# Cosmological N–body Simulations and their Analysis

Joachim Gerhard Stadel

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

University of Washington

2001

Program Authorized to Offer Degree: Astronomy

University of Washington

Graduate School

This is to certify that I have examined this copy of a doctoral dissertation by

Joachim Gerhard Stadel

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Chair of Supervisory Committee:

Thomas Quinn

Reading Committee:

Thomas Quinn

George Lake

Craig Hogan

Date:

University of Washington

Abstract

Cosmological N–body Simulations and their Analysis

by Joachim Gerhard Stadel

Chair of Supervisory Committee

Professor Thomas Quinn
Astronomy

The goal of this work is to show how large cosmological N-body simulations are computed, how structures within such simulations can be identified with observed galaxies, and how the weak gravitational lensing effect of large scale structure introduces changes in the observed brightness of high redshift sources. We describe PKDGRAV, a fully parallel N–body code that can both spatially adapt to large ranges in particle densities, and temporally adapt to large ranges in dynamical timescales. Care has been taken to insure that the code runs efficiently on a variety of parallel architectures. This has lead us to using a non-standard data structure for efficiently calculating the gravitational forces, a variant on the k-D tree, and a new method for treating periodic boundary conditions. A major problem with the interpretation of cosmological N–body simulations has always been identifying the positions and velocities of the "galaxies" in simulations. Observational constraints almost always concern the position and velocities of galaxies, not of the dark matter background. We present our grouping algorithm, SKID, which can identify galaxy halos independent of the environment in which the halo is found, a property that is very important given the large dynamic range in background densities present in cosmological simulations. PKDGRAV has been used to perform many very high resolution cosmological N–body simulations. We give an overview of some of the scientific program that has been enabled by this code. We present the magnification distributions due to weak lensing through all the intervening large

scale structure from an observer to a source at $z = 1$. We show that the distribution is a highly skewed one with a shifted mode implying a slight demagnification bias and an extended tail toward high magnification. We investigate the implications of these results on the observations of high redshift type-Ia supernovae and show how these can be used to provide a new cosmological test.

# TABLE OF CONTENTS

## LIST OF FIGURES

## ACKNOWLEDGMENTS

I have been fascinated with how things work and why from the time of my earliest childhood memories, through school where such interests were called science and math, on through university where professors really put me to task. All along the way I have had the good fortune to have met and been influenced by so many inspiring people. These people have kept the pursuit of science exciting, rewarding and just plain fun. In chronological order then, Graham Avon, my grade 7 teacher who not only inspired many students in the classroom, but also took the time to take science learning outside of it. He also introduced me to computers by taking me to the university and having a "job card" made for me. The rest of that year saw me rushing to the university after dinner and programming Fortran on punch cards, I was hooked. I happened to be lucky enough to have Graham Avon once again as my Grade 12 biology teacher, and he was still just as inspiring then. Doug Potter, first and foremost for being my friend. Doug showed me (and is still showing me) new ways of programming, and working on various projects with him have been some of the most intensely creative experiences of my life. Doug also got me into teaching computers for kids classes with him, while eating up many of our weekends we both look back fondly on this rewarding experience. Austin Gulliver, for the enjoyable summer research experience and along with John Rice for making physics a lot of fun at Brandon University. Bernd Klecker, for giving me the tremendous opportunity of working at the Max Planck Institute (MPE) in Garching Germany for one year after my 3rd year of university. I had many great friends there, Lynn, Gregg, Herbert, Manfred, and Lane. Leaving them and the exciting environment of the Institutes was very difficult after such a wonderful year. Douglas Pickering and Michael Roddy, who gave me a year of pure math at Brandon University for my 4th year and for also giving me a summer research job in mathematics. As half my time at Brandon University was spent in the Mathematics and Computer Science

Department, I should also like to thank Gwyn Richards, Don Eastman, George McMaster, Ellie Crystal, Gerry Dueck for many engaging courses and discussions. Boyd Tolton and Cameron Wicentowich, my physics roommates at Toronto who provided both sanity and insanity in equally necessary doses. Ray Carlberg, for drawing me into Astrophysics and getting me into cosmological N-body simulations. His "project" for me, that of writing an N-body code for parallel computers, kept me busy well after leaving Toronto. George Lake, a.k.a. fearless leader, who brought me together with a dynamic group at the University of Washington, the N-body shop, and provided for a continually stimulating and creative atmosphere within this group. I also thank him for his support and friendship throughout my time in Seattle, both as an employee and as a student. Thomas Quinn, whose office I invaded more than anyone else and who I have been fortunate to have as my supervisor. With Tom it has been possible to turn ideas into reality and much would not have been possible without him. Ben Moore, climbing partner with whom I have had many adventures on rock and snow, not to mention engaging discussions, scientific and otherwise, at the "pub". Along with Ben, I have had the pleasure of working with the postdocs who have passed through the N-body shop, in particular Neal Katz, Fabio Governato, Derek Richardson, Frank van den Bosch, Geraint Lewis, James Wadsley. They assured that things never got dull in our suite. Craig Hogan, whose discussions I always left with a sense of deeper understanding and Chris Stubbs for giving me the opportunity to use a "really big" telescope and giving me a much better understanding of the observational side of astronomy. Thanks to all the great graduate students, in particular Mark Pratt, with whom I have also had some great climbing adventures, and Andrew Becker, who certainly kept things interesting for me.

Finally,

Mom, Dad, Angela, Tonia and Beatrice

Tracy

⋆

Chapter 1

# INTRODUCTION

Historically the N-body problem in astrophysics dealt with the evolution of the Solar System under the influence of gravity. Over the last several decades most of the attention and computing resources in astrophysics has been focused on cosmological N-body simulations with larger and larger numbers of particles. These are indeed some of the largest simulations to be performed in all of computational science. However, modern questions about the stability and dynamical chaos of the Solar System require simulations using many billions of timesteps and sophisticated integration techniques, making this old problem a present day computational challenge as well. Between this big and small $N$ there is a diverse range of astrophysical problems open to N-body simulations, particularly when gas and collisional physics are incorporated into simulation codes. This range is shown in Figure 1, where $T$ is the number of timesteps that a present day (2000) "state-of-the-art" simulation would take. Although the N-body method is being applied to a variety of new astrophysical problems across this $N$-$T$ spectrum, new results continue to be obtained from very large cosmological simulations.

The purpose of cosmological N-body simulations is to study the formation and evolution of large scale structure in the Universe. The isotropy of the microwave background radiation tells us that the early universe was exceptionally smooth, with density fluctuations smaller than $10^{-4}$ to $10^{-5}$ times the average density of the universe. However, the densities within galaxies and clusters of galaxies today are on the order of $10^6$ and $10^2-10^3$ times the average density of the universe respectively. Moreover, the galaxies have arranged themselves into very large coherent structures, the conspicuous walls and voids seen in redshift surveys. This raises the question of how the universe got from its early smooth state, to the present

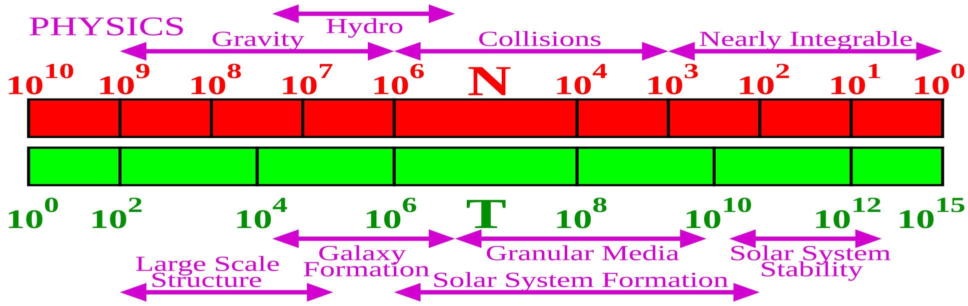


Figure 1.1: The spectrum of astrophysical $N$-body simulations, showing increasing number of particles, $N$, and decreasing number of timesteps, $T$. The product of $N$ and $T$ is roughly proportional to computational work, the combined scales corresponding to present day "state-of-the-art" simulations. Shown are the various ranges of physics that addressed by PKDGRAV. Nearly Integrable: systems where some form of perturbation theory can be used in the calculation, such as protoplanets orbiting the Sun. Collisions: systems where collisions between bodies are most relevant to the creation or evolution of structure. Hydro: systems where gas physics (along with gravity) is used, mainly to address the problem of galaxy formation. Gravity: systems where gravity alone dictates structure. On the bottom of the figure the ranges for some of the scientific questions addressed by our code are shown.

day wealth of structure we observe. It is now clear that gravitational instability causes the small "seed" fluctuations in the early universe to collapse, forming ever larger density contrasts. The statistics of the resulting structures are very sensitive to the power spectrum of the initial fluctuations, to the average density in the universe, and to the nature of the, as yet unknown, dark matter which is the dominant component of matter in the universe. It is because of this that the study of large scale structure formation is a cornerstone of modern cosmology.

Fluctuations on smaller scales quickly grow to the point where there is significant nonlinear evolution and nonlinear coupling between different scales. This limits the usefulness of a purely analytical description of structure formation and large numerical simulations have become the principle theoretical tool in this field. These simulations are among the most expensive in all of science because they must follow the evolution of a fully self gravitating system which forms density contrasts with a dynamic range of over 6 orders of magnitude. Parallel computers have been very effective in this field despite the difficulties presented by the non-uniform, spatially and temporally adaptive data structures that are required. This

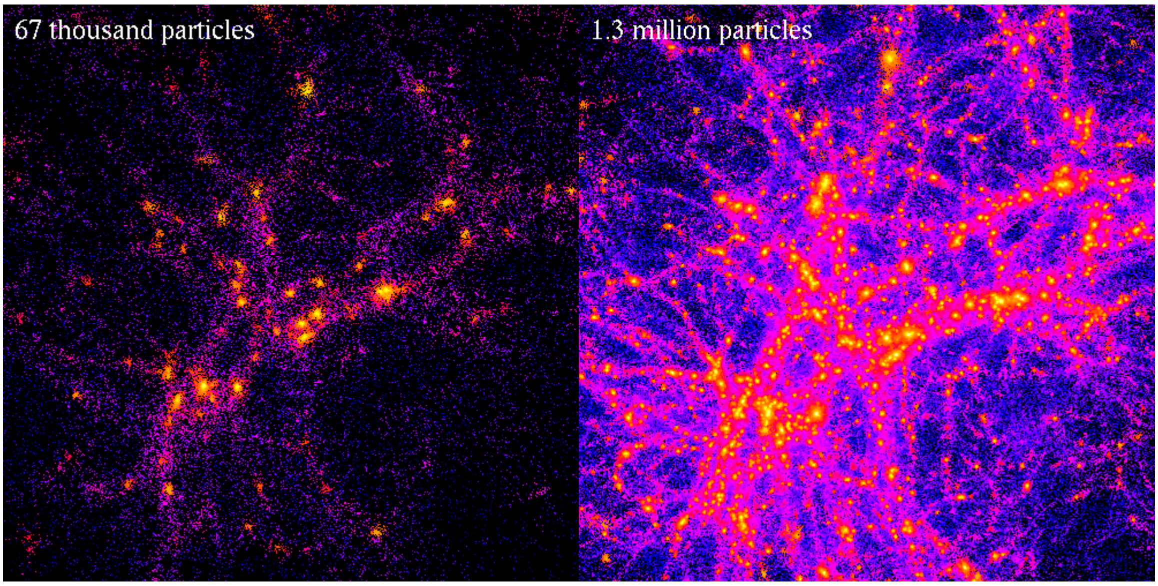


Figure 1.2: The effect of increasing the number of particles in an N-body simulation of the same cosmological system. The system shown is an early stage in the formation of a cluster of galaxies, similar in size to the Virgo Cluster. Note the absence of substructure (structures within structures) in the low resolution simulation, a feature that is quite evident in the high resolution simulation. The abundance of small filaments are also lost in the low resolution simulation.

has vastly expanded the scope of investigations and the level of realism that can be obtained since simulations which would take 10 years on the best workstation can be completed in a few days on a large parallel computer. In figure 1.2 we illustrate the importance of increasing the number of particles, and hence the resolution, for cosmological N-body simulations. In simulating a cluster of galaxies, a large number of particles is required to resolve the dark matter halos of individual galaxies orbiting within the cluster. As the number of particles are increased we resolve substructure halos to smaller and smaller mass scales. On the order of a million particles are required to resolve, within such a cluster, all galaxy halos down to a mass scale comparable to our own galaxy.

Resolving the dark matter halos of galaxies is a critical step in making detailed comparisons between the cosmological N-body simulations and the observable universe, where the luminous matter of galaxies is most prominent. Semi-analytical techniques allow one to make predictions about the type and luminosity of galaxy that could exist within a dark matter halo based on the halo's formation history[1] within the simulation. This further enhances the level of detail with which the theoretical framework can be compared with observations. Thus we can begin to bridge the gap between the dark matter, which dominates the dynamics of structure formation in the universe, and luminous matter in galaxies, which dominates our observational understanding of the universe.

Finally, large volumes of the universe are required for these simulations in order to obtain reasonable statistics and correctly model the larger dynamical environment of galaxy halos. The combined requirements of large volumes and high resolution explains the large computational demands of cosmological N-body simulations. This, in turn, motivates the development of improved numerical algorithms as well as the use of parallel computing for performing such simulations. This topic will be the focus of chapters 2 and 3, where we will present the details of PKDGRAV, which is a high performance parallel N-body code developed by the author. Chapter 4 presents SKID, a group finding algorithm developed by the author which is ideally suited to finding the galaxy halos within cosmological N-body simulations. Chapter 5 outlines some of the scientific program that has been undertaken using PKDGRAV and SKID. The final chapter examines the weak gravitational lensing effect of the cosmic structure on very distant sources, which provides a way to directly "observe" the dark matter in the universe. These effects are subtle and detailed numerical models of the distribution of matter in the universe are required to accurately characterize them. Detecting the weak lensing effect by large scale structure is still an observational challenge, but is likely to develop into one of the most powerful tools for investigating the large scale cosmic structure.

[1] This is more commonly referred to as the "merger history" of the halo.

Chapter 2

# PKDGRAV: A SCALABLE PARALLEL HIERARCHICAL TREECODE FOR COSMOLOGICAL N–BODY SIMULATIONS

## *2.1 Introduction*

The present day distribution of galaxies and galaxy clusters is a critical constraint on any theory of cosmology. However, the formation of such structures is a highly nonlinear process that is, in general, not analytically tractable. Therefore, the only way to compare the consequences of a given theory with the observed distribution of galaxies is via a numerical simulation of structure formation. It is widely accepted that the predominant physics driving the formation of structure is the gravitational collapse of initially small fluctuations in the mass density. In order to follow the dynamics of this collapse, one must solve the collisionless Boltzmann equation with the appropriate initial conditions. This is most commonly done using an N–body simulation: the phase space is discretized into particles which then follow the phase space flow. The discretization introduces unphysical artifacts into the simulation which in general can only be reduced by using a large number of particles. This requires significant computing resources both in terms of floating point operations and memory capacity, naturally suggesting the use of massively parallel computers.

In this paper we describe PKDGRAV, a fully parallel N–body code that can both spatially adapt to large ranges in particle densities, and temporally adapt to large ranges in dynamical timescales. Care has been taken to insure that the code runs efficiently on a variety of parallel architectures. This has lead us to using a non-standard data structure for efficiently calculating the gravitational forces, a variant on the k-D tree (Bentley 79), and a new method for treating periodic boundary conditions.

### 2.2 Calculating Gravity

In order to evolve a self gravitating system, realized as a large number of particles, one must determine the gravitational acceleration on a given particle due to the mass of all other particles. The simple approach of calculating all pairwise interactions between the particles becomes a prohibitive expense, scaling as $\mathcal{O}(N^2)$, as the number of particles, $N$ is increased. Two basic approaches have been used to reduce this scaling, thus allowing very large N-body simulations to be performed. The first of these approaches rely on the speed of FFT algorithms and includes the PM (particle-mesh), $P^3M$ (particle-particle particle-mesh), and $AP^3M$ (adaptive $P^3M$) methods. The second class of methods uses multipole expansions to approximate the gravitational effect of more distant mass. These methods use a hierarchical representation of the mass implemented as some form of tree structure (Barnes & Hut 86). These *meshcodes* and *treecodes* reduce the cost of calculating the mutual gravitational force to $\mathcal{O}(N \log N)$ or even $\mathcal{O}(N)$ in certain cases. Since PKDGRAV is a treecode, we will describe in detail how it calculates the gravitational force using multipole expansions and a hierarchical (tree) representation of the mass.

#### *2.2.1 Multipole Expansions*

The mathematical machinery of spherical harmonics allows the simple expression of multipole expansions and their transformations to arbitrary order (Greengard and Rokhlin 87). However elegant, at low orders of expansion the use of Cartesian multipoles is both simpler and considerably faster since the calculation of trigonometric functions is avoided. For this reason most treecodes have used Cartesian multipole expansions taken to second (quadrupole) order. PKDGRAV extends the multipole expansion to fourth (hexadecapole) order which can be almost twice as efficient at the same accuracy as quadrupole order. At the same accuracy, a mass distribution which is well approximated by a fourth order multipole will need to be subdivided into several smaller second order multipoles, for which the stream of floating point calculations requires more square root calculations and suffers from not being as tightly pipelined for the present generation of microprocessors. Furthermore, for parallel computation, the use of higher order multipole expansions increases the

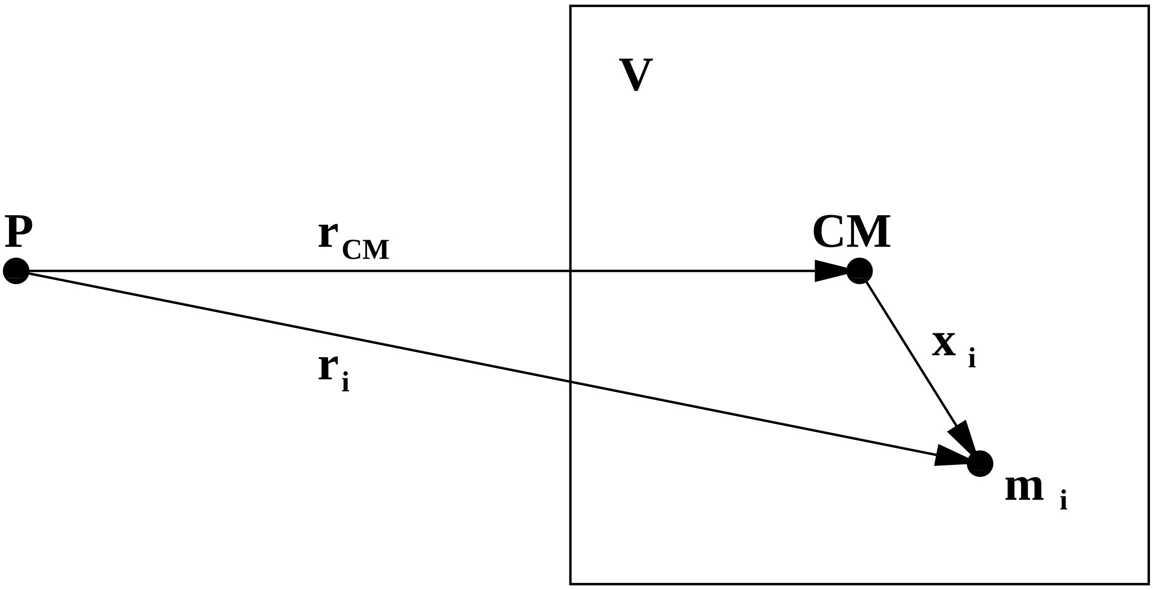


Figure 2.1: Orientation of vectors for the multipole moment approximation.

computation to communication ratio for parts of the mass distribution not residing in local memory.

Expressing higher order multipole expansions in Cartesian coordinates is notationally cumbersome, as the order of the expansion is increased, the rank of the tensors required also increases (see Salmon 94). However, by introducing a new notation which takes advantage of the fact that all the tensors we will require are symmetric under permutation of their indices, we can cleanly express multipole calculations in Cartesian coordinates taken to arbitrary order. In what is to follow we use an underlined index to a tensor to indicate the *number* of indices it would carry in traditional tensor notation. For example, given the vector $\mathbf{x}$ we could write, $x^{\underline{2}} \equiv x^{ij} \equiv x^{ji} \equiv x^i x^j$.

The gravitational potential at a point in space due to a mass distribution over the volume $V$ is given by,

$$\Psi = \int_V \gamma(|\mathbf{r}|)\rho(\mathbf{r})d\mathbf{r}, \tag{2.1}$$

where $\gamma(|\mathbf{r}|)$ denotes the Green's function; for unsoftened gravity this is given by $\gamma(r) = -1/r$ (setting the gravitational constant $G = 1$). For a distribution of particles the mass density distribution is given by,

$$\rho(\mathbf{r}) = \sum_{i \in V} \delta(\mathbf{r}_i - \mathbf{r}) m_i, \tag{2.2}$$

where $m_i$ and $\mathbf{r}_i$ are the mass and position of particle $i$. Substituting this mass density we obtain,

$$\Psi = \sum_{i \in V} m_i \gamma(|\mathbf{r}_i|) = \sum_{i \in V} m_i \gamma(|\mathbf{r}_{\mathbf{cm}} + \mathbf{x}_i|), \tag{2.3}$$

where the definition of the vectors $\mathbf{r}_{\mathbf{cm}}$ and $\mathbf{x}_i$ is shown in figure 2.2. A formal Taylor expansion of the Green's function in this expression gives,

$$\Psi = \sum_{i \in V} m_i \sum_{n=0}^{\infty} \frac{1}{n!} \left[\partial_{\underline{n}} \gamma(r_{\mathbf{cm}})\right] x_i^{\underline{n}}, \tag{2.4}$$

where we have used the shorthand for partial derivatives, $\partial_{\underline{n}} \equiv \partial / \partial r^{\underline{n}}$, which is an $n$-th order derivative in this new *rank notation*. We have also dispensed with the use of magnitude symbols and bold face for vectors where the meaning is clear. The raised and lowered indices indicate the Einstein summation convention, just as in the traditional tensor calculus. We now define the multipole moment,

$$M^{\underline{n}} = \sum_{i \in V} m_i x_i^{\underline{n}}, \tag{2.5}$$

where $M^{\underline{0}}$ is the monopole, $M^{\underline{1}} \equiv M^i$ the dipole, $M^{\underline{2}} \equiv M^{ij}$ the quadrupole, $M^{\underline{3}} \equiv M^{ijk}$ the octopole, and $M^{\underline{4}} \equiv M^{ijkl}$ the hexadecapole. Note that the dipole, $M^{\underline{1}} = 0$ when taken about the center of mass. With this definition we can now write,

$$\Psi = \sum_{n=0}^{\infty} \frac{1}{n!} \left[\partial_{\underline{n}} \gamma(r_{\mathbf{cm}})\right] M^{\underline{n}}. \tag{2.6}$$

What remains is the evaluation of the $n$-th order derivative of the Green's function. For this the following useful recursion relation is needed,

$$\gamma(r) \equiv \gamma_0 = -\frac{1}{r}; \qquad \gamma_{m+1} = -\frac{(2m+1)}{r^2} \gamma_m \tag{2.7}$$

for which we can write the 1st order derivative,

$$\partial^{\underline{1}} \gamma_m = \gamma_{m+1} r^{\underline{1}}. \tag{2.8}$$

We also require the product rule for derivatives,

$$\partial^{\underline{n}}(A\,B) = \sum_{m=0}^{n} \left(\partial^{(\underline{m}} A\right)\left(\partial^{\underline{n-m})} B\right), \tag{2.9}$$

where the braces about the indices indicates all *unique* permutations of the enclosed indices. For example,

$$\partial^{(\underline{2}} A\,\partial^{\underline{1})} B \equiv \partial^{(ij} A\,\partial^{k)} B \equiv \partial^{ij} A\,\partial^{k} B + \partial^{jk} A\,\partial^{i} B + \partial^{ik} A\,\partial^{j} B.$$

Equations 2.7, 2.8 and 2.9 lead us to the more general expression,

$$\partial^{\underline{n}}\gamma_k = \sum_{m=0}^{\lfloor \frac{n}{2} \rfloor} \frac{1}{m!}\gamma_{k+n-m}\delta^{(\underline{2m}} r^{\underline{n-2m})}, \tag{2.10}$$

where we have defined $\delta^{\underline{2m}}$ as the product of $m$ delta symbols. Finally, substituting equation 2.10 into equation 2.6, we obtain the multipole expansion of $\Psi$ to arbitrary order (letting $r \equiv r_{\mathbf{cm}}$),

$$\Psi = \sum_{n=0}^{\infty} \frac{1}{n!} \sum_{m=0}^{\lfloor \frac{n}{2} \rfloor} \frac{1}{m!}\gamma_{n-m}\delta_{(\underline{2m}} r_{\underline{n-2m})} M^{\underline{n}}. \tag{2.11}$$

This equation can be further simplified, greatly reducing the number of terms which need to be evaluated, if the Green's function is harmonic, satisfying the Laplace equation,

$$\delta^{\underline{2}}\partial_{\underline{n+2}}\,\gamma(r) = 0; \qquad\qquad r \neq 0. \tag{2.12}$$

If we substitute the moment $Q$,

$$Q^{\underline{n}} = M^{\underline{n}} + \delta^{\underline{2}} C^{\underline{n-2}}, \tag{2.13}$$

for $M$ in equation 2.6 we see that $C^{\underline{n-2}}$ can be *any* tensor we wish, since the second term will cancel when contracted with the derivatives of the Green's function by equation 2.12. This freedom in the choice of $C$ allows us to make $Q$ completely trace free over any two indices,

$$\delta_{\underline{2}} Q^{\underline{n}} = 0. \tag{2.14}$$

The required trace–free tensor $Q$ is called the *reduced multipole moment* and is given by,

$$Q^{\underline{n}} = \sum_{m=0}^{\lfloor \frac{n}{2} \rfloor} (-1)^m \frac{(2n-2m-1)!!}{m!(2n-1)!!}\delta^{(\underline{2m}} T_m^{\underline{n-2m})}(M^{\underline{n}}), \tag{2.15}$$

where $T_m^{\underline{n-2m}}(M^{\underline{n}})$ is the $m$-fold trace of $M^{\underline{n}}$,

$$T_m^{\underline{n-2m}}(M^{\underline{n}}) = \delta_{\underline{2m}} M^{\underline{2m}\,\underline{n-2m}}. \tag{2.16}$$

We point out that because $Q^{\underline{n}}$ is trace–free, the number of independent elements in this tensor has been reduced to $n+1$ from the $(n+1)(n+2)/2$ elements required for $M^{\underline{n}}$. It should also come as no surprise that this is exactly the number of independent elements of a spherical harmonic of the same order. This reduction in the required storage of multipole moments becomes quite significant at higher orders. It is useful to introduce the following definition. Let,

$$Q_n^{\underline{m}} = \frac{1}{(n-m)!} r_{\underline{n-m}} Q^{\underline{m}\,\underline{n-m}}\,; \qquad (m \leq n) \tag{2.17}$$

then,

$$\partial^{\underline{1}}\, Q_n^{\underline{m}} = Q_n^{\underline{m+1}}\,; \qquad (m+1 \leq n), \tag{2.18}$$

and,

$$Q_n^{\underline{m}} = \frac{1}{n-m}\, r_{\underline{1}} Q_n^{\underline{m+1}}. \tag{2.19}$$

$Q_n^{\underline{m}}$ also remains completely trace–free,

$$\delta_{\underline{2k}} Q_n^{\underline{m}} = 0\,; \qquad (2k \leq m). \tag{2.20}$$

Equations 2.17-2.20 now allow us to simplify equation 2.11 to,

$$\Psi = \sum_{n=0}^{\infty} \gamma_n Q_n^{\underline{0}}. \tag{2.21}$$

The expansion for the gravitational acceleration is also easily obtained using equations 2.8 and 2.18,

$$a^{\underline{1}} = \partial^{\underline{1}}\Psi = \sum_{n=0}^{\infty} \left[\gamma_{n+1} r^{\underline{1}} Q_n^{\underline{0}} + \gamma_n Q_n^{\underline{1}}\right]. \tag{2.22}$$

The above are still formal multipole expansions, meaning that we need to impose extra conditions to ensure that they are valid, convergent and have a tolerable error when truncated at some order $p$. We require that,

$$\max_{i \in V}(x_i + 2h_i) < r, \tag{2.23}$$

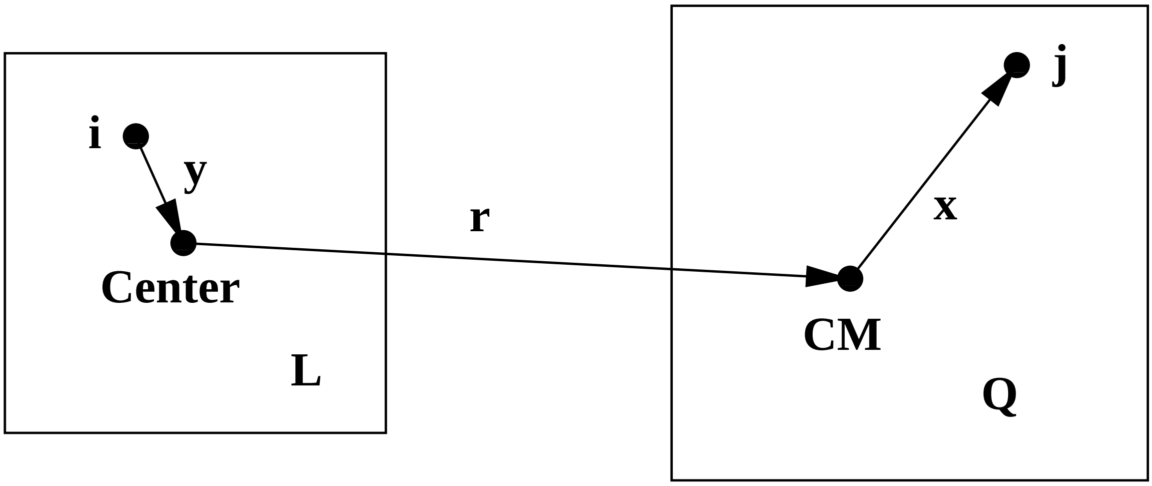


Figure 2.2: Orientation of vectors for the local expansion of the multipole moment approximation. Note that the center of the local expansion is somewhat arbitrary, it is best to use the center of the bounding sphere of the particles in the local volume, but one could also use the center of mass of these particles.

where $2h_i$ is the radius beyond which the potential of particle $i$ is *unsoftened* $(1/r)$. This condition ensures that the expansion is both valid and convergent. PKDGRAV uses the above expansions to fourth order, $n \leq p = 4$, for which tolerable force errors are obtained for most applications by requiring that,

$$\max_{i \in V}(x_i) < \theta\, r, \tag{2.24}$$

with $\theta$ ranging from about 0.5 to 0.8. We will examine the error properties of the code in more detail in section 2.5. These conditions imply that when determining the contributions to the force on a given particle, larger and larger volumes can be treated as a single multipole, as we consider more and more distant parts of the mass distribution. Each of these contributions will be accumulated with comparable relative force accuracy, $|\delta \mathbf{a}_V|/|\mathbf{a}_V|$.

We now pursue the prospect of approximating the local gravitational field due to one, or many, multipoles which represent more distant parts of the mass distribution. This near field approximation is the dual to the approximation we developed above and is often referred to as the *local expansion.* If there are several particles in a very localized region of space we can combine all sufficiently far field multipoles into a *single* local expansion which is then applied to each of these particles. Since many hundreds of multipoles required by each of the particles would be identical, approximating their combined effect in the local region can lead to considerable savings in computation. In fact, this local expansion is the basis for calculating gravity in $\mathcal{O}(N)$ time, an improvement in the asymptotic scaling from $\mathcal{O}(N \log N)$ where only the far field expansion in employed (see section 2.2.2 for further discussion). Developing the local expansion is reasonably straight forward using the rank notation. We begin again by writing a formal Taylor expansion, this time about some local center. The local center we choose is somewhat arbitrary, as long as it is "close" to all the particles in the local volume. To minimize the maximum error due to this expansion, it is best to use the center of the bounding sphere of the particles in the local volume, but for symmetry reasons the center of mass of these particles may be more appropriate. The Taylor expansion about the local center is given by,

$$
\begin{aligned}
\Psi(\mathbf{r}+\mathbf{y}) &= \sum_{l=0}^{\infty} \frac{1}{l!} L^{\underline{l}} y_{\underline{l}} && (2.25)\\
a^{\underline{1}}(\mathbf{r}+\mathbf{y}) &= \sum_{l=0}^{\infty} \frac{1}{l!} L^{\underline{l+1}} y_{\underline{l}} && (2.26)
\end{aligned}
$$

where,

$$
L^{\underline{l}} = \partial^{\underline{l}} \Psi(\mathbf{r}) \tag{2.27}
$$

What we are seeking is a way to transform a far field multipole, $Q$, into the terms of the local expansion, $L$. Linearity allows us to transform each sufficiently distant multipole and add it to $L$. We begin by substituting equation 2.21, which has been truncated to a maximum order of $p$.

$$
\begin{aligned}
L^{\underline{l}} &= \sum_{n=0}^{p} \partial^{\underline{l}} \left(\gamma_n Q_n^{\underline{0}}\right) \\
&= \sum_{n=0}^{p} \sum_{k=0}^{l} \left(\partial^{(\underline{l-k}} \gamma_n\right) \left(\partial^{\underline{k})} Q_n^{\underline{0}}\right) \\
&= \sum_{n=0}^{p} \sum_{k=0}^{\min l,n} \left(\partial^{(\underline{l-k}} \gamma_n\right) Q_n^{\underline{k})} \\
&= \sum_{k=0}^{l} \sum_{n=k}^{p} \left(\partial^{(\underline{l-k}} \gamma_n\right) Q_n^{\underline{k})},
\end{aligned}
$$

where we have used the chain rule equation 2.9 and equation 2.18, taking care to limit the sum over $k$ appropriately, and finally switching the order of the two sums. We now apply equation 2.10 to arrive at,

$$L^{\underline{l}} = \sum_{k=0}^{l} \sum_{n=k}^{p} \sum_{m=0}^{\lfloor \frac{l-k}{2} \rfloor} \frac{1}{m!} \gamma_{n-m+l-k} \, \delta^{(\underline{2m}} \, r^{\underline{l-k-2m}} \, Q_{n}^{\underline{k})}. \tag{2.28}$$

Although this equation is somewhat complicated, with some reordering of summations and relabeling of indices we can obtain a more elegant and efficient way of calculating $L^{\underline{l}}$. This is given by the following sequence of equations,

$$R_{l}^{\underline{k}} = \sum_{n=k}^{p} \gamma_{n+l-k} Q_{n}^{\underline{k}} \tag{2.29}$$

$$S_{l}^{\underline{k}} = \sum_{m=0}^{\lfloor \frac{l-k}{2} \rfloor} \frac{1}{m!} \delta^{(\underline{2m}} R_{l-m}^{\underline{k-2m})} \tag{2.30}$$

$$L^{\underline{l}} = \sum_{k=0}^{l} r^{(\underline{l-k}} S_{l}^{\underline{l})}, \tag{2.31}$$

where of particular note is the fact that $L$ is trace–free (as is $R$), resulting in considerable reduction in the number of operations needed in this transformation. At this stage our notation has carried us so far from actual implementation that the use of machine generated code via algebraic engines becomes essential in ensuring that such expressions are correctly realized. Evaluating these equations to fourth order in $l$ and $n$ is only about 4 times as expensive as evaluating both of equations 2.21 and 2.22. This means that using the local expansion becomes profitable when it can be applied to more than 4 particles. Once again, to be considered a valid and convergent expansion we need to impose the following condition,

$$\max_{i \in L}(y_i + 2h_i) + \max_{j \in V}(x_j + 2h_j) < r. \tag{2.32}$$

To ensure tolerable errors we can impose the criterion,

$$\max_{i \in L}(y_i) + \max_{j \in V}(x_j) < \theta \, r, \tag{2.33}$$

similar to the way we did before. We note that these criteria are symmetric, in the sense that it does not matter which volume is the local one and which is the far field. This property can be further exploited to make all expansions symmetric with respect to the two volumes, leading to perfect momentum conservation (see Dehnen (2000) who has developed this technique to second order).

### *2.2.2 Tree Structure*

The central data structure in PKDGRAV is a tree structure which forms the hierarchical representation of the mass distribution. Each node of this tree represents a rectangular subvolume of the simulation volume (a cell), containing the mass, center–of–mass, and higher moments of the mass distribution in the enclosed volume. The *root–cell* represents the entire simulation volume, and as we proceed deeper down this tree we get an ever finer representation of the mass distribution. Each cell is divided into subcells until we reach the *leaves* of the tree which contain only a few particles, sometimes only one. Although all treecodes use such a structure, unlike the more traditional *oct–tree* used in the Barnes–Hut algorithm, we use a binary tree.

The primary motivation for choosing a binary tree data structure over the classical oct–tree (Barnes and Hut 86) is that it can be naturally extended to a parallel, distributed tree structure over an arbitrary number of processors in a very natural way. We have found that efficiency is greatly improved if the gravity data structure matches the domain decomposition data structure. The domain decomposition should form some tiling of gravity tree cells instead of just dividing up the particles. To achieve this we simply subdivide the simulation volume recursively until we have subvolumes for every processor of approximately equal computational work. Then each processor continues from there to build its own local part of the tree. For the oct–tree this is not quite as straight forward, but is accomplished by labeling the tree nodes with hash keys in a way that creates a linear ordering that retains the property of spatial locality for the cells (Salmon, Winckelmans and Warren 94). This ordering is again subdivided into pieces of approximately equal work for each processor. This technique is also quite efficient, although it lacks the convenient property of having an entire subtree for each processor. We will discuss the details of domain decomposition and load balancing further in section 2.4.3.

PKDGRAV gets its name from the type of binary tree which it originally used, the *k–D*

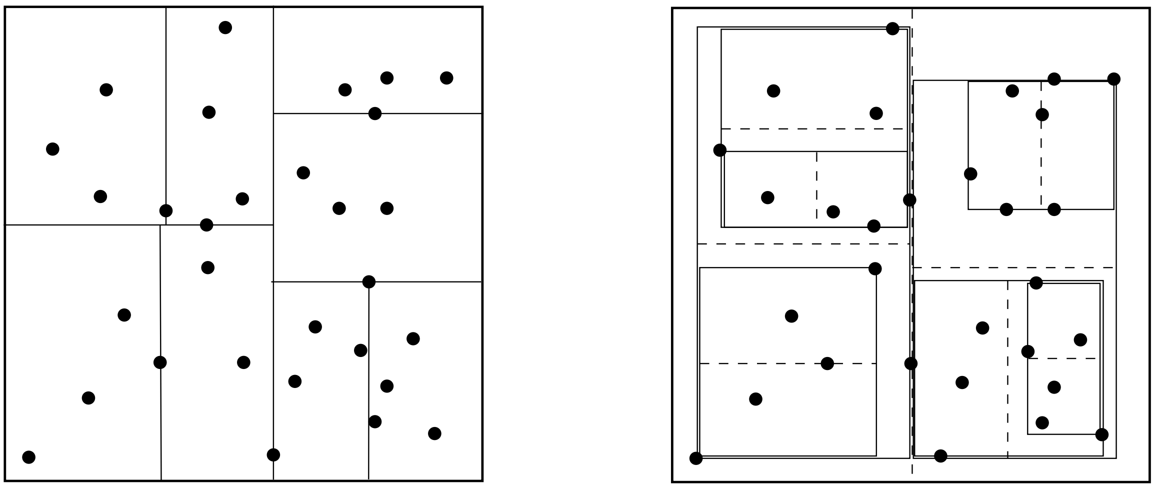

Figure 2.3: A comparison of the balanced k–D tree and the spatial binary tree used by PKDGRAV. On the left the k–D tree is shown, constructed with a maximum of 4 particles per leaf cell, $b = 4$. On the right we show the equivalent spatial binary tree, where we have squeezed the bounds of the cells before bisecting them at each level. The bisectors are shown with the dashed lines. Note that the depth of the spatial binary tree is not the same for all particles, as is the case for the k–D tree. The k–D tree also assures that we get either $b$ or $b - 1$ particles per leaf cell, which is also not the case for the spatial binary tree. For such a uniform distribution (and such small numbers of particles), the differences between the two structures are not as pronounced as they would be for a more highly clustered distribution, where the the cells of the spatial binary tree maintain a much higher degree of spatial locality than for the k–D tree.

*tree*, where k–D stands for k–Dimensional. In our case k always equals 3. To build a k-D tree, we start from the root-cell and recursively bisect the cells through their longest axis, so that an equal number of particles lie in each sub-volume, and that higher moments of cells are kept to a minimum (see Figure 2.3). This bisection is accomplished using Hoare's median finding algorithm, which is an $\mathcal{O}(N)$ average time operation per level of the tree, making the tree building process an $\mathcal{O}(N \log_2 N)$ operation. The depth of the tree is chosen so that we end up with at most some small number of particles in the leaf cells (*buckets*). We have found eight particles per bucket, $b = 8$, to be near optimal for parallel gravity calculation. The k–D tree has been considered "optimal" for nearest neighbor finding, multi–dimensional key searches and other applications. The benefits of a balanced tree include the fact it has either $b$ or $b - 1$ particles per leaf node. The availability of fast median finding algorithms allow for very efficient tree construction. Pointers are unnecessary since each node in the tree can be indexed so that the finding of children, parent and sibling nodes are simple bit-shift operations. This and the use of buckets, by which only $2\lceil N/8 \rceil$ nodes are required, makes the tree structure memory-efficient. However, for gravity calculations, the balanced k–D tree can behave pathologically for certain particle distributions. (See Anderson 99, for a discussion of the details.) In particular, distributions such as those used in "renormalized volumes" (Katz *et al.* 94) can lead both to systematic force errors and $\mathcal{O}(N^2)$ performance.

To solve this problem we modified the tree construction such that instead of bisecting cells to obtain an equal numbers of particles in each subvolume, we spatially bisect the bounding box containing all the particles in the volume. Dividing the cells in this fashion reduces the higher order multipole moments resulting in more accurate forces, and does not suffer from the pathology described above. PKDGRAV now uses this *spatial binary tree* exclusively for both the gravity calculation and nearest neighbor searching, although for nearest neighbor searching the balanced k–D tree seems to work well for all particle distributions we have encountered. PKDGRAV has successfully used both tree structures, spatial binary trees for gravity and balanced k–D trees for nearest neighbor searching. However, we have recently found that the spatial binary tree nearly matches the balanced k–D tree in nearest neighbor searching performance. In our opinion, this makes the use of the balanced k–D tree obsolete, since the constant switching between the two tree structures during a calculation incurs significant overheads, particularly when implementing Smoothed Particle Hydrodynamics (SPH).

