Contents

Acknowledgement

At the outset, I would like to thank my parents for all that they have smilingly endured and sacrificed for my sake since my childhood. It is obviously plain to me that it would have been well nigh impossible for me to go the distance without their strength and support which I always seem to have taken for granted. I would also like to acknowledge, with all due sincerity, the tremendous amount of selfless effort put in by my thesis supervisor Jasjeet. I had definitely not made the task any easier for him and I hope that in the near future I would be able to take advantage of the strong launching pad that Jasjeet has strived to create for me today.

I would like to thank Sriram and Sesh for being a constant source of encouragement and support to me. I am also indebted to Biswarupda whose has been a reassuring presence for me — someone whose advice and counsel I could always rely on. So many of the academic courses offered at HRI over the years have been enriching experiences for me — I take back with me invaluable lessons from Ashokeda, Pinakida, Biswarupda, Rajesh, Sumathi, Professor Mani, Sesh, Shiv and Jasjeet. I would like to recall the encouragement offered by Sumathi during the early days of graduate school when things were rather uphill for many of us.

Outside of home, I have spent the greatest part of my life at HRI. This is the most beautiful place I have ever lived in. I have made invaluable friends at HRI — bonds that would hopefully last a lifetime. I am deeply indebted to my batchmates Sourin, Partha and Swapan for being always there when I needed them. I would like to acknowldge the support and sympathy of Purusottam, Sanoli and Siddhartha through all these years.

I would not have been able to tide over many a moment of personal crisis without my friends. Poonam has been a wonderful sister to me. She has been a perennial source of strength and I can never thank her enough for all that she's done for me.

I have enjoyed countless moments of joy and camaraderie with Sanjeev, Debajyotida, Kalyanda, Soumenda, Anjanda, Yashpal, Bhanu and Pintu on the football ground. I would like to fondly remember the time I have spent in the pleasant company of Parthada, Sharmishthadi, Sourovda, Anindyada, Debrupadi, Somadi, Shripad, Tribikramda, Aditidi, Roopam, Biswarupda, Ashokeda, Sumathi, Maneesh, Aarti, Kalyanda, Rashmi, Riyan and Arnabda. I would like to acknowledge Seema for being the radiant ever-smiling face of HRI administration. I would also like to thank Anju and Farooqui for attending to all my computational needs with patience and admirable efficiency over the years. I have really enjoyed my stay at HRI as a graduate student and it is a privilege to express my gratitude towards all members of the HRI community who made this experience a memorable one for me.

Numerical experiments for this thesis were carried out at the cluster computing facility at the Harish-Chandra Research Institute (http://cluster.mri.ernet.in). This research has made use of NASA's Astrophysics Data System.

Abstract

The primary goal of this thesis is to study aspects of non-linear gravitational clustering in an expanding Universe using N-Body simulations. The thesis can, in principle, be divided into two distinct parts. The first part deals with the development and study of performance characteristics of the TreePM method for cosmological simulations with emphasis on optimisation, improvement and parallelisation of the method. Because N-Body simulations are a very important tool in the study of formation of large scale structures, we also try to address, from a general broader perspective, some of the limitations of N-Body methods in this part of the thesis. The remainder of the thesis primarily focusses on using the TreePM code to study a few problems in gravitational clustering. This includes a study of aspects of clustering of overdense regions in power law models as well as an analysis of clustering and non-linear scaling relations in a scenario of two-dimensional gravitational collapse.

The thesis is organised as follows. Chapter 1 summarises the current status of some important cosmological observations and reviews some of the more general analytical formalisms (big bang cosmology, perturbation theory, etc) which form the backbone of the thesis. It also defines and sets the notation for various physical and statistical quantities of interest (correlation functions, bias parameter, etc.) in the thesis. Chapter 2 gives an overview of cosmological N-Body simulations, describing, in general terms, the various modules that make up a cosmological N-Body code. The chapter also reveiws some of the important algorithms for cosmolological N-Body simulations. Each of the remaining chapters of the thesis addresses a separate problem, a chapterwise summary of which

follows.

The TreePM code is a hybrid technique for carrying out large N-Body simulations to study the formation and evolution of large scale structures in the Universe. It is a combination of the Barnes and Hut (1986) tree code and a Particle-Mesh (PM) code. In Chapter 3, we study the performance characteristics of the TreePM code. We also perform a comprehensive analysis of the error budget for the code. It is shown that the choice of filter for splitting the inverse square force into short and long range components suggested in Bagla (2002) is close to optimum - it turns out to be the best among the filters we have studied. We show that the error in the long range component of the force contributes very little to the total error in force. Errors introduced by the tree approximation for the short range force are different from those for the inverse square force and these errors dominate the total error in force. We calculate the distribution function for error in force for clustered and unclustered particle distributions. This gives an idea of the error in realistic situations for different choices of the parameters of the TreePM code. Errors for 99% of particles are found to be below 1.5% for an unclustered distribution and below 0.8% for a clustered distribution for the recommended configuration of the TreePM code. These numbers compare favourably with other methods such as implementations of the tree code like GADGET (Springel, Yoshida and White, 2001) and hybrid methods such as the TPM (Xu, 1995). We also test the code by simulating a few power law models and checking for scale invariance.

The emergence of Beowulf clusters as an affordable platform for high performance computing has given an impetus to the development of algorithms that can be parallelised easily on such platforms. In Chapter 4, we describe a method for parallelising the TreePM code. The TreePM method has two convenient parallelisms inherent in the sequential algorithm. In all N-Body codes, including the TreePM code, the force on all particles can be computed concurrently. Over and above that, the TreePM method splits force computation into two parts, that of the long range and the short range forces. Therefore, the long range force can be computed concurrently on a processor not involved in the computation of the short range force. This makes the TreePM algorithm comparatively simple to parallelise. We use both functional and domain decompositions for parallelising the TreePM code. Functional decomposition is used to separate the computation of the long range and short range forces as well as for the task of coordinating communications between the different components. Short range force calculation is time-consuming and benefits from the use of domain decomposition. We test the parallel code on a Linux cluster. We get a speedup of 31.4 for a 128^3 particle simulation on 33 processors; speedup being better for larger simulations. The time taken for one timestep per particle is found to be 6.5μs for a 256^3 particle simulation on 65 processors. Thus a simulation that runs for 4000 timesteps takes 5 days on the Linux cluster.

Many problems in gravitational clustering are intractable because of the limited dynamic range of N-Body simulations in three dimensions. The problem of dynamic range can be circumvented by simulating a two-dimensional system as long as one studies generic features like the scaling relations which are likely to be independent of dimension. We have extended the TreePM method to simulations in two dimensions. Chapter 5 describes the 2d TreePM code. In two dimensions, the gravitational force falls as $1/r$ and not as $1/r^2$. The splitting of force between the PM and the tree parts is described. Though the key features of the error analysis for the 3d TreePM code carry over to the 2d code as well, we also estimate the error in the 2d force for a realistic configuration. Error for 99% of the particles is found to be below 4% for an unclustered distribution and below 2% for a clustered distribution. A comparison of the 2d TreePM and Particle-Mesh methods is also carried out in this chapter of the thesis. The aim is to highlight the efficacy of the 2d TreePM method for two-dimensional cosmological simulations. We show that a PM simulation underestimates the clustering at small scales and that haloes are better resolved in TreePM simulations. The 2d TreePM code offers greater dynamic range and superior resolution in comparison to a 2d Particle-Mesh code.

Much of the work in this thesis is based on the development of methods for N-Body simulations as well as analysis of results obtained from N-Body simulations. Also much of the progress in understanding the physics of the high-redshift Universe and comparisons with observations would not have been possible without N-Body simulations. Given the importance of this tool, it is essential to understand its limitations, as ignoring them can easily lead to interesting, but wrong results. In Chapter 6, we analyse some of the limitations arising out of the finite size of the simulation volume in any N-Body simulation. A finite size implies that fluctuations at scales larger than the size of the simulation volume are ignored and a truncated power spectrum is simulated. We propose a measure based on mass functions that can be used to test whether the simulation volume is large enough for us to ignore contribution of scales larger than the dimensions of the simulation volume. If the simulation volume is large enough, then the number density of massive objects expected from the full power spectrum and that from the truncated power spectrum should match. The same scale is also an appropriate choice for the transition scale when tools like *MAP* (Tormen and Bertschinger, 1996), that add the contribution of the missing power, are used.

Scaling relations in gravitational clustering in an expanding background indicate that there are three prominent regimes in the evolution of gravitational clustering. The highly non-linear asymptotic regime is poorly understood. Several studies have attempted to understand the nature of the asymptotic regime, mainly with help of N-Body simulations. One of the reasons for the inability to resolve the confronting issues in the asymptotic regime has been the limited dynamic range of N-Body simulations. We can circumvent the problem of dynamic range by simulating a two-dimensional system. Understanding the nature of the asymptotic regime in two dimensions can help solve the problem in three dimensions, even though it may not be possible to map the solution directly to the full problem in three dimensions. In Chapter 7, we investigate scaling relations for two-dimensional gravitational collapse using the 2d TreePM code and study the strongly non-linear regime ($\bar{\xi} \simeq 100$) for power law models. We find the evolution of these models to be scale-invariant in all our simulations. We find that the stable clustering limit is not reached, but there is a model-independent non-linear scaling relation in the asymptotic regime. This confirms results from an earlier study (Bagla, Engineer and Padmanabhan,

1998) which only probed the mildly non-linear regime ($\bar{\xi} < 40$). The correlation function in the extremely non-linear regime is found to be a less steep function of scale than reported in earlier studies. We show that this is due to coherent transverse motions in massive haloes. We also study density profiles and find that the scatter in the inner and outer slopes is large and there is no single universal profile that fits all cases. Also the difference in typical density profiles for different models is found to be smaller than expected from the stable clustering hypothesis. Transverse motions induced by substructure are a likely reason for this difference being small.

Current models for the formation of large scale structures in the Universe are based on the assumption that gravitational amplification of density perturbations results in the formation of collapsed, virialised objects. These models can be used to make predictions about the distribution of matter. This is conveniently codified in terms of correlation functions and the moments of the matter distribution. However, all of these models deal with the distribution of dark matter, whereas what we see are galaxies. Bias is the thread that links the two. We define bias in a statistical manner. It is the ratio of the power spectrum for galaxies and dark matter. Understanding clustering of galaxies in the non-linear regime is complicated by the issue of bias. It has been hypothesised that measurements of higher order correlations and moments allow a determination of bias. However, in the absence of a clean analytical formulation in the non-linear regime, there is no clear method of making use of higher order moments for this purpose. In Chapter 8 of the thesis we try to get a handle on the relation of galaxy clustering and the initial power spectrum of density fluctuations by computing the moments of distribution of overdense regions in N-Body simulations of power law models using the TreePM code. In the study, high density regions are identified and isolated. These are assumed to host galaxies in proportion to mass. We study the effect of redshift space distortions and high density cutoffs on mass distributions and we find that the skewness for overdense regions in redshift space *does not depend on the initial power spectrum* in any significant manner. This is work in progress.

The thesis is based on the following :

- (1) J.S. Bagla and **S. Ray** : Performance Characteristics of TreePM Codes, New Astronomy **8**, 665, 2003.
- (2) **S. Ray** and J.S. Bagla : A Parallel Implementation of the TreePM Code, astro-ph/0405220.
- (3) **S. Ray** : TreePM Method for Two-Dimensional Cosmological Simulations, astro-ph/0406009, accepted for publication in JAA.
- (4) **S. Ray**, J.S. Bagla and T. Padmanabhan : Gravitational Collapse in an Expanding Universe : Scaling Relations for Two-Dimensional Collapse Revisited, astro-ph/0410041, accepted for publication in MNRAS, pending minor revisions.
- (5) J.S. Bagla and **S. Ray** : Comments on the Size of the Simulation Box in Cosmological N-Body Simulations, astro-ph/0410373, accepted for publication in MNRAS.
- (6) J.S. Bagla and **S. Ray** : Patterns in Clustering of Overdense Regions, manuscript in preparation.

Other Publications / Preprints / Work in Progress :

- (1) J.S. Bagla, J. Prasad and **S. Ray** : Gravitational Collapse in an Expanding Background and the Role of Substructure I : Planar Collapse, astro-ph/0408429, accepted for publication in MNRAS.
- (2) J.S. Bagla and **S. Ray** : The Adaptive TreePM Method (ATreePM) for Cosmological Simulations, work in progress.

Chapter 1

Introduction

Cosmology is the scientific study of properties of the Universe as a whole. The aim of cosmology is to understand the origin, evolution and ultimate fate of the Universe. The prevailing theory about the origin and evolution of the Universe is the so-called big bang model which postulates that the Universe has expanded from a hot dense state into the vast and much cooler cosmos we currently inhabit. The big bang model of cosmology rests on two key ideas that date back to the early 20th century : Einstein's general theory of relativity and the cosmological principle. The cosmological principle states that the Universe is homogeneous and isotropic. We assume that the matter in the Universe is distributed uniformly on large scales and general relativity is used to compute the evolution of the Universe.

1.1 Cosmology and Observations

Observations reveal that the Universe is homogeneous at very large scales. It has been shown (Bharadwaj, Mittal and Seshadri, 1999) that the distribution of galaxies in the Las Campanas Redshift Survey (Shectman et al., 1996) is consistent with the Universe being homogeneous at large scales and the transition to homogeneity occurs in the range of scales $80h^{-1}$ Mpc to $100h^{-1}$ Mpc [1]. That the early Universe was highly homogeneous and isotropic is also indicated by observations of temperature anisotropies with a very small amplitude $[\frac{\Delta T}{T} \sim 10^{-5}]$ in the cosmic microwave background radiation (Smoot et al., 1992; Bennett et al., 2003).

The first important discovery in observational cosmology was that we live in an expanding Universe. Hubble discovered (Hubble, 1929) from observations of galaxies that these are moving away from us with a speed proportional to their distance from us. This is called the Hubble law and the constant of proportionality is called the Hubble constant H_0. The value of H_0 is equal to 72^{+5}_{-5} (Km/sec)/Mpc for a 68% confidence range (Freedman et al., 2001; Spergel et al., 2003; Bennett et al., 2003). The form of the Hubble law is also consistent with a homogeneous and isotropic Universe. WMAP and other observations (Perlmutter et al., 1999; Spergel et al., 2003; Bennett et al., 2003) indicate that the total energy density in the Universe is close to the critical density (defined later) and hence the geometry of the Universe is flat. There is considerable evidence that the energy content of the Universe is dominated by *dark* matter and *dark* energy, components distinct from normal matter (baryons) and radiation.

The dynamical mass of galaxies is found to be about ten times larger than the mass associated with stars, gas and dust. This excess mass is also inferred from observations of gravitational lensing (Schneider, Ehlers and Falco, 1992; Chitre et al., 2001; Van Waerbeke et al., 2000), the phenomenon of bending of light predicted by the general theory of relativity. The material that exerts a gravitational pull, but does not emit nor absorb light is called dark matter (Trimble, 1987). The exact nature of dark matter is unknown, but there are a number of plausible candidates. Dark matter could be brown dwarfs or supermassive black holes; it could also be made of weakly interacting particles. Cosmologists call these hypothetical particles WIMPs (Weakly Interacting Massive Particles). There are strong limits on the interaction cross-sections of WIMPs from astrophysical considerations (Kochanek and White, 2000; Hennawi and Ostriker, 2002). Observational

[1] 1 Mpc is 1 Megaparsec and is equal to 3.0856×10^{22} metres

evidence indicates that dark matter is non-relativistic (Spergel et al., 2003; Tegmark et al., 2003). Non-relativistic, non-interacting (collisionless) dark matter is called *cold* dark matter. Observations indicate that about a quarter of the matter content of the Universe is cold dark matter.

Measurements based on the luminosity distance to distant Type Ia supernovae (Perlmutter et al., 1999) suggest that the expansion of the Universe is accelerating which, in turn, implies the existence of a form of matter with a strong negative pressure. This form of matter is called dark energy. A prime candidate for this is the cosmological constant (Λ). Dark energy constitutes about two-thirds of the energy content of the Universe and it plays a significant role in the evolution of the Universe.

Though the Universe is homogeneous and isotropic at the largest scales probed, observations reveal a wealth of detail on smaller scales. There are clear departures from homogeneity from the scale of single galaxies to that of structures larger than 100 Mpc. Fig.1.1 shows the distribution of galaxies observed by the 2dF Galaxy Redshift Survey (http://www.mso.anu.edu.au/2dFGRS). The observed field is covered by a complex network of structures such as filaments, voids, clusters and superclusters (Peacock et al., 2001).

Large scale structures like galaxies and clusters of galaxies are believed to have formed by the amplification of small perturbations due to gravitational instability (Doroshkevich and Zeldovich, 1964; Dicke et al., 1965; Zeldovich and Podurets, 1966; Doroshkevich, Zeldovich and Novikov, 1967; Zeldovich and Novikov, 1967; Peebles, 1980; Peacock, 1998; Padmanabhan, 2002a; Bernardeau et al., 2002). Fluctuations detected in the cosmic microwave background temperature are signatures of such primordial density perturbations in the early Universe.

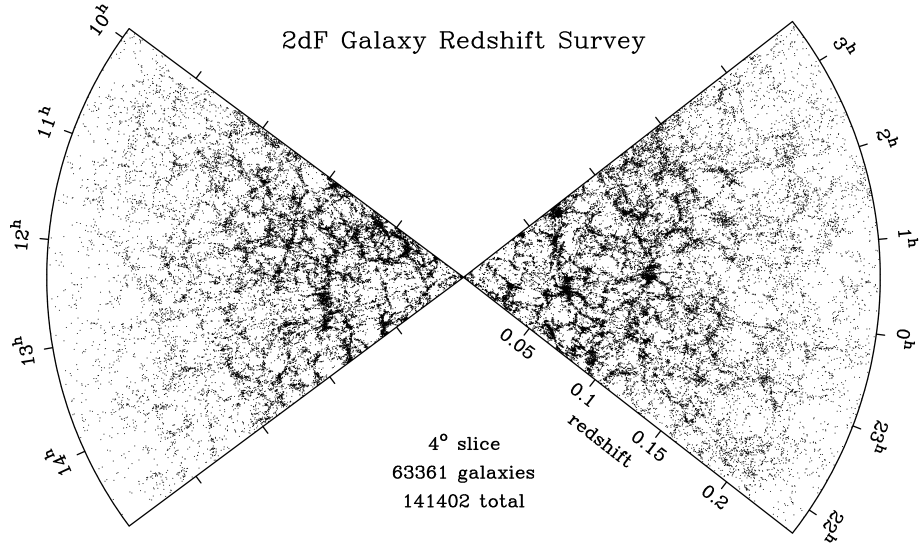

Figure 1.1: This figure shows the distribution of galaxies observed by the 2dF Galaxy Redshift Survey (http://www.mso.anu.edu.au/2dFGRS).

1.2 The Background Universe

In this section, we will briefly review the relativistic formalism for a homogeneous and isotropic background Universe in the standard big bang model of cosmology (Peebles, 1993; Padmanabhan, 1993; Narlikar, 1993; Peacock, 1998). For a homogeneous and isotropic Universe, many of the features of the metric can be deduced from symmetry alone (Robertson, 1935; Walker, 1936). The Robertson-Walker metric describes a homogeneous and isotropic space evolving by a simple scaling of coordinates with fundamental observers maintaining fixed positions in comoving coordinates $\mathbf{x}$. The metric has the form

$$ds^2 = c^2dt^2 - a^2(t)\left[\frac{dx^2}{1-kx^2} + x^2(\sin^2\theta d\phi^2 + d\theta^2)\right] \tag{1.1}$$

where $a(t)$ is the scale factor and k is the curvature constant. It can be shown that k takes values $+1$, 0, or -1 and the Universe is called closed, flat or open respectively. k can be restricted to these values only if the present value of a is not assumed to be unity for the open and closed models. t is the cosmic time.

We can relate comoving co-ordinates $\mathbf{x}$ to the observationally measured coordinates

$\mathbf{r}$, also called the proper coordinates : $\mathbf{r}(t) = a(t)\mathbf{x}$. Then,

$$\dot{\mathbf{r}}(t) = \left[\frac{\dot{a}(t)}{a(t)}\right] a(t)\mathbf{x} = \left[\frac{\dot{a}(t)}{a(t)}\right] \mathbf{r}(t) = H(t)\mathbf{r}(t) \tag{1.2}$$

implying that $H(t)$ is the Hubble parameter. Its present value $H_0 = \left(\frac{\dot{a}}{a}\right)_{t=t_0}$ is the Hubble constant. For an increasing $a(t)$ we get an expanding Universe.

Einstein's field equations are :

$$R_{\mu\nu} - \frac{1}{2}g_{\mu\nu}R - \Lambda g_{\mu\nu} = -8\pi G T_{\mu\nu} \tag{1.3}$$

where $R_{\mu\nu}$, $g_{\mu\nu}$ and R are respectively the Ricci tensor, the metric tensor and the curvature scalar. $T_{\mu\nu}$ is the energy-momentum tensor, Λ the cosmological constant and G the gravitational constant. Under the assumptions of homogeneity and isotropy, the tensor $T_{\mu\nu}$ has the following form for perfect fluids :

$$T_{\mu\nu} = dia[\bar{\rho}(t), -p(t), -p(t), -p(t)] \tag{1.4}$$

Here p is the pressure. On computing the Ricci tensor and the curvature scalar from the Robertson-Walker metric, eqn.1.3 reduces to two independent equations $(c = 1)$:

$$\frac{\dot{a}^2 + k}{a^2} = \frac{8\pi G}{3}\bar{\rho} + \frac{\Lambda}{3} \tag{1.5}$$

$$2\frac{\ddot{a}}{a} + \frac{\dot{a}^2 + k}{a^2} = -8\pi G p + \Lambda \tag{1.6}$$

These two equations, combined with an equation of state $p = p(\bar{\rho})$, completely determine the three functions $a(t)$, $p(t)$ and $\bar{\rho}(t)$. The initial conditions can be specified in terms of the densities contributed by the various forms of energy and matter in the Universe at the present time. The densities at $t = t_0$ are conveniently expressed in dimensionless form by

$$\Omega_i = \frac{\bar{\rho}_i(t_0)}{\rho_c} \tag{1.7}$$

where the density contributed by the ith component of matter or energy is scaled by the critical density ρ_c :

$$\rho_c = \frac{{H_0}^2}{8\pi G} \tag{1.8}$$

From eqn.1.5, we can see that

$$\bar{\rho}a^3 = \left(\frac{3a}{8\pi G}\right)\left(\dot{a}^2 + k - \frac{\Lambda a^2}{3}\right) \tag{1.9}$$

Differentiating this expresssion and using equation 1.6, we obtain the following :

$$\frac{d}{da}(\bar{\rho}a^3) = -3a^2 p \tag{1.10}$$

This equation is the same as $T^{ij}_{;j} = 0$, the conservation equation. Given an equation of state $p = p(\bar{\rho})$, we can integrate eqn.1.10 to obtain $\bar{\rho} = \bar{\rho}(a)$. For an equation of state of the form $p = w\bar{\rho}$, density varies as

$$\bar{\rho} \propto a^{-3(1+w)} \tag{1.11}$$

Radiation and relativistic matter (Ω_r) have an equation of state $p = \frac{1}{3}\bar{\rho}$, non-relativistic matter (Ω_{nr}) an equation of state $p \cong 0$ and the cosmological constant $p = -\bar{\rho}$. Using eqn.1.11, we have $\bar{\rho}_{nr} \propto a^{-3}$, $\bar{\rho}_{rad} \propto a^{-4}$ and $\bar{\rho}_\Lambda = \textit{constant}$, independent of a. Substituting any of these relations into eqn.1.5, we can determine $a(t)$. The energy density contributed by radiation and relativistic matter in our Universe is negligible at present, though these were the dominant components in the early Universe. It can be shown that the evolution of a Universe dominated by non-relativistic matter, Λ and k is governed by the following equation ($\Omega_r \simeq 0$) :

$$\frac{\dot{a}^2 + k}{a^2} = {H_0}^2\left[\Omega_{nr}\left(\frac{a_0}{a}\right)^3 + \Omega_\Lambda\right] \tag{1.12}$$

where $\Omega_\Lambda = \Lambda/3{H_0}^2$ and we have neglected all relativistic matter as well as radiation. Observations are consistent with a flat Universe ($k = 0$) with $\Omega_{tot} = 1.02 \pm 0.02$ (Spergel et al., 2003).

1.2.1 The Einstein deSitter Model

Let us consider a flat ($k = 0$) Universe containing only non-relativistic matter ($\Omega_{nr} = 1$); this is the Einstein deSitter Universe. Using $p = 0$ in eqn.1.10, we have

$$\frac{d}{da}(\bar{\rho}a^3) = 0 \Rightarrow \bar{\rho}(t) = \bar{\rho}(t_0)\left[\frac{a(t_0)}{a(t)}\right]^3 \tag{1.13}$$

Substituting the above expression for $\bar{\rho}$ in eqn.1.5, we get

$$\frac{\dot{a}^2}{a^2} = \frac{8\pi G}{3}\bar{\rho}(t_0)\left[\frac{a(t_0)}{a(t)}\right]^3 \tag{1.14}$$

Solving this with the initial condition $a(0) = 0$, we find

$$a(t) = [6\pi G\bar{\rho}(t_0)]^{1/3}a(t_0)t^{2/3} \tag{1.15}$$

Assuming $a(t_0) = 1$, we get

$$6\pi G\bar{\rho}(t_0) = t_0^{-2} \Rightarrow a(t) = \left(\frac{t}{t_0}\right)^{2/3} \tag{1.16}$$

The Einstein deSitter Universe expands forever with a decelerating rate of expansion and the density decreases as

$$\bar{\rho}(t) = \frac{1}{6\pi G t^2} \tag{1.17}$$

1.2.2 The Newtonian Limit

In principle, the process of structure formation should be studied in a fully covariant manner within the general theory of relativity. But it is more convenient to work in the Newtonian limit. In the Newtonian limit, we can reduce an arbitrary metric to the following form (Landau and Lifshitz, 1975) :

$$ds^2 = \left(c^2 + 2\Phi_{av}\right)dt^2 - d\mathbf{r}^2 \tag{1.18}$$

with $\Phi_{av}/c^2 \ll 1$. Here Φ_{av} is the effective Newtonian gravitational potential. The space-space components of the metric of the same order as Φ_{av} can be dropped if particle velocities are non-relativistic, i.e. $v/c \ll 1$.

