~ 0.05 (in Hénon units). In the case of the semimajor axis, the data representing the cold collapse simulations behave differently, because the width is much larger than the width for the data representing the initially virialised systems.

If we regard the results given by Brutus and Hermite as random variables drawn from the same distribution, then we can write the variance in a certain statistic, in this example the eccentricity, as:

$$\langle (e_H - e_B)^2 \rangle = \langle e_H^2 \rangle + \langle e_B^2 \rangle - 2\langle e_H \rangle \langle e_B \rangle. \qquad (4.5)$$

Here e stands for eccentricity and the subscripts for Brutus and Hermite. For a thermal eccentricity distribution (Eq. 4.1), we obtain a standard deviation of $1/3$. However, this only applies to inaccurate Hermite results, which had enough time to diverge through outcome space. If we multiply the theoretical standard deviation calculated above by the inaccurate fraction, we obtain a range in the standard deviation from 0.17 to 0.27, as η ranges from the most precise value to $\eta = 10^{-1}$.

Symmetry of the Error Distribution

To measure the symmetry of the error distribution, we count the fraction of positive errors (Fig. 4.7, bottom panels). Again for an $\eta < 10^{-1}$, this fraction converges to 0.5. A more detailed comparison is given in Fig. 4.8, where we compare distribution functions of positive and negative errors. In Sec. 2.2.3, we mentioned that in our experiment we define the Brutus solution to be converged when at least 3 decimal places of every coordinate have converged. To investigate the symmetry up to higher precision, we repeated a subset of 1000 simulations. We did this only for the initial conditions with equal-mass stars picked randomly from a virialised Plummer distribution and this time we obtain solutions converged up to the first 15 decimal places.

We observe that the majority of errors are larger than $\sim 10^{-3}$ and within the statistical error, the positive and negative errors have a similar distribution. For the smallest errors however, we observe an asymmetry in the sense that there are more negative, small errors. The magnitude of the error where this excess occurs is determined by the precision of the integration. For the smallest η, the excess is below double-precision and thus not observable anymore (see Sec. 4.3.2 for more explanation).
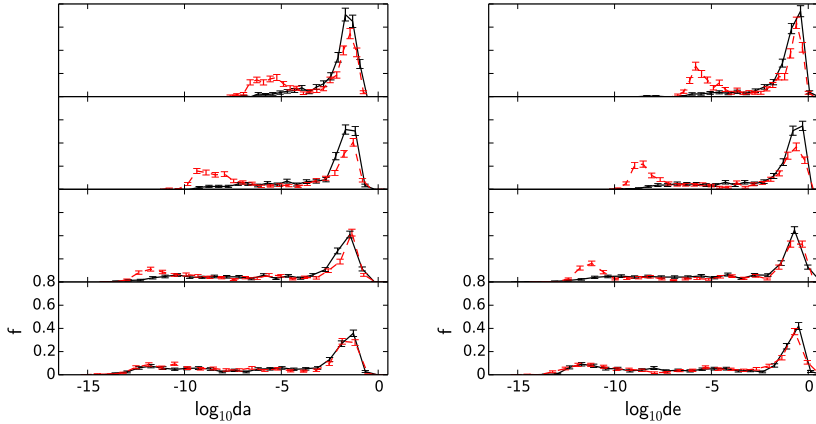
Figure 4.8: Symmetry of the error distributions. We show distributions of the errors in semimajor axis (left column) and eccentricity (right column) of the binaries formed in the equal-mass Plummer data set. This is shown separately for the positive errors (solid line) and negative errors (dashed line), to investigate the symmetry of the error distribution. From the panels at the top to the bottom, the time-step parameter for Hermite varies as $2^{-5}, 2^{-7}, 2^{-9}$ and 2^{-11}. An asymmetry can be observed at the smallest errors.

4.2.5   Escaper Identity

In this section we compare the solutions obtained with Hermite and Brutus individually, by looking at which star eventually becomes the escaper and which form the binary. We define preservation if the Hermite and the Brutus solution both have the same star as the escaper. We define it as exchange if the escaping star is different. A further distinction can be made in the preservation category, if the Hermite simulation is also accurate. We can typify each Hermite simulation as follows:

- **Accurate**: The coordinates are accurate, up to at least two digits.

- **Preservation**: The coordinates are inaccurate, but same star escapes.

- **Exchange**: Different star escapes.

In Fig. 4.9 we present the fraction of each category as a function of time. As expected, systems which dissolve quickly, hardly have time to develop errors and are categorized as accurate simulations. In time

Figure 4.9: The evolution of the relative fraction of categories. The differ-
ent curves represent the different categories: accurate (solid curves), preservation
(dashed curves) and exchange (dotted curves). These three categories are defined
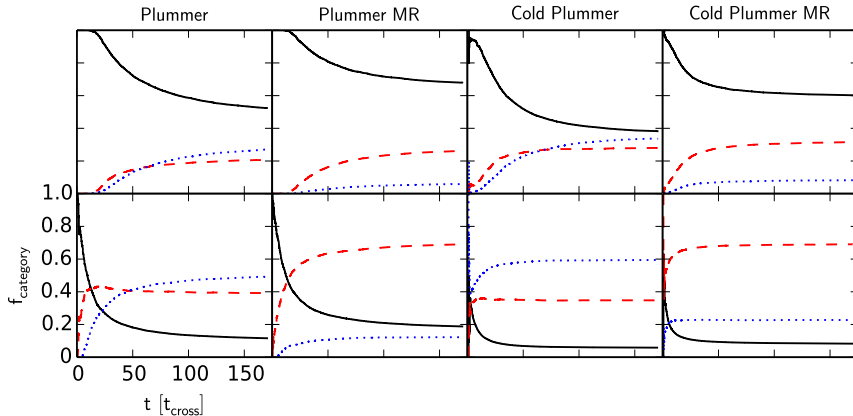in the text. From left to right, the data are from the Plummer, Plummer with
different masses, cold Plummer and cold Plummer with different masses data sets.
In the top panels we show the results for a Hermite time-step parameter $\eta = 2^{-11}$
and in the bottom for $\eta = 2^{-3}$.

however, because errors grow exponentially, the solutions become in-
accurate. The fractions of preservation and exchange start to grow.
For a small time-step parameter ($\eta = 2^{-11}$, top row in Fig. 4.9), this
growth starts after ~ 20 crossing times for the initially virialised sys-
tems. For the initially cold systems, the inaccurate fractions already
start to grow after a single crossing time.

The cold collapse with equal-mass stars is the hardest problem to
integrate as the accurate fraction is of comparable magnitude as the
preservation and exchange fractions. The accurate fraction generally
remains dominant, with a final fraction varying from about 0.4 for the
equal-mass cold Plummer to about 0.7 for the Plummer with different
masses. For the lesser precision ($\eta = 2^{-3}$, bottom row in the figure),
the accurate fractions decrease to below 0.2.

In the panels in Fig. 4.9, which include the data for the systems with
different masses, preservation is more common than exchange. This
can be understood, because due to energy equipartition, the lightest
particle will be more likely to escape and therefore the identity is more
often correct than in the equal mass case. For the equal mass case,
the fraction of preservation and exchange is comparable, except in the
case of the equal-mass cold Plummer with the low precision ($\eta = 2^{-3}$,
the bottom row). If we regard the identity of the escaping star to be
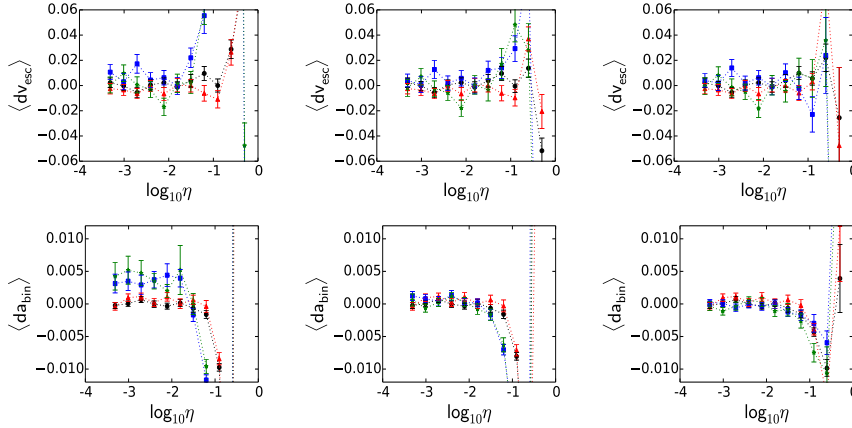
Figure 4.10: The effect of cuts in final relative energy conservation. We plot the average error in the velocity of the escaping star (top row) and the error in the binary semimajor axis (bottom row) as a function of Hermite time-step parameter η (with the same relation between the curves and the data sets as in Fig. 4.6). The three columns differ in the maximum allowed level of relative energy conservation. In the left column we show the results for the total ensemble of solutions, in the middle column for a maximum level of unity and in the right column for 10^{-1}. The bias in the left column for the binary semimajor axis is caused by solutions which grossly violate energy conservation. Note that this only happens for the cold collapse simulations. When these outliers are taken out of the ensemble, the bias vanishes.

completely random once the solution has become inaccurate, we would expect the fraction of exchange to be twice the fraction of preservation. This is roughly what we observe in the equal mass cold collapse case with low precision. Because of the low precision and the initial close encounter, solutions will diverge very quickly. In the panel with the higher precision this trend is not observed because the solutions are less randomised. The preservation category includes solutions which slightly differ from the converged solution only in the escape angle of the escaper. Also the long-lived triples are not taken into account here, which will alter these fractions.

4.3   DISCUSSION

4.3.1   Energy Conservation

In every ensemble of Hermite solutions there are some that grossly violate conservation of energy $|\Delta E/E| > 0.1$. This deformation of the energy hyper-surface in phase-space can allow solutions to reach

parts of phase-space which are theoretically forbidden. This affects the global statistical distributions. In Fig. 4.10, we replot the average error in the binary semimajor axis as a function of the time-step parameter. We produce similar diagrams as presented in Fig. 4.7, but this time we introduce a maximum allowed error in the energy. If we filter out simulations with a relative energy conservation $|\Delta E/E| > 1$, or $|\Delta E/E| > 0.1$, we observe that the bias in the average error of the semimajor axis of the binaries vanishes. We conclude that this bias is caused by a few simulations which grossly violate energy conservation. A similar bias in the velocity of the escaping star is less pronounced.

Time-reversible, symplectic integrators should in principle conserve energy to a better level than non-symplectic integrators, since there is no drift present in the energy error. Therefore, by using a symplectic integrator, the number of simulations with large energy error could be reduced. Using a Leapfrog integrator with constant time-steps, we tested this assumption and we find that for resonant 3-body interactions, it is challenging to obtain accurate solutions. The main reason is that, contrary to regular systems like, for example, the solar system, resonant 3-body interactions often include very close encounters, which need a very small time-step size to be resolved accurately. This is especially the case for the initially cold systems. Adopting such a small time-step size for the whole simulation, will increase the wall-clock time to that of Brutus or beyond.

4.3.2 Asymmetry at Small Errors

In Sec. 4.2.4, we discussed an asymmetry at small errors. In Fig. 4.11, we present similar diagrams as in Fig. 4.8 for the positive and negative errors. This time we add the errors in the total energy and angular momentum of the system and the error in the velocity of the escaper.

We also vary the integration method because different methods produce different (biased) error distributions in energy and angular momentum. We use a standard Leapfrog integrator, a standard Hermite integrator and a Hermite integrator which uses the $P(EC)^n$ method (we adopted n=3) (**?**). This last method adds an iterative procedure to the algorithm to improve the predictions and corrections, which improves the time-symmetry. For each method we implement a shared, adaptive time-step criterion as in Eq. 2.1, with a time-step parameter $\eta = 2^{-7}$. As a consequence they will not be time-symmetric nor symplectic.

We first look at the error distributions in the total energy and angular momentum. We observe that none of them are symmetric, in the
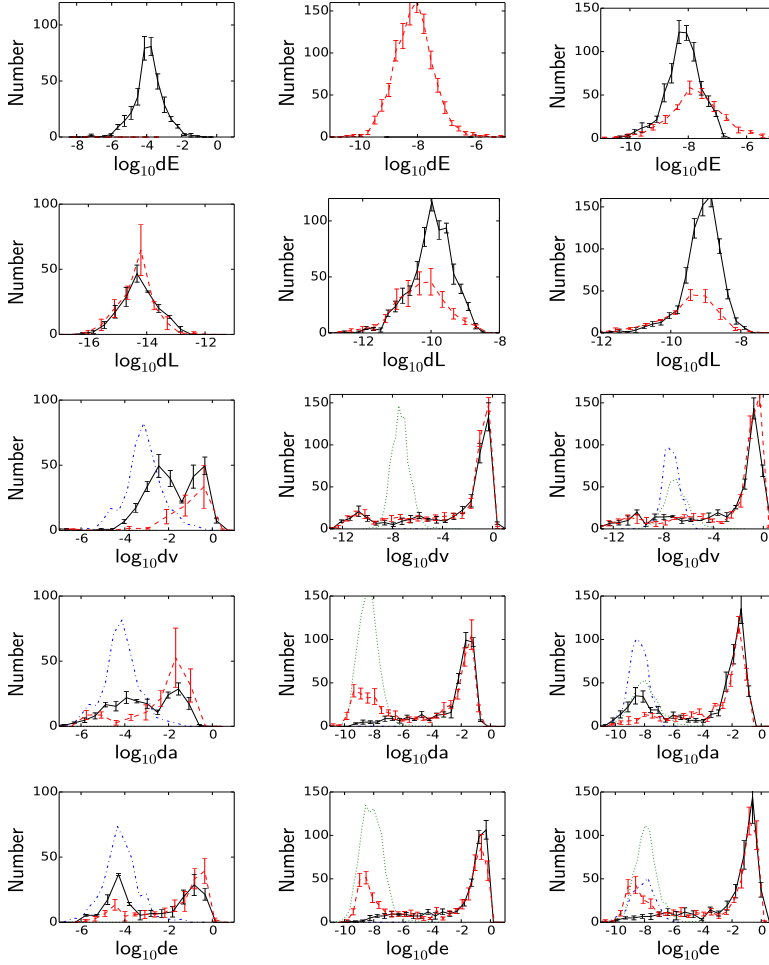
Figure 4.11: Explanation of the asymmetry at small errors. We show distributions of the positive (solid curves) and negative (dashed curves) errors in the total energy (top row), total angular momentum (second row), escaper velocity (third row), binary semimajor axis (fourth row) and eccentricity (bottom row). This is shown for different algorithms: Leapfrog (left column), standard Hermite (middle column) and Hermite with $P(EC)^n$ method (right column, $n = 3$). Each method implements a shared, adaptive time-step criterion according to Eq. 2.1, with a time-step parameter $\eta = 2^{-7}$. Each of these three integrators has a different asymmetry in the conservation of energy and angular momentum. By propagating these asymmetric errors as a small perturbation to the converged solution, we can estimate the resulting asymmetry in the derived quantities. These estimated error distributions are also given separately for the positive (dot-dash, light curves) and negative (dotted, light curves) errors. We observe that the estimated error distributions are located at the asymmetry in the empirical error distributions. The asymmetry at small errors is caused by a bias in the integrator.

sense that the positive and negative errors have identical distributions, except for the angular momentum in the Leapfrog simulations. The Leapfrog solutions tend to gain energy, whereas the standard Hermite loses energy. The Hermite with the $P(EC)^n$ method produces both positive and negative errors in the energy, but not in a symmetric manner.

To investigate whether the bias in energy and angular momentum conservation propagates to a bias in the binary and escaper properties, we estimate what the errors should be if we regard the error in the energy and angular momentum as a small perturbation to the converged solution. For the error in the velocity of the escaper, using the derivative of the kinetic energy with respect to velocity, we obtain the following expression:

$$\delta v = \frac{1}{mv} \delta E.$$

(4.6)

Here m is the mass of a star, v the velocity as obtained by Brutus, δE the energy error and δv the error in the velocity due to this energy error. For the binary semimajor axis we obtain:

$$\delta a = \frac{2}{m^2} a^2 \delta E.$$

(4.7)

Here a is the semimajor axis from the Brutus solution. For the eccentricity we obtain:

$$\delta e = \frac{1}{\sqrt{1 + \frac{2\epsilon l^2}{\mu^2}}} \left(\frac{l^2}{\mu^2} \delta\epsilon + \frac{2\epsilon l}{\mu^2} \delta l \right).$$

(4.8)

Here μ is the total mass of the binary, ϵ and l the specific energy and specific angular momentum of the binary as obtained by Brutus. The error in the eccentricity δe has contributions from errors in the energy $\delta\epsilon$ and angular momentum δl.

If we compare the resulting error distributions to the actual error distributions, we find that the approximated error distribution is positioned at the asymmetry in the empirical error distribution. This is most clearly seen for the semimajor axis and eccentricity (see Fig. 4.11).

The reason why the approximated error distribution overestimates the excess, is because not all errors are solely due to an error in the energy and angular momentum. In time, the numerical solution diverges from the true solution and this error due to divergence will become

more dominant. With this in mind, we can approximate the error in a statistic as follows:

$$\delta S = \delta S_{\text{conservation}} + \delta S_{\text{divergence}}. \tag{4.9}$$

Here S is a statistic that is related to energy and/or angular momentum, $\delta S_{\text{conservation}}$ is the error due to a small perturbation in the energy and/or angular momentum and $\delta S_{\text{divergence}}$ is the error due to divergence of the solution. When the solution has not diverged appreciably yet, the first type of error will dominate and possible biases can be observed. When the second type of error dominates, we observe that the symmetry is restored to within the statistical error.

Upon inspection of the velocity data, we observe no asymmetry in the Hermite results. When we measure which fraction of the energy error is reserved for the binary and which fraction for the escaper, we find that in most cases the error propagates to the binary. For the Leapfrog however, the asymmetry is still present.