Building the tree is accomplished in two phases. In the first phase we build the basic tree structure by using what is essentially a *quick–sort* algorithm, the only difference being that we partition the particles about the bisector of the longest axis of the parent cell. This $\mathcal{O}(N \log_2 N)$ average time operation produces a spatially local order to the particles in memory, which greatly improves performance of gravity calculation and nearest neighbor searching. Before choosing a bisector of a cell, we first calculate a new tight bounding box for the particles within the cell (see Figure 2.3). This added operation, which we call *squeezing* the cells, produces a significantly better tree structure resulting in a larger average number of particles per bucket (typically about $0.65b$) and fewer cells for a given $N$. Although redundant cells, those where one of the subcells contains no particles, can be removed, this squeezing automatically ensures that each cell always has two non–empty subcells, a property that makes tree traversal algorithms simpler and more efficient. Since particle structures can be quite large, from 80 to over 160 bytes depending on the application, the continuous re-shuffling of these large structures during this phase completely dominates the tree building cost. For this reason, we shuffle only pointers to the actual particles, following this by $\mathcal{O}(N)$ transpositions of the particle structures in memory. Finally, we note that setting $b = 16$, resulting roughly 10 particles per bucket on average, is close to optimal for gravity when both compute speed and memory usage are considered.

This previous, *top–down*, phase has constructed the basic tree structure, determined the bounds of cells, and ordered the particles in memory. What remains is a second, *bottom–up*, phase to calculate cell quantities, such as the center–of–mass (CM) and reduced multipole moments. These quantities can be obtained, for all but the leaf cells, from the two child cells, and for the leaf cells the calculation is trivial from its particles. However, calculating the reduced multipole about a new CM from a given reduced multipole presents a bit more difficulty. For regular multipoles, $M^{\underline{n}}$, this is accomplished using the *parallel axis theorem*, extended to arbitrary order,

$$M^{\underline{n}}_{\mathbf{d}} = \sum_{m=0}^{n} M^{(\underline{n-m}} d^{\underline{m})}, \tag{2.34}$$

where the vector $\mathbf{d} = \mathbf{r_{cm}} - \mathbf{r_{cm}^{new}}$, the distance from the new CM to the original CM. Let us define the trace free version of the tensor $d^{\underline{m}}$ as $f^{\underline{m}}$. This is identical to equation 2.15 with $d$ and $f$ substituted for $M$ and $Q$ respectively. We do not at present[1] have a general expression like the above for reduced multipoles, and instead provide the equations up to fourth order.

$$\begin{aligned}
Q_{\mathbf{f}}^{ij} &= Q^{ij} + Qf^{ij} \\
Q_{\mathbf{f}}^{ijk} &= Q^{ijk} + Qf^{ijk} + Q^{(ij}f^{k)} - \tfrac{2}{5}\delta^{(ij}Q_m^{k)}f^m \\
Q_{\mathbf{f}}^{ijkl} &= Q^{ijkl} + Qf^{ijkl} + Q^{(ijk}f^{l)} + Q^{(ij}f^{kl)} \\
&\quad -\tfrac{2}{7}\left(\delta^{(ij}Q_m^{kl)}f^m + \delta^{(ij}\delta_m^k Q_n^{l)}f^{mn} - \tfrac{1}{5}\delta^{(ij}\delta^{kl)}Q_{mn}f^{mn}\right)
\end{aligned} \tag{2.35}$$

These equations have been presented in traditional tensor notation and the $Q$ used here should not to be confused with the definition given by equation 2.17. These equations allow us to efficiently calculate all cell quantities in a bottom–up fashion, a procedure which takes $\mathcal{O}(N)$ time. In fact, the only $\mathcal{O}(N \log N)$ operation required for tree building is the the sorting procedure itself, and $\mathcal{O}(N)$ sorting algorithms are known where the underlying distribution of sort keys is Riemann integrable (Knuth 98, Section 3.3). This would suggest that gravity calculation (including tree building) can be accomplished in $\mathcal{O}(N)$ time for underlying particle distributions which become continuous (non–fractal) below some scale. Unfortunately, the cold dark matter model (CDM) for structure formation in the Universe produces fractal–like distributions to well below the scales that have been investigated with N–body simulations.

### *2.2.3 Tree Walking*

In this section we describe the manner in which the interactions from both nearby particles and more distant multipoles are determined for each particle, using the tree structure described above. This top–down procedure, called *tree walking*, produces two interaction lists, a particle–particle list (p–list) and a particle–cell list (c–list), which can subsequently be evaluated to determine the gravitational force on the given particle. The traditional tree code determines these interaction lists for each individual particle, an operation that

[1] Although we are quite confidant that such an expression can be obtained.

has become the dominating cost with the rapid evolution of floating point performance in microprocessors. PKDGRAV amortizes the tree walk, taking advantage of the fact that the interaction lists for nearby particles would be almost identical. Instead of applying the interaction lists to a single particle, PKDGRAV applies it to all the particles in a given bucket, or leaf cell. Recently, it has been found that even this level of amortization is insufficient to keep pace with the continual improvements in floating point performance. This has led us to develop a new algorithm which once again ensures that floating point operations dominate the gravity calculation.

In the tree building phase, PKDGRAV assigns to each cell of the tree an *opening radius* about its center–of–mass. This is defined as,

$$r_{open} = \frac{2}{\sqrt{3}} \frac{r_{max}}{\theta}, \tag{2.36}$$

where $r_{max}$ is the distance from the center–of–mass to the most distant corner of the cell, which closely approximates the distance to the furthest particle in the cell. This equation is simply a restatement of equation 2.24. The geometrical factor is added to make this $\theta$ more closely match the traditional $\theta$ opening angle of the Barnes–Hut treecode. However, we note that because the cell shapes in PKDGRAV are not perfectly square, and because of the use of a fourth order expansion, this $\theta$ must still be interpreted differently when considering the accuracy of the code.

The basic tree walking algorithm for collecting the interactions for a bucket of particles begins from the root–cell and proceeds in a top–down manner. In this recursive procedure, if a cell's opening ball (given by $r_{open}$ and its CM) intersects the bucket then we cannot accept its multipoles in the expansion and need to *open* this cell, considering each of its child cells in turn. If the opening ball of a cell does not intersect the bucket, meaning it is sufficiently distant, then we place its multipole moments onto the interaction list (c–list) for this bucket. This is shown in Figure 2.4. When we reach the leaves, or buckets, of the tree and are forced to open them, we simply place all the particles in such buckets onto the p–list. Once we have completed the construction of the particle and cell interaction lists in this way, we evaluate them for each of the particles of the bucket in question. This last operation accounts for the bulk of the floating point operations required for calculating gravity. For

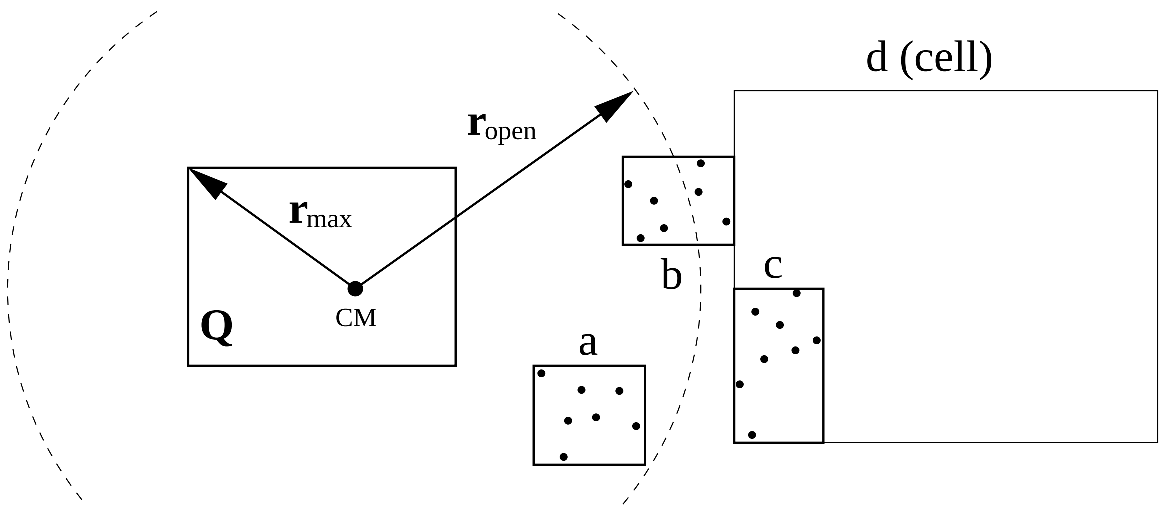


Figure 2.4: The multipole $Q$ of the cell on the left is placed on the interaction list of a bucket, if this given bucket lies outside of the opening radius, $r_{open}$, of the cell. The only bucket of particles which satisfies this criterion is bucket $c$. The buckets $a$ and $b$, which are intersected by (or lie inside) this radius, must reject this multipole and *open* the cell, considering further the two subcells for inclusion in their interaction lists. The opening radius of a cell is given by some constant factor times $r_{max}$, the distance from the center–of–mass to the most distant corner of the cell. Some of the particles in bucket $b$ would have accepted multipole $Q$ in the traditional treecode, but the benefits of amortizing the tree walk over all particles of a bucket far outweighs such a consideration. Finally, note that multipole $Q$ will be acceptable to all buckets in the cell $d$, this observation being the basis for the new tree walking algorithm used by PKDGRAV.

this reason, careful optimization of this relatively small section of code can greatly benefit the overall throughput. Furthermore, since the number of cell interactions generally dominates over the particle–particle interactions, most of the optimization attention is on the efficient evaluation of multipoles.

Next we describe the new walk algorithm used by PKDGRAV, which greatly enhances the level of amortization that is achieved by the basic algorithm. This algorithm is based on the observation that two nearby buckets will show only a small number of differences in their interaction lists, namely in the closest particles and in the closest, and hence smallest, cells. If a cell $a$ is entirely contained by the opening ball of another cell $b$, then cell $b$ will need to be opened by *all* particles of $a$. If $a$ lies entirely outside of the opening ball of $b$, then the multipole of $b$ will be acceptable to all particles of $a$. The following algorithm, presented as pseudo–code in Figure 2.5, uses these two rules to walk the tree from top down in a single pass for all the particles in the simulation, making the minimal required number of changes to the interaction lists.

The algorithm recursively descends the tree from the root–cell to the buckets. A *check list* of cells which still need to be considered for inclusion on the interaction list is processed by the algorithm before descending deeper in the tree. If the opening ball of a cell on the check list entirely contains the current cell we are considering then we remove it and add its two child cells to the end of the check list, effectively opening it for cells deeper within the tree. If the opening ball of a cell on the check list does not intersect the current cell then it is removed from the check list and added to the current interaction list. If neither of these two cases applies then the cell on the check list must remain, and be reconsidered when we proceed deeper in the tree. Once we have processed all cells on the check list in this way we recursively descend to the current cell's children, making sure to add the sibling of each to the remaining checklist. When we have descended to the point where the current cell is a bucket, we open a check list cell if its ball intersects the current bucket cell. This last condition ensures that we end up removing all cells from the check list and that we have constructed a complete interaction list for the bucket. Although not immediately obvious, this produces the same set of interactions as the basic algorithm. We note that although the above algorithm was presented in a recursive form, it is not too difficult to

```
Walk(Cell,CheckList,Interact)
    forall (c ∈ CheckList) do
        if (Cell ⊂ OpenBall(c) ∨ (isBucket(Cell) ∧ Cell ∩ OpenBall(c) ≠ ∅)) then
            CheckList = CheckList − c
            if (isBucket(c)) Interact = Interact + Particles(c)
            else CheckList = CheckList + ChildCells(c)
        else if (Cell ∩ OpenBall(c) = ∅) then
            CheckList = CheckList − c
            Interact = Interact + Moments(c)
        end if
    end do
    if (isBucket(Cell)) then
        Calculate Gravity on Particles(Cell) from Interact.
    else
        LowerCheckList = CheckList + Cell→upper
        UpperCheckList = CheckList + Cell→lower
        Walk(Cell→lower,LowerCheckList,Interact)
        Walk(Cell→upper,UpperCheckList,Interact)
    end if
end Walk
```

Figure 2.5: Pseudo–code for the new walk algorithm used by PKDGRAV. To walk the entire tree we would make the call **Walk**(root–cell,$\emptyset$,$\emptyset$). Note that the forall loop is not fixed, since new elements can be added to the end of CheckList within the loop body. For clarity we have used a single interaction list for both particles and multipoles, in reality this is implemented as two separate interaction lists. The two references, Cell$\rightarrow$lower and Cell$\rightarrow$upper, refer to each of the two subcells of Cell.

recast it to a non–recursive form. It is important to do this because the function calling overhead of a recursive algorithm significantly impacts performance, even on modern RISC processors. This new walk algorithm can be up to 5 times faster than the basic algorithm at constructing the interaction lists. A further unexpected benefit is a speedup in the evaluation of the interaction lists (which remains the same for both algorithms). This is presumably due to having a minimal amount of change in the interaction lists thereby enhancing cache utilization within the microprocessor.

### *2.2.4 Force Softening*

All cosmological N–body simulations must incorporate some form of force softening for interactions at small separations between particles. A pure Newtonian interaction would produce strong scattering amongst particles when they happen to have close encounters, or where they are representing regions of high density. This type of strong scattering violates the conditions underlying the collisionless Boltzmann equation, which governs the evolution of the density distribution for cosmological systems. The particles can be thought of as representing "clouds" of mass, which should be able to freely pass through each other without scattering. This picture is basically the same as that obtained by applying the method of characteristics to discretize the collisionless Boltzmann equation, where the particles represent discrete packets of the underlying phase fluid. The simplest form of force softening that has been used by N–body simulations is the modification of the Green's function to

$$\gamma(r) = -\frac{1}{\sqrt{r^2 + \epsilon^2}}, \tag{2.37}$$

which is known as Plummer softening, with $\epsilon$ being the characteristic softening radius.

The main disadvantage of using the above type of softening is that it is not harmonic. This Green's function does not satisfy equation 2.12, resulting in a very significant performance penalty since we cannot use the reduced multipole formalism. Furthermore, this differs from Newtonian gravity at all radii, leading to slightly different equilibria from theoretically calculated equilibrium models. This results in some amount of undesired oscillation when such theoretical models are evolved. PKDGRAV uses a *spline softened gravity kernel*, which becomes Newtonian beyond a radius of $2h$, $h$ being the softening radius. This is based on the density kernel used by Smoothed Particle Hydrodynamics codes, and is given by,

$$
\begin{aligned}
u &= \frac{|\mathbf{r}|}{h} && (2.38)\\
\rho &= \frac{m}{\pi h^3}\begin{cases} 1-\frac{3}{4}(2-u)u^2, & u<1 \\ \frac{1}{4}(2-u)^3, & 1\le u<2 \end{cases} && (2.39)\\
\phi &= \frac{m}{h}\begin{cases} \frac{7}{5}-u^2(\frac{2}{3}-\frac{1}{10}u^2(3-u)), & u<1 \\ -\frac{1}{15}u^{-1}+\frac{8}{5}-u^2(\frac{4}{3}-u(1-\frac{1}{10}u(3-\frac{1}{3}u))), & 1\le u<2 \end{cases} && (2.40)\\
\mathbf{a} &= \mathbf{r}\frac{m}{h^3}\begin{cases} \frac{4}{3}-\frac{1}{2}u^2(\frac{12}{5}-u), & u<1 \\ -\frac{1}{15}u^{-3}+\frac{8}{3}-u(3-u(\frac{6}{5}-\frac{1}{6}u)), & 1\le u<2 \end{cases} && (2.41)
\end{aligned}
$$

These expressions can be efficiently calculated using a small lookup table.

### *2.2.5 Periodic Boundary Conditions*

One disadvantage of tree codes is that they must deal with periodic boundary conditions explicitly, unlike grid codes where this aspect is taken care of implicitly. Although this adds complexity to any tree code, it is possible to incorporate periodic boundary conditions efficiently by using the Ewald summation technique (Hernquist, Bouchet and Suto 91). Ewald summation uses the decomposition of the Green's function,

$$\frac{1}{r} = \frac{\mathrm{erfc}(\alpha r)}{r} + \frac{\mathrm{erf}(\alpha r)}{r}, \tag{2.42}$$

where the first term converges quickly in real space and the second term converges quickly in Fourier space, for a periodic lattice. The problem with simply replicating the simulation volume and using the $1/r$ Green's function is that this process converges very slowly, and an unacceptably large number of replicated volumes would have to be considered to obtain even modest accuracy. The modified Green's function for gravity over a periodic lattice is given by,

$$g(\mathbf{r}) = \frac{\pi}{\alpha^2 L^3} - \sum_{\mathbf{l}} \frac{\mathrm{erfc}(\alpha|\mathbf{r}-\mathbf{l}L|)}{|\mathbf{r}-\mathbf{l}L|} - \sum_{\mathbf{h}\neq\mathbf{0}} \frac{1}{h^2\pi L}\exp\left(-\frac{\pi^2 h^2}{\alpha^2 L^2}\right)\cos\left(\frac{2\pi}{L}\mathbf{h}\cdot\mathbf{r}\right), \tag{2.43}$$

where $L$ is the size of the periodic box, $\mathbf{l}$ and $\mathbf{h}$ are integer triples, and $\alpha$ is a parameter that determines the convergence of the two sums. The sum over $\mathbf{l}$ is truncated for $|\mathbf{r}-\mathbf{l}L| > \beta_l L$

and the sum over $\mathbf{h}$ is truncated for $|\mathbf{h}| > \beta_h$. By default, PKDGRAV uses the values of $\alpha = 2/L$, $\beta_l = 2.6$ and $\beta_h = 2.8$, these values being on the very conservative side. For most simulations, the values of $\beta_l = 2.3$ and $\beta_h = 2.5$ are usually quite acceptable, significantly reducing the number of terms in the Ewald sum.

PKDGRAV differs significantly from the prescription given by Hernquist *et al.* , which is ill suited to a parallel code because it requires a large static array to store the differences between the non-periodic and periodic forces. A separate copy of this array would be needed for each processor. Instead we use a prescription more like that described by Ding (1992). For a given particle we first calculate the direct, non–periodic forces due to all particles in the fundamental cube and in a number of periodic replicas. The number of replicas is typically the 26 neighbors of the fundamental cube, but this can be made higher for more accurate forces. One could use the opening criterion for the fundamental cube to determine the number of replicas needed for a given accuracy. Including the direct, non–periodic contribution of each of the replicas of the fundamental cube is a simple matter of placing these cells on the check list when calling **Walk** for the root–cell (see Figure 2.5). We then calculate the forces due to the Ewald sum of the multipole moments of the entire fundamental cube on the particle. The Ewald sum is slightly modified so that replicas for which we have already done a direct force only contribute the difference between the direct force and the periodic force. This is discussed in more detail below, and is also shown in Figure 2.6.

The significant difference between our scheme and that of Ding is that whereas they use a small number of particles to represent the multipole moments of the fundamental cube, we explicitly calculate the Ewald sum of the multipole moments. The advantages of our scheme are 1) we avoid errors from the near field that are introduced when representing the multipole moments as a finite number of particles, and 2) we significantly reduce the number of floating point operations for a given accuracy. The disadvantage of our scheme is its algebraic complexity in calculating the Ewald sum of the multipole moments of higher order than the monopole. Fortunately, we have discovered a recursion relation that allows the convenient calculation of the Ewald summation of the higher order moments.

We develop the multipole expansion of the Ewald summation in the same way as was

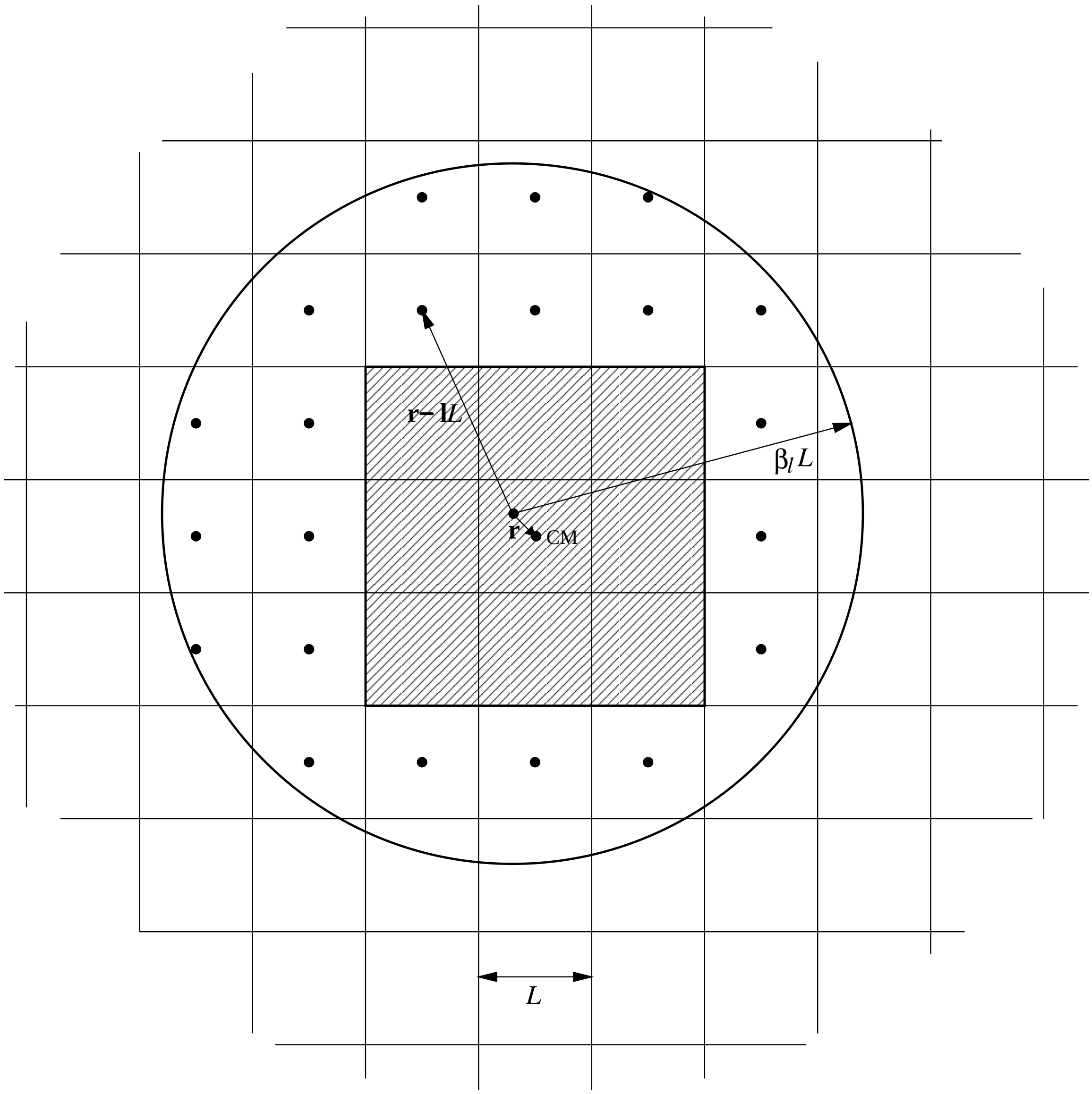


Figure 2.6: Ewald summation over the periodic replicas of the entire simulation volume (the central cube). The summation is truncated at a radius of $\beta_l L$ about the particle in question. The shaded region is treated somewhat differently since it already has the $1/r$ Green's function calculated for it. We apply a correction to the Ewald summation for this near field part to account for this. Each replica CM lying within a radius of $\beta_l L$ of the particle contributes to the Ewald multipole expansion. The regions lying outside of this are taken care of by the Fourier part of the expansion. When the distance between the particle and the CM of the root–cell becomes small we need to use a Taylor expansion for the derivatives of the Green's function to avoid excessive round–off errors.

done in Section 2.2.1. For convenience we rewrite equation 2.43 as,

$$g(\mathbf{r}) = \frac{\pi}{\alpha^2 L^3} + \sum_{\mathbf{l}} \gamma(|\mathbf{r} - \mathbf{l}L|) + \sum_{\mathbf{h}\neq\mathbf{0}} b(\mathbf{r};\mathbf{h}) \tag{2.44}$$

$$b(\mathbf{r};\mathbf{h}) = -\frac{1}{h^2\pi L} \exp\left(-\frac{\pi^2 h^2}{\alpha^2 L^2}\right) \cos\left(\frac{2\pi}{L}\mathbf{h}\cdot\mathbf{r}\right), \tag{2.45}$$

where for the far field part we use,

$$\gamma(r) \equiv \gamma_0 = -\frac{\operatorname{erfc}(\alpha r)}{r}, \tag{2.46}$$

and for the near field part where only the difference to the direct force is to be calculated (see equation 2.42) we use,

$$\gamma(r) \equiv \gamma_0 = -\frac{\operatorname{erf}(\alpha r)}{r}. \tag{2.47}$$

Just as in equation 2.7, we need a recursion relation to provide us with the derivatives of this new Green's function to arbitrary order. This is given by,

$$\gamma_{m+1} = -\frac{(2m+1)}{r^2}\gamma_m - 2^m \alpha^{2m} a \tag{2.48}$$

$$a = \frac{2\alpha}{\sqrt{\pi} r^2} \exp\left(-\alpha^2 r^2\right), \tag{2.49}$$

which applies for both of the above definitions of $\gamma_0$, and for which the 1st order derivative has the identical form to equation 2.8,

$$\partial^{\underline{1}} \gamma_m = \gamma_{m+1} r^{\underline{1}}. \tag{2.50}$$

This means that equation 2.10 provides the derivatives to arbitrary order as in section 2.2.1. The derivatives of $b(\mathbf{r};\mathbf{h})$ are easily shown to be,

$$\partial_{\underline{n}} b(\mathbf{r};\mathbf{h}) = -\frac{(-1)^{\lfloor \frac{n+1}{2} \rfloor}}{h^2 \pi L} \left(\frac{2\pi}{L}\right)^n \exp\left(-\frac{\pi^2 h^2}{\alpha^2 L^2}\right) \left\{ \begin{array}{lll} \cos\left(\frac{2\pi}{L}\mathbf{h}\cdot\mathbf{r}\right), & n & \text{even} \\ \sin\left(\frac{2\pi}{L}\mathbf{h}\cdot\mathbf{r}\right), & n & \text{odd} \end{array} \right\} h_{\underline{n}}. \tag{2.51}$$

The multipole approximation to the Ewald summation is then given by,

$$\Psi(\mathbf{r}) = \frac{\pi}{\alpha^2 L^3} M^{\underline{0}} + \sum_{n=0}^{\infty} \frac{1}{n!} D_{\underline{n}}(\mathbf{r}) M^{\underline{n}} \tag{2.52}$$

$$a_{\underline{1}}(\mathbf{r}) = \sum_{n=0}^{\infty} \frac{1}{n!} D_{\underline{n+1}}(\mathbf{r}) M^{\underline{n}}, \tag{2.53}$$

where $M^{\underline{n}}$ are the multipole moments of the entire fundamental volume about its center–of–mass. In the above, $D_{\underline{n}}$ is given by,

$$D_{\underline{n}}(\mathbf{r}) = \sum_{\mathbf{l}} \partial_{\underline{n}}\gamma_0(|\mathbf{r}-\mathbf{l}L|) + \sum_{\mathbf{h}\neq\mathbf{0}} \partial_{\underline{n}} b(\mathbf{r};\mathbf{h}) \tag{2.54}$$

$$= \sum_{\mathbf{l}} \sum_{m=0}^{\lfloor \frac{n}{2} \rfloor} \frac{1}{m!}\gamma_{n-m}(|\mathbf{r}-\mathbf{l}L|)\delta_{(\underline{2m}}\,(\mathbf{r}-\mathbf{l}L)_{\underline{n-2m})} + \sum_{\mathbf{h}\neq\mathbf{0}} \partial_{\underline{n}} b(\mathbf{r};\mathbf{h}), \tag{2.55}$$

where we have left the substitution of equation 2.51 for the sake of clarity. It would be nice to be able to use the reduced multipole moment formalism here, however this is not technically correct since the Green's functions here are not harmonic. For this reason we have to explicitly calculate the complete multipoles for the fundamental volume. Although not significant in terms of computational cost, this operation adds some complexity to the code.[2]

Finally, there is one added difficulty, the expressions given above for $\gamma_m$ can involve high powers of $|\mathbf{r}-\mathbf{l}L|^{-1}$ for large $m$. These terms tend to cancel out as $|\mathbf{r}-\mathbf{l}L|$ approaches zero, leading to significant round-off errors. This error can be avoided by using the Taylor expansion of the expressions for $\gamma_m$ about zero whenever $|\mathbf{r}-\mathbf{l}L|$ is much less than $1/\alpha$. The Taylor expansions to second order for $\gamma_0$ is,

$$\gamma_0 = -c_0 + \frac{c_0\alpha^2}{3}r^2, \tag{2.56}$$

where $c_0$ is given by,

$$c_0 = \frac{2\alpha}{\sqrt{\pi}}. \tag{2.57}$$

Expressions for the Taylor expansion of the higher derivatives can be gotten from the recursion relation,

$$\gamma_m = -\frac{c_m}{(2m+1)} + \frac{c_m\alpha^2|\mathbf{r}-\mathbf{l}L|^2}{(2m+3)}; \quad c_m = 2c_{m-1}\alpha^2. \tag{2.58}$$

Although this stage is somewhat complicated, it is very efficient. Note that once the derivative tensor $D_{\underline{n}}$ has been calculated, only a single contraction with the multipoles of

[2]Despite this, we have tried using the much faster reduced multipole formulation here, and it seems to produce correct forces at the accuracy we desire. Exactly why this seems to work is not entirely clear to us, and for this reason we refrain from using it in the stock versions of PKDGRAV.

the fundamental volume is required. Furthermore, most of the **h**–sum can be precalculated for all particles. Using local expansions is also possible and would make this stage an almost negligible cost, although this has not been actively pursued by the authors. The most significant point is that this stage is $\mathcal{O}(N)$ and that it is perfectly parallelized, no communication being required to calculate the Ewald contribution. We have seen that as the number of particles of our simulations has increased, the costs of the Ewald summation have become less relevant.

### *2.3 Hierarchical Time Steps*

As the number of particles in an N-body simulation grows, so do the density contrasts. Hierarchical methods can follow extremely large dynamic ranges in densities at modest additional cost per force evaluation. However, large ranges in densities also imply a large range in time scales ($\propto 1/\sqrt{\text{density}}$). If we take the final state of a simulation and weight the computational work done on particles not uniformly but inversely with their natural timesteps, we find a potential gain of $\sim 50$. Temporal adaptivity is one of the last algorithmic areas where we can target an order of magnitude improvement. PKDGRAV uses an adaptive leapfrog integrator, a method where particles are on adjustable individual timesteps. This integrator is discussed in full detail in Chapter 3.

The performance gains achievable by using a particular adaptive timestepping scheme are limited by the "fixed costs" required for any level of the timestep hierarchy. For example, a tree code requires that the tree is built for all the particles regardless of the number of particles requiring force evaluations. Indeed, for a standard particle mesh code, the forces at every grid point are either calculated or not. The only part of the force evaluation that is not part of the fixed cost would be the trivial interpolation of the force onto particles. In this case, essentially all the costs are fixed and no real gain is possible. In addition to these fixed costs, the method of determining the timestep level assigned to each particle can incur a significant cost and must also be accounted for.

Finally, inefficiency in calculating forces for a few particles on modern pipelined, vector or parallel processors, can add an extra "hidden" cost to any adaptive timestepping scheme.

The amortization gains of having several particles per bucket are lost if only one of them needs its force calculated. However, despite these limitations the multistepping speedups for gravity simulations are typically a factor of 4 or more. With future efforts focused on reducing or eliminating these fixed costs, speedups closer to the theoretical upper limits should be realized. Since tree building is currently the dominant fixed cost limiting PKDGRAV, the idea of repairing only parts of the tree, instead of rebuilding it is being investigated.

## 2.4 Parallel Implementation

The implementation of a parallel algorithm can offer new challenges to readability and maintainability of code. Even a very straightforward algorithm can be easily obfuscated by the extra "glue" needed to get all the processors working on their respective data. Our solution to this problem is to divide the programming into a hierarchy of layers which naturally divide the algorithms into their parallel and serial parts. The top layer we refer to as the *Master* layer. This layer controls the overall flow of the program, such as how many timesteps to take, when to write outputs, and the sequence of steps in calculating forces and updating velocities. This is essentially a serial layer. It dispatches parallel work by calling the next layer down which is the *Processor Set Tree* or PST layer. This layer provides the parallel glue to get all processors working on a particular task, and returns the results to the Master layer. At the single processor level, the PST layer calls the *Parallel KD* or PKD layer. This is also a serial layer executing simultaneously on each processor in a MIMD (multiple-instruction-multiple-data) mode. This layer alone has access to the actual particle data. Finally there is a *Machine Dependent Layer* or MDL which provides an interface to parallel primitives.

Note that almost all of the scientific computing is done in either the Master or PKD layers which are written in serial. This clean separation of the serial and parallel parts of the code allows for easy modification of the code. A case in point is the ease in which PKDGRAV, a code designed for cosmology, was adapted for use in studying planet formation (Richardson *et al.* 00).

### *2.4.1 The Machine Dependent Layer*

Portability is critical to the implementation of a production quality simulation code. For parallel applications, *de facto* standards (e.g. PVM) and *de jure* standards (e.g. MPI) have been implemented on most major machines. However, a given standard is not necessarily most efficient for a given machine. Therefore we have isolated the code likely to change between architectures into a machine dependent layer (MDL). This allows us to quickly port our code to the most efficient or "native" parallel library for each machine. The MDL layer provides a set of primitives to handle all the parallel aspects of the code. This includes 1) an asynchronous remote procedure call, 2) a memory swapping primitive, 3) a memory sharing primitive for accessing non-local data, and 4) utilities for printing diagnostics.

On a machine with a shared address space, such as the KSR or Convex Exemplar, the memory sharing is easily implemented. For distributed memory machines, such as a cluster of workstations, we maintain a local software cache on each processor. When a processor requests a tree node or particle that is not local to the processor or in its cache, the request is sent to the appropriate processor that owns the data. While waiting for the data to be sent back, the requesting processor handles cache-requests sent from other processors. The owner processor sends back a cache line comprised of more than a single tree node or particle, in an attempt to amortize the effects of latency and message passing overhead. This cache line is inserted into the local cache, and a pointer to the requested element is returned. This cache mechanism allows us to minimize inter-processor communication without the danger of absorbing large amounts of memory as in schemes involving a locally essential tree. (c.f. Dubinski 96)

### *2.4.2 The Master Layer*

The master layer only runs on a single processor. This processor is normally determined by the MDL. This is accomplished by all processors initially calling an MDL initialize routine. Only the master processor returns from this call to run the "main" program. All other processors enter a loop to process requests at the PST level. The master processor then determines the parameters of the simulation, such as the number of timesteps, the size of the

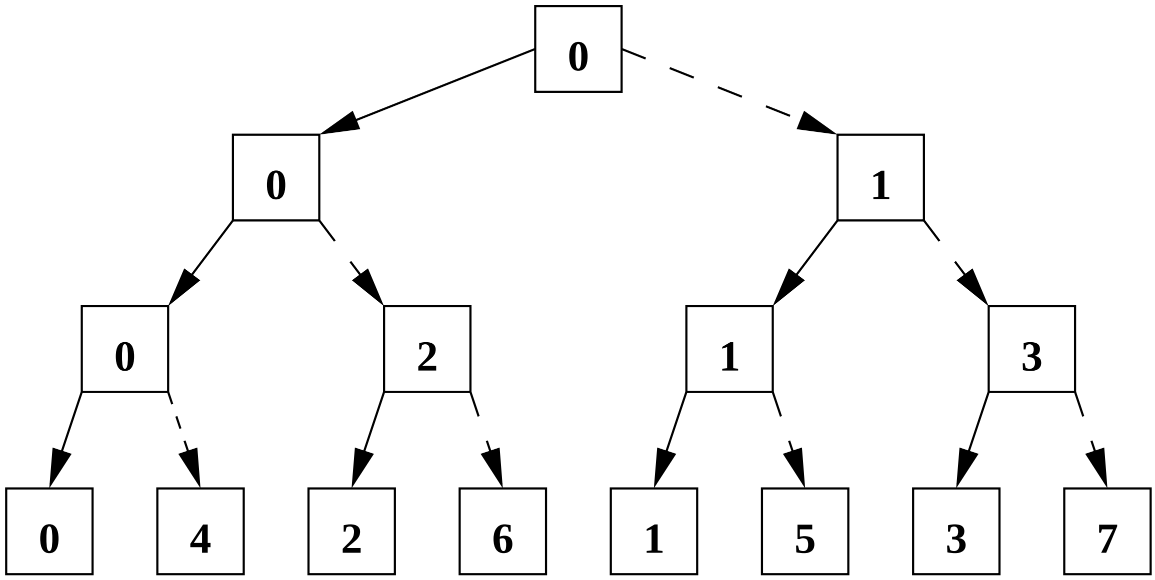


Figure 2.7: The structure of the PST. Solid lines represent actual pointers on the same processor, while dashed lines indicate a reference to another processor. The ordering of processor numbers (or identifiers) is shown for each node of this tree.

timesteps and the force accuracy. For architectures that can only do I/O on one processor, the master processor organizes the I/O via calling PKD level routines directly. Otherwise the I/O is handled via calls to the PST layer.

### *2.4.3 The Processor Set Tree*

The processors (of arbitrary number) are organized in to a binary tree: the Processor Set Tree (PST). Each non-leaf node of the PST contains a pointer to another processor and a pointer to the next lower node. Leaf nodes correspond to individual processors. Therefore, the portion of the tree on each processor is a linked list, where each non-terminal node on the list contains a reference to a child processor. (See figure 2.7.) Parallel procedures are dispatched to all processors by the recursive descent of this tree, which can be accomplished in $\log_2 n_p$ time, where $n_p$ is the number of processors. Any result returned by the individual threads can be combined with results from other threads and passed back up the PST to the master layer.

Achieving effective parallelism requires that work be divided equally amongst the processors in a way which minimizes interprocessor communication during the gravity calculation. Since we only need a crude representation for distant mass, the concept of data locality translates directly into spatial locality within the simulation. Each particle can be assigned a *work-factor*, proportional to the cost of calculating its gravitational acceleration in the prior time-step. Therefore, during *domain decomposition*, we divide the particles into spatially local regions of approximately equal work.

Experience has shown that using a data structure for the domain decomposition that does not coincide with the hierarchical tree for gravity calculation, leads to poor memory scaling with number of processors and/or tedious book-keeping. That is the case, for instance, when using an Orthogonal Recursive Bisection (ORB) tree for domain decomposition and an oct-tree for gravity. Current domain decomposition techniques for the oct-tree case involve forming "costzones," that is, processor domains out of localized sets of oct-tree cells (Singh 93), or "hashed oct-trees" (Warren and Salmon 95). PKDGRAV uses the ORB tree structure to represent the domain decomposition of the simulation volume. The ORB structure, which is implemented using the PST, is completely compatible with the binary tree structure used for the gravity calculation. A root finder is used to recursively subdivide the simulation volume so that the sums of the work-factors in each processor domain are equal. Once this has been done, each processor builds a local tree from the particles within its domain. This entire domain decomposition and tree building process are fully parallelizable and incur negligible cost to the overall gravity calculation.

### *2.4.4 The PKD Layer*

This layer alone has access to the particle and tree data structures. The code within this layer is executed by all the processors simultaneously as dictated by the PST layer above it. The PKD layer consists of predominantly serial code with a few MDL calls to access data on other processors. Not surprisingly, almost all of the computational work is performed by this layer. This subdivision into MASTER, PST and PKD layers along with the separate MDL library has simplified development and extension of PKDGRAV and has also facilitated porting it to a diverse set of parallel architectures.

### *2.5 Accuracy*

Analyzing the behavior of errors of an N–body code with changes in accuracy parameters and in the type of simulation, is a very difficult task. Each type of N–body code behaves somewhat differently with respect to errors, particularly when different applications are being compared. Compounding the difficulties is the fact that there is always a trade–off between computational speed and sufficient control of the error properties of a code. It is dangerous to ignore the error distributions, allowing all possible compromises to achieve higher computational throughput. On the other hand, it is equally fruitless to consume valuable compute cycles attaining very good error behavior, when a simpler approach could yield mean errors which are orders of magnitude lower. The art in all modern N–body codes is achieving a compromise between these two extremes.

The approach taken in what follows is to examine the acceleration error properties for a suite of particle distributions. Instead of using equilibrium models, such as a king model, model galaxy or other such uniform distributions, we focus on the hardest tests of the code which we have encountered. These are the cosmological simulations which exhibit a hierarchy of structures on all scales, and require the use of our multipole Ewald summation technique to incorporate the periodic boundary conditions. Particularly difficult are the volume renormalized simulations, where a very large number of particles is used to focus on a structure of interest, such as a galaxy cluster within a very large cosmological volume, with the remaining particles representing the tidal fields of the more distant structures. This type of simulation has pushed PKDGRAV in terms of its ability to efficiently follow extremely large dynamic ranges in density and time scale. We show these test distributions of particles in Figure 6.3, where test4 is a volume renormalized galaxy cluster, similar to the Virgo cluster, in a cosmological volume of 50 $h^{-1}$ Mpc$^3$.