Consider the following coordinate transformations for the Robertson-Walker metric :

$$\begin{aligned}\mathbf{r} &= a(t)\mathbf{x}\\ \bar{t} &= t - t_0 + \frac{1}{2}\frac{H(t)a^2(t)}{c^2}x^2 + O(x^4) = t - t_0 + \frac{H^{-1}}{2}\frac{a^2x^2}{{d_H}^2} + O(x^4)\end{aligned} \tag{1.19}$$

The expansion of the Universe sets a natural scale $d_H = c(\dot{a}/a)^{-1} = c/H$. This scale, called the Hubble radius, is the length scale over which physical processes can act coherently

(Peebles, 1980; Padmanabhan, 1993). In the above transformation, spatial coordinates have been changed from the comoving to the proper frame and time has been corrected for gravitational redshift. The coordinate transformations lead to the metric

$$ds^2 = \left[1 - \frac{\ddot{a}}{a}\frac{r^2}{c^2}\right] c^2 d\bar{t}^2 - \left[1 + \left(\frac{\dot{a}}{a}\right)^2 \frac{r^2}{c^2} + \frac{k}{a^2} r^2\right] dr^2 - r^2 \left[d\theta^2 + \sin^2\theta d\phi^2\right] \tag{1.20}$$

We have ignored terms of cubic and higher order in $r\dot{a}/ac = r/d_H$ because we are interested in scales that are small in comparison with the Hubble radius. The coefficient of dr^2 in eqn.1.20 can be set equal to unity if $r \ll d_H$, $r \ll d_k$ and $v \ll c$. Here $d_k = a/\sqrt{k}$ is the curvature scale. The line element then reduces to

$$ds^2 = \left[1 - \frac{\ddot{a}}{a}\frac{r^2}{c^2}\right] c^2 d\bar{t}^2 - dr^2 - r^2 \left[d\theta^2 + \sin^2\theta d\phi^2\right] \tag{1.21}$$

Comparing with eqn.1.18, we can identify the effective Newtonian potential due to the homogeneous and isotropic background Universe as

$$\Phi_{av} = \frac{1}{2}\frac{\ddot{a}}{a} r^2 = \frac{2}{3}\pi G \left(\bar{\rho} + 3p\right) \tag{1.22}$$

For a Universe with non-relativistic matter and a cosmological constant, the effective potential becomes

$$\Phi_{av} = \left(\frac{2}{3}\pi G \bar{\rho}_{nr}(t) - \frac{1}{6}\Lambda\right) = \frac{1}{4} {H_0}^2 \Omega_{nr} \frac{{a_0}^3}{a^3} - \frac{1}{2} {H_0}^2 \Omega_\Lambda \tag{1.23}$$

Here $\bar{\rho}_{nr}(t) = \bar{\rho}_{dm}(t) + \bar{\rho}_b(t)$ is the total density of non-relativistic matter at time t. This has contributions from ordinary baryonic matter $\bar{\rho}_b$ as well as cold dark matter $\bar{\rho}_{dm}$.

For length scales $l \ll d_H$, Newtonian gravity provides an accurate description, i.e., in any region small compared to d_H, one can set up a coordinate system in which the *proper* coordinates of a particle $\mathbf{r}(t) = a(t)\mathbf{x}(t)$ satisfy the Newtonian equations with Φ_{av} as the Newtonian potential. Since non-linear effects due to gravitational clustering also occur at scales $l \ll d_H$, this framework is adequate for our purpose. For us to use the Newtonian limit for studying structure formation, formally we require the following :

- The scale of homogeneity should be much smaller than the Hubble radius. The scale of homogeneity is close to $100h^{-1}$ Mpc (Bharadwaj, Mittal and Seshadri, 1999). This is definitely much smaller than the Hubble radius ($\sim 3000h^{-1}$ Mpc).

- At all scales of interest, the gravitational potential should satisfy $\Phi_{av}/c^2 \ll 1$. Let us consider a homogeneous mass distribution, where the gravitational potential energy for mass M contained within a sphere of proper radius R is $\Phi_{av} \sim GM/R \sim G\bar{\rho}_{nr}R^2 \sim (HR)^2$. At $R \sim 100h^{-1}$ Mpc at the present epoch, we get $\Phi_{av}/c^2 \sim 10^{-3}$. Even highly overdense regions like clusters of galaxies have $\Phi/c^2 \sim 10^{-5}$.

- Particle motions should be non-relativistic. Galaxies and cores of clusters of galaxies are believed to be in virial equilibrium. Therefore, within an order of magnitude, the average velocity of objects, for example, in the core of a cluster of galaxies is given by $v \sim \sqrt{\Phi}$, which gives $v/c \sim 10^{-2.5}$ for the core of a cluster of galaxies.

Given that all the necessary conditions are satisfied, we can safely apply the Newtonian limit in the study of dynamics of clustering of matter.

1.3 Linear Theory of Perturbations

Gravity dominates over all other forces of nature at sufficiently large scales. As most of the matter in the Universe is dark and dark matter responds only to gravity, the role played by gravitational forces becomes doubly important. The complications due to gas physics can be mostly ignored at scales of interest ($l > 1h^{-1}Mpc$) and we can assume that the evolution of density perturbations at large scales is governed by gravitational forces due to dark matter.

Gravitational instability leads to the growth of density perturbations. Overdense regions accrete matter at the expense of underdense regions and in the process inhomogeneities in the density field grow. When the amplitude of density fluctuations is small compared to the average density, it is possible to study the growth of fluctuations analytically using linear perturbation theory (Peebles, 1974a, 1980). Galaxies and clusters of galaxies are however clearly non-linear objects and we need to go beyond linear theory in order to study the process of structure formation. Nevertheless, the linear theory of density perturbations is important and in this section we present a brief review of the formalism.

Motions of non-relativistic matter can be studied in the fluid limit and we can treat non-relativistic matter as an ideal pressureless fluid. This requires that the velocity dispersion be much less than the bulk velocity at the scales of interest. The velocity **V** and the total density ρ, considered as functions of the proper coordinates **r** and time t, satisfy the continuity equation :

$$\left(\frac{\partial \rho}{\partial t}\right)_{\mathbf{r}} + \nabla_{\mathbf{r}} \cdot (\rho \mathbf{V}) = 0 \tag{1.24}$$

Since we ignore pressure, Euler's equation of motion is

$$\left(\frac{\partial \mathbf{V}}{\partial t}\right)_{\mathbf{r}} + (\mathbf{V} \cdot \nabla_{\mathbf{r}})\mathbf{V} = -\nabla_{\mathbf{r}}\Phi \tag{1.25}$$

and Poisson's equation is

$$\nabla_{\mathbf{r}}{}^2\Phi = 4\pi G\rho \tag{1.26}$$

Φ is the total potential corresponding to ρ. The subscript in ∇ indicates that the independent variables are t and **r**. In terms of comoving coordinates, the velocity can be written as

$$\dot{\mathbf{r}} = \mathbf{V} = \dot{a}\mathbf{x} + \mathbf{u}(\mathbf{x}, t) = H\mathbf{r} + \mathbf{u}(\mathbf{x}, t) \tag{1.27}$$

Here $H\ \mathbf{r} = \dot{a}\ \mathbf{x}$ is the Hubble flow and the peculiar velocity is $\mathbf{u}(\mathbf{x}$,t) $= a\dot{\mathbf{x}}$. We write density as

$$\rho(\mathbf{x}, t) = \bar{\rho}[1 + \delta(\mathbf{x}, t)] \tag{1.28}$$

where $\delta(\mathbf{x}, t)$ is the density contrast. It is the fractional departure of the local density from the average background density and is a function of both time t and position **x**. The continuity equation in comoving coordinates is

$$\left(\frac{\partial}{\partial t} - \frac{\dot{a}}{a}\mathbf{x} \cdot \nabla\right) \bar{\rho}(t)[1 + \delta] + \frac{\bar{\rho}}{a}\nabla \cdot [(1 + \delta)(\dot{a}\mathbf{x} + \mathbf{u})] = 0 \tag{1.29}$$

Recalling that $\dot{\bar{\rho}} = -3\bar{\rho}\dot{a}/a$, we get

$$\frac{\partial \delta}{\partial t} + \frac{1}{a}\nabla \cdot [(1 + \delta)\mathbf{u}] = 0 \tag{1.30}$$

In this equation and in the discussion below the spatial derivatives are with respect to the comoving coordinates $\mathbf{x}$. Poisson's eqn.1.26 is

$$\frac{1}{a^2}\nabla^2\Phi = 4\pi G\bar{\rho}(1+\delta) - \Lambda \tag{1.31}$$

We can separate the contribution of the background by writing

$$\Phi = \phi(\mathbf{x},t) + \Phi_{av} = \phi(\mathbf{x},t) + \frac{2}{3}\pi G\bar{\rho}a^2x^2 - \frac{1}{6}\Lambda a^2x^2 \tag{1.32}$$

This gives us

$$\nabla^2\phi = 4\pi G\bar{\rho}a^2\delta \tag{1.33}$$

Euler's equation can be written as

$$\begin{aligned}\left(\frac{\partial}{\partial t} - \frac{\dot{a}}{a}\mathbf{x}\cdot\nabla\right)(\dot{a}\mathbf{x}+\mathbf{u}) \quad &+ \quad \frac{1}{a}(\dot{a}\mathbf{x}+\mathbf{u})\cdot\nabla(\dot{a}\mathbf{x}+\mathbf{u}) \\ &= \quad -\frac{1}{a}\nabla\phi - \frac{4}{3}\pi G\bar{\rho}a\mathbf{x} + \frac{1}{3}\Lambda a\mathbf{x} \qquad (1.34) \\ \Rightarrow \frac{d\mathbf{u}}{dt} + \frac{\dot{a}}{a}\mathbf{u} = \frac{\partial\mathbf{u}}{\partial t} + \frac{\dot{a}}{a}\mathbf{u} \quad &+ \quad \frac{1}{a}(\mathbf{u}\cdot\nabla)\mathbf{u} = -\frac{1}{a}\nabla\phi \qquad (1.35)\end{aligned}$$

The term $\mathbf{u}\dot{a}/a$ indicates that the expansion of the Universe makes the peculiar velocity decay as $\mathbf{u} \propto 1/a(t)$. Eqns.1.30, 1.33 and 1.35 describe the evolution of density fluctuations in an expanding Universe.

If the perturbations are small ($|\delta| \ll 1$), then quadratic terms in perturbed quantities (like $u\delta$ or u^2) can be dropped. In this limit, eqns.1.30 and 1.35 are

$$\begin{aligned}\frac{\partial\delta}{\partial t} \quad &+ \quad \frac{1}{a}\nabla\cdot\mathbf{u} = 0 \\ \frac{\partial\mathbf{u}}{\partial t} \quad &+ \quad \frac{\dot{a}}{a}\mathbf{u} + \frac{1}{a}\nabla\phi = 0 \end{aligned} \tag{1.36}$$

Combining these with the Poisson eqn.1.33, we get

$$\frac{\partial^2\delta}{\partial t^2} + 2\frac{\dot{a}}{a}\frac{\partial\delta}{\partial t} = 4\pi G\bar{\rho}\delta = \frac{3}{2}\frac{\delta}{a^3}{H_0}^2\Omega_{nr} \tag{1.37}$$

The last equality follows from the definition of the Hubble constant and eqn.1.8. This equation describes the evolution of the density contrast δ in linear perturbation theory. It can be shown that, for a Universe with no relativistic component, the Hubble parameter itself is a solution of the linear equation (Heath, 1977). That however is the decaying solution for density perturbations. We can obtain the second solution by using the Wronskian (Heath, 1977).

$$\begin{aligned} X &= 1 + \Omega_{nr}\left(\frac{1}{a} - 1\right) + \Omega_\Lambda\left(a^2 - 1\right) \\ b(t) &\propto \frac{X^{1/2}}{a}\int^a \frac{da}{X^{3/2}} \end{aligned} \tag{1.38}$$

Here $b(t)$ is the growing mode of density perturbations. In the Einstein deSitter model, the scale factor varies as $a \propto t^{2/3}$ (eqn.1.15) and eqn.1.37 becomes

$$\frac{\partial^2\delta}{\partial t^2} + \frac{4}{3t}\frac{\partial\delta}{\partial t} = \frac{2}{3t^2}\delta \tag{1.39}$$

with the solution

$$\delta = At^{2/3} + Bt^{-1} \tag{1.40}$$

where A and B are constants. The general solution to eqn.1.37 is of the form $\delta = A(\mathbf{x})b(t) + B(\mathbf{x})c(t)$, where b and c are linearly independent and represent the growing and the decaying solutions respectively. We can safely assume that the growing mode dominates at late times. We can then ignore the decaying solution and write $\delta = A(\mathbf{x})b(t)$.

The velocity field in linear perturbation theory can be obtained as follows. The continuity equation (the second of eqns.1.36), using the growing solution above, takes the form

$$\nabla\cdot\mathbf{u} = -a\frac{\partial\delta}{\partial t} = -a\delta\frac{\dot{b}}{b} \tag{1.41}$$

We can express the velocity field as the sum of a part with no divergence and an irrotational part. The divergence-free part does not contribute to the evolution of the density contrast and decays as $a(t)^{-1}$ (Peebles, 1980). The solution for the irrotational part is

$$\mathbf{u}(\mathbf{x}) = a\frac{fH}{4\pi}\int\frac{\mathbf{y}-\mathbf{x}}{|\mathbf{y}-\mathbf{x}|^3}\delta(\mathbf{y})d^3y \tag{1.42}$$

where

$$f = \frac{a}{\dot{a}}\frac{\dot{b}}{b} = \frac{1}{H}\frac{\dot{b}}{b} = \frac{d\log b}{d\log a} \tag{1.43}$$

Comparing eqn.1.41 with the equation for the acceleration $\mathbf{g} = -\nabla\phi/a$, we see that the peculiar velocity can be written as

$$\mathbf{u} = \frac{fH}{4\pi G\bar{\rho}}\mathbf{g} = \frac{2}{3}\frac{f}{\Omega_{nr}H}\mathbf{g} \tag{1.44}$$

In the Einstein deSitter model, where $\Omega_{nr} = 1 = f$ and $H = 2/(3t)$, the peculiar velocity has the simple form $\mathbf{u} = \mathbf{g}t$.

For $\Omega_\Lambda = 0$ and at redshift $z = 0$, f can be expressed in terms of the Hypergeometric function (Lahav et al., 1991) :

$$f = -\Omega_{nr}/2 - 1 + \frac{5}{2}{\Omega_{nr}}^{3/2}/F\left(\frac{3}{2}, \frac{5}{2}; \frac{7}{2}, 1 - \Omega_{nr}^{-1}\right) \tag{1.45}$$

Useful approximations are $f \sim \Omega_{nr}^{0.6}$ (Peebles, 1980) and $f \sim \Omega_{nr}^{4/7}$ (Lightman and Schechter, 1990).

1.4 The Equation of Motion

In the last section we have derived the evolution equations of the density and velocity fields in the fluid limit. We now switch to a particle representation and replace the density and velocity fields by a discrete distribution of particles with individual positions and velocities. There is no velocity dispersion at early times. The distribution of particles evolves due to gravitational interaction and it can be used to compute the density field.

The dynamical evolution of a system of self-gravitating particles is described by the following equations :

$$\begin{aligned}
\ddot{\mathbf{r}}_i = & - & \nabla_{\mathbf{r}_i}\Phi \\
\nabla^2_{\mathbf{r}}\Phi & = & \nabla^2_{\mathbf{r}}[\Phi_{av} + \phi] = 4\pi G\rho - \Lambda = (4\pi G\bar{\rho} - \Lambda) + 4\pi G\bar{\rho}\delta \\
\rho(\mathbf{r}) & = & \sum_i m_i\delta^3_D(\mathbf{r} - \mathbf{r}_i)
\end{aligned} \tag{1.46}$$

Here $\mathbf{r}_i$ is the position vector of the ith particle in proper coordinates and m_i is the mass of the ith particle. ρ is the density field for the given set of particles, δ the corresponding density contrast field. Φ is the total gravitational potential field, being the sum of contributions from the background Universe Φ_{av} and perturbations ϕ. δ_D is the Dirac Delta function. We will drop the subscript i from particle positions for convenience. Transforming eqns.1.46 to comoving coordinates $\mathbf{x} = \mathbf{r}/a(t)$ and using eqn.1.32, we obtain

$$\begin{aligned} \ddot{\mathbf{x}} &+ 2\frac{\dot{a}}{a}\dot{\mathbf{x}} = -\frac{1}{a^2}\nabla_x\phi \\ \nabla_x{}^2\phi &= 4\pi G a^2\bar{\rho}\delta = \frac{3}{2}H_0{}^2\Omega_{nr}\frac{\delta}{a} \end{aligned} \tag{1.47}$$

The last equality in the second equation follows from the definition of the Hubble constant and eqn.1.8. The equation of motion (eqns.1.47) can be cast in a useful form by using the growing mode $b(t)$ as "*time*" :

$$\begin{aligned} \frac{d\mathbf{v}}{db} &= -\frac{Q}{b}(\mathbf{v} - \mathbf{g}) \\ \nabla^2\psi &= \frac{\delta}{b} \end{aligned} \tag{1.48}$$

with

$$\begin{aligned} \mathbf{g} &= -\nabla\psi = -\frac{2}{3H_0{}^2\Omega_{nr}}\left(\frac{a}{b}\right)\nabla\phi \\ Q &= \left(\frac{\bar{\rho}}{\rho_c}\right)\left(\frac{\dot{a}b}{a\dot{b}}\right)^2 = \left(\frac{\bar{\rho}}{\rho_c}\right)\left(\frac{d\log a}{d\log b}\right)^2 = \left(\frac{\bar{\rho}}{\rho_c}\right)\frac{1}{f^2} \end{aligned} \tag{1.49}$$

Here $\mathbf{v} = d\mathbf{x}/db$ is the generalised peculiar velocity defined with the parameter b as the time, ψ is the generalised gravitational potential and all the other symbols carry their usual meaning. We have dropped the subscript x from spatial derivatives for convenience. This form of the equation of motion is particularly useful as the acceleration $d\mathbf{v}/db = 0$ in the linear regime. This allows us to relate the velocity $\mathbf{v}$ to the gravitational force $\mathbf{g}$ in a simple mannner. For the Einstein deSitter model, $b(t) = a(t)$ and $Q = 1$. Therefore, the equation of motion reduces to

$$\frac{d\mathbf{v}}{da} = -\frac{3}{2a}(\mathbf{v} - \mathbf{g}) \tag{1.50}$$

1.5 Statistical Analysis of Clustering

Models of structure formation do not explain the exact structures we see in our Universe; at best, theorists can hope to construct models of the Universe that are statistically equivalent. In this section, we give a brief summary of the various statistical indicators that are used for comparing models with observations.

1.5.1 Correlation Functions and Power Spectra

Spatial properties of a statistically homogeneous set of points (be it galaxies or simulation particles) are fully characterized by n-point correlation functions (de Vaucouleurs, 1971; Peebles, 1971, 1980). The first of these is the two-point correlation function. This is defined as follows :

$$\xi(\mathbf{r}) = \langle \delta(\mathbf{x})\delta(\mathbf{x}+\mathbf{r})\rangle \tag{1.51}$$

The angular brackets indicate an averaging extending over all space. $\xi(\mathbf{r})$ is therefore the *auto-correlation* function of the density contrast field.

$$\xi(\mathbf{r}) = \frac{1}{V}\int d^3x\delta(\mathbf{x})\delta(\mathbf{x}+\mathbf{r}) \tag{1.52}$$

If we express δ in terms of its Fourier transform and note that $\delta(\mathbf{x})$ is real, so that $\delta(-\mathbf{k}) = \delta^*(\mathbf{k})$, we obtain

$$\begin{aligned}
\xi(\mathbf{r}) &= \frac{1}{V}\int d^3x \int \frac{d^3k}{(2\pi)^3}\delta_{\mathbf{k}}\exp[-\imath\mathbf{k}\cdot\mathbf{x}]\int\frac{d^3k_1}{(2\pi)^3}\delta_{\mathbf{k_1}}\exp[-\imath\mathbf{k_1}\cdot(\mathbf{x}+\mathbf{r})] \\
&= \int\frac{d^3k}{(2\pi)^3}\delta_{\mathbf{k}}\int\frac{d^3k_1}{(2\pi)^3}\delta_{\mathbf{k_1}}\exp[-\imath\mathbf{k_1}\cdot\mathbf{r}]\left[\frac{1}{V}\int d^3x\exp[-\imath(\mathbf{k}+\mathbf{k_1})\cdot\mathbf{x}]\right] \\
&= \int\frac{d^3k}{(2\pi)^3}\delta_{\mathbf{k}}\int\frac{d^3k_1}{(2\pi)^3}\delta_{\mathbf{k_1}}\exp[-\imath\mathbf{k_1}\cdot\mathbf{r}]\delta_D(\mathbf{k}+\mathbf{k_1}) \\
&= \int\frac{d^3k}{(2\pi)^3}\left|\delta_{\mathbf{k}}\right|^2\exp[-\imath\mathbf{k}\cdot\mathbf{r}] \\
&= \int\frac{d^3k}{(2\pi)^3}P(\mathbf{k})\exp[-\imath\mathbf{k}\cdot\mathbf{r}]
\end{aligned} \tag{1.53}$$

The two-point correlation function is therefore the Fourier transform of the power spectrum $P(\mathbf{k})$.

In an isotropic Universe, the density perturbation spectrum cannot contain a preferred direction and therefore we must have an isotropic power spectrum such that $\left|\delta_{\mathbf{k}}\right|^2$ depends only on the magnitude of $\mathbf{k}$ and not on its direction. The same argument is valid for the correlation function and hence $\xi(\mathbf{r}) = \xi(r)$. The angular part of the k-space integral (eqn.1.53) can then be performed. In three dimensions, this yields

$$\xi(r) = \frac{1}{(2\pi)^3} \int P(k) \frac{\sin kr}{kr} 4\pi k^2 dk = \int \frac{k^3 P(k)}{2\pi^2} \frac{\sin kr}{kr} \frac{dk}{k} \tag{1.54}$$

The 2d analogue of this formula is

$$\xi(r) = \frac{1}{(2\pi)^2} \int P(k) J_0(kr) 2\pi k dk \tag{1.55}$$

The power spectrum is a measure of density fluctuations at scale k. We usually express the power spectrum in dimensionless form in terms of the quantity $\Delta^2(k)$ which is defined as

$$\Delta^2(k) = \frac{k^3 P(k)}{2\pi^2} \tag{1.56}$$

$\Delta^2(k)$ is the power spectrum of density fluctuations, but in logarithmic intervals in k. Thus $\Delta^2(k) = 1$ means that density fluctuations have a typical amplitude equal to unity for modes in a logarithmic bin around wavenumber k. $\Delta^2(k)$ is therefore the Fourier-space counterpart to the dimensionless quantity $\xi(r)$.

If the density field is a Gaussian random field, it is completely described by the power spectrum and all the moments of the distribution can be expressed in terms of integrals over the power spectrum. In most cosmological models the initial density field is chosen to be a Gaussian random field.

An alternative definition of the two-point correlation function is in terms of the excess number of neighbours at a given distance from a randomly chosen point/particle. We can define a hierarchy of three-point, four-point and n-point correlation functions in terms of the excess number of triplets, quadruplets, etc. Thus, if a point is chosen from a distribution, the average number of neighbours at a distance r in an infinitesimal volume element δV is

$$\delta N_2 = \bar{n} \delta V [1 + \xi(r)] \tag{1.57}$$

Here n is the mean number density in the distribution. Similarly, the three-point correlation function can be defined in terms of the number of pairs to complete a triangle with a randomly chosen point :

$$\delta N_3 = \bar{n}^2 \delta V_2 \delta V_3 [1 + \xi(r_{12}) + \xi(r_{23}) + \xi(r_{31}) + \zeta(r_{12}, r_{23}, r_{31})] \tag{1.58}$$

Here $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$; $i, j = 1, 3$ form the three sides of the triangle.

The usual method to compute ξ is based on counting pairs of points as a function of pair separation. Higher-order correlations are often estimated and characterized by their volume averages, the irreducible moments (cumulants) of counts in cells μ_N (de Vaucouleurs, 1971; Peebles, 1971, 1980). The volume averaged correlation function $\bar{\xi}(r)$ may be interpreted in terms of the excess number of neighbours with separation less than r and is given by

$$\bar{\xi}(r) = \frac{3}{r^3} \int_0^r y^2 \xi(y) dy = 3 \int_0^\infty \frac{dk}{k} \Delta^2(k) \left(\frac{\sin(kr) - kr\cos(kr)}{k^3 r^3} \right) \tag{1.59}$$

1.5.2 Smoothed Fields

A useful tool in the study of density fields is filtering, where the density field is smoothed over some length scale. This allows us to focus on averaged quantities and many observational results are expressed in this form. Some common three-dimensional filter functions are as follows :

$$\begin{aligned} \textit{Gaussian} \quad &: \quad F = \frac{1}{2\pi^{3/2} {R_G}^3} e^{-r^2/2{R_G}^2} \Rightarrow F_k = e^{-k^2 {R_G}^2/2} \\ \textit{Spherical Tophat} \quad &: \quad F = \frac{3}{4\pi {R_T}^3};\ r \leq R_T \Rightarrow F_k = \frac{3}{y^3} (\sin y - y\cos y)\ ;\ y \equiv kR_T \end{aligned}$$

F_k here is the Fourier transform of the filter function. F is taken to be equal to zero beyond $r = R_T$ for the spherical tophat filter function.

We are often interested in the variance of the mass distribution at a given scale. This is given by

$$\sigma_R{}^2 = \frac{1}{2\pi^3} \int P(k) \left| F_k(R) \right|^2 d^3k \tag{1.60}$$

An important measure is σ_8 which is the r.m.s. amplitude of mass fluctuations when smoothed with a spherical tophat filter of radius $8h^{-1}$ Mpc. It is an absolute measure of fluctuations at the given scale and, unlike the correlation function, it is a positive definite quantity.