4.3.3   Preservation of the Macroscopic Properties

Valtonen et al. (**?**) state that the final statistical distributions forget the specific initial conditions and only depend on globally conserved quantities. This assumption makes predictions which are verified by our experiment. The results show that for a time-step parameter $\eta < 2^{-5}$, the distributions are statistically indistinguishable, even though at least half of the solutions diverged from the converged solution. If however, energy conservation is grossly violated, biases are introduced in the statistics. In our experiment, a maximum level of relative energy conservation of $|\Delta E/E| = 0.1$ was sufficient to remove the biases. This is a much milder constraint than the $|\Delta E/E| \sim 10^{-6}$ usually adopted in collisional simulations. Whether 0.1 is also sufficient for systems with more stars, should be verified experimentally. Heggie (**?**) for example, finds that the energy of escaping stars in higher-N systems, depends sensitively on integration accuracy. The maximum required level of energy conservation should be such that it is below the energy taken away from the cluster by the escaping stars.

The chaoticity of the 3-body problem is illustrated by the scatter diagrams in Fig. 4.5. For a certain value of a statistic obtained by Brutus, any other value in the allowed outcome space is reachable for the Hermite integrator. For example, if the converged solution gives an eccentricity for the binary of 0.6, a diverged solution can produce any eccentricity between 0 and 1. Once the solution has diverged from the true solution, it will start a random walk through or near the allowed

phase-space until the 3-body system has dissolved. We observed that this randomisation happens in such a way that the available outcome space is still completely sampled and that it preserves global statistical distributions.

In Sec. 4.2.3, we discussed that the lifetime of an unstable triple does not depend on the integrator used nor on the accuracy of that integrator. This last point should be interpreted in the sense that when more effort is put into performing simulations with higher precision, that this does not change the global statistics, even though individual solutions will change with precision (see for example the Hermite results in Fig. 2.1). If instead we continue to decrease the precision, there will be a point where biases start to appear. Urminsky (**?**) analysed the 3-body Sitnikov problem and showed that the precision of the integration influences the average lifetime of triple systems, contrary to our results. The integration times in our experiment however, are much shorter. Obtaining a converged solution for a resonant 3-body system for longer than 200 crossing times, is still computationally challenging. Therefore any statistical difference on the long term will not be visible in our experiment.

4.4 CONCLUSION

Brutus is an N-body code that uses the Bulirsch–Stoer method to control discretisation errors, and arbitrary-precision arithmetic to control round-off errors. By using the method of convergence, where we systematically vary the Bulirsch–Stoer tolerance parameter and the word-length, we can obtain a solution for a particular N-body problem, for which the first p digits in the mantissa are independent of the time-step size and word-length. We call this solution converged to p decimal places.

Obtaining the converged solution is computationally very expensive, mainly because of the exponential divergence of the solution. In some cases, Bulirsch–Stoer tolerances of 10^{-100} are needed to reach convergence. We estimate that the time for simulating a star cluster up to core collapse, until convergence, scales approximately exponentially with the number of stars. Simulations with 256 stars however, may be performed within a year of computing time.

The motivation to obtain expensive, converged solutions is to test the assumption that the statistics of an ensemble of approximate solutions, are indistinguishable from the statistics of an ensemble of true solutions. To put this assumption to the test, we have investigated the statistics on the breakup of 3-body systems. In our experiment, a

bound triple system will eventually dissolve into a binary and an es-
caping star. Solutions to every initial realisation were obtained using
the standard Hermite integrator and using Brutus.

For systems with a long lifetime it is challenging to obtain the con-
verged solution. Due to repeated ejections and resonances, many ac-
curate digits will be lost and so a very small Bulirsch–Stoer tolerance
is required. Therefore, we have set an integration limit at ~ 180 cross-
ing times. For equal-mass, virialised systems, $\sim 40\%$ of the random
initial realisations were not dissolved by this time. For the initially
cold systems with different masses this was $\sim 10\%$. Hermite and Bru-
tus are consistent on the average lifetime of an unstable triple system.
However, possible differences on the long term are not visible in this
experiment.

When we compare the results on an individual basis, we find that on
average about half of the Hermite solutions give accurate results, i.e. at
most a 1% relative difference compared to Brutus. For the inaccurate
results, the error distribution becomes unbiased and symmetric for a
time-step parameter $\eta \leq 2^{-5}$ and implementing a maximum level of
relative energy conservation of $|\Delta E/E| < 0.1$.

Once the conventional solution has diverged from the converged so-
lution, it will start a random walk through or near the allowed region
in phase space. such that any allowed outcome of a statistic is reach-
able. This randomisation process completely samples the available
outcome space of a statistic and it also preserves the global statistical
distributions.

Kolmogorov–Smirnov tests were performed to compare the global
distributions produced by Hermite and Brutus. No significant differ-
ences were detected when using the criteria mentioned above for the
time-step parameter η and relative energy conservation. This research
for the 3-body problem supports the assumption that results from
conventional N-body simulations are valid in a statistical sense. We
observed however that a bias is introduced for the smallest errors, if
the algorithm used to solve the equations of motion, is biased in the
conservation of energy and angular momentum. In this research how-
ever, this bias did not have an appreciable effect. It is important to
see whether this remains true for statistics of higher-N systems or sys-
tems with a dominant mass. An example of a higher-N system where
precision might play a role is a young star cluster (without gas) going
through the process of cold collapse (**?**). At the moment of deepest
collapse, a fraction of stars will obtain large accelerations, so that a
small error in the acceleration can cause large errors in the position

and velocity. The rate of divergence can increase up to about 5 digits per Hénon time unit for 128 particles and it increases with N.

5

The Quasi-ergodic Property
of Gravity also known as
"Nagh-Hoch"

The conservation of energy, linear momentum, and angular momentum are important drivers of our physical understanding of the evolution of the universe. These quantities are also conserved in Newton's laws of motion under gravity. Numerical integration of the associated equations of motion is extremely challenging, in particular due to the steady growth of numerical errors (by round-off and discrete time-stepping and the exponential divergence between two nearby solutions. As a result, numerical solutions to the general N-body problem are intrinsically questionable. Using brute force integrations to arbitrary numerical precision we demonstrate empirically that ensembles of different realizations of resonant three-body interactions produce statistically indistinguishable results. Although individual solutions using common integration methods are notoriously unreliable, we conjecture that an ensemble of approximate three-body solutions accurately represents an ensemble of true solutions, so long as the energy during integration is conserved to better than 1/10. We therefore provide an independent confirmation that previous work on self-gravitating systems can actually be trusted, irrespective of the intrinsically chaotic nature of the N-body problem.

5.1   INTRODUCTION

Newton's law of gravitation is one of the fundamental laws in the universe that holds everything together. Although formulated in the

17th century, scientists today still study the consequences, in particular those of many-body systems, like the solar system, star clusters, and the Milky Way. General analytic solutions to the N-body problem only exist for configurations with one mass, commonly referred to as $N = 1$ solutions, and for two masses (equivalently named $N = 2$; Kepler (1609); Newton (1687)). Problems for $N \to \infty$ can be reduced via Liouville's theorem for Hamiltonian systems to the collisionless Boltzmann equation (**??**, but see also **?**), and therefore analytic solutions for the global distribution function exist.

Solutions for N in between these two limits are generally realized by computer simulations. These so-called N-body simulations have a major shortcoming in that the solution to any initial realization can only be approximated. The main limiting factors in numerically obtaining a true solution include errors due to round-off and approximations both in the integration and in the time-step strategy (**????**). These generally small errors are magnified by the exponentially sensitive dependence on the 6N-dimensional phase-space coordinates, position and velocity (**??**). As a consequence, the solution for a numerically integrated self-gravitating system of N masses diverges from the true solution (**??**). This error can be controlled to some degree by selecting a phase-space volume-preserving or a symplectic algorithm (**?**) and by reducing the integration time step (**??**). The latter however, cannot be reduced indefinitely due to the accumulation of numerical round-off in the mantissa, which is generally limited to 53 bits (64 bits in total, but 11 bits are reserved for the exponent, resulting in only about 15 significant digits). The exponential divergence subsequently causes this small error to propagate to the entire system on a dynamical time-scale (**?**), which is the time-scale for a particle to cross the system once. The result of these errors, together with the exponential divergence, is the loss of predicting power for a numerical solution to a self-gravitating system with $N > 2$ after a dynamical time-scale. One can subsequently question the predicting qualities of N-body simulations for self-gravitating systems, and thereby their usefulness as a scientific instrument.

We address this question for $N = 3$ using brute-force numerical integration to arbitrary precision. The choice of $N = 3$ is motivated by the realization that this represents the first fundamental irregular configuration with the smallest possible number of objects that cannot be solved analytically and cannot be addressed with collisionless theory. In addition, three-body encounters form a fundamental and frequently occurring topology in any large N-body simulation, and therefore also drive the global dynamics of these larger systems.

5.2 VALIDATION OF THE UNRESTRICTED PRECISION INTEGRATION

The divergence between two different, approximate solutions to the N-body problem can be quantified by the phase-space distance in the positions r and velocities v of the N particles (in dimensionless N-body units):

$$\delta^2 = \frac{1}{6N} \sum_{i=1}^{N} \left[(\mathbf{r}_A - \mathbf{r}_B)^2 + (\mathbf{v}_A - \mathbf{v}_B)^2 \right]. \qquad (5.1)$$

Values of δ are obtained by comparing the configurations from solution A and solution B at any moment in time. Each star has a position and velocity in solution A and (generally) a different position and velocity in solution B. For each star we calculate its phase-space distance between the two solutions. By dividing by $6N$, δ can be thought of as the average difference per coordinate. The two different runs can be performed either with the same code at a different precision, or with two different codes, all having exactly the same initial realization. A value of $\delta \gtrsim 0.1$ indicates that the results of the two simulations have diverged beyond recognition. We consider a solution to be converged to p decimal places when, for any time $t > 0$, $\delta < 10^{-p}$. (In stable hierarchical few-body systems the value of δ can vary substantially across the orbital phase (**?**), and one has to be assured that temporarily large deviations can diminish again at a later instant.)

To investigate the build-up of numerical errors and the corresponding exponential divergence, we developed an N-body solver for self-gravitating systems which solves the N-body problem to arbitrary precision. This code, named Brutus (**?**), is composed of a Bulirsch–Stoer integrator (**?**), which conserves energy to the level of the Bulirsch–Stoer tolerance. This tolerance is a parameter that can be interpreted as the discretisation error per integration step. The round-off error is controlled by choosing the word length with which all floating point numbers in the computer code are represented. By decreasing the Bulirsch–Stoer tolerance and increasing the word length, we can obtain solutions to the N-body problem to arbitrary precision.

We tested Brutus by adopting a three-body system of identical particles, which are located on the vertices of an equilateral triangle, with initial velocities such that the orbits are on a circle around the center of mass (Lagrange, 1772). Because this system is intrinsically unstable, small perturbations in the position and velocity vectors cause the triangular configuration to dissolve quickly. The time at which this

happens depends on precision. Using Brutus we can reach arbitrary precision, but in this validation experiment we stopped reducing the time step and increasing the word length once the energy was conserved up to 75 decimal places, which is sufficient to demonstrate our point. For any pre-determined time of stability there is a combination of word length and Bulirsch–Stoer tolerance for which Brutus converges. We define a solution to be converged when the first p decimal places become independent of the size of the time step and the word length. This is equivalent to saying that δ is always below 10^{-p}; for $p = 3$ (at least the first three digits have converged), then $\delta < 10^{-3}$ at all times.

5.3   RESULTS

Having established the possibility of integrating a self-gravitating N-body system to arbitrary precision we can study the reliability of N-body simulations in general. We limit ourselves to the problem of three bodies, generating a database of different three-body problems and solving them until a converged solution is achieved. The positions of the particles are taken randomly from a Plummer distribution (**?**) and are either cold (zero kinetic energy) or virialised. In the cold case we ensured that the mutual distances between the particles are initially comparable (within an order of magnitude). We performed runs with identical masses and with the masses in a ratio of 1:2:4. For each of the four selected ensembles of initial conditions we generated 10^4 random realizations. The masses and coordinates for these systems are specified in standard double-precision to ensure that the double-precision calculations use exactly the same initial realizations as the arbitrary-precision calculations. Every initial condition is integrated using the Leapfrog–Verlet (**?**, we adopted the implementation available from http://nbabel.org) and the fourth-order Hermite predictor-corrector scheme in a code called ph4 (**?**). (Both codes, Brutus and ph4, are assimilated in the public AMUSE framework which is available at http://amusecode.org; **?**). The integration continues until the system has been dissolved into a permanent binary and a single escaper (Heggie, 1975; Hut & Bahcall, 1983). Dissolution is declared upon the first integral dynamical time upon which one particle is unbound, outside a sphere of two initial virial radii around the barycentre, and receding from the center of mass (Hut & Bahcall, 1983). A particle is considered unbound if its kinetic energy in the center of mass reference frame exceeds the absolute value of its potential energy, which is stricter than adopted in Hut & Bahcall (1983). For a fraction of the
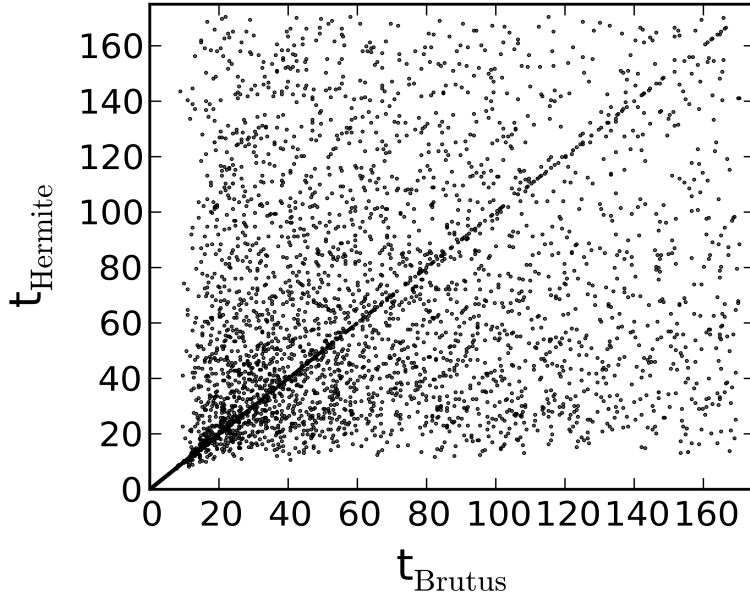
Figure 5.1: Individual comparison of the dissolution time of three-body systems. Each point represents one unique initial realization of three equal-mass bodies taken randomly from a Plummer distribution in virial equilibrium. The time to dissolution given by Hermite (using $\eta = 2^{-5}$) is on the ordinate and the converged value given by Brutus on the abscissa. About 50 percent of the data points lie on the diagonal which represents the cases for which Hermite and Brutus gave very similar results. The scatter around the diagonal is symmetric. For very short dissolution times (< 10 dynamical times), there is insufficient time to grow errors and the results are in agreement. Once the divergence becomes important the Hermite integrator can return any value allowed in the experiment irrespective of the converged dissolution time.

Figure 5.2: Cumulative distribution for the difference in time until dissolution for Hermite compared to converged Brutus solutions for three different values of $\eta = 2^{-2}$ (solid), $\eta = 2^{-3}$ (dashes) and $\eta = 2^{-9}$ (dotted curves). For each value of η, there are two curves, one for the case when Hermite lasted longer and the other for when Brutus lasted longer. Each of 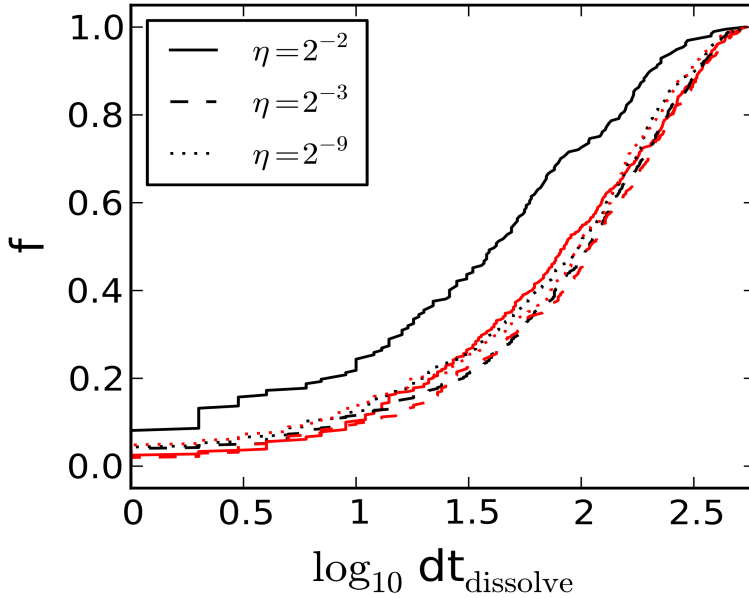these pair of curves for $\eta \leq 2^{-3}$ is statistically indistinguishable, and the mean difference is centred around the origin.

simulations (see Figure 5.3), the dissolution time turns out to be very long as the evolution consists of a sequence of ejections where a particle almost escapes, but then still returns to once again enter a three-body resonance. We therefore put a constraint on the integration time and use the fraction of long-lived systems as a measurable statistic. We obtain ensembles of solutions using the Hermite and Leapfrog integrators with a time-step parameter $\eta = 2^{-1}, 2^{-2}, ..., 2^{-11}$. Here we adopted the definition for η given by **?**.

We subsequently recalculate each of these initial realizations with Brutus using the same tolerance. In subsequent calculations we systematically reduce the time-step size and increase the word length until we obtain a converged solution (as we discussed in Section 5.2 for $p = 3$) for every realization of the initial conditions. This converged solution is then compared to the earlier simulations performed with the Hermite and Leapfrog integrators.

We now have three solutions for each initial realization of the three-body problem, one of which is the converged solution. We compare the three solutions for the time of dissolution, the semimajor axis (or equivalently the reciprocal of the orbital energy) of the surviving binary, its eccentricity (equivalent to the angular momentum), and the escaper's velocity and direction.

In Figure 5.1 we individually compare the time to dissolution for a certain initial realization as given by the Hermite integrator and the converged solution as given by Brutus. About half of the individual Hermite solutions lie along the diagonal representing the accurate solutions. The other half is scattered around the diagonal. These solutions have diverged away from the converged solution, producing a binary and an escaper with completely different properties. For dissolutions within ~ 10 dynamical times, there is insufficient time for the solution to diverge and the results of the various numerical methods are consistent. But once the Hermite or Leapfrog solutions have diverged away from the converged solution the entire parameter space of the numerical experiment is sampled. A similar statement holds when instead of comparing the dissolution time, we compare the properties of the binaries or the escapers.