For each test distribution we calculate the exact forces using a direct ($\mathcal{O}(N^2)$) sum over all pairwise interactions, where the interaction we use between the particles is the direct Ewald sum given by equation 2.43. Having to calculate the periodic boundary conditions

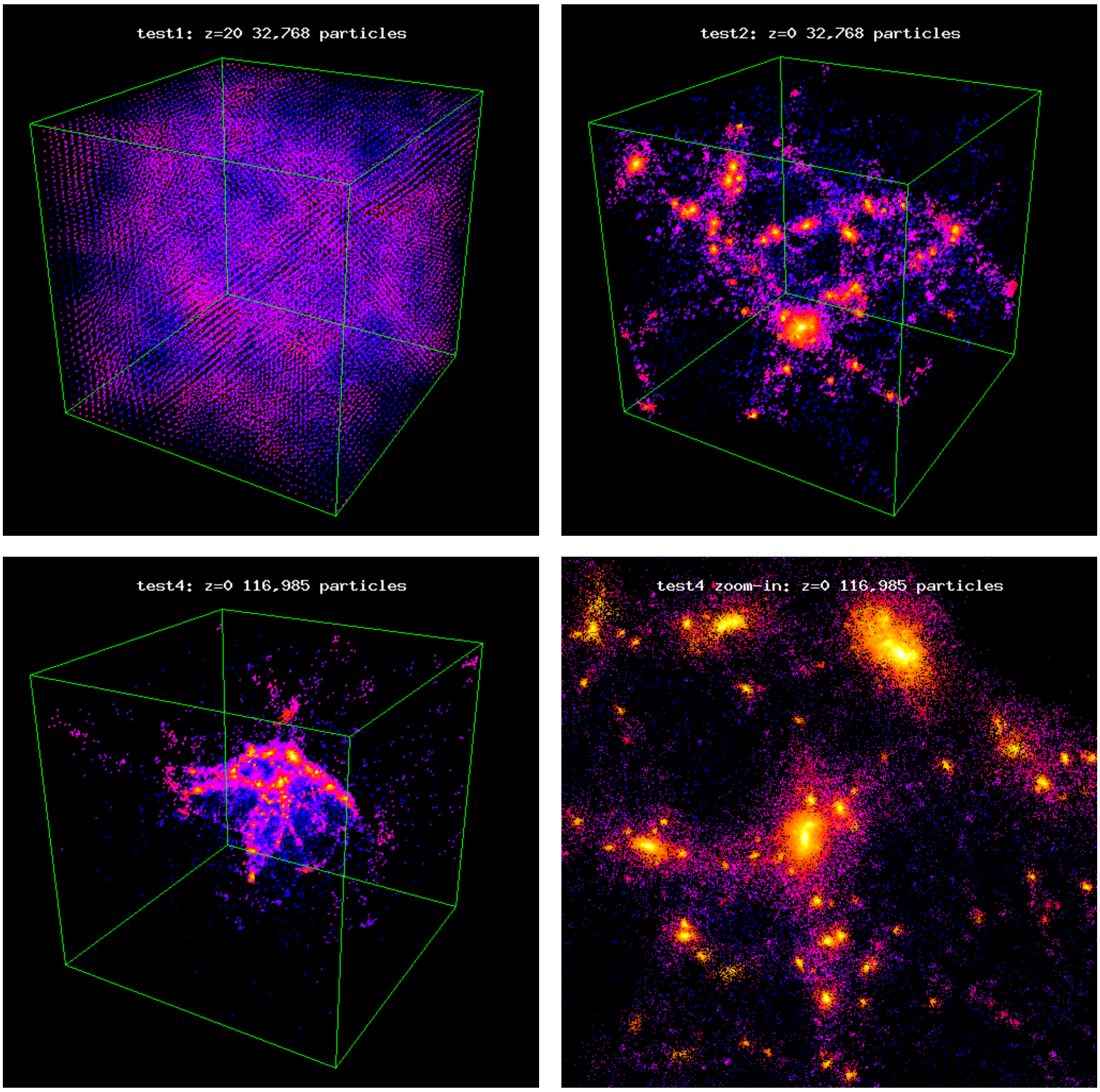


Figure 2.8: Cosmological test particle distributions used for error analysis of PKDGRAV. Colors represent local mass density which range from $\log \delta\rho/\rho$ of -1 (black) to 5 (white) for all except test1, where the range is from -0.2 to 0.7 to enhance contrast in the nearly uniform distribution.

directly in this way is extremely expensive, restricting the error analysis of cosmological particle distributions to smaller numbers of particles. This restriction is not a serious one, since non–periodic tests on larger numbers of particles show that the error properties do not change significantly. We have run PKDGRAV on these tests, investigating changes in a variety of parameters including: $\theta$; expansion order; $\beta_l$, $\beta_h$ and the number of replicas for the Ewald approximation; the number of particles per bucket; modifications of the tree structure; and the number of processors. Of these we will focus only on the error distributions resulting from change in $\theta$ and the use of 2nd vs. 4th order multipole expansions. Sample error distributions are show in full in Figure 2.9. The top two panels show the relative and absolute acceleration error distributions. The remaining panels check for any correlations of the errors with the axes of the simulation box, and with the direction of the acceleration at each particle. The difference seen in the two distributions of the middle–right panel are normal, and arise from the difference in dimensionality of the parallel and perpendicular spaces. However, a significant difference in the RMS–values in any of these four panels would indicate a potential problem. These plots have been very useful in the development of PKDGRAV, and have successfully identified a number of bugs and other problems. They have also been of aid in assessing the relative quality of various methods, looking at the general shape of the distributions, how extended the tails are, and how steeply the distributions fall off towards larger error.

Figure 2.10 shows the compiled results for each of the tests, where the dependence of the RMS relative error on both $\theta$ and the order of the expansion are shown. Note the substantial improvement in the error when using the 4th order expansion. The relative acceleration errors for test1 are so much larger because this is a nearly uniform grid of particles, and the actual accelerations are quite small. Both of test2 and test4 are highly clustered distributions (particularly test4), and are more representative of particle distributions encountered by an N–body code throughout most of a cosmological simulation. Since $\theta$ controls the relative error of each individual multipole contribution, the large amount of cancellation which takes place amongst these contributing multipoles for nearly uniform distributions creates much larger errors in the total force than expected. With PKDGRAV we recognize the need for smaller $\theta$ during the early stages of a cosmological simulation and switch to a more

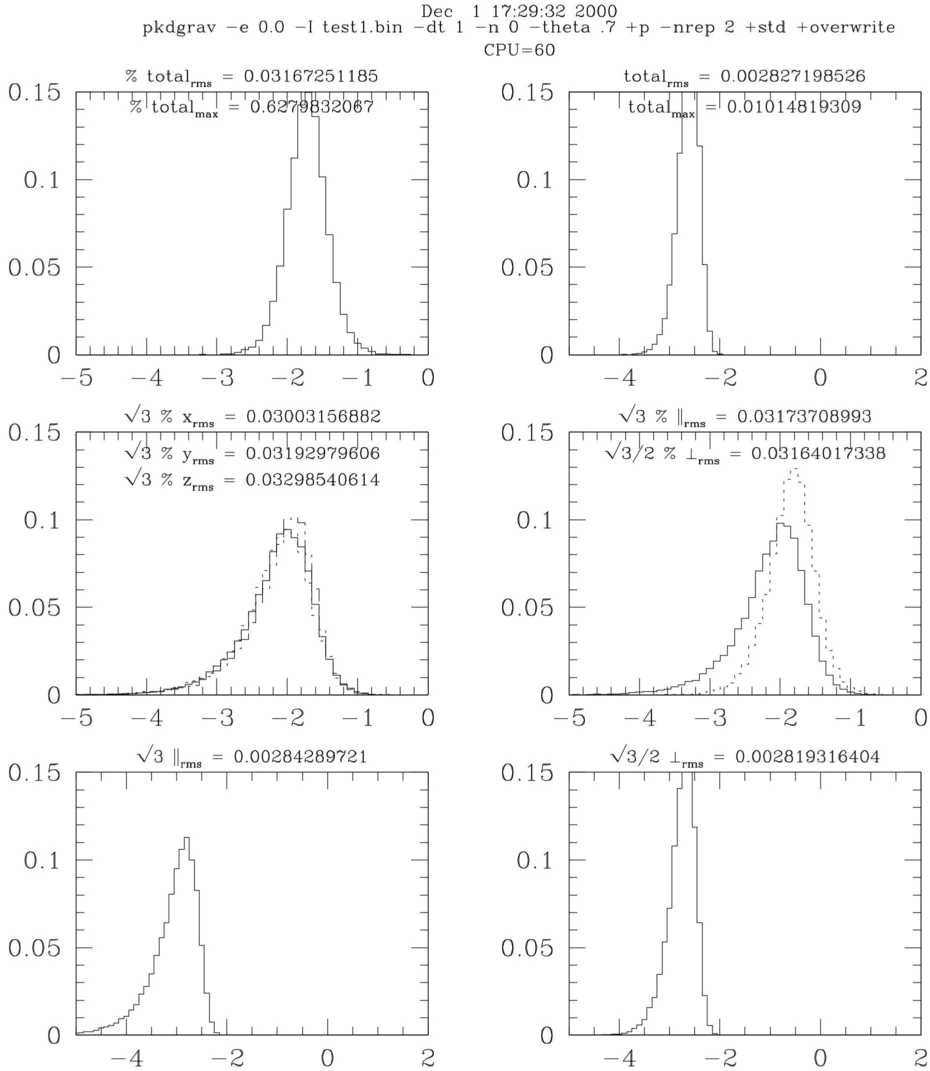


Figure 2.9: Error distributions for test1 using $\theta = 0.7$.

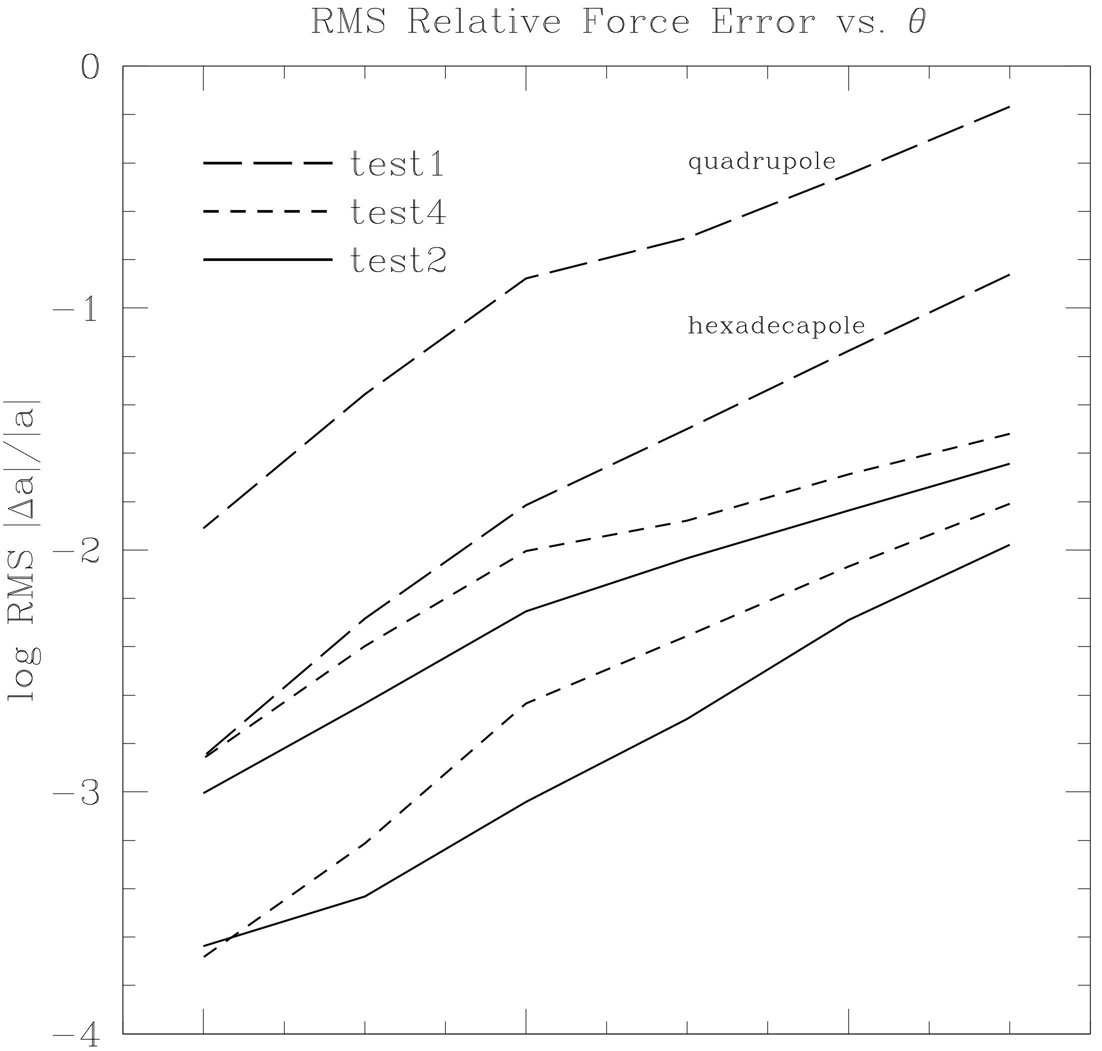


Figure 2.10: RMS relative force error as a function of $\theta$ and the order of multipole expansion used.

representative value once clustering is well developed. We make this switch in $\theta$ at a redshift of 2, and have typically used the values of $\theta_{z>2} = 0.55$ and $\theta_{z<2} = 0.7$ for most cosmological simulations.

In Figure 2.11 we show the maximum relative error for each of the tests, a measure which is noisier than the RMS-value. This shows the absolute extent of the large error tail in the distributions, which is different from the 99th, or other, percentile quoted by some authors (although these are also useful measures, see Dehnen 2000). We note that the 4th order expansion shows better control of the maximum error as $\theta$ is decreased, this trend is also seen in the shapes of the error distributions. Figure 2.12 shows the cost of evaluating the interaction list (not including tree walking or Ewald summation) versus the RMS relative acceleration error. This clearly shows that using the 4th order multipole expansion is significantly more efficient than only using 2nd order, at all levels of accuracy we have examined. The reason that the multipole expansion to 3rd order has not been considered here is that this does almost nothing to improve the errors beyond the 2nd order expansion. The reason for this is that the box–like symmetry of the particles within the cells still dominates the moments at this low order. Finally, we note that the cost (or number of multipole interactions) scales almost exactly as, $C \propto \theta^{-2\frac{1}{3}}$ for these cosmological test cases.

These are all static force tests, and dangers can still lurk in the form of correlated errors of various forms. The final test must always be in the form of complete simulations. For cosmological simulations, the expensive but simple use of convergence tests is effective, where a series of simulations with ever tighter tolerances in the accuracy parameters are performed at a lower number of particles. In such simulation runs we look for convergence in the central density profile of the dark matter halos, this being one of the most stringent tests. Besides this one can make sure that the fundamental mode of the cosmological volume grows in accordance with linear theory, this being a strong test of the periodic boundary conditions code and its accuracy parameters. Besides these we typically also look for convergence in the mass function of halos, to make sure that we are not inadvertently wiping out the lowest end of the resolvable mass scale. Since the initial development of PKDGRAV, we have run many instances of a 1.3 million particle volume renormalized cluster, which now serves as a dynamic litmus test of changes in code, error parameters, periodic boundary conditions,

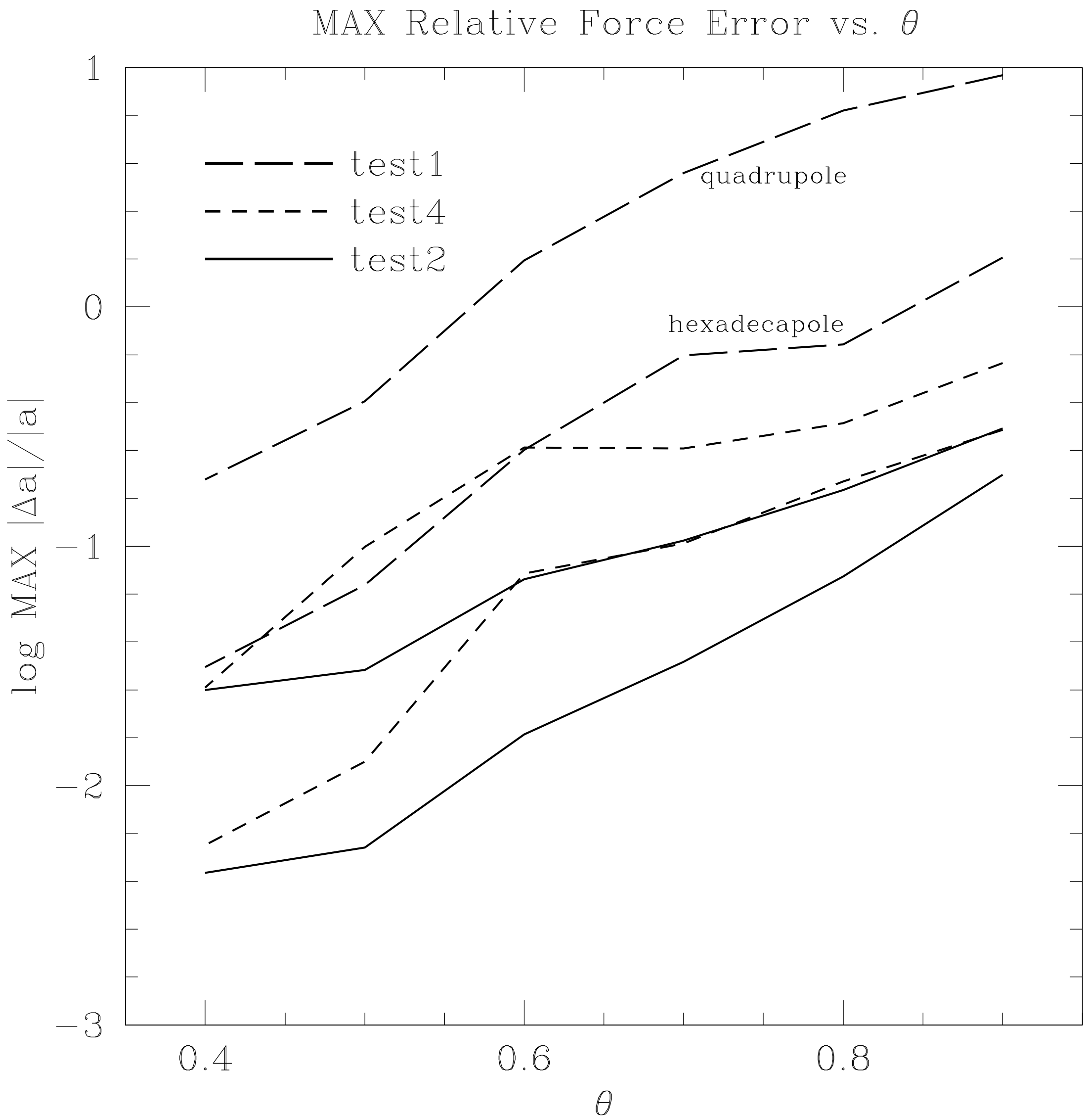


Figure 2.11: Maximum relative force error as a function of $\theta$ and the order of multipole expansion used.

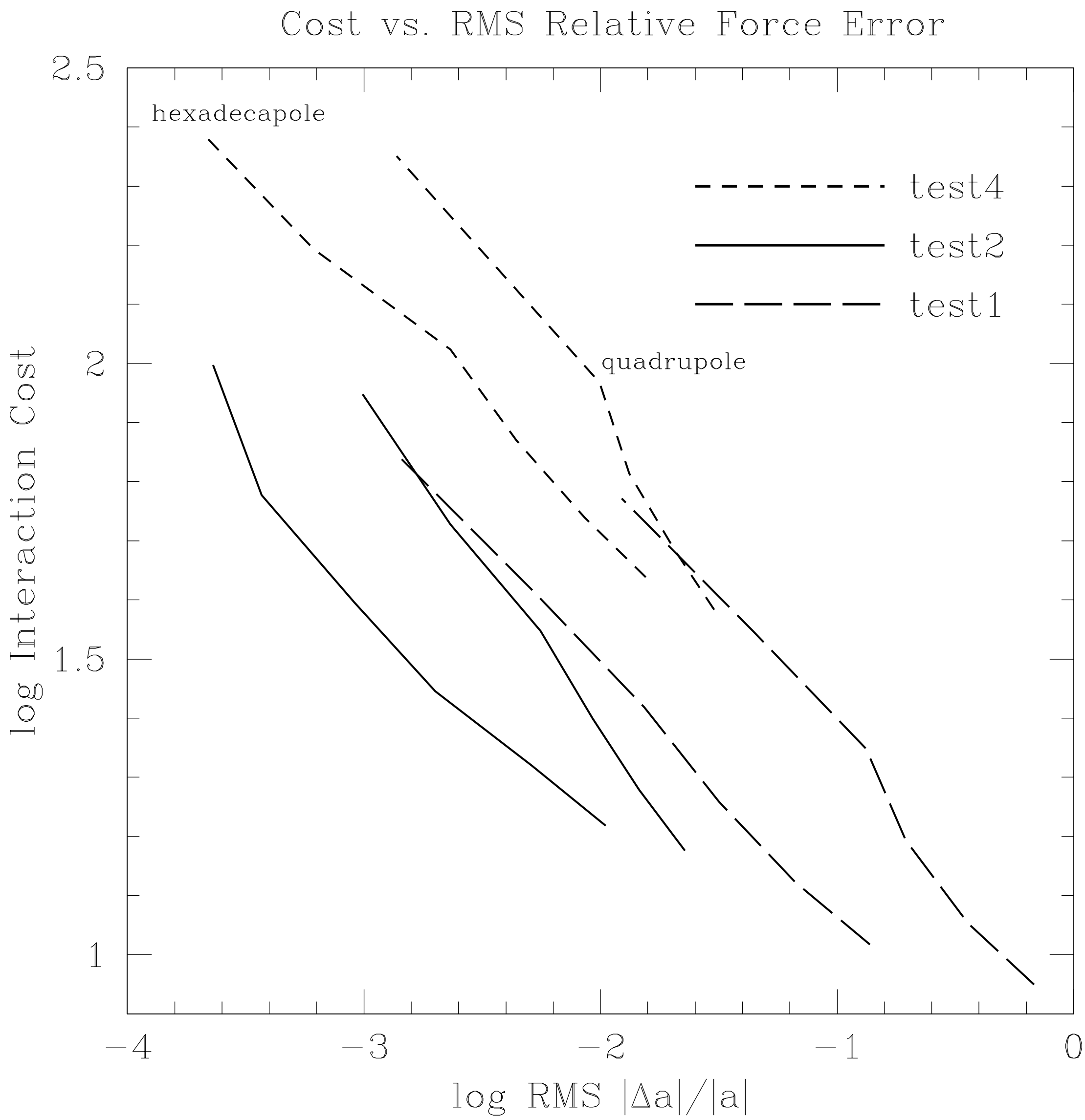


Figure 2.12: Relative cost of evaluating the interaction list as a function of the RMS relative force error and multipole expansion order. This shows that at a given force accuracy, the use of 4th order multipoles is more efficient.

time stepping and softening.

## 2.6 Performance

Although theoretically the aggregate computing power and memory of a parallel computer are $P$–times greater ($P$ being the number of available processors) than that of a single processor machine, in practice the complete use of these aggregate resources towards a complex task such as an N–body simulation is not achievable. Furthermore, algorithms with asymptotic linear scaling with $P$ often fall very short of scaling the best serial algorithm by a factor of $P$. Two factors limit the scaling of an N–body code with number of processors. The first is the ability to effectively balance the work load amongst an increasing number of processors. The second is maintaining locality of the data references so that communication between processors is minimized. An imbalance as small as 1% in the work assigned to processors will significantly impact the throughput of a parallel computing task when a large number of processors (of order 100) are used. Furthermore any serial overheads will come to limit the speedup that is achieved when increasing the number of processors.

Figure 6.6 shows the speedup with number of processors for the gravity and tree building phases of PKDGRAV. It can be seen that PKDGRAV scales very close to linearly out to 256 processors, achieving about 80% linear speedup from single processor performance at 256 processors. In figure 6.6 we also show the result of algorithmic improvements ("old"[3] vs. "new"), where the gravity has been improved by a factor of 2 and the tree build by a factor of 5. The improvements in the tree build are particularly important since this fixed cost currently limits the speedups attainable when using multistepping, because although gravity can be calculated for a small $N_{\rm active}$ subset of the particles, building the tree is an operation which currently can only be accomplished in $\mathcal{O}(N)$ time at best. Future efforts will be directed toward "repairing" the tree in $\mathcal{O}(N_{\rm active})$ time instead of rebuilding it. These recent improvements are part of a general trend of algorithmic improvements which over the last several decades have rivaled the dramatic technological improvements in expanding the scope of cosmological N-body simulations.

[3] All simulations performed before 2001 used this version of PKDGRAV.

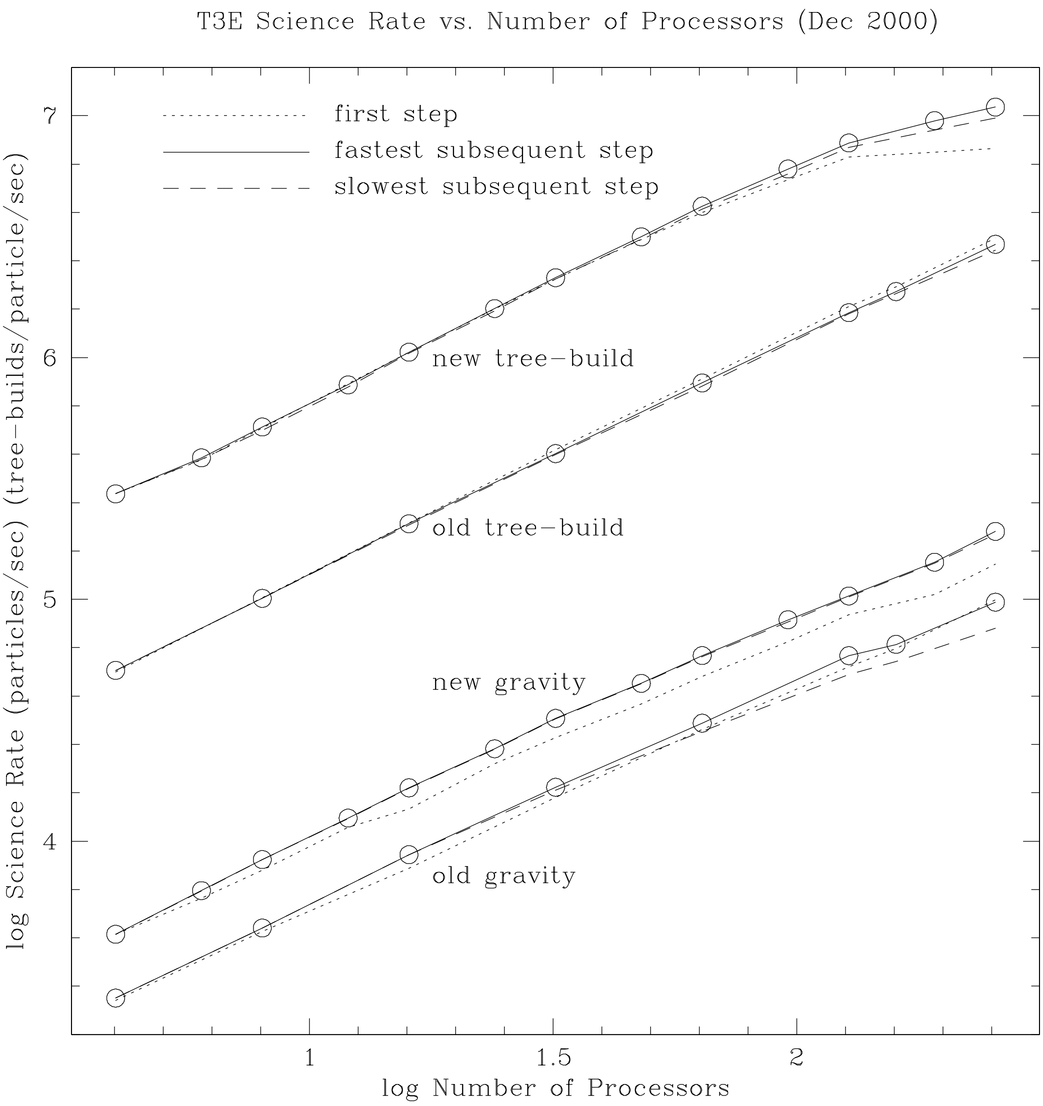


Figure 2.13: Science rate scaling with number of processors for the Cray-T3E parallel computer. A clustered cosmological particle distribution of 3 million particles was used with Θ set to 0.8. Scaling curves for the first step and the fastest and slowest subsequent steps are shown to illustrate the effects of domain decomposition. The “new” curves show a recently improved version of PKDGRAV.

Figure 6.6 also shows the speedup for the first gravity step and the fastest and slowest of 5 subsequent steps. This illustrates the importance of careful work decomposition. In the first step PKDGRAV must assume all particles have the same work factor associated with them, thereafter it can use the actual computational work that was required for each particle in the preceding step. This results in a much better domain decomposition, significantly improving the scaling for large numbers of processors. Note the pathologies in the relationship of these curves for the "old" code beyond about 50 processors, where the zero'th order assumption of all particles having equal work eventually surpassed subsequent attempts at achieving an improved domain decomposition. Furthermore, the fluctuations in the performance of subsequent steps showed that the method of determining the work factors for particles could cause some instability in the domain decomposition process. The new code always improves from the first step and its performance remains stable for subsequent steps. The raw gravity throughput of PKDGRAV can be obtained from the *Science Rate* (particles/sec) given on the vertical axis of figure 6.6, knowing that the curves correspond to a 3 million particle clustered test case and used $\Theta = 0.8$ which with 4th-order multipoles gives good force accuracy. For example, using 256 processors we can calculate forces on all 3 million particles in 15.6 seconds, a science rate of 192 thousand particles per second.

It is interesting to note that PKDGRAV does not require such a large bandwidth of communication to scale well. Roughly 20 MBytes/sec are sufficient with the software MDL cache to service request for data from other processors. This means that 100 Mb Ethernet in a Beowulf cluster performs very well. However, latency is a more critical factor for the current code. Low overhead message passing libraries and use of low latency communication switches can significantly impact the performance of the code.

### *2.7 Summary and Future Prospects*

We have presented PKDGRAV a fully parallel N-body treecode for cosmological simulations. PKDGRAV takes advantage of high order multipoles to improve the speed of gravity calculations at a fixed error tolerance. It uses a binary tree instead of the more traditional oct-tree, which is allows the local tree on a processor to be a simple extension of the domain decomposition tree. The binary tree is also more efficient since it simplifies all tree operations including building, walking, and searching. A high degree of amortization over particles in the walking phase reduces the cost of collecting interactions and intensifies the floating point rate in evaluating the interactions. Implementing periodic boundary conditions required for cosmological simulations efficiently in a treecode is non-trivial. PKDGRAV uses a new technique for approximating the Ewald method of incorporating periodic boundary conditions which is very efficient and is ideal for parallel N-body tree codes. The use of hierarchical timestepping greatly increases the throughput of PKDGRAV for cosmological N-body simulations since these simulations exhibit a very large dynamic range in timescales. The use of the separate high level MDL library to isolate architecture dependent aspects of the code, allows PKDGRAV to take advantage of many types of parallel architectures and also different message passing libraries and parallel software environments. The attention given to software design throughout the development of the code has allowed several authors to successfully contribute in this development.

There are several areas where future development of PKDGRAV will be focused. Currently local expansions are not used, but once added they will provide significant speedup in the gravity calculation. MDL needs to be extended to allow issuing of prefetch request for data on other processors, since the new tree walking algorithm can take advantage of this. This will effectively hide most of the latency penalty for PKDGRAV and further improve parallel scaling, particularly on Beowulf class machines. However, by far the most important area where the throughput of PKDGRAV can be increased is in multistepping. Multistepping presents several new algorithmic challenges. The first of these is the elimination of the $\mathcal{O}(N)$ tree building fixed cost which currently becomes the dominant cost in the code when a very large hierarchy of timescales is present. Various schemes of incrementally repairing the tree need to be investigated. The second difficulty with multistepping is that it is extremely difficult to obtain a good distribution of the computational load for all of the timestep hierarchy. Finally, although PKDGRAV was presented here only in the context of cosmological simulations, it has been extended to incorporate collisional physics (see Richardson *et al.* 2000) which has lead to many new solar system applications. PKDGRAV is now also being used to study the physics of granular dynamics which relies exclusively on the collisional physics within the code. A large effort has gone into adding Hydrodynamics, radiative cooling, star formation and supernova feedback to PKDGRAV. This effort is largely complete and cosmological simulations to study galaxy formation are underway using this new code known as GASOLINE. We hope that PKDGRAV will continue to see such application into other areas of computational science.

# Chapter 3

# TIME STEPPING LARGE N-BODY SIMULATIONS

## *3.1 Introduction*

Over the past decade, spatially adaptive methods have been developed to calculate gravitational forces in N-body simulations. These include tree codes (Appel, 1985; Barnes and Hut, 1986), fast multipole methods (Greengard, 1987), and adaptive grids (Couchman, 1991). As the number of particles in an N-body simulation grows, so do the density contrasts. Hierarchical methods can follow extremely large dynamic ranges in densities at modest additional cost per force evaluation. However, large ranges in densities also imply a large range in time scales ($\propto 1/\sqrt{\text{density}}$). If we take the final state of a simulation and weight the computational work done on particles not uniformly but inversely with their natural timesteps, we find a potential gain of $\sim 50$. Temporal adaptivity is one of the last algorithmic areas where we can target an order of magnitude improvement. Hence, we seek a hierarchical integrator, *i.e.*, a method such that particles are on adjustable individual timesteps.

The most commonly used time integration scheme for N-body simulations is the leapfrog method. Leapfrog has several advantages over other methods. 1) For second order accuracy only one force evaluation and one copy of the physical state of the system is required. This is particularly beneficial for N-body simulations where the cost of a force evaluation is very expensive and memory usage is often a critical concern. 2) The force field in an N-body simulation is not very smooth, so higher order does not necessarily mean higher accuracy (Press, 1992, p. 711). 3) It is a symplectic integrator, *i.e.*, it preserves properties specific to Hamiltonian systems. See Channell and Scovel (1990) for a review of symplectic integrators. Gravitational N-body systems are Hamiltonian, and therefore they should benefit from the use of an integrator that conserves phase space volume and has no spurious dissipation. This is especially important in self-gravitating systems where a dissipation time scale that is linked to the dynamical time can lead to a runaway to spuriously large densities. *Leapfrog is a second order symplectic integrator requiring only one costly force evaluation per timestep and only one copy of the physical state of the system.* These properties are so desirable that we concentrate on making an adaptive leapfrog.

Hierarchical leapfrogs have been used before (Porter 1985; Ewell 1988; Hernquist and Katz 1989), but they were not symplectic (§4). There exist symplectic integrators with individual but *fixed* timesteps for either particles or modes (Saha & Tremaine 1994; Skeel & Biesiadecki 1994; Lee, Duncan & Levison 1997). Hut, Makino & McMillan (1995) proposed an iterative scheme for choosing timesteps for a single particle in a rapidly varying potential, but each iteration involves a force evaluation that is prohibitively expensive for large N simulations (§2).

Section 2 describes the theory behind a hierarchical integrator. Section 3 describes the choice of timestep. Section 4 describes tests of this integrator on a single particle in a potential, and Section 5 will present tests of the integrator on full N-body systems. Performance issues will be addressed in Section 6. We discuss the implications of these results in Section 7.

### *3.2 Symplectic Integrators*

A symplectic integrator is an exact solution to a discrete Hamiltonian system that is close to the continuum Hamiltonian of interest. Therefore, it preserves all the Poincaré invariants (Arnold 1978) and places stringent conditions on the global geometry of the dynamics. An obvious example is total energy conservation in a system with a time-independent Hamiltonian or the conservation of angular momentum in axisymmetric systems (Zhang & Skeel 1995). A symplectic integrator will exactly conserve the energy in the discrete Hamiltonian that is an approximation to the true energy of the system. This approximate energy oscillates about the true energy without any numerical dissipation.

The difference between the discrete and continuum Hamiltonians can be viewed as a small perturbation given by the truncation error of the integrator. The error is a Hamiltonian! If the error Hamiltonian is a sufficiently small perturbation, then the KAM theorem

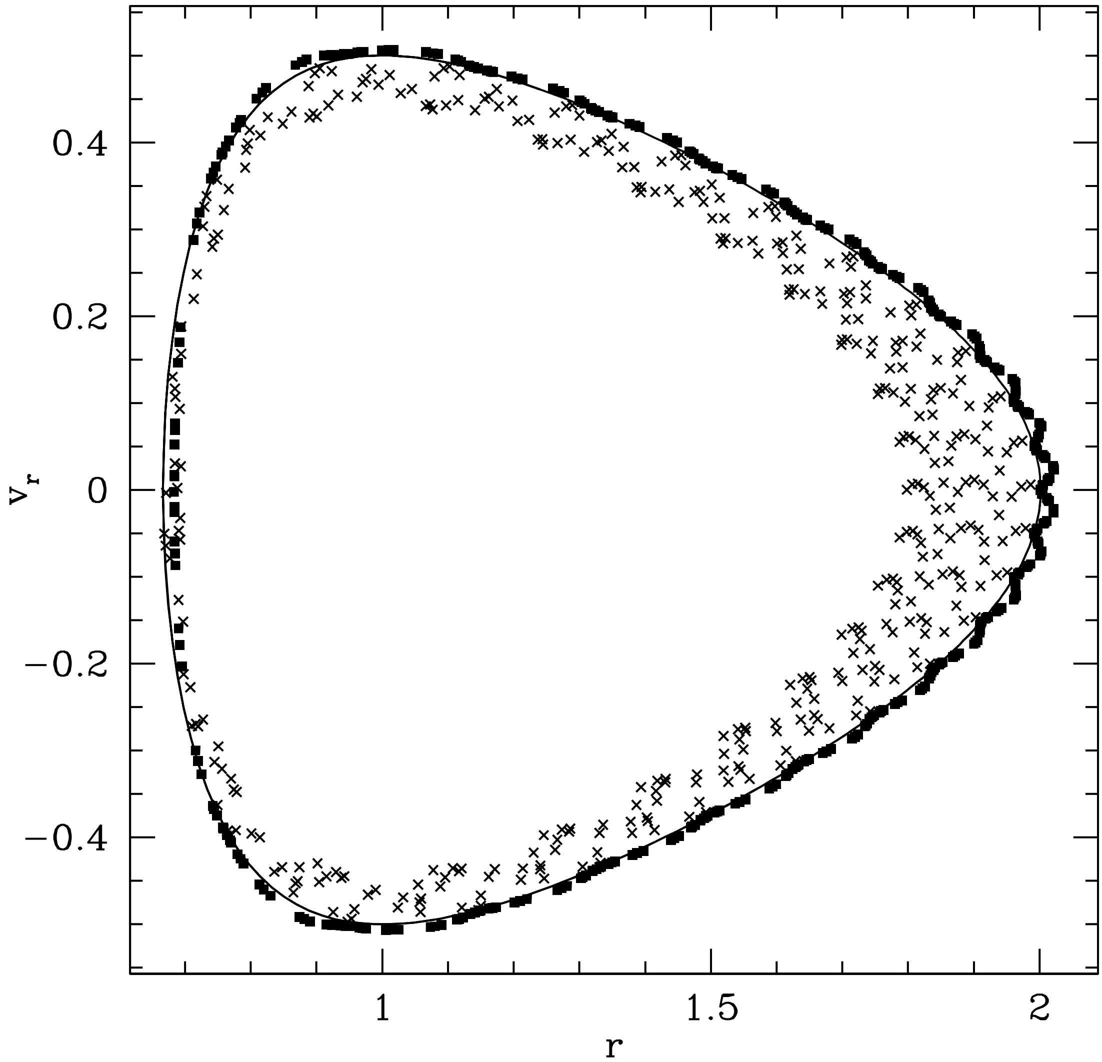


Figure 3.1: A comparison of symplectic and non-symplectic integrators is made. The squares are a second order leapfrog integrator; the crosses are a 4th order Runge-Kutta integrator with the same timestep, and the solid line is the exact solution.

(Arnold 1978) guarantees that the invariant curves destroyed are a set of finite measure. In other words, almost all orbits that are stable in the real system will continue to be stable in the numerical system. An illustration of these advantages is shown in Figure 3.1. Here the radial velocity, $v_r$, is plotted against the radius, $r$, for an ellipticity, $e = 0.5$ Kepler orbit using a leapfrog integrator and using a fourth order Runge-Kutta integrator. In each integration, approximately 24 steps were taken per orbit, and the integrations ran for 16 orbits. Note how the leapfrog integrator oscillates about the true solution but always remains on a one dimensional surface. This indicates that it is indeed conserving an energy-like quantity, *i.e.* having the orbit constrained to a one dimensional surface shows the existence of an isolating integral of motion. On the other hand, the Runge-Kutta orbit slowly becomes more circular. The poor performance of the Runge-Kutta integrator is remarkable given that it is a fourth order integrator and uses four times as many force evaluations as the leapfrog integrator. Also note the large wiggles in the leapfrog integration at apoapse. These are indicative of the proximity of resonant islands that would lead to a instability for larger timesteps.

The leapfrog integrator can be written as

$$\begin{aligned}
\mathbf{r}_{n+1/2} &= \mathbf{r}_n + \tfrac{1}{2}\tau\mathbf{v}_n, \\
\mathbf{v}_{n+1} &= \mathbf{v}_n + \tau\mathbf{a}(\mathbf{r}_{n+1/2}), \\
\mathbf{r}_{n+1} &= \mathbf{r}_{n+1/2} + \tfrac{1}{2}\tau\mathbf{v}_{n+1},
\end{aligned}$$

where $\mathbf{r}$ is the position vector of a particle, $\mathbf{v}$ is the velocity, $\mathbf{a}$ is the acceleration (a function of position only), and $\tau$ is the timestep. When several of these steps are put together, the two position updates can be combined to a single update $\mathbf{r}_{n-1/2}$ to $\mathbf{r}_{n+1/2}$, and the resulting alternation between updating $\mathbf{r}$ and $\mathbf{v}$ gives leapfrog its name. The symplectic nature of leapfrog can be seen by noting that the position update is equivalent to evolving the system exactly under the Hamiltonian $H_D = \frac{1}{2}v^2$, and the velocity update is equivalent to evolving the system under the Hamiltonian $H_K = V(\mathbf{r})$, where $V(\mathbf{r})$ is the potential generating the accelerations. We will call the operator that evolves the system under $H_D$ the "drift" operator, $D$, and the operator that evolves the system under $H_K$ the "kick" operator, $K$.[1]

[1] If one expresses Hamilton's equations as $\dot{\mathbf{z}} = \{\mathbf{z}, H\}$, where $\mathbf{z}$ are the phase space coordinates and $\{\ ,\ \}$ are Poisson brackets, then formally $D(\tau)\mathbf{z} = \exp(\tau\{\mathbf{z}, H_D\})$, and $K(\tau)\mathbf{z} = \exp(\tau\{\mathbf{z}, H_K\})$.

It can be shown that (see Saha and Tremaine, 1992 for details) the combination of operators $D(\tau/2)K(\tau)D(\tau/2)$ evolves the system under a Hamiltonian

$$H_N = H_D + H_K + H_{err} = \frac{1}{2}\mathbf{v}^2 + V(\mathbf{r}) + H_{err},$$

where $H_{err}$ is of order $\tau^2$, reflecting that this is a second order integration method. Furthermore, the existence of this surrogate Hamiltonian ensures that leapfrog is symplectic. Since the Hamiltonian is separable the combination of operators $K(\tau/2)D(\tau)K(\tau/2)$ is also a symplectic second order integrator. Explicitly, this is

$$\begin{aligned}
\mathbf{v}_{n+1/2} &= \mathbf{v}_n + \tfrac{1}{2}\tau\mathbf{a}(\mathbf{r}_n),\\
\mathbf{r}_{n+1} &= \mathbf{r}_n + \tau\mathbf{v}_{n+1/2},\\
\mathbf{v}_{n+1} &= \mathbf{v}_{n+1/2} + \tfrac{1}{2}\tau\mathbf{a}(\mathbf{r}_{n+1}).
\end{aligned}$$

Higher order symplectic integrators can be constructed from combinations of leapfrog steps (Yoshida 1990). Each $N$th order leapfrog integrator requires $N-1$ force evaluations and only one copy of the physical state of the system.

Unfortunately, constructing a variable step-size method by choosing a new timestep after each leapfrog step gives very disappointing results (Calvo and Sanz-Serna, 1993). Some simple schemes can be shown to have no rigorous stability criterion for any step size (Skeel 1993). Several explanations have been given for this behavior. The simplest is to note that a variable step integrator is evolving a dynamic system with state variables ($\mathbf{r}$, $\mathbf{v}$, $\tau$). The projection onto the phase space coordinates ($\mathbf{r}$, $\mathbf{v}$) cannot be described by a Hamiltonian. Skeel and Gear (1992) consider a one step symplectic operator of the form

$$(\mathbf{r}_{n+1}, \mathbf{v}_{n+1}) = F(\mathbf{r}_n, \mathbf{v}_n, \tau(\mathbf{r}_n, \mathbf{v}_n)),$$

and show that if $F$ is symplectic for a constant step size, it will not in general be symplectic for variable step size. Another way to see this problem is to notice that the time reversibility has been broken: if we step forward in time and then step backward, we do not end up at the same point because of the change in timestep. Since symplectic implies time reversible, such an operator is not symplectic. Integrators that are time reversible are referred to as reflexive by Kahan (1993; see also Kahan and Li 1997) who argues that reflexivity is the key to the robust properties of "updating formulae" rather than symplecticity.