1.5.3 Moments of Counts in Cells

Given a distribution of points, the volume averaged n-point correlation functions can be computed easily using the *counts-in-cells* method. In this method, we randomly distribute a large number of cells in the volume containing an arbitrary distribution of points. We compute the distribution of counts of points in cells and we can relate the moments of the distribution to the volume averaged n-point correlation functions as shown here. We reproduce some of the general results (Peebles, 1980) for the case where the points are counted in a randomly placed cell.

A convenient way to compute the moments of the count N of points in a cell with volume V is to imagine that the cell is divided into infinitesimal elements with n_1 points in the element δV_1. The probability that $n_1 = 1$ is $\bar{n}\delta V_1$ and the probability that $n_1 > 1$ is an infinitesimal of higher order.

$$\bar{n}\delta V_1 = \langle n_1 \rangle = \langle {n_1}^3 \rangle = \langle {n_1}^2 \rangle = ... \tag{1.61}$$

to order δV because ${n_1}^m = n_1$ if $n_1 = 0, 1$. The product $n_1 n_2$ for the counts in the disjoint elements δV_1, δV_2 is equal to unity if there are points in both elements and the probability for this is given by

$$\delta P = \bar{n}^2 \delta V_1 \delta V_2 [1 + \xi(r_{12})] \tag{1.62}$$

where points in elements δV_1 and δV_2 are separated by distance r_{12}. Therefore, it follows that

$$\begin{aligned} \langle n_1 n_2 \rangle &= \bar{n}^2 \delta V_1 \delta V_2 (1 + \xi_{12}) \\ \langle (n_1 - \langle n_1 \rangle) \rangle \langle (n_2 - \langle n_2 \rangle) \rangle &= \bar{n}^2 \delta V_1 \delta V_2 \xi_{12} \end{aligned} \tag{1.63}$$

The count in the cell is

$$N = \sum n_1 \tag{1.64}$$

By eqn.1.61, the average is

$$\langle N\rangle = \sum\langle n_1\rangle = \int_V \bar{n}dV = \bar{n}V \tag{1.65}$$

The second moment is

$$\left\langle N^2\right\rangle = \sum\langle {n_1}^2\rangle + \sum\langle n_1 n_2\rangle \tag{1.66}$$

The squared terms are given by eqn.1.61, the cross terms by eqn.1.63 :

$$\begin{aligned} \left\langle N^2\right\rangle &= \bar{n}V + (\bar{n}V)^2 + I_2 \\ \mu_2 &= \left\langle (N-\bar{n}V)^2\right\rangle = \bar{n}V + I_2 \end{aligned} \tag{1.67}$$

with

$$I_2 = \bar{n}^2 \int_V dV_1 dV_2 \xi_{12} \tag{1.68}$$

If $\xi = 0$, this reduces to $\mu_2 = \bar{n}V$ as for a Poisson distribution. If the points are correlated ($\xi > 0$), the dispersion in N is increased. We can perform a change of variables from r_1 and r_2 to $r = |\mathbf{r}_1 - \mathbf{r}_2|$ and r_1 and integrate out r_1 in I_2. Eqn.1.67 for the second moment then reduces to

$$\mu_2 = \bar{n}V + \bar{n}^2 V \int d^3r\xi(r) = \bar{n}V + \bar{n}^2V^2\bar{\xi}(r) = \langle N\rangle + \langle N\rangle^2 \bar{\xi}(r) \tag{1.69}$$

Similarly, the third moment is expressible as

$$\mu_3 = 3\mu_2 - 2\bar{n}V + I_3 \tag{1.70}$$

with

$$I_3 = \bar{n}^3V^3\bar{\zeta}(r) \tag{1.71}$$

For ready reference, the equations relating the moments of counts of points to the two, three and four-point correlation functions are as follows :

$$\begin{aligned} \langle N\rangle^2 \bar{\xi}(r) &= \mu_2 - \langle N\rangle \\ \langle N\rangle^3 \bar{\zeta}(r) &= \mu_3 - 3\mu_2 + 2\langle N\rangle \\ \langle N\rangle^4 \bar{\eta}(r) &= \mu_4 - 6\mu_3 - 3{\mu_2}^2 + 11\mu_2 - 6\langle N\rangle \end{aligned} \tag{1.72}$$

where η is the four-point correlation function.

1.5.4 The Reduced Moments

The reduced moments are defined as :

$$S_Q(r) = \frac{\bar{\xi}_Q(r)}{\bar{\xi}_2(r)^{Q-1}} \tag{1.73}$$

where $Q = 3, 4, 5$, etc. and $\bar{\xi}_2 = \bar{\xi}$, $\bar{\xi}_3 = \bar{\zeta}$, $\bar{\xi}_4 = \bar{\eta}$, etc. Most cosmological models presume an initial Gaussian random density field. Non-linear evolution however leads to a non-Gaussian distribution at small scales. The lowest-order non-Gaussian signatures are probability distributions for density that display asymmetry (skewness) or a non-Gaussian degree of peakiness (kurtosis). These deviations are respectively measured through the reduced moments S_3 and S_4. From second order perturbation theory (Peebles, 1980; Colombi, Bouchet and Hernquist, 1996), it can be shown that S_Q should not depend on scale in the weakly non-linear regime ($|\delta| \ll 1$). The expected results (in the weakly non-linear regime) are (Juszkiewicz, Bouchet and Colombi, 1993; Bernardeau, 1994)

$$\begin{aligned} S_3 &= \frac{34}{7} - (n+3) \\ S_4 &= \frac{60712}{1323} - \frac{62}{3}(n+3) + \frac{7}{3}(n+3)^3 \end{aligned} \tag{1.74}$$

Here n is the index of the initial density power spectrum.

In the strongly non-linear regime, no general analytical solutions to the BBGKY hierarchy have been found. However, if we can assume local statistical equilibrium at small-enough scales, corresponding to virialised objects, then S_Q is expected to be independent of scale (Davis and Peebles, 1977; Colombi, Bouchet and Hernquist, 1996). Simulations are consistent with a phenomenological fit for S_3 (Fry, Melott and Shandarin, 1993; Colombi, Bouchet and Hernquist, 1996) given by

$$S_3 = \frac{9}{n+3} \tag{1.75}$$

Methods of Computing the Reduced Moments

There are two ways of computing the n-point correlation functions for a distribution of objects like galaxies or points obtained from, say a N-Body simulation. One is a direct but brute-force method in which we draw spheres of radius R with arbitrary centres within the simulation volume. Then we determine the probablility distribution $P_N(R)$ of counts N of points within spheres, compute moments of counts N using

$$\mu_Q(R) = \left\langle (N - \bar{N})^Q \right\rangle = \sum_0^\infty (N - \bar{N})^Q P_N(R), \tag{1.76}$$

and finally compute $\bar{\xi}_2(R), \bar{\xi}_3(R)$, etc. using eqns.1.72. This method gives accurate results at all scales, but is computationally expensive, i.e. requires a large amount of CPU time because the computing time $t \propto N_p N_{sph} \left[R/L_{box}\right]^3$, where N_{sph} is the no. of spheres of radius R drawn and ${L_{box}}^3$ is the volume of the simulation box containing N_p particles. This becomes more and more expensive as we get to larger R.

The second method takes advantage of the speed of fast Fourier transforms (FFT). In this, we first create an equispaced grid in the simulation volume. Next, starting from the distribution of points, we define a number density $n(x_i)$ on the grid using an appropriate interpolating function (compute time $t \propto N_p$), Fourier transform the number density (compute time $t \propto N_p \log N_p$) and convolve the number density with a spherical tophat filter function of radius R (compute time $t \propto N_p$). The inverse Fourier transform of $\tilde{N}(k) = \tilde{n}(k)\tilde{F}_R(k)$ i.e. $N(x_i)$ gives us the number of points (or counts) within a sphere of radius R at grid point x_i (compute time $t \propto N_p \log N_p$). We can determine the probablility distribution $P_N(R)$ of counts N and as before compute the quantities of interest in subsequent steps. Though this method is faster [overall O$(N_p \log N_p)$, where N_p is the total number of points as before], results are expected to be inaccurate at scales comparable with the grid scale. This is illustrated in fig.1.2 where we have plotted the second and the third moments of particle distribution S_2 (upper panel) and S_3 (lower panel) respectively for a power law simulation with $n = -0.5$ using the TreePM code. S_2 and S_3 are plotted as functions of scale r. In both panels, the solid lines show the statistics computed using the direct method, while the dashed lines show the same for the method using FFT. The scale r is in units of the spacing in the grid constructed for the FFT method. The two methods give identical answers at scales large compared to the grid spacing, but at scales comparable to the grid spacing, the FFT method leads to large inaccuracies in the computed S_2 and S_3. The convergence scale for the two methods is higher for S_3 compared to S_2.

1.5.5 Redshift Space Statistics

In a Universe that obeys the cosmological principle, the clustering of galaxies is statistically isotropic. But in galaxy surveys, the distance to a galaxy is inferred from the redshift z, making the line of sight ($\hat{z}$) a special direction. Peculiar velocities of galaxies produce an anisotropy in redshift-space clustering on all scales (Kaiser, 1987). On small scales, the random motions of galaxies in virialised systems stretch groups and clusters into so-called "fingers-of-god". On large scales, coherent flows created by gravity compress overdense regions along the line of sight and stretch underdense regions correspondingly. The redshift distance of a galaxy differs from its true distance by its peculiar velocity $u_g \equiv \hat{z} \cdot \mathbf{u}_g$ along the line of sight $\hat{z}$. The effect of peculiar velocities on galaxy distributions has been extensively studied with simulations (Matsubara, 1994). This is also used to determine cosmological parameters with the help of observations, e.g. as in the 2dF Galaxy Redshift Survey (Verde et al., 2002). The statistics of interest in redshift space are the redshift space correlation function $\xi(\mathbf{s})$ as well as higher order correlation functions. Formally, the redshift space correlation function $\xi^{\mathbf{s}}(\pi, \sigma)$ is the mean fractional excess of galaxy neighbours of a galaxy at separations π and σ parallel and perpendicular to the line of sight respectively. The statistic is a function only of the components π and σ of the pair separation if the angle between the positions of any two galaxy pairs is small enough such that the line-of-sight redshift distortions are effectively plane-parallel.

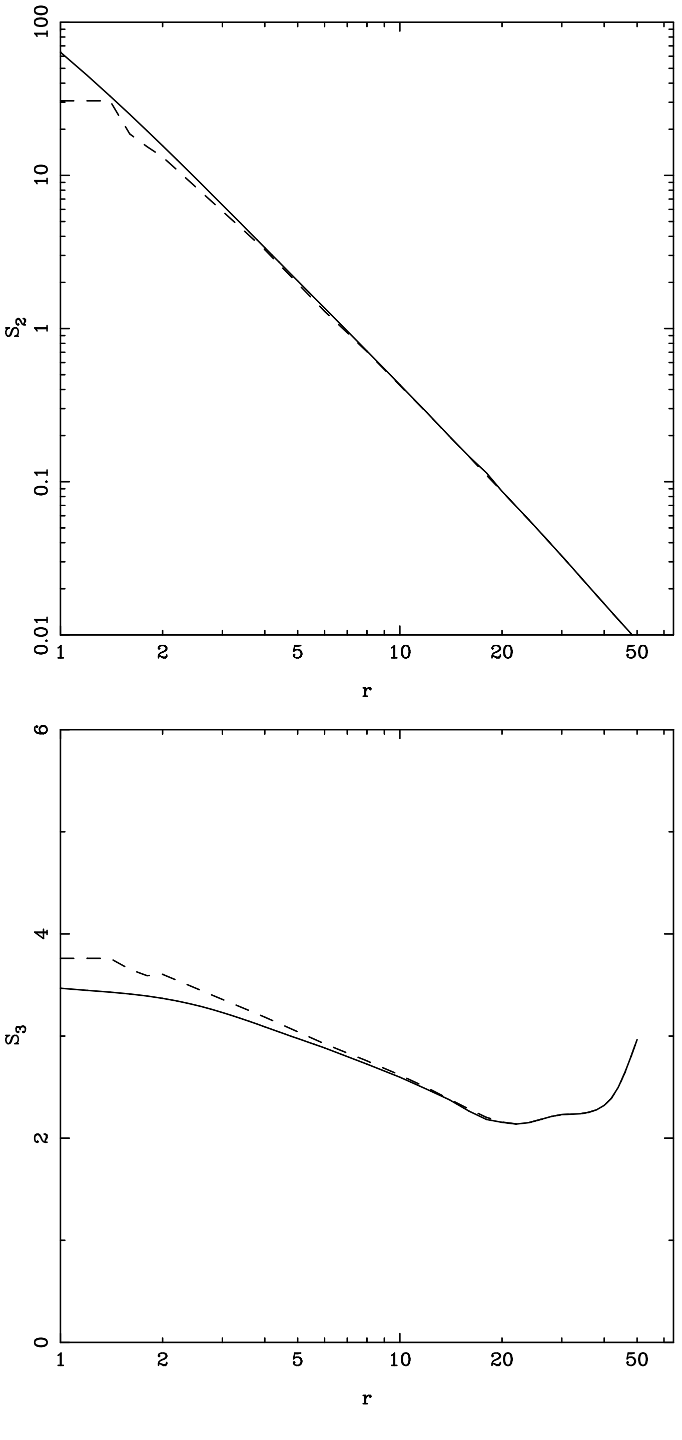

Figure 1.2: This figure shows the second moment S_2 (upper panel) and the third moment S_3 (lower panel) as a function of scale r for a power law simulation with $n = -0.5$ using the TreePM code. In both panels, the solid lines show the statistics computed using the direct method described in the text, while the dashed lines show the same for the method using FFT. The horizontal axis r is in units of the grid spacing. The values of S_2 and S_3 obtained using FFT are inaccurate at small scales, but match their direct determination at larger scales.

1.5.6 The Radial Pair Velocity

The last statistical indicator of interest to us is the dimensionless radial pair velocity h. It is defined as

$$h(\mathbf{r}, a) = -\mathbf{v}_p(\mathbf{r}, a) \cdot \mathbf{r}/Hr^2 \tag{1.77}$$

where $\mathbf{v}_p(\mathbf{r}, a)$ denotes the mean relative velocity of pairs of objects at separation $\mathbf{r}$ at epoch a. h is a ratio of the radial component of the mean relative velocity $\mathbf{v_p}$ of pairs and the velocity of the Hubble expansion, both evaluated at the same scale $\mathbf{r}$. This makes h a dimensionless quantity. In the linear limit, i.e. when $\bar{\xi} \ll 1$, $h = (2/3)\,\bar{\xi}$ (Peebles, 1980; Hamilton et al., 1991; Nityananda and Padmanabhan, 1994). In two dimensions, the corresponding relation is $h = \bar{\xi}$. In the extreme non-linear limit, if we can ignore the effect of mergers, virialised systems should maintain their sizes in proper coordinates (the *stable clustering* limit) and peculiar motion can be assumed to compensate for the Hubble expansion to form bound structures. In that case $h = 1$ (Peebles, 1980).

1.6 Light and Mass and Bias

Observations suggest that baryonic matter makes up only a small portion of the Universe - about 5% (Ω_b), while cold dark matter (Ω_{dm}) and dark energy (Ω_Λ) contribute about 25% and 70% respectively. In other words, almost 90% of the matter in the Universe which can give rise to structures is dark. Observations show that the dynamics of large structures like galaxies, clusters of galaxies, etc. is dominated by dark matter. Dark matter behaves like non-relativistic dust at large scales.

Cosmologists believe that galaxies and clusters of galaxies formed by accretion of matter on to small inhomogeneities present in the early Universe. In the standard scenario of structure formation, dark matter aggregates into clumps, the virialised parts of which are usually called haloes. Galaxies then form by the cooling and condensation of baryonic matter (gas) within these haloes (Hoyle, 1953; Rees and Ostriker, 1977; Silk, 1977; White and Rees, 1978) subject to the condition that the baryonic gas cloud has a mass greater than the Jeans mass. Pressure opposes gravity effectively and a gas cloud will not collapse under its own gravity unless it attains a mass greater than the Jean's mass given by

$$M_J = \rho {\lambda_J}^3 \tag{1.78}$$

where λ_J is the Jeans length given by

$$\lambda_J = c_s\sqrt{\frac{\pi}{G\rho}} \tag{1.79}$$

Here c_s is the velocity of sound and ρ is the density of the gas cloud. Physically, λ_J is the length at which sound-crossing time is equal to the collapse time $t_{coll} = G\rho^{-1/2}$.

Evolution of galaxy clustering and its relation to the clustering in the underlying dark matter mass distribution is an important question that needs to be addressed before we can interpret observations of clustering of galaxies. Models of structure formation make predictions about the distribution of dark matter in terms of correlation functions and higher moments of the matter distribution function. However, all we see are galaxies. Observations tell us only about the clustering properties of galaxies and the distribution of galaxies does not necessarily trace the underlying dark matter density field. Thus an additional function that describes the difference between clustering of dark matter and galaxies is required. This function is called bias. Bias can be defined in a statistical manner as

$$b^2 = P_{gal}(k,t)/P_{dm}(k,t) \tag{1.80}$$

A similar definition of bias in real space is

$$b^2 = \bar{\xi}_{gal}(r,t)/\bar{\xi}_{dm}(r,t) \tag{1.81}$$

Here all the quantities carry their usual meaning. Bias defined as above is called deterministic bias. Many studies have shown that the relation between galaxy clustering and that of the underlying dark matter is not simple and this relation can, in general, be a complicated function of scale, redshift and galaxy type (Gelb and Bertschinger, 1994a; Brainerd and Villumsen, 1994; Mo and White, 1996; Bagla, 1998b; Tegmark and Peebles, 1998; Dekel A. and Lahav O., 1999; Sheth and Lemson, 1999; Sheth, Mo and Tormen, 2001; Norberg, 2001, 2002).

1.6.1 Linear Redshift Distortions

In the preceding section we have introduced the concept of bias. In this subsection, we will briefly review the analytical framework within which it is possible to estimate bias from observations using linear perturbation theory. Observations of large scale structures are carried out primarily in redshift space.

The Hubble law can be restated as follows : the recession velocity cz of a galaxy is proportional to its distance d from us, i.e. $cz = H_0 d$. The recession velocity cz of a galaxy can be measured from the redshift z of its spectrum a great deal more easily and accurately than its true distance d. This has been a primary motivation for redshift surveys which map the Universe in three dimensions using the recession velocity cz of each galaxy as a measure of its distance. Galaxies also have peculiar velocities $\mathbf{u}$ relative to the general Hubble expansion. Thus it is necessary in general to distinguish between a galaxy's *redshift distance* s :

$$s \equiv cz \tag{1.82}$$

and its true distance r (in the same units) :

$$r \equiv H_0 d \tag{1.83}$$

The redshift distance s of a galaxy differs from the true distance r by its peculiar velocity $u_g \equiv \hat{r}.\mathbf{u}_g$ along the line of sight :

$$s = r + u_g \tag{1.84}$$

The peculiar velocities of galaxies thus cause them to appear displaced along the line of sight in redshift space. These displacements lead to redshift distortions in the clustering of galaxies in redshift space. Although such distortions complicate the interpretation of redshift maps as positional maps, they have the advantage of bearing information about the dynamics of galaxies. Details of how peculiar velocities lead to redshift distortions is illustrated schematically in fig.1.3. The dots are 'galaxies' and the arrows represent their peculiar velocities. At large scales, the galaxies can be seen to be undergoing infall towards a spherical overdensity; the peculiar velocity of an infalling shell is small compared to its

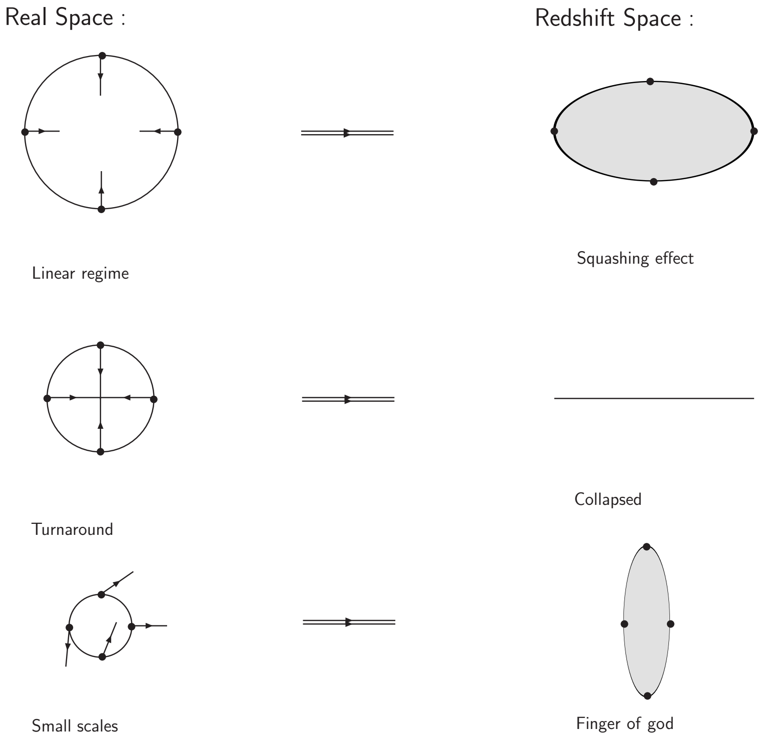

Figure 1.3: A simple schematic representation of redshift space distortion.

radius and the shell appears squashed. The shell that is just at turnaround, its peculiar velocity just cancelling the general Hubble expansion, appears collapsed to a single velocity in redshift space. At small scales, not only is the radius of a shell smaller, but also its peculiar velocity tends to be larger. At small scales, large *random* peculiar motions in virialised structures give rise to the *finger-of-god* effect.

The amplitude of redshift distortions on large scales is related to the linear redshift distortion parameter β which is related to the density parameter Ω_{nr} by

$$\beta = \frac{f(\Omega_{nr})}{b} \approx \frac{{\Omega_{nr}}^{0.6}}{b} \tag{1.85}$$

b here is the bias factor and $f(\Omega_{nr})$ is the factor defined in eqn.1.43. A measurement of beta, in principle, combined with an independent accurate measurement of the cosmological parameters by observations like the Supernova Cosmology Project (Perlmutter et al., 1999) and WMAP (Spergel et al., 2003) can be used to get a handle on the linear bias parameter. For example, if we combine WMAP measurements ($\Omega_{nr} \simeq 0.3$) with the 2dF

Galaxy Redshift Survey results (Verde et al., 2002), which give a value for $\beta \simeq 0.43$, we find that the linear bias parameter $b \simeq 1$.

1.6.2 Identification of Haloes

In order to quantify bias through systematic N-Body simulation studies, one needs to identify particles in "galaxies" from a full distribution of particles evolved in a simulation. We may postulate that galaxies populate overdense regions in a uniform manner. There are different methods of identifying overdense regions in numerical simulations. We will describe a few of the methods here. We have used these methods to study the distribution of overdense regions and haloes obtained using N-Body simulations.

- *Friends-Of-Friends (FOF) Method* : This is an efficient algorithm for halo identification (Davis et al., 1985). The FOF method involves scanning through the list of particles in a distribution. For each particle, one looks for particles whose distances from it are smaller than some length l. This length is called the linking length. The magnitude of the linking length specifies a minimum density cutoff for the environment surrounding any particle. If the distance to a particle is smaller than the linking length, one makes a "link" and then looks around the linked particle in a similar manner to identify new links. Each linked particle is also marked at the same time as taken, i.e. it is assumed to belong to a unique group or halo. The entire process is repeated till no more neighbours, i.e. particles which are within a linking length of any linked particle, are to be found. The process has to be repeated for all unlinked particles to generate other independent groups or haloes. In FOF, every particle in a halo gets linked to every other particle in the same halo either through a direct link (friend) or through links mediated by particles which share a direct link with it (friend of friend). FOF sometimes fails to find separate groups in cases when those groups are obviously present. It is known to link dynamically distinct haloes (Gelb, 1992; Gelb and Bertschinger, 1994). The problem originates from the tendency of FOF to "percolate" through bridges connecting, for example, interacting galaxies or galaxies in high density backgrounds (Klypin, 2000).

- *Lagrangian Density Method* : We compute the Lagrangian density at the positions of particles. This is done by ordering neighbouring particles by distance from the particle and then using the volume in which N particles are enclosed to estimate the density. Recommended values of N are of order of 10, certainly less than 100. The volume enclosed is taken to be a sphere of radius R, where R is the average of the distance to the Nth and $(N + 1)$th particles. All particles in high density regions are identified by specifying a cutoff on the density. This method on its own does not identify individual haloes or groups in isolation. However it can be adapted for that purpose, as for example, in the group-finding algorithm *HOP* (Eisenstein and Hut, 1998).

We have also studied the density profiles of individual haloes obtained using N-Body simulations. The method that we have used for this purpose is called the *Method of Density Peaks* (Tormen, Bouchet and White, 1997), where we find points of maxima of density. Then we identify particles which belong to any maximum, i.e. lie within a certain distance, say r, from it. A density field is constructed on a uniform grid from the particle distribution using an interpolating kernel (Cloud-in-Cell or CIC in our case). The density field is smoothed by convolution with a spherical tophat window function of an appropriate smoothing radius. Local maxima of density are identified by scanning the nearest grid points around every grid point. Each maximum or peak identified is mapped to a cluster or a halo and the cluster centre is identified as the most dense point of the clump, found iteratively as the centre of mass of spheres of increasingly smaller radius. One starts with a large enough radius and the iteration is stopped when a fixed small number of particles, say m, is left in the sphere. The position of the centre of mass of these m particles is then identified as the final cluster centre. The centre found with this method corresponds with very high accuracy to the highest density peak even in cases of ongoing mergers between different clumps (Tormen, Bouchet and White, 1997). The only free parameter in this method is the number m that determines when the iterative calculation should stop. It has been found that the scatter in the locations of the halo centres due to different choices of m is negligible. We chose $m = 15$ for our studies.