In Figure 5.2 we present the cumulative distribution function of the difference between the time to dissolution of the Hermite and Brutus calculations: $dt_{dissolve} = t_{Hermite} - t_{Brutus}$ for three different values of $\eta = 2^{-2}$, $\eta = 2^{-3}$ and $\eta = 2^{-9}$. The differences for $\eta \leq 2^{-3}$ are symmetric around the origin with a dispersion of ~ 70 N-body time units, but for $\eta \geq 2^{-2}$ it is not symmetric. The distributions in the differences in semi-major axis, eccentricity, and the direction of the escaper (polar and azimuthal angles with respect to the binary plane) at the time we stop the experiment for $\eta = 2^{-3}$ down to $\eta = 2^{-11}$ are symmetric with respect to the origin. The global distributions are statistically indistinguishable using a Kolmogorov–Smirnov test. We empirically determine that for a value of the time-step parameter $\eta = 2^{-3}$ the majority of the ensemble conserves energy to better than $1/10$.

In Figure 5.3 we present the fraction of undissolved systems in time. The coloured symbols give the converged solutions, whereas the curves give the results obtained using the Hermite integrator. The two solutions for each ensemble of initial realizations for $\eta \leq 2^{-3}$ (as well as those obtained with the Leapfrog integrator, not shown) are statistically indistinguishable after comparing 10^4 realizations of the initial conditions. The distributions obtained using $\eta \geq 2^{-2}$ are not symmetric.
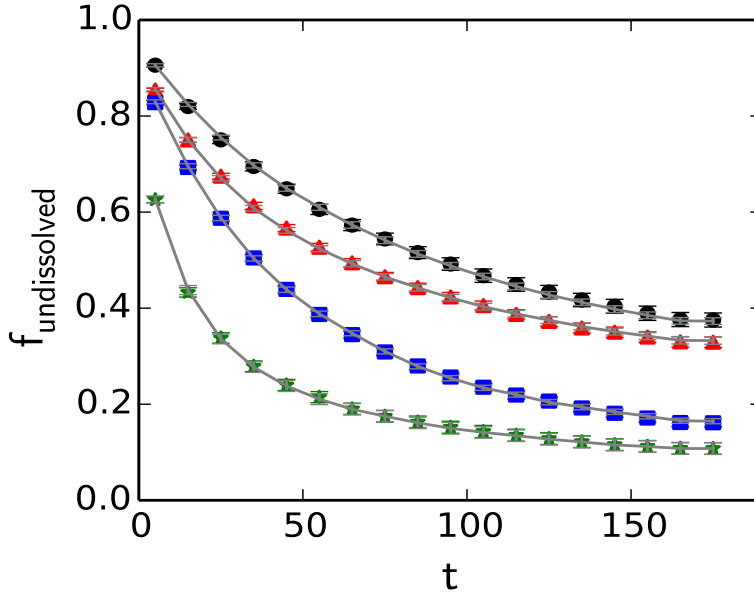
Figure 5.3: Fraction of surviving systems as a function of time for the four sets of initial conditions. The symbols give the results calculated with Brutus, the curves give the linear interpolation between the points calculated with the Hermite integrator using $\eta = 2^{-5}$. The virialised Plummer sphere with identical masses is represented by the bullets, and with the range in masses as triangles. The squares and stars give the results for the cold Plummer distribution without and with different masses, respectively. The results of the runs with Hermite are statistically indistinguishable from those with Brutus.

The duration of stability was studied as a function of accuracy by **?** using the Sitnikov problem (**?**). They found that the remaining time for the system to stay bound depends on the integration accuracy. Our simulations did not reveal this effect, because we study systems that dissolve on a much shorter time-scale.

5.4   CONCLUSION

The properties of the binary and the escaper of a three-body system can be described in a statistical way. This is consistent with the findings in previous analytic (**?**) and numerical (**?**) studies. This behaviour was named quasi-ergodicity by **?**. We confirm that this behaviour remains valid also for converged three-body solutions.

Based on the symmetry of the distribution in dissolution times (see Figure 5.2), the final parameters of the binary and escaper, as well as the consistency of the mean and median values of the inaccurate simulations when compared to the converged solution (see Figures 5.2 and 5.3) we argue that global statistical distributions are preserved irrespective of the precision of the calculation as long as energy is preserved to better than 1/10th of the initial energy of the system. Although we have tested only three algorithms for solving the equations of motion we conjecture that the statistical consistency may be also preserved for some other direct N^2 methods, and these may also require that energy and angular momentum are preserved to $\leq 1/10$th. If such direct N-body methods have the same statistical behaviour as collisionless ($N \gg 3$) systems, it will be interesting to investigate how other—non-N^2—algorithms, like the hierarchical-tree method (**?**) or particle-mesh methods (**?**) also behave in this respect.

In studies of self-gravitating systems which adopt the fourth-order Hermite integrator, energy and angular momentum are generally conserved up to $\leq 10^{-6}$ per dynamical time. Only those simulations in which this requirement is met are often considered reliable and suitable for scientific interpretation. Proof for this seemingly conservative choice has never been provided, and it is unknown whether or not the numerical error and the exponential divergence are not preventing certain parts of the parameter space to be accessed, or new physically inaccessible parts in the parameter space to be explored. We argued that for the resonant three-body problem the error made during the integration of the equations of motion poses no problem for obtaining scientifically meaningful results so long as energy is conserved to better than about one-tenth of the initial total energy of the system. In that case resonant three-body interactions should be treated as an ensemble average, and individual results only contribute statistically.

By means of numerical integration, until a converged solution is obtained, we find that the statistical properties of the binary and the escaper resulting from a three-body resonant encounter are deterministic. This behaviour is not guaranteed to propagate to larger N (see also **?**; $N > 3$ requires independent testing, because these introduce more complex solutions in the form of, for example, binary-binary outcomes and hierarchical triples. The more extended parameter space for increasing N from 3 to $N = 4$ is quite dramatic, in particular for solving the system until a converged solution is reached.

6

Event-driven Chaos
in Dynamical Systems

In 1207BC, a year after Odysseus returned to Ithaca, the comet P1/Halley should have been visible from the Ionian islands, but apparently was not sighted (**?**). Since then the short-period comet P1/Halley has intrigued astronomers, policy makers (**?**), religious leaders (**??**), artists and the general public for its splendour.

More recently, Halley gained considerable interest because of its importance for understanding the stability of the solar system. The comet probably migrated towards its current orbit in the last 200,000 years (**?**). Small variations in its time of sighting over the last millennium have prompted astronomers to the possible chaotic nature of the comet's orbit (**?**). In particular the recent discussion of the chaotic nature of its orbit, as derived by **?** requires us to revisit the chaotic nature of Halley's orbit, the origin of its chaos and its short Liapounov time.

We construct a general model for the growth of perturbations in a few-body dynamical system. Using a map to describe the time evolution of the orbital frequency of Halley's orbit, we find that a sequence of close encounters with Jupiter causes exponential growth with a Liapounov time of order 300 years. This short Liapounov time is a natural consequence of the density of close encounters and the strength of each encounter. Numerical integrations however, show that Venus is currently the dominant source of chaos in Halley's orbit.

6.1   INTRODUCTION

Whether a dynamical system, such as a planetary system, a star cluster or a galaxy, is stable or unstable is an important property of the system. For example, from an observational point of view, it tells us

about the likelihood to observe a system in a certain state. If the configuration is unstable, it will evolve to a different configuration on a relatively short time scale, so that statistically, it is unlikely to be observed. From a numerical perspective, the stability tells us about the time scale on which we can accurately predict the orbits of the constituent bodies. For a chaotic system, initially nearby solutions will quickly diverge away from each other, resulting in very different outcomes (**????**).

The stability of a configuration is determined by perturbing the configuration and calculating how the perturbation grows in time. If this growth follows a power law, the system is considered regular. Examples of regular dynamical systems are two-body (point-particle) binary stars, strictly hierarchical triples (e.g. **?**) and the three-body figure-8 orbit (**??**). If the growth instead is exponential, then the system is considered to be chaotic. Examples of chaotic three-body problems are the Pythagorean problem (**???**), the equilateral triangle (e.g. **?**) and the Sitnikov problem (e.g. **?**).

Chaos in larger systems was investigated by **?**. For time scales shorter than a million years, the divergence of the planets in our solar system closely resembles a power law. At later times however, it turns out that the divergence is really an exponential with an e-folding time of about 5 Myr (**?**). The time over which the stability has to be determined is thus an important factor.

It is also possible for a system to evolve from order to chaos and vice versa. In a different study of the solar system **?** reproduced the exponential divergence found by **?**. After ~ 50 Myr however, they observed a transition to a faster exponential growth (**?**, Fig. 1). The origin of this transition is not known. It might be physical and related to a different chaoticity for the terrestrial and Jovian planets. A numerical artefact is however not excluded.

A more intuitive example of a transition can be constructed for a three-body system consisting of a binary star and a single star that is relatively far but still bound, moving towards the binary star. Since at the start of the experiment the single and the binary star are well separated, the system will behave regularly. At small separations, however, the interplay of the three particles becomes much more irregular, resulting in a prolonged chaotic phase during the resonant encounter (**?**). In the resonant phase perturbations can grow by orders of magnitude (**?**).

Dynamical chaos is also present in star clusters, i.e. open clusters and globular clusters. **?** measured an exponential growth of perturbations for a small stellar system. **?** estimate the e-folding time for

this growth to be on the order of a crossing time, which is the typical time for a star to cross the cluster once. Due to this relatively short time scale, accurate long term integrations of star clusters are virtually impossible. The exponential divergence can be fought using high-precision integrations, but they are prohibitively expensive (**?**). Transitions in the rate of divergence also occur for star clusters. Starting with a cluster consisting of single stars, close encounters and interactions with a binary star during the moment of core collapse significantly increases the rate of divergence (**?**, Fig. 3).

Much work has been done to measure the stability of dynamical systems. Less work has been done on the origin of chaos in dynamical systems and transitions in the rate of divergence. **?** construct a model for the growth of perturbations in a star cluster. In a somewhat analogous way as in the derivation of the relaxation time of a stellar system (Chandrasekhar, 1942), they relate a linear growth of error to the linear growth in separation after a deflection due to a 2-body encounter (**?**, Fig 1). A sequence of 2-body encounters can result in the accumulation of power laws, which approximates exponential divergence. This is analogous to a feature already present in a hard-sphere gas (**?**).

We present a new model for the rate of divergence in few-body dynamical systems, which is based on 2-body Keplerian orbits being perturbed by a third body. Using the fact that a 2-body Keplerian system shows linear divergence and that the accumulation of power laws can produce exponential growth, we are able to model both regular, chaotic and transitional behaviour. We explain the model in more detail in Sec. 6.2.

Next we turn to the case of Halley's Comet (hereafter just Halley), which is perturbed by the planets. Its chaoticity has been verified in several studies (e.g. **????**). The e-folding time for the exponential divergence has been determined to be on the order of the orbital period of Halley or less (< 76 years). One of the aims of this study is to understand the origin of this short time scale. To this end, in Sec. 6.3 we define a map similar to those in **?** and **?**, which uses kick-functions to model the perturbations due to the planets on Halley. Using this map we investigate the onset of exponential growth of perturbations. In Sec. 6.4 we measure the rate of divergence between neighbouring solutions using precise N-body integrations of the orbit of Halley in the solar system. We compare the data to our semi-analytical model to gain a better understanding of the origin of chaos, the cause of its short e-folding time scale, i.e. the Liapounov time, and the physical mechanism responsible for transitions in the rate of divergence.
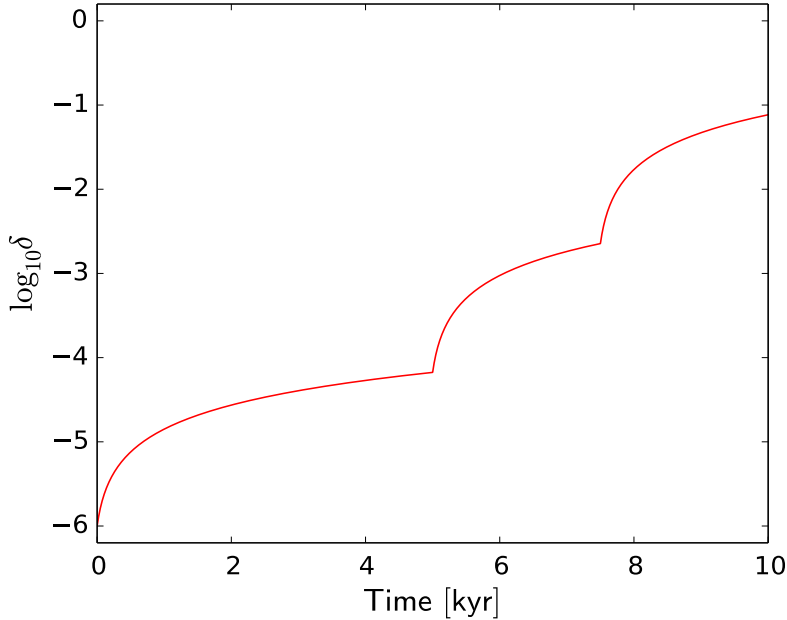
Figure 6.1: Growth of displacement between neighbouring solutions during two scattering events. The diagram, which is intended to be schematic, was plotted using eqs.(6.1), (6.3) and (6.4) for $\delta_0 = 10^{-6}$, a constant period $P = 76$ years, $f = 1/P$ and encounter times $t_1 = 5000$ and $t_2 = 7500$ years.

6.2   EVENT-DRIVEN CHAOS

Consider Kepler motion with initial semi-major axis a_0, total mass m and initial frequency $f_0 = \sqrt{m/a_0^3}$ (gravitational constant $G = 1$). Let a neighbouring solution be separated by a small displacement δ_0 initially. (For simplicity we also suppose that the difference in velocity is small.) This displacement has components along and transverse to the orbit, and we assume that they are both of equal magnitude. The cross-orbit component gives rise to a difference in semi-major axis of the same order, i.e. $\Delta a_0 \sim \delta_0$. The resulting difference in frequency is $\Delta f_0 \sim \delta_0 \sqrt{m/a_0^5}$. By time $t > 0$ the displacement along the orbit will have grown to an amount of order

$$\delta\left(t\right) \sim \delta_0 + a_0 \Delta f_0 t \sim \delta_0 \left(1 + f_0 t\right). \tag{6.1}$$

This growth is linear in t, but the growth in δ from t_0 to t leads to no growth in Δa, because the growth is along the direction of orbital motion.

Now suppose a short-lived significant perturbation acts on the motion at time t_1, and that the velocity of the Kepler motion changes direction significantly. The displacement δ_1 at that time now does have a significant component which is not along the new direction of orbital motion. Thus the variation in semi-major axis is now

$$\Delta a_1 \sim \delta_1, \tag{6.2}$$

and this leads to a difference in orbital frequency $\Delta f_1 \sim \Delta a_1 f_1/a_1$ at time t_1. Thus for $t > t_1$, the displacement varies as

$$\delta\left(t\right) \sim \delta_1 + \Delta a_1 f_1\left(t - t_1\right) \sim \delta_1\left(1 + f_1\left(t - t_1\right)\right). \tag{6.3}$$

This is again a linear growth, but with a different frequency and initial perturbation. If a second strong perturbation occurs at time $t_2 > t_1$, we can see from eqs.(6.1) and (6.3) that the displacement is

$$\delta_2 \sim \delta_0\left(1 + f_0 t_1\right)\left(1 + f_1\left(t_2 - t_1\right)\right), \tag{6.4}$$

with a subsequent growth of similar form as Eq. 6.3. A schematic plot of eqs.(6.1), (6.3) and (6.4) is given in Fig.6.1. The result qualitatively resembles the numerical result of **?**, Fig. 3. The main difference is that their numerical result has a regular oscillation superposed on the trend illustrated, because the motion in the numerical example is eccentric.

If the perturbations recur at roughly comparable intervals Δt, and if f does not change by a large factor, it can be seen that the displacement at some large time t will be

$$\delta\left(t\right) \sim \delta_0\left(1 + f\Delta t\right)^{t/\Delta t}. \tag{6.5}$$

In this way we see that the linear growth of Eq.(6.1) transforms into exponential growth, and can easily estimate that the corresponding Liapounov exponent is of order f if $f\Delta t \lesssim 1$. This means it is of order the reciprocal of the crossing time. The case $f\Delta t \gtrsim 1$ is also of interest, and leads to a smaller estimate of order $\ln(f\Delta t)/\Delta t$.

Up to a point we can think of a resonant three-body scattering event as a prolonged sequence of perturbations of Kepler motion. As long as the three bodies remain at comparable distances and are of comparable mass the perturbations in any of the three two-body motions will be of order 1 and will take place at intervals of order the crossing time. Therefore, in accordance with the above discussion, the Liapounov exponent will be of order $1/t_{cr}$, with t_{cr} the crossing time.

Indeed the numerical examples of **?** show that the separation of neighbouring solutions grows roughly exponentially until dissolution of

the resonance. The lifetime of the Pythagorean problem, for example, is about 16 crossing times (Aarseth, 2003, p. 238), and the growth of the separation of neighbouring solutions in this time is about 8.5 dex (**?**). Thus the finite-time Liapounov exponent is of order $1/t_{cr}$.

If the evolution of the triple system is dominated by protracted excursions of the third body, of order $T \gg t_{cr}$, then the estimate will decrease to one of order $1/T$ (in accordance with the result for the case $f\Delta t \gtrsim 1$, and neglecting a logarithm). Usually, the evolution is a mix of prolonged excursions interspersed with periods of frequent interplay (Szebehely, 1972), and the Liapounov exponent, λ, will be intermediate between limits $1/T \lesssim \lambda \lesssim 1/t_{cr}$, where T is the duration of the longest excursion.

The model we have used neglects the fact that there are, even in the two-dimensional problem we have discussed, four components of the deviation to take into consideration, i.e. two in configuration space and two in velocity space. But the only one of these which can grow secularly (between perturbations) is the component of δ along the orbit, and its growth is accounted for approximately in our model.