A strategy to make leapfrog reflexive is clear from a cursory look at its evolution operators. The operation $D(\tau/2)K(\tau)D(\tau/2)$ is reversible if we use a "select" operator to choose the timestep, $S$, such that the time reversibility is retained. Hut, Makino, and McMillan (1995) achieve this by using an implicit definition of the timestep:

$$\tau = \frac{1}{2}[\tau(\mathbf{r}_n, \mathbf{v}_n) + \tau(\mathbf{r}_{n+1}, \mathbf{v}_{n+1})].$$

So, the beginning and end of the step are required to agree on the timestep. One can solve for $\tau$ iteratively, and very good results are obtained even with only one or two iterations. (This result is also clear from Kahan 1993). This iterative scheme poses several problems when applied to a large N-body simulation. It requires backing up timesteps, throwing away expensive force calculations, using auxiliary storage and one must specify a method for synchronizing the particles for mutual force evaluations.

However, we can also see an alternative means to restore reflexivity. Let us choose a select operator $S$ that commutes with $K$, so that $DSKD$ is equivalent to $DKSD$. Since $K$ only changes the velocities and not the positions, an $S$ operator that depends entirely on positions satisfies this commutation requirement. As we shall show below, this scheme is not strictly time reversible. However, since the operators read the same forward and backward, we will refer to $DSKD$ as a "palindromic" integrator. We will refer to the traditional method of choosing a timestep at the beginning of the integration step as $SDKD$. Synchronization can be maintained by only allowing timesteps that are a power-of-two subdivision of the largest timestep, $\tau_s$. That is,

$$\tau_i = \frac{\tau_s}{2^{n_i}},$$

where $\tau_i$ is the timestep of a given particle, and $n_i$ is an integer (See Hernquist and Katz, 1989). Combining this synchronization procedure with the $DSKD$ evolution operation we have the following (recursive) algorithm for a timestep. 1) Drift the particles forward $\tau_i/2$. 2) Apply the select operator. If it accepts this step, then finish the step with $K(\tau_i)$ and $D(\tau_i/2)$. Otherwise, drift the particles back $\tau_i/2$ and take two timesteps using $\tau_i/2$. The algorithm is expressed as pseudo-code in Figure 3.2.

As an example of how this procedure works, consider Figure 3.3. In this diagram we consider the case where there are particles on three separate timesteps. The arcs represent

```
Timestep(τ)
{
    Drift(τ);
    Select(τ);
    if τ = τ_min then
    {
        Kick(τ);
        Drift(τ);
    }
    else
    {
        Drift(−τ);
        Timestep(τ/2);
        Kick(τ);
        Timestep(τ/2);
    }
}
```

Figure 3.2: Pseudo–code for the top–down timestepping algorithm. Here, `Drift(τ)` applies $D(\tau)$ to all particles, `Kick(τ)` applies $K(\tau)$ to particles with timestep $\tau$ and `Select()` decides to which timestep the particles belong, with the side effect of finding the minimum timestep, $\tau_{min}$.

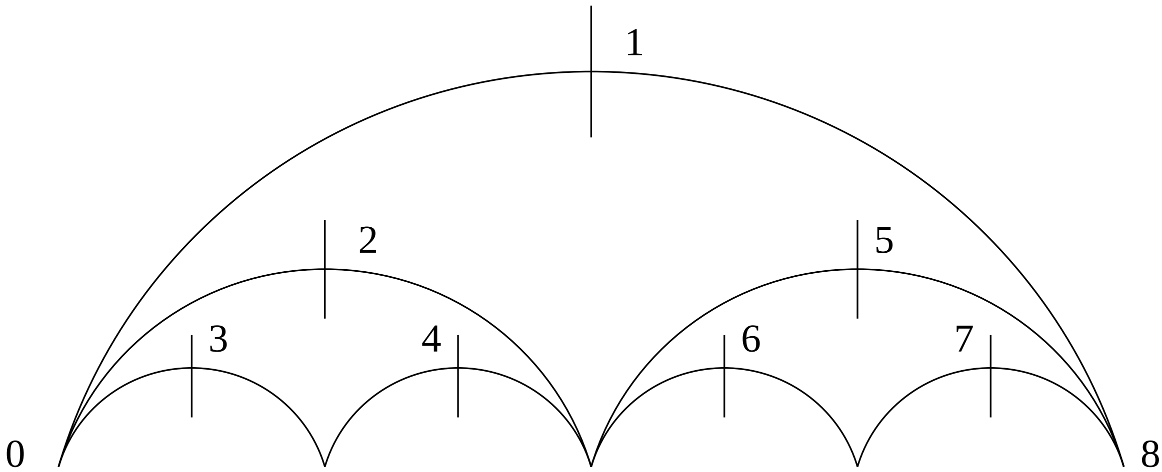


Figure 3.3: Diagram of particles on three separate timesteps, where arcs represent “drifts”, and vertical lines represent “kicks”. For $DSKD$, the order of flow through the diagram is 0, 1, 2, 3, 2, 4, 1, 5, 6, 5, 7, 8.

drifting and the vertical slashes represent either a select or a kick of particles at that timestep. Note that we always drift *all* of the particles, such that their positions are always time synchronized. Following the diagram, we first drift to 1 and evaluate which particles are on the largest timestep. We continue in the same way, drifting to 2 and 3, each time evaluating which particles belong to the given timestep. At this point in our example all particles will belong to one of the three timesteps and we can continue by applying a kick to the smallest timestep. We then drift from 3 to 2, where the middle timestep gets a kick and on to 4, where the smallest timestep gets its second kick. Drifting from 4 to 1, we see that we have completed a single middle timestep and two small timesteps. Now at 1, we apply the kick to the largest timestep. At this point we recognize the possibility that the distribution of particles amongst the middle and small timesteps can change, so we drift to 5 and again to 6 to determine this new distribution using the select operator. We will assume here that no particles were found to be on a yet smaller timestep, although this is possible. We continue as before, applying the appropriate kicks at points 6, 5, and 7 allowing us to finally complete a single large timestep.

There are several things to note about the method in general. Firstly, it is not unique.

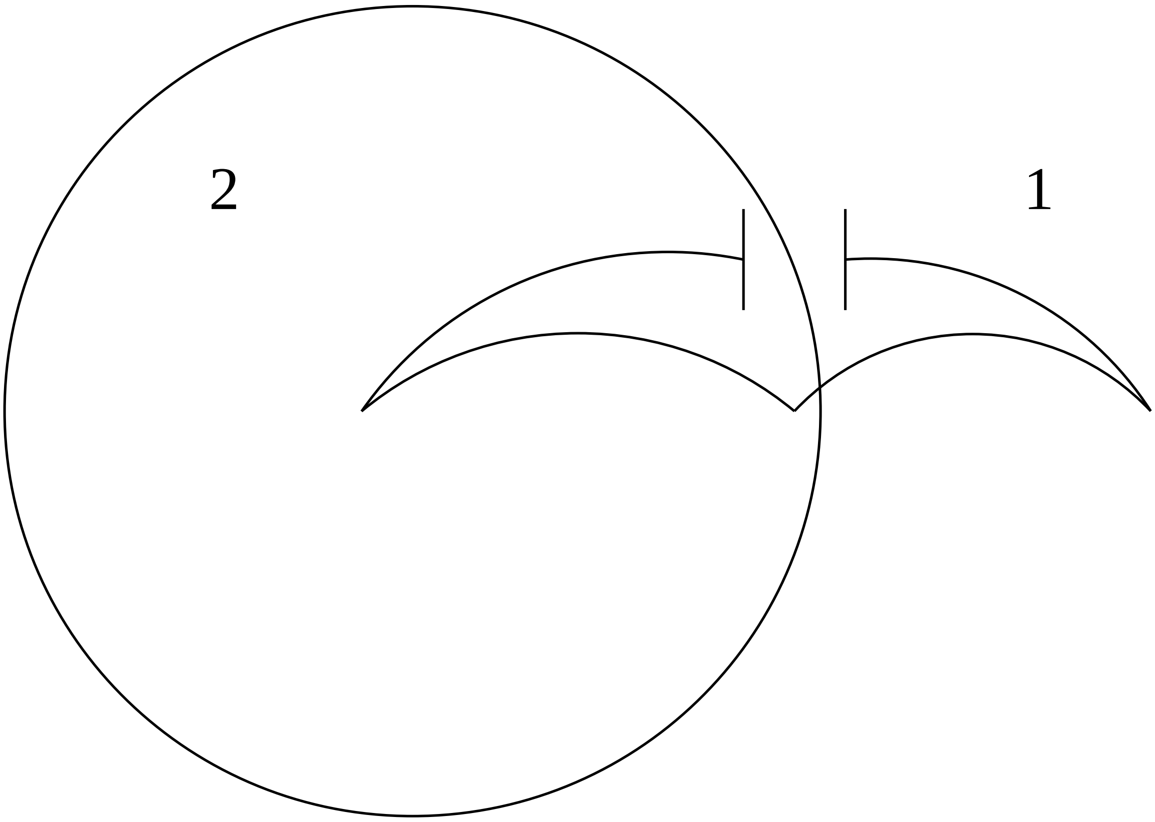


Figure 3.4: An example of a non-time-symmetric scenario is shown. If the step starts in region 2, two steps are taken. If the step starts in region 1, only one step is taken.

It uses a "top down" approach by trying a given largest timestep and reducing it until it is deemed appropriate. A "bottom up" scheme can be implemented where a given smallest timestep is tried first. Secondly, this algorithm is not exactly time reversible. Scenarios can easily be constructed where a forward step followed by a backward step will not come back to the initial conditions, as is shown in Figure 3.4. Suppose that interior to the circle the select operator will pick a timestep which is half of that required exterior to it. In the pathological case shown, a particle inside the circle evaluates the select operator at a point just interior to this region. This indicates that two smaller timesteps are to be taken, bringing the particle into the exterior region. From this point, we try to back up the particle using the method, but find that this time the select operator is applied just exterior to the region. Therefore, only a single larger step is taken going backward and we do not arrive at the point from which we started. Similar problems to this loss of reflexivity can occur in the iterative schemes (Hut, Makino, & McMillan 1995; Kahan 1993) if the starting guess of an iteration is preconditioned by the history of the particle. Finally, since several selects may be done for every kick, the efficiency of this algorithm depends upon `Select` being able to quickly decide appropriate timesteps given the particle positions.

### *3.3 Choosing Timesteps*

The operator formalism for describing leapfrog allows the derivation of a suitable timestep criterion by requiring the error Hamiltonian to be small compared to the actual Hamiltonian. For the DKD integrator the error Hamiltonian is (see Saha & Tremaine, 1992):

$$H_{err} = \frac{\tau^2}{12}\left\{\{H_D, H_K\}, H_K + \frac{1}{2}H_D\right\} + \mathcal{O}(\tau^4).$$

Using $H_D = \frac{1}{2}\mathbf{v}^2$, and $H_K = V(\mathbf{r})$, this evaluates to

$$H_{err} = \frac{\tau^2}{12}\left[-\frac{1}{2}v_i v_j \frac{\partial^2 V}{\partial r_i \partial r_j} + \frac{\partial V}{\partial r_i}\frac{\partial V}{\partial r_i}\right],$$

where implicit sums are taken over repeated indices. If we require that $|H_{err}|/|H|$ be less than some small parameter, $\eta^2$, then we have the criterion:

$$\tau^2 < 12\eta^2 \frac{|1/2\mathbf{v}^2 + V|}{\left|-\frac{1}{2}v_i v_j \frac{\partial^2 V}{\partial r_i \partial r_j} + \frac{\partial V}{\partial r_i}\frac{\partial V}{\partial r_i}\right|}.$$

If we evaluate this for a harmonic potential, we have

$$\tau^2 < 12\eta^2 \frac{|1/2\mathbf{v}^2 + V|}{|-2\pi G\rho\mathbf{v}^2 + \mathbf{a}^2|},$$

where $\rho$ is the mass density distribution generating the potential.

To implement the palindromic integrator, we require a timestep criterion that depends on only the positions. If we restrict ourselves accordingly, the above suggests the following criterion:

$$\tau < \eta \min\left(\frac{1}{\sqrt{G\rho}}, 3.5\frac{\sqrt{|V|}}{|\mathbf{a}|}\right). \tag{3.1}$$

The first term of this criterion is just the local dynamical time. The second poses two problems. A pragmatic one is that it is more expensive to calculate than the density; a more fundamental problem is that it is not gauge-invariant. A constant addition to the potential does not change the equations of motion, but does change the timestep criterion. Indeed, in cosmological simulations, the potential is not well defined at all, since it does not converge as one adds perturbations on larger scales with the standard Harrison-Zel'dovich power spectrum. However, in a typical non-cosmological N-body simulation without external potentials, the zero point of the potential is unambiguous, being the energy required to bring the particles in from infinity to their current configuration.

The above criteria differ significantly from the criteria commonly used in large simulations (*e.g.* see Katz, Weinberg & Hernquist, 1996), which are usually based on $\epsilon/v$ or $\sqrt{\epsilon/a}$, where $\epsilon$ is the length scale of the force softening used in the calculation. By using the force softening, both of these criteria assume that the dynamical times are set by two-body encounters. For a particle orbiting in a cluster, it is the cluster density rather than two-body encounters that set the dynamical time. Indeed, for large simulations that are trying to be collisionless, the two-body timescale should be irrelevant.

Because of its straightforward physical interpretation and relatively cheap cost, we will focus on the density criterion as we perform our tests. However we will also look at the other criteria as we run into problems with the density criterion.

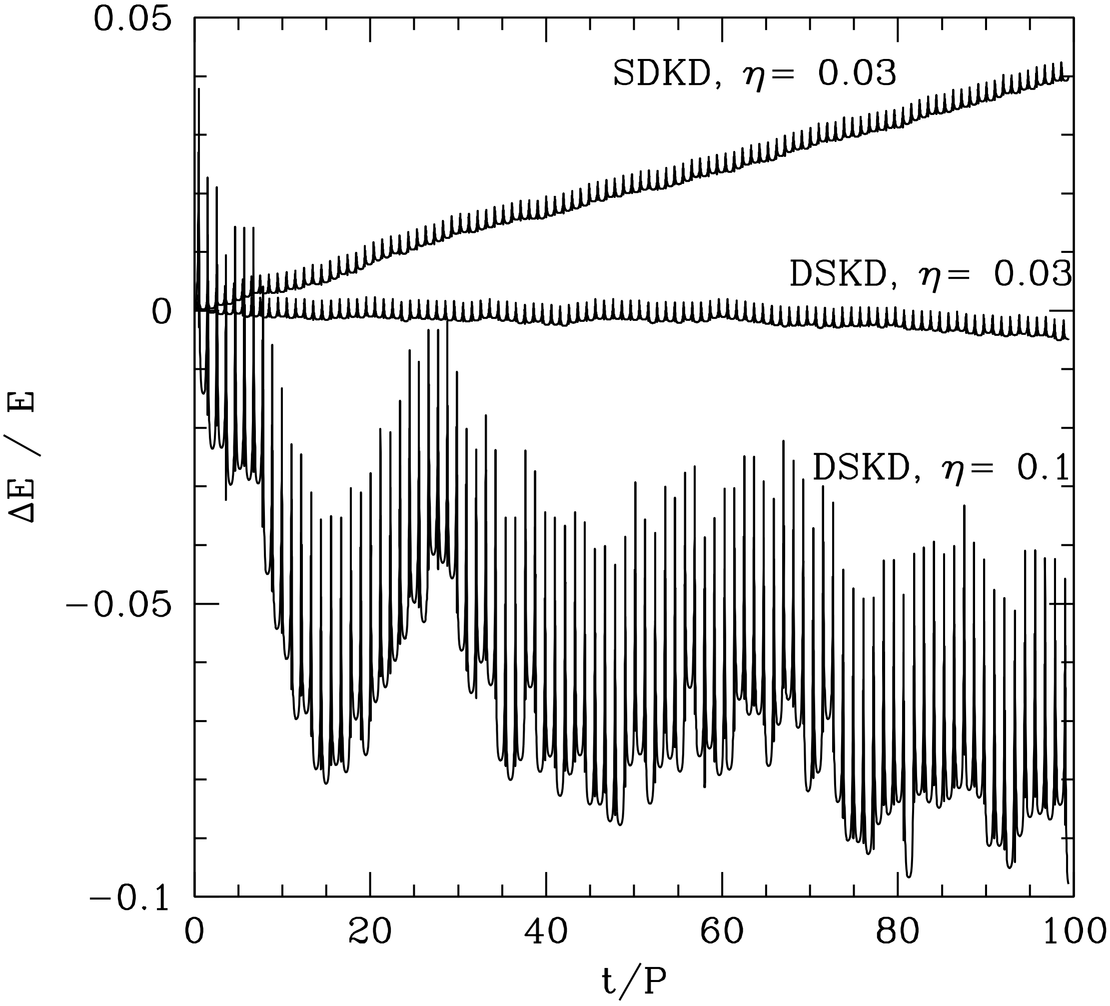


Figure 3.5: Relative change in total energy is plotted as a function of time in units of the orbital period for several integrators. Curves are plotted for $DSKD$ with $\eta = 0.1$ and $\eta = 0.03$, and $SDKD$ with $\eta = 0.03$.

### 3.4 Single Particle Tests

Our first test is the motion of a massless particle in the one dimensional effective potential of the Kepler problem. Since the local density in equation 3.1 is zero for the Kepler problem, we determine the timestep by using the enclosed density, $\rho_e = 3M/4\pi r^3$ where $M$ is the mass of the central object. Figure 3.5 shows the relative change in total energy as function of time of an eccentricity $e = 0.9$ orbit. It appears that the $DSKD$ method with $\eta = 0.03$ is not quite symplectic since there remains a small drift in the total energy, but it is stable, which cannot be claimed for $\eta = 0.1$. We also see that the $SDKD$ method with the same value of $\eta$ experiences a substantially larger drift in total energy. This difference between the two methods is even more striking when we compare them for orbits with much higher eccentricities. With the same value of $\eta$ (0.03) the $DSKD$ method only experiences a small drift as above, while the $SDKD$ method drifts in energy by a factor of 2.5 in 100 orbits for $e = 0.99$ and drifts by more than an order of magnitude in only 12 orbits for $e = 0.999$. The savings in force evaluations is also significant. Keeping energy errors comparable, fixed timesteps requires about 20 times as many force evaluations as adaptive timesteps for the $e = 0.9$ orbit.

A more realistic test is the motion of a massless particle in the potential of an isothermal sphere. As well as having a local density defined everywhere, the potential is similar to that of globular clusters, galaxies and clusters of galaxies. Figure 3.6 shows the relative change in total energy as function of time for a 3.2:1 (apocenter to pericenter) orbit, typical for orbits in an isotropic isothermal distribution (van den Bosch, et. al. 1998). The non-symmetric method shows a large secular drift in the energy, while in this case the symmetric method shows no appreciable drift in energy and the variations in energy are quasiperiodic, both characteristics of a symplectic method.

### 3.5 N-Body Tests

For our purposes, the real proof of a given integration method is how well it performs in the case of many mutually gravitating bodies. Although in practice the integration method will be used to study the complex evolution of self gravitating systems, it is useful to first

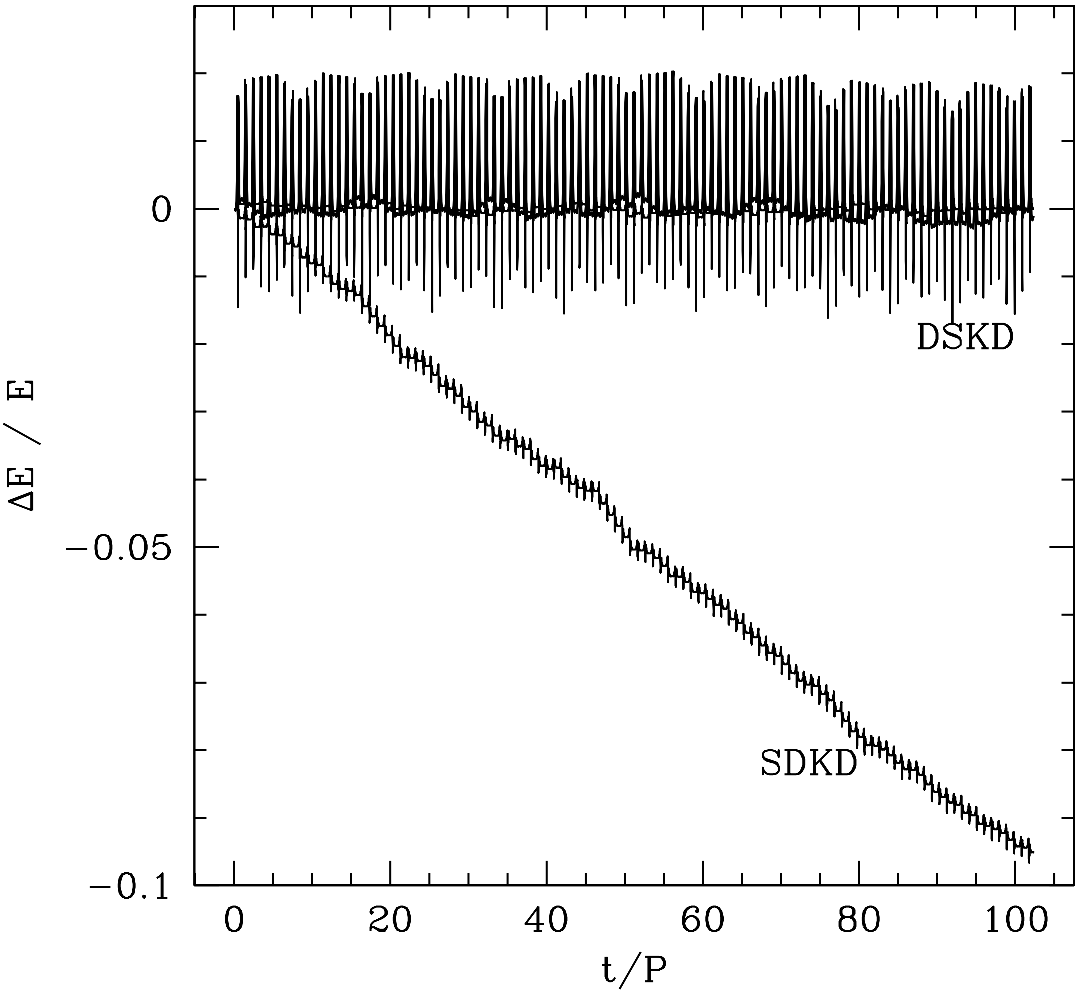


Figure 3.6: Relative change in total energy is plotted as a function of time in units of the orbital period for several integrators in an isothermal potential. Curves are plotted for $DSKD$ with $\eta = 0.1$ and $\eta = 0.03$, $SDKD$ with $\eta = 0.03$.

consider the case of a "non-evolving" N-body system. An example of such a system is a King model, whose distribution function, in the absence of collisional relaxation, should remain static. This King model test will act as a precursor to a real application of the method, namely to a cosmological simulation of a universe dominated by cold dark matter (CDM). This latter case is very dynamic, with structure evolving on all scales. The following tests used the parallel treecode, *PKDGRAV* (Stadel & Quinn, in preparation), which is ideally suited to performing very large cosmological N-body simulations. To mitigate the effect of force errors which could confuse the issue of integration errors, we have used 4th-order multipole expansions within this code as well as setting a very conservative *opening criterion* of $\theta = 0.55$ (a force accuracy parameter).

When considering such N-body systems we are immediately faced with the question of how we should evaluate the performance of a given integration method. Basing this evaluation solely on the conservation of energy, on stability or on quasiperiodic behavior seems to be inadequate and/or inappropriate for large N-body systems. The requirement that is relevant is the convergence of scientifically interesting quantities, and we will introduce and base our evaluations on these in the following tests. For the King model the radial density profile is a natural test of the integration method since it should remain constant in a collisionless simulation.

We will test a $W = 9$ King model realized with 100,000 particles evolved for 100 central dynamical times. This is representative of a galaxy halo over its lifetime in a typical cosmological N-body simulation. In Figure 3.7 we compare the density profile of the initial King model with that at the end of a simulation using fixed step leapfrog with timesteps of either 1.0, 0.3, 0.1, or 0.03 times the central dynamical time. From the figure, we see that $\tau/t_d < 0.1$ is needed to maintain the central density of the King model. Having established an appropriate fixed timestep for leapfrog we can now compare the performance of the adaptive timestepping method using the local density criterion with various values of $\eta$. We find that, as in the single particle case, $\eta \leq 0.03$ is needed to integrate this model with reasonable accuracy. We note in passing that both the fixed and adaptive methods require better than 3% total energy conservation to maintain the density profile over 100 dynamical times. We see that the $\eta$ parameter corresponds well to $\tau/t_d$. This is not surprising since the central dynamical time ($1/\sqrt{G\rho}$) should equal the orbital period to within a constant of order unity. At a comparable accuracy, the adaptive method performs roughly 5 times fewer force evaluations than fixed timestep leapfrog, reassuring us that considerable gains can still be made in the full N-body case.

Before testing timestepping in a cosmological simulation we first note that the standard leapfrog difference equations used in cosmological simulations with comoving coordinates are not done in canonical coordinates and are apparently not symplectic. However, as shown in the appendix, a symplectic integrator can easily be derived, for which the "drift" and "kick" operators are

$$\begin{aligned} D(\tau) &\equiv \mathbf{r'}_{t+\tau} = \mathbf{r'}_t + \mathbf{p'} \int_t^{t+\tau} \frac{dt}{a^2} \\ K(\tau) &\equiv \mathbf{p'}_{t+\tau} = \mathbf{p'}_t - \nabla'\phi' \int_t^{t+\tau} \frac{dt}{a}, \end{aligned}$$

where the variables are as defined in the appendix. We use this integrator in the tests that follow. This integrator has another advantage over the standard leapfrog difference equations in that it can be used to implement both the $DKD$ and $KDK$ form of leapfrog. The standard equations can only be derived for $DKD$. The $KDK$ form of leapfrog can have advantages over $DKD$ (see the discussion of costs in §5).

Energy conservation may be a an even poorer criterion for determining the accuracy of a cosmological integration. This owes to the possibility of making small errors in calculating the energy of the uniform background that can dominate the total error budget and mask any problems in the structure of interest. Nonetheless, it pays to examine the evolution of the energy. It was transients in the energy at the beginning of the cosmological simulations that made us suspect that the integration was not in canonical coordinates, leading to the results in Appendix A. But, these large transients in energy had little effect on the three dimensional structure, under-scoring our point that it may be a poor diagnostic. Small changes in the way that two simulations handle global energy may be an interesting clue to other aspects of the integrator.

Since there is no analytic model for the full three dimensional formation of non-linear cosmological structure, we test the goodness of a given timestepping algorithm by examining its convergence to a simulation with a large number of timesteps for all particles. This

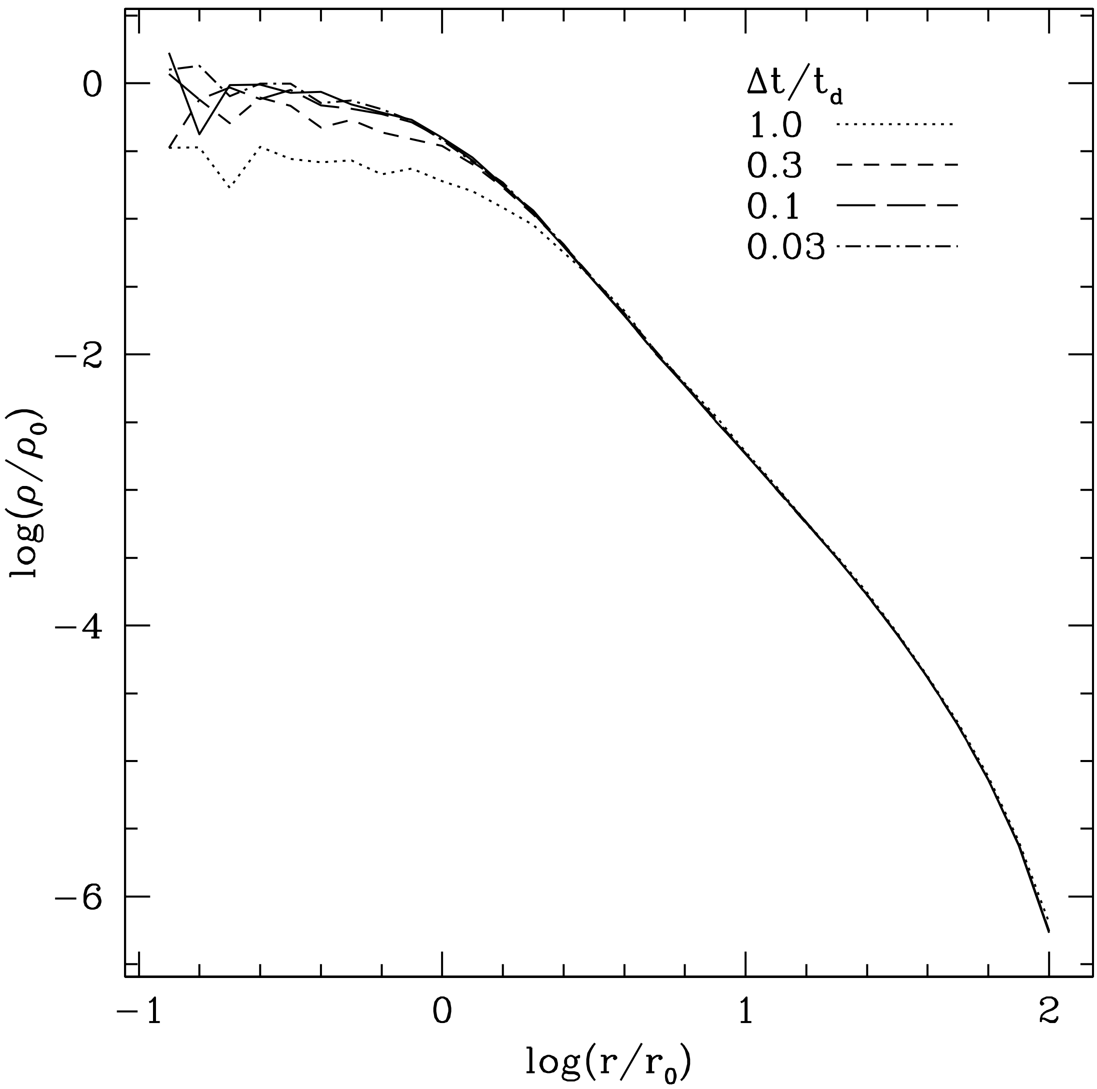


Figure 3.7: The spherically averaged density in units of the central density is plotted against radius in units of the core radius for the end state of several simulations. The solid line is the density profile of the initial King model. All integrations were done with a fixed step leapfrog integrator with the timestep in units of the central dynamical time given in the legend.

convergence will be assessed based on properties of the groups that are formed and the internal structure of the largest group.

The fiducial simulation is of a standard CDM dominated universe in a periodic box of size 22.2 Mpc ($h = 0.5$) on a side realized on a grid of $32^3$ particles. This implies a particle mass of $2.3 \times 10^{10}$ $M_{\odot}$. A spline softening is used with a softening length of 20 kpc. The initial conditions were imposed on the particles at a redshift of 49 using the Zel'dovich approximation. The small size of this box means it does not accurately represent large scale structure, but it does guarantee significant non-linear evolution and a stringent test for the time integrator.

The first convergence test is the density profile at the end of the simulation of the largest halo as identified using a friends-of-friends grouping algorithm. The object in question has a total mass of $1.23 \times 10^{14}$ $M_{\odot}$ within the virial radius and a maximum circular velocity of 639 kms$^{-1}$. This test looks at how the timestep algorithm affects the structure of well resolved objects in the simulation. Since structure forms hierarchically, the objects in the largest clusters will have experienced a complicated history with several dynamical times in previous generations of structure. Any effect that accumulates with dynamical time is most likely to betray itself in the center of the richest cluster.

Comparisons of the density profile of this object in various integrations are shown as differences in Figure 3.8. The fixed step run used 8000 steps in a Hubble time. The $DSKD$ and $SDKD$ runs used $\eta = 0.03$. We also performed simulations using the acceleration timestep criterion of $\tau < \eta\sqrt{\epsilon/a}$ with $\eta = .3$. The symmetric and non-symmetric versions of these integrations are indicated in the figures by *DSKD* (accel.) and *SDKD* (accel.), respectively. As might be expected, the timestep criterion used for the King model simulation performs well in accurately modeling the density profile of this cluster-size object. However, note that integrations using both versions of the density criterion and the SDKD version of the acceleration criterion show a reduced central density in comparison with the fixed step integrations. The increased density of these simulations at about .15 Mpc appears to be the spreading out of the mass that is in the central part of the fixed step simulation. Looser criteria ($\eta \geq 0.1$) fail to capture the central density of the object, just as in the static King model.

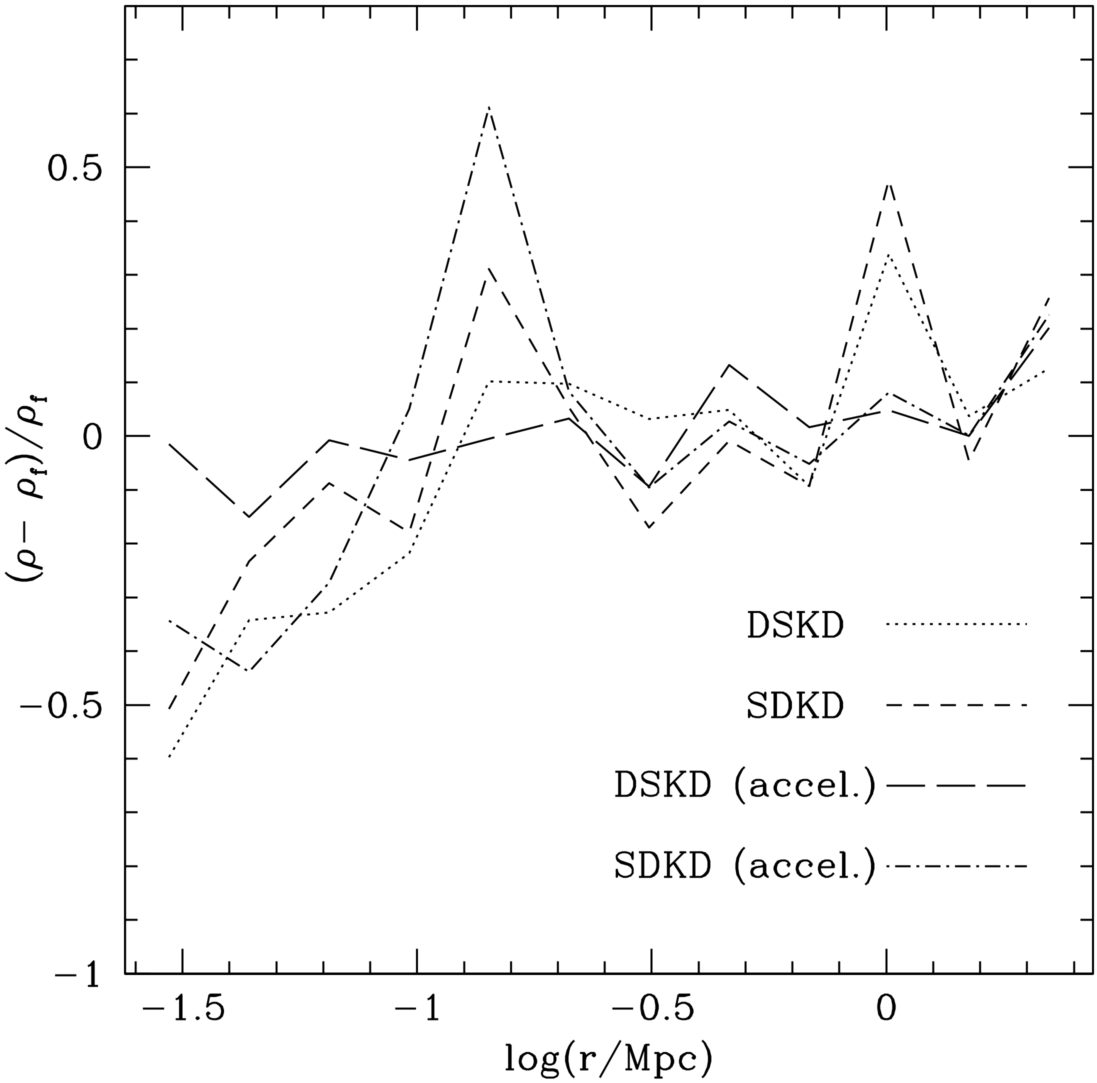


Figure 3.8: The difference in density averaged in spherical shells is plotted as a function of radius for the largest halo in the cosmological simulation. All integrations were compared with the fixed timestep integration.

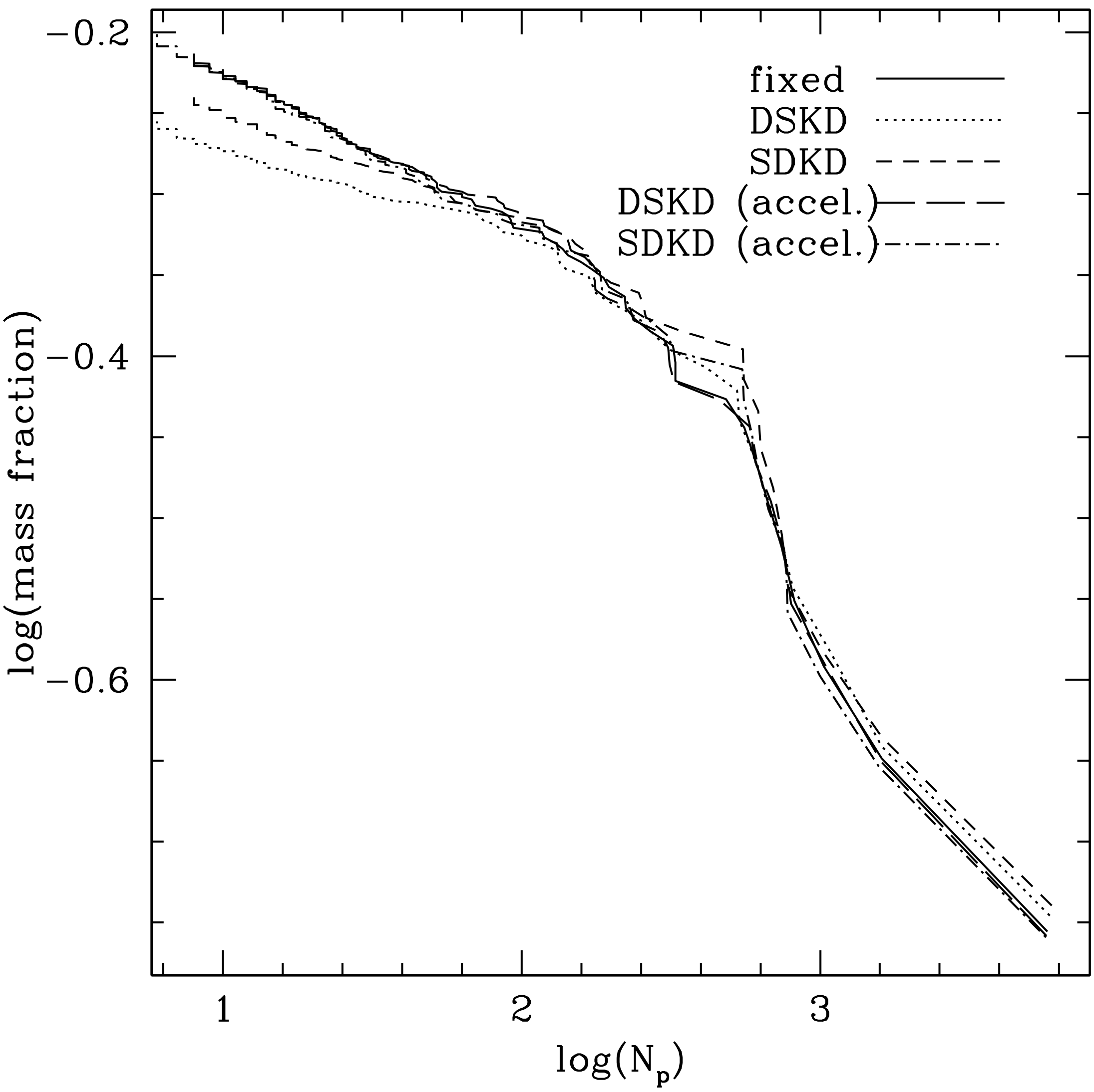


Figure 3.9: The cumulative mass fraction in groups of a given particle number or higher is plotted.

Our second criterion is the cumulative mass fraction in groups, also as identified by the friends-of-friends algorithm with a linking length of 0.26 times the mean interparticle separation. This linking length corresponds to an enclosed density equal to the virial density assuming an isothermal sphere. Here the motivation is to see the effect of the timestep algorithm on the smallest objects we can resolve. The cumulative mass fraction of halos is plotted in Figure 3.9 for the same simulations as shown in Figure 3.8.

For the smallest objects, the density criterion does not do well. There is a significant decrease in the number of objects formed with less than one hundred particles in the simulation using the density timestep criterion. It may not be surprising that particles residing in halos with comparable or fewer particles than that used by the smoothing kernel are unable to accurately estimate their local density. On the other hand, both simulations using the acceleration criterion reproduces the numbers of objects seen in the fixed step simulation all the way down to 8 particles. (The discrepancy seen at about 500 particles per group is due to a merger occurring at the end of the simulation.)

Let us compare the above results with commonly used criteria in cosmological simulations. In order to do this with different criteria, some assumptions need to be made. Consider a simulation in which the largest halo has a circular velocity, $v_c$, that is roughly constant with radius down to some core radius, $r_0$. If we wish to accurately model this halo, then the density timestep criterion must be

$$\Delta t = \frac{\eta}{\sqrt{G\rho_0}} = \frac{\eta r_0}{v_c}\sqrt{\frac{4\pi}{3}},$$

where $\rho_0$ is the (roughly constant) central density, and we have assumed spherical symmetry. Now consider a simulation with a halo with $v_c = 1000$ kms$^{-1}$ where our goal is to resolve its structure within 10 kpc. In units of the Hubble time, $1/H_0$, we have

$$H_0\Delta t \approx 1 \times 10^{-3}\eta \left(\frac{r_0}{10 \text{ kpc}}\right)\left(\frac{v_c}{1000 \text{ kms}^{-1}}\right)^{-1}.$$

For $\eta = 0.03$, this gives 33,000 steps per Hubble time. Previous estimates of the timestep conclude that $6000(10 \text{ kpc}/\epsilon)$ timesteps are needed per Hubble time, where $\epsilon$ is the gravitational softening length (Lake *et al.* 1995). This is consistent with our estimate if we assume that the gravitational softening is 1/5 the core radius we wish to resolve.

### 3.6 Performance

The large dynamic range in timescales present in most N-body systems indicates that significant performance gains should be realizable by using temporal adaptivity in the integration of these systems. In the prior sections we have characterized these performance gains by stating the reduction of the number of force evaluations over a comparable integration using fixed timesteps. The performance gains achievable by using a particular adaptive timestepping scheme are limited by the "fixed costs" required for any level of the timestep hierarchy. For example, a tree code requires that the tree is built for all the particles regardless of the number of particles requiring force evaluations. Indeed, for a standard particle mesh code, the forces at every grid point are either calculated or not. The only part of the force evaluation that is not part of the fixed cost would be the trivial interpolation of the force onto particles. In this case, essentially all the costs are fixed and no real gain is possible. In addition to these fixed costs, the method of determining the timestep level assigned to each particle can incur a significant cost and must also be accounted for. Finally, inefficiency in calculating forces for a few particles on modern pipelined, vector or parallel processors, can add an extra "hidden" cost to any adaptive timestepping scheme. In this section we will address each of these issues and provide implementation independent expressions which estimate the performance that can be realized by our adaptive leapfrog integration method.