Chapter 2

Cosmological N-Body Simulations

Numerical simulations of structure formation in the Universe are a powerful tool to understand and interpret cosmological observations. Simulations bridge the gap that often exists between basic theory and observations. Simulations are used for testing and calibrating methods used for processing observations, to provide insight into non-linear gravitational clustering, etc. The task of numerical simulations is to evolve density and velocity perturbations with time in a given cosmological model.

Cosmological N-Body simulations that take only gravitational interactions into account can be used to obtain the large scale distribution of galaxies and clusters of galaxies. Cosmological simulations can incorporate a wide range of physics like gas dynamics (Hernquist and Katz, 1989; Katz and Gunn, 1991; Hernquist, 1993; Steinmetz and Mueller, 1993), chemistry and radiative transfer (Cen et al., 1990; Hernquist et al., 1996; Meiksin and White, 2001). Inclusion of gravitational effects is mandatory and the rest can be looked upon as refinements. State-of-the-art simulations can be used for studying a variety of problems like the nature of the inter-galactic medium and the formation of first stars. Our focus in this thesis is isolating and understanding the generic aspects of evolution of gravitational clustering and we concentrate only on simulations involving gravitational interactions.

2.1 Struture of a General N-Body Code for Cosmological Simulations

In cosmological simulations, we use comoving coordinates and periodic boundary conditions. Density and velocity distributions are represented by particles sampling the phase space distribution. Trajectories are evolved using eqns.1.49.

An N-Body code consists of two modules - one computes the force field for a given configuration of particles and the other moves the particles in this force field. These two are called at each step so that the force field and the distribution of particles evolve in a self-consistent manner. Apart from these, we need to set up the initial conditions.

The following important physical requirements have to be taken into account while constructing a viable N-Body code for cosmological simulations.

- Density averaged over larger and larger scales in the Universe tends towards a constant value. Thus the simulation volume V cannot be assumed to exist in isolation. Periodic boundary conditions are a simple solution to this problem (Peebles, 1980).

- The average density of the simulation volume (usually a cube in three dimensions) should be equal to the average density of the Universe. Thus perturbations averaged at the scale of the simulation box and at larger scales must be ignorable at all times in the model of interest i.e. $\sigma(R = V^{1/3}) \ll 1$. For example, in the case of the ΛCDM model this implies that the box should be at least $150h^{-1}$ Mpc at $(z = 0)$ in extent (Bagla and Ray, 2004).

- We must probe scales that are sufficiently non-linear in order to justify the use of N-Body simulations. If we are interested in comparing the results of a N-Body simulation with observations, simulations should be able to study the formation of galaxies. If we want to resolve masses of the order of galaxies, the masses of individual particles must be much less than the mass of a galaxy $(M_{part} \ll M_{gal})$

and hence the minimum number of particles required for a box of comoving size, say $150h^{-1}$ Mpc will be

$$N_p \geq \frac{M(150h^{-1}Mpc)}{0.1M_{galaxy}} \simeq \frac{5 \times 10^{17} M_\odot}{10^{10} M_\odot} \simeq 5 \times 10^7 \tag{2.1}$$

Thus we need a very large number of particles for realistic simulations. Cosmological N-Body methods are oriented towards reducing the number of operations and stored quantities per particle in order to facilitate large simulations.

- In an N-Body simulation, we have to approximate a large number of particles by a single particle. Therefore particles in an N-Body simulation must interact in a purely collisionless manner at small separations. The desirable force is a softened inverse square law representing the force between two finite-sized particles with extended mass distribution, say $W(r, \epsilon)$. ϵ here is called the softening length and it represents the finite size of a particle. The force due to a finite-sized particle can be obtained by solving the Poisson equation with a source term given by the mass distribution of the particle :

$$\rho(r) = mW(r, \epsilon) \tag{2.2}$$

For a normalised cubic spline kernel used in the SPH formalism (Monaghan, 1992), $W(r, \epsilon)$ has the following form :

$$W(r, \epsilon) = \left(\frac{16}{3\epsilon^3}\right) \begin{cases} 1 - 6\left(\frac{r}{\epsilon}\right)^2 + 6\left(\frac{r}{\epsilon}\right)^3, & 0 \geq \frac{r}{\epsilon} \leq 0.5 \\ 2\left(1 - \frac{r}{\epsilon}\right)^3, & 0.5 < \frac{r}{\epsilon} \leq 1.0 \\ 0\ , & \frac{r}{\epsilon} > 1.0 \end{cases} \tag{2.3}$$

The cubic spline-softened force $\mathbf{f}$ is given by the following expression :

$$\mathbf{f}(\mathbf{r}) = \begin{cases} -\left\{\frac{2}{3} - \frac{12r^2}{5\epsilon^2} + \frac{2r^3}{\epsilon^3}\right\} \frac{16Gm}{\epsilon^3}\mathbf{r}\ , & 0 \geq \frac{r}{\epsilon} \leq 0.5; \\ \left\{\frac{\epsilon^3}{15r^3} - \frac{64}{3} + \frac{48r}{\epsilon} - \frac{192r^2}{5\epsilon^2} + \frac{32r^3}{3\epsilon^3}\right\} \frac{Gm}{\epsilon^3}\mathbf{r}\ , & 0.5 < \frac{r}{\epsilon} \leq 1.0; \\ -\frac{Gm}{r^3}\mathbf{r}\ , & \frac{r}{\epsilon} > 1.0 \end{cases} \tag{2.4}$$

The softening length for the force is the smallest scale that can be resolved in an N-Body simulation and this should be of the same order as the mean inter-particle separation for collisionless evolution. Smaller force softening requires more timesteps and also leads to increased two-body relaxation (Huang, Dubinski and Carlberg, 1993; Melott et al., 1997; Binney and Knebe, 2002).

2.1.1 Initial Conditions

Initial conditions for simulations of structure formation consist of specifying the background cosmological model and a description of perturbations to be studied. The background model is generally taken to be a Robertson-Walker spacetime with a specified mix of dark matter, baryons, cosmological constant, etc. The nature of dark matter must be specified as well. The statistical nature of the primordial fluctuations depends on how these are generated. We will assume the initial density field to be a Gaussian random field with a specified power spectrum $P_\delta(k)$ in our simulations (Klypin and Shandarin, 1983; Efstathiou et al., 1985; Bagla and Padmanabhan, 1997a; Hoffman and Ribak, 1991; Bertschinger, 2001). Gaussian random fields are simple to study as these are fully specified by one function, the power spectrum $P(k)$ (Peacock and Heavens, 1985; Bardeen et al., 1986). The Fourier components of a Gaussian random field (both the real and the imaginary parts) are random numbers which follow a normal distribution with variance proportional to the power spectrum of the random field. This property is used to generate a Gaussian random field in Fourier space and an inverse transform gives us the field in real space.

Once the initial linear density fluctuation field (or the initial gravitational potential) has been specified at some initial time, the next task is to generate particle positions and velocities. There are two schemes for generating the initial density field.

- The particles are distributed uniformly and their masses are chosen so that $M = \bar{\rho}(t_{in})(1+\delta_{in})dV$, where δ_{in} is evaluated at the positions of the particles. Here dV is the volume in the simulation box per particle and $\bar{\rho}(t_{in})$ is the average density at time t_{in}. We can either start with zero velocities, in which case we have to increase the amplitude of ϕ_{in} by a factor 5/3 to account for the presence of the decaying mode. Alternatively we can choose to put the system in the growing mode and assign an appropriate velocity to each particle.

- The second method for dark matter simulations is to displace equal-mass particles from a uniform distribution using the Zeldovich approximation (Zeldovich, 1970; Doroshkevich et al., 1980; Dekel, 1982; Efstathiou et al., 1985). It is important to ensure that in this case the maximum displacement for any particle is smaller than the average inter-particle separation in the simulation box. Then the resulting distribution of particles will represent the required density field (Bagla and Padmanabhan, 1997a; Bagla, 2004b) if the initial distribution did not have any inhomogeneities. We can retain the initial velocity field used to displace particles. If the amplitude of displacements used is larger than the average inter-particle separation, it becomes necessary to recompute the potential from displaced positions and assign initial velocities with this potential (Efstathiou et al., 1985). Such large displacements can lead to an incorrect realisation of the power spectrum.

The source term in the Poisson equation (eqn.1.48) is constant in the linear regime, implying that the potential and hence the gravitational force **g** do not evolve with time in this regime. It can also be shown that the generalised velocity **v** is also a constant in this regime. This implies

$$\mathbf{v}(\mathbf{x}, b) = \mathbf{v}(\mathbf{x}) = \mathbf{g}(\mathbf{x}, b) = \mathbf{g}(\mathbf{x}) \tag{2.5}$$

This equation is valid only in the regime where the amplitude of perturbations is small i.e. $|\delta| \ll 1$. However, one can extrapolate this equation beyond the regime where it is strictly valid. This extrapolation leads to the Zeldovich approximation where velocities are constant in Lagrangian space. Thus the velocity of each particle is constant and equals its initial velocity given by

$$\mathbf{v}(\mathbf{x}, b) = \mathbf{v}(\mathbf{q}) = \mathbf{g}(\mathbf{q}) \tag{2.6}$$

where **q** is the Lagrangian position of the particle, same as its initial position and **x** is its Eulerian position. Then

$$\mathbf{x} = \mathbf{q} + b(t)\mathbf{g}(\mathbf{q}) \tag{2.7}$$

Particles move with a uniform velocity that is determined by the initial potential. The Zeldovich approximation is a one-step mapping between the initial and the final condition.

Approximate constancy of velocities results from two opposite effects acting over similar time scales, namely the expansion of the Universe and the gravitational force of density perturbations.

To generate initial conditions in N-Body simulations by the second method described above, particles are displaced from a uniform grid according to the prescription given by eqn.2.7. $\mathbf{q}$ labels the position of a particle on the grid. We need to first generate the gravitational potential ψ in order to obtain $\mathbf{g} = -\nabla\psi$ in eqn.2.7. As the potential ψ and the initial density contrast are related through a linear equation (eqn.1.48), it follows that

$$P_\psi(k,a) = P_\delta(k,a)/a^2k^4 = P_\delta^{lin}(k,a=1)/k^4 \tag{2.8}$$

Here $P_\delta^{lin}(k,a=1)$ is the linearly extrapolated power spectrum of initial density fluctuations.

2.1.2 Integrating the Equations of Motion

We integrate the equation of motion using a second-order accurate leap-frog integration scheme as this requires only one force evaluation per timestep.

$$\begin{aligned} x_i(t+\Delta t) &= x_i(t) + \Delta t v_i(t+\Delta t/2) + O(\Delta t^3) \\ v_i(t+3\Delta t/2) &= v_i(t+\Delta t/2) + \Delta t a_i(t+\Delta t) + O(\Delta t^3) \end{aligned} \tag{2.9}$$

This method is called the leap-frog method (Hockney and Eastwood, 1988) because it updates velocities halfway between the step that is used to update the position. Force evaluation is very time-consuming and one has to minimise the number of force evaluations per timestep. It is mainly due to this reason that cosmological N-Body simulations use the leap-frog method as it requires only one evaluation of force, but still gives errors of $O(\Delta t^3)$. We can recast the leap-frog method so that velocities and positions are defined at the same instant (Hut, Makino and McMillan, 1995).

It has been shown that with a timestep such that that all particles move by less than the softening length (the Courant condition), the correlation function and other quantities converge to correct values (Quinn et al., 1997). Also timestep is chosen so that momentum is conserved and energy evolves according to the Irvine-Layzer equation (Layzer, 1963). Monitoring consistency with the Irvine-Layzer equation requires care and adds considerably to the number of operations to be carried out in an N-Body simulation. Hence it is usual to carry out test runs and fix the value of timestep Δt. The ideal Δt will depend on the distribution of particles and as the distribution evolves, the desirable value of Δt changes. We allow Δt to vary with time.

For velocity-dependent forces, as is the case for the equations that we use, the leap-frog method is replaced by a predictor-corrector integrator. Here the equation for the velocity is solved iteratively. Higher-order schemes of integration provide more accurate trajectories with longer timesteps. But these are rarely used in cosmological simulations as these require multiple force evaluations per step.

2.1.3 Scale Invariant Evolution of Power Law Spectra

Hierarchical clustering from scale-free initial conditions i.e. initial power spectra $P(k) \propto k^n$ in an Einstein deSitter Universe is expected to evolve in a self-similar manner. A powerful test for any N-Body simulation is the requirement that the evolution of power law models be scale-invariant even in the strongly non-linear regime. Although scale-free initial conditions differ from realistic models, they are useful for understanding non-linear gravitational instability and have therefore been studied extensively. The only scale in purely gravitational evolution of a scale-free initial power spectrum is the scale at which non-linearity sets in, defined by $\xi(r_{nl}) = 1$. As $\xi(r) \propto a^2$ in linear theory, we have

$$r_{nl}(t) \sim a^{2/(n+3)} \tag{2.10}$$

For clustering in two dimensions,

$$r_{nl}(t) \sim a^{2/(n+2)} \tag{2.11}$$

If evolution is scale invariant, $\xi(r, a) = \xi(r/r_{nl})$ at all epochs. This requirement can be used to fix the value of the timestep Δt for integration of the equation of motion.

2.2 Algorithms for N-Body Simulations

Integration of the equation of motion in an N-Body code is a process of O(N_p), where N_p is the total number of particles in the simulation. But the calculation of force, if performed by summing the force over all pairs, is a process of O(${N_p}^2$). Evaluating the force by direct summation over all particle pairs is prohibitive except with special purpose computers like GRAPE (Toshikazu et al., 1993). Many schemes have been proposed to replace direct summation over pairs by approximate methods of force evaluation.

2.2.1 The Particle-Mesh Method

The Particle-Mesh (PM) method is based on representing the gravitational potential on a Cartesian grid (with a total of N_g grid points) and solving the Poisson equation on this grid. The Fast Fourier Transform (FFT) algorithm (Cooley and Tukey, 1965) is used to solve the Poisson equation with ($N_g \log N_g$) operations. Periodic boundary conditions are the default in this method, making this algorithm natural for cosmological simulations (Klypin and Shandarin, 1983; Bouchet and Kandrup, 1985). The PM method has been discussed in great detail in Hockney and Eastwood (1988).

The PM algorithm has three major components. The density field represented by particles is computed on a grid. The Poisson equation is then solved for the gravitational potential ϕ and the force $\mathbf{g} = -\nabla\phi$. Finally, the potential or the force on the grid is interpolated on to particle positions.

The first step is called mass assignment : $\rho(x,t)$ is computed on the grid from discrete particle positions and masses. The simplest method assigns each particle to the nearest grid point (NGP), with no contribution of mass to any other grid point. This method produces rather large errors (Efstathiou et al., 1985; Hockney and Eastwood, 1988). The most commonly used assignment scheme is the Cloud-in-Cell (CIC), which uses multilinear interpolation to the eight grid points defining the cubical mesh cell containing the particle.

This procedure effectively treats each particle as a uniform-density cubical cloud. The sharp edges introduce force fluctuations, which can be reduced by using a higher-order interpolation scheme (e.g. Triangular Shaped Cloud or TSC, which uses the nearest 27 grid points). These discretization errors are similar to aliasing errors that occur in image processing. They may be reduced further using a suitable anti-aliasing filter [Quiet PM (Hockney and Eastwood, 1988)].

The Poisson equation is then solved in Fourier space :

$$\phi_{\mathbf{k}}(t) = -4\pi G \frac{\rho_{\mathbf{k}}(t)}{k^2} \tag{2.12}$$

Here $\phi_{\mathbf{k}}$ and $\rho_{\mathbf{k}}$ are the discrete Fourier transforms of the density and potential respectively. One may replace the inverse Laplacian operator $-1/k^2$ by another multiplicative factor in Fourier space, including an anti-aliasing filter. The gravitational field is then obtained by transforming the potential back to the spatial domain and approximating the gradient by finite differences, or by multiplication by $\imath\mathbf{k}$ in the Fourier domain. The latter method requires twice as many FFTs and more memory allocation for its implementation. But it does not necessarily require more computing cycles compared to the former and leads to more accurate forces (Ferrell and Bertschinger, 1994).

The third step is to interpolate the gravitational force from the grid back to the particles. The same interpolation scheme should be used here as in mass assignment to ensure that self-forces on particles vanish (Hockney and Eastwood, 1988) and momentum is conserved. The PM method has the advantage of speed, requiring $\mathrm{O}(N_g \log N_g)$ operations to evaluate the force on all particles. For typical grid sizes (with as many grid points as particles in each dimension), it requires less memory and is faster per timestep than other algorithms. However, the force approximates the inverse square law poorly for particles separated by less than several grid spacings. Each particle has an effective diameter of about one grid spacing and a non-spherical shape. PM codes are widely used for large scale cosmological simulations. The PM algorithm has been parallelized (Ferrell and Bertschinger, 1994; Merz, Pen and Trac, 2004). For more details on the Particle-Mesh method, we refer the reader to other sources (Hockney and Eastwood, 1988; Bouchet and

Kandrup, 1985).

2.2.2 The P^3M and the Adaptive P^3M Methods

The main shortcoming of the Particle-Mesh method is that its force resolution is limited by the spatial grid. This limitation can be removed by supplementing the force with a direct sum over pairs separated by less than a few grid spacings, resulting in the particle-particle/particle-mesh (P^3M) algorithm. This hybrid algorithm, first developed for plasma physics by Hockney et al. (1974), was applied in cosmology by Efstathiou and Eastwood in 1981. It is described in detail in Hockney and Eastwood (1988) and Efstathiou et al. (1985). The P^3M method achieves high resolution in force through the combination of mesh-based and direct summation forces. The mesh can be regarded as simply a convenience for providing periodic boundary conditions and removing much of the burden of computation from direct pair summation. However, when clustering becomes strong, the cost of the direct summation dominates and this degrades the performance of P^3M codes. The mesh-refined P^3M algorithm (Couchman, 1991) presents an elegant solution to this bottleneck. A refined mesh is placed over regions of high density. This shifts some of the burden of force evaluation away from pair summation and over to a mesh calculation with isolated boundary conditions. Pair summation is still required, but only for pairs with separation less than two spacings of the refined mesh. In this method, multiple levels of refinement can be used to further reduce pair summation in dense regions. Because the pair summation no longer dominates as it does in P^3M, the force computation of the adaptive P^3M (AP^3M) scales as $\mathrm{O}(N_p \log N_p)$. The P^3M algorithm has been parallelized (Ferrell and Bertschinger, 1994; Pearce and Couchman, 1997). The P^3M algorithm has a few shortcomings. The correction for the long range force is assumed to be isotropic, while the standard PM force has anisotropies at the grid scale due to the anisotropic mesh structure. Thus the resulting force (long range + short range correction) is anisotropic at the grid scale. Also, short range correction in force is added only upto 1.5 to 2.0 grid lengths, whereas the PM method is known to underestimate the force out to a much larger distance (Bouchet and Kandrup, 1985). Finally, the refined inter-particle force is softened at scales much smaller than the average inter-particle separation which can lead to two-body scattering and relaxation, which, in turn, may modify the nature of statistical quantities of interest like the two-point correlation function, etc. (Melott et al., 1997; Binney and Knebe, 2002; Bagla, 2004b).

2.2.3 The Tree Method

An alternative to Fourier space based methods is the tree method. In this scheme, the particles are arranged in a hierarchy of groups. When the force on a particular particle is computed, the forces exerted by distant groups are approximated by their lowest multipole moments. In this way, the computational cost for a complete force evaluation is reduced to $\mathrm{O}(N \log N)$ (Appel, 1985). The forces become more accurate if the multipole expansion is carried out to higher order, but the increasing cost of evaluating higher moments makes it more efficient to terminate the multipole expansion and use a larger number of nodes to achieve the desired accuracy (McMillan and Aarseth, 1993).

The Barnes and Hut (BH) method (1986) is commonly used for tree construction. In this scheme, the computational domain is hierarchically partitioned into a sequence of cubes, where each cube contains eight siblings, each with half-the-side length of the parent cube. These cubes form the nodes of an oct-tree structure. The tree is constructed such that each node (cube) contains either exactly one particle or eight smaller nodes, in which case the node carries the monopole (and/or quadrupole moments) of all the particles that lie inside. A schematic illustration of the BH tree is shown in fig.2.1.

Force computation then proceeds by summing up appropriate contributions from tree nodes. In the standard BH tree walk, the multipole expansion of a node of size d is used only if

$$\theta = \frac{d}{r} \leq \theta_c \tag{2.13}$$

where r is the distance of the point of reference to the center-of-mass of the cell and θ_c is a prescribed accuracy parameter also called the cell-opening angle. If a node fulfills

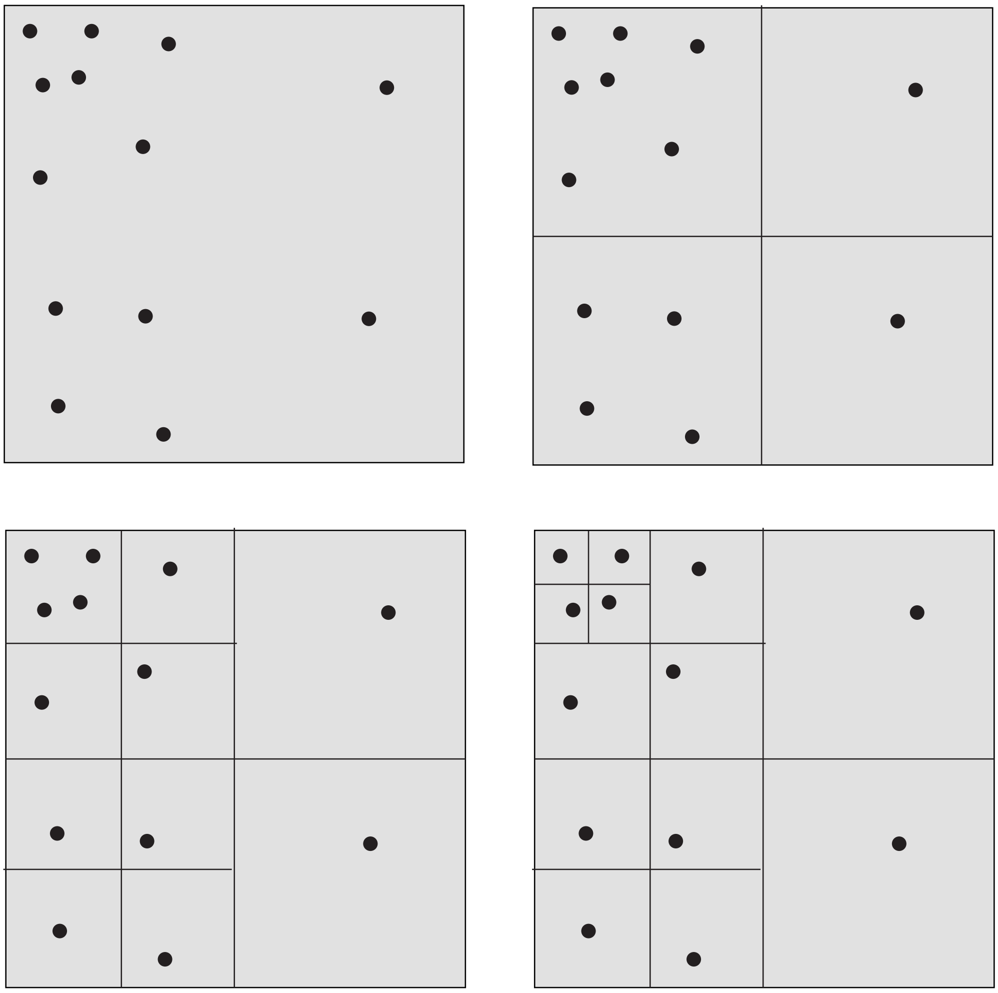

Figure 2.1: Schematic illustration of the Barnes and Hut oct-tree in two dimensions. The particles are first enclosed in a square (root node). This square is then iteratively subdivided into four squares of half the size, until exactly one particle is left in each final square (leaves of the tree). In the resulting tree structure, each square can be progenitor of up to four siblings. Note that empty squares need not to be stored.

the so-called cell acceptance criterion (2.13), the tree walk along this branch can be terminated, otherwise the node is ‘opened’ and the walk is continued with all its siblings. For smaller values of the opening angle the forces become more accurate, but it requires more computing time. Other cell acceptance criteria have been proposed to obtain higher efficiency (Salmon and Warren, 1994; Springel, Yoshida and White, 2001).

A technical difficulty arises when the gravitational force is softened. In regions of high density, it can happen that nodes satisfy the cell acceptance criterion and simultaneously $r < \epsilon$, where ϵ is the force softening length. In this situation, the multipole moments of the softened gravitational field are required. It is possible to work around this situation by opening all nodes for $r < \epsilon$, but this can slow down the code significantly. Another solution is to use the proper multipole expansion for the softened potential (Springel, Yoshida and White, 2001).

The tree algorithm has a number of important advantages (Hernquist, 1987). Foremost is its speed as already mentioned : $\mathrm{O}(N_p \log N_p)$ operations are required to compute all forces on N_p particles. Force errors can be made small by choosing θ_c to be small and by including quadrupole and high-order moments. The hierarchical tree automatically refines resolution where needed. While mesh-based codes are generally much faster for close-to-homogeneous particle distributions, tree codes can adapt to any distribution without significant variation in speed. This Lagrangian nature is a great advantage if a large dynamic range in density is required. Further, tree codes are basically free from any geometrical restrictions and these can be easily used with integration schemes that advance particles on individual timesteps.

One drawback of tree codes is the large amount of memory required - 20 to 30 variables are required per particle. Another complication is that the tree algorithm, like direct summation, does not provide periodic boundary conditions in an obvious manner. This can be incorporated using Ewald summation (Ewald, 1921), leading to a tree code for cosmological applications (Rybicki, 1986; Bouchet and Hernquist, 1988; Hernquist, Bouchet and Suto, 1991; Springel, Yoshida and White, 2001). The tree algorithm can be easily parallelized (Salmon and Warren, 1994; Dubinski, 1996).

PM and tree solvers have been combined into hybrid tree-Particle-Mesh codes (Xu, 1995; Bode, Ostriker and Xu, 2000; Bagla, 2002; Dubinski, 2004). In this approach, the speed of the PM method for computing the long range part of the gravitational force is combined with a tree computation of the short range force. At a basic level, this can be seen as a replacement of the direct summation (particle-particle) part in P^3M codes with a tree algorithm. We will discuss the TreePM method in greater detail in later chapters.