The result of the model (that the Liapounov exponent λ is of order $1/t_{cr}$ for comparable masses) is consistent with the results in **?**, who considered the general N-body problem. This is rather independent confirmation, as their model was based on assuming that the deviation between neighbouring solutions grows as a result of two-body encounters.

6.2.1   Generalization

The secular growth of a perturbation in a two-body system is proportional to the difference in orbital frequency, Δf, between the two neighbouring solutions. Due to events such as close encounters (or other events such as moments of significant mass loss (**?**)), the difference in orbital frequency will generally be a function of time, $\Delta f(t)$. Every time Δf changes, the subsequent growth of the perturbation is linear, proportional to the new value of Δf. In general we can write

$$\delta(t) \sim \delta_0 + \int_0^t \Delta f(T)\, dT. \tag{6.6}$$

If $\Delta f(t)$ is a constant, Δf, we obtain Eq. (6.1) for linear growth of the perturbation. Other types of behaviour are also possible. For example, if Δf keeps flipping sign, e.g. $\Delta f(t) \sim \sin(t)$, the integral will be zero on average and we obtain no secular growth of perturbations.

Exponential growth of perturbations is obtained if $\Delta f(t)$ grows exponentially. As described in Sec. 6.2, this can be obtained if there is a regular sequence of close encounters, e.g. changes in Δf. We can estimate a finite-time Liapounov exponent by equating Eq. (6.6) to an exponential with Liapounov exponent λ, resulting in

$$\lambda(t) = \frac{1}{t} \ln \left(1 + \frac{C}{\delta_0} \int_0^t \Delta f(T)\, dT \right). \tag{6.7}$$

Here C is a constant depending on the configuration and this approximate expression will approach the Liapounov exponent at large t.

6.3   THE ONSET OF EXPONENTIAL DIVERGENCE

In the previous section we have shown that an event, such as a close encounter with a third body, can cause the rate of divergence between two neighbouring solutions to increase. The growth of an initially small perturbation is thus related to the encounter history of the binary with a third body. In this section we measure the growth of perturbations using a map similar to those of **?** and **?**, with a kick function to model the effect of encounters. The model is particularly adapted to a case like that of Comet Halley, in which the masses of the perturbers are small, unlike the case of comparable masses considered in Sec.2.

6.3.1   Map for Changes in Orbital Frequency

We are interested in the time evolution of the difference in orbital frequency, Δf, since this quantity drives the growth of perturbations. We consider a system similar to the three-body system consisting of the sun, Jupiter and Comet Halley. Each orbital period Halley will encounter Jupiter at a certain distance, R, depending on the orbital phase of Jupiter, ϕ. We define the kick function $K(\phi)$ to be the 2π-periodic function that gives the change in orbital frequency δf as a function of ϕ. Depending on the geometry of the configuration, different kick functions are possible. A sawtooth-like function is appropriate for the configuration under discussion (**??**).

The map is given by

$$\phi_{n+1} = \phi_n + 2\pi \left(\frac{f_J}{f_n} \right) \tag{6.8}$$

$$f_{n+1} = f_n + K(\phi_{n+1}), \tag{6.9}$$
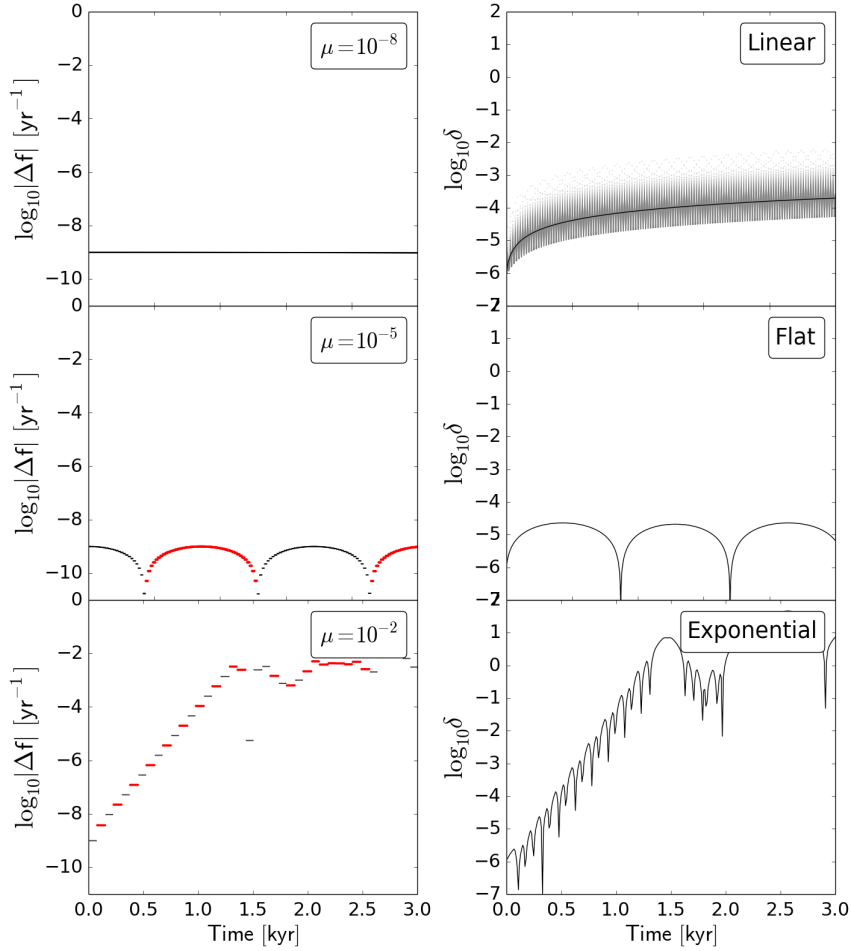
Figure 6.2: Illustration of the three types of behaviour for the growth of perturbations. The time evolution of the difference in orbital frequency Δf between two neighbouring solutions (left column) and the consequent growth or perturbation (right column), are presented for three different encounter strengths $\mu = 10^{-8}$, 10^{-5} and 10^{-2}. In the left column, the data is represented by a thin line if $\Delta f > 0$ and fat otherwise.

where ϕ_n is the phase (i.e. longitude) of Jupiter at the nth perihelion passage, f_n is the frequency of Halley after the nth perihelion passage, and f_J is the (constant) frequency of Jupiter. The times can be obtained recursively from

$$t_{n+1} = t_n + \frac{1}{f_n}. \tag{6.10}$$

Time is measured in years, f in yr^{-1} and semi-major axis, when we need it, in AU. The orbital periods of Halley and Jupiter are given by $P_h \simeq 75.3$ yr and $P_J \simeq 11.9$ yr respectively, and we use these values to compute the exact starting value $f_0 = 1/P_h$, and the value of f_J. Note that they are approximately in a 3:19 resonance.

To study the separation of neighbouring solutions we construct the tangent map, i.e. the linearisation of the above map, given by

$$\Delta\phi_{n+1} = \Delta\phi_n - 2\pi \frac{f_J}{f_n^2} \Delta f_n \tag{6.11}$$

$$\Delta f_{n+1} = \Delta f_n + \Delta\phi_{n+1} K'(\phi_{n+1}). \tag{6.12}$$

We see from eqs.(6.11), (6.12) that the change in $\Delta\phi$ is a negative multiple of Δf, whereas if $K' > 0$ the change in Δf is a positive multiple of $\Delta\phi$. This interplay causes interesting behaviour as we will illustrate in Sec.6.3.2 below.

When the right side of Eq.(6.12) is expressed in terms of $\Delta\phi_n$ and Δf_n, it takes the form

$$\Delta f_{n+1} = \Delta f_n + \left(\Delta\phi_n - 2\pi \frac{f_J}{f_n^2} \Delta f_n \right) K'(\phi_{n+1}). \tag{6.13}$$

Combining with Eq.(6.11), we see that the matrix of the linearised map is given by

$$A = \begin{pmatrix} 1 & -2\pi \dfrac{f_J}{f_n^2} \\ K'(\phi_{n+1}) & 1 - 2\pi \dfrac{f_J}{f_n^2} K'(\phi_{n+1}) \end{pmatrix}. \tag{6.14}$$

This matrix has determinant one, showing that our map is *symplectic* (i.e. area-preserving). Thus although the variables f, ϕ are not canonical in the usual sense (energy and phase would be better), the map preserves the main geometrical constraint of a canonical mapping. The eigenvalues of A, which will be useful below, are

$$\lambda = 1 - \pi \frac{f_J}{f_n^2} K' \pm \sqrt{\pi \frac{f_J}{f_n^2} K' \left(\pi \frac{f_J}{f_n^2} K' - 2 \right)}, \tag{6.15}$$
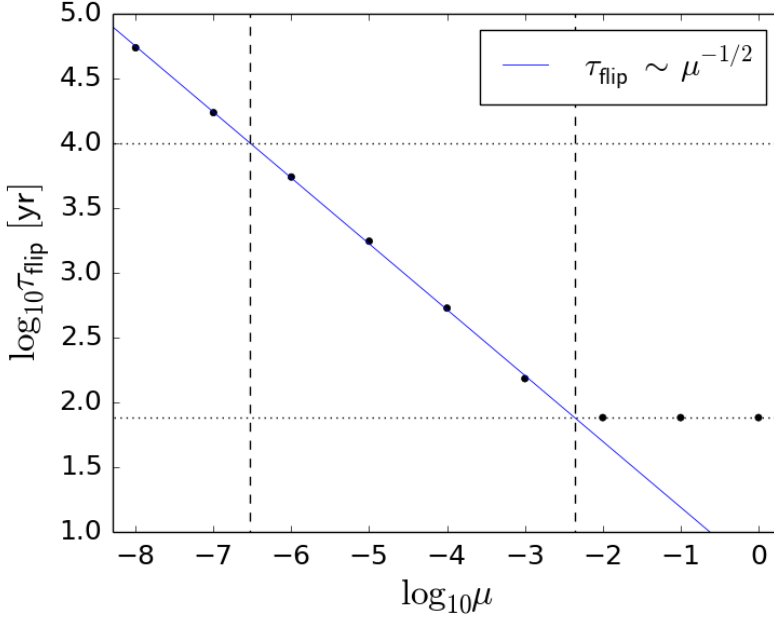
Figure 6.3: The time until the first change of sign, τ_{flip}, as a function of the strength of the perturbation, μ. The top dotted line marks the 10 kyr time of interest, whereas the bottom dotted line marks the orbital period of Halley. Exponential divergence occurs when τ_{flip} is smaller than the orbital period.

where $K' = K'(\phi_{n+1})$.

For numerical purposes we sometimes measure the difference between two solutions by a formula adapted from Eq.(6.6), whose right side can be computed from

$$\delta_{n+1} = \delta_n + C\Delta f_n \frac{1}{f_n}. \tag{6.16}$$

Another technicality of these numerical calculations is that ϕ_n and $\Delta\phi_n$ are stored mod 2π.

6.3.2   Linear Kick Function

We first consider the idealised case in which the kick function is the 2π-periodic function defined by the relation

$$K(\phi) = \frac{\mu}{2\pi}\phi \quad \text{on } [0, 2\pi) \tag{6.17}$$

with μ a free parameter. Note that the derivative of the kick function is a constant (except at the discontinuity when ϕ is an integral multiple of 2π). In this model every kick has the same contribution to the growth of perturbation in the orbital frequency. This is approximately the case when there is a regular sequence of similar encounters, i.e. ϕ_n is almost independent of n, mod 2π.

We start with the initial values $t = 0$, $\phi = 0$, $\Delta\phi = 0$ and $\Delta f = 10^{-9}$ (which corresponds to a $\Delta a \simeq 10^{-6}$). We vary the value of $\mu = 10^{-8}, 10^{-5}$ and 10^{-2} to model encounters of different strengths. In Fig. 6.2 we plot the time evolution of Δf (top row) and the consequent growth of perturbation (bottom row) through Eqs. (6.8)–(6.16).

For the weakest perturbations no variation in Δf is detectable on the scale of the plot, thus we get a linear growth of the perturbation. The data points which are also plotted in this panel (bottom left) are data from a numerical simulation where we integrated the orbit of Halley around the sun without perturbers. A good fit is obtained for a proportionality constant $C = 20$ (see Eq. 6.7). Although this value has no particular significance, it gives an impression of how the simple model we are considering may apply to Halley, and we use it in the remainder of this study.

If we make the perturbations somewhat stronger (Fig. 6.2 central panels) we observe an oscillatory behaviour such that the perturbation in Δf never exceeds its initial value but alternates in sign. To understand this oscillatory behaviour better, we compute the eigenvalues of the matrix A. From eqs.(6.15) and (6.17) we readily find that

$$\lambda = 1 - \frac{\mu f_J}{2 f_n^2} \pm \sqrt{\frac{\mu f_J}{2 f_n^2}\left(\frac{\mu f_J}{2 f_n^2} - 2\right)} \qquad (6.18)$$

$$\simeq 1 \pm i\sqrt{\frac{\mu f_J}{f_n^2}} \qquad (6.19)$$

when $|\mu| \ll 1$. This shows that the evolution is expected to be oscillatory (if $\mu > 0$), and the period (in years) is given approximately by

$$P = \frac{2\pi}{\sqrt{\mu f_J}}, \qquad (6.20)$$

which gives a value of 6854 yr for $\mu = 1 \times 10^{-5}$, i.e. very consistent with what would be inferred from Fig.6.2 (middle column).

For the strongest perturbation (Fig. 6.2 right panels) we observe exponential growth of perturbations. This growth saturates at $\log_{10} \delta \sim$

0 since the growth is limited by $\Delta\phi < 2\pi$. In other words, the perturbation has grown to the size of the system. We observe again the sign flipping of Δf, however this time it occurs after every orbital period. The exponential growth is explained again by Eq.(6.18), which shows that the eigenvalues λ are real if

$$\mu > \frac{4f_n^2}{f_J} = \frac{4P_J}{P_h^2} \simeq 0.0084. \qquad (6.21)$$

Also, their product is unity, and so one eigenvalue has magnitude $|\lambda| > 1$, resulting in exponential growth.

To understand the relation between the period of the oscillatory behaviour and the secular growth of perturbations in more detail, we plot in Fig. 6.3 the time to the first change of sign. (Note that this will be $0.25P$, in the notation of Eq.(6.21), as it is evident from Fig.6.2, central column, that the oscillation starts at maximum.) First, we observe that towards increasing values of μ, the flip time scale decreases as $\tau_{flip} \sim \mu^{-1/2}$, as predicted in Eq.(6.20). To the left of the vertical, dashed line at $\log_{10}\mu \sim -6.5$, the period is longer than 10 kyr, so that on such a time scale we observe a constant Δf and a linear growth of perturbation. In between the two vertical dashed lines we observe oscillatory behaviour with decreasing period. A transition in the behaviour occurs once the period is of the order the orbital period of Halley (marked by vertical dashed line at $\log_{10}\mu \simeq -2.4$). The critical value of $\mu \simeq 0.004$ and for larger values we obtain exponential growth of perturbations. The theoretical prediction that exponential growth occurs for $\mu \gtrsim 0.0084$, i.e. $\log_{10}\mu \gtrsim -2.08$, is consistent with the numerical data plotted in Figs. 6.2 and 6.3.

6.3.3   Saw-tooth Kick Function

In a more realistic kick function there will be both weak and strong encounters present. We use the following derivative of the kick function

$$\frac{dK(\phi)}{d\phi} = \frac{\mu_{max}}{2\pi}, \phi < \phi_c, \qquad (6.22)$$

$$\frac{dK(\phi)}{d\phi} = \frac{\mu_{min}}{2\pi}, \phi \geq \phi_c. \qquad (6.23)$$

To investigate transitions in the rate of divergence we take our map from Sec. 6.3.1, and vary the free parameters in the kick function (μ_{max}, μ_{min}) as follows: $(10^{-2}. 10^{-5})$, $(10^{-2}. 10^{-8})$ and $(10^{-5}. 10^{-8})$. These pairs of values correspond to the different regimes of behaviour
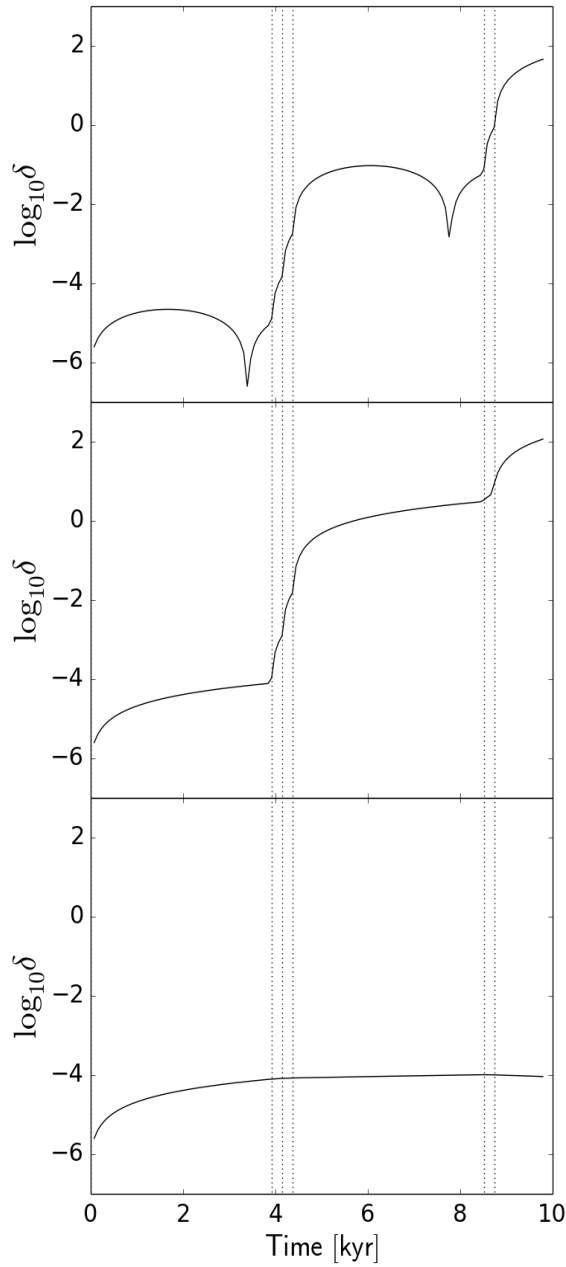
Figure 6.4: Transitions in the rate of divergence as a consequence of the close encounter history between Halley and Jupiter. The close encounter times are marked by the vertical dotted lines. The panels illustrate different types of transitions: oscillatory to exponential (top), linear to exponential (middle) and linear to oscillatory (bottom).