For illustration, let us first examine the traditional, non–symmetric, $SDKD$ multistepping algorithm. Here we will consider only the fixed costs associated with the gravity calculation, given here by $c_f$, a ratio of these fixed costs to the cost of calculating the gravitational forces on all of the particles. A single base step with $r$ different timesteps separated by a factor of two, will have a cost of

$$C \sum_{i=1}^{r} 2^{i-1}(c_f + f_i),$$

where $f_i$ is the fraction of the total number of particles on a timestep $i$ of size $2^{1-i}$ times the base timestep. Note that $C$, the cost of calculating the gravitational forces on all of the particles, is an $O(NlogN)$ operation for N-body codes and that for treecodes, $c_f$ is dominated by the cost of building the gravity tree. For a fixed timestep simulation with all particles at the smallest timestep, the cost would be

$$C\, 2^{r-1}(c_f + 1).$$

Comparing these two costs, it is easy to see that there is a maximum speedup of $(c_f+1)/2c_f$ for large $r$, where almost all of the particles are on the largest timestep. Note that for an $SKDK$ scheme this maximum speedup would be a factor of two larger since gravity evaluations for larger timesteps are synchronized with those of smaller timesteps, and the same tree can be used for both. In the case of treecodes, we see that although the tree building costs are usually considered negligible when compared to the cost of calculating forces, for temporally adaptive methods these costs become much more significant to the total throughput of the code. Since these fixed costs place a limit on the gains achievable with a multistepping method, reducing them is well worth the effort, even where these costs are small in comparison to $C$.

We now consider the symmetric $DSKD$ scheme, taking into account the cost of the select operator,

$$C \sum_{i=1}^{r} \left[ 2^{i-1}((c_f + c_f') + (1 + 2c_s)f_i) - c_s f_i \right],$$

where $c_f'$ is the ratio of fixed cost associated with the select operator to the cost of calculating gravity on all particles, $c_s$ is the cost of evaluating the select operator relative to evaluating gravity, and $f_i$ is once again the fraction of the total number of particle on timestep rung $i$. For the density select operator $c_s$ is roughly 0.05 to 0.1 and $c_f'$ corresponds to the cost of building the nearest neighbor searching tree used in calculating the local density of particles. If the tree build costs and per particle costs for selecting the timestep and evaluating the forces are similar, then the symplectic scheme would be a factor of two more expensive than the non-symplectic scheme. This is the case for the cosmological integrations using the acceleration criterion described above where the maximum speedup is 11. However, for *PKDGRAV* running on a 100,000 particle King model, the per particle cost of the density criterion is a factor of 19 smaller than the per particle cost of a force evaluation, and the tree build for a density tree is 50% faster than for a gravity tree. The maximum speedup assuming an optimum particle distribution is then about 13.

For realistic particle distributions, the speed up can be much smaller, especially since a few particles (0.1% in the 100,000 particle King model) end up on timesteps smaller than the single stepping timestep. The speedup of the multistep run over the single step run as calculated from the above formulae is about a factor of 4. Note that we have neglected some costs such as particle pushing and the inefficiency of calculating forces for a few particles on modern pipelined processors.

Although the above discussion has focused on a particular implementation (*PKDGRAV*) of a particular algorithm (tree-codes), the results can be easily generalized to an arbitrary implementation. By determining fixed costs ($c_f$ and $c'_f$ above), and per particle costs ($c_s$ above), the above formulae can be used to determine the speedup of any given N-body integrator using our time stepping scheme.

For non-equilibrium situations such as cosmological simulations, a multistep scheme offers additional speedups since it can adapt to the changing time scales in the simulation. An example of the timestep evolution is shown in Figure 3.10. Here, the fraction of particles at a particular timestep or larger is plotted as a function of time. Note how almost all particles start out on a very small timestep at the beginning of the simulation when the Universe is very dense, and generally migrate to larger timesteps as the simulation progresses. One could make adjustments for this trend in a single step simulation by transforming to an appropriate variable, *e.g.* expansion factor instead of time (as in Efstathiou *et al.* 1985), but as can be seen from the figure, a simple monotonic transformation would not capture all the complexities of the changing time scales. Furthermore, one would need different transformations for different cosmological models. Finally, keeping the integrations in canonical coordinates would require that the gravitational softening length be fixed in coordinates that were different from either comoving or physical (see Section 3.8).

### *3.7 Discussion*

From our pragmatic standpoint as users of N-body simulations to model complex phenomena, the gain from symplectic integrators is their ability to tolerate greater truncation errors in numerical simulations of Hamiltonian systems. This, in turn, allows for longer

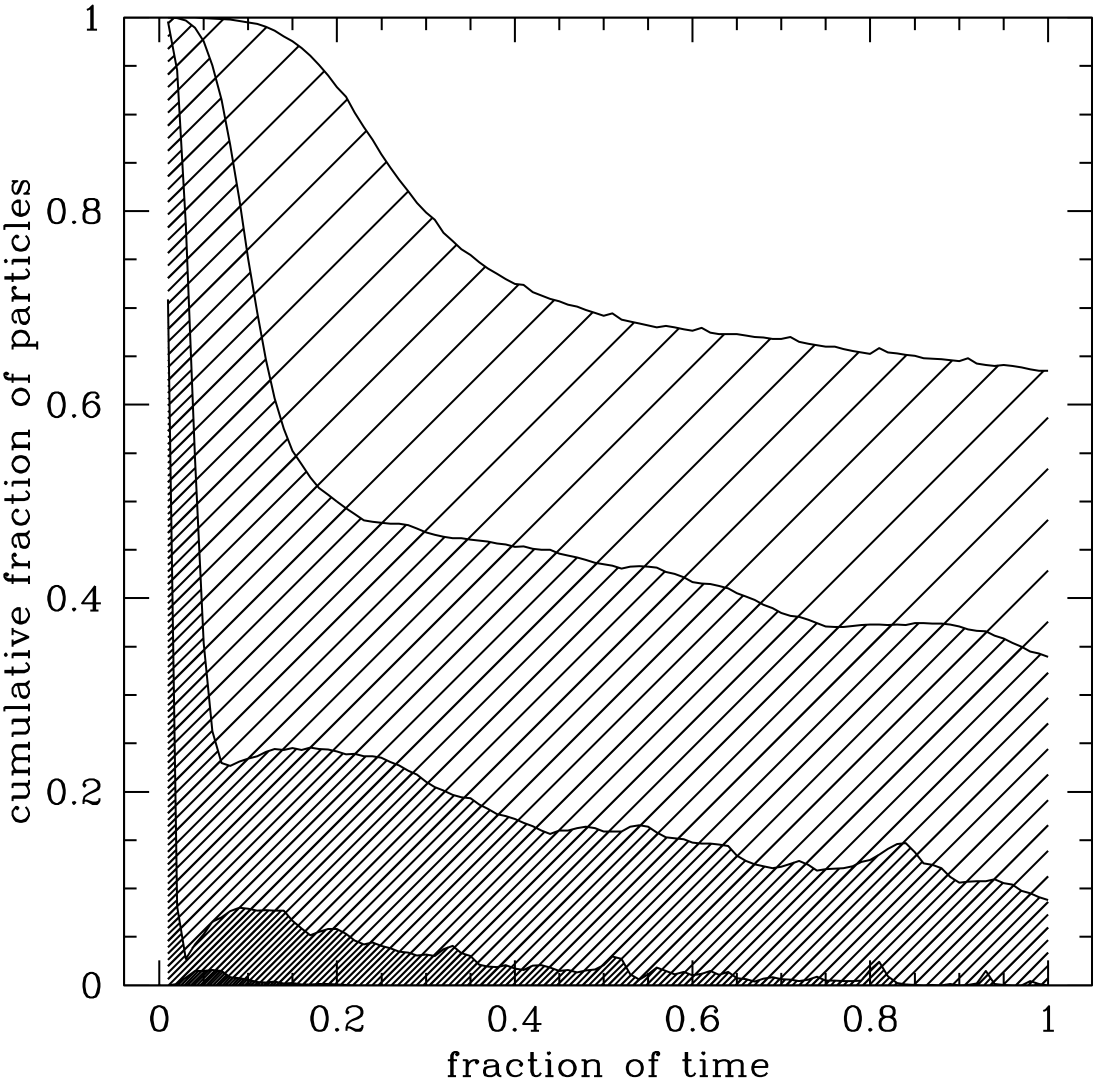


Figure 3.10: The cumulative fraction of particles on a given timestep or smaller is plotted as a fraction of the age of the Universe in a cosmological simulation. The lowest line is for the smallest timestep taken, and each upper line is for a factor of two larger timestep. The shading is proportional to the cost of integrating for a fixed time period.

timesteps and shortens the computer time needed to complete the simulation. Similarly, individual timesteps in a particle simulation allows one to concentrate the computational effort on those particles with the shortest dynamical times that require it. Either of these is beneficial only if their advantages outweigh their cost, whether in computational effort or algorithmic complexity.

For a system like the Solar System that is simulated for billions of dynamical times, a symplectic integrator has obvious advantages. Any dissipation introduced by truncation error has dire consequences that are easily observable—a planet may spiral into the Sun. The alternatives to symplectic integrators are either very high order integrators or extremely short timesteps, or both. Even a computationally expensive symplectic integrator can be an overall winner against these alternatives. (Wisdom and Holman, 1991; Saha and Tremaine, 1992.)

For other systems, such as galaxies or large scale structure, the situation is not so clear. These systems are simulated for at most a few hundred dynamical times, and drifts in conserved quantities due to truncation error will not be as noticeable. In these marginal cases, a symplectic integrator will only be beneficial if it is simple to implement and computationally cheap. In this paper we focus on a criterion based on the local density for this very reason: the algorithm for estimating the local density is very fast compared to calculating gravity. Unfortunately, this criterion is not ideal, particularly for cosmological simulations. On the other hand, the acceleration criterion works well in cosmological simulations, but the gains in multistepping can be canceled by the expense of evaluating the criterion.

Is there a better criterion? If we use the kick-drift-kick form of leapfrog, one could imagine a timestep criterion that depends only on the velocity, and therefore commutes with the drift operator. An example would be $\tau < \eta\epsilon/v$, where $\epsilon$ is the force softening. Three objections can immediately be raised against such a criterion. First, it is not Galilean invariant. A particle moving with respect to the coordinate system of the simulation will be on a different timestep than one that is at rest irrespective of the forces acting on the particle. Second, if the acceleration of a particle is nearly opposite to its velocity at the beginning of a timestep, then the timestep can be chosen such that the velocity is nearly zero at the middle of the timestep where the `Select` decision is made. Such particles will thereby be always put on a very long timestep. Third, the choice of $\epsilon$ to set the scale assumes that the dynamical times are set by two-body encounters. For a particle orbiting in a cluster, it is the cluster density rather than two-body encounters that set the dynamical time.

On the other hand, in a cluster with significant substructure the timestep should be set by the encounters with this substructure. In this case, $\epsilon/v$ might be an accurate estimate of the timestep. To evaluate whether such a criterion is necessary to resolve high speed encounters we ran tests of energy conservation for hyperbolic encounters between two particles. For our test we tracked the energy conservation of the orbit of a test particle in the spline softened Kepler potential. The test particle started out at about 20 softenings away with an impact parameter of one softening length and an initial velocity of between 1 and 64 times the escape velocity at twice the softening radius, $v_{e2h}$, in factor of 2 increments. We used both the $DSKD$ and $SDKD$ versions of our integrator with the enclosed density and $\eta = 0.03$, and we used an integrator with a timestep criterion such that $\tau < \eta\epsilon/v$ with $\eta = .3$. For each initial velocity we calculated 100 orbits, each starting at a slightly different phase along the orbit in order to find the worst case relative energy error. For both the $DSKD$ and $SDKD$ integrators, the worst case error is the largest at an initial velocity of about $16v_{e2h}$, getting smaller for velocities both larger and smaller than this. For $DSKD$ the worst case relative error was .2% while for $SDKD$, the error was 4%. Furthermore, for $DSKD$ the errors were symmetrically distributed around 0.0 whereas the $SDKD$ errors were skewed to one side. For the initial velocity of $16v_{e2h}$ the mean relative error across phases for the $DSKD$ integration was $5 \times 10^{-6}$ (consistent with round-off) while for $SDKD$, the mean was .4%. The integrator with the $\epsilon/v$ criterion also showed a worst case error, but at only $2v_{e2h}$ where the maximum relative energy error was .09%. At an initial velocity of $4v_{e2h}$ and above, this integrator had energy errors consistent with round-off.

Whether the errors seen in these high velocity tests can be tolerated in a simulation depends upon the scientific goals of the simulation. However, some general statements can be made: 1) The trend toward lower errors at high velocity can be derived from the impulse approximation. In this limit, the amount of energy transferred in the collision goes down with increasing velocity, and one would expect any integrator to do better. In fact the $\epsilon/v$

criterion seems to be overkill for very large velocities. With an initial velocity of $64v_{e2h}$ the $eps/v$ criterion requires as many as 1000 steps to complete the integration whereas the density criterion requires only 24 steps. 2) The $DSKD$ method proves to be significantly better than $SDKD$ even in the case of a single high speed collision. The reason for this is that the $DSKD$ method selects the timestep at the point which the force will be evaluated. If this point happens to lie very close to the closest approach, then a suitably small timestep will be selected. On the other hand, the point at which $SDKD$ selects the timestep can be significantly different than the point at which the force is calculated. 3) The symplectic nature of $DSKD$ becomes apparent when one looks at an ensemble of collisions at different phases. For the same reason a symplectic integrator shows no energy drift over time it also shows no energy drift over phases.

Based on the success of the acceleration criterion, one could also imagine criteria based on higher order terms, such as the magnitude of the divergence of the acceleration, or the time derivative of acceleration, but these tend to be even more expensive to calculate than the acceleration. However, according to the cost calculations above, it is possible to achieve significant speedups using $DSKD$ and an acceleration criterion in an ideal situation (most particles on the largest timestep).

Although our $DSKD$ method performs well both in single particle tests and our example N-body problems, it is not clear whether the extra computational cost incurred is worth it. We investigated the local density criterion extensively because it is cheap, but we found that there were instances where it did not select a proper timestep. This problem was foreshadowed by the inability to define the local density in the Kepler case and having to resort to a global criterion that used the enclosed density. When calculated in a cosmological simulation by averaging over the nearest $m$ particles, it erased substructure which consisted of somewhat less than $m$ particles. On the other hand, an acceleration criterion does well in a cosmological simulation, but at significant cost. We have not exhausted the search for an ideal timestep criterion, and there yet may be something that is a good compromise. Nevertheless, the criterion we have explored in this paper may be very useful in simulations where structure is defined by large numbers of particles, such as in the modeling of galactic structure.

For more general cosmological N-body simulations we advocate either the $SDKD$ or $SKDK$ method using $\eta\sqrt{\epsilon/a}$ to choose the timestep. Using tests similar to those described for our symplectic integrator we determined that an appropriate choice for $\eta$ is 0.3 (for Plummer softening 0.4) to insure stability of the integration. Although this algorithm is not symplectic, it appears to give accurate results for quantities of interest in a cosmological N-body simulation. Note that this timestepping method is subtly different from previous methods in the literature. In particular, we do not make the correction described in section II.c.v of Hernquist and Katz (1989). The difference arises because in our formulation of leapfrog, which defines the positions and velocities to be known at the same time, rather than offset by a half timestep as in the traditional formulation.

This work was supported in part by NASA HPCC/ESS grant NAG 5-2213 and by NASA Astrophysics Theory grants NAGW-2523 and NAG 5-4242. We also wish to acknowledge useful discussions with Scott Tremaine.

### *3.8 Appendix: Symplectic Integrators in Comoving Coordinates*

The equations of motion in a comoving coordinate frame are traditionally presented as

$$\begin{aligned}
&\dot{\mathbf{v}}' + 2H(t)\mathbf{v}' = -\frac{\nabla'\phi'}{a^3} \\
&\dot{\mathbf{r}}' = \mathbf{v}' \\
&\nabla'^2\phi' = 4\pi G(\rho' - \rho'_b),
\end{aligned}$$

where the primed quantities refer to the comoving coordinate frame, $a$ is the expansion factor, $H$ is Hubble's constant, and $\rho'_b$ is the mean background density. These can be integrated using the difference equations

$$\begin{aligned}
\mathbf{r}'_{n+1/2} &= \mathbf{r}'_n + \tfrac{1}{2}\tau\mathbf{v}'_n \\
\mathbf{v}'_{n+1} &= \mathbf{v}'_n\frac{1 - H(t)\tau}{1 + H(t)\tau} + \frac{\nabla'\phi'(\mathbf{r}'_{n+1/2})\tau}{a^3(1 + H(t)\tau)} \\
\mathbf{r}'_{n+1} &= \mathbf{r}'_{n+1/2} + \tfrac{1}{2}\tau\mathbf{v}'_{n+1}.
\end{aligned}$$

(See Hockney and Eastwood, 1981.) However, these equations do not appear to be symplectic, *i.e.*, there is no known generating function that produces this transformation.

However, by making a suitable canonical transformation, one can easily derive an integrator that is symplectic. The Lagrangian for the particle motion in the comoving frame is

$$\mathcal{L} = \tfrac{1}{2}(a\mathbf{v}' + \dot{a}\mathbf{r}')^2 - \phi.$$

With the (time dependent) generating function $\psi = (1/2)a\dot{a}\mathbf{r}'^2$, this transforms to

$$\mathcal{L} = \tfrac{1}{2}a^2\mathbf{v}'^2 - \phi'/a,$$

where

$$\phi' = a\phi + \tfrac{1}{2}\ddot{a}a^2\mathbf{r}'^2.$$

(See Peebles, 1980.)

Switching to the Hamiltonian formalism, the momentum canonical to $\mathbf{r}'$ is $\mathbf{p}' = a^2\mathbf{v}'$, and the Hamiltonian is

$$H = \frac{\mathbf{p}'^2}{2a^2} + \frac{\phi'}{a}.$$

Although this Hamiltonian is time-dependent[2], it is separable, so the "drift" and "kick" operators are easily derived as:

$$\begin{aligned} D(\tau) &\equiv \mathbf{r}'_{t+\tau} = \mathbf{r}'_t + \mathbf{p}' \int_t^{t+\tau} \frac{dt}{a^2} \\ K(\tau) &\equiv \mathbf{p}'_{t+\tau} = \mathbf{p}'_t - \nabla'\phi' \int_t^{t+\tau} \frac{dt}{a}, \end{aligned}$$

where $\tau$ is the step size. Note that it is assumed here that there is no explicit time dependence in $\phi'$; *e.g.* any softening of the potential must be constant in comoving coordinates. For standard cosmologies the integrals in the above operators can be easily evaluated. For example in a critical density universe,

$$\int_t^{t+\tau} \frac{dt}{a^2} = \frac{2}{H_0}\left[a^{-1/2}(t) - a^{-1/2}(t+\tau)\right],$$

and

$$\int_t^{t+\tau} \frac{dt}{a} = \frac{2}{H_0}\left[a^{1/2}(t+\tau) - a^{1/2}(t)\right].$$

[2] An energy can still be defined, as in the Layzer-Irvine energy equation. See Peebles, 1980.

For non-flat matter dominated universes, it is convenient to use the parametric solutions (Peebles, 1980),

$$a = A(1 - \cos\eta), \qquad t = B(\eta - \sin\eta); \qquad \Omega > 1$$
$$a = A(\cosh\eta - 1), \qquad t = B(\sinh\eta - \eta); \qquad \Omega < 1,$$

where the constants $A$ and $B$ are

$$A = 4/3\pi G\rho_b a^3 |R|^2, \quad B = A|R|,$$

$R$ is the curvature radius and the integrals are

$$\int_t^{t+\tau} \frac{dt}{a} = \frac{B}{A}\left[\eta(t+\tau) - \eta(t)\right],$$

and

$$\begin{aligned}\int_t^{t+\tau} \frac{dt}{a^2} &= \frac{B}{A^2}\left[\cot\frac{\eta(t)}{2} - \cot\frac{\eta(t+\tau)}{2}\right]; \qquad \Omega > 1 \\ &= \frac{B}{A^2}\left[\coth\frac{\eta(t)}{2} - \coth\frac{\eta(t+\tau)}{2}\right]; \quad \Omega < 1.\end{aligned}$$

The above operators can be used to build leapfrog or higher order symplectic integrators for particle motion in comoving coordinates.

For the leapfrog case, the timestep criteria introduced in section 3 need to be modified in the obvious ways. The criteria become:

$$\tau < \eta a^{3/2} \min\left(\frac{1}{\sqrt{G(\rho' - \rho_b')}}, 3.5\frac{\sqrt{|\phi'|}}{|\nabla\phi'|}\right).$$

This criterion compares well with the timestepping of cosmological simulations investigated by Efstathiou *et al.*, 1985. They investigated a timestep of the form $\Delta p = C$, where $C$ is a constant and $p = a^\alpha$. For constant $\rho'$ or $\sqrt{\phi}/\nabla\phi$, our timestep criterion is equivalent to $\alpha = 0$, which is what Efstathiou *et al.* derive for an orbiting binary pair at constant comoving separation. For the growing mode of a perturbation in a critical density universe, $\rho' - \rho_b'$ grows as $a$, which is equivalent to $\alpha = 1/2$. Efstathiou *et al.* found $p = a^{1/2}$ was one of the better choices for a timestep variable in terms of conservation of energy.

Chapter 4

# SKID: FINDING GROUPS IN N–BODY SIMULATIONS

## *4.1 Introduction*

A major problem with the interpretation of cosmological N–body simulations has always been identifying the positions and velocities of the "galaxies" in the simulation. Observational constraints almost always concern the position and velocities of galaxies, not of the dark matter background. Although the galaxy identification problem is particularly acute in simulations that only contain a dark matter component, the problem still exists to a lesser degree in simulations that contain a gas component and/or a stellar component. One still needs a dependable method to group together the particles into "galaxies".

The most common approach is the so called friends–of–friends (FOF) algorithm. The algorithm recursively links together all particles within a distance $b$ of each other. $b$ is the linking length and is a fraction of the mean interparticle separation. On average the friends–of–friends algorithm links together all particles within an isodensity contour of $1/b^3$ times the mean density. There are two main difficulties with the friends–of–friends algorithm. First, the appropriate choice of $b$ is somewhat arbitrary. Second, no global choice of $b$ seems to work well. A large $b$ allows one to find groups (in this discussion "groups" refers to groups of particles) that clearly exist in the particle distribution in regions where the mean background density is low but links together obviously separate groups in regions where the mean background density is high. A small $b$ suffers from the opposite problem, it works well in regions where the mean background density is high but not where it is low.

The DENMAX algorithm (Bertschinger and Gelb 1991, Gelb 1992) avoids many of these difficulties. Groups are centered on density maxima in the particle distribution and all the particles within the last closed isodensity contour are assigned to the group. However, the implementation of this simple idea is quite difficult. The density throughout the volume is first determined on a regular rectangular grid. This is accomplished by assigning a fraction of each particle's mass to the 27 grid points surrounding the grid point nearest the particle using the triangular cloud algorithm. Then the densities are smoothed using a Gaussian smoothing kernel with a smoothing radius appropriate for galaxies. The gradients of the density at each grid point are then determined using the same triangular cloud interpolation by using the derivatives of the triangular cloud weights. Particles are slowly slid along these density gradients until they reach the density maxima. As described below, unbound group members are then eliminated through an iterative process.

We propose a variant of the DENMAX algorithm, SKID (Spline Kernel Interpolative Denmax), which uses an SPH–like approach to calculate densities and density gradients. The advantage to this is that simulations which exhibit an extremely large dynamic range in the scale of structures present can be analyzed without having to set a fixed grid resolution which would ultimately set a resolution below which structures would be smoothed out. SKID relies on a fast nearest neighbor searching algorithm based on a k-D tree structure which forms a hierarchical representation of the mass distribution. The smoothing scale associated with a given particle will be given by its nearest $N_{smooth}$ neighbors. This approach allows SKID to dispense with the grid altogether and makes it independent of any artificially set smoothing scale. This approach is more compatible with the Lagrangian nature of N-body simulations. Furthermore, for large simulations the algorithm outlined below is much faster than the original DENMAX group finder.

## *4.2 General Description*

The Denmax method of finding groups can be likened to finding watershed regions given a topographical map. In the case of an N-body snapshot our valleys are regions of high density and our hills, regions of low density. Instead of water flowing, we have particles flowing along the local density gradient[1], toward regions of higher density. We wish to associate, or group, particles which converge to a common high density region when allowed

[1]This is the density gradient of the snapshot's initial configuration, i.e., we keep the "topography" fixed while the particles are allowed to flow.

to flow. Once all the particles have been grouped in this manner we remove particles which are not gravitationally bound to their group. This produces groups whose member particles are both geometrically and gravitationally associated with each other.

SKID can be broken up into several general phases of computation. Some of these phases vary somewhat depending on the type of input that is being analyzed. Three different types of particles are recognized by SKID, these being **gas particles**, **dark matter particles** and **star particles**. Although this results in seven possible input types, the essential differences in the way SKID finds groups are a result of the presence of gas particles in the input. For example, in order for a galaxy to be gravitationally bound, we must take into consideration its dark matter halo. Thus, when dark matter and gas are present, SKID must take into account some of a group's dark matter environment in deciding whether its gas particle members are bound to it. This consideration is made during the **unbinding** phase, which is described in more detail below. In addition, the type of input determines which particles are considered for grouping and also how densities and density gradients are calculated. The following lists the different types of input and a brief description of how SKID treats each one.

| Skid Grouping Rules | | | | |
|---|---|---|---|---|
| particles | group | cut criterion | gradient | unbind |
| dark only | dark | $\rho_{\mathrm{dark}} \geq \rho_{\mathrm{min}}$ | $\nabla\rho_{\mathrm{dark}}$ | Type I |
| gas only | gas | $\rho_{\mathrm{gas}} \geq \rho_{\mathrm{min}}$<br>$T_{\mathrm{gas}} \leq T_{\mathrm{max}}$ | $\nabla\rho_{\mathrm{gas}}$ | Type I |
| gas & dark | gas | $\rho_{\mathrm{gas}} \geq \rho_{\mathrm{min}}$<br>$T_{\mathrm{gas}} \leq T_{\mathrm{max}}$ | $\nabla\rho_{\mathrm{gas}}$ | Type II |
| star only | star | none | $\nabla\rho_{\mathrm{star}}$ | Type I |
| star & dark | star | none | $\nabla\rho_{\mathrm{star}}$ | Type II |
| star & gas | star | none | $\nabla\rho_{\mathrm{star}}$ | Type II |
| star, gas & dark | star<br>gas | none<br>$\rho_{\mathrm{star,gas}} \geq \rho_{\mathrm{min}}$<br>$T_{\mathrm{gas}} \leq T_{\mathrm{max}}$ | $\nabla\rho_{\mathrm{star,gas}}$ | Type II |

The first column contains the various possible input file type in terms of the type of particles that are present. The second column indicates which type of particle will be considered for grouping. This is followed by a column indicating which of these particles will be further considered for grouping. The fourth column indicates which particles will determine the local gradient indicating the direction towards higher density. The final column indicates the type of unbinding procedure that will be used to remove particles from the groups such that only gravitationally bound particles form the final group. How each of these factors enters into each phase of the SKID group finding process will be described in what follows.

### *4.2.1 Density Calculation*

The first phase of SKID calculates the densities of all particles which are to be considered for grouping (as outlined above for the different input types). We will refer to these particles as the *active particles* in the subsequent discussion. First we determine the smoothing length, $h$, of each active particle. The smoothing length of a particle is chosen such that there are $N_{smooth}$ other active particles within a sphere of radius $2h$. $N_{smooth}$ is an input parameter typically chosen to be between 16 and 256, although these are in no way hard limits for the algorithm. A larger value for $N_{smooth}$ decreases the noise in the estimates of the density and the density gradient but also degrades the resolution. Generally using 32 nearest neighbors is appropriate for SKID.

Once $h$ has been determined for each active particle, a density is assigned to each of these active particles using the combined gather and scatter approach as described in Hernquist and Katz (1989). The density for each active particle is thus given by,

$$\rho(\mathbf{r}_i) = \sum_{j=1}^{N} m_j \frac{1}{2} \left[ W(r_{ij}, h_i) + W(r_{ij}, h_j) \right]. \tag{4.1}$$

The sum is over all the active neighbors of particle $i$. $m_j$ is the mass of the $j^{th}$ particle and $W(r,h)$ is the SPH smoothing kernel. This is given by,

$$W(r,h) = \frac{1}{\pi h^3} \begin{cases} 1 - \frac{3}{2}\left(\frac{r}{h}\right)^2 + \frac{3}{4}\left(\frac{r}{h}\right)^3 & 0 \le \frac{r}{h} \le 1 \\ \frac{1}{4}\left(2 - \frac{r}{h}\right)^3 & 1 \le \frac{r}{h} \le 2 \\ 0 & \frac{r}{h} \ge 2 \end{cases} \tag{4.2}$$

It is worth noting that densities are always calculated, even where they are not used as is the case for star particles. This is done because we need to know the $h$'s for all the active particles in the subsequent density gradient calculations.

### *4.2.2 Density/Temperature Cut*

Once the density for the active particles has been determined we may want to only consider particles above some density threshold. This mainly done for performance reasons, since particles that are not part of a dark matter halo will later be removed in the unbinding phase. When finding dark matter halos setting the density threshold to the local density at the viral radius, or somewhat lower is sensible. The local over density at the viral radius of an isolated halo is about 55 and this is the value that is typically used. When we are grouping gas particles we may also want to set a maximum temperature, so that only cold gas is grouped. The temperature is assumed to be provided in the input for the gas particles.

### *4.2.3 "Flowing" Tracer Particles*

At this point we create a massless tracer particle for each active particle which passed the density/temperature cut. The initial position of each tracer particle is set to the position of its associated active particle. We now push the tracer particles along the density gradient, toward higher densities, in small increments. This involves calculating the density gradient contributed by the active particles (which are kept in the initial configuration), on the tracer particles. Use of the gather–scatter approach in calculating the density gradient at each tracer particle is not necessary since they are massless, instead we simply scatter from *all* of the active particles to the tracer particles. We use the same method of calculating the gradient of the density as described by Hernquist and Katz (1989), where the gradient of the smoothing kernel is used to estimate gradients of local fluid properties. For the gradient of the local density this has the particularly simple form,

$$\nabla\rho(\mathbf{r}) = \sum_{j=1}^{N} m_j \nabla W(\mathbf{r} - \mathbf{r}_j, h_j). \tag{4.3}$$

The sum here is over all the active particles and $\nabla W(\mathbf{r}, h)$ is the gradient of the SPH smoothing kernel used above. This is given by,

$$\nabla W(\mathbf{r}, h) = \frac{\mathbf{r}}{\pi h^5} \begin{cases} -3 + \frac{9}{4}\left(\frac{r}{h}\right) & 0 \leq \frac{r}{h} \leq 1 \\ -3\left(\frac{r}{h}\right)^{-1} + 3 - \frac{3}{4}\left(\frac{r}{h}\right) & 1 \leq \frac{r}{h} \leq 2 \\ 0 & \frac{r}{h} \geq 2 \end{cases} . \tag{4.4}$$

In this scatter approach, the smoothing length used for each active particle is the one found during the initial density calculation, since we wish to keep the gradient at each point in space fixed. This also means that we do not need to do full nearest neighbor searching, instead we simply find all tracer particles in a ball of radius $2h$, about each active particle. The density gradient evaluation must be performed at each new position of a tracer particle, as it is pushed toward a local maxima in density. We now describe this iterative procedure in more detail.

A fundamental parameter to the way SKID finds groups, is the approximate resolution limit of the simulation, $\epsilon$. This is typically set to the gravitational softening length that was used in the simulation. With each iteration the tracer particles are pushed a distance of $\epsilon$ in the direction of the density gradient. If a tracer particle is within a distance $2\epsilon$ of where it was 5 iterations before, we no longer bother to push this particle as it is then oscillating about a local maxima in density. This reduces the computing cost of the iterations as more and more of the tracer particles are localized. We simply continue this iterative process until there are no tracer particles left to push. At this point all of the tracer particles have been localized into roughly $2\epsilon$–sized balls about their local maxima in density.

One further optimization is employed during this phase of Denmax, in the case where the input contains only dark matter particles. Active particles which do not scatter to any tracer particle in the *first* gradient evaluation, will never again need to scatter. The reason for this is because the tracer particles will always be pushed toward regions of higher density, such that the surfaces bounding them at any iteration are always enclosed in the surfaces bounding them in their initial configuration. This is not true of gas particles, for which a maximum temperature criterion is included in selecting the tracer particles. For the dark matter only case, this significantly reduces the number of active particles which will need to scatter during subsequent iterations. Note, however, that it is not correct to continue reducing the number of scatterers with each iteration, since only the initial positions of the tracer particles can be considered physical.

### *4.2.4 Grouping particles*

At this stage we are ready to form the initial groups. Here we simply use the friends–of–friends algorithm with a linking length of $4\epsilon$ on all of the localized tracer particles. Each tracer particle belonging to a group will be within $4\epsilon$ of some other member of its group. In this way, each tracer particle is assigned a group number, which then also becomes the group number of its corresponding active particle in the initial configuration. Groups with less than a certain specified minimum number of members are discarded. It is important to stress that the friends–of–friends algorithm was chosen at this point purely for convenience. Since the tracer particles are already localized within balls of roughly $2\epsilon$ any simple prescription for associating them with each other is sufficient here. Figure 4.1 shows particles which SKID has grouped together with lines connecting the initial and final positions of the tracer particles.

### *4.2.5 Finding the center of each group*

Although we have fairly localized regions about the local density maxima, we sometimes wish to refine our estimate for the centers the density maxima somewhat. When determining if gas particles are bound to their group, the unbinding procedure uses this as the center of the group when including surrounding dark matter. It is also output as the center of each group by Denmax. To find these centers, we push all of the tracer particles which belong to a group for five more iterations. However, this time we push them only a distance of $\epsilon/10$ along the density gradient. This has the effect of localizing them in a ball of roughly $\epsilon/5$. We use the mean position of the tracer particles in each group its center. This refinement has little impact for many applications and should be considered optional.

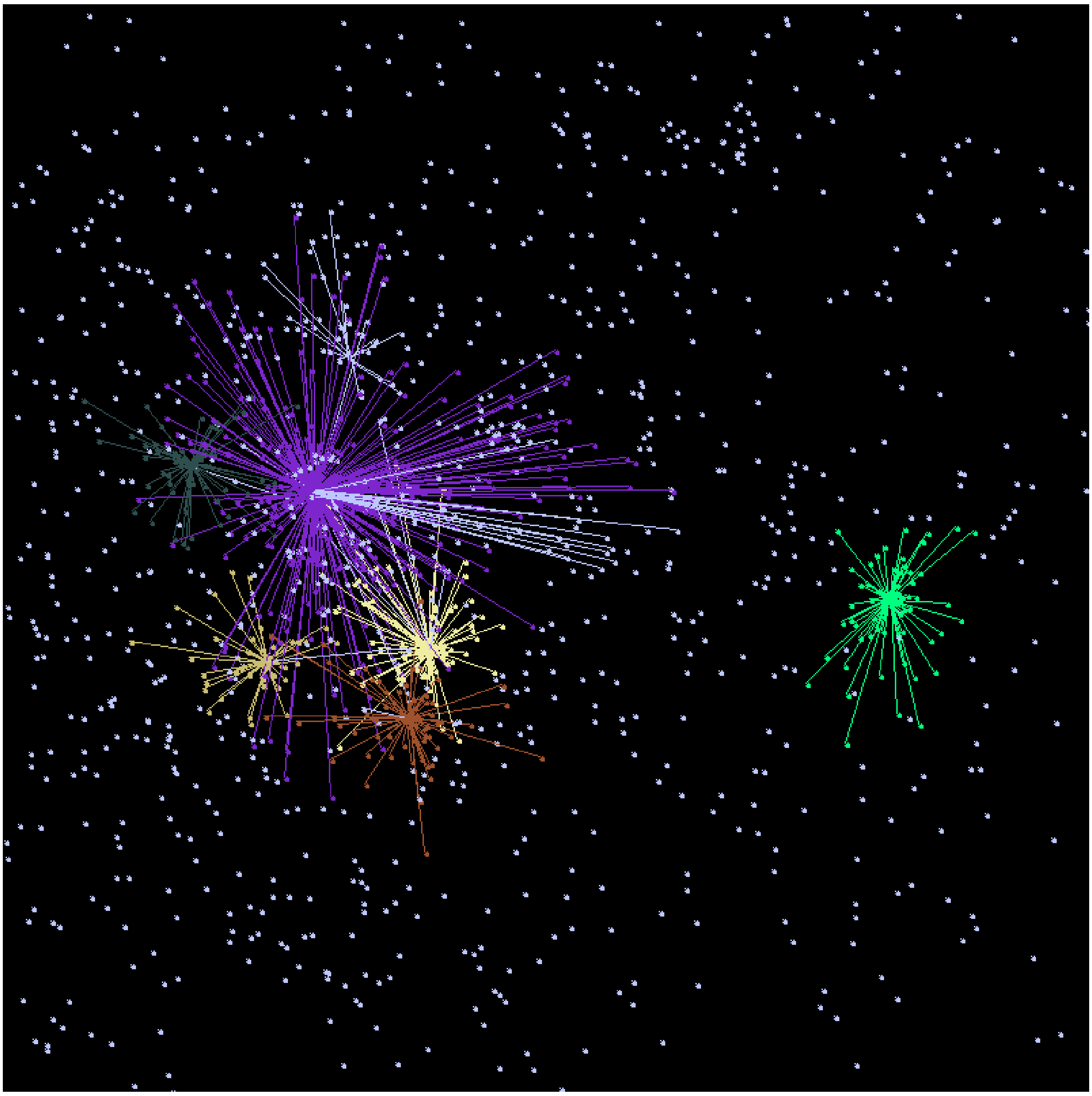

Figure 4.1: SKID groups found in a small sample of particles. The lines connect the initial and final positions of the tracer particles, showing the positions of the density maxima. The actual paths of the tracer particles during the iterations are not straight lines.

*4.2.6 Unbinding*

The final stage of SKID is to remove particles which are not gravitationally bound to their group. Only the initial configuration of the particles and their group assignments are relevant to this stage. The tracer particles, which were used in making the group assignments, are no longer needed. Calculating the binding energy of a particle with respect to its group tells us whether it is bound or not, a negative binding energy indicating a bound particle. The binding energy of a particle is given by the sum of its kinetic and potential energies, where the kinetic energy is calculated with respect to the center of mass velocity of the group,

$$E_b = \frac{1}{2} m |\mathbf{v} - \mathbf{v}_{cm}|^2 + U. \tag{4.5}$$

Here $m$ is the mass of the particle, $\mathbf{v}$ its velocity and $U$ its potential energy. Since cosmological simulations typically use comoving coordinates, we need to add corrections to this equation to obtain the true binding energy.

$$E_b = \frac{1}{2} m \left[ \frac{|\mathbf{v} - \mathbf{v}_{cm}|}{1+z} + H_0 \sqrt{1 + \Omega_0 z}\, |\mathbf{r} - \mathbf{r}_{cm}| \right]^2 + (1+z)\, U. \tag{4.6}$$

For a given group we calculate the binding energy of all its member particles and begin by removing the one with the largest non-negative binding energy, if it exists. Once this particle has been removed all the binding energies of the remaining particles must be recalculated since the center–of–mass and center–of–mass velocity will have changed. we then continue removing particle in the same way until all the particles in the group have negative binding energies. This process may well dissolve a group of particles completely, but in general this only occurs for groups of very few initial particles.

When we are unbinding gas particles we need to include the gravitational effect of dark matter surrounding the gas group. This is what is meant by Type II unbinding in the preceding table, Type I being the basic method described above. The size of the surrounding volume of dark matter that should be considered cannot in general be precisely defined. However, we have determined that using a radius twice that which encloses the gas group preserves gas groups that clearly remain bound in the simulations while still removing "inter-loping" gas particles. Other than including this extra gravitational effect, the unbinding proceeds as discussed above.

One problem arises when SKID must unbind groups of with a very large number of particle. In these cases the continual $\mathcal{O}(N^2)$ gravity calculations become very expensive. One example of when this happens is when SKID is unbinding the group which corresponds to the center of a cluster halo, which has an order of magnitude more particles than any of the substructure halos orbiting within it. For such cases the unbinding procedure does not need to iterate, a single removal of all unbound particles is sufficient. A treecode could be used for calculating the potential energy once for all particles in this case, reducing the computational complexity to $\mathcal{O}(N \log N)$ for groups with very large numbers of particles.

### *4.3 Summary*

SKID is a group finder which finds the *smallest* scale of substructure within the simulation, and only that. For example it will not produce a group with the mass of an entire cluster, when such a cluster resolves substructure within it. However, it is very robust at finding the smallest bound structures and has almost no input parameters. If we define $\epsilon_m ax$ as the value of $\epsilon$ at which the maximum number of groups are found, then for values of $\epsilon$ up to several times this value SKID produces group catalogs that agree with each other quite well. Setting this single parameter to the gravitational softening used in the simulation and up to about twice this value is generally the most appropriate choice. In figure 4.2 we compare SKID to the friends–of–friends method of finding groups. The simulation used was a 1.3 million particle cluster halo simulation, similar in mass to the Virgo cluster of galaxies. This simulation has very good resolution and easily resolves individual galaxy halos which are falling into or orbiting within the cluster halo. We see that FOF finds the main cluster halo and halos outside of this, but even these outer halos are biased with respect to the background density. While SKID ignores the main cluster halo, it successfully finds all the halos orbiting within the cluster. Furthermore it ensures that all the particles in a halo are gravitationally associated with each other. This makes SKID ideal for studying the dynamics and properties of galaxy halos in different environments.