Chapter 3

Performance Characteristics of TreePM Codes

In this chapter we will describe the TreePM method in detail with particular emphasis on the error budget for the method[1]. All methods of N-Body simulations use some approximation or the other to speed up the calculation of force. Approximations are used because direct summation over all pairs of particles scales as $O({N_p}^2)$, where N_p is the number of particles, making the calculation very time consuming for large N_p. The use of these approximations reduces the number of calculations required to $O(N_p \ln N_p)$ or less. These approximations also introduce inaccuracies in the computed force. In this chapter, we analyse errors in estimation in force and also study errors for different distributions of particles for the TreePM method.

3.1 The TreePM Method

Several attempts have been made to combine a tree code with a PM code (Xu, 1995; Dubinski, 2004). A common failing of all such methods is that these continue to use the usual PM force which is known to have large errors and anisotropies at scales comparable to the grid scale (Bouchet and Kandrup, 1985). These errors are present in the final force

[1] This chapter is based on *Performance Characteristics of TreePM Codes* (J.S. Bagla and S. Ray), New Astronomy **8**, 665, 2003.

and there is no natural way of reducing the errors in any of these methods.

The TreePM method is a hybrid N-Body method which combines the Barnes and Hut (1986) tree method with a Particle-Mesh method. However, the philosophy of TreePM method is to modify the long range force in order to have a better control over errors in force at small scales. This is done by an explicit division of the potential and the force into long range and short range components. The PM method is then used to compute only the long range component and the tree method is used to calculate the short range component (Bagla, 2002). Use of a tree code enhances the force resolution. The gravitational potential can be split into two parts in Fourier space (Ewald, 1921) :

$$\begin{aligned}
\varphi_k &= -\frac{4\pi G \varrho_k}{k^2} \\
&= -\frac{4\pi G \varrho_k}{k^2} \exp\left(-k^2 r_s^2\right) - \frac{4\pi G \varrho_k}{k^2}\left[1 - \exp\left(-k^2 r_s^2\right)\right] \\
&= \varphi_k^l + \varphi_k^s \\
\varphi_k^l &= -\frac{4\pi G \varrho_k}{k^2} \exp\left(-k^2 r_s^2\right) \qquad (3.1) \\
\varphi_k^s &= -\frac{4\pi G \varrho_k}{k^2}\left[1 - \exp\left(-k^2 r_s^2\right)\right] \qquad (3.2)
\end{aligned}$$

where φ^l and φ^s are the long range and the short range potentials respectively. The splitting is done at the scale r_s. G is the gravitational coupling constant and ϱ the density. The short range potential and force (for a single particle) are evaluated in real space, using the following expressions :

$$\begin{aligned}
\varphi^s(\mathbf{r}) &= -\frac{GM}{r} \mathrm{erfc}\left(\frac{r}{2r_s}\right) \\
\mathbf{f}^s(\mathbf{r}) &= -\frac{GM\mathbf{r}}{r^3}\left[\mathrm{erfc}\left(\frac{r}{2r_s}\right) + \frac{r}{r_s\sqrt{\pi}} \exp\left(-\frac{r^2}{4r_s^2}\right)\right] \qquad (3.3)
\end{aligned}$$

Here, erfc is the complementary error function. The expression for the long range force for a single particle in real space is :

$$\mathbf{f}^l(\mathbf{r}) = -\frac{GM\mathbf{r}}{r^3}\left[\mathrm{erf}\left(\frac{r}{2r_s}\right) - \frac{r}{r_s\sqrt{\pi}} \exp\left(-\frac{r^2}{4r_s^2}\right)\right]. \qquad (3.4)$$

We have plotted the long range (eqn.3.4) and the short range force (eqn.3.3) as a function of r in fig.3.1 to show their dependence on scale. We have chosen $r_s = 1$ here.

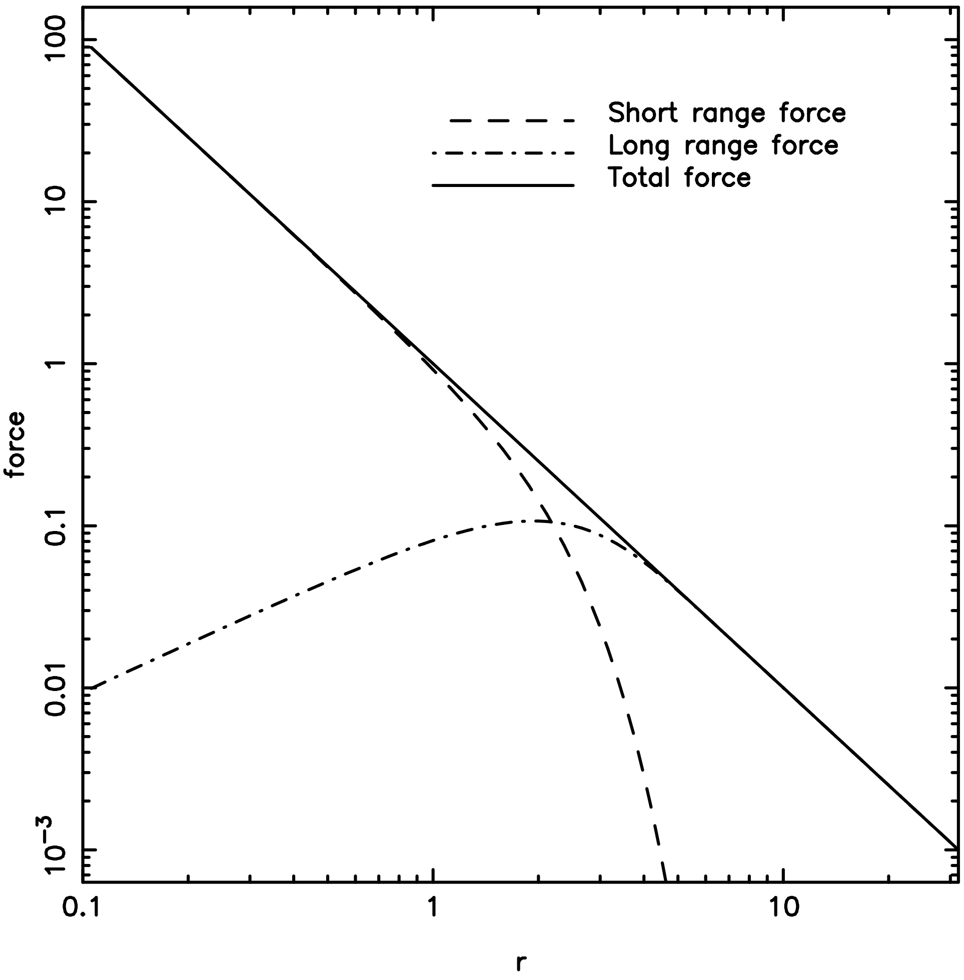

Figure 3.1: This figure shows the long and the short range forces as a function of scale. The inverse square force is shown by the solid line, the long range force by the dot-dashed line and the short range force by the dashed line. We have taken $r_s = 1$ here.

The total $1/r^2$ force is shown by the solid line. The short range force (dashed line) follows the total force up to about $2r_s$ and decreases rapidly at larger scales, its magnitude falls below 1% of the total force by $5r_s$. So the short range force needs to be taken into account only over a small region around every particle - we denote the radius of this region as r_{cut}. The long range force reaches a peak around $2r_s$. It makes up most of the total force beyond $3.5r_s$. It falls linearly with scale below $2r_s$, becoming negligible below $r_s/2$. We take the long range force into account at all scales.

Eqn.3.1 and eqn.3.3 provide a prescription for computing the long range and the short range forces independently. As mentioned above, the short range force is computed in real space using the tree method and the long range force is computed in Fourier space using the PM method. Both of these are computed for every particle in a simulation.

3.1.1 Filter Functions

The filter function used in Ewald's method of splitting the gravitational potential is $F = \exp\left[-k^2 r_s^2\right]$ because this function has a useful feature that it is positive definite, as is its Fourier transform. One can generalise to a case in which the partition of unity can be described in terms of equations as follows :

$$\varphi_{\mathbf{k}} = \varphi^s_{\mathbf{k}} + \varphi^l_{\mathbf{k}} \tag{3.5}$$

$$\varphi^l_{\mathbf{k}} = \varphi_{\mathbf{k}} F(kr_s, \ldots) \tag{3.6}$$

$$\varphi^s_{\mathbf{k}} = \varphi_{\mathbf{k}} \left(1 - F(kr_s, \ldots)\right) \tag{3.7}$$

Here $F(kr_s, \ldots)$ is a filter function that is unity for $kr_s \ll 1$ but decreases rapidly for $kr_s > 1$. As before, r_s is used to partition the short range and long range forces. F can also depend on some other parameters. This manner of partitioning the force allows us to control errors in both components of force. We require a function $F(kr_s, \ldots)$ that decreases sufficiently quickly at $kr_s > 1$. Similarly in real space, the short range potential and force should become negligible at scales larger than a few r_s and the long range force should be negligible below r_s.

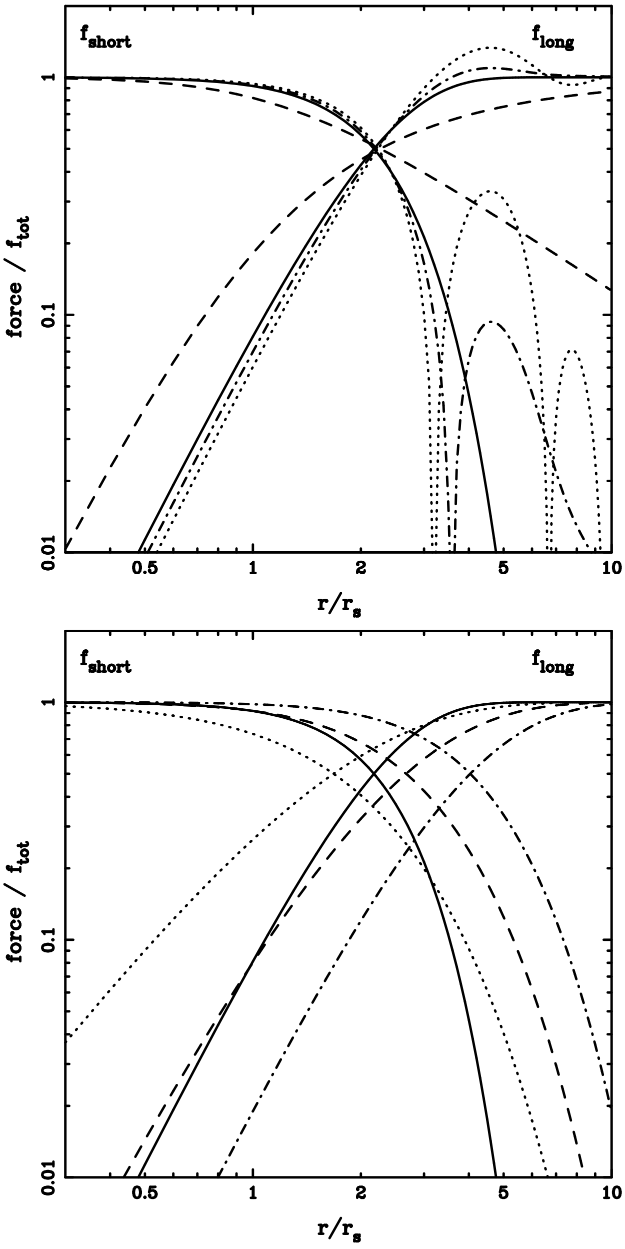

Figure 3.2: In this figure we have plotted the ratio of the short range/long range force and the inverse square force for a choice of filters as a function of distance. The upper panel shows curves for filters described by eqn.3.8 and the lower panel that for filters described by eqn.3.9. For reference, we have also plotted the curve for $\alpha = 2$ in the lower panel. In the upper panel, the dashed curve, the solid curve, the dot-dashed curve and the dotted curve correspond to $\alpha = 1.0, 2.0, 2.5$ and 4.0 respectively. In the lower panel, the solid curve refers to $\alpha = 2.0$ and the dotted, dashed and dot-dashed curves correspond to $n = 1, 2$ and 4 respectively.

We have studied filters belonging to two families of functions :

$$F(kr_s, \alpha) = \exp\left[-\left(k^2 r_s^2\right)^{\alpha/2}\right] \quad (3.8)$$

$$F(kr_s, n) = \left(1 + k^2 r_s^2\right)^{-n} \quad (3.9)$$

Here α and n are additional parameters. For $\alpha = 2$ in eqn.3.8, we recover the one-parameter filter used in Bagla (2002). Fig.3.2 shows the ratios $|f^s/f^{tot}|$ and $|f^l/f^{tot}|$ as functions of scale r for these models. We have chosen $r_s = 1$ for this figure. The upper panel shows these ratios for the models described by eqn.3.8 and the lower panel shows the same for models described by eqn.3.9. In the upper panel, we see that the short range force for $\alpha < 2$ decreases very slowly at large scales, making it an unsuitable choice. For $\alpha > 2$, the short range force oscillates at large scales and the amplitude of the peaks decreases very slowly with scale. Hence these models too are not useful as an alternative to the $\alpha = 2$ model. Models described by eqn.3.9 are shown in the lower panel. In this case the behaviour is better than models with $\alpha \neq 2$ in that the long and the short range forces both fall off much faster. However, the model with $\alpha = 2$ is better than any of these models. This model is plotted in the lower panel as the solid curve. We would like the overlap between the short and long range forces to be as small as possible. For one, a rapidly decreasing short range force implies that we need to sum this over a small region in space. Computation of the short range force is the most time consuming part and hence a sharply falling short range force is required for a well optimised code. The second reason, which requires us to have a long range force that decreases rapidly below some scale, is that the mesh introduces large errors at around the grid scale. The main motivation of considering an explicit division of force into a long range and a short range components is to avoid carrying over the errors in the mesh force by truncating the long range force. Thus we want to use the model for which the range of scales at which both the long range and the short range components are relevant is the narrowest. The Gaussian filter is the optimum one in this regard and no other filter comes close to this. We shall not discuss other filters from this point onwards as fig.3.2 suggests that the Gaussian filter is the best option.

3.2 Error Analysis

We wish to estimate the error in force computed using the TreePM method. We first study errors in the force field of one particle. As the tree approximation does not introduce any errors for a single particle, we can focus exclusively on errors in the long range force.

3.2.1 Errors in the Long Range Force

The long range force is computed by solving eqn.3.4 in Fourier space. The long range force computation is essentially done using the PM method, but with a different kernel. We have described the Particle-Mesh method in Chapter 2. We use the cloud-in-cell (CIC) interpolating function (Hockney and Eastwood, 1988). In three dimensions, the interpolating function is a product of three one-dimensional interpolating functions.

$$W(x, x_i) = 1 - |x - x_i|/L \qquad |x - x_i| \leq L \tag{3.10}$$

$$= 0 \qquad |x - x_i| > L \tag{3.11}$$

Here L is the separation between adjacent mesh points. The Fourier transform of W is given by

$$W(k) = 4\sin^2(kL/2)/(kL)^2 \tag{3.12}$$

Fig.3.3 is a plot of this interpolating function in Fourier space against k/k_g. $k_g = 2\pi/L$ is twice the Nyquist frequency. In the continuum limit we require the interpolating function to be like the Dirac Delta function, the Fourier transform of which will be a constant equal to one. We can see from the figure that the departure from unity for W increases with k, i.e. the error that is induced by the interpolating function is significant at small scales. This leads to a serious shortfall in the force computed in a simulation at small scales. This becomes clear in Fourier space analysis (Bouchet and Kandrup, 1985; Hockney and Eastwood, 1988).

In the Particle-Mesh model, the density field on the mesh is Fourier transformed using a Fast Fourier Transform (FFT). The next step is to solve eqn.3.1 instead of the usual

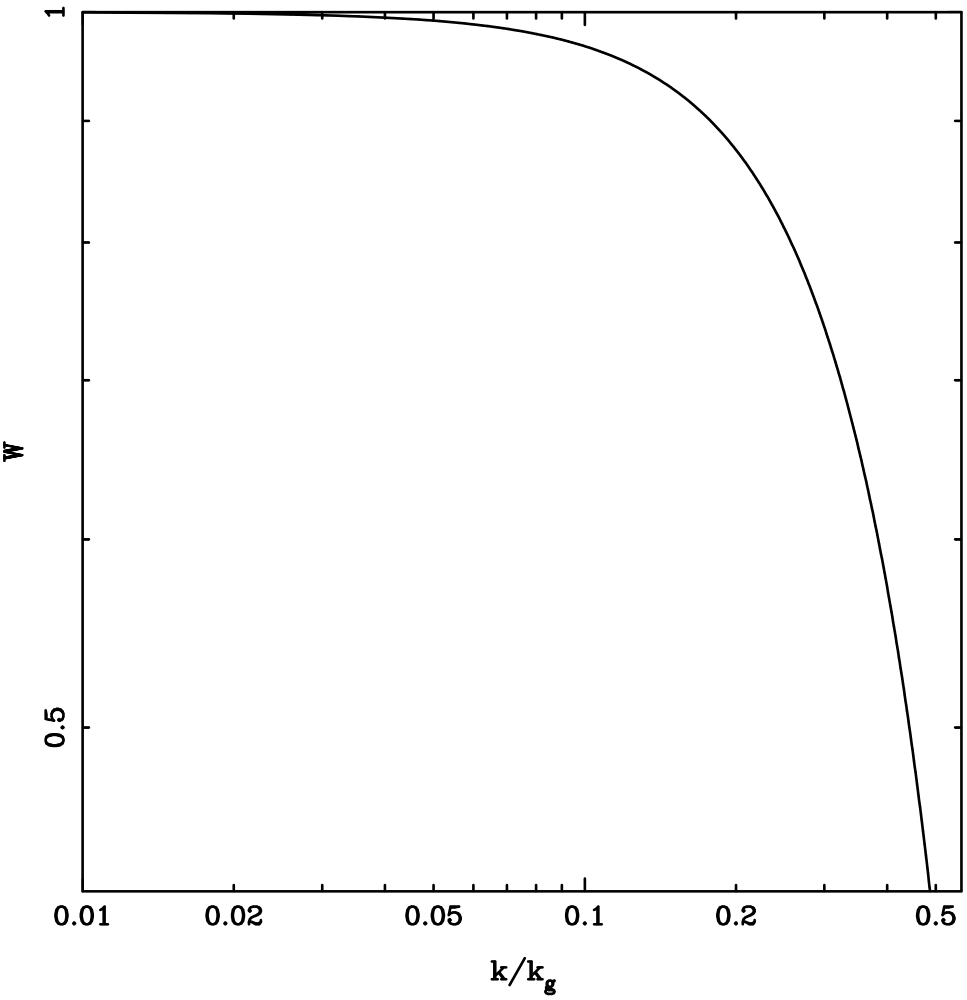

Figure 3.3: We have plotted the CIC interpolating function as a function of k/k_g, where k_g is twice the Nyquist frequency. The Fourier transform of a Dirac Delta function is a straight line at $y = 1$ parallel to the x-axis in this figure.

Poisson equation (eqn.2.12) to get the long range potential on the mesh. We have to multiply this with $\imath\mathbf{k}$ to get the negative gradient of the long range potential i.e. the long range force in Fourier space.

We compare the errors introduced by the interpolating function W in the k-space total force and the k-space long range force for one particle. In our model, the long range force on a unit point mass due to a unit point mass source at the origin is

$$\mathbf{f}^l(\mathbf{k}) = W(-k_x)W(-k_y)W(-k_z)\frac{4\pi G}{k^2}\exp(-k^2r_s^2)\mathbf{k} \tag{3.13}$$

The corresponding total force (the standard Particle-Mesh force) is

$$\mathbf{f}^{pm}(\mathbf{k}) = W(-k_x)W(-k_y)W(-k_z)\frac{4\pi G}{k^2}\mathbf{k} \tag{3.14}$$

The expected long range force is

$$\mathbf{f_{th}}^l(\mathbf{k}) = \frac{4\pi G}{k^2}\exp(-k^2r_s^2)\mathbf{k} \tag{3.15}$$

And the expected total force is

$$\mathbf{f_{th}}(\mathbf{k}) = \frac{4\pi G}{k^2}\mathbf{k} \tag{3.16}$$

We have plotted the expected total force, the expected long range force as well as the total force and the long range force computed from the model as a function of k/k_g in fig.3.4. For simplicity, we have restricted ourselves to one dimension, i.e. we have computed the force for a single component of $\mathbf{k}$ with the interpolating function W also evaluated only along that direction. We have chosen $r_s = 2$ for the plot. For small values of k/k_g, the forces computed by the Particle-Mesh model closely match the expected values. However, deviations grow at large k due to errors induced by the interpolating function.

We define errors in the total force and the long range force in our model as follows :

$$\epsilon^{pm} \equiv \left|\frac{|\mathbf{f_{th}}| - |\mathbf{f}^{pm}|}{|\mathbf{f_{th}}|}\right| \tag{3.17}$$

$$\epsilon^{l} \equiv \left|\frac{|\mathbf{f_{th}}^l| - |\mathbf{f}^{l}|}{|\mathbf{f_{th}}|}\right| \tag{3.18}$$

at all points $\mathbf{k}$. We have plotted ϵ^{pm} and ϵ^l as functions of wavenumber k in fig.3.5. The error in the standard Particle-Mesh force increases monotonically with k, while the error

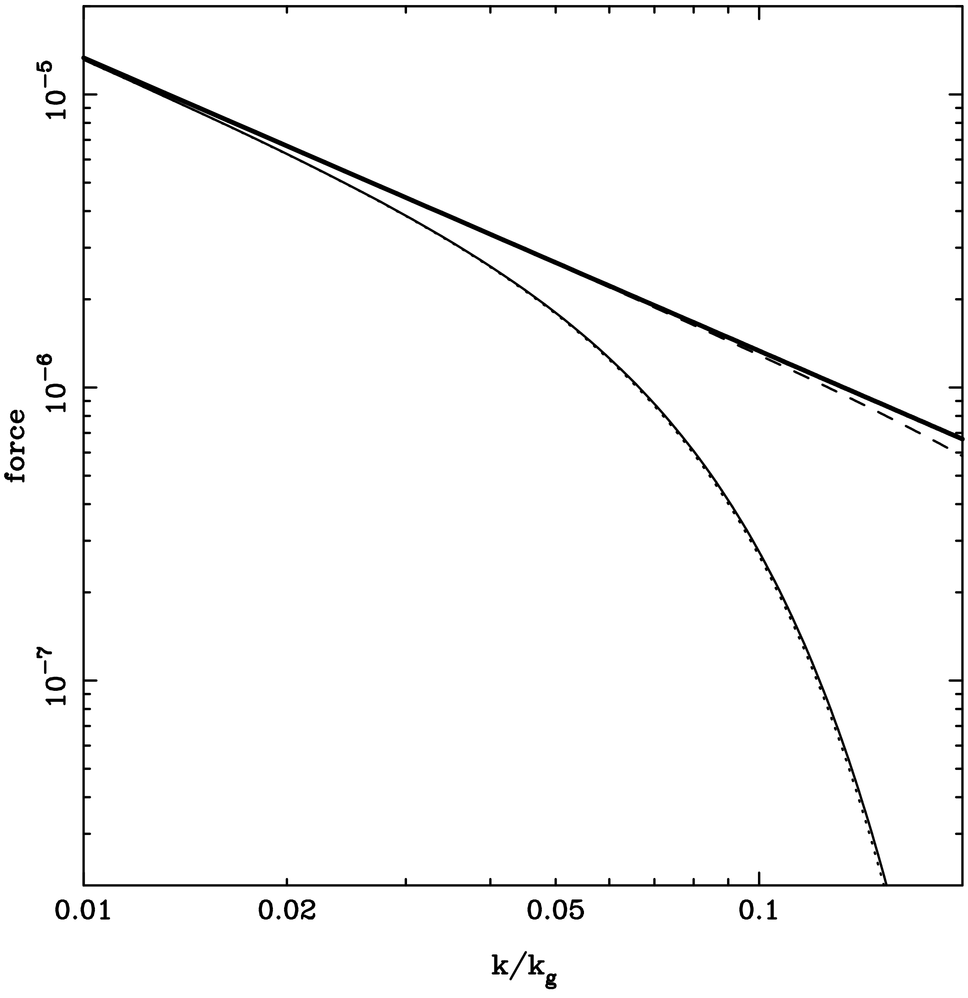

Figure 3.4: We have plotted the Fourier transforms of the expected total force (thick full line), the expected long range force (full line) as well as the total (dashed line) and long range (dotted line) forces obtained from the PM model as a function of k/k_g in 1-d, where k_g is twice the Nyquist frequency. The four curves match at small wavenumbers. But the PM force and the long range force in the model start deviating from the expected total and long range forces respectively at large k or at small scales where the interpolating function starts to play a significant role. We have chosen $r_s = 2$ for this plot.