(see Fig. 6.3). We create a small window for the strong perturbations to occur by setting $\phi_c \sim 0.3$, which corresponds to a periodic sequence of roughly two or three strong encounters[1]. We illustrate the results belonging to the encounter history given by $\phi_0 = 0$ in Fig. 6.4. The close encounter events are marked by the vertical, dotted lines.

In the left panel, the growth of the perturbation starts out oscillatory ($\mu = 10^{-5}$). After about 4 kyr there is a sequence of three close encounters causing the characteristic accumulation of power laws or "hops" in the divergence as explained in Sec. 6.2. Once these encounters are over however, the growth becomes oscillatory again.

In the central panel of Fig. 6.4 we observe a similar behaviour, but instead of the oscillatory behaviour we have a linear growth since $\mu = 10^{-8}$ in those intervals. It is clear from these examples that these transitions are caused by a sequence of close encounters. Finally, in the right panel we observe a transition from power law to oscillatory divergence which effectively produces no secular growth.

6.3.4   Liapounov Time

A characteristic time scale for divergence is somewhat difficult to determine for a solution showing transitional behaviour. Once a solution does diverge exponentially it does so with a rate that depends on the strength of the perturbation and the density of close encounters.

To estimate a lower limit for the Liapounov time, we set the orbital period of Halley to $P_h = 76$ yr and that of Jupiter to $P_J = 12$ yr, so that they are exactly in a 3:19 resonance. Therefore, if Halley once experiences a close encounter with Jupiter, it does so every three orbital periods. We use the saw-tooth kick function with $\mu_{min} = 10^{-8}$ (linear growth) and we vary μ_{max} in the exponential regime to measure the Liapounov time as a function of the strength of the perturbation. We measure $\Delta f(t)$ using the map from Sec. 6.3.1 and calculate the approximate Liapounov time using Eq. 6.7, evaluated at the moment when the perturbation $\delta = 1$. We show the result in Fig. 6.5.

The variation in the Liapounov time decreases as μ increases through the critical value corresponding to $\log_{10} \mu \simeq -2.08$, i.e. the transition from periodic behaviour to exponential growth (dashed, green line). From values exceeding a thousand years it quickly drops to a value of the order the orbital period of Halley (blue, horizontal line). Very

---

[1]There is a small drift in the orbital phase because Halley and Jupiter are not exactly in a 3:19 resonance. In fact $3P_h - 19P_j = 0.2$yr if $P_h = 75.3$yr and $P_j = 11.9$yr. It is then easy to see that these sequences of close encounters recur at intervals of about 4.5kyr.
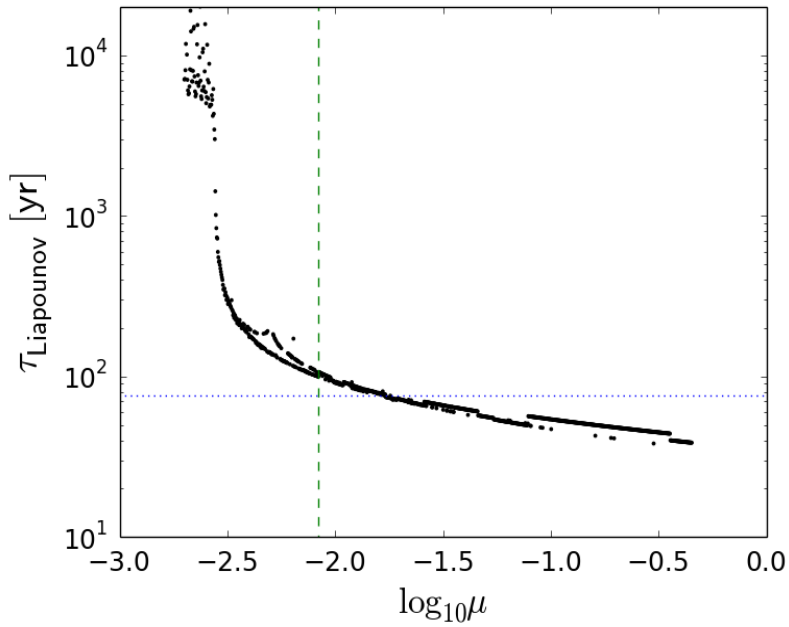
Figure 6.5: Estimated Liapounov time as a function of encounter strength, for a configuration where Jupiter and Halley are in a 3:19 mean motion resonance. The horizontal dotted line gives the orbital period of Halley, and the vertical dashed line marks the critical value of μ as derived in Eq. 6.21.

strong perturbations decrease the Liapounov time to values as low as $30 - 40$ yr. What intuitively seemed like a very short time scale for the exponential growth, actually follows naturally from the recurrence of close encounters in time and the strength of each perturbation. The saturation at low values of μ is due to the finite integration time, whereas the small scatter along the curve is due to errors in calculating the Liapounov time from Eq. 6.7 and the slight variation in the final value of $\delta < 1$.

It must be stressed that these estimates of the Liapounov time are based on the artificial imposition of a resonance ensuring that close encounters recur at every third perihelion passage, i.e. every 228yr. Their frequent recurrence will also ensure that f changes, throwing the system out of resonance and increasing the Liapounov time. For example, in the situation shown in Fig.6.4, corresponding to the present-day periods, close encounters recur on average once every 1.5kyr. Furthermore we have focused on the case in which $\mu > 0$. In case $\mu < 0$ we can see from Eq.(6.18) that the eigenvalues of A are always real, giving exponential growth. When $-1 \ll \mu < 0$, the Liapounov time can be estimated from

$$T_{Liapounov} \simeq \frac{1}{\sqrt{-\mu f_j}}. \tag{6.24}$$

In reality, the kick function $K(\phi)$ is approximately a saw-tooth function, as we have been assuming, but there is a significant difference. Since it is a periodic function, its derivative cannot always be positive. Indeed, as shown by **?**[2] $K(\phi)$ (the kick due to Jupiter) is an increasing function of ϕ, except for a small range of ϕ in which $K' < 0$. Thus K resembles the function considered in this section, except that $\mu_{max} < 0$. (The "max" may be taken to refer to the magnitude of μ.) It therefore seems likely that most encounters are of what we have called the oscillatory type, while there is a minority in which the behaviour is of exponential type.

6.4   N-BODY SIMULATIONS OF HALLEY'S ORBIT

In this section we describe several experiments in which we perform a series of N-body simulations to measure the growth of an initial perturbation in Halley's orbit. We model the dynamical evolution of the solar system according to Newtonian dynamics, in which the bodies

---

[2]Their kick function F is defined as the change in twice the binding energy of Halley, and the binding energy is an increasing function of f. Therefore F and K have the same sign for a given phase.

are mathematical point-particles. Non-gravitational effects, such as radiation pressure from the sun, Halley's mass loss due to the interaction with the stellar wind or internal processes, are neglected. This makes our results less realistic, but for our study on the origin of chaos in Halley's orbit, the gravitational interaction with the sun and planets is sufficient. Relativistic effects, especially the orbital precession of Mercury, will also be neglected. Its influence might become important were it to be shown that Mercury affects the chaoticity of Halley in the Newtonian limit.

We use the N-body code Brutus (**?**), that solves the N-body problem to a pre-defined precision. To make sure that numerical errors do not bias our results, we vary the precision until convergence as described in **?**.

The dominant force in Halley's motion is the sun. Small perturbations are superposed due to the interactions with the planets. solar system bodies smaller than the planets are unlikely to be the cause of chaos in Halley's orbit. We therefore only consider the sun, the eight planets and Halley in our N-body simulations. We obtain the initial conditions from the JPL Horizons database [3].

The orbital elements of Halley are known to about six decimal places (**?**). Two initial realizations which differ within the observational uncertainty are both equally valid representations of the system. When we measure the growth of perturbations, we will use the fiducial initial realization and compare it to a perturbed initial realization in which a single coordinate (usually the x-coordinate of Halley) is perturbed by 10^{-6} AU (similar as in **?**).

6.4.1   Phase Space Distance

A wide variety of methods are available to measure the rate of divergence for a particular orbit (e.g. variational equations (**?**) or finite-time Liapounov exponent (e.g. **?**). We adopt a simple, direct approach. We take a fiducial initial condition for a certain system of bodies. This initial condition is integrated with a pre-determined precision until the end time. We also take the perturbed initial condition, where we translate the position of Halley along the x-direction by the observational uncertainty 10^{-6} AU. This new initial realization is also integrated with the same precision, until the end time. The phase space distance as a function of time between these two solutions is calculated similar as in **?**

---

[3]http://ssd.jpl.nasa.gov/, JDCT = 2456934.5 = A.D. 2014-Oct-04 00:00:00.0000 (CT)
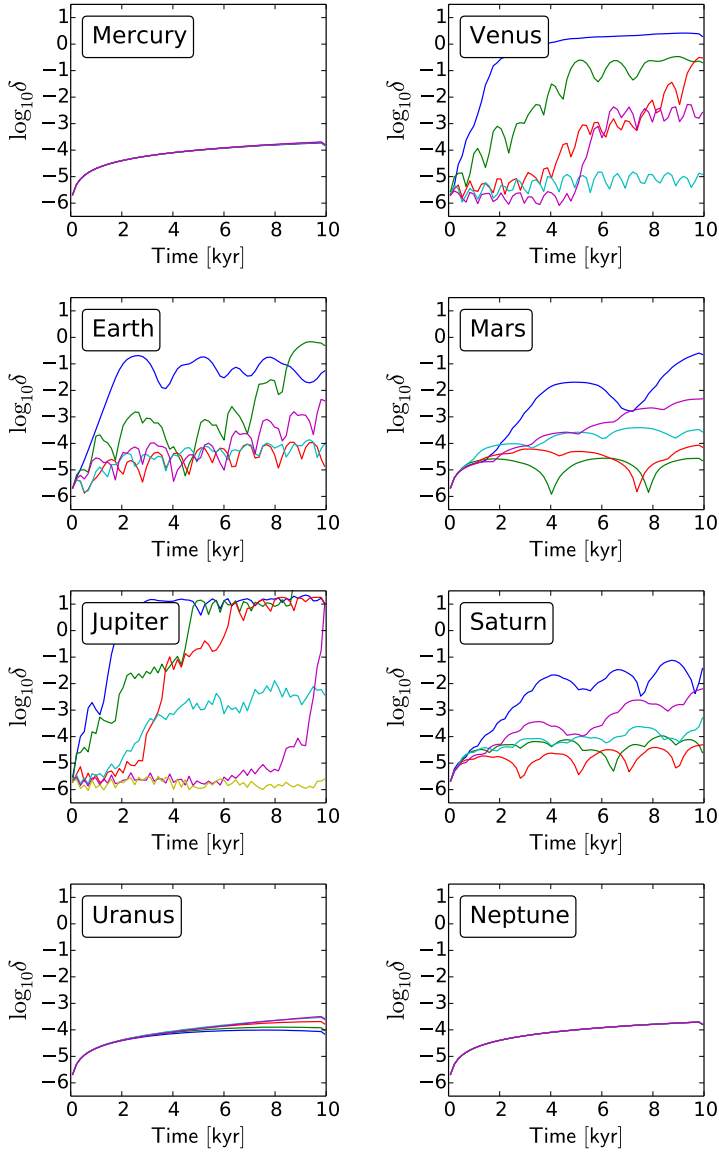
Figure 6.6: Divergence between neighbouring solutions in the $N = 3$ sun, planet and Halley system. We show a subset of solutions to illustrate the different behaviour when we vary the initial orbital phase of the planet around the sun. As a consequence, every solution has a different encounter history with that planet. Mercury, Uranus and Neptune do not influence Halley's chaoticity significantly. The other planets are able to cause exponential growth, most notably Jupiter and Venus.

$$\delta_{A,B}^2 = \sum_{i=1}^{N} \sum_{j=1}^{6} (q_{A,i,j} - q_{B,i,j})^2 . \qquad (6.25)$$

Here q denotes the phase space coordinate for solutions A and B. The sums are over all particles and their phase space coordinates.

If two solutions with the same initial realization are compared, but they were obtained with different precisions, numerical divergence might distort the physical rate of divergence. Therefore, we always compare solutions integrated with the same precision, but with a small, physical perturbation in the initial conditions.

6.4.2   Three-body Divergence: sun, planet and Halley

We already showed some results of our simulations in Fig. 6.2 (bottom left panel), where we integrated a two-body system consisting of the sun and Halley. This result confirms the linear growth of perturbations in two-body systems.

We now introduce a perturbing planet to the system. For each planet we generate an ensemble of a thousand initial conditions, where we vary the initial orbital phase of that planet. In every subsequent integration, Halley will experience a different encounter history with the planet, which should produce different rates of divergence as was already illustrated in Sec. 6.3. We show a subset of illustrative cases in Fig. 6.6.

We first observe the results by Jupiter. The rates of divergence vary widely. There are solutions which stay almost constant within a time span of 10^4 years (yellow curve). In the other extreme are solutions that grow exponentially and have completely diverged within a few thousand years (blue, green and red curves). In between, there are solutions with different kind of transitions in the divergence. After an initial flat phase of a certain duration, a transition to an exponential growth is possible (red and purple curves), but it is also possible for this exponential growth to convert into a power law divergence (cyan curve).

The influence of Saturn on Halley's stability is less strong, but some solutions still grow exponentially for a few thousand years, after which they make a transition to a power law divergence. The magnitude of the perturbation never really becomes the size of the system. The slope in the exponential part of the blue curve is also shallower than the slope in Jupiter's results. The remaining outer planets show a

power law growth and thus have a negligible contribution to Halley's chaoticity.

The influence of the terrestrial planets varies. Mercury shows regular behaviour irrespective of its encounter history with Halley. It is therefore likely that relativistic effects are unimportant for the chaoticity of Halley's orbit. Venus on the other hand shows a variety of solutions similar to Jupiter. The most rapid growing solution looks similar to the one of Jupiter. The majority of Earth and Mars solutions show a power law divergence superposed with periodic variations. Note however, that they are able to generate a rapid rate of divergence in some situations.

6.4.3 Hopping Between Planets

In this experiment we do not randomize the initial orbital phase, but we take the fiducial initial conditions so that we can measure the actual encounter histories of the planets with Halley. We consider the 3-body systems including the sun, a planet and Halley, to measure the independent rates of divergence. Based on the results of Sec. 6.4.2, we neglect Mercury, Uranus and Neptune. We compare these results with a simulation including all the relevant planets collectively. The results are given in Fig. 6.7. We averaged the data over bins of two orbital periods to reduce the short term oscillatory behaviour.

We observe that only Venus (green curve) and Jupiter (yellow curve) produce an exponential divergence. Initially the perturbation due to Venus dominates, but it is overtaken by Jupiter after about 3000 years. The solution including multiple planets (black curve), follows this transition, first following the perturbations due to Venus and then hopping onto the perturbations by Jupiter. Other effects are present since the black curve does not lie perfectly on top of the green and yellow curves. The superposition of independent growth rates is however a reasonable approximation in this example.

From the time evolution of the perturbation in the complete system (black curve) we calculate the average Liapounov time up to the point where $\delta = 1$ resulting in 299 yr $\pm$ 62 yr, where the uncertainty is the standard deviation in the variation of the Liapounov time from $t = 0$ onwards.

To investigate the dependency on the direction of the perturbation, we varied the initial perturbation in Halley's orbit to lie along the x, y or z-direction. We find that in each case Venus is dominant for at least 3000 yr. For the y-direction, Venus remains dominant up to 4000 yr. The rate of divergence due to Venus depends sensitively on Halley's
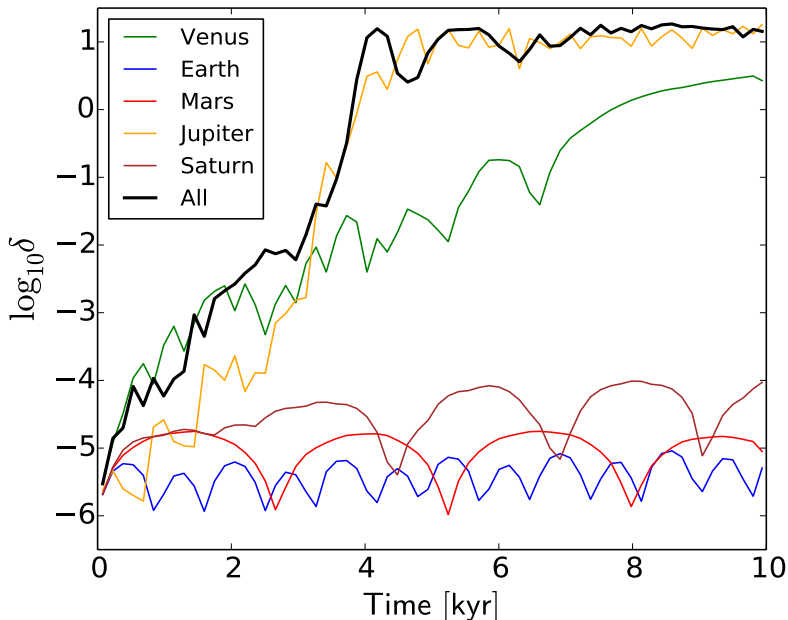
Figure 6.7: Growth of perturbations in time for the different planets independently and with the planets collectively (fat curve). Up to 3000 years, Venus is the dominant perturber of Halley's orbit. Then a transition occurs and Jupiter becomes the main perturber. The transition in the rate of divergence for the solution including all planets is explained by the superposition of independent rates of divergence of the planets.

orbit. We also performed a similar experiment where we integrated backwards in time. We find that both Venus and Jupiter show an exponential divergence, reaching $\log_{10}\delta = 1$ after $3 - 4000$ yr.