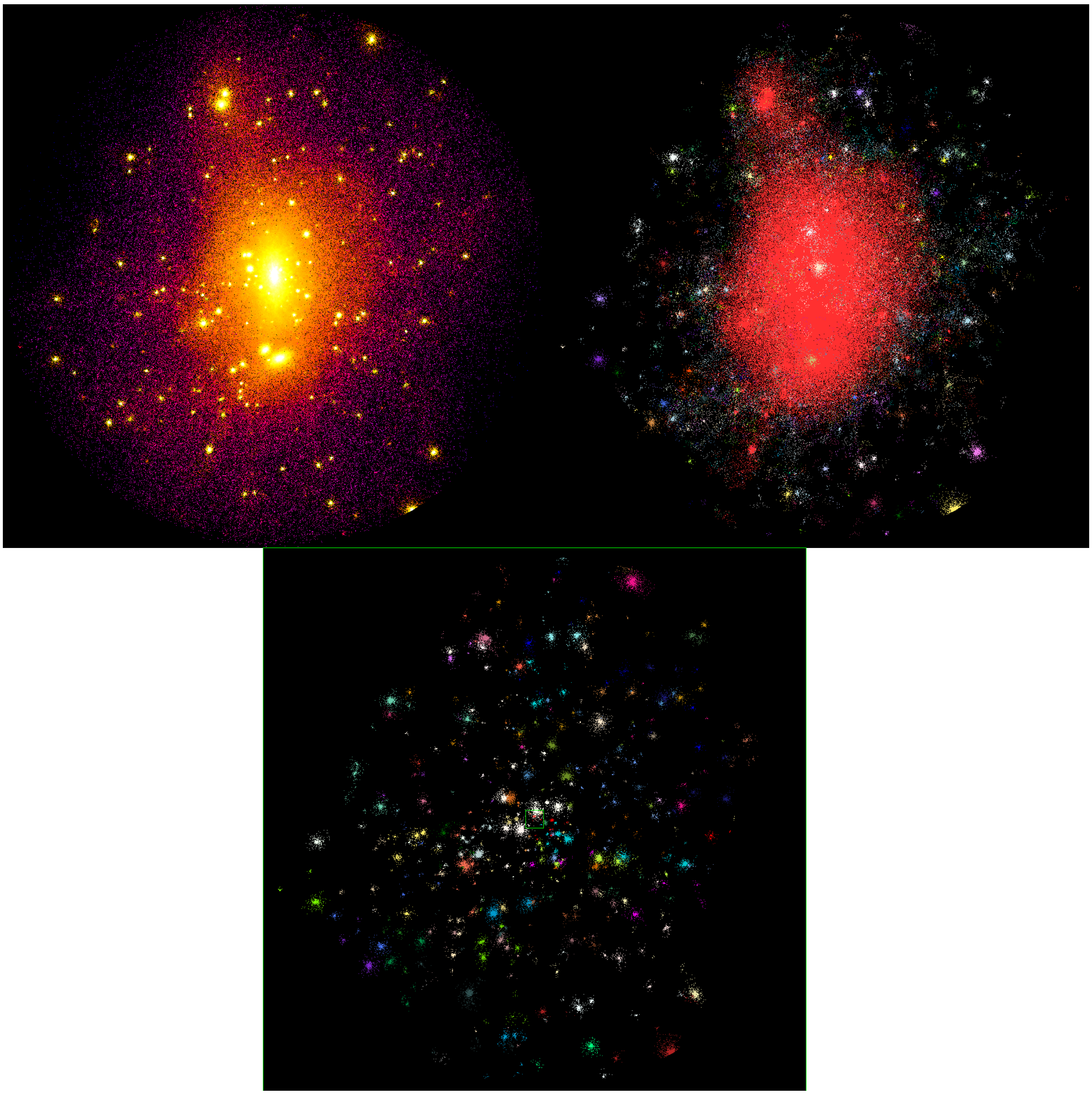

Figure 4.2: Comparison of SKID groups to those produced by friends–of–friends. The top left panel shows all particles colored by the log of their local density, with blue corresponding to an over density of 1 and white to $10^5$. The top right panel shows the FOF halos found, where each halo is given a random color. The main halo is the large red structure which extends out to roughly the virial radius of the cluster. The bottom panel shows the SKID halos in the same way. A small square green box indicates the center of the cluster in the bottom panel.

# Chapter 5

# SUMMARY OF LARGE SCALE STRUCTURE RESULTS USING PKDGRAV AND SKID

High resolution simulations of dark matter on massively parallel computers using PKDGRAV enables us for the first time to realistically model the full wealth of structures present in the Universe from the largest voids to the smallest satellite galaxies. By resolving the dark matter halos of galaxies within a larger cosmological context, in particular resolving them as substructure of larger cluster halos, we have reached a critical threshold with cosmological N-body simulations. We can confront our models with the the observations of galaxies, thereby placing substantial constraints on the theory of galaxy formation and on the global cosmological parameters.

This chapter outlines the scientific program that the author has been involved in, and which has been enabled by PKDGRAV and SKID.

## 5.1 Structure and Dynamics of Dark Matter Halos

Resolving substructure within halos is a computational challenge that has required the use of new algorithms and parallel computing. On the order of a million particles within the virial radius of a cluster are required to resolve a significant number of substructure halos. Since such halos would harbor individual galaxies, we can compare their distribution and orbital properties to galaxies in observed clusters. We have made several predictions on the properties of halos within a cluster similar in size to the Virgo cluster of galaxies.

Within the observable universe there are two scales on which substructure is most evident. The first of these, as mentioned, are the galaxies within galaxy clusters. However, the dwarf galaxies of the Milky Way and the Andromeda galaxy are also conspicuous examples of substructure on a somewhat smaller scale. The latter of these is quite interesting because our cold dark matter models predict that there should be many more satellite galaxies in the Milky Way than are observed. Our very high resolution simulations of a system very similar to our own Local Group were the first to point out this interesting discrepancy. One possibility is that these dark matter halos are actually present, and for some reason failed to produce a significant number of stars. Another possibility being considered is that these halos are again present and observed as the High Velocity Clouds (HVCs), in which case the HVCs must be placed at significantly larger distances and be more massive than has previously been thought. The final possibility is that these halos are simply not there, and that we have a fundamental challenge to the cold dark matter model.

Finally, an even greater computational challenge is resolving the very central structure of dark matter halos. Our high resolution simulations pointed out that the central density profile of dark matter halos is very steep having a power-law index of $-1.2$ to $-1.5$. The resulting rotation curves appear to be in conflict with the observed rotation curves of low surface brightness galaxies. This discrepancy has also seen a lot of attention, on the observational front as well as the theoretical. On the scale of clusters, the central profile could alter the gravitational lensing arcs observed in distant clusters of galaxies, particularly an observation of *radial* arcs.

We have found that the core structure is independent of the merger history and seems to be a universal outcome of the dark matter collapse process. This is very significant since it seems to provide a constraint on the fundamental nature of the dark matter. A way to significantly modify the central profile of dark matter halos is if one considers dark matter that is not cold, but has some amount pressure support. This introduces a scale which would break the universality of the central profile shape. It is therefore important to observationally investigate the central density profile of both clusters and galaxies (and other scales where it becomes possible).

The following three paragraphs are the abstracts to papers addressing these issues on the structure and dynamics of dark matter halos.

### Dark Matter Halos within Clusters

Ghigna, Moore, Governato, Lake, Quinn and Stadel (1998)

*We examine the properties of dark matter halos within a rich galaxy cluster using a high resolution simulation that captures the cosmological context of a cold dark matter universe. The mass and force resolution permit the resolution of 150 halos with circular velocities larger than 80 km $s^{-1}$within the cluster's virial radius of 2 Mpc. This enables an unprecedented study of the statistical properties of a large sample of dark matter halos evolving in a dense environment. The cumulative fraction of mass attached to these halos varies from 0% at 200 kpc, to 13% at the virial radius. Even at this resolution the overmerging problem persists; halos that pass within 200 kpc of the cluster center are tidally disrupted. Additional substructure is lost at earlier epochs within the massive progenitor halos. The median ratio of apocentric to pericentric radii is 6:1; the orbital distribution is close to isotropic, circular orbits are rare, radial orbits are common. The orbits of halos are unbiased with respect to both position within the cluster and with the orbits of the smooth dark matter background and no velocity bias is detected. The tidal radii of surviving halos are generally well-fit using the simple analytic prediction applied to their orbital pericenters. Halos within clusters have higher concentrations than those in the field. Within the cluster, halo density profiles can be modified by tidal forces and individual encounters with other halos that cause significant mass loss - "galaxy harassment". Mergers between halos do not occur inside the clusters virial radius.*

### Resolving the Structure of Cold Dark Matter Halos

Moore, Governato, Quinn, Stadel and Lake (1998)

*We examine the effects of mass resolution and force softening on the density profiles of cold dark matter halos that form within cosmological N-body simulations. As we increase the mass and force resolution, we resolve progenitor halos that collapse at higher redshifts and have very high densities. At our highest resolution we have nearly 3 million particles within the virial radius, several orders* *of magnitude more than previously used and we can resolve more than one thousand surviving dark matter halos within this single virialized system. The halo profiles become steeper in the central regions and we may not have achieved convergence to a unique slope within the inner 10% of the virialized region. Results from two very high resolution halo simulations yield steep inner density profiles, $\rho(r) \sim r^{-1.4}$. The abundance and properties of arcs formed within this potential will be different from calculations based on lower resolution simulations. The kinematics of disks within such a steep potential may prove problematic for the CDM model when compared with the observed properties of halos on galactic scales.*

**Cold Collapse and the Core Catastrophe**

Moore, Quinn, Governato, Stadel and Lake (1999)

*We show that a universe dominated by cold dark matter fails to reproduce the rotation curves of dark matter dominated galaxies, one of the key problems that it was designed to resolve. We perform numerical simulations of the formation of dark matter halos, each containing $\gtrsim 10^6$ particles and resolved to 0.003 times the virial radius, allowing an accurate comparison with rotation curve data. A good fit to both galactic and cluster sized halos can be achieved using the density profile $\rho(r) \propto [(r/r_s)^{1.5}(1 + (r/r_s)^{1.5})]^{-1}$, where $r_s$ is a scale radius. This profile has a steeper asymptotic slope, $\rho(r) \propto r^{-1.5}$, and a sharper turnover than found by lower resolution studies. The central structure of relaxed halos that form within a hierarchical universe has a remarkably small scatter (unrelaxed halos would not host disks). We compare the results with a sample of dark matter dominated, low surface brightness (LSB) galaxies with circular velocities in the range 100–300 km $s^{-1}$. The rotation curves of disks within cold dark matter halos rise too steeply to match these data which require a constant mass density in the central regions. The same conclusion is reached if we compare the scale free shape of observed rotation curves with the simulation data. It is important to confirm these results using stellar rather than HI rotation curves for LSB galaxies. We* *test the effects of introducing a cut-off in the power spectrum that may occur in a universe dominated by warm dark matter. In this case halos form by a monolithic collapse but the final density profile hardly changes, demonstrating that the merger history does not play a role in determining the halo structure.*

## 5.2 Implications of Cold Dark Matter Structure for Galaxies

With our simulations we can directly test the implications of the cold dark matter structure on galaxy properties and dynamics. Using SKID we can identify all the dark matter halos within the simulation and examine their masses, orbital properties, the way the matter is distributed within them, their environments and how their mass was assembled over time through merging. These factors allow us to compare such dark matter halos with observed galaxies, even though we do not follow the gas physics of the baryons in such simulations. Future cosmological simulations which include gas physics and star formation will should be able to form galaxies *ab initio* with properties that can be directly compared to observed galaxies.

At the simplest level, we can identify the circular velocity of halos and their orbital properties and attempt to match these with known galaxy systems. An obvious system to study is our own Local Group of galaxies, the Milky Way and Andromeda system along with its satellite galaxies such as the Large and Small Magellanic Clouds. We have done this by selecting artificial local group systems from a large volume, high resolution simulation by matching the circular velocities, separation and distance from the nearest cluster of galaxies to our own Milky Way and Andromeda system. One finds several candidate systems in a large enough volume, but none of these systems have the "cold" local environment we observe in the Local Group. That the Universe could be dominated by vacuum energy, $\Omega_\Lambda \approx 0.7$, along with the fact that our Local Group of galaxies probably lives in a somewhat special environment (between two large voids) may help to explain this result.

At a more sophisticated level, we have used a semi-analytical model to predict the luminosity and type of galaxy that is likely to be harbored within a particular dark matter halo, based on the halo's mass and particularly its merger history. This allows us to identify within the simulation which halos are likely to correspond with a particular type of galaxy and thereby study the clustering properties of that type within the model. We have done this to show that the high degree of clustering of primeval galaxies (the Lyman Break galaxies), which at first seems to contradict theory, can be explained by the fact that these early galaxies are in fact the progenitors of present day clusters. The cosmological N-body simulations, along with semi-analytical modeling allows us to make this important connection. In a similar manner we have studied the progenitors of early-type galaxies and the merger history of these galaxies. We find that the hierarchical clustering picture within a cold dark matter Universe can explain the the populations of early-type galaxies that are observed. From the simulations we are able to make predictions on the number density of interacting bright binary galaxies as a function of redshift, which differs depending on the cosmological model used. Such predictions provide a way to test our understanding of galaxy formation within the hierarchical clustering picture of structure formation in a cold dark matter dominated universe.

Finally, we can replace halos within the simulation with high resolution equilibrium models containing the three principal components of dark matter halo, disk, and bulge. Although these galaxies have not formed *ab initio* within the halo, they are a fully dynamical model which can test the effect of environment on the structure of the galaxy within a halo. Our N-body simulations allow us to directly follow the evolution of the stellar component of both a low surface brightness galaxy (LSB) and a high surface brightness galaxy (HSB), as the halo containing them participates in the formation of a large galaxy cluster. We find that the LSB galaxy makes a morphological transformation to a galaxy which resembles a dwarf spheroidal galaxy while the HSB galaxy remains stable during cluster formation. This helps us to understand the distribution of galaxies we observe within clusters as compared to the field.

The following four paragraphs are abstracts to papers addressing the implications of cold dark matter structure for galaxies as outlined above.

**The Local Group as a Test of Cosmological Models**

Governato, Moore, Cen, Stadel, Lake and Quinn (1997)

*The dynamics of Local Group and its environment provide a unique challenge to cosmological models. The velocity field within 5*$h^{-1}$ *Mpc of the Local Group (LG) is extremely "cold". The deviation from a pure Hubble flow, characterized by the observed radial peculiar velocity dispersion, is measured to be* $\sim$ *60km* $s^{-1}$*. We compare the local velocity field with similarly defined regions extracted from N-body simulations of Universes dominated by cold dark matter (CDM). This test is able to strongly discriminate between models that have different mean mass densities. We find that neither the* $\Omega = 1$ *(SCDM) nor* $\Omega = 0.3$ *(OCDM) cold dark matter models can produce a single candidate Local Group that is embedded in a region with such small peculiar velocities. For these models, we measure velocity dispersions between 300–700km* $s^{-1}$ *and 150–300km* $s^{-1}$ *respectively, more than twice the observed value. Although both CDM models fail to produce environments similar to those of our Local Group on a scale of a few Mpc, they can give rise to many binary systems that have similar orbital properties as the Milky Way–Andromeda system. The local, gravitationally induced bias of halos in the CDM "Local Group" environment, if defined within a sphere of 10 Mpc around each Local Group is* $\sim$ *1.5, independent of* $\Omega$*. No biasing scheme could reconcile the measured velocity dispersions around Local Groups with the observed one. Identification of binary systems using a halo finder (named Skid) based on a local density maxima search instead of a simple linking algorithm, gives a much more complete sample. We show that a standard "friend of friends" algorithm would miss about 40% of the LG candidates present in the simulations.*

**The Seeds of Rich Galaxy Clusters in the Universe:**

Governato, Baugh, Frenk, Cole, Lacey, Quinn and Stadel (1998)

*The discovery (Steidel et al. 96) of a population of young galaxies at an epoch when the universe was about one tenth of its current age has shed new light on the question of when and how galaxies formed. Within the context of popular models*

*(Baugh et al. 98) this is the population of primeval galaxies that built themselves up to the size of present–day galaxies through the process of repeated mergers. But the recent detection (Steidel et al. 98) of a large concentration of these primeval galaxies appears to be incompatible with hierarchical clustering models, which generally predict that clusters of this size are fully formed later in time. Here we use a combination of two powerful theoretical techniques –semi-analytic modeling and N-body simulations– to show that such large concentrations should be quite common in a universe dominated by cold dark matter, and that they are the progenitors of the rich galaxy clusters seen today. We predict the clustering properties of primeval galaxies which should, when compared with data that will be collected in the near future, test our understanding of galaxy formation within the framework of a universe dominated by cold dark matter.*

**On the Origin of Early Type Galaxies and the Evolution of the Interaction Rate in the Field**

Governato, Gardner, Stadel, Quinn and Lake (1999)

*Using cosmological N-body simulations of critical (SCDM) and open ($\Omega$=0.3, OCDM) cold dark matter models we have identified dark matter halos which are associated with the progenitors of present day bright early–type galaxies. By following their merging history, we show how early–type galaxies that formed within massive halos at redshift* $\simeq 3$ *are now preferentially residing in clusters and groups. On the other hand, those that formed through major merging events between redshift 1 and the present have not yet been accreted into larger, virialized structures. This result is in agreement with analytical predictions in hierarchical clustering models. CDM models are able to explain both the ancient and uniform population of ellipticals that dominates in clusters together with the more recent and heterogeneous population of field ellipticals. Predictions for the co-moving number density of bright early–type galaxies are given, and are shown to be consistent with the observed luminosity function. We predict that the number density of interacting bright binary galaxies, from which the field population of* *ellipticals may have originated, is proportional to (1+z)*$^{4.2\pm0.28}$ *and (1+z)*$^{2.5\pm0.42}$ *in SCDM and OCDM respectively. This result is consistent with previous analytical estimates and is discussed together with recent observational constraints.*

**On the Survival and Destruction of Spiral Galaxies in Clusters**

Moore, Lake, Quinn and Stadel (1999)

*We follow the evolution of disk galaxies within a cluster that forms hierarchically in a cold dark matter N-body simulation. At a redshift* $z = 0.5$ *we select several dark matter halos that have quiet merger histories and are about to enter the newly forming cluster environment. The halos are replaced with equilibrium high resolution model spirals that are constructed to represent examples of low surface brightness (LSB) and high surface brightness (HSB) galaxies. Varying the disk and halo structural parameters reveals that the response of a spiral galaxy to tidal encounters depends primarily on the potential depth of the mass distribution and the disk scale length. LSB galaxies, characterized by slowly rising rotation curves and large scale lengths, evolve dramatically under the influence of rapid encounters with substructure and strong tidal shocks from the global cluster potential – galaxy harassment. We find that up to 90% of their stars are tidally stripped and congregate in large diffuse tails that trace the orbital path of the galaxy and form the diffuse intra-cluster light. The bound stellar remnants closely resemble the dwarf spheroidals (dE's) that populate nearby clusters. HSB galaxies are stable to the chaos of cluster formation and tidal encounters. These disks lie well within the tidally limited dark matter halos and their potentials are more concentrated. Although very few stars are stripped, the scale height of the disks increases substantially and no spiral features remain, therefore we speculate that these galaxies would be identified as S0 galaxies in present day clusters.*

### *5.3 Statistical Properties of Large Scale Structure*

There are many statistical measures which can be applied towards the Large Scale Structure in a Universe dominated by Cold Dark Matter. We have focused on two very different measures, but both which apply to the largest non-linear structures. The first of these is the weak gravitational lensing effect that large scale structure has on standard candles, such as Type-Ia supernovae, observed at high redshift. The intervening structure will tend to magnify and demagnify (with respect to a perfectly homogeneous cosmology) the light observed depending on the distribution of structure pierced by the particular line of sight through the universe. To study this we have simulated the lensing effects for many tens of thousands of lines of sight through each of three fiducial cosmological models. The details of this statistical study are the topic of Chapter 6.

Finally we have investigated the high order statistics of $N$-point correlation functions within a very large cosmological simulation. A thorough understanding of such statistics is needed to properly assess the results from large redshift surveys, such as SLOAN, on the clustering statistics of galaxies. In particular, this work allows one to place constraints on cosmological parameters from such measurements. We show that the clustering statistics in the EDSGC survey strongly supports a cosmological model with $\Omega < 1$ with no biasing. The following abstract is taken from the paper on our investigation of the $N$-point correlation functions of large cosmological simulations.

**N–Point Correlations in CDM and OCDM Simulations**

Szapudi, Quinn, Stadel and Lake (1999)

*Higher order statistics are investigated in* $(\Omega)$*CDM universes by analyzing* 500 Mpc *high resolution tree* $N$*-body simulations with both* $\Omega = 1$, *and* $\Omega < 1$. *The amplitudes of the* $N$*-point correlation functions are calculated from moments of counts-in-cells determined by a pair of new algorithms especially developed for large simulations. This approach enables massive oversampling with* $\simeq 10^{9-14}$ *cells for accurate determination of factorial moments from up to* 47 *million particles in the scale range of* 8 kpc $-$ 125 Mpc. *Thorough investigation shows that there are three scale ranges in the simulations:* $\geq 8$ Mpc, *weakly non-linear* *regime, where perturbation theory applies with utmost precision,* 1 Mpc − 8 Mpc*, the non-linear plateau, and finally* $\leq 1$ Mpc*, a regime where dynamical discreteness effects dominate the higher order statistics. In the physically relevant range of* 1 Mpc − 125 Mpc *the results i) confirm the validity of perturbation theory in the weakly non-linear regime, ii) establish the existence of a plateau in the highly non-linear regime similar to the one observed in scale free simulations iii) show extended perturbation theory to be an excellent approximation for the non-linear regime iv) find the time dependence of the* $S_N$*'s to be negligible in both regimes v) in comparison with similar measurements in the EDSGC survey, strongly support* $\Omega < 1$ *with no biasing vi) show that the formulae of Szapudi & Colombi (1996) provide a good approximation for errors on higher order statistics measured in* $N$*-body simulations.*

Chapter 6

# TOWARDS BREAKING DEGENERACIES IN THE INTERPRETATION OF TYPE-IA SUPERNOVA DATA USING THE WEAK LENSING INDUCED SIGNATURE OF LARGE SCALE STRUCTURE

## *6.1 Introduction*

The effects of matter inhomogeneities on the propagation of light in the Universe has been intensely studied over the past decade (see Schneider, Ehlers & Falco 1992 and Mellier 1998 for an excellent review of the subject and of recent developments). Among the more dramatic examples of gravitational lensing are the splitting of a high-redshift source into multiple images and creation of highly distorted giant luminous arcs. Such strong gravitational lensing occurs when light rays from a source pass through the densest regions of the Universe, such as cores of galaxies or clusters of galaxies. These regions are rare, and strong lensing typically affects $\sim 1$ in $10^3$ random lines of sight. Most of the matter in the Universe is distributed in more ubiquitous, less dense, structures whose form has often been described as a cosmic web, a sponge-like cellular network of filaments and sheets (Bond *et al.* 1996; Yess & Shandarin 1996). The gravitational influence of the cosmic web will also lens the light beams from distant sources, albeit weakly, amplifying or deamplifying the observed flux as well as inducing asymmetric stretching in the images of the sources. And because of the ubiquitous nature of the density inhomogeneities, light beams along all lines of sight will be affected.

The weak gravitational lensing effects of smoothly distributed large-scale density inhomogeneities were first considered by Gunn (1967). The first calculations using matter distribution predicted by the currently favored class of models for the formation and evolution of structure in the Universe, the CDM family of models, were carried out by Jaroszyński, Park, Paczyński & Gott (1990), Babul & Lee (1991), Blandford *et al.* (1991), Miralda-Escude (1991), Kaiser (1992), and Frieman (1997). These theoretical studies, as well as those that have followed, indicate that weak lensing effects, though not detectable in any individual image, ought to be measurable by studying the observed properties of a large number of cosmologically distant sources.

Measurement of the large-scale weak gravitational lensing signal would constitute a direct probe of the matter distribution in the Universe independent of how the light and the baryonic matter in the Universe are distributed. Consequently, a significant effort has been, and continues to be, devoted to searching for the weak lensing signal in images of cosmologically distributed optical (Valdes, Tyson & Jarvis 1983; Mould *et al.* 1994; Schneider *et al.* 1998; Whittman *et al.* 2000; van Waerbeke *et al.* 2000) as well as radio (see, for example, Kamionkowski, Babul, Cress & Refregier 1998) sources.

Most of these searches have concentrated on detecting small but coherent shearing of the images of high redshift sources. However, as discussed in a series of publications (Wagoner 1986; Schneider & Wagoner 1987; Linder, Schneider & Wagoner 1988; Babul & Lee 1991; Wambsganss *et al.* 1997; Metcalf 1998), if there exists a class of distant sources that are effectively standard candles, then weak gravitational lensing can also be detected through its amplifying/deamplifying effects on the observed flux from these sources. The resulting signal can, in principle, be used to test models for the distribution and evolution of inhomogeneities in the Universe. Current models for structure formation suggest that the most non-linear of the structures are those on megaparsec scales and smaller, and lensing is very sensitive to the presence of these strongly non-linear, small-scale structures (Babul & Lee 1991; Jain & Seljak 1997). It is, therefore, most natural to use numerical simulations to model the distribution of matter in the Universe as a function of cosmic time and propagate light beams through these simulations.

The first study to use numerical simulations to investigate the effects of large-scale weak gravitational lensing in a CDM universe was that of Jaroszyński, Park, Paczyński & Gott (1990). However, the resolution of their simulations was relatively poor. Subsequent improvements in computational techniques and dramatic increase in available computing power, has resulted in lensing studies being carried out using simulations with ever-increasing dynamic range (Bartelmann & Schneider 1991; Jaroszyński 1991; Wambsganss *et al.* 1995; 1997; 1998; Premadi, Martel & Matzner 1998; Tomita 1998; Couchman, Barber & Thomas 1998; Jain, Seljak & White 1999; Hamana, Martel & Futamase 1999; White & Hu 1999). Most of these simulation studies are still limited either by insufficient resolution or by the size of the volume simulated. A thorough study requires numerical simulations which resolve scales of order $50h^{-1}$ kpc, the scale of the cluster cores. At the same time, the simulation volume must also be large enough, on the order of $250h^{-1}$ Mpc across, to encompass a representative volume of the Universe. Smaller volumes carry the risk that they may become dominated by a few structures such as a pair of massive clusters, a large void, etc. and the corresponding weak lensing results thus biased. Simulations with even this minimum required dynamic range are still computationally quite expensive. As a workaround, a few of the authors have used innovative techniques such as semi-analytic schemes (Jaroszyński 1991; Premadi, Martel & Matzner 1998), convolution of large but low resolution simulations with small but high resolution simulations (Wambsganss, Cen & Ostriker 1998) and "tiling" (White & Hu 1999) to extend the dynamic range of the simulations. With recent development in simulation software and improvement in computing hardware for floating point calculations, it is now possible to directly simulate the requisite large, high resolution, cosmological volumes, mitigating the need to introduce such approximations.

For many years now, studies of Type Ia supernovae have suggested that they have a very narrow range in peak luminosities and hence, are excellent candidates for standard candles (Sandage & Tammann 1982; Cadonau, Sandage & Tammann 1984; Miller & Branch 1990). Based on correlations between the peak flux and the width of the supernova's light curve, the absolute magnitude of these supernovae is thought to be known to within 0.15 mag (Hamuy et al. 1996; Riess, Press & Krishner 1996). This, coupled with detections of Type Ia supernovae at high redshifts, has allowed the luminosity distance–redshift relationship to be used to constrain the fundamental cosmological parameters (Perlmutter *et al.* 1997; 1998; Garnavich *et al.* 1998; Riess *et al.* 1998; Schmidt *et al.* 1998). This constraint is best expressed as, $0.8\Omega_M - 0.6\Omega_\Lambda = -0.2 \pm 0.1$, which emphasizes the $\Omega$–$\Lambda$ degeneracy which exists for these studies.

It is well known that the small spread in the determined peak absolute magnitudes of Type Ia supernovae limits the precision with which the cosmological parameters can be determined. In addition several studies have pointed out that gravitational lensing will cause an additional spread in the observed flux (Kantowski, Vaughan & Branch 1995; Friemann 1997; Wambsganss, Cen, Xu & Ostriker 1997; Holz & Wald 1998; Kantowski 1998; Holz 1998). However, there is a general consensus that, for $z < 1$ supernovae, neither is sufficiently large to alter the current constraint on cosmological parameters. At higher redshift the spread in the brightness distribution due to the intervening structure becomes comparable to the intrinsic spread of Type-Ia supernovae. For a large sample of high redshift supernovae, this opens up the possibility of using the measured brightness distribution itself as an independent test of cosmological models. This is the focus of the present paper.

Specifically, we show that the characteristics of the lensing induced magnification distribution is quite insensitive to $\Lambda$, which offers ways to break the $\Omega$–$\Lambda$ degeneracy noted previously. Alternatively, if $\Omega$ and $\Lambda$ are known, for example through microwave background experiments, then the weak lensing induced brightness distribution will distinguish itself from intergalactic forms of gray absorption. Greatly increasing the supernova sample and extending it to higher redshifts can be achieved through surveys such as SNAP (http://snap.lbl.gov/), which are currently being proposed. Such large samples of Type Ia supernovae, along with any possible decrease in the intrinsic spread of their determined peak absolute magnitudes, will make such investigations possible.

This paper is organized as follows: In Section 6.2 we discuss our method of determining the weak lensing effects of high redshift sources using large volume (1 Gpc$^3$), high resolution ($4.7 \times 10^7$ particles) N-body simulations for three standard cosmological models. Readers interested primarily in the Magnification Distribution Test may wish to skip ahead to Section 6.4. In Section 6.3 we discuss tests that we carried out in order to convince ourselves that our results are robust with respect to numerical resolution and smoothing. In Section 6.4 we present the resulting convergence and magnification distributions for each of three cosmological models. Finally, in Section 6.5 we discuss the implications of our study and the application of the Magnification Distribution Test to Type-Ia supernovae.

## 6.2 Numerical Method

### 6.2.1 The Simulations

We have simulated structure formation within a periodic cube of comoving length $L = 500h^{-1}$ Mpc, realized with $360^3$ ($\approx 4.66 \times 10^7$) particles, for three "fiducial" CDM models: a critical density model ($\Omega = 1, h = 0.5$), an open model ($\Omega_0 = 0.3, h = 0.75$) and a flat lambda model ($\Omega_0 = 0.3, h = 0.7$). Although each of these models used a normalization given by $\sigma_8 = 1$, models with different normalization are obtained from earlier outputs ($z > 0, \sigma_8 < 1$) by appropriate scaling of the cosmological parameters. This allows us to also examine the weak lensing properties of less extremely clustered cases than those given by the fiducial models.

The initial conditions were set using the Bardeen *et al.* (1986) transfer function for CDM, and realized at $z = 49$ using the Zel'dovich approximation on a uniform grid of $360^3$ particles. By using the same set of random phases for the fluctuations of all three initial conditions, we ensured that similar structures form in the same general locations within each of the simulations. The simulations were computed using *PKDGRAV*, a parallel treecode that allows for periodic boundary conditions and individual timesteps per particle (Stadel & Quinn, 2001). This code uses a $4^{th}$ order multipole expansion for cells of the tree which gives very good force accuracy when used with a cell opening angle of $\Theta = 0.8$. For these simulations we used $\Theta = 0.45$ for $z > 2$ to address the need for higher relative force accuracy at higher redshift in order to correctly evolve the fluctuations on the nearly homogeneous background. A cubic spline softening for particles of $\epsilon = 50$, 62.5 and 50 $h^{-1}$ kpc was used for the critical, open and lambda models respectively. With this softening, the overall structure of clusters in the simulation is well resolved. The critical model was run using a fixed number of 800 timesteps per particle to reach redshift zero. The open and lambda models used individual timesteps per particle, roughly equivalent to having used 1600 fixed timesteps but at a fraction the cost. Timesteps were constrained to $\Delta t < 0.3\sqrt{\epsilon/a}$ for the open model and $\Delta t < 0.2\sqrt{\epsilon/a}$ for the lambda model, where $\epsilon$ is the softening length and $a$ is the magnitude of the acceleration on a given particle (Quinn *et al.* 1997). The critical model was run on 128-256 nodes of a Cray-T3D, taking an estimated total of 150,000 node hours in 1996. The open model was run on 64-128 nodes of a Cray T3E-600 in 1998 and the lambda model was run on 50 nodes of a Cray T3E-900 in 1999 using only 17,500 node hours, reflecting the improvements in hardware and in the *PKDGRAV* code, particularly the use of individual timesteps.

The extremely large volumes simulated, coupled with a reasonable mass resolution and very good force resolution made possible by the use of a treecode, allow us to study in detail the evolution of structures ranging in size from groups of galaxies, made up of several tens of particles each, to very rich clusters that contain a few thousand particles (see, for example, Szapudi *et al.* 1998 and Governato *et al.* 1999). Here we will focus on the gravitational impact of these structures on the propagation of light rays though the universe.

### *6.2.2 Skewering the Universe: Generating Beams*

In considering the effects of the cosmic structure between the observer and a high redshift source, we must take into account the significant evolution of this structure in look-back time. Our simulations capture the largest scale of structure and extend to a redshift of about 0.1 from any present day observer without replicating the periodic volume of the simulation. However, the outputs of these simulations are all at a fixed proper time, and simply replicating them does not capture the look-back history of the evolution of structure. One way to address this issue is to output the entire lightcone for one or more observers. To do this one must output shells of decreasing radius working in towards the center ($z = 0$) as the simulation progresses (see Yoshida *et al.* 2001). For our $\mathrm{Gpc}^3$ "Sloan–volume" simulations, a single lightcone output extending to $z = 1$ would be an enormous file containing roughly 8.8 billion particles.

An alternate approach, which is suitable to high redshift studies, is to generate pencil beam surveys from the fixed proper time outputs of the simulation. Since we have 15 simulation outputs from $z = 1$ to $z = 0$, we can account for the look-back history of structure formation within these beams by jumping to successively earlier and earlier proper time outputs of the simulation as we extend the beam to higher redshift. Furthermore, this approach allows for flexibility in choosing the observer's position (e.g. we can explore the consequences of constraining the observer to reside in specific types of environment), choosing the width and depth of the beam and choosing the angular coverage of the beams (to account for correlations between nearby beams). In this way one can tailor the generated beams to suit a particular high redshift study, thereby optimizing the use of available data and computational resources.

We have written a general purpose tool called BEAM [1], which extracts a column of particles from simulation outputs in the manner described, such that the look-back history of structure formation is accounted for. BEAM was designed to handle the analysis of supercomputer scale simulations on workstation scale computing. This is accomplished by never requiring all of the particles of the simulation to be stored in memory simultaneously. For each output file of the simulation we determine the section along the beam for which the output contains the appropriate epoch of structure. For output file $i = (0..n-1)$ at proper time $t_i$, this is given by the range in comoving distance $[r_i^c, r_{i+1}^c]$, where

$$r_i^c = c\int_{\frac{1}{2}(t_i+t_{i-1})}^{t_0} \frac{dt}{a(t)} \qquad r_n^c = c\int_{t_n}^{t_0} \frac{dt}{a(t)} \qquad t_i < t_{i-1};\ i = (1..n), \tag{6.1}$$

and $t_n$ is the proper time corresponding to the maximum redshift that the beam will extend to. Particles are then read sequentially from the simulation outputs and tested for containment in the corresponding sections of the beam. The result is a continuous and "seamless" column of structure which clearly shows the emergence of larger over-densities and larger voids as one progresses through this column (see figure 6.1).

While BEAM has many potential uses, we will apply it to study the weak lensing effects on the Type-Ia supernovae. For this purpose we selected 10,000 randomly oriented lines of sight within each simulation. This removes bias due to the local environment of the observer from our weak lensing analysis of Type-Ia supernovae. Of the 15 available simulation outputs, 5 were used to span the redshift range in question. Since each output file is 1.6 Gbytes, storage requirements necessitated this compromise. However, for our weak lensing application, 5 outputs quite adequately captures the evolution of structure along the line of sight. In this way, we obtained a dataset of 10,000 random beams with a comoving radius

[1] http://www-hpcc.astro.washington.edu/stadel/Software/Beam

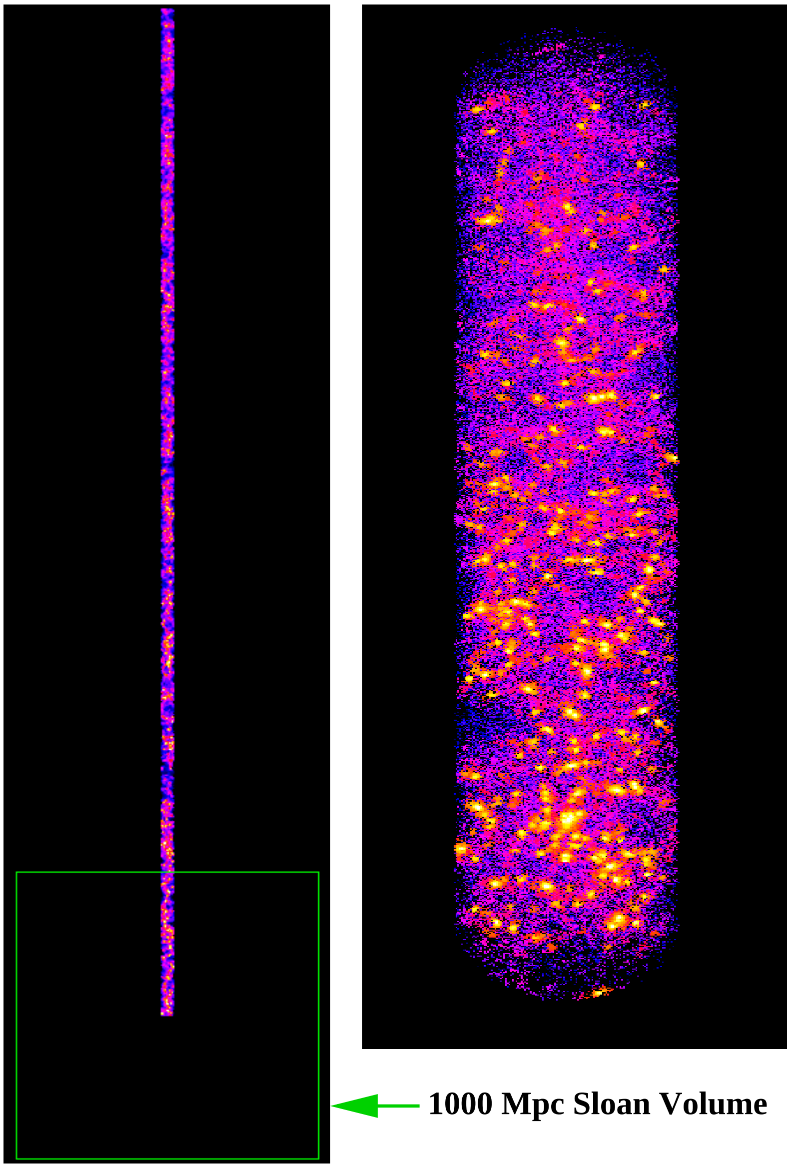


Figure 6.1: A sample beam of radius 20 Mpc, extending to $z = 1$, extracted from the $\Omega = 1$ "Sloan–volume" ($\mathrm{Gpc}^3$) simulation. The right shows this beam looking down–the–barrel, enhancing the evolution of structure along the line of sight. The onset of cluster formation between $z = 1$ and the present is quite evident in this view.

of 20 Mpc, extending to $z = 1$ (a proper depth of $\approx 2.6$ Gpc), for each of the cosmological simulations discussed in Section 6.2.1.

### *6.2.3 Lensing*

Gravitational lensing by an extended distribution of matter is usually studied within the framework of the multiple lens plane approximation (Blandford & Narayan 1986; Kovner 1987) In order to determine the gravitational lensing effects of a particular beam of matter, we follow the evolution in the shape of a bundle of light as it travels from one end of the beam to the other. The procedure for this approach has been described in considerable detail elsewhere (see Schneider, Ehlers & Falco 1992; Jaroszyński, Park, Paczyński & Gott 1990; Lee, Babul, Kofman & Kaiser 1997).

As part of its execution, BEAM projects the column of particles along a line of sight onto a specified number of planes. The planes are spaced equidistant in comoving distance along the column and the matter surface density on each plane is generated by projecting a density kernel for each particle onto the pixels of the nearest plane. Just as in Smooth Particle Hydrodynamics (SPH), we find the use of the spherically symmetric spline kernel proposed by Monaghan and Lattanzio (1985) to be suitable for this purpose. The smoothed density contribution from a particle of mass $m$ is given by

$$\rho(r,h) = \frac{m}{\pi h^3} \left\{ \begin{array}{cc} 1 - \frac{3}{2}\left(\frac{r}{h}\right)^2 + \frac{3}{4}\left(\frac{r}{h}\right)^3 & 0 \le \frac{r}{h} < 1 \\ \frac{1}{4}\left(2 - \frac{r}{h}\right)^3 & 1 \le \frac{r}{h} < 2 \\ 0 & \frac{r}{h} \ge 2 \end{array} \right\}, \tag{6.2}$$

where $r$ is the distance from the position of the particle and $h$ is the smoothing scale of the particle. From this we can calculate the projected surface density, given by

$$\Sigma(r,h) = \frac{m}{\pi h^2} \left\{ \begin{array}{cc} 4f(\frac{1}{2}\frac{r}{h}) - f(\frac{r}{h}) & 0 < \frac{r}{h} < 1 \\ 4f(\frac{1}{2}\frac{r}{h}) & 1 \le \frac{r}{h} < 2 \\ 0 & \frac{r}{h} \ge 2 \end{array} \right\} \tag{6.3}$$

$$f(x) = \left(\frac{1}{2} - \frac{13}{4}x^2\right)\sqrt{1-x^2} - \left(3 + \frac{3}{4}x^2\right)x^2 \ln\frac{1+\sqrt{1-x^2}}{x}. \tag{6.4}$$

The smoothing scale of these density kernels, $h$, can be fixed or it can vary from particle to particle such that the kernel always encloses a certain number of neighboring particles.

We opted for the latter, defining $h$ such that a spherical volume of radius $2h$ about the particle encloses a fixed number (typically 32) of nearest neighbor particles. To do this quickly, BEAM relies upon the high performance smoothing tool, SMOOTH [2]. This variable smoothing is more suited to the Lagrangian nature of N-body simulations. However, smoothing in this way is likely to underestimate the "clumpiness" of dark matter below the resolution scale of the simulation. The magnitude of this bias and its effect on the weak lensing analysis is investigated in Section 6.3.

For the purpose at hand, we used 50 planes of 255x255 pixels from which several quantities depending solely on the projected mass distribution are derived. The first of these is the convergence, $\kappa$, which is a measure of the matter inside the ray bundle and results in a uniform expansion or contraction of the bundle. This quantity is given by

$$\kappa = \frac{\Sigma}{\langle\Sigma\rangle} \ , \tag{6.5}$$

where $\langle\Sigma\rangle$ is the projection of the mean universal density of the plane. Gravitational tidal forces by matter outside of the ray bundle acts to distort it asymmetrically, resulting in shearing of the ray bundle. The shear matrix is completely described by two components, $S_{11}$ and $S_{12}$, which are given by

$$S_{11} = \frac{1}{\pi}\sum_{i,j\neq 0,0} \kappa_{ij}\frac{i^2-j^2}{(i^2+j^2)^2} \quad S_{12} = \frac{1}{\pi}\sum_{i,j\neq 0,0} \kappa_{ij}\frac{2ij}{(i^2+j^2)^2} \tag{6.6}$$

where the summation is taken over the surface mass density pixels, centered upon the location where the line of sight intersects the plane. The summation is only undertaken over the $(+,+)$ quadrant, utilizing several symmetries to cover the plane. We note that the pixels are only used in calculating the shear and that the convergence, which is simply the normalized surface density along the line of sight, depends only on the form of smoothing and not the choice of pixel scale.