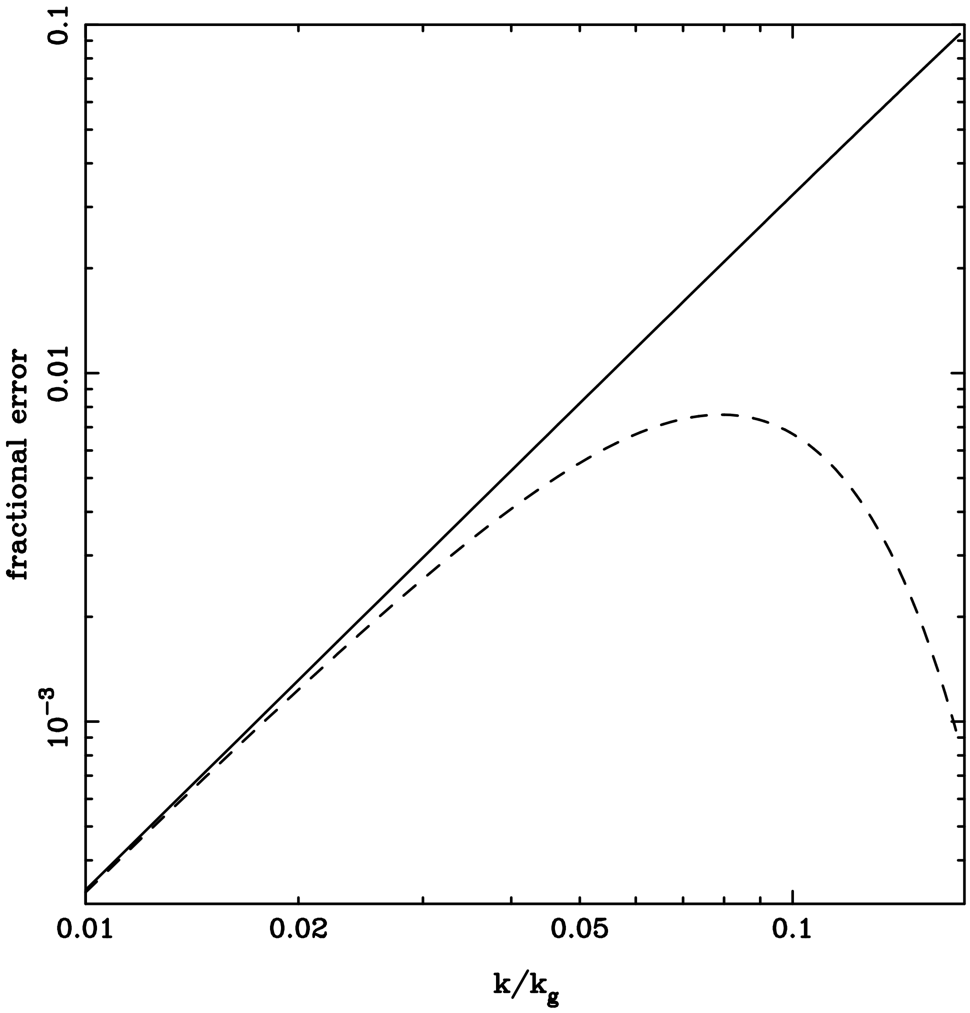

Figure 3.5: This figure shows the fractional error (with respect to the expected total force) in the long range force and the total force in the PM model as a function of k/k_g, where k_g is twice the Nyquist frequency. At large k, the long range force is exponentially small and hence contributes less to the error in the total force. We have taken $r_s = 2$ here.

in the long range force in the model reaches a peak and then decreases with increasing wavenumber. At short scales, the error in the long range force is lower than the error in the total force. This is because the long range force becomes negligible at small scales (refer to fig.3.4) and hence the error contributed to the total force is small.

As the next step we study the long range force in real space. An inverse transform of the long range force in k space using FFT gives us the long range force in real space at mesh points. This is then interpolated back to particle positions using the same interpolating function W as before. In our model, the mesh and the interpolating function W are the main sources of anisotropy. To estimate errors in the long range force in real space, we follow the method prescribed by Efstathiou et al. (1985). We place a particle at a random position in a mesh cell and use the TreePM code to find the long range force at another point. The force is evaluated at a large number of random points within some distance from the particle in question. This process is repeated a number of times with a different position of the particle in the mesh cell. We use force calculated this way to compute the average long range force as a function of distance as well as the dispersion about this average. Fig.3.6 shows a plot of the average long range force (dot-dashed curve) and the expected long range force given by eqn.3.4 (solid curve) as functions of scale. We used $r_s = 1$ for this plot. There is a clear under-estimation of the computed average long range force at small scales.

The difference between the expected force and the computed force is caused mainly by the interpolating function. Fourier space analysis (Bouchet and Kandrup, 1985; Hockney and Eastwood, 1988) suggests that we should be able to recover the expected long range force by de-convolving the interpolating function. We deconvolve[2] the interpolating function twice as we use it twice in force calculation - once for calculating the density on the mesh and then for calculating the force at the positions of particles.

Deconvolution of force will not work in general as it will enhance anisotropies introduced by the mesh. However in this case the contribution of wave modes near the Nyquist frequency is negligible due to the Gaussian filter. Partial correction can be obtained by deconvolving for the effect at small wave numbers by considering the Taylor expansion of W_k about $k = 0$ (White, 2000a).

The effect of deconvolution on the long range force is seen in fig.3.6 and fig.3.7. The force obtained in the simulation (dashed line) almost coincides with the expected force over the entire range of scales. We have plotted the difference of the average long range

[2] We divide the potential in k-space by the square of the Fourier transform of the interpolating function. Since the three dimensional interpolating function is a product of three one-dimensional interpolating functions, we are essentially dividing φ_k by $W_{k_x}^2 W_{k_y}^2 W_{k_z}^2$. The square of each of these needs to be taken as we need to correct for two instances of interpolation.

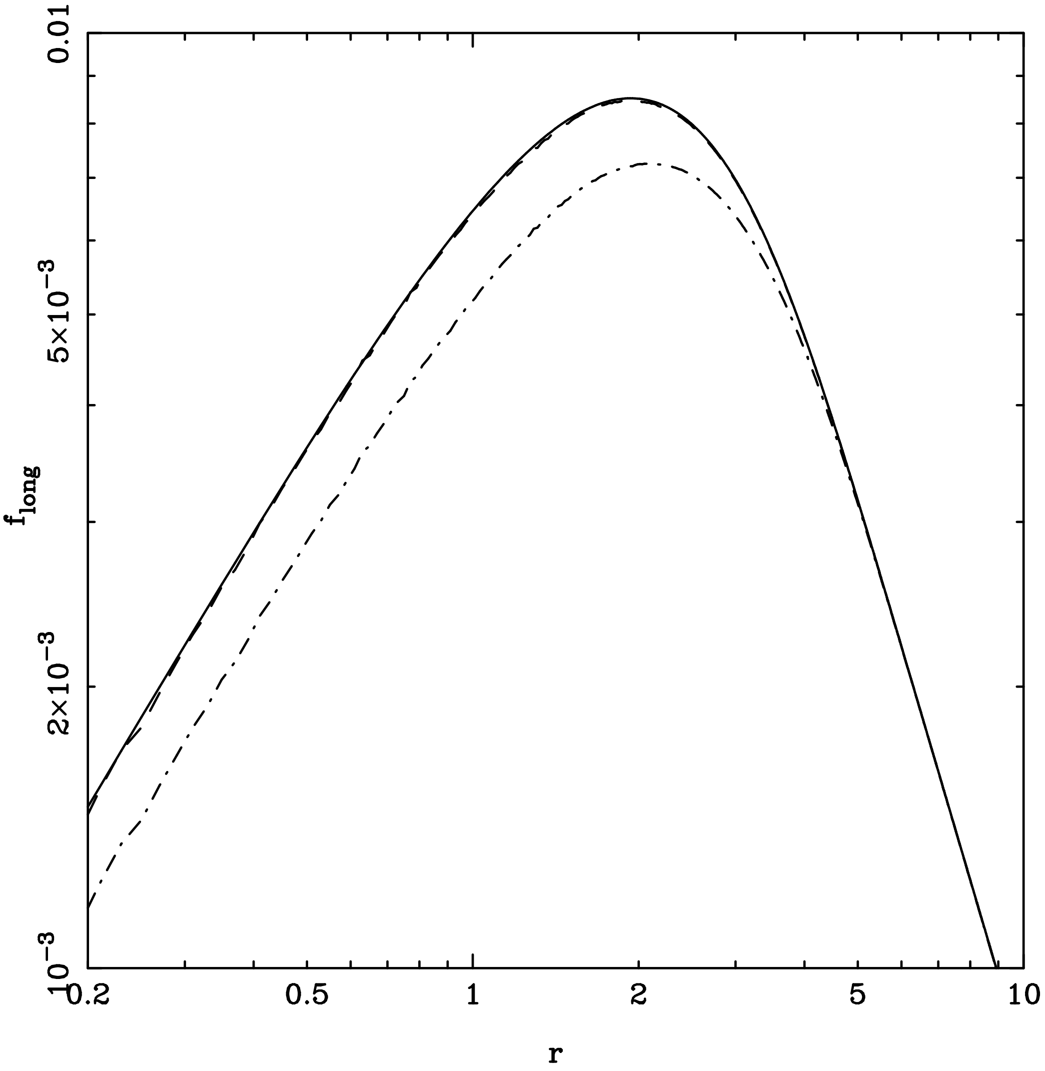

Figure 3.6: Long range force for a single particle is shown as a function of distance. The solid line shows the expected long range force (eqn.3.4), the dot-dashed curve the long range force obtained in the simulation and the dashed line the long range force obtained in the simulation after removing the effect of the interpolating function. This curve was drawn for $r_s = 1$. One can see that the difference between the expected force and that from the simulation decreases strikingly after deconvolution.

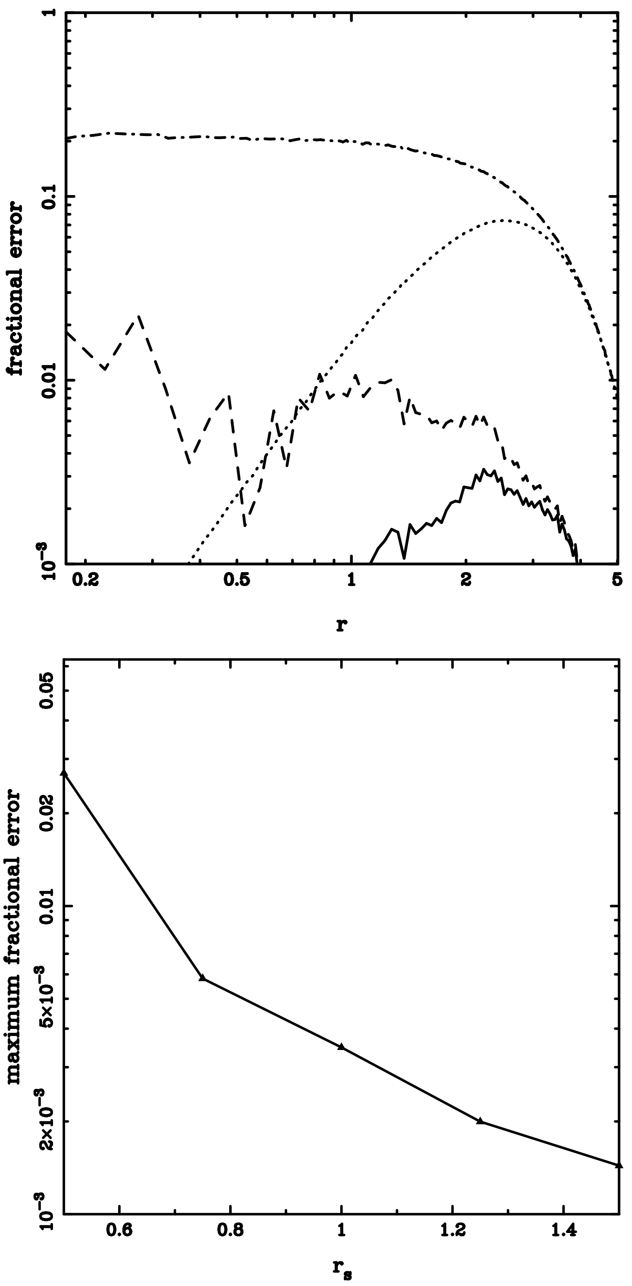

Figure 3.7: The upper panel shows the fractional error in the long range force due to a particle as a function of scale. Difference of the computed long range force from the expected long range force (eqn.3.4) is scaled by the expected long range force for the dot-dashed curve. The same difference is also scaled by the inverse square force and is shown by the dotted curve. The latter curve is more relevant for studying the contribution to the error in force. Curves showing the fractional error after removing the effect of interpolating function are also plotted here. The dashed curve shows the fractional error when scaled with the long range force and the solid curve shows the fractional error scaled with the inverse square force. For this case the peak fractional error when scaled by the inverse square force is less than 0.4% for $r_s = L$. The lower panel shows the peak fractional error as a function of r_s (in units of L).

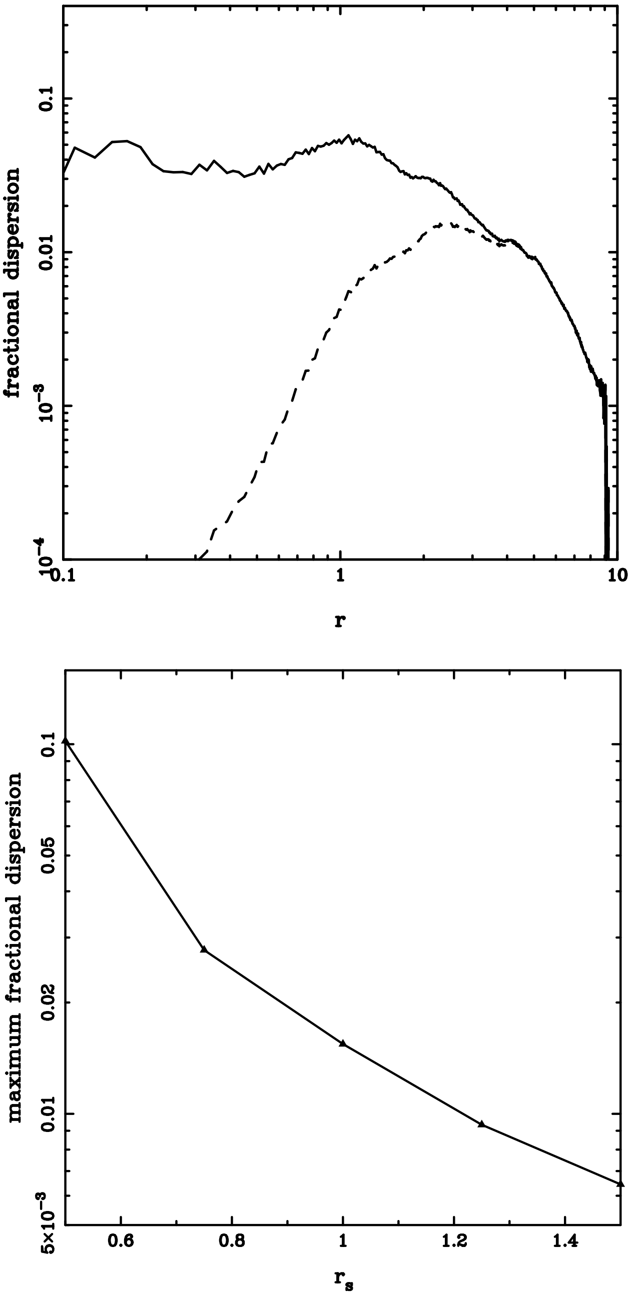

Figure 3.8: The upper panel shows the dispersion about the average long range force for $r_s = L$ with deconvolution of the interpolating window function. This dispersion scaled by the long range force is shown by the solid curve. Dispersion scaled by the inverse square force is shown by the dashed line. The peak dispersion in this case is just over 1.4% of the inverse square force. The lower panel shows the maximum value of dispersion scaled by the inverse square force as a function of r_s (in units of L).

force and the expected long range force (eqn.3.4) divided by the expected long range force in fig.3.7. We have also plotted the difference divided by the total force as this is more relevant for errors. We have plotted these for the average force as well as the average force obtained after deconvolution. The difference between the average long range force and the expected long range force (eqn.3.4) normalised by the total force drops by nearly one order of magnitude after deconvolution. Fractional error is below 0.4% for the entire range of scales. One can lower this further by using a larger r_s, or use a smaller r_s if larger errors are acceptable. Fig.3.7 also shows the maximum fractional error as a function of r_s. We use the long range force with deconvolution of the interpolating function W in the TreePM code.

The upper panel of fig.3.8 shows the dispersion in the long range force due to a single particle about the average value in a simulation. We have plotted *rms* dispersion about the average force as a function of r in the figure. We have normalised this with the long range force (eqn.3.4) and the total force. The latter curve is relevant for errors in the total force. The rms dispersion about the average long range force reaches a peak of about 1.4% near $r = 2r_s$. The dispersion decreases as we increase r_s. The lower panel of fig.3.8 shows the variation of the maximum *rms* dispersion, normalised by the inverse square force, as a function of r_s.

We can summarise our investigations of the errors in the long range force for a particle as follows : Maximum deviation of the average long range force from the theoretical expectation is around 0.4% and is much smaller at most scales. Dispersion about the average peaks at a value of 1.4% and is well below 1% for most scales. These values are for $r_s = L$ and the error is maximum around $2r_s$, errors fall off rapidly on both sides of $2r_s$. Errors decrease as we increase r_s.

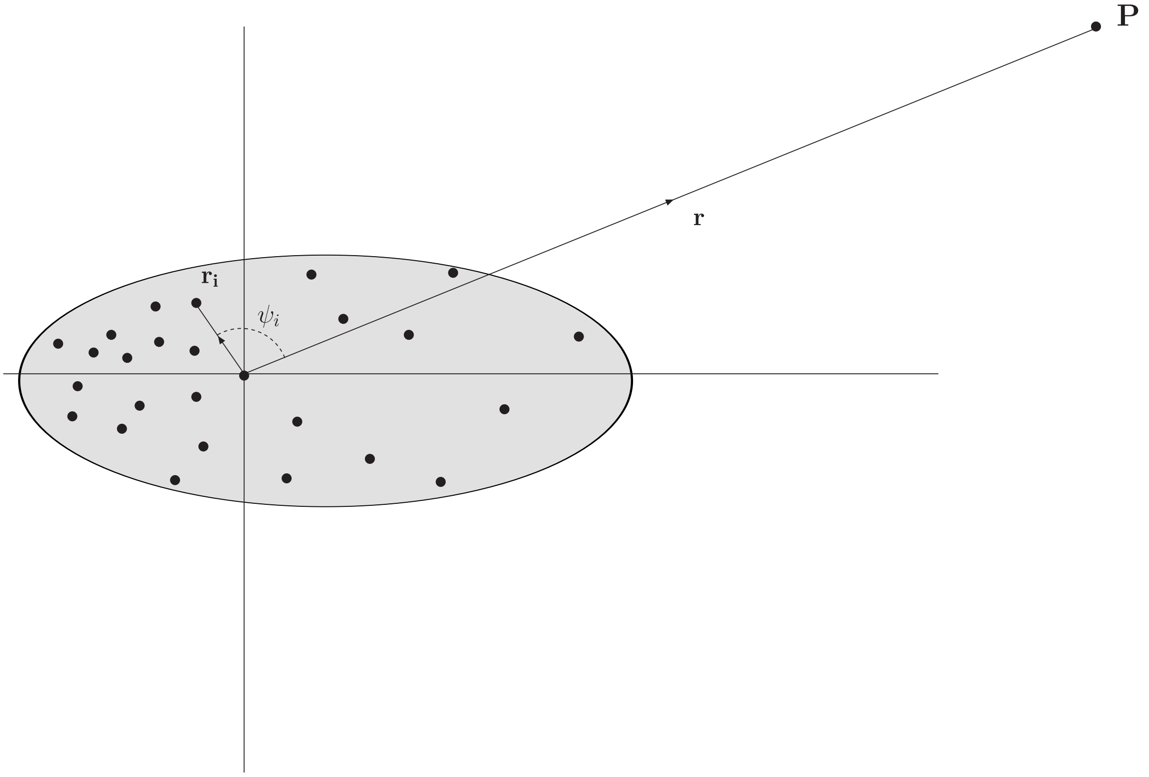

Figure 3.9: This shows a general distribution of point masses ($m = 1$) with position vectors $\mathbf{r}_i$ ($i = 1, n$). The centre of mass of the system is at the origin. We calculate the potential at a point, say P, with position vector $\mathbf{r}$ w.r.t the centre of mass. The size of the system is assumed to be small compared to $|\mathbf{r}|$.

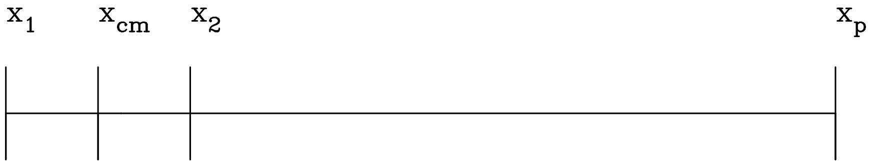

Figure 3.10: Schematic representation of a situation when two particles are at the diagonally opposite ends of a cell. Refer to text for details.

3.2.2 Errors in the Short Range Force

The tree approximation is used here for computing the short range force which differs from the usual inverse square force. We expect errors to be different in this case. Consider a general distribution of point masses ($m = 1$) with position vectors $\mathbf{r}_i$ ($i = 1, n$) and assume that the centre of mass of the system is at the origin. We calculate the potential at a point P with position vector $\mathbf{r}$ w.r.t the centre of mass. We assume that the size of the system is small compared to $|\mathbf{r}|$. Fig.3.9 is a simple schematic representation of such a system. Then

$$V_{tot} = -\sum_i \frac{1}{|\mathbf{r} - \mathbf{r}_i|} = -\sum_i \frac{1}{r\sqrt{1 + \left(\frac{r_i}{r}\right)^2 - 2\left(\frac{r_i}{r}\right)\cos\psi_i}} \tag{3.19}$$

where ψ_i is the angle between $\mathbf{r}_i$ and $\mathbf{r}$. Expanding this in a Taylor series and retaining terms upto order $(r_i/r)^2$,

$$V_{tot} = -\sum_i \frac{1}{r} + \frac{1}{2r}\sum_i \left(\frac{r_i}{r}\right)^2 \left[1 - 3\cos^2\psi_i\right] \tag{3.20}$$

The second term in the above expression is the quadrupole moment term and for a generic distribution of particles it is the leading order error term in the tree approximation as we use it here. It is apparent from the quadrupole term that in the worst case scenario all the particles lie along a straight line coincident with the direction of $\mathbf{r}$. The error is maximum in such a case.

We now estimate the error in the worst case scenario for two particles, i.e. when the particles are at the diagonally opposite ends of a cell and the point at which the force is being calculated is along the same diagonal. Distance between the particles is $d\sqrt{3}$, where

d is the size (one side) of the cell under consideration. We have represented the situation in a simple schematic form in fig.3.10. In the figure, we have two unit point masses at positions x_1 and x_2 separated by a distance 2ϵ such that $2\epsilon = d\sqrt{3}$. We will calculate the force at a point x_p at a distance r from the centre of mass x_{cm} of the system under the assumption that $\epsilon \ll r, r_s$. Using the expression for the potential in equation 3.20, we obtain

$$\frac{\delta}{f(r)} = \frac{3}{4}\left(\frac{2\epsilon}{r}\right)^2 = \frac{9}{4}\left(\frac{d}{r}\right)^2 \tag{3.21}$$

where

$$f(r) = \frac{2G}{r^2} \tag{3.22}$$

is the the total force from the centre of mass at the point x_p and δ is the error corresponding to the quadrupole moment term in the potential. $(d/r)^2$ in equation 3.21 is nothing but the parameter θ_c. Therefore we have

$$\frac{\delta}{f(r)} = \frac{9}{4}{\theta_c}^2 \tag{3.23}$$

The fractional error due to the quadrupole moment term in the tree force is a simple constant proportional to ${\theta_c}^2$. If the two particles are separated along an edge of the cell instead of the diagonal, the error is $3\theta_c^2/4$.

We now repeat this estimate for the short range force. The short range potential at any point $\mathbf{r}$ is

$$V_{sh} = -\sum_i \frac{1}{|\mathbf{r}-\mathbf{r}_i|}\text{erfc}\left(\frac{|\mathbf{r}-\mathbf{r}_i|}{2r_s}\right) \tag{3.24}$$

Again, retaining terms upto order $(r_i/r)^2$, we have

$$\begin{aligned} V_{sh} &= -\sum_i \frac{1}{r}\text{erfc}\left(\frac{r}{2r_s}\right) + \frac{1}{2r}\text{erfc}\left(\frac{r}{2r_s}\right)\sum_i\left(\frac{r_i}{r}\right)^2\left[1-3\cos^2\psi_i\right] \\ &- \frac{1}{\sqrt{\pi}r}\exp\left(-\frac{r^2}{{r_s}^2}\right)\sum_i\left(\frac{r_i}{r}\right)^2\left[\left(\frac{r}{r_s}\right)^3\cos^2\psi_i - \frac{r}{2r_s}\left(1-3\cos^2\psi_i\right)\right] \end{aligned} \tag{3.25}$$

The first two terms in the above expression correspond to terms in the potential for the total force. We consider, as before, the worst case scenario for the error in the short range force. The error is given by

$$\begin{aligned}\frac{\delta}{f(r)} &= \frac{1}{4}\left(\frac{2\epsilon}{r}\right)^2\left[3-\frac{6}{\sqrt{\pi}}\int_0^{r/2r_s}\exp\left(-t^2\right)dt+3\left(\frac{r}{r_s}\right)\exp\left(-\frac{r^2}{4r_s^2}\right)\right]\\ &+ \frac{1}{4}\left(\frac{2\epsilon}{r}\right)^2\left[\frac{2}{\sqrt{\pi}}\left(\frac{r}{r_s}\right)^3\exp\left(-\frac{r^2}{4r_s^2}\right)+\frac{1}{8\sqrt{\pi}}\left(\frac{r}{r_s}\right)^5\exp\left(-\frac{r^2}{4r_s^2}\right)\right] \qquad (3.26)\end{aligned}$$

where $f(r)$ is the total force and δ is the error corresponding to the quadrupole moment term in the short range potential. It follows that

$$\begin{aligned}\frac{\delta}{f(r)} &= \frac{3\theta_c^2}{4}\left[\text{erfc}\left(\frac{r}{2r_s}\right)+\left(\frac{r}{r_s}\right)\exp\left(-\frac{r^2}{4r_s^2}\right)\right]\\ &\quad +\frac{\theta_c^2}{4}\left[\frac{2}{\sqrt{\pi}}\left(\frac{r}{r_s}\right)^3\exp\left(-\frac{r^2}{4r_s^2}\right)+\frac{1}{8\sqrt{\pi}}\left(\frac{r}{r_s}\right)^5\exp\left(-\frac{r^2}{4r_s^2}\right)\right] \qquad (3.27)\end{aligned}$$

Therefore the quadrupole moment error term in the short range force is a complicated function of r and not a simple constant proportional to ${\theta_c}^2$ as it is for the total force. In this derivation we have assumed that the separation between the two particles is much smaller than r_s. Also the error here has been calculated for the case when the two particles are separated along an edge of the cell (and not along a diagonal) and these two points and the point where the force is being calculated are co-linear. We have plotted the error as a function of distance r in the upper panel of fig.3.11. We can see that the error in the short range force is higher than that for the total force for the same value of θ_c.