6.4.4 Ensemble Simulations

In the previous section we varied the perturbation in Halley's orbit along three different directions in space. In this experiment we want to vary the perturbation in all directions in space. Instead of comparing a fiducial and a perturbed solution, we take an ensemble of a hundred Halley-like objects, which are distributed around the fiducial initial position, in a three-dimensional Gaussian distribution with a dispersion of 10^{-6} AU. This eliminates any chance effects of preferred spatial directions. We only consider the perturbations due to the sun and Jupiter. Starting points are the current positions of the sun, Jupiter and Halley. The simulations are done with the Huayno

integrator (**?**). To study the influence of the strength of the perturbations, we vary the mass of Jupiter by multiplying it by a factor ranging from zero to five. We measure the spread in the positions of the Halley-like objects, i.e. the standard deviation in the position of the ensemble, as a function of time.

We observe in Fig. 6.8 that if the planet has zero mass, we get a linear growth in the dispersion of the positions of the swarm, as expected from the imposed distribution in orbital periods. For small Jupiter masses, i.e. a mass smaller than the actual Jupiter mass, we get a sub-linear growth with an oscillatory behaviour, which we now can understand from our previous analysis to be due to the weak encounter nature of the interactions. Comparing the cases of $0.2 \times M_{jup}$ and $0.5 \times M_{jup}$ we see that the mass of Jupiter is of little influence, until there is a strong perturbation, which happens after 9000 yr for half of Jupiter's mass (red curve). The increase in mass versus $0.2 \times M_{jup}$ remarkably does not increase the growth at all by weak perturbations before that time, it is just increasing the probability of eventually encountering a strong interaction. For heavier Jupiters (i.e. $1 \times M_{jup}$ and heavier), we obtain a rather fast exponential divergence due to prompt strong interactions.

Note that the experiment conducted here considers the evolution of an ensemble of Halley-like objects, but the results equally apply to a swarm of objects (e.g. the result of an asteroid collision or dust emitted from a cometary nucleus). This means that in configurations where the orbit does not encounter strong interactions, but is affected by weak perturbations such a swarm will survive as a coherent group longer than might be expected from the linear spreading with time.

6.5   DISCUSSION AND CONCLUSIONS

6.5.1   The Liapounov Time

Previous studies have considered the value of the Liapounov time for the growth of perturbations in Halley's orbit. **?** gave an estimate of a lower bound of 34 yr for the Liapounov time, and our estimate is consistent with this. Our estimate is, however, inconsistent with the results of **?**, who found a value around 70 yr. This was based on an initial perturbation in the y-coordinate of Halley, but they also gave results for an initial perturbation in the x-coordinate (their Fig.7) which would give a Liapounov time only slightly longer. We note, however, that their plot of the growth of the deviation between two orbits (their Fig.6) indicates growth in δ (their measure of the separation of two
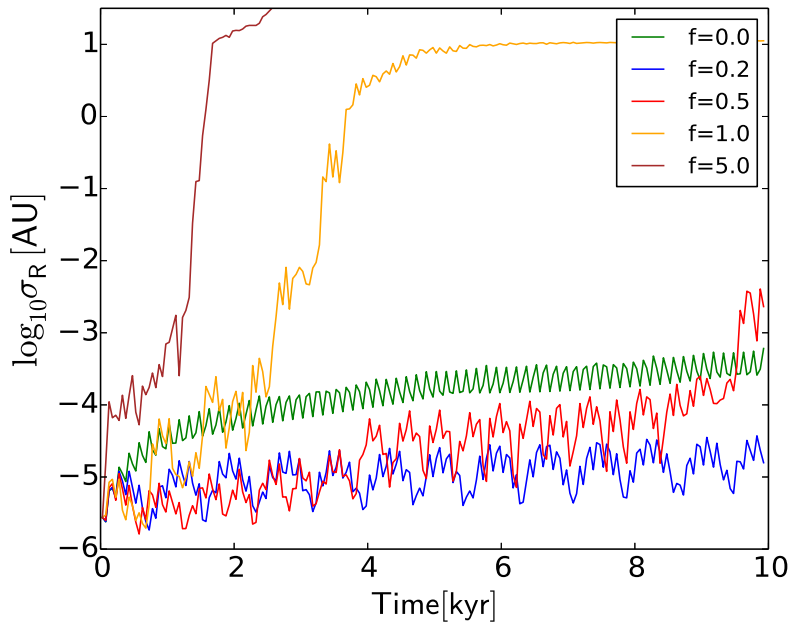
Figure 6.8: Growth of the spread in position of an ensemble of Halley-like objects. We vary the mass of Jupiter by multiplying it by a fraction given in the legend. We reproduce the linear, sub-linear and exponential growth, depending on the strength of the perturbation.

orbits) by about 5 dex in 3.5 kyr, implying a Liapounov time of order 300 years, very similar to ours. Our value for the growth in the separation of two orbits in 3.5 kyr is similar to theirs (see Fig.6.7), and so we suspect an error in their computation of the Liapounov time from correct data.

The Liapounov time of Halley's Comet is determined principally by perturbations due to Venus and Jupiter (Fig.6.6). The influence of Earth, Mars and Saturn is smaller during the next few millennia, and that of Mercury, Uranus and Neptune is negligible. The comparable importance of Jupiter and Venus could not have been guessed from their relative masses. The surprising fact that the mass of a planet does not directly measure its influence on the Liapounov time is illustrated in the very interesting calculations of the kick function (as a function of phase ϕ) by **?**, their Fig.2. That due to Venus has a maximum value about one hundredth of the maximum kick due to Jupiter, which is roughly in proportion to their mass, even though these two planets contribute roughly equally to the Liapounov exponent for Halley. The reason for this is the contribution also depends on the distance of closest approach. This is made apparent by the fact that the divergence caused by these two planets depends strongly on the initial phase (Fig.6.6 again). Indeed **?** draw attention to a forthcoming relatively close encounter with Jupiter after about 3.4 kyr, and its influence is visible in Fig.6.7. We drew attention to the importance of a near-resonance in the motions of Halley and Jupiter, and its importance for the growth of divergence between neighbouring orbits (Fig.6.4), and for different planets such configurations will occur at different periods, as the orbits of the system evolve. The importance of Venus to Halley's chaoticity can be explained by noting that Halley crosses the orbital plane of the solar system close to the orbit of Venus.

Much of our focus in Sec.6.3 was on the parameter μ, which measures the derivative of our kick function $K(\phi)$. This also can be estimated from the results of **?**, bearing in mind that their kick function $F(x)$ is the change (per perihelion passage) in twice the binding energy of Halley, as a function of $x = \phi/(2\pi)$. For Venus the largest value of $|F'|$ occurs over a range of x of order 0.1 in which F decreases between values of about $\pm 0.5 \times 10^{-4}$. Thus we estimate $F' \simeq -10^{-3}$, and infer that $K' \simeq -10^{-5}$, though care has to be taken with the different units used in the two studies. This results in $\mu \simeq -6 \times 10^{-5}$ and so, using Eq.(6.24), we estimate that the corresponding Liapounov time is of order 400 yr. This is of the correct order to account for the most rapid growth in Fig.6.6 (second panel), but it would only occur for phase values within a fairly narrow range. For Jupiter, similar estimates give

a Liapounov time an order of magnitude smaller, again over a similar, limited range of phases. For Venus there is actually another larger range of phase with $K' < 0$, but $|K'|$ is smaller than the estimate we have given, and the Liapounov time correspondingly longer. For both planets the magnitude of K' is smaller than these upper limits, and so when $K' > 0$ Halley remains in the regime of oscillatory "growth" (Sec.6.3). When the phases are such that this occurs, it is interesting to note that these perturbations make Halley more stable compared to having no perturbations at all.

Even neglecting the other planets, Venus and Jupiter clearly both contribute strongly to the growth of the divergence between neighbouring orbits. Fig.6.7 illustrates that a sort of superposition principle is approximately valid. It can be seen that the yellow curve for Jupiter alone does not rise quickly immediately, but only after about 2 kyr. The green curve due to Venus gives the opposite behaviour, from a transition from rapid growth to one of slower growth. Meanwhile the black curve (for the entire planetary system) exhibits a transition from the green onto the yellow curve, much as if the two latter effects were superposed. The rate of divergence of the collective system will closely follow that of the most rapidly diverging individual perturbation.

Many factors have been ignored in our work. As shown by **?**, Fig. 5, the kick functions are not constant in time. Small variations in the orbital elements alter the maximum value of the derivative of the kick function. The Liapounov time changes in time. In this work the changes in Δf were assumed to be mainly caused by close encounters with a planet. Our model is however more general and other events could cause a change in the orbital frequency as well. For example, Halley lost a significant amount of mass during an event in 1991 (**?**), the origin of which is unknown. Even though the mass loss will be the same in both neighbouring solutions, the difference in orbital elements will cause the effect of the mass loss event to be slightly different.

6.5.2 Conclusions

The orbit of Comet Halley is chaotic (**????**) with a Liapounov time which we find to be 299 ± 62 yr (measured over approximately the next 4 kyr). The aim of this study is to understand the origin of chaos in Halley's orbit and its relatively short Liapounov time.

We present a model to explain the origin of chaos in few-body systems with comparable masses. We start by considering a two-body system. As is well known, in this system the difference between two solutions grows linearly, proportional to the difference in orbital fre-

quency, Δf, between the two solutions. When more than two bodies are present, a sequence of events (e.g. close encounters with a third body) changes Δf. A sequence of strong encounters produces an accumulation of power laws which resembles exponential growth.

Next we develop a model better adapted to the case of Comet Halley, where the perturbations are almost always weak, because the masses of the planets are small and very close encounters are rare. This model uses a map to describe the evolution of Δf in Halley's orbit. For very weak perturbations, Δf remains roughly constant, which results in a linear growth of perturbation or weak exponential divergence, depending on the sign of the perturbation. For somewhat stronger encounters the perturbation shows either oscillatory behaviour without secular growth (thus making Halley's orbit more stable in a sense), or exponential growth, again depending on the sign of the perturbation. Above a certain threshold in the size of the perturbation, we demonstrate that a sequence of close encounters is able to produce exponential growth, irrespective of the sign of the perturbation, with a Liapounov time of order the orbital period of Halley.

To calculate the growth of perturbations in Halley's orbit more accurately, we perform numerical N-body integrations. We find the surprising result that Venus is the dominant cause of chaos in Halley's orbit (see Fig. 6.7).

7

Future Directions

Several directions in the field of research of N-body systems and simulations are discussed, where further improvement is necessary. The first one is speed of N-body simulations in order to be able to handle large systems such as globular clusters. Especially the presence of binaries causes a major slow down. The second direction is precision, which is crucial when simulating systems including millisecond pulsars for example. Through high precision observations and simulations we can test theories of gravity to many decimal places. Finally, a better understanding of the quasi-ergodic property of gravity and the growth of perturbations will provide clues for designing new N-body methods and a more fundamental understanding of gravity and the evolution of dynamical systems.

7.1   SPEED

The first N-body simulations in the 1960s calculated the orbits of a few dozen stars (von Hoerner, 1960; Aarseth, 1963; **?**). Each decade later, this number increased by an order of magnitude, due to advances in computer hardware (**?**). The first star by star simulation of a rich globular cluster, with nearly half a million stars, has been completed recently (**?**), and now we are on the brink of the first such simulations with a million bodies (**?**). The comparison between a million-body star cluster simulation and observations of globular clusters, will involve less assumptions than comparing with simulations of a smaller number of bodies. For example, the relaxation time of a cluster depends on the number of bodies, and also rare events, such as collisions between stars or the fraction of exotic objects in the cluster, such as pulsars and black holes, depend on the number of bodies in the system.

The development of N-body methods is still an ongoing process, since there are certain types of configurations that are still hard to simulate efficiently. For example, star clusters are born with a fraction of primordial binary stars (**?**). The ratio between the crossing time of the cluster and the orbital period of a binary can be as high as six orders of magnitude. Integrating these systems directly is very

computationally expensive. Approximation schemes, such as replacement by a center of mass body or nearest-neighbour methods (Aarseth, 2003), have to be implemented to be able to integrate these systems at all. Handling binary and higher-order systems in star clusters is challenging and new methods have to be constructed to improve the performance, while maintaining sufficient accuracy..

7.2   PRECISION

Some dynamical systems can be observed very precisely. The orbit of Halley's Comet for example, is known to about six decimal places (**?**). Since the comet is relatively nearby, it was possible to sent a spacecraft in 1986 to explore the comet from up close. Another example is that of a system including a pulsar. Pulsars are fast rotating neutron stars from which we receive a signal twice every rotational period (**?**). The timing of these signals can be precise up to 15 decimal places, making them one of the most accurate clocks in nature (**?**). Variations in these timing measurements are caused by internal processes in the pulsar, but can also hold information about its dynamical environment.

These precise systems need to be modelled using both accurate physical models and integration techniques. Starting with Newtonian dynamics and a precise N-body integrator such as Brutus (**?**), the simulation data can be compared to the observations. If there are residuals left in the comparison, higher-order effects have to be taken into account to solve for them. These could be relativistic effects in systems with high masses or large velocities, finite size effects if objects approach each other closely, or simply reducing the effect of numerical errors. Through studies such as these, and applying them to systems of extreme gravity including pulsars or black holes, we can test theories of gravity to high precision.

7.3   RELIABILITY AND CHAOS

In this thesis, we investigated the reliability of numerical integration for a subset of three-body systems (**?**). The fact that we confirm the preservation of statistics under divergence of solutions, does not guarantee that other N–body systems do the same (**?**). Starting with a four-body system, one body can escape, leaving behind a three-body system. The configuration of this three-body system however, is probably different than the three-body systems we investigated. Therefore, in principle, every value of N should be tested independently.

Such studies can only show whether the results from N-body simulations are correct in a statistical sense. They do not however, explain why this is the case. More theoretical work is needed to understand the quasi-ergodic behaviour of gravity (**?**). For what type of systems does it hold, and are there systems for which it breaks down? This last case would be particularly interesting, because results of simulations of such systems would have to be reconsidered.

Some light in this largely unexplored field of research, might be shed by investigating a related topic on the dynamical stability and growth of perturbations in dynamical systems. In this thesis, we started by constructing a model for the growth of perturbations, explaining both regular and chaotic behaviour in the orbit of Halley's Comet (Boekholt et al., In prep.). This study should be expanded by including more general orbits and configurations. By understanding how perturbations or errors propagate through a dynamical system, we will also be in a better position to design new N-body methods, which are adapted to our understanding of error growth. Apart from improving our simulation methods, it will also provide us with a more fundamental understanding of the way gravity works, which will lead to a better understanding of the evolution of dynamical systems in the universe

The field of N-body methods and simulations has become increasingly important for astronomers. It has improved our understanding of dynamical systems, and it will continue to make a significant impact in astronomy. In this age of big data, where many large surveys, such as the Gaia mission[1], will acquire more complex and deeper observations, more sophisticated and realistic modelling is required, in order to interpret the observations from a dynamical point of view (**?**).

---

[1]http://sci.esa.int/gaia/

Bibliography

Aarseth S. J., 1963, Monthly Notices of the Royal Astronomical Society, 126, 223

Aarseth S. J., 2003, Gravitational N-Body Simulations

Aarseth S. J., Anosova J. P., Orlov V. V., Szebehely V. G., 1994, Celestial Mechanics and Dynamical Astronomy, 58, 1

Aarseth S. J., Henon M., Wielen R., 1974, Astronomy and Astrophysics, 37, 183

Aarseth S. J., Hoyle F., 1964, Astrophysica Norvegica, 9, 313

Aarseth S. J., Lecar M., 1975, Annual Review of Astronomy and Astrophysics, 13, 1

Ahmad A., Cohen L., 1973, Journal of Computational Physics, 12, 389

Babadzhanyants L. K., 1981, Pisma v Astronomicheskii Zhurnal, 7, 752

Barnes J., Hut P., 1986, Nature, 324, 446

Bédorf J., Gaburov E., Fujii M. S., Nitadori K., Ishiyama T., Portegies Zwart S., 2014, ArXiv e-prints

Bédorf J., Portegies Zwart S., 2012, European Physical Journal Special Topics, 210, 201

Binney J., Tremaine S., 1987, Galactic dynamics

Boekholt T., Portegies Zwart S., 2015, Computational Astrophysics and Cosmology, 2, 2

Boekholt T. C. N., McMillan S. L. W., Portegies Zwart S. F., 2015, In preparation

Boltzmann L., 1868, Wiener Berichte, p. 517–560

Bulirsch R., Stoer J., 1964, Numerische Mathematik, pp 413–427

Burrau C., 1913, Astronomische Nachrichten, 195, 113

Caputo D. P., de Vries N., Portegies Zwart S., 2014, Monthly Notices of the Royal Astronomical Society, 445, 674

Chambers J. E., 1999, Monthly Notices of the Royal Astronomical Society, 304, 793

Chandrasekhar S., 1942, Principles of stellar dynamics

Chenciner A., Montgomery R., 2000, ArXiv Mathematics e-prints

Chirikov R. V., Vecheslavov V. V., 1989, Astronomy and Astrophysics, 221, 146

Conway B. A., 1986, Celestial Mechanics, 39, 199

Copernicus N., 1543, De revolutionibus orbium coelestium.

Dejonghe H., Hut P., 1986, in Hut P., McMillan S. L. W., eds, The Use of Supercomputers in Stellar Dynamics Vol. 267 of Lecture Notes in Physics, Berlin Springer Verlag, Round-Off Sensitivity in the N-Body Problem. p. 212

Duncan M. J., Levison H. F., Lee M. H., 1998, Astronomical Journal, 116, 2067

Euler L., 1767, Novi Commentarii academiae scientiarum Petropolitanae, 11, 103

Faber N. T., Boily C. M., Portegies Zwart S., 2008, Monthly Notices of the Royal Astronomical Society, 386, 425

Fujii M., Iwasawa M., Funato Y., Makino J., 2007, Publications of the Astronomical Society of Japan, 59, 1095

Galilei G., 1610, Sidereus Nuncius.