With these, we can define a magnification matrix for a plane, normalizing it with respect to the cosmological parameters;

$$\mathbf{A}_i = \frac{3}{2}(1+z_i)^3 r(0,z_i)\Delta r_i^p \Omega_0 \begin{pmatrix} \kappa_i - S_{11,i} & S_{12,i} \\ S_{12,i} & \kappa_i + S_{11,i} \end{pmatrix} , \tag{6.7}$$

---

[2]http://www-hpcc.astro.washington.edu/stadel/Software/Smooth

where $\Delta r_i^p$ is the proper width of the region that was collapsed onto the plane, and $r(0, z_i)$ is the angular diameter distance between an observer at redshift 0 and the plane (note that both these have been normalized such that $D = cr/H_0$). The angular diameter distances employed here are 'empty beam' ([20, 21, 22]). For a $\Omega = 1$, the empty beam angular diameter distances are given by

$$\begin{aligned} r(z_1, z_2) &= (1 + z_1)\left[w(z_2) - w(z_1)\right] \\ w(z) &= \tfrac{2}{5}\left[1 - (1+z)^{-\frac{5}{2}}\right]. \end{aligned} \tag{6.8}$$

For more general cosmological models $r(z_1, z_2)$ must be calculated numerically (see Linder *et al.* 1988). The final step is to combine the the properties of each plane to produce an over all evolution of the ray bundle as it traverses the beam. This procedure is recursive and is of the form;

$$\mathbf{B}_k = \mathbf{I} - \sum_{j=1}^{k-1} \frac{r(z_j, z_k)}{r(0, z_k)} \mathbf{A}_j \mathbf{B}_j \ , \tag{6.9}$$

where $\mathbf{B}_1 = \mathbf{I}$. The magnification of the ray bundle emanating from plane j, with respect to the empty beam universe, is simply $1/Det(\mathbf{B}_j)$, and the apparent angular diameter distance is $\sqrt{Det(\mathbf{B}_j)}$ of that in the empty beam universe.

The agreement of this multiple lensing plane approximation with the analytical result for a perfectly homogeneous universe is excellent when 50 planes are used. The error introduced in the magnification of a $z = 1$ source is less than 0.01% in this case. To ensure that the shear contribution is correctly calculated we used a single plane with a model surface density distribution (see Schneider, Ehlers & Falco 1992, pages 244-246) and compared the resulting magnification to the analytical result. Although the accuracy of the shear contribution depends on the number of pixels and the extent of the lensing plane being considered, we converge on the analytical result as these factors are increased. When considering the shear due to large scale structure we are mainly concerned about the extent about the beam over which it is calculated. We have limited our calculation of the shear to a radius of 20 Mpc about the line of sight, which captures most of the shear due to large scale structure. Since the shear introduces only a small correction to the magnification when compared with the net convergence, this restriction has no implication for the results presented.

### 6.3 Testing the Robustness of the Results: Resolution and Smoothing

One of the primary concerns that we faced was the question of how much the structure below the resolution scale of our simulations could affect the resulting weak lensing magnification distributions. Such structures clearly have implications for the few strongly lensed lines of sight, but how much bias this could introduce to the majority of weakly lensed lines of sight was not clear. The simulations needed to be large enough to capture the largest voids which act to deamplify the light with respect to a uniform matter distribution. At the same time the smallest structures resolved by the simulations have to extend down to a mass scale where their filling factor is sufficiently small so as to have a minimal impact on the net convergence of the majority of lines of sight. The requirements of both a large volume and high resolution mandated the supercomputer scale N-body simulations we used. However, in order to investigate the effect of resolution on our results, we performed a higher resolution $\Omega = 1$ simulation in a smaller volume of 100 $\mathrm{Mpc}^3$ with 64 times better mass resolution than the $\mathrm{Gpc}^3$ simulations.

In figure 6.2 we compare the convergence distributions obtained from the $\mathrm{Gpc}^3$ "Sloan–Volume" simulation with the smaller volume, higher resolution simulation. Note that this is the distribution of the convergence, *kappa*, due to each individual lensing plane at the indicated redshift. We see that the increase in resolution does produce a non-negligible change in the skewness of this distribution. This means that, for any individual lensing plane, the resolution of the large volumes used was not quite high enough to capture the detailed character of the convergence distribution. However, when we consider the net effect of all planes between $z = 0$ and $z = 1$ on the final magnification distribution of a $z = 1$ source, this difference becomes less significant. We show this in figure 6.3, where the solid and dashed lines show the magnification distributions for the $\Omega = 1$ "Sloan-Volume" and 100 $\mathrm{Mpc}^3$ high resolution simulations respectively. The resolution of our large volume simulations appear to be sufficient for obtaining the general character of the magnification distributions, although the distribution obtained for the higher resolution simulation does tend toward slightly lower magnifications.

In addition to gauging the effects of resolution in the simulations, we also needed to

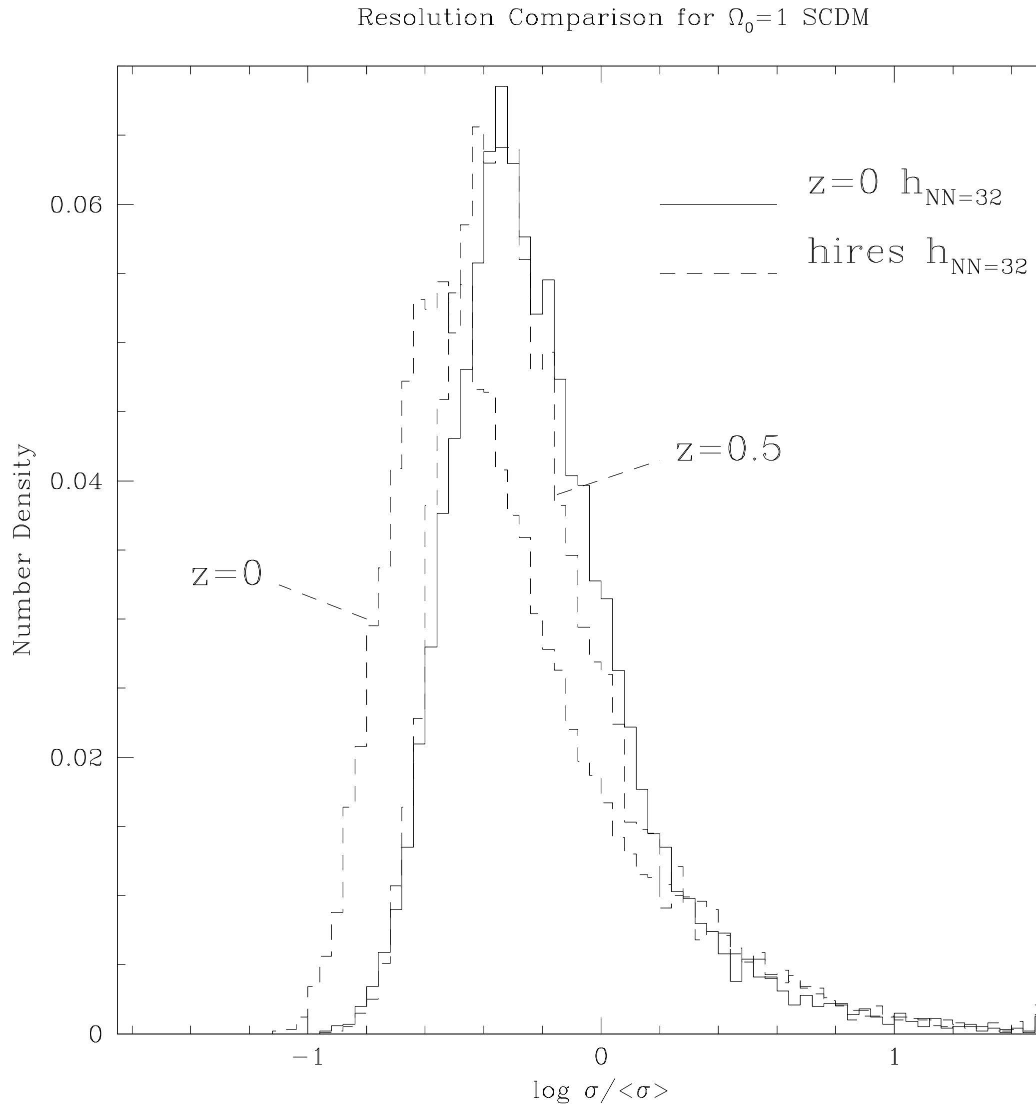


Figure 6.2: A comparison of the effect of simulation resolution on the convergence distribution. The solid line shows the convergence distribution of a single lensing plane at redshift zero for the $\Omega = 1$ "Sloan–Volume" simulation. The dashed line shows the same, for two different redshifts of the higher resolution, smaller volume, simulation.

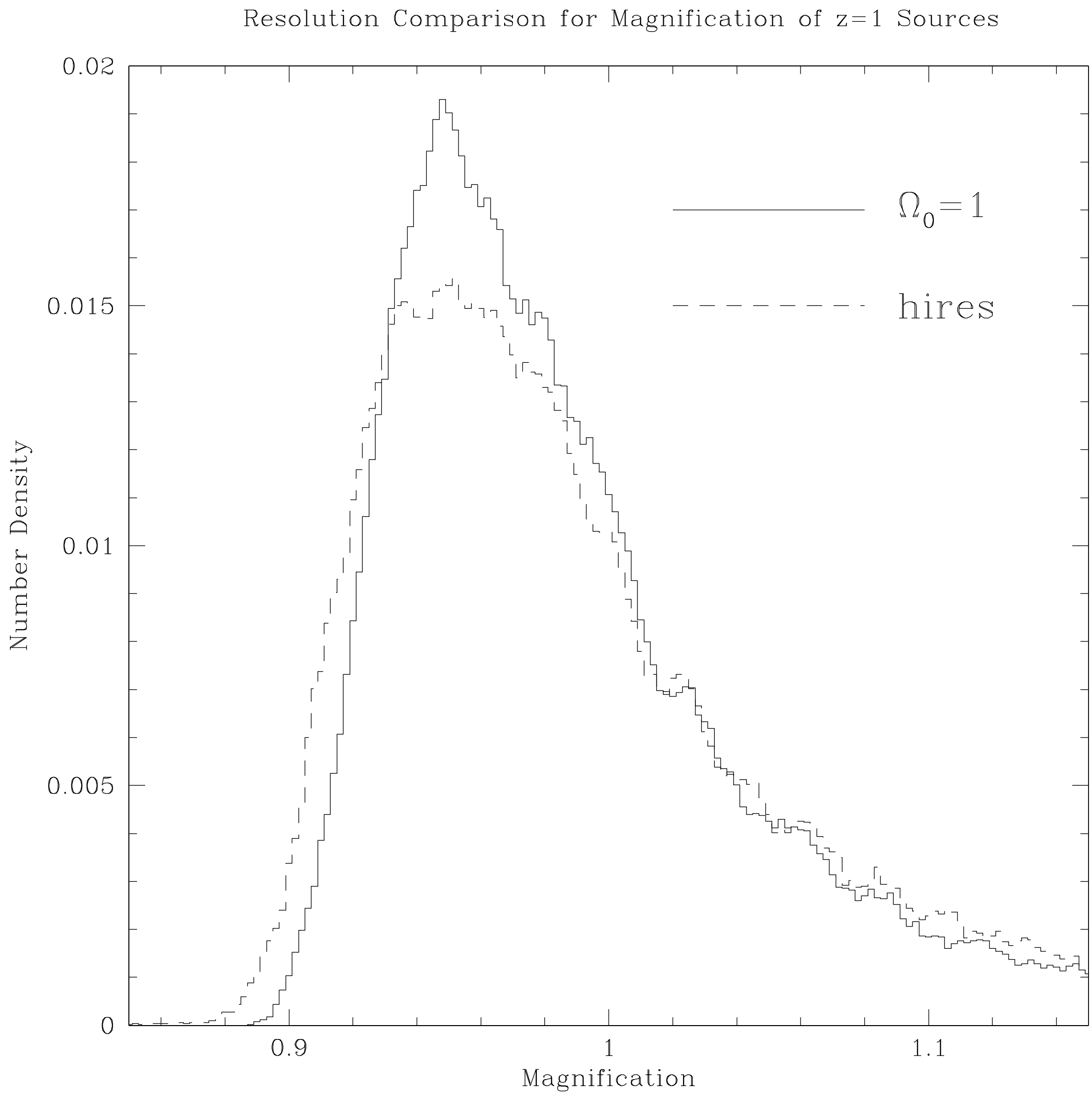


Figure 6.3: Resolution comparison for the magnification of $z = 1$ sources in an $\Omega = 1$ cosmological model. The solid and dashed lines show the magnification distributions for the $\Omega = 1$ "Sloan-Volume" and 100 $\mathrm{Mpc}^3$ high resolution simulations respectively.

examine the choice of smoothing used in calculating the projected surface density at each lensing plane. To study the impact of this we performed all of the analysis with a smaller smoothing scale for each of the models. Figure 6.4 compares the convergence from a single lensing plane for different smoothing. Just as in figure 6.2, the solid curve shows the convergence distribution for a single plane at redshift zero in the large volume $\Omega = 1$ SCDM simulation. It corresponds to a smoothing where $2h$ encloses 32 nearest neighbor particles and $h$ is used as the smoothing scale for the kernel in equation 6.3. The dashed curve shows the same but with the smoothing scale set to half the previous value, giving each particle a four fold increase in effective surface density. We can see that the change in smoothing scale, while producing a lower convergence tail, does little to change the general character of the distribution. This is also true for the other two cosmological models.

Finally, figure 6.2 also shows the interesting fact that the convergence distribution of the high resolution simulation at $z = 0.5$ matches almost exactly that for the larger volume at $z = 0$. This indicates that the larger volume is acting like a cosmological model of a lower *effective* normalization, in this case $\sigma_8/1.5$, when the convergence of a single plane is being considered. Again, we note that the final magnification distribution is not strongly affected by this, a fact which is both good and bad. On one hand, it seems to imply that the magnification distributions are not very sensitive to the details of the large scale structure. However, as a cosmological test, this is exactly what we want, a measure which depends on very few parameters.

## 6.4 Results

### 6.4.1 Convergence Distribution

Comparing the single plane convergence distribution of the three cosmological models in figures 6.5-6.7 we see the effects of the different look-back histories in the evolution of structure. This is most pronounced in figure 6.7 for the $\Omega_0 = 0.3, \Omega_\Lambda = 0.7$ model. The evolution in the convergence distribution is continuous in the two low $\Omega_0$ models. However, the lack of evolution in the convergence distribution for the $\Omega = 1$ model between redshift 0 and 0.5 in figure 6.5 is striking, particularly in light of the fact that this is the era when

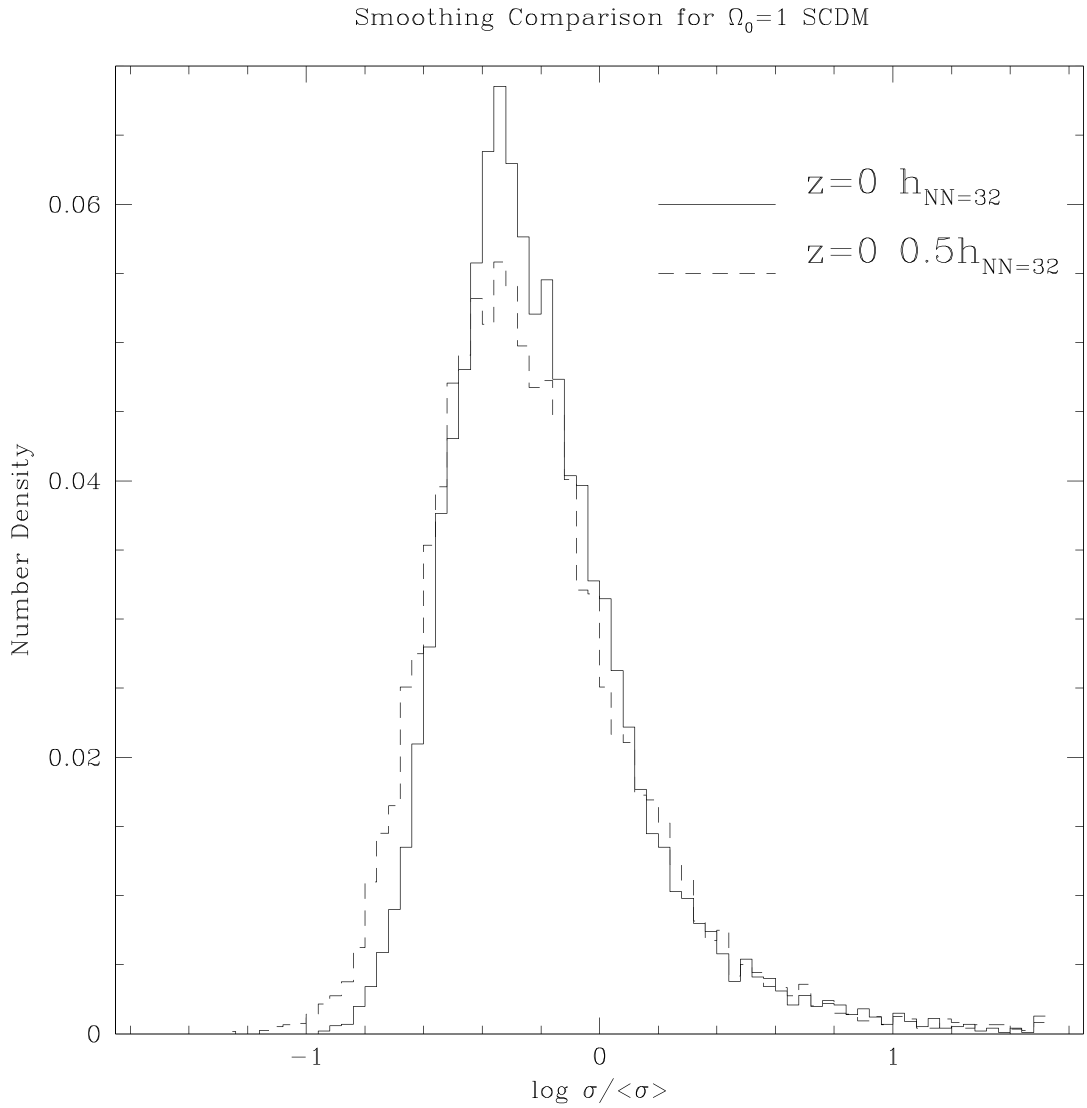


Figure 6.4: A comparison of the the effect of different smoothing on the convergence distribution. The solid line shows the convergence distribution of a single lensing plane at redshift zero for the $\Omega = 1$ "Sloan–Volume" simulation. The dashed line shows the same, but with the density kernel of particles reduced by a factor of 2.

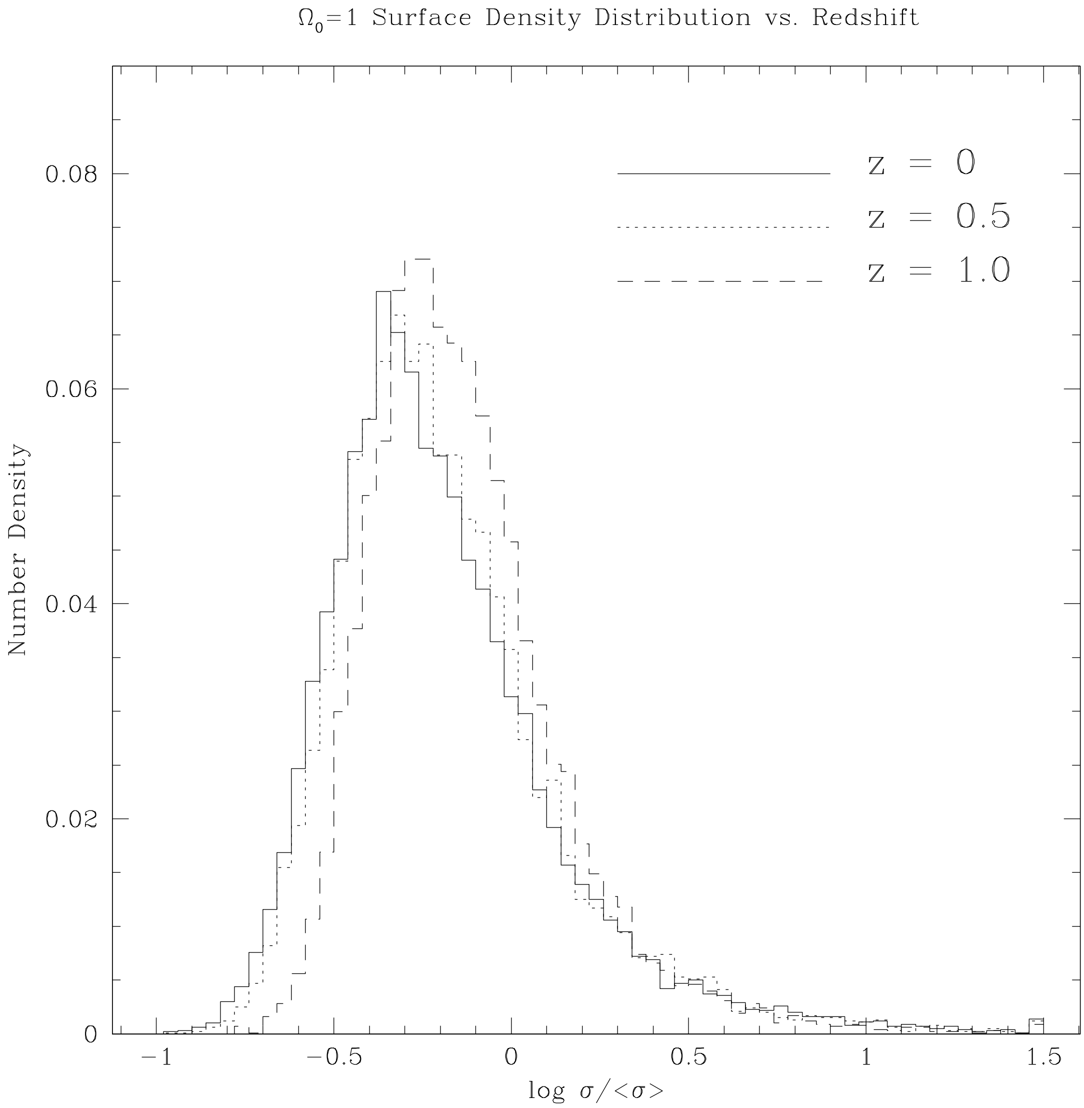


Figure 6.5: $\Omega_0 = 1$ log–convergence distribution for a single lensing plane at $z = 0$, 0.5 and 1.0.

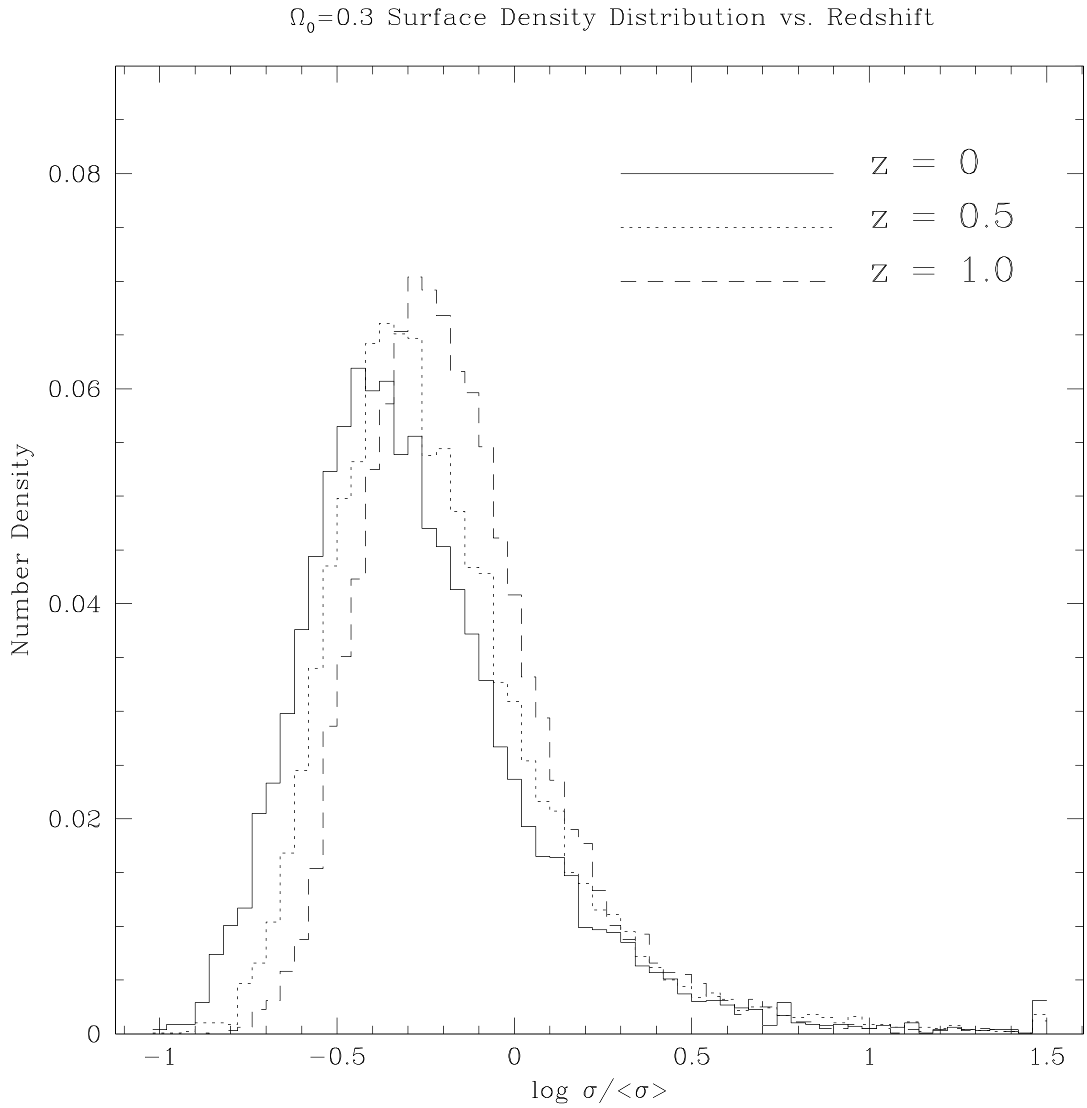


Figure 6.6: $\Omega_0 = 0.3$ log–convergence distribution for a single lensing plane at $z = 0$, 0.5 and 1.0.

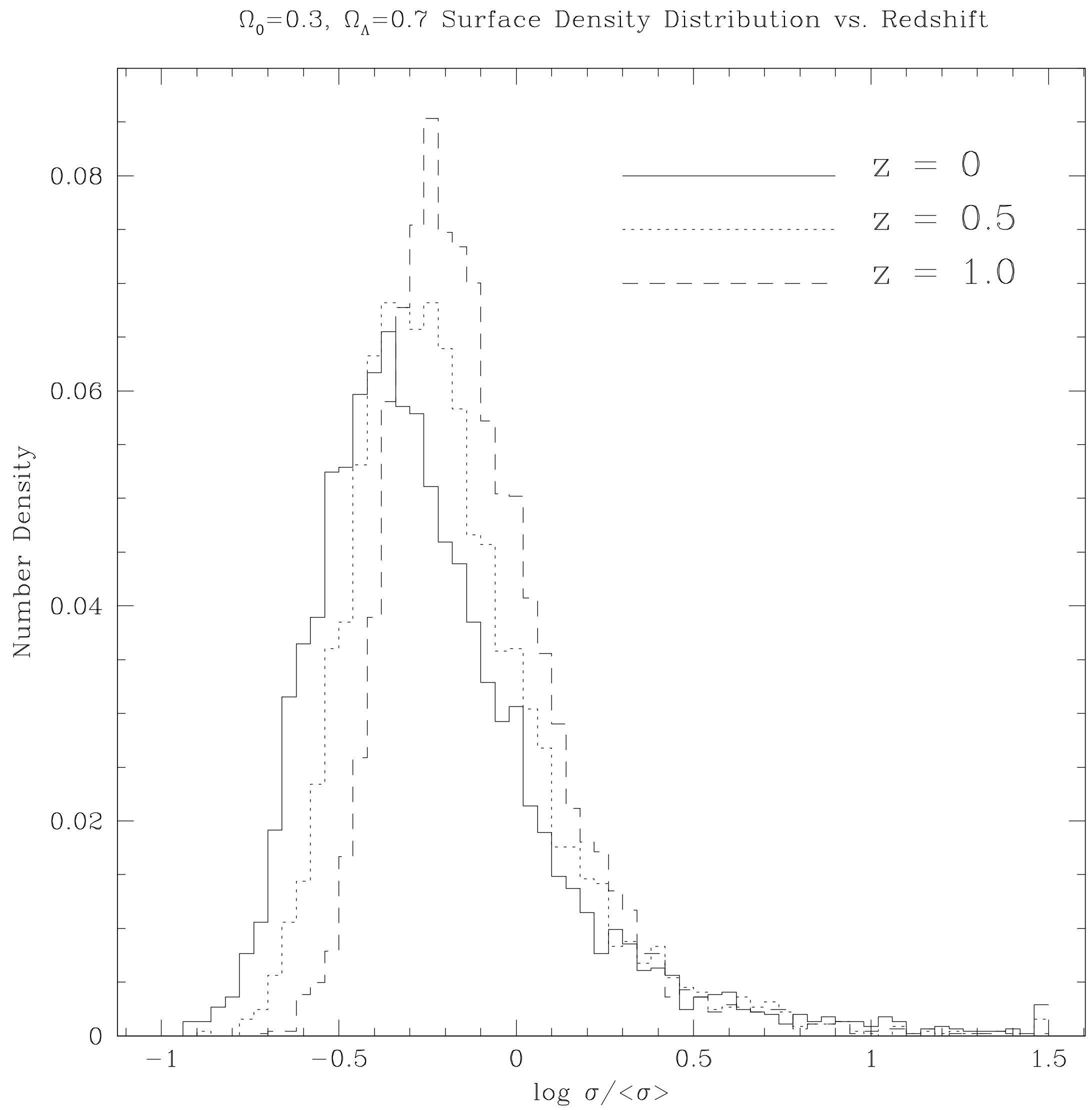


Figure 6.7: $\Omega_0 = 0.3, \Omega_\Lambda = 0.7$ log–convergence distribution for a single lensing plane at $z = 0$, 0.5 and 1.0.

cluster formation is taking place. This may indicate that the assembly of walls and filaments is of more importance to the weak lensing statistics than the formation of clusters, since their filling factor is quite small. However, it could also indicate a more general speed with which space on a large range of scales is emptied. It is clear from these distributions that the look-back history of structure evolution should not be ignored when considering the weak lensing amplification/deamplification of high redshift sources.

The convergence distribution of a single plane is a highly skewed one (the distributions are plotted are of $\log \sigma/\langle\sigma\rangle$), yet the surface density along the entire line of sight is very close to the mean. Most of a line of sight passes through under-dense regions occasionally piercing small regions of high density such that the resulting magnification of a high redshift source is generally quite close to that predicted for a uniform cosmology. Although this result is not surprising, we can define a path length over which a line of sight will generally make up the difference. This is also a length scale on which we can consider the universe to be homogeneous in the mean as far as the convergence is concerned. This length scale is about 400 Mpc, showing that simulations of about this scale should be used for studying this type of weak lensing statistic. *This should be defined more precisely! It is a bit hand-wavy, but the main point is the last one.*

### *6.4.2 Magnifications Distribution*

Figures 6.8 and 6.9 show the distribution of the resulting weak lensing magnifications relative to that predicted by a uniform mass distribution, for sources at $z = 1$ and $z = 0.5$ respectively. We note that the prediction for a homogeneous cosmological model would be a delta function at unit magnification in these plots. The distributions shown are the results of drawing over 10,000 random lines of sight from each of the three cosmological N-body simulations. These figures show that the weak lensing effect of the intervening large scale structure produces a distribution which is skewed towards lower magnifications, with a sharp decline at lower magnification and an extensive high magnification tail. A small sample of high redshift standard candles will appear to be slightly fainter than predicted for a homogeneous cosmological model. For small samples this systematic bias is better

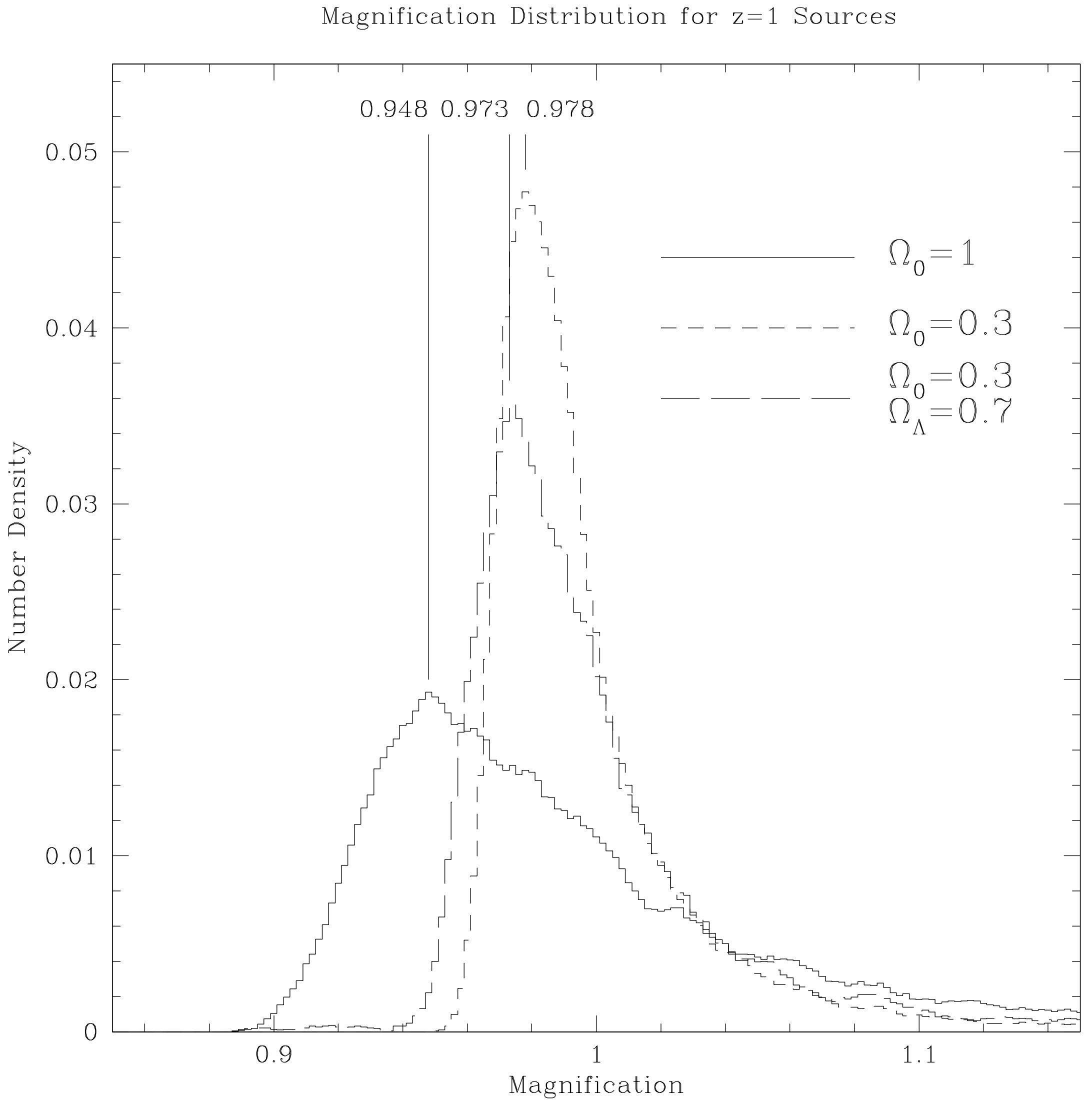


Figure 6.8: Distribution of weak lensing magnifications relative to that predicted by a uniform mass distribution, for sources at $z = 1$. The resulting distributions for the three cosmological models are shown including the smaller volume, higher resolution $\Omega = 1$ simulation.

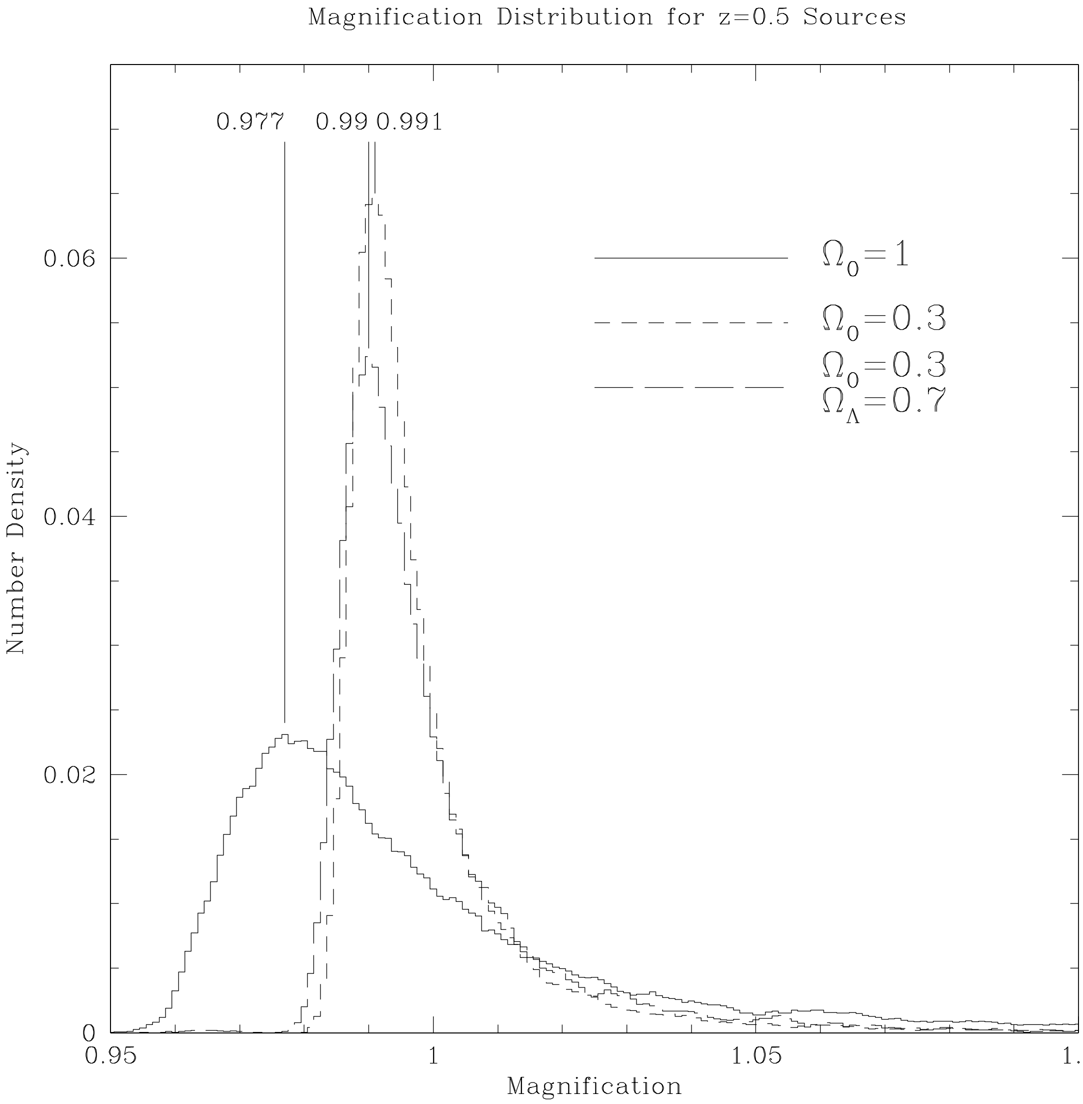


Figure 6.9: Distribution of weak lensing magnifications relative to that predicted by a uniform mass distribution, for sources at $z = 0.5$. The resulting distributions are shown for the three cosmological models.

characterized by the modes of these skewed distributions, which we indicate at the top of figures 6.8 and 6.9. The mode of the distributions correspond to a bias in magnitude of $\Delta m = +0.058, +0.030, +0.024$ for $z = 1$ sources and $\Delta m = +0.025, +0.011, +0.010$ for $z = 0.5$ sources for the each of the critical, flat–lambda, and open cosmological models respectively. Clearly, the effects of weak lensing on high redshift sources becomes much more pronounced at higher redshift. As has previously been pointed out this bias is insufficient to weaken the case for an accelerating universe that is implied by the high redshift Type-Ia SNe data (Riess *et al.* 2000).

From both figures we see that the spread in the distributions of both of the low $\Omega_0$ models is smaller than the $\Omega = 1$ model. The fact that the spread in the magnification distribution is almost independent of $\Lambda$, or an almost pure measure of $\Omega_0$, means that this could provide an independent cosmological test which could be applied to a large set of high redshift supernovae. We investigate this possibility in the final section of this paper. Finally, we point out that although the distribution for the flat–lambda model at $z = 0.5$ is essentially identical to the open model, by $z = 1$ it is beginning to distinguish itself. Such differences will become even more pronounced at greater redshifts.

## *6.5 Discussion*

It is clear that the effects of weak lensing deamplification due to large scale structure must be accounted for when using high redshift standard candles such as Type-Ia supernovae in determining cosmological parameters. This is particularly true as the limiting redshifts of such datasets are extended in future surveys, since the bias due to weak lensing becomes larger at larger redshifts. Aside from introducing a bias to the predicted magnifications of high redshift standard candles, the weak lensing due to large scale structure introduces a significant spread in their brightness. If this spread can be detected through any intrinsic dispersion in the brightness of the standard candle, then it can be used as a powerful cosmological test. We examine only the simplest application of this idea, that of simply distinguishing two models with differing $\Omega_m$.

Figure 6.10 shows the "smearing" effect of the intrinsic brightness distribution of Type-

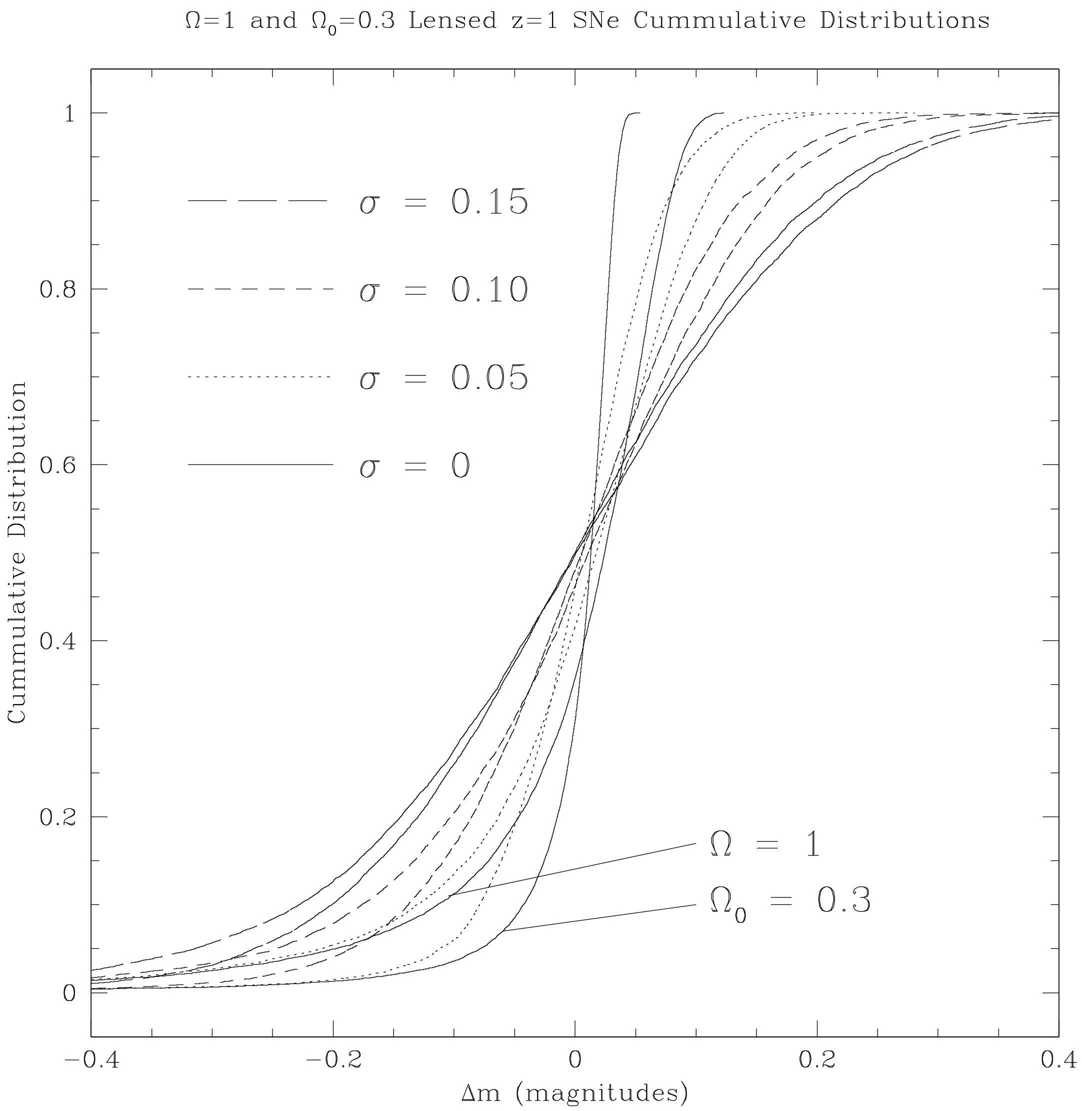


Figure 6.10: Cumulative brightness distributions for the $\Omega = 1$ and $\Omega_0 = 0.3$ models as a function of the intrinsic dispersion in the high redshift standard candles. A larger intrinsic dispersion acts to smear out the effect of the weak lensing induced dispersion due to large scale structure.