The next task is to do a more realistic estimate of the error for a random distribution of particles in a cell instead of considering pathological situations which give rise to large errors. We take a cubical cell of size d and distribute point masses randomly in the cell. Our aim is to calculate the error in the short range force in the tree approximation on a single particle due to the group of particles in the cubical cell described above. To calculate the error we need to compare the short range force in the tree approximation ($\mathbf{f}^s_{cm}$) with the sum of short range forces due to all the particles in the cell ($\mathbf{f}^s$). We also have the inverse square force in the tree approximation $\mathbf{f}_{cm}$. To quantify error in short range force, we define ϵ_s as a measure of the error :

$$\epsilon_s = \frac{|\mathbf{f}^s_{cm}-\mathbf{f}^s|}{|\mathbf{f}_{cm}|} \qquad (3.28)$$

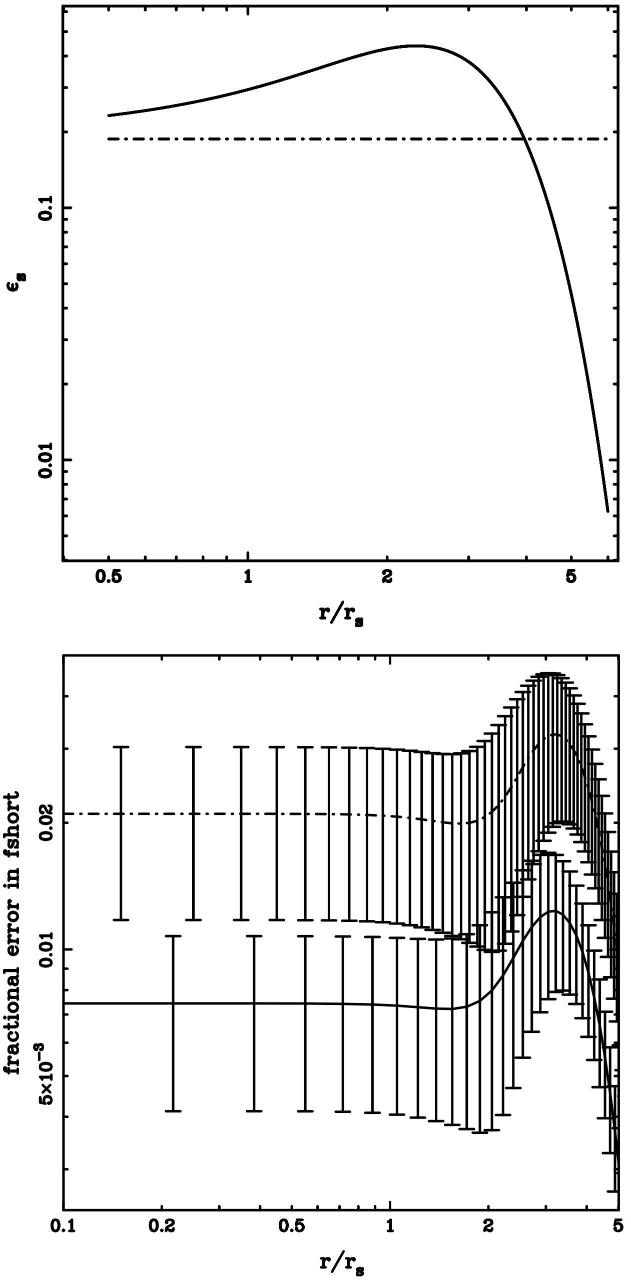

Figure 3.11: The fractional error in the tree approximation in the worst case is shown in the upper panel as a function of distance r for the short range force (solid curve). This is the error up to leading order in θ_c (see text). For reference, we have also plotted the corresponding error in the case of the inverse square force (dot-dashed curve). This error is computed for a situation when the two particles are separated along an edge of the cell and the two particles and the point where the force is being calculated are along a line. The lower panel shows the fractional error in the short range force measured using the method described in the text. Here the solid curve corresponds to $\theta_c = 0.3$ and $N_p = 30$ and the dot-dashed curve is for $\theta_c = 0.5$ and $N_p = 30$. Error bars mark 1σ variations about the average error measured for 100 different distributions of particles.

Lower panel of fig.3.11 shows the fractional error ϵ_s in the short range force as a function of distance r. The error is constant at small scales, increases around the scales where the contribution of short range force is decreasing rapidly, then plummets to very small values at larger scales as the short range force itself becomes very small. The range of scales where the error is large are those where the short range force contributes a significant fraction to the total force and *its variation with distance is significantly different from the inverse square force*. In the figure, the error is plotted for two values of θ_c. It is lower for smaller θ_c : error for $\theta_c = 0.3$ is a nearly a factor of three smaller than that for $\theta_c = 0.5$. Here the error has been averaged over 100 different distributions of $N_p = 30$ particles in the cell. The error is smaller for larger number of particles as chances of an extreme distribution of particles become smaller. Even for $N_p = 10$, the error is considerably smaller than the extreme collinear case discussed above. The error for $\theta_c = 0.5$ and $N_p = 30$ is about 2% - slightly larger than the maximum error due to the long range force (1.4% for $r_s = L$). Errors for $\theta_c = 0.3$ are smaller and closer to the errors in the long range force. However, we should add that these are errors due to particles in a single cell.

3.2.3 Errors in the TreePM Force

We turn to the question of errors in the TreePM force calculated by adding the short range and the long range forces. Errors are calculated with respect to a reference force computed with the following configuration of the TreePM code. We chose $\theta_c = 0.01$; hence the errors in short range force are negligible as these errors decrease in proportion to θ_c^2. We take $r_s = 4.0$ and the error in the long range force is thus below 0.1% if we extrapolate the maximum dispersion in the long range force. With this setup, we can safely assume that the error in the reference force (as compared to the inverse square force with periodic boundary conditions that we should use) is below 0.1% at all scales and this is sufficient for studying errors in the TreePM force calculated using more pragmatic values of r_s and θ_c where we aim to keep errors below the 1% level. We calculate errors for two distributions of particles, one a random distribution of particles with uniform density (an unclustered distribution) and the other a clustered distribution generated by an N-Body simulation. We plot the cumulative distribution for fractional error; errors are calculated for $2^{21} \simeq 2 \times 10^6$ particles.

We present the variation of errors with r_{cut} in fig.3.12. r_{cut} is the radius within which we sum the short range force around each particle. We need to sum out to a distance where the contribution of the short range force to the inverse square force drops to an ignorable fraction. From fig.3.2, we know that the short range force is less than 1% of the inverse square force at $5r_s$, so we expect that r_{cut} should be comparable to this value. For this figure we used $r_s = L$ and $\theta_c = 0.5$. We find that for the unclustered distribution, the distribution of errors is very different for $r_{cut} = 3r_s$ as compared to $r_{cut} \geq 4r_s$, whereas for all $r_{cut} \geq 4r_s$ the distribution of errors is very similar. This can result if particles beyond $4r_s$ are not contributing to the force. This is to be expected in a uniform distribution of particles where force due to local fluctuations dominates. The distribution of errors for the clustered distribution is shown in the lower panel of fig.3.12. For the clustered distribution, particles at larger distances contribute significantly to the force on each particle. As this contribution increases, the force from $r > 4r_s$ becomes more relevant. The error distribution for the clustered distribution shows some difference between $r_{cut} = 4r_s$ and $r_{cut} \geq 5r_s$. From these figures we conclude that $r_{cut} = 5r_s$ is a safe choice, but $r_{cut} = 4r_s$ may suffice for most situations.

Fig.3.13 shows the distribution of errors for different values of θ_c. As before, we have shown results for the unclustered distribution in the upper panel and the clustered distribution in the lower panel. We used $r_s = L$ and $r_{cut} = 6r_s$ for these plots. There is a marked difference in the errors for different values of θ_c in the case of the unclustered distribution. Error decreases with θ_c and then saturates as we get to smaller values of θ_c. The situation is different for the clustered distribution as errors for different values of θ_c are very similar. This suggests that the errors are dominated by the long range force calculation for a clustered distribution. For an unclustered distribution,the total force on each particle is small, whereas the force due to a cell with many particles is large and many large contributions cancel out to give a small net force. Numerical errors in adding and

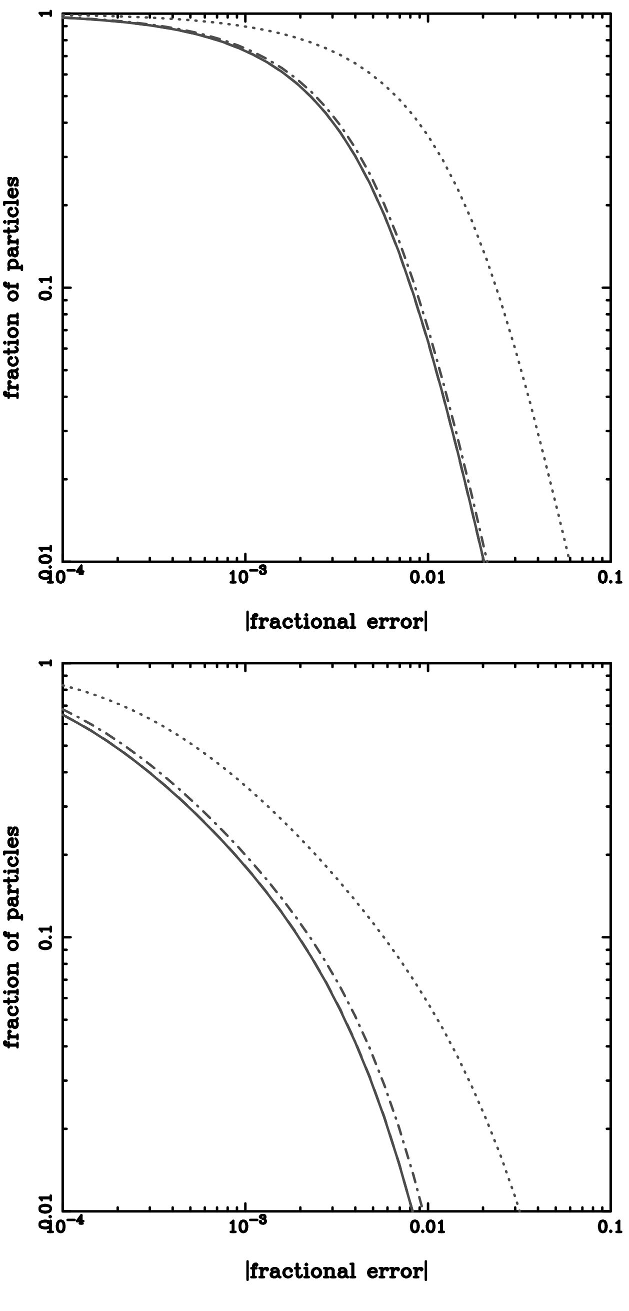

Figure 3.12: The distribution of fractional error in the TreePM force is shown here for two distributions of particles. The upper panel shows cumulative errors for a uniform distribution of particles for different choices of r_{cut}. The lower panel shows the same for a clustered distribution of particles. We used $r_s = L$ and $\theta_c = 0.5$ for these figures. The solid curve corresponds to $r_{cut} = 6r_s$, the dashed curve to $r_{cut} = 5r_s$ and the dot-dashed and dotted curves correspond to $r_{cut} = 4r_s$ and $3r_s$ respectively. The error is very large for both particle distributions for $r_{cut} = 3r_s$. The error for all $r_{cut} \geq 4r_s$ is essentially the same. There is some difference in the error for $r_{cut} = 4r_s$ and $r_{cut} \geq 5r_s$ for the clustered distribution as anisotropies in the particle distribution start to make the force due to particles at these distances more and more significant.

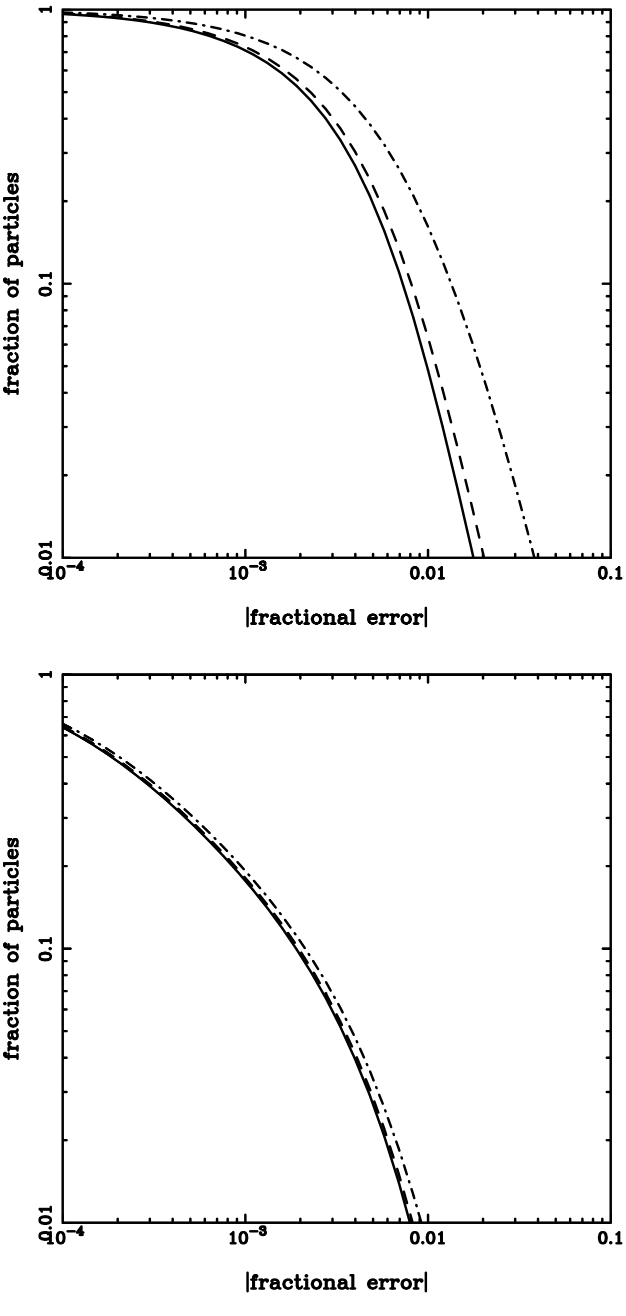

Figure 3.13: The distribution of fractional error in the TreePM force is shown for different values of θ_c. The upper panel shows these for a uniform distribution of particles and the lower panel shows the same for a clustered distribution of particles. We used $r_s = L$ and $r_{cut} = 6r_s$ for these figures. A full line, a dashed line and a dot-dashed line correspond to $\theta_c = 0.3, 0.5$ and $\theta_c = 0.7$ respectively in both panels. The order of curves in both the panels is as expected - error is smaller for smaller θ_c. The difference in errors for different values of θ_c reduces with increasing clustering.

subtracting large numbers seem to dominate errors in case of the uniform distribution. Contribution of cells is large for larger θ_c; hence we get a significant variation with θ_c for the unclustered distribution.

Lastly, we look at the variation in errors with r_s. Fig.3.14 shows the distribution of errors for different values of r_s. We fixed the values of the other parameters to $r_{cut} = 6r_s$ and $\theta_c = 0.5$. The upper panel of the figure is for the unclustered distribution and the lower panel is for a clustered distribution of particles. Errors for the clustered distribution increase monotonically as r_s is decreased as expected from our analysis of errors in the long range force. The fractional error in the TreePM force for 99% of the particles for $r_s = L$ is less than 0.8%. The variation of errors with r_s for the unclustered distribution is not monotonic. Errors decrease till r_s increases to unity and then increase again. This happens because the issue of the small force on each particle for a uniform distribution again comes into play. For larger r_s, the force due to individual cells is large and the numerical error in adding all the large contributions to get a small net force is also large. The fractional error in the TreePM force for 99% of the particles for $r_s = L$ is less than 1.5%; thus the variation in error from an unclustered to a clustered situation is small compared to that in tree codes. From this figure, we conclude that $r_s = L$ is the optimum choice for the TreePM code.

3.3 CPU Time Requirements

In this section we list the CPU time required for one timestep of the TreePM code for a simulation with 2^{11} particles. We also study the variation of the CPU time with the parameters of the TreePM method. The computer on which these timings were obtained is powered by a 1.6GHz Pentium 4 CPU. Programs were compiled using the Intel Fortran compiler (version 5.0) and double precision variables were used throughout. Programs are written in Fortran 90.

We used $r_s = L$, $\theta_c = 0.5$ and $r_{cut} = 5r_s$. In this case one timestep takes 740 sec-

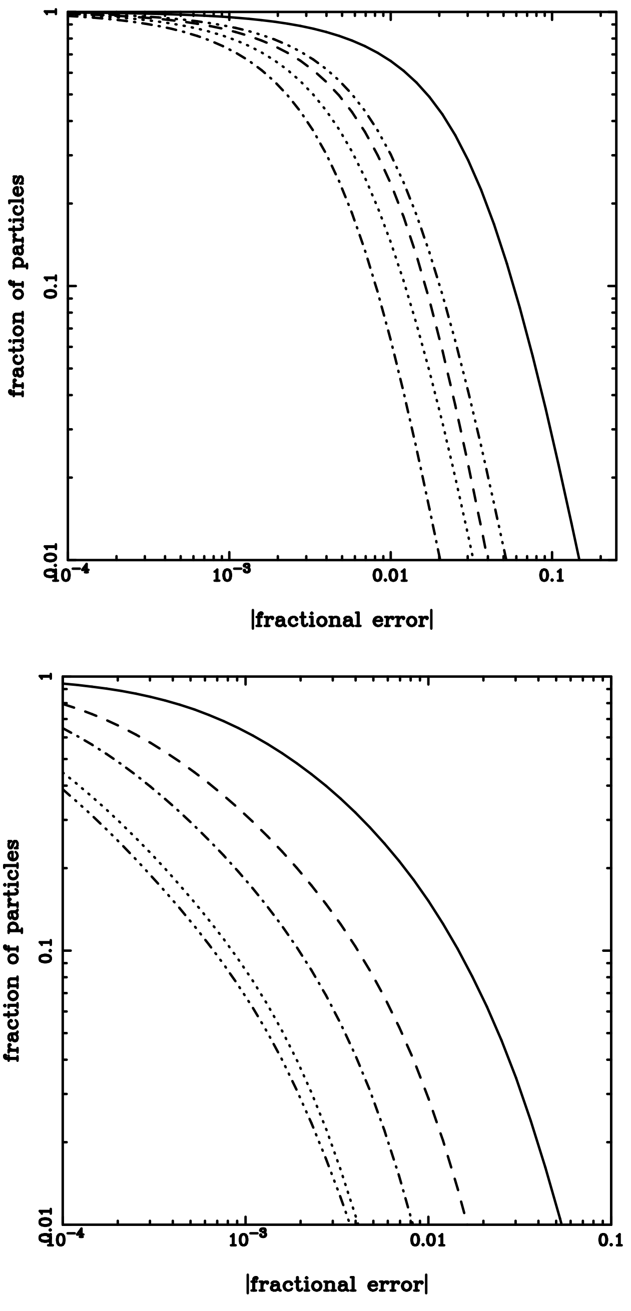

Figure 3.14: This figure shows the variation of errors with r_s. The upper panel shows the fractional error in the TreePM force for a uniform distribution of particles and the lower panel shows the same for a clustered distribution of particles. We used $\theta_c = 0.5$ and $r_{cut} = 6r_s$ for these figures. Different line styles are used for different values of r_s. The solid curve is for $r_s = 0.5L$, the dashed curve for $r_s = 0.75L$, the dot-dashed curve for $r_s = L$, the dotted curve for $r_s = 1.5L$ and the dot-dot-dashed curve for $r_s = 2L$. The error decreases monotonically with increasing r_s for the clustered distribution. The behaviour is different for the unclustered distribution where the error decreases at first, but then increases again. Refer to the text for an explanation.

onds. Time taken for building the tree, calculating the long range force, calculating the short range force and moving particles are, for one timestep, equal to 6.38 seconds, 26.70 seconds, 711.51 seconds and 0.25 seconds respectively. This implies that the time taken for each particle per step is 0.35ms. These numbers are for an unclustered (uniform) distribution of 128^3 particles.

The time required for one timestep of the TreePM code varies with the choice of the values of the parameters r_{cut}, r_s and θ_c. We expect the time taken to increase with increasing r_s and r_{cut} and decrease with increasing θ_c. The behaviour for these cases is shown in fig.3.15. It is seen that the time taken falls sharply as we increase θ_c, whereas errors are insensitive to the choice as long as $\theta_c \leq 0.5$. Therefore it makes sense to use $\theta_c = 0.5$. Time taken rises sharply with increasing r_{cut}. Again, by comparing with the variation of errors, we recommend $r_{cut} \approx 4.5 r_s$. Lastly, the time taken increases rapidly as we increase r_s. We recommend $r_s = L$ as the error is minimum for this choice for the unclustered distribution.

3.4 Scale Invariant Evolution of Power Law Spectra

In Chapter 2 we have seen that a powerful test for any N-Body method is the requirement that the evolution of clustering of power law models be scale-invariant. We have tested the TreePM code for power law models. Fig.3.16 shows $\bar{\xi}$ as a function of $r/r_{nl}(t)$ for several epochs. Here $r_{nl}(t)$ is the scale which is going non-linear at time t and it varies in proportion with $a^{2/(n+3)}$ in the Einstein-deSitter model. n is the index of the power spectrum. We have used $n = -1$ for the upper panel and $n = 1$ for the lower panel. We have plotted $\bar{\xi}$ at scales more than four times larger than the artificial softening length used in the simulation. It is obvious that the evolution is scale invariant for both the spectra. This implies that we can safely probe the highly non-linear regime in gravitational clustering with this code.

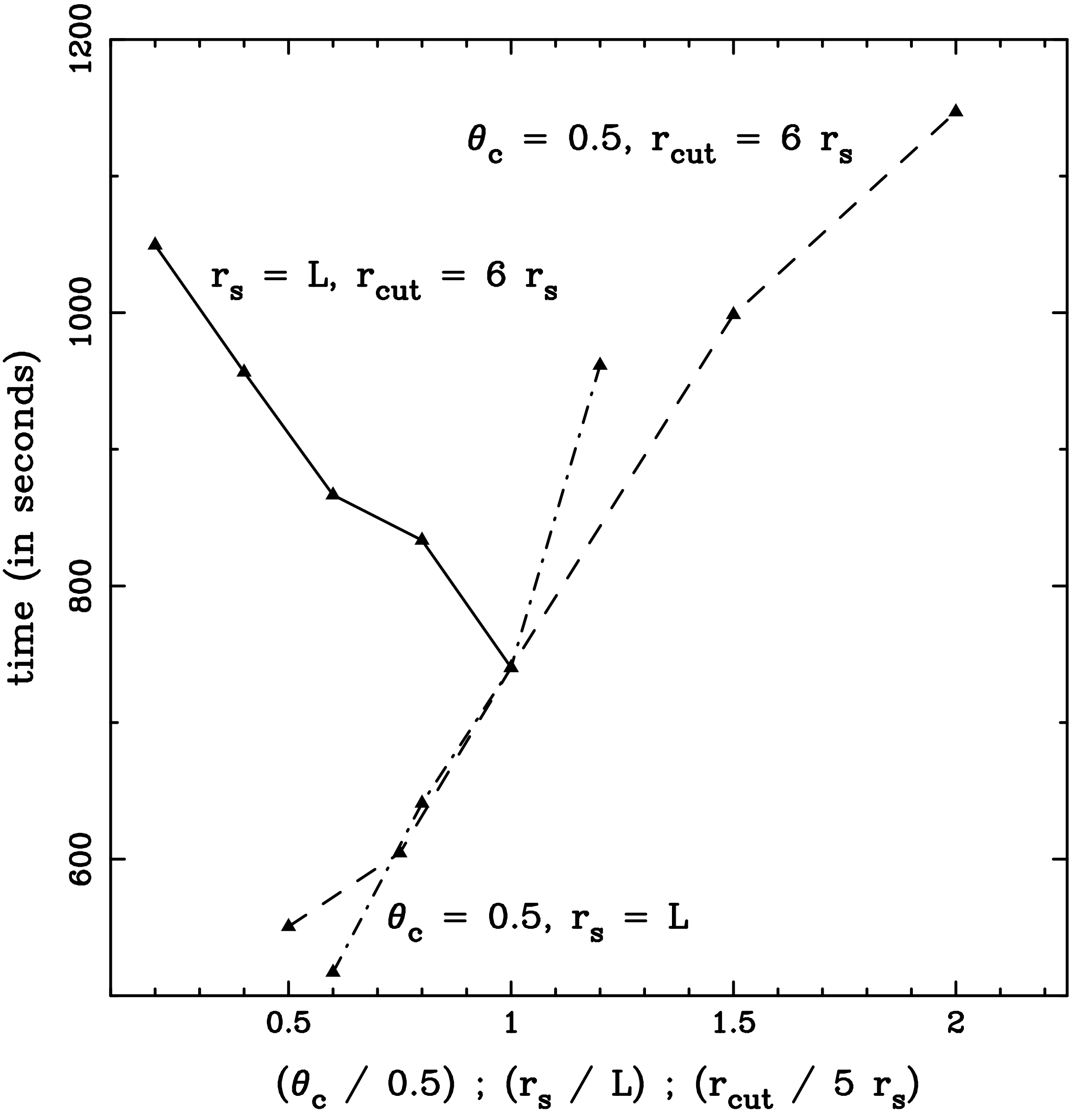

Figure 3.15: The variation of CPU time required per timestep is shown as a function of the values of the parameters in the TreePM code. The thick line shows the variation in CPU time with $\theta_c/0.5$. Points, from left to right, along this curve are for $\theta_c = 0.1, 0.2, 0.3, 0.4$ and 0.5. Values of the other parameters were fixed to $r_s = L$ and $r_{cut} = 6r_s$. The dashed line shows variation with respect to r_s/L at $r_{cut} = 6r_s$ and $\theta_c = 0.5$. The dot-dashed line shows variation with $r_{cut}/5r_s$ at $\theta_c = 0.5$ and $r_s = L$.