Goodman J., Heggie D. C., Hut P., 1993, Astrophysical Journal, 415, 715

Gragg W. B., 1965, SIAM Journal on Numerical Analysis, pp 384–403

Hairer E., Lubich C., Wanner G., 2002, Geometric numerical integration : structure-preserving algorithms for ordinary differential equations. Springer series in computational mathematics, Springer, Berlin, Heidelberg, New York

Halley E., 1705, Synopsis of the Astronomy of Comets.

Hartnett J. G., Luiten A. N., 2011, Rev. Mod. Phys., 83, 1

Hayes W. B., 2008, Monthly Notices of the Royal Astronomical Society, 386, 295

Heggie D., Hut P., 2003, The Gravitational Million-Body Problem: A Multidisciplinary Approach to Star Cluster Dynamics

Heggie D. C., 1975, Monthly Notices of the Royal Astronomical Society, 173, 729

Heggie D. C., 1991, in Roeser S., Bastian U., eds, Predictability, Stability, and Chaos in N-Body Dynamical Systems Chaos in the N-body problem of stellar dynamics.. pp 47–62

Heggie D. C., 2000, Monthly Notices of the Royal Astronomical Society, 318, L61

Heggie D. C., 2014, Monthly Notices of the Royal Astronomical Society, 445, 3435

Heggie D. C., Mathieu R. D., 1986, in Hut P., McMillan S. L. W., eds, The Use of Supercomputers in Stellar Dynamics Vol. 267 of Lecture Notes in Physics, Berlin Springer Verlag, Standardised Units and Time Scales. p. 233

Hénon M. H., 1971, apss, 14, 151

Hernquist L., Hut P., Makino J., 1993, Astrophysical Journal Letters, 402, L85

Hockney R. W., Eastwood J. W., 1988, Computer simulation using particles

Hughes D. W., 1985, Monthly Notices of the Royal Astronomical Society, 213, 103

Hut P., 1983, Astronomical Journal, 88, 1549

Hut P., 1993, Astrophysical Journal, 403, 256

Hut P., Bahcall J. N., 1983, Astrophysical Journal, 268, 319

Hut P., Heggie D. C., 2002, Journal of Statistical Physics, 109, 1017

Hut P., McMillan S., Goodman J., Mateo M., Phinney E. S., Pryor C., Richer H. B., Verbunt F., Weinberg M., 1992, Publications of the Astronomical Society of the Pacific, 104, 981

Ito T., Tanikawa K., 2002, Monthly Notices of the Royal Astronomical Society, 336, 483

Jenkins R. M., 2004, Journal of the British Astronomical Association, 114, 336

Jílková L., Portegies Zwart S., Pijloo T., Hammer M., 2015, ArXiv e-prints

Johnstone D., Rucinski S. M., 1991, pasp, 103, 359

Kandrup H. E., 1996, in Jantzen R. T., Mac Keiser G., Ruffini R., eds, Proceedings of the Seventh Marcel Grossman Meeting on recent developments in theoretical and experimental general relativity, gravitation, and relativistic field theories Chaos, regularity, and noise in self-gravitating systems.. pp 167–182

Kepler J., 1609, Astronomia Nova.

Kinoshita H., Yoshida H., Nakai H., 1991, Celestial Mechanics and Dynamical Astronomy, 50, 59

Kokubo E., Yoshinaga K., Makino J., 1998, Monthly Notices of the Royal Astronomical Society, 297, 1067

Kolmogorov A., 1933, 1st. Ital. Attuari. G., 4, 1–11

Lagrange J. L., 1772, Prix de l'Académie Royale des Sciences de Paris, 6, 292

Landgraf W., 1986, Astronomy and Astrophysics, 157, 245

Laskar J., 1989, Nature, 338, 237

Laskar J., 2008, Icarus, 196, 1

Laskar J., Gastineau M., 2009, Nature, 459, 817

Lyapunov A. M., Walker J. A., 1994, JAM, 61, 226

Lyne A. G., 1996, in van Paradijs J., van den Heuvel E. P. J., Kuulkers E., eds, Compact Stars in Binaries Vol. 165 of IAU Symposium, A Review of Galactic Millisecond Pulsar Searches. p. 225

Makino J., 1994, in Gurzadyan V. G., Pfenniger D., eds, Ergodic Concepts in Stellar Dynamics Vol. 430 of Lecture Notes in Physics,

Berlin Springer Verlag, Smoothing of the Cosmic Background Radiation by Multiple Gravitational Scattering. p. 274

Makino J., Aarseth S. J., 1992, Publications of the Astronomical Society of Japan, 44, 141

Mardling R. A., 2008, in Vesperini E., Giersz M., Sills A., eds, IAU Symposium Vol. 246 of IAU Symposium, Resonance, Chaos and Stability in the General Three-Body Problem. pp 199–208

Martínez-Barbosa C. A., Brown A. G. A., Portegies Zwart S., 2015, ArXiv e-prints

Maxwell J. C., 1867, Phil. Trans. R. Soc. Lond., 157, 49

McMillan S., Portegies Zwart S., van Elteren A., Whitehead A., 2012, in Capuzzo-Dolcetta R., Limongi M., Tornambè A., eds, Advances in Computational Astrophysics: Methods, Tools, and Outcome Vol. 453 of Astronomical Society of the Pacific Conference Series, Simulations of Dense Stellar Systems with the AMUSE Software Toolkit. p. 129

McMillan S. L. W., 1986, in Hut P., McMillan S. L. W., eds, The Use of Supercomputers in Stellar Dynamics Vol. 267 of Lecture Notes in Physics, Berlin Springer Verlag, The Vectorization of Small-N Integrators. p. 156

McMillan S. L. W., Hut P., 1996, Astrophysical Journal, 467, 348

Mikkola S., Aarseth S., 2002, Celestial Mechanics and Dynamical Astronomy, 84, 343

Mikkola S., Tanikawa K., 1999a, Monthly Notices of the Royal Astronomical Society, 310, 745

Mikkola S., Tanikawa K., 1999b, Celestial Mechanics and Dynamical Astronomy, 74, 287

Miller R. H., 1964, Astrophysical Journal, 140, 250

Monaghan J. J., 1976, Monthly Notices of the Royal Astronomical Society, 176, 63

Moore C., 1993, Phys. Rev. Lett., 70, 3675

Moser J., ed. 1973, Stable and random motions in dynamical systems. With special emphasis on celestial mechanics

Muñoz-Gutiérrez M. A., Reyes-Ruiz M., Pichardo B., 2015, Monthly Notices of the Royal Astronomical Society, 447, 3775

Newton I., 1687, Philosophiae Naturalis Principia Mathematica.

Nyland L., Harris M., Prins J., 2007, in Nguyen H., ed., , GPU Gems 3. Addison Wesley Professional, Chapt. 31

Papamarinopoulos S. P., Preka-Papadema P., Antonopoulos P., Mitropetrou H., Tsironi A., Mitropetros P., 2013, Mediterranean Archaeology & Archaeometry, 13, 69

Pelupessy F. I., Jänes J., Portegies Zwart S., 2012, New Astronomy, 17, 711

Plummer H. C., 1911, Monthly Notices of the Royal Astronomical Society, 71, 460

Poisson E., Will C. M., 2014, Gravity

Portegies Zwart S., Bédorf J., 2014, ArXiv e-prints

Portegies Zwart S., Boekholt T., 2014, Astrophysical Journal Letters, 785, L3

Portegies Zwart S., McMillan S., Groen D., Gualandris A., Sipior M., Vermin W., 2008, New Astronomy, 13, 285

Portegies Zwart S., McMillan S., Pelupessy I., van Elteren A., 2012, in Capuzzo-Dolcetta R., Limongi M., Tornambè A., eds, Advances in Computational Astrophysics: Methods, Tools, and Outcome Vol. 453 of Astronomical Society of the Pacific Conference Series, Multi-physics Simulations using a Hierarchical Interchangeable Software Interface. p. 317

Portegies Zwart S., McMillan S. L. W., van Elteren E., Pelupessy I., de Vries N., 2013, Computer Physics Communications, 183, 456

Portegies Zwart S. F., McMillan S. L. W., Hut P., Makino J., 2001, Monthly Notices of the Royal Astronomical Society, 321, 199

Preto M., Tremaine S., 1999, Astronomical Journal, 118, 2532

Prialnik D., Bar-Nun A., 1992, Astronomy and Astrophysics, 258, L9

Quinlan G. D., Tremaine S., 1992, Monthly Notices of the Royal Astronomical Society, 259, 505

Rollin G., Haag P., Lages J., 2014, ArXiv e-prints

Seargent D. A. J., 2009, The Greatest Comets in History

Shevchenko I. I., 2007, in Valsecchi G. B., Vokrouhlický D., Milani A., eds, IAU Symposium Vol. 236 of IAU Symposium, On the Lyapunov exponents of the asteroidal motion subject to resonances and encounters. pp 15–30

Shimada M., Yoshida H., 1996, Publications of the Astronomical Society of Japan, 48, 147

Simon L., 1986, La Recherche, 17, 854

Smirnov N., 1948, The Annals of Mathematical Statistics, 19, 279

Smith Jr. H., 1979, Astronomy and Astrophysics, 76, 192

Spitzer L., 1987, Dynamical evolution of globular clusters

Sussman G. J., Wisdom J., 1992, Science, 257, 56

Szebehely V., 1972, Proceedings of the National Academy of Science, 69, 1077

Szebehely V., Peters C. F., 1967, Astronomical Journal, 72, 876

Tanikawa A., Hut P., Makino J., 2012, New Astronomy, 17, 272

Urminsky D., 2008, in Vesperini E., Giersz M., Sills A., eds, IAU Symposium Vol. 246 of IAU Symposium, On the Calculation of Average Lifetimes for the 3-body Problem. pp 235–236

Urminsky D. J., 2010, Monthly Notices of the Royal Astronomical Society, 407, 804

Urminsky D. J., Heggie D. C., 2009, Monthly Notices of the Royal Astronomical Society, 392, 1051

Valtonen M., Karttunen H., 2006, The Three-Body Problem

Valtonen M., Mylläri A., Orlov V., Rubinov A., 2004, in Byrd G. G., Kholshevnikov K. V., Myllri A. A., Nikiforov I. I., Orlov V. V., eds, Order and Chaos in Stellar and Planetary Systems Vol. 316 of Astronomical Society of the Pacific Conference Series, Statistical Approach to the Three-Body Problem. p. 45

van Albada T. S., 1968, Bulletin of the Astronomical Institutes of the Netherlands, 19, 479

van Zon R., van Beijeren H., Dellago C., 1998, Physical Review Letters, 80, 2035

Vecheslavov V. V., Chirikov B. V., 1988, Pisma v Astronomicheskii Zhurnal, 14, 357

Verlet L., 1967, PhysRev, 159, 98–103

Vlasov A. A., 1968, Soviet Physics Uspekhi, 10, 721

von Hoerner S., 1960, Z. Astrophys, 50, 184

Wang L., Spurzem R., Aarseth S., Nitadori K., Berczik P., Kouwenhoven M. B. N., Naab T., 2015, Monthly Notices of the Royal Astronomical Society, 450, 4070

Wisdom J., Holman M., 1991, Astronomical Journal, 102, 1528

Yoshida H., 1990, Physics Letters A, 150, 262

Zadunaisky P. E., 1979, Celestial Mechanics, 20, 209

Nederlandse
Samenvatting

"Ik denk vaak dat de nacht levendiger en kleurrijker is dan de dag."
– Vincent van Gogh

DYNAMISCHE SYSTEMEN

In het heelal is een oneindig aantal voorbeelden van dynamische systemen aanwezig. Elk systeem dat bestaat uit een verzameling van objecten, ook wel lichamen genoemd, die bewegen door ruimte en tijd onder de invloed van hun onderlinge krachten, is een geldige kandidaat.

Ons zonnestelsel is een bekend voorbeeld bestaande uit verschillende soorten lichamen. We hebben de zon in het midden (Copernicus, 1543), de planeten in ellipsbanen rond de zon (Kepler, 1609), manen in banen rond de planeten (Galilei, 1610), en dan zijn er nog zeer veel relatief kleine lichamen, zoals asteroïden en kometen (Halley, 1705) (zie Fig. 1.1, bovenste paneel).

Deze hemellichamen hebben allemaal één ding gemeen, ze hebben allen massa. Sinds het baanbrekende werk van Newton (1687), weten we dat lichamen met massa elkaar aantrekken. Elk object voelt de aantrekkingskracht van alle andere objecten. Het gevolg is dat ons zonnestelsel niet statisch is, maar evolueert, omdat alle lichamen bewegen. Als alle lichamen bewegen, dan is het ook mogelijk dat twee van hen rakelings langs elkaar heen scheren, zoals komeet Halley en de aarde in 1910.

Een ander voorbeeld van een dynamisch systeem is een cluster van sterren, ook wel een bolhoop genoemd (zie Fig. 1.1, onderste paneel). Het merendeel van de lichamen in zo'n systeem is een gewone ster, maar vaak komen er ook exotische objecten in voor, zoals neutronensterren en zwarte gaten. Een bolhoop is anders dan ons zonnestelsel, omdat er nu niet één dominante massa aanwezig is (de zon is duizend keer zwaarder dan alle planeten in ons zonnestelsel bij elkaar), maar alle massa's zijn nu gelijkwaardig. Dit zorgt voor een andere dynamica en evolutie van het systeem als geheel.

Het aantal sterren varieert per bolhoop, van enkele tientallen tot miljoenen. Het aantal is belangrijk omdat het invloed heeft op de

banen van de sterren. Stel dat onze zon zich in een cluster van sterren bevindt (men denkt dat dit zo was toen de zon nog jong was (Martínez-Barbosa et al., 2015; Jílková et al., 2015)), en er is alleen maar een handjevol andere sterren aanwezig, dan is de aantrekkingskracht van elke ster op de zon een significante fractie van het totaal. Als er echter een miljoen andere sterren aanwezig zijn, is elke individuele contributie nietig. Het gevolg is dat de baan van de zon bepaald wordt door alle sterren als geheel, waardoor de baan gladder is en op een langere tijdschaal verandert. Deze tijdschaal wordt ook wel de relaxatietijd genoemd (Chandrasekhar, 1942) en geeft de typische tijdschaal voor een bolhoop om zijn structuur te veranderen.

In de kernen van massieve bolhopen worden interacties tussen enkele sterren echter weer belangrijk. Vooral drie-lichamen interacties vormen een categorie op zich. Door de hoge dichtheid in massieve bolhopen, kunnen drie sterren zo dicht bij elkaar komen, dat ze voor een relatief korte tijd effectief geïsoleerd van de rest van het systeem zijn. Deze drie sterren wisselen energie uit en het kan gebeuren dat één van de drie sterren genoeg energie steelt van de andere twee sterren, zodat het genoeg bewegingsenergie heeft om uit de kern van het cluster te schieten. De overgebleven twee sterren hebben energie verloren en raken gebonden aan elkaar. Ze vormen samen een twee-lichamen systeem, oftewel een dubbelster (Szebehely, 1972). Deze dubbelster zal weer interacties aangaan met de andere nabije sterren, waarbij het bewegingsenergie weggeeft. Dit zorgt ervoor dat de structuur in de kern van de bolkoop verandert. Veel theoretische en numerieke studies zijn er gedaan van de precieze energie-uitwisseling tussen enkele en dubbelsterren in bolhopen (bijvoorbeeld Heggie, 1975; Hut & Bahcall, 1983; McMillan & Hut, 1996; Boekholt et al., 2015).

Planetaire systemen, bolhopen en drie-lichamen systemen zijn maar enkele voorbeelden van dynamische systemen in het heelal. Er zijn er nog veel meer, bijvoorbeeld sterrenstelsels die bestaan uit miljarden sterren, of kernen van sterrenstelsels die een supermassief zwart gat bevatten. Van al deze systemen zijn er enkele dingen die we graag zouden willen weten. Om precies te zijn, hoeveel en wat voor type lichamen zijn er aanwezig in het systeem, hoe zijn ze verdeeld in de ruimte, hoe verandert de globale structuur in de tijd en hoe zal het systeem tot zijn einde komen? Als een specifiek voorbeeld nemen we weer ons zonnestelsel, maar we beschouwen alleen de zon en de planeten. De massa's, posities en de snelheden zoals ze vandaag de dag zijn, zijn heel precies gemeten. Eén van de open vragen is hoe de banen van de planeten evolueren over een tijdschaal van miljarden jaren. Sommige studies laten zien dat het zonnestelsel altijd min of meer het-

zelfde zal blijven, oftewel het zonnestelsel is stabiel (Ito & Tanikawa, 2002). Andere studies concluderen dat het mogelijk is dat de banen significant kunnen veranderen, wat kan resulteren in botsingen tussen planeten (Laskar & Gastineau, 2009) of een planeet die het zonnestelsel uitgeschoten wordt (Laskar, 2008). Om de oorsprong en evolutie van dynamische systemen te onderzoeken, hebben we een wiskundig model nodig dat het gedrag van bewegende lichamen beschrijft.

HET N-LICHAMEN PROBLEEM

Newton (1687) definieerde een wiskundig model, genaamd het N-body (N-lichamen) probleem, dat het volgende beschrijft: stel we hebben een dynamisch systeem van N lichamen (met $N = 1, 2, 3, ...$), met elk een massa, positie en snelheid. Wat zijn de posities en snelheden op elke tijd in de toekomst of in het verleden? Als de lichamen elkaar niet zouden voelen, dan zegt Newton's eerste wet dat ze zich blijven voortbewegen in rechte paden met een constante snelheid. Als de lichamen elkaar wel voelen, door de gravitationele aantrekkingskracht, dan worden de lichamen versneld en zullen dan gekromde paden afleggen. Dit gedrag is vastgelegd in Newton's tweede bewegingswet

$$F = ma, \tag{1}$$

met F de kracht, m de massa en a de acceleratie. Toen de appel Newton's hoofd raakte, realiseerde hij zich dat de attractieve kracht die actief is tussen alle lichamen in het heelal met massa, de gravitatiekracht is, ook wel zwaartekracht genoemd. Deze kracht is evenredig aan de massa's van de twee lichamen en omgekeerd evenredig aan het kwadraat van de afstand tussen de twee lichamen

$$F = \frac{GMm}{r^2}. \tag{2}$$

In deze formule staat G voor de gravitationele constante, M en m voor de twee massa's en r voor de afstand. Om iets preciezer te zijn, de massa in vergelijking (1) is de inertiaal massa (massa is traag), terwijl de massa in vergelijking (2) de gravitatie massa is (massa is zwaar). Diverse experimenten hebben echter laten zien dat deze twee gelijk zijn tot op tenminste 13 plaatsen achter de komma (Poisson & Will, 2014). Daarom kunnen we vergelijking (1) en (2) samenvoegen en tegelijkertijd de som nemen over alle lichamen in het systeem

$$\vec{a}_i = G \sum_{j \neq i} \frac{m_j}{r_{ij}^3} \vec{r}_{ij}. \tag{3}$$

Hier staat $\vec{a}_i$ voor de totale acceleratie ervaren door lichaam i, als gevolg van alle andere lichamen met massa m_j op afstanden $r_{ij} = |\vec{r}_j - \vec{r}_i|$. Voor elk object rekenen we de versnelling uit veroorzaakt door de aantrekking van elk ander object. De volgende stap is om deze versnelling te gebruiken om de posities en snelheden te berekenen op een bepaalde tijd in de toekomst. Dit is het lastigste gedeelte.