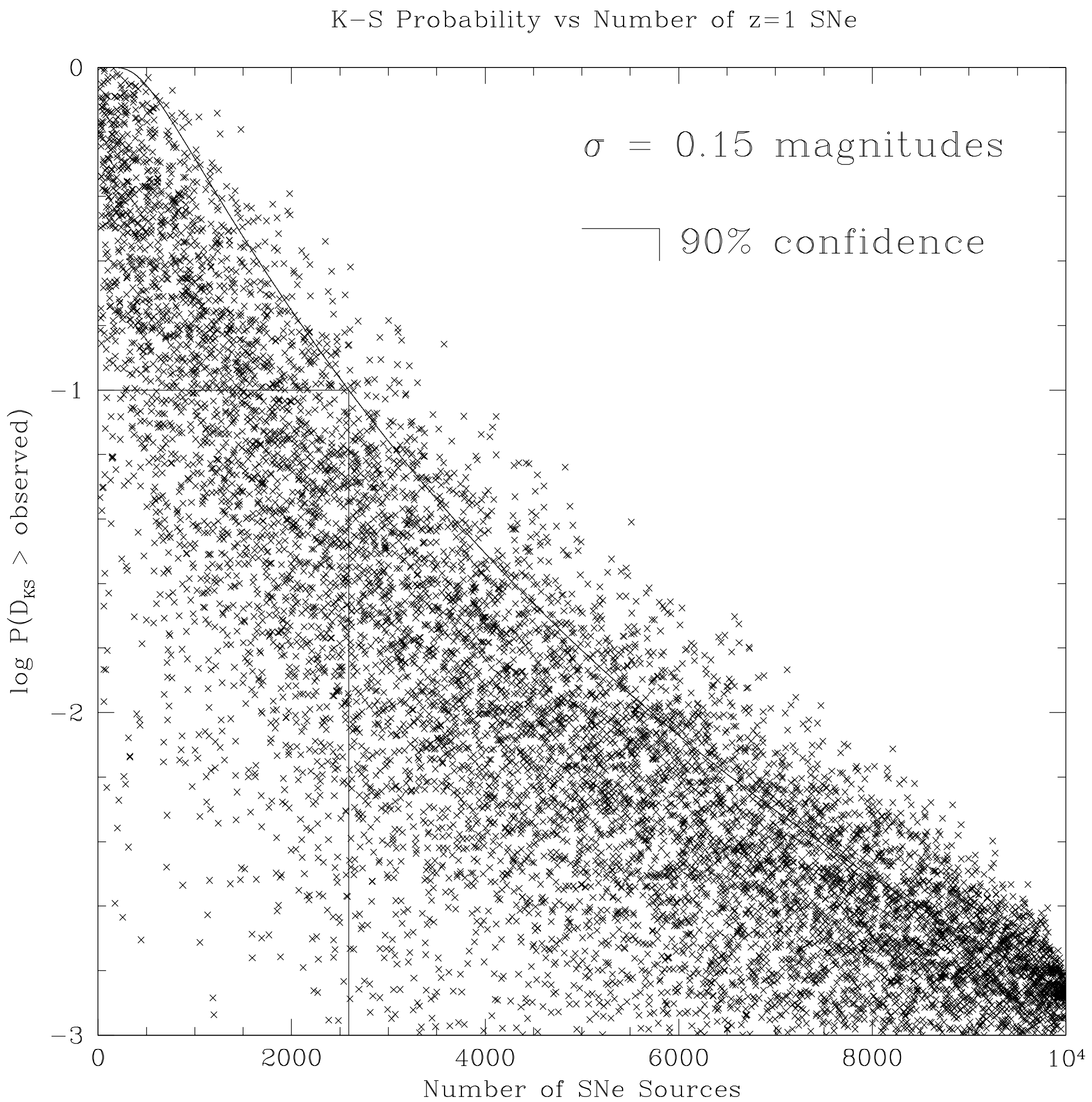


Figure 6.11: Log of the K–S probability that the brightness distributions of an $\Omega = 1$ and $\Omega_0 = 0.3$ model are indistinguishable for an intrinsic dispersion of $\sigma = 0.15$ in the distance modulus of the standard candles. This is shown as a function of the number of standard candles in the sample. The curve shows the analytical K–S probability while the crosses indicate a Monte Carlo check on the validity of the K–S statistic using randomly drawn distributions. We indicate the number of standard candles needed to distinguish the two models at a 90% confidence level.

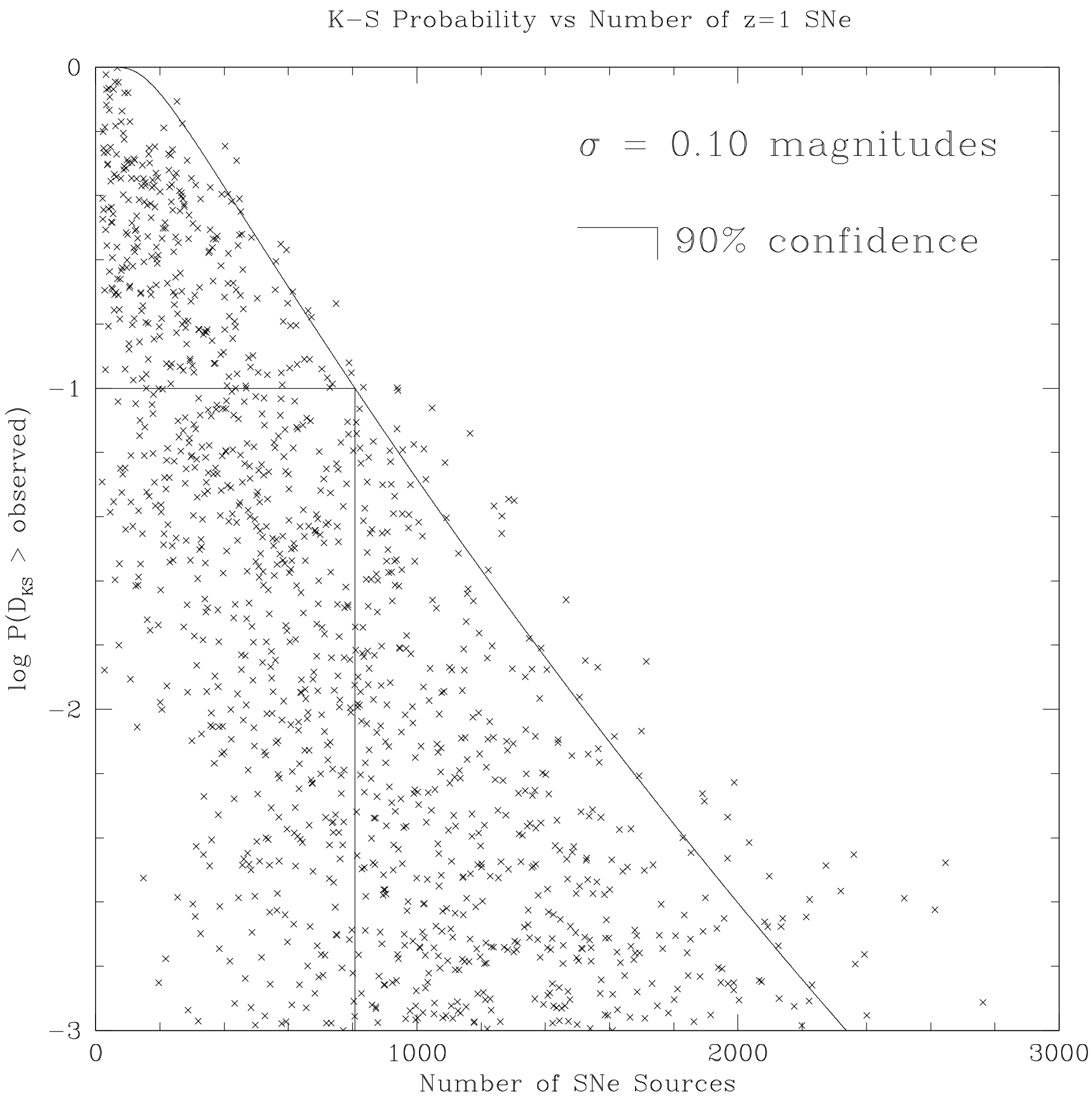


Figure 6.12: Log of the K–S probability that the brightness distributions of an $\Omega = 1$ and $\Omega_0 = 0.3$ model are indistinguishable for an intrinsic dispersion of $\sigma = 0.10$ in the distance modulus of the standard candles. This is shown as a function of the number of standard candles in the sample. The curve shows the analytical K–S probability while the crosses indicate a monte–carlo check on the validity of the K–S statistic using randomly drawn distributions. We indicate the number of standard candles needed to distinguish the two models at a 90% confidence level.

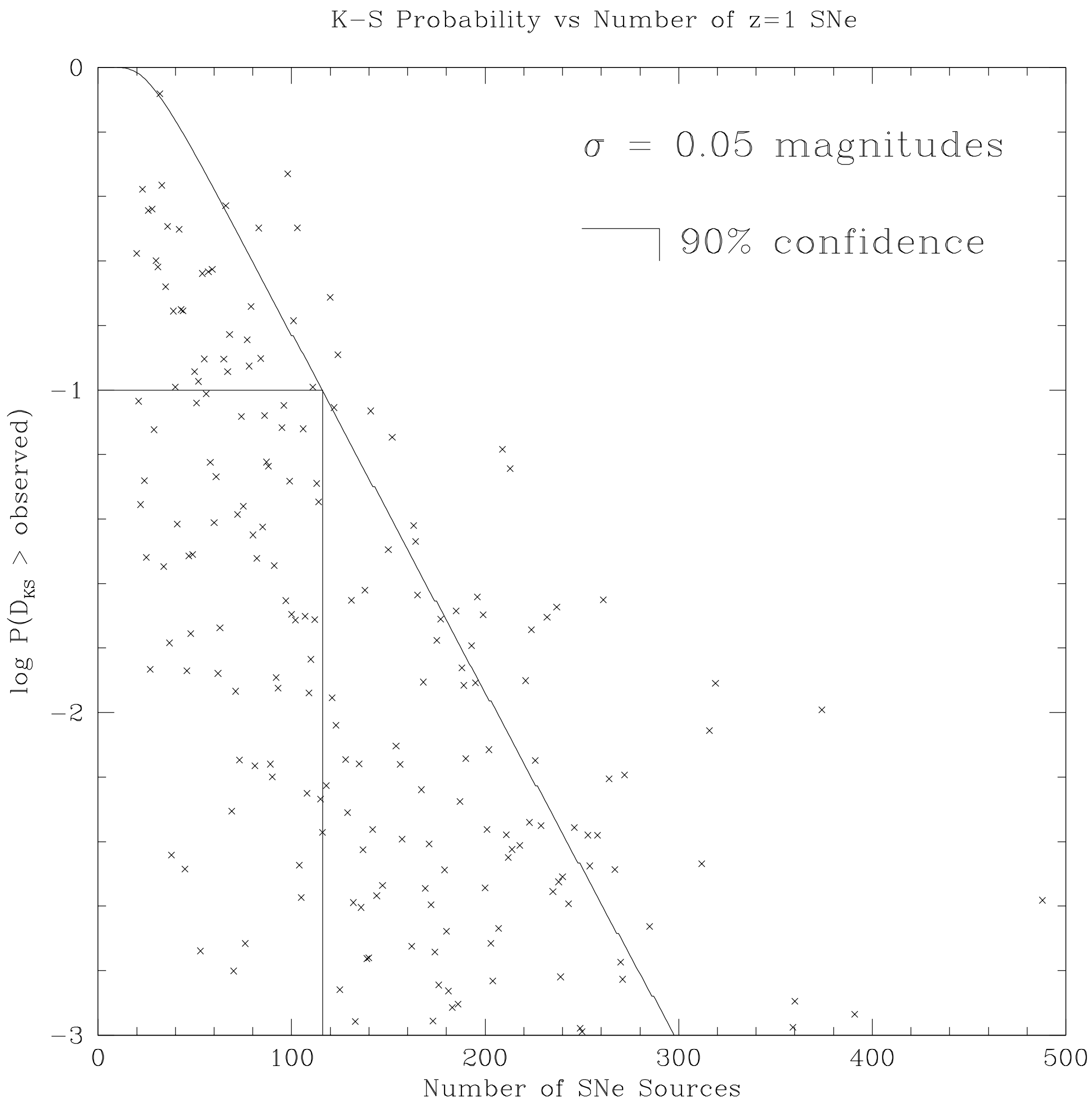


Figure 6.13: Log of the K–S probability that the brightness distributions of an $\Omega = 1$ and $\Omega_0 = 0.3$ model are indistinguishable for an intrinsic dispersion of $\sigma = 0.05$ in the distance modulus of the standard candles. This is shown as a function of the number of standard candles in the sample. The curve shows the analytical K–S probability while the crosses indicate a monte–carlo check on the validity of the K–S statistic using randomly drawn distributions. We indicate the number of standard candles needed to distinguish the two models at a 90% confidence level.

Ia supernovae on the cumulative magnification distribution due to weak lensing. Clearly as the intrinsic dispersion is increased it becomes more an more difficult to distinguish an $\Omega = 1$ model from and $\Omega_0 = 0.3$ model. A K-S test allows us to answer the question of how many supernovae are needed to distinguish the two models from the brightness distribution alone, given an intrinsic dispersion (after correction) of the distance modulus of the supernova sample. Figures 6.11–6.13 show the K–S probability that the two distributions are indistinguishable as a function of the number of supernovae in the sample. We find that for the present intrinsic dispersion in the supernova distance modulus of $\sigma \sim 0.15$ that a sample of order 2500 high redshift supernovae would be required to differentiate the two models. This is indeed a very large sample, and makes the prospects for applying such a test difficult. However, should further astrophysical or empirical corrections to Type-Ia supernovae brightness become possible such that the intrinsic dispersion could be reduced to 0.1 or 0.05 magnitudes, prospects for such a test improve greatly. Furthermore, these weak lensing effects increase at higher redshift, so that a smaller sample of very high redshift supernovae could also make such a test possible. We point out that this cosmological test would be useful because it is independent of many factors which complicate the interpretation of the supernovae data, such as the $\Omega$–$\Lambda$ degeneracy and the possibility of some form of uniform grey absorption in the universe. Since microwave background experiments will place very strong constrains on $\Omega_m$ and $\Omega_\Lambda$, this method could potentially also be used to test whether the intrinsic dispersion of Type-Ia supernovae changes at higher redshift. With current proposals in existence for increasing the high redshift supernovae dataset by up to two orders of magnitude and extending to higher redshifts (see for example: http://snap.lbl.gov/ where the proposed rate of discovery will be 2000 SNe per year out to $z = 1.7$) using the underlying weak lensing magnification as a cosmological or astrophysical test is worth consideration.

# BIBLIOGRAPHY

[1] R. Anderson. Tree data structures for n–body simulations. SIAM Journal on Computing, 28(6):1923–1940, 1999.

[2] A. W. Appel. SIAM J. Sci. Stat. Comput., 6:85+, 1985.

[3] V. I. Arnold. *Mathematical methods of classical mechanics.* Graduate texts in mathematics, New York: Springer, 1978, 1978.

[4] A. Babul and M. H. Lee. Gravitational lensing by smooth inhomogeneities in the universe. *MNRAS*, 250:407–413, May 1991.

[5] J. Barnes and P. Hut. A hierarchical o(nlogn) force-calculation algorithm. *Nature*, 324:446–449, December 1986.

[6] M. Bartelmann and P. Schneider. A large-scale structure model for gravitational lensing. *AAP*, 259:413–422, June 1992.

[7] C. M. Baugh, S. Cole, C. S. Frenk, and C. G. Lacey. The epoch of galaxy formation. *ApJ*, 498:504+, May 1998.

[8] J. L. Bentley. Multidimensional binary search trees in database applications. *IEEE Transactions on Software Engineering*, SE-5(4):333–340, July 1979.

[9] E. Bertschinger and J. M. Gelb. Cosmological n-body simulations. *Computers and Physics*, 5:164–175, April 1991.

[10] R. Blandford and R. Narayan. Fermat's principle, caustics, and the classification of gravitational lens images. *ApJ*, 310:568–582, November 1986.

[11] R. D. Blandford, A. B. Saust, T. G. Brainerd, and J. V. Villumsen. The distortion of distant galaxy images by large-scale structure. *MNRAS*, 251:600–627, August 1991.

[12] R. Cadonau, G. A. Tammann, and A. Sandage. Type i supernovae as standard candles. In *LNP Vol. 224: Supernovae as Distance Indicators*, pages 151–165, 1985.

[13] M. P. Calvo and J. M. Sanz-Serna. SIAM J. Sci. Comput., 14:936+, 1993.

[14] P. J. Channell and C. 1990 Scovel. Nonlinearity, 3:231+, 1990.

[15] H. M. P. Couchman. Mesh-refined p3m - a fast adaptive n-body algorithm. *ApJ Letters*, 368:L23–L26, February 1991.

[16] H. M. P. Couchman, A. J. Barber, and P. A. Thomas. Measuring the three-dimensional shear from simulation data, with applications to weak gravitational lensing. *MNRAS*, 308:180–200, September 1999.

[17] W. Dehnen. A very fast and momentum-conserving tree code. *ApJ Letters*, 536:L39–L42, June 2000.

[18] H. Q. Ding, N. Karasawa, and Goddard W. A. The reduced cell multipole method for coulomb interactions in periodic systems with million-atom unit cells. *Chemical Physics Letters*, 196:6–10, August 1992.

[19] J. Dubinski. A parallel tree code. *New Astronomy*, 1:133–147, October 1996.

[20] C. C. Dyer and R. C. Roeder. The distance-redshift relation for universes with no intergalactic medium. *ApJ Letters*, 174:L115–+, June 1972.

[21] C. C. Dyer and R. C. Roeder. Distance-redshift relations for universes with some intergalactic medium. *ApJ Letters*, 180:L31–+, February 1973.

[22] C. C. Dyer and R. C. Roeder. Observations in locally inhomogeneous cosmological models. *ApJ*, 189:167–176, April 1974.

[23] G. Efstathiou, M. Davis, S. D. M. White, and C. S. Frenk. Numerical techniques for large cosmological n-body simulations. *ApJ Supp.*, 57:241–260, February 1985.

[24] M. W. J. Ewell. *Dissipative N-body simulations of the formation of single galaxies in a cold dark matter cosmology.* PhD thesis, Princeton Univ., NJ., 1988.

[25] J. A. Frieman and *et al.* . Group report: are we making progress in relating cosmology and fundamental theories? In *The Evolution of the Universe: report of the Dahlem Workshop on the Evolution of the Universe*, pages 141+, 1997.

[26] P. M. Garnavich and *et al.* . Constraints on cosmological models from hubble space telescope observations of high-z supernovae. *ApJ Letters*, 493:L53–+, February 1998.

[27] J. M. Gelb. *Large Simulations of Gravitational Clustering in the Universe.* PhD thesis, MASSACHUSETTS INSTITUTE OF TECHNOLOGY., January 1992.

[28] S. Ghigna, B. Moore, F. Governato, G. Lake, T. Quinn, and J. Stadel. Dark matter haloes within clusters. *MNRAS*, 300:146–162, October 1998.

[29] F. Governato, C. M. Baugh, C. S. Frenk, S. Cole, C. G. Lacey, T. Quinn, and J. Stadel. The seeds of rich galaxy clusters in the universe. *Nature*, 392:359–361, 1998.

[30] F. Governato, J. P. Gardner, J. Stadel, T. Quinn, and G. Lake. On the origin of early-type galaxies and the evolution of the interaction rate in the field. *AJ*, 117:1651–1656, April 1999.

[31] F. Governato, B. Moore, R. Cen, J. Stadel, G. Lake, and T. Quinn. The local group as a test of cosmological models. *New Astronomy*, 2:91–106, May 1997.

[32] L. Greengard. PhD thesis, Yale Univ., CT., 1987.

[33] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. J. Comp. Phys., 73:325–348, December 1987.

[34] J. E. Gunn. On the propagation of light in inhomogeneous cosmologies. i. mean effects. *ApJ*, 150:737+, December 1967.

[35] T. Hamana, H. Martel, and T. Futamase. Statistics of weak gravitational lensing in cold dark matter models: Magnification bias on quasar luminosity functions. *ApJ*, 529:56–68, January 2000.

[36] M. Hamuy and *et al.* . The morphology of type ia supernovae light curves. *AJ*, 112:2438+, December 1996.

[37] L. Hernquist, F.R̃. Bouchet, and Y. Suto. Application of the ewald method to cosmological n-body simulations. *ApJ Supp.*, 75:231–240, February 1991.

[38] L. Hernquist and N. Katz. Treesph - a unification of sph with the hierarchical tree method. *ApJ Supp.*, 70:419–446, June 1989.

[39] R. W. Hockney and J. W. Eastwood. *Computer Simulation Using Particles*. Computer Simulation Using Particles, New York: McGraw-Hill, 1981, 1981.

[40] D. E. Holz. Lensing and high-z supernova surveys. *ApJ Letters*, 506:L1–L5, October 1998.

[41] P. Hut, J. Makino, and S. McMillan. Building a better leapfrog. *ApJ Letters*, 443:L93–L96, April 1995.

[42] B. Jain and U. Seljak. Cosmological model predictions for weak lensing: Linear and nonlinear regimes. *ApJ*, 484:560+, July 1997.

[43] B. Jain, U. Seljak, and S. White. Ray-tracing simulations of weak lensing by large-scale structure. *ApJ*, 530:547–577, February 2000.

[44] M. Jaroszynski, C. Park, B. Paczynski, and J. R. I. Gott. Weak gravitational lensing due to large-scale structure of the universe. *ApJ*, 365:22–26, December 1990.

[45] W. Kahan. *Lecture Notes.* CS Division, Dept. of EECS, UC Berkeley, October 1993.

[46] W. Kahan and R-C. Li. J. Comp. Phys., 134:316+, 1997.

[47] N. Kaiser. Weak gravitational lensing of distant galaxies. *ApJ*, 388:272–286, April 1992.

[48] M. Kamionkowski, A. Babul, C. M. Cress, and A. Refregier. Theory and statistics of weak lensing from large-scale mass inhomogeneities. *MNRAS*, 301:1064–1072, December 1998.

[49] R. Kantowski. The effects of inhomogeneities on evaluating the mass parameter omega_m and the cosmological constant lambda. *ApJ*, 507:483–496, November 1998.

[50] R. Kantowski, T. Vaughan, and D. Branch. The effects of inhomogeneities on evaluating the deceleration parameter q 0. *ApJ*, 447:35+, July 1995.

[51] N. Katz, T. Quinn, E. Bertschinger, and J. M. Gelb. Formation of quasars at high redshift. *MNRAS*, 270:L71–+, October 1994.

[52] N. Katz, D. H. Weinberg, and L. Hernquist. Cosmological simulations with treesph. *ApJ Supp.*, 105:19+, July 1996.

[53] D. E. Knuth. *Sorting and Searching, Second Edition.* Addison-Wesley, 1998.

[54] Donald E. Knuth. *The TEX book.* Addison-Wesley, 1984.

[55] I. Kovner. The thick gravitational lens - a lens composed of many elements at different distances. *ApJ*, 316:52–69, May 1987.

[56] G. Lake, N. Katz, T. Quinn, and J. Stadel. In *Proceeding of the 7th SIAM conference on Parallel Processing for Scientific Computing*, pages 307+, 1995.

[57] M. H. Lee, M. J. Duncan, and H. F. Levison. A multiple timestep symplectic algorithm for integrating close encounters in planetary systems. In *American Astronomical Society Meeting*, volume 191, pages 6903+, December 1997.

[58] E. V. Linder, R. V. Wagoner, and P. Schneider. Gravitational lensing statistics of amplified supernovae. *ApJ*, 324:786–793, January 1988.

[59] Y. Mellier. Probing the universe with weak lensing. *ARAA*, 37:127–189, 1999.

[60] R. B. Metcalf. The gravitational lensing of supernovae and the composition of dark matter. In *American Astronomical Society Meeting*, volume 193, pages 11707+, December 1998.

[61] D. L. Miller and D. Branch. Supernova absolute-magnitude distributions. *AJ*, 100:530–539, August 1990.

[62] J. Miralda-Escude. The correlation function of galaxy ellipticities produced by gravitational lensing. *ApJ*, 380:1–8, October 1991.

[63] B. Moore, S. Ghigna, F. Governato, G. Lake, T. Quinn, J. Stadel, and P. Tozzi. Dark matter substructure within galactic halos. *ApJ Letters*, 524:L19–L22, October 1999.

[64] B. Moore, F. Governato, T. Quinn, J. Stadel, and G. Lake. Resolving the structure of cold dark matter halos. *ApJ Letters*, 499:L5–+, May 1998.

[65] B. Moore, G. Lake, T. Quinn, and J. Stadel. On the survival and destruction of spiral galaxies in clusters. *MNRAS*, 304:465–474, April 1999.

[66] B. Moore, T. Quinn, F. Governato, J. Stadel, and G. Lake. Cold collapse and the core catastrophe. *MNRAS*, 310:1147–1152, December 1999.

[67] J. Mould and *et al.* . A search for weak distortion of distant galaxy images by large-scale structure. *MNRAS*, 271:31–38, November 1994.

[68] P. J. E. Peebles. *The large-scale structure of the universe.* Research supported by the National Science Foundation. Princeton, N.J., Princeton University Press, 1980. 435 p., 1980.

[69] S. Perlmutter and *et al.* . Measurements of the cosmological parameters omega and lambda from the first seven supernovae at z andqgt;= 0.35. *ApJ*, 483:565+, July 1997.

[70] S. Perlmutter and *et al.* . Discovery of a supernova explosion at half the age of the universe. *Nature*, 391:51+, January 1998.

[71] D. H. Porter. *A study of hierarchical clustering of galaxies in an expanding universe.* PhD thesis, California Univ., Berkeley., 1985.

[72] P. Premadi, H. Martel, and R. Matzner. Light propagation in inhomogeneous universes. i. methodology and preliminary results. *ApJ*, 493:10+, January 1998.

[73] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical recipes in C. The art of scientific computing.* Cambridge: University Press, —c1992, 2nd ed., 1992.

[74] D. C. Richardson, T. Quinn, J. Stadel, and G. Lake. Direct large-scale n-body simulations of planetesimal dynamics. *Icarus*, 143:45–59, January 2000.

[75] A. G. Riess and *et al.* . Observational evidence from supernovae for an accelerating universe and a cosmological constant. *AJ*, 116:1009–1038, September 1998.

[76] A. G. Riess, W. H. Press, and R. P. Kirshner. Is the dust obscuring supernovae in distant galaxies the same as dust in the milky way? *ApJ*, 473:588+, December 1996.

[77] P. Saha and S. Tremaine. Symplectic integrators for solar system dynamics. *AJ*, 104:1633–1640, October 1992.

[78] P. Saha and S. Tremaine. Long-term planetary integration with individual time steps. *AJ*, 108:1962–1969, November 1994.

[79] John K. Salmon and Michael S. Warren. Skeletons from the treecode closet. J. Comp. Phys., 111(1):136–155, 1994.

[80] John K. Salmon, Grégoire S. Winckelmans, and Michael S. Warren. Fast parallel treecodes for gravitational and fluid dynamical N-body problems. Intl. J. Supercomputer Appl., 8, 1994.

[81] A. Sandage and G. A. Tammann. Steps toward the hubble constant. viii - the global value. *ApJ*, 256:339–345, May 1982.

[82] B. P. Schmidt and *et al.* . The high-z supernova search: Measuring cosmic deceleration and global curvature of the universe using type ia supernovae. *ApJ*, 507:46–63, November 1998.

[83] P. Schneider, J. Ehlers, and E. E. Falco. *Gravitational Lenses*. Gravitational Lenses, XIV, 560 pp. 112 figs.. Springer-Verlag Berlin Heidelberg New York. Also Astronomy and Astrophysics Library, 1992.

[84] P. Schneider and *et al.* . Detection of shear due to weak lensing by large-scale structure. *AAP*, 333:767–778, May 1998.

[85] P. Schneider and R. V. Wagoner. Amplification and polarization of supernovae by gravitational lensing. *ApJ*, 314:154–175, March 1987.

[86] P. Schneider and A. Weiss. Light propagation in inhomogeneous universes - the ray-shooting method. *ApJ*, 330:1–15, July 1988.

[87] J. P. Singh. *Parallel Hierarchical N-Body Methods and their Implications for Multiprocessors*. PhD thesis, Stanford University, March 1993.

[88] R. D. Skeel and J. J. Biesiadecki. Ann. Numer. Math., 1:191+, 1994.

[89] R. D. Skeel and C. W. Gear. Physica D, 60:311+, 1992.

[90] C. C. Steidel, K. L. Adelberger, M. Dickinson, M. Giavalisco, M. Pettini, and M. Kellogg. A large structure of galaxies at redshift z approximately 3 and its cosmological implications. *ApJ*, 492:428+, January 1998.

[91] C. C. Steidel, M. Giavalisco, M. Pettini, M. Dickinson, and K. L. Adelberger. Spectroscopic confirmation of a population of normal star-forming galaxies at redshifts z andgt; 3. *ApJ Letters*, 462:L17–+, May 1996.

[92] I.;̃. Szapudi, T. Quinn, J. Stadel, and G. Lake. N-point correlations in cdm and omegacdm simulations. *ApJ*, 517:54–63, May 1999.

[93] K. Tomita. Cosmological simulation of gravitational lensing and angular diameter distances. In *Abstracts of the 19th Texas Symposium on Relativistic Astrophysics and Cosmology, held in Paris, France, Dec. 14-18, 1998. Eds.: J. Paul, T. Montmerle, and E. Aubourg (CEA Saclay).*, pages E310–+, December 1998.

[94] F. Valdes, J. F. Jarvis, and J. A. Tyson. Alignment of faint galaxy images - cosmological distortion and rotation. *ApJ*, 271:431–441, August 1983.

[95] F. C. van den Bosch, G. F. Lewis, G. Lake, and J. Stadel. Substructure in dark halos: Orbital eccentricities and dynamical friction. *ApJ*, 515:50–68, April 1999.

[96] D. Walsh, R. F. Carswell, and R. J. Weymann. 0957 + 561 a, b - twin quasistellar objects or gravitational lens. *Nature*, 279:381–384, May 1979.

[97] J. Wambsganss, R. Cen, and J. P. Ostriker. Testing cosmological models by gravitational lensing. i. method and first applications. *ApJ*, 494:29+, February 1998.

[98] J. Wambsganss, R. Cen, J. P. Ostriker, and E. L. Turner. Testing cosmogonic models with gravitational lensing. *Science*, 268:274+, April 1995.

[99] J. Wambsganss, R. Cen, G. Xu, and J. P. Ostriker. Effects of weak gravitational lensing from large-scale structure of the determination of q 0. *ApJ Letters*, 475:L81–+, February 1997.

[100] Michael S. Warren and John K. Salmon. A parallel, portable and versatile treecode. In *Seventh SIAM Conference on Parallel Processing for Scientific Computing*. SIAM, 1995.

[101] M. White and W. Hu. A new algorithm for computing statistics of weak lensing by large-scale structure. *ApJ*, 537:1–11, July 2000.

[102] J. Wisdom and M. Holman. Symplectic maps for the n-body problem. *AJ*, 102:1528–1538, October 1991.

[103] C. Yess and S. F. Shandarin. Universality of the network and bubble topology in cosmological gravitational simulations. *ApJ*, 465:2+, July 1996.

[104] H. Yoshida. Phys. Lett. A., 150:262+, 1990.

[105] M. Q. Zhang and R. D. Skeel. J. Comp. Chem., 16:365+, 1995.

# Curriculum Vitae - Joachim Gerhard Stadel

University of Victoria
Dept. of Physics and Astronomy
Elliot Bldg, 3800 Finnerty Rd
Victoria, BC Canada V8P 1A1

Work: (250) 721-8656
Home: (250) 381-2076
Fax: (250) 721-7715
stadel@astro.washington.edu

**Research astrophysicist with extensive experience in numerical computing and parallel algorithms applied to astrophysical problems ranging from the evolution of cosmic structure to long-term solar system stability. While working under the NASA HPCC initiative, developed the parallel N-body gravity code, PKDGRAV, which has been used by several research groups around the world and has provided new predicitions on the clustering properties of dark matter in the Universe. Is involved at present in a larger effort devoted to understanding galaxy formation in detail using GASOLINE, the hydrodynamics extension of PKDGRAV.**

Born April 4, 1968 in Hamilton Ontario Canada of German parents, thus having dual Canadian and German citizenship and fluency in both English and German.

**Postdoctoral Position: (2000 - present) University of Victoria**

**Graduate Studies: (1995 - 2000)**
**University of Washington**
**PhD in Astronomy, March 2001**
***Cosmological N-Body Simulations and their Analysis***

**Graduate Studies: (1991 - 1993)**
**University of Toronto**
**Physics and Astronomy**

**Undergraduate Studies: (1986 - 1989, 1990 - 1991)**
**Brandon University**
**4-year BSc in Physics and Mathematics with Honors**

**High School: (1983 - 1986)**
**Vincent Massey High School, Brandon Manitoba**

## Academic Awards

- **2000: CITA National Fellowship (Canadian Institute for Theoretical Astrophysics)**
- **2000: PIMS Postdoctoral Fellowship (Pacific Institute for Mathematical Sciences)**
- **1991: NSERC 1967 Graduate Scholarship (50 Awarded in Canada, value $21,000/yr)**
- **1990: *Brandon University Silver Medal* for highest overall achievement in Science.**
- **1991: *Arthur & Abbie Vining* Memorial Graduate Scholarship in Physics**
- **1988: *Robert James Lang* Memorial Scholarship in Mathematics**
- **1988: *Digital Equipment of Canada Limitted* Award of Merit in Computer Science**
- **1988: *W. G. Wong* Scholarship in Physics**
- **1987: *C. B. Clark* Memorial Scholarship in Physics**

## Teaching Experience

- **1999: Teaching Assistant for a graduate course: Cosmology, and for a second year undergraduate course: Gravity in the Universe.**
- **1998: Development of several interactive WWW labs for introductory astronomy courses.**
- **1995 - 1998: Teaching Assistant for introductory astronomy courses.**
- **1988 - 1989: Teaching Assistant for introductory physics courses.**

## Computer Experience

- **High level of proficiency in modern programming languages: C, C++, FORTRAN, Java**
- **Author of the PKDGRAV code, used for running large N-body simulations on the present generation of parallel supercomputers. This code is considered state-of-the-art and is used by several research groups around the world.**
- **Developed a suite of applications for the execution and analysis of large cosmological N-body simulations. This suite of tools is made publically available to other researchers.**
- **Familiarity with astronomical data reduction using IRAF.**

## Research Work Experience

- **1993 - 1995: Research Astronomer for the NASA HPCC project at the University of Washington. Studied the formation of Large Scale Structure using parallel supercomputer simulations.**
- **1991: Sunmer NSERC Research Grant, Brandon University. Research in Mathematics on Lattice Theory and Partially Ordered Sets (see publication).**
- **1990: Max-Planck-Institute fuer extraterrestrische Physik (MPE), Munich Germany. Created a simulation code used in the design of space-based particle detectors.**
- **1988: Summer NSERC Research Grant, Brandon University. Research on spectral line identification of stars (see publication).**

## Ongoing Research Projects and Interests

- **GASOLINE: development of a new parallel SPH hydrodynamics code for cosmological structure formation simulations including gas dynamics, gravity, and radiative cooling.**

- **Solar system chaos and stability: integration of the planetary orbits over a span of 10 billion years, including the effects of general relativity.**
- **Solar system formation: direct simulation of planetesimal dynamics.**
- **Large scale structure formation: dynamics of cluster formation, artificial galaxy catalogs.**
- **Using planetary nebulae as tracers of intercluster stars and dynamics of clusters.**

## Papers in Refereed Journals


1. **The Descendants of Lyman Break Galaxies in Galaxy Clusters: Spatial Distribution and Orbital Properties**
   **F. Governato, S. Ghigna, B. Moore, T. Quinn, J. Stadel, G. Lake, ApJ 547:555, 2001**
2. **Tidal Stirring and the Origin of the Dwarf Spheroidals in the Local Group**
   **L. Mayer, F. Governato, M. Colpi, B. Moore, T. Quinn, J. Wadsley, J. Stadel, G. Lake, ApJ 547:123, 2001**
3. **Density Profiles and Substructure of Dark Matter Halos: Converging Results at Ultra-High Numerical Resolution**
   **S. Ghigna, B. Moore, F. Governato, G. Lake, T. Quinn, J. Stadel, ApJ 544:616, 2000**
4. **Cold Collapse and the Core Catastrophe**
   **B. Moore, T. Quinn, F. Governato, J. Stadel, G. Lake, MNRAS 310:1147, 1999**
5. **Time Stepping N-Body Simulations**
   **T. Quinn, N. Katz, J. Stadel, G. Lake, astro-ph/9710043, submitted to ApJ**
6. **Dark Matter Substructure within Galactic Halos**
   **B. Moore, S. Ghigna, F. Governato, G. Lake, T. Quinn, J. Stadel, P. Tozzi, ApJ 524:L19, 1999**
7. **Direct Large-Scale N-Body Simulations of Planetesimal Dynamics**
   **D. C. Richardson, T. Quinn, J. Stadel, G. Lake, *Icarus* in press.**
8. **N-Point Correlations in CDM and OCDM Simulations**
   **I. Szapudi, T. Quinn, J. Stadel, G. Lake, ApJ 517:54, 1999**
9. **On the Survival and Destruction of Spiral Galaxies in Clusters**
   **B. Moore, G. Lake, T. Quinn, J. Stadel, MNRAS 304:465, 1999**
10. **Structure in Dark Halos: Orbital Eccentricities and Dynamical Friction**
    **F. C. van den Bosch, G. F. Lewis, G. Lake, J. Stadel, ApJ 515:50, 1999**
11. **On the Origin of Early Type Galaxies and the Evolution of the Interaction Rate in the Field**
    **F. Governato, J. Gardner, J. Stadel, T. Quinn, G. Lake, AJ 117:1651, 1999**
12. **Dark Matter Haloes within Clusters**
    **S. Ghigna, B. Moore, F. Governato, G. Lake, T. Quinn, J.Stadel, MNRAS 300:146, 1998**
13. **The Seeds of Rich Galaxy Clusters in the Universe**
    **F. Governato, C. Baugh, C. S. Frenk, S. Cole, C. Lacey, T. Quinn, J. Stadel, Nature 392:359, 1998**
14. **Resolving the Structure of Cold Dark Matter Halos**
    **B. Moore, F. Governato, T. Quinn, J. Stadel, G. Lake, ApJ Letters 499:L5, 1998**
15. **A Parallel Integration Method for Solar System Dynamics**
    **P. Saha, J. Stadel, S. Tremaine, AJ 114:409, 1997**
16. **The Local Group as a Test of Cosmological Models**
    **F. Governato, B. Moore, R. Cen, J. Stadel, G. Lake, T. Quinn, NewA 2:91, 1997**
17. **Prospects for Cosmology with Cluster Mass Profiles**

**M. Crone, F. Governato, J. Stadel, T. Quinn, ApJ Letters 477:L5, 1997**
18. **The Strong Fixed Point Property for Small Sets**
**D. Pickering, M. Roddy, J. Stadel, Order 8:29, 1991**
19. **Automated Spectral Line Identification**
**A. F. Gulliver and J. Stadel, PASP 102:587, 1990**

## Papers in Conference Proceedings

1. **The Structure of Cold Dark Matter Halos**
**B. Moore, S. Ghigna, F. Governato, G. Lake, T. Quinn, J. Stadel,**
**Large Scale Structure: Tracks and Traces, Proceedings of the 12th Potsdam Cosmology Workshop, 1997**
2. **Parallel Gravity from the 9 Planet Problem to Billions and Billions**
**G. Lake, T. Quinn, D. C. Richardson, J. Stadel, High Performance Computing 1997: Grand Challenges in Computer Simulation -- Proceedings of the 1997 Simulation MultiConference, p.86.**
3. **A Performance Study of Cosmological Simulations on Message-Passing and Shared Memory Multiprocessors**
**M. Dikaiakos and J. Stadel, ICS Conference Proceedings 1996 (Refereed Publication)**
4. **Cosmological N-Body Simulation**
**G. Lake, N. Katz, T. Quinn, J. Stadel, Parallel Processing for Scientific Computing, SIAM 1995, p.307.**

## References

Thomas Quinn
University of Washington
Department of Astronomy
Box 351580
Seattle, WA 98195-1580
trq@astro.washington.edu
Phone: (206) 685-9009

George Lake
Institute for Systems Biology
4225 Roosevelt Way NE
Suite 200
Seattle, WA 98105-6099
glake@systemsbiology.org
Phone: (206) 732-1225

Arif Babul
University of Victoria
Dept. of Physics & Astronomy
Elliot Bldg, 3800 Finnerty Rd
Victoria, BC Canada V8P 1A1
babul@alMuhit.phys.UVic.CA
Phone: (250) 721-8844

## Others Familiar with my Research

Craig Hogan
University of Washington
Department of Astronomy
Box 351580
Seattle, WA 98195-1580
hogan@astro.washington.edu
Phone: (206) 685-2112

Carlos Frenk
University of Durham
Department of Physics,
Science Laboratories
South Road,
Durham DH1 3LE, UK
c.s.frenk@dur.ac.uk
Fax: (01144) 191 374-3749

Ben Moore
University of Durham
Department of Physics,
Science Laboratories
South Road,
Durham DH1 3LE, UK
moore@dur.ac.uk
Phone: (01144) 191 374-4613