De versnelling als een functie van tijd is in het algemeen geen simpele functie, maar eerder complex en chaotisch. Behalve voor bepaalde configuraties met $N = 2$ (Newton, 1687) en $N = 3$ (Euler, 1767; Lagrange, 1772), kunnen we banen analytisch oplossen. Voor alle andere gevallen moeten we de versnelling stap voor stap meten. Deze discretisatie van het N-lichamen probleem introduceert een kleine fout in de oplossing, maar maakt het ideaal voor het gebruik van computers.

N-LICHAMEN SIMULATIES

Niet lang na de uitvinding van de computer, werden de eerste N-lichamen simulaties al uitgevoerd. Tussen de eersten om het N-lichamen probleem op te lossen op de computer waren von Hoerner (1960) en Aarseth (1963), met het doel om de banen van sterren te berekenen onder de invloed van hun onderlinge gravitatiekracht.

Het oplossen van een N-lichamen probleem op een computer bestaat uit twee elementen: een integratiemethode en een tijdstap-criterium. De integratie methode bepaalt hoe de nieuwe positie en snelheid worden berekend uit de huidige positie, snelheid en acceleratie. De simpelste integrator is de Euler methode:

$$r\left(t + \Delta t\right) = r\left(t\right) + v\left(t\right) \Delta t, \tag{4a}$$

$$v\left(t + \Delta t\right) = v\left(t\right) + a\left(t\right) \Delta t. \tag{4b}$$

Hier staat t voor de huidige tijd, Δt voor de tijdstap grootte, r voor de positie en v voor de snelheid. Door iteratief de vergelijkingen (3) en (4) uit te voeren, kunnen we de banen van de lichamen berekenen en dus de evolutie van dynamische systemen bestuderen.

De Euler methode is heel simpel en duidelijk, maar niet echt precies. Als we het zouden beschouwen als een eerste orde Taylor-expansie, dan worden de hogere orde termen verwaarloosd, wat zorgt voor een afkappingsfout in de oplossing. Eén manier om dit type fout tegen te gaan, is door een goed tijdstap-criterium te kiezen. In principe moet de tijdstap zo klein mogelijk zijn, maar dit zal het aantal integratiestappen vermeerderen, wat weer tot gevolg heeft dat de CPU tijd stijgt.

Daarom is er een balans tussen de precisie en de snelheid. Er bestaan vele tijdstap-criteria in de literatuur, zoals

$$\Delta t = \mathsf{min}\left\{\frac{r_{ij}}{v_{ij}}\right\},\tag{5}$$

$$\Delta t = \mathsf{min}\left\{\sqrt{\frac{r_{ij}}{a_{ij}}}\right\},\tag{6}$$

Het eerste criterium zorgt ervoor dat het dichtste paar van lichamen niet kan botsen binnen een enkele tijdstap, omdat de grootte van de tijdstap kleiner wordt naarmate lichamen dichterbij elkaar komen. Het tweede criterium is vergelijkbaar, maar werkt ook als alle snelheden nul zijn. Voor verdere details over integratoren, tijdstap-criteria, algoritmes en simulaties verwijs ik de lezer naar Aarseth (2003) en "The Art of Computational Science project"door Hut and Makino[2].

Als de software eenmaal geschreven is, moet het gerund worden op een bepaalde type hardware. Voor simulaties met een paar sterren is een desktop genoeg. De hoogste snelheid kan behaald worden door gebruik te maken van de snelste processors beschikbaar op de markt. Voor grotere dynamische systemen wordt het efficiënt om het N-lichamen probleem in parallel op te lossen. Zoals beschreven in de vorige sectie, als we de acceleratie van één lichaam willen bepalen, moeten we itereren over alle andere lichamen, en dit is een operatie van orde N. Echter, we willen de acceleraties van alle objecten weten, wat nog een factor N oplevert, resulterend in een hoeveelheid arbeid dat schaalt als N^2. Dit werk verdelen over zoveel mogelijk CPU kernen kan leiden tot een significantie versnelling van de simulatie. Het is echter wel zo, dat met meer kernen er ook meer communicatie tussen de kernen aanwezig is, wat weer tot vertraging leidt. Ook hier is er weer een balans die gevonden moet worden (Portegies Zwart et al., 2008).

Omdat het N-lichamen probleem niet exact opgelost kan worden, is er veel tijd gestoken in het vinden van nieuwe, betere algoritmes. Zo zijn er methodes die specifiek ontwikkeld zijn voor een bepaald probleem. Er zijn codes die gaan voor snelheid, zodat ze grote dynamische systemen als sterrenstelsels aankunnen. Andere codes gaan meer voor precisie, bijvoorbeeld voor het berekenen van de banen van planeten. De zoektocht naar de ultieme N-lichamen methode die snel en precies is duurt voort.

---

[2]http://www.artcompsci.org/

CHAOTISCHE DYNAMICA

Een belangrijk concept in dit proefschrift is de groei van een kleine verstoring in het dynamische systeem. Deze verstoring zal in twee contexten behandeld worden. Ten eerste kan de kleine verstoring een numerieke fout zijn, bijvoorbeeld de afkappingsfout. Deze numerieke fout zal groeien door heel het systeem, waardoor de numerieke oplossing divergeert van de echte oplossing. Dit is een probleem dat te allen tijde in de gaten gehouden moet worden door gebruikers van N-lichamen simulaties. De tweede toepassing is wanneer de verstoring fysisch is. We meten hoe twee dichtbijzijnde oplossingen uit elkaar groeien in de tijd. Als dit heel lang duurt vergeleken met de tijdschaal van het probleem, dan is het dynamische systeem stabiel tegen kleine verstoringen. Als blijkt dat een kleine verstoring exponentieel groeit, dan is het systeem instabiel en zal de structuur van het systeem veranderen. Dit is te vergelijken met het voorspellen van het weer; het weer voor morgen kunnen we redelijk precies voorspellen, voor de dag erna is de onzekerheid al groter, en voor het einde van de week weten we het alleen heel globaal. Door het meten van de groei van een kleine verstoring in een dynamisch systeem, kunnen we iets leren over de stabiliteit van het systeem. Dit is belangrijk voor de dynamische interpretatie van waarnemingen die vaak maar een enkele snapshot van het systeem geven.

DIT PROEFSCHRIFT: CHAOTISCHE DYNAMICA IN N-LICHAMEN SYSTEMEN

Dit proefschrift bestaat uit vijf studies aangaande N-lichamen algoritmes, de betrouwbaarheid van N-lichamen statistiek en de oorsprong van chaos in dynamische systemen. Hieronder wordt kort beschreven wat de belangrijkste resultaten zijn.

In hoofdstuk 2 beschrijven we een nieuwe computercode voor het oplossen van het N-lichamen probleem. Deze code is de meest precieze code op aarde, maar ook de langzaamste. Het lost het N-lichamen probleem op met brute kracht en heet daarom Brutus. De reden voor deze hele precieze code is om te proberen het echte antwoord te vinden. Veel systemen laten chaotisch gedrag zien waardoor kleine verstoringen exponentieel toenemen, inclusief numerieke fouten. Op zich is tegen de groei zelf niets te doen, maar door een extreem hoge precisie te gebruiken kunnen we de numerieke fout zo klein maken, dat de verstoring binnen de integratietijd relatief klein blijft. Door de precisie dus

flink op te schroeven kunnen we de echte oplossing vinden en bepalen hoe accuraat conventionele, niet-precieze oplossingen zijn.

Hoofdstuk 3 gaat over een nieuwe N-lichamen code, genaamd Sakura, die voor snelheid gaat. De vraag die we hier stelden is of het N-lichamen probleem opgelost kan worden door alle twee-lichamen problemen in het systeem op te lossen. We kunnen immers een $N = 2$ probleem analytisch oplossen en het superpositie principe geldt voor het berekenen van de acceleraties. We laten zien dat dit inderdaad mogelijk is en leiden een tweede orde, niet-symplectisch algoritme af. Deze code is heel efficiënt op parallele computers door de verdeling van alle Kepler-problemen. Het is voor deze code dat we de Wim Nieuw-poort Award 2013 ontvingen van SURFSara te Amsterdam voor het efficiënte gebruik van hun supercomputer Cartesius.

In hoofdstuk 4 toetsen we de aanname dat resultaten van N-lichamen simulaties statistisch accuraat zijn, ofschoon individuele oplossingen besmet zijn met numerieke ruis. We doen een numeriek experiment waarbij we drie sterren nemen die elkaar gravitationeel beïnvloeden. Ze wisselen energie met elkaar uit, totdat één van de sterren genoeg bewegingsenergie heeft om te ontsnappen. We hebben dan een enkele ster die de ene richting op beweegt en een dubbelster in de tegen-overgestelde richting. We maken een ensemble van initiële condities en integreren deze heel precies met Brutus, en ook met conventionele precisie. Daarna vergelijken we de statistiek van deze twee ensembles van oplossingen. We vinden dat, naargelang de totale energie tot op tenminste 10 procent is behouden, de globale statistiek behouden is onder divergentie van individuele oplossingen. Dit is goed nieuws voor de N-lichamen gemeenschap.

In hoofdstuk 5 gaan we wat dieper in op de gevolgen van de resulta-ten in het vorige hoofdstuk. Hoe komt het dat resultaten met nume-rieke ruis toch de globale statistische distributies behouden? Achter dit resultaat schuilt een fundamentele werking van de gravitatie die doet denken aan quasi-ergodiciteit. Een numerieke oplossing diver-geert continu van het ene oplossingspad naar het andere, en dwaalt door (een groot deel van) de fase-ruimte. Het doet dit zodanig dat het ensemble heel de fase-ruimte bestrijkt en met de juiste frequentie zo-dat statistische distributies behouden blijven. Meer theoretisch werk is nodig om dit fenomeen beter te begrijpen.

We hebben laten zien in de voorgaande hoofdstukken dat chaotische systemen een exponentiële groei van kleine verstoringen laten zien. In hoofdstuk 6 willen we beter begrijpen waar deze groei vandaan komt, hoe relateert het zich tot het fysische systeem? Waarom zijn sommige systemen chaotisch en anderen ordelijk? We stellen een wiskundig

model op dat gebaseerd is op de groei van verstoringen in het twee-lichamen probleem dat weer verstoord wordt door een derde lichaam. We passen dit model toe op komeet Halley en vinden dat de chaos het gevolg is van sterke interacties met Jupiter en Venus. De groei neemt toe op een tijdschaal van ongeveer 300 jaar, zodat over 6000 jaar we niet meer kunnen voorspellen waar de komeet zich zal bevinden.

Het onderzoeksveld van N-lichamen codes en simulaties is volwassen geworden sinds de eerste simulaties in de jaren 1960. Er zijn echter nog veel onderwerpen die onderzocht moeten worden en veel onderwerpen die verbeteringen behoeven. Met de vele waarnemingsmissies gaande, zoals de Gaia en Kepler missies, hebben we behoefte aan codes die sneller en precieser zijn. Hiermee kunnen we het optimale uit de waarnemingen halen en zo meer leren over de werking van dynamische systemen in het heelal.

Curriculum Vitae

"Life is nothing without a little chaos to make it interesting."
– Amelia Atwater-Rhodes

I was born on 12 May 1985 (Mother's day) in Leidschendam, only 15 km away from Leiden. After a few months we moved to Waddinxveen, where I later attended elementary school at the Koningin Beatrix school. My main interests in life were playing football at the local club V. V. Waddinxveen and painting landscapes in the popular style of Bob Ross, which resulted in an exhibition at the local library in 1997.

Following elementary school, I remained in Waddinxveen to attend high school at the Coenecoop College (Atheneum, Natuur & Techniek). It was when I was aged 13 that I watched the TV series "From the Earth to the Moon" on the Apollo missions, and seeing Neil and Buzz bounce around on the Moon triggered the realisation that the world is bigger than my local street. I started reading books on astronomy, including "A Brief History of Time" by Stephen Hawking. This book inspired my final high school research project on "de Oerknal" (the Big Bang) under supervision of mr. Guiking and mw. Persoon. My free time was spent playing rock guitar.

Whereas many hobbies came and went, my passion for astronomy remained and I decided to study Astronomy at Leiden Observatory. My Bachelor research project was in collaboration with Willem de Pous, under supervision of Michiel Hogerheijde and Olja Panić. We simulated the emission of a proto-planetary disk using the code RATRAN. This was my first experience with numerical modelling of astronomical systems.

Having obtained the BSc title, I continued at Leiden to do my Masters in Astronomy. For my minor Master research, under supervision of Jarle Brinchmann, I data-mined the SDSS catalogue for candidate dual AGN. We worked mostly with IDL, which introduced me to scripting and the handy program TOPCAT. I was also student assistant for the first year course Practical Astronomy.

During the annual Science Day at Leiden Observatory, I attended a talk by a newly arrived professor in Computational Astrophysics. As soon as the particles moved over the screen, I instantly knew that

this was the research field for me, and immediately arranged to do my major Master research with Simon Portegies Zwart on stellar dynamics and N-body methods. It was from this research that the N-body code Brutus was developed, which uses arbitrary-precision arithmetic. I was fortunate to have found the right topic for me, with the right supervisor, at the right time.

On completion of my Masters, I left academia to work at the space division of the software company Logica in Rijswijk. Only one week into this role I was given the opportunity to continue my astronomy career at Leiden Observatory, and therefore resigned. Over the last four years I have worked on my PhD under supervision of Simon Portegies Zwart, where we investigated the topics presented in this thesis.

I have attended many conferences and workshops in a variety of places, including Leiden (Modest11), Ameland (NAC12), Noordwijkerhout (NAC14), Nunspeet (NAC15), Dwingeloo (NOVA school), Bonn (Modest14, Aarseth N-body Meetings), Heidelberg (Computational Astrophysics), Berlin (Formation of the Solar System II), Daresbury (Hardware Acceleration), Paris (Stellar Dynamics), Sesto (N-body dynamics), Philadelphia (work visit to Steve McMillan) and New Orleans (Supercomputers). On several occasions I presented my work through a poster and in Bonn and Paris I gave a talk on my work on Brutus and Comet Halley.

Publications

1. *Punctuated Chaos in Dynamical Systems*
   Boekholt, T. C. N., Pelupessy, F. I., Heggie, D. C. and Portegies Zwart, S. F.
   Submitted to MNRAS (Chapter 6)

2. *On the Reliability of N-body Simulations*
   Boekholt, T. C. N. and Portegies Zwart, S. F.
   ComAC, Vol. 2, p. 2, March, 2015 (Chapter 2 and 4)

3. *A Keplerian-based Hamiltonian splitting for gravitational N-body simulations*
   Gonçalves Ferrari, G., **Boekholt, T. C. N.** and Portegies Zwart, S. F.
   MNRAS, 440, 719-730, May, 2014 (Chapter 3)

4. *On the Minimal Accuracy Required for Simulating Self-gravitating Systems by Means of Direct N-body Methods*
   Portegies Zwart, S. F. and **Boekholt, T. C. N.**
   ApJL, 785, L3, April, 2014 (Chapter 5)

5. *The evolution of the Sun's birth cluster and the search for solar siblings in the simulated Gaia catalogue*
   Martınez-Barbosa, C. A., Brown, A. G. A., **Boekholt, T. C. N.**, Portegies Zwart, S. F., Antiche, E. and Antoja T.
   Submitted to MNRAS

Acknowledgements

"We know the age of the Universe to within one percent. For most of my friends I have no idea what their exact age is to within one percent. The Universe is probably my most accurate acquaintance at the moment." – Piet Hut

Four years of PhD research has made me grow both scientifically and as a person. Apart from studying, doing research and attending conferences, I learnt that communication and discussion with colleagues is crucial, both for receiving constructive feedback and for getting new ideas. I have had the privilege to meet many interesting people at Leiden Observatory and at conferences abroad.

First of all, I would like to thank my supervisor Simon Portegies Zwart. Four years ago you made me an offer I could not refuse [3]. Without your guidance and belief in my capabilities, this thesis would not be here. I have enjoyed our collaboration and look forward to many more projects to come.

Secondly, I wish to thank my direct collaborators, some of whom I produced a publication with. A special thanks goes out to Douglas Heggie, I still remember my mobile phone disrupting our first meeting at Modest11. I have learnt a lot from our collaboration since, both as a scientist and as a person and also greatly appreciate your feedback and contributions to this entire thesis.

It is a pleasure to thank Guilherme Gonçalves Ferrari for our many discussions on N-body related topics. I truly learnt a lot from you, and I hope we can investigate many more topics in the future[4].

I would like to thank Steve McMillan for receiving me twice at Drexel University in Philadelphia. I enjoyed these work visits very much and learnt a lot about three-body scattering. Many thanks to Joshua Wall and Michael Brewer for their hospitality during my visits.

It was an honour to be part of the AMUSE team, including Arjen, Inti, Nathan, Evghenii, Niels, Michiko, Masaki, Bernadetta, Steven, Jeroen, Dan, Alex, Carmen, Davide, Stefania, Edwin, Tjibaria, Adrian, Lucie, Silvia, Nora, Thomas, Martha and Maria. The weekly group

---

[3]The Godfather, 1972

[4]Both astronomical and non-astronomical!