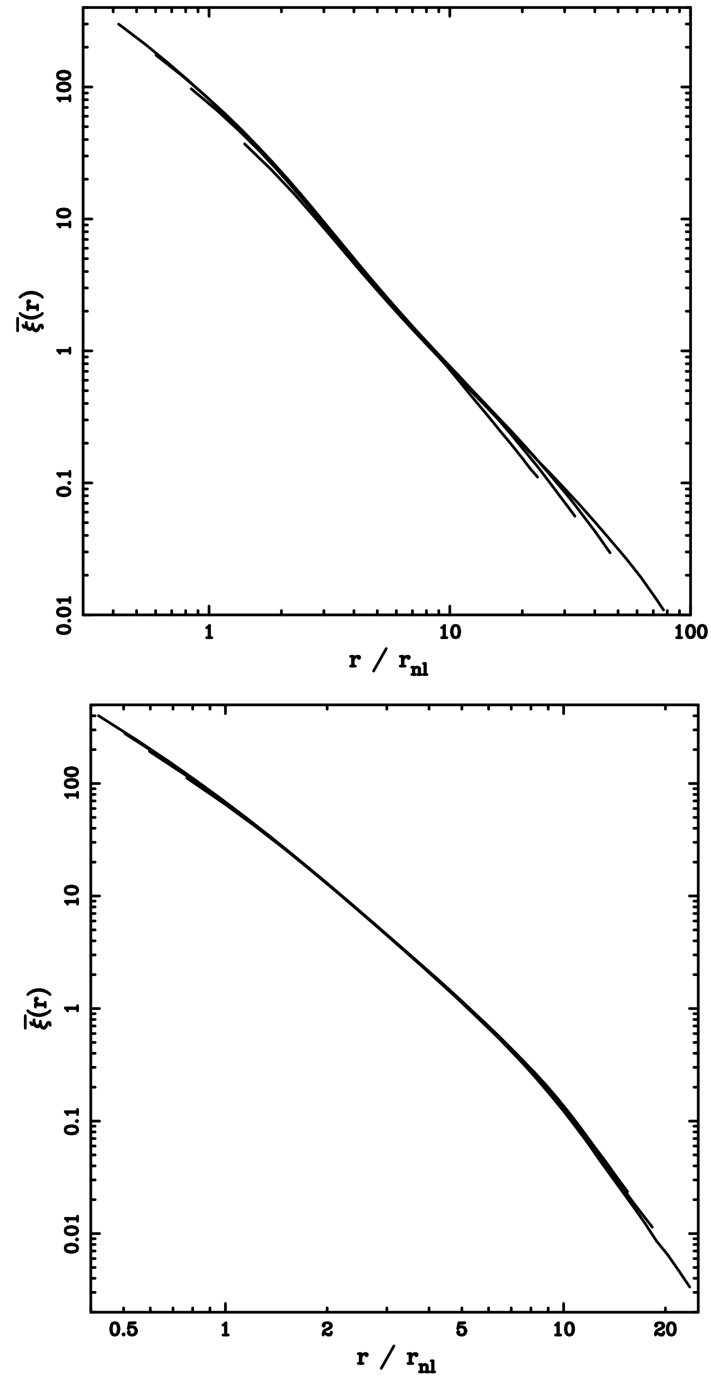

Figure 3.16: This figure shows $\bar{\xi}$ as a function of $r/r_{nl}(t)$ for several epochs. Here $r_{nl}(t)$ is the scale which is going non-linear at time t and it varies in proportion with $a^{2/(n+3)}$ in the Einstein-deSitter model. n is the index of the power spectrum. We have used $n = -1$ for the upper panel and $n = 1$ for the lower panel. We have plotted $\bar{\xi}$ at scales more than four times larger than the artificial softening length used in the simulation.

3.5 Comparison with P^3M and AP^3M Methods

The TreePM code retains the accuracy of the tree code while speeding up the force calculation by a factor of 4.5 or more. A code that provides comparable performance is the P^{3}M code (Efstathiou et al., 1985; Couchman, 1991). However, there are two main differences between the P^{3}M and the TreePM codes. One is that most P^{3}M codes use the natural cutoff provided by the grid for the long range force. This implies that the long range force falls very slowly at small scales. Thus anisotropies due to the mesh lead to anisotropies in force at scales comparable to the grid scale. In contrast, the TreePM method uses Ewald's method for effecting the separation between the long and the short range forces and anisotropies at the grid scale are reduced considerably as shown in §3.1 of this chapter. The second difference between the P^{3}M code and the TreePM code is that the small scale force is added for each pair of particles with separation smaller than some r_{cut}. This process is of order $O\left[N_p n r_{cut}^3\left(1+\bar{\xi}\left(r_{cut}\right)\right)\right]$, where N_p is the number of particles in the simulation, n is the number density of particles and $\bar{\xi}(r_{cut}) = 3J_3(r_{cut})/r_{cut}^3$. At early times this reduces to $O(Nnr_{cut}^3)$, but at late times, when the density field has become highly non-linear [$\bar{\xi}(r_{cut}) \gg 1$], it becomes $O\left[Nnr_{cut}^3\bar{\xi}\left(r_{cut}\right)\right]$. Thus, as the density field becomes more and more non-linear, the number of operations required for computing the short range force increases rapidly. The number of operations required for adding the short range correction using the tree method varies much more slowly being of $O\left[N\log\left(nr_{cut}^3\left(1+\bar{\xi}\left(r_{cut}\right)\right)\right)\right]$. The linear and the non-linear limits of this expression are $O\left[N\log\left(nr_{cut}^3\right)\right]$ and $O\left[N\log\left(nr_{cut}^3\bar{\xi}\left(r_{cut}\right)\right)\right]$ respectively. Thus the variation in the number of operations with increase in clustering is much less for a TreePM code compared to a P^{3}M code. The problem is not as severe for the Adaptive P^{3}M code (Couchman, 1991), but it still persists. Therefore the TreePM code has a significant advantage over the P^{3}M and AP^{3}M codes for simulations of models where $\bar{\xi}(r_{cut})$ is very large (Bagla, 2002). In turn, P^{3}M codes have one significant advantage over the TreePM code as these codes require much less memory.

3.6 Discussion and Summary

In this chapter, we have presented a detailed study of performance characteristics of the TreePM method. We have analysed different sources of error and developed workarounds for the main sources of error. We have shown that the choice of filter for splitting the inverse square force into short and long range components suggested in Bagla (2002) is the best among options studied here. We have also shown that the error in the long range component of the force contributes very little to the total error in force. Errors introduced by the tree approximation for the short range force are different from those for the inverse square force and these errors dominate the total error in force. We have calculated the distribution of errors in force for clustered and unclustered particle distributions. The analysis of errors in realistic situations shows that the TreePM method performs well. We have also tested the code by simulating a few power law models and checking for scale invariance.

This method compares favourably with other comparable methods such as implementations of the tree code like GADGET (Springel, Yoshida and White, 2001) and hybrid methods such as the TPM (Xu, 1995; Bode, Ostriker and Xu, 2000). From the numbers available in these papers, we find that the errors for the recommended configuration of the TreePM method are comparable with those in GADGET and lower than those in the TPM method. In terms of CPU time taken per step per particle, we again find that the numbers are comparable. Of course, it is not possible to make a detailed comparison of this quantity as the whole approach is different, e.g. we do not use multiple timesteps whereas GADGET relies heavily on these to optimise on speed. On the other hand, GADGET and TreePM have a uniform force resolution, whereas TPM does not and hence the time taken is likely to vary more with the amplitude of clustering for the TPM code as compared to the other two.

In summary, we can state that the TreePM method is a competitive method for doing cosmological N-Body simulations. The TreePM method, with only three well-defined parameters, has a cleaner mathematical model compared with other methods. Explicit use of the three parameters gives users control over errors and CPU time requirements.

Chapter 4

A Parallel TreePM Code

An inherent parallelism in all N-Body codes is that forces on particles can be calculated concurrently. Hybrid nature of the TreePM method brings in multiple parallelisms and we have to adopt a somewhat involved scheme for parallelisation. The tree method is used in the TreePM code to calculate the short range force. Computation of the short range force is the dominant component in terms of the fraction of CPU time used. We make use of the scheme commonly used for parallelising a tree code to compute the short range force. The long range force is computed concurrently on a processor not involved in the computation of the short range force. We use the Message Passing Interface (MPI) (Snir et al., 1999) for parallel computing. We discuss our parallel implementation of the TreePM method in the following sections of this chapter[1].

4.1 Scheme of Parallelisation

In a Barnes and Hut (1986) tree code, the simulation volume is divided into cells and only a small subset of the details of the particle distribution in distant cells is needed for computing the force. Thus it is natural to divide the simulation volume into domains with equal computational load and the forces on particles in a given domain can be

[1] This chapter is based on *A Parallel Implementation of the TreePM Code* (S. Ray and J.S. Bagla), astro-ph/0405220.

computed by one processor. We compute the short range force in the TreePM code by this method. In order to form domains, the simulation volume is bisected recursively. Bisections may or may not be along orthogonal directions and each bisection is carried out in such a way that the computational load is equal on both sides of the cut (Salmon, 1991; Dubinski, 1996; Springel, Yoshida and White, 2001; Dubinski, 2004). After m bisections, the simulation volume is divided into 2^m domains - all with equal computational load. These are now assigned to different processors (we call these processors the grid nodes) and the calculation of force can be carried out concurrently on these processors. A separate communicator is set up for processors involved in the short range force calculation, i.e. the grid nodes. This communicator is given the Cartesian grid topology (Snir et al., 1999) for easy bookkeeping and the grid nodes consequently get labelled by "coordinates". Fig.4.1 schematically represents the process of domain docomposition and assignment of coordinates to processors within a Cartesian grid. For ease of presentation we have used a two-dimensional example here. In the figure, the two panels at the top demonstrate recursive bisections of the simulation volume along the x-axis. The lower left panel in the figure shows a further bisection of each of the domains so created along an orthogonal direction, i.e. y. We end up with $2^3 = 8$ domains with equal computational load, each of which is assigned to a grid node with a unique set of Cartesian coordinates, represented by a pair of numbers in brackets in the lower right panel of the figure. We call the node with coordinates $(0, 0)$ the origin node.

The process of domain decomposition adds some overhead, but it is small compared to the gain due to parallelisation, i.e. to the time saved by using multiple processors for force calculation. Of course, this overhead increases as we increase the number of processors for domain decomposition. For long range forces like gravity, each processor needs information from all the other processors and hence the number of communications required is significant. This can be a serious impediment on distributed memory machines with a large number of processors. The problem is less serious for the TreePM code as the short range force calculation requires communications with a much smaller number of processors.

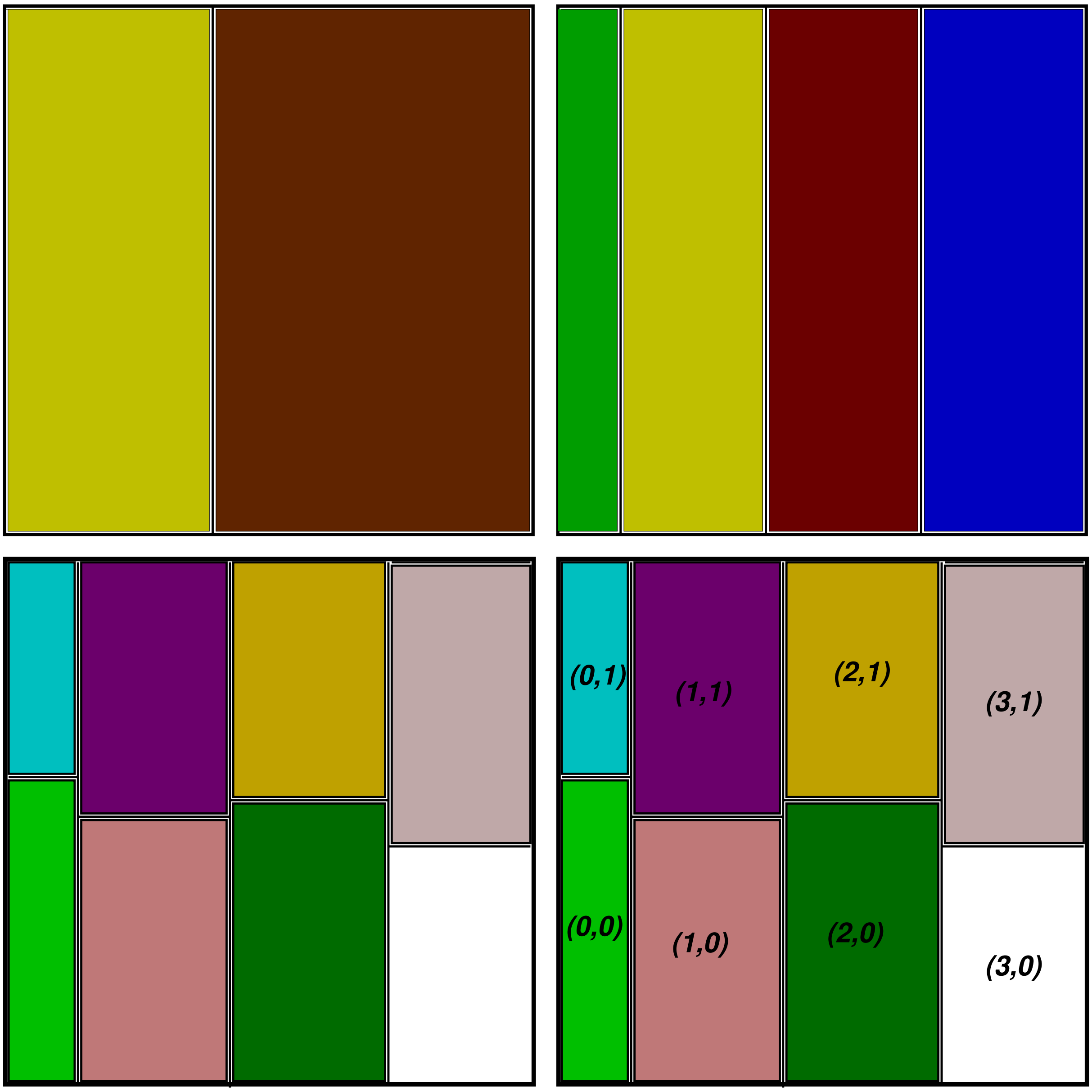

Figure 4.1: A schematic representation of domain decomposition in two dimensions. The numbers in brackets represent the coordinates of processors on the Cartesian grid created by the Message Passing Interface (MPI).

In the TreePM method, the force computation is divided into two parts, the long range and the short range forces, and these two components can be evaluated independently. This is a parallelism inherent in the TreePM algorithm. The method described above serves to compute the short range force, exploiting another implicit parallelism. We need to use the two parallelisms in the algorithm for a successful implementation of the parallel TreePM code. However the presence of two independent parallelisms makes the task of load balancing somewhat non-trivial and gives rise to complexities discussed below.

Only a small fraction of the CPU time is used for computing the long range force in the sequential version of the code. Thus the number of processors used for the long range force calculation can be much smaller than the total number of processors being used. In fact, only one processor (say, the PM node) for the long range force calculation is sufficient for most cases. An obvious problem that arises is that load balancing is achieved only for a specific number of processors when the time taken for the long range and short range forces coincides and load balancing is less than perfect for a smaller number of processors. If the number of processors used for computing the short range force is smaller than a critical number then the long range force calculation will finish well before the short range force calculation. Given that the long range force computation takes much less time as compared to the calculation of the short range force, only the processor(s) involved in the long range force calculation are made to wait in this situation. As this is much smaller than the number of processors used for the short range force calculation, load imbalance does not degrade efficiency significantly. If, however, the number of processors is larger than the number required for optimum load balancing, then the processors doing the short range force calculation will have to wait. The situation can be remedied by spreading the task of long range force calculation over more than one processor at the cost of adding to the total number of processors (N_{proc})[2].

We now proceed to describe the detailed algorithm that we have adopted and summarise various options that we have considered at each step.

[2] We will use N_{proc} to denote the total number of processors and N_{short} for the number of processors used to compute the short range force.

4.1.1 Short Range Force

We use domain decomposition for computing the short range force and recursive bisection to create domains with equal number of particles (Salmon, 1991; Dubinski, 1996). As long as the number of particles in each domain is sufficiently large, we find that dividing the simulation volume into domains with equal number of particles is sufficient for load balancing and we need not explicitly create domains with equal computational load.

The short range force needs to be taken into account over a radius r_{cut} around every particle. Therefore each domain needs information about particles upto a distance r_{cut} from its boundary walls. Communicating information about particles in neighbouring domains for completing the calculation of the short range force is a tricky problem. The direct approach, which also ensures load balancing, is to request the processors corresponding to neighbouring domains for the relevant information (Salmon, 1991; Dubinski, 1996). A variant of this method is to send positions of particles near the boundary of domains and seek the partial force information, i.e. the processor for a neighbouring domain receives a list of particles and calculates the short range force on these particles due to particles in its domain and returns the partial force. The problem with these approaches is that communication overheads are significant. For a three-dimensional simulation, where each domain is larger than r_{cut}, communications are restricted to nearest neighbours amongst domains. The number of nearest neighbours is never greater than 26, but two-point communications with 26 processors per processor can add a significant amount of overhead. There is an overhead in starting communications as even high performance networks have a latency of 5μs or more and, of course, some time is taken for transmission of data as well. It is not possible to manage several concurrent communications and hence all the exchanges of information between particles have to be staggered. There is also a computational overhead as lists of relevant particles at tree nodes are generated and information from all the neighbours is put together, or partial force is calculated in the alternative approach mentioned above. However, this is the only choice available when the number of processors is large or when the memory available per processor is limited and a single processor cannot handle the entire information about the positions of all the particles. When this is not the case, pairwise communications are not the optimal solution. Some alternatives that we have tried are :

1. List *non-local* particles along with recursive bisection. Thus the list of non-local particles needed for calculating the short range force is made at the same time as domain decomposition.

2. Send positions of all particles to all processors; each processor identifies the list of non-local particles that are needed.

The first option adds to the time before the calculation of the short range force can commence. It nearly doubles the time take for domain decomposition, though there is no additional overhead beyond this step. We find that the second option listed here is the faster of the two, but adds large overheads in terms of memory requirements for each processor because it requires that we be able to store all the particle positions in each processor. As long as memory is not a limitation, this is the best option and we choose this for our implementation. The overhead that can be attributed to different options is subject to the nature of the interconnect used.

In our scheme, each grid node shortlists non-local particles by identifying all particles within a distance r_{cut} of its domain boundaries. This is demonstrated by a simple schematic diagram in fig.4.2. Here the simulation volume is taken to be a square for convenience. The domain under consideration is shaded in dark grey, while the region surrounding it, shaded in light grey, is the area to be scanned to identify non-local particles that contribute to the short range force calculation for particles in the domain. The width of the region containing non-local particles is r_{cut}. The edges of the region in light grey, in principle, ought to arcs of circles, but we have used sharp edges for ease of computation.

While constructing the tree on each node, we use the full simulation volume as the

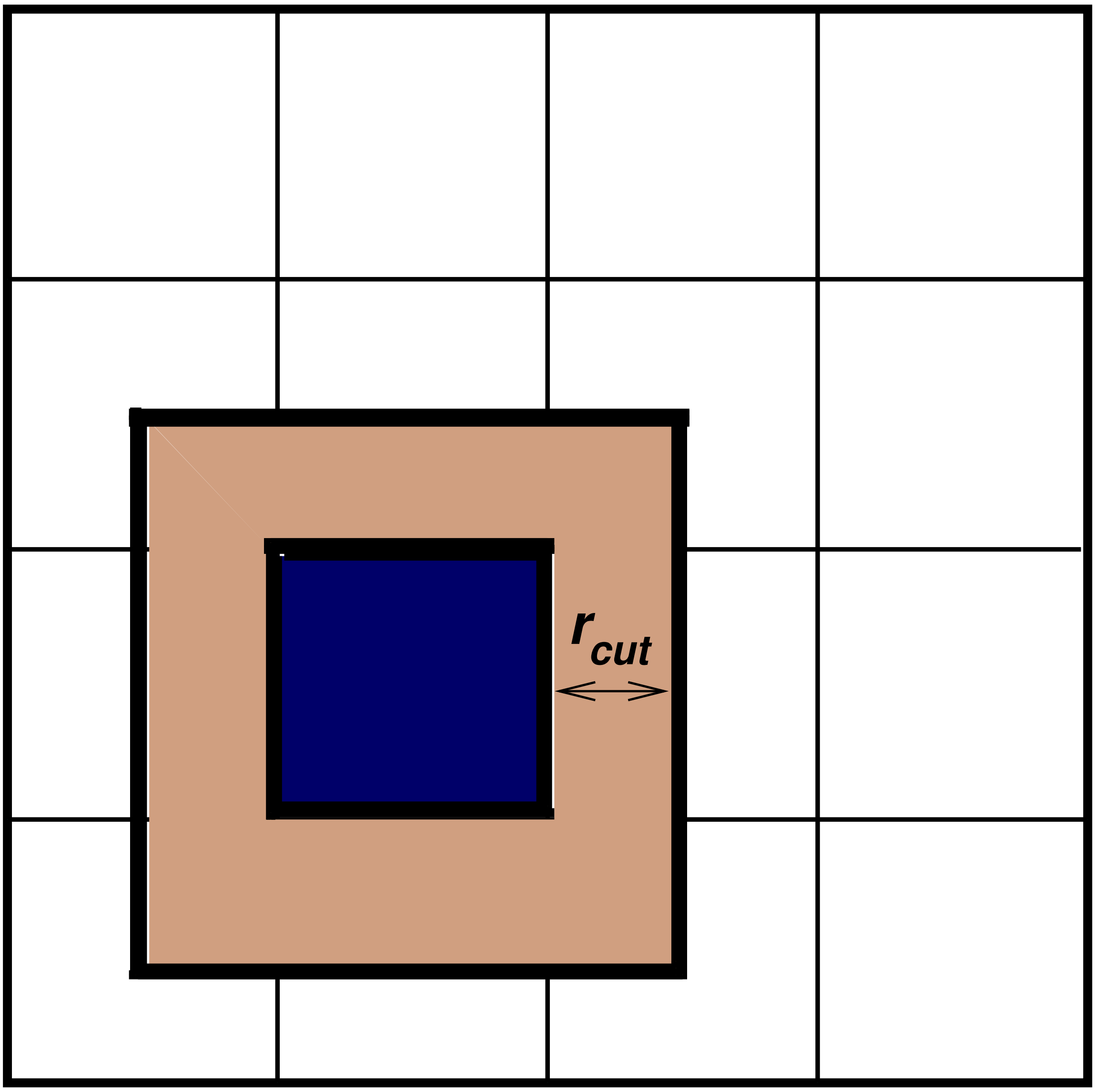

Figure 4.2: This figure shows a simulation area (two dimensions). The area is subdivided into domains. For a particular domain (shaded in dark grey), the area to be scanned, to identify the non-local particles that will contribute to the short range force calculation in the domain, is shaded in light grey.

largest cell. Periodic boundary conditions are applied for calculating the force and also to wrap particle positions after every timestep.

4.1.2 Long Range Force

Long range force is calculated in Fourier space using the PM method but using a different kernel (see eqn.3.1). We use FFTW (http://www.fftw.org) for computing Fourier transforms in this calculation. The force is computed on a single processor and is then communicated to all the other nodes using a *broadcast* (Snir et al., 1999). We achieve good load balancing between calculation of the short range and the long range forces when the short range force computation is done on 64 processors for a simulation with 128^3 particles and the long range force is calculated on a single processor. In case we wish to use a larger number of CPUs, we will have to use multiple processors for the long range force calculation as well in order to get load balancing. For testing our code, we have used the 84-processor Linux cluster *Kabir* (http://cluster.mri.ernet.in) and hence we can use atmost 64 processors for the short range force calculation.

4.1.3 Communications

At the start of each timestep, particle positions are *broadcast* by the origin node (coordinates $x = 0$, $y = 0$ and $z = 0$ on the Cartesian grid) to all the processors on the Cartesian grid as well as to the processor which computes the long range force. The origin node then initiates the process of domain decomposition and the particles are grouped into smaller and smaller units until they are passed on to processors where the short range force is to be computed. As domain decomposition proceeds along a tree structure, several processors can contribute concurrently to the process. At the outset, the origin node performs a bisection of the simulation volume along the x-axis. Several communicators are constructed in order to optimise communications for concurrent message passing between distinct subsets of nodes along independent channels of communication. In this, the communication overhead is reduced significantly. We'll describe the communicators in some detail in the following paragraphs.

We first define *Comm_x* which serves as a communicator dedicated only to communications of the origin node with nodes which have coordinates $y = 0$, $z = 0$. During the process of bisection, the origin node implicitly assigns nodes, including itself, in *Comm_x* to the domains it creates within the simulation volume in the first step of domain decomposition. All nodes which are members of *Comm_x* receive the entire set of particle positions together with the corresponding velocities as well as coordinates of the points of bisection along the x-axis of the simulation volume from the origin via a series of *MPI_Broadcast*s. These nodes then create their own local arrays of particle positions and velocities using the corresponding bisection point information. The remaining portions of the original position and velocity arrays are discarded.

At the next step, each node which is a member of *Comm_x* creates a communicator *Comm_y*(x_i) which channelises communication with all nodes which have y-coordinates which range over the entire set of possible y values on the Cartesian grid as well as $z = 0$ and $x = x_i$, where x_i is its own x-coordinate. Each of these nodes performs a bisection along the y-axis followed by a series of *MPI_Broadcast*s (similar to that by the origin node) to communicate its local arrays of particle positions and velocities as well as the coordinates of the points of bisection along the y-axis to all the nodes in *Comm_y*. Nodes in *Comm_y* similarly create their own local arrays of particle positions and velocities using the corresponding bisection point information, discarding the remaining portions of the available position and velocity arrays.

Finally, each node in *Comm_y* creates a communicator *Comm_z*(x_i,y_j) which facilitates communication with all nodes which have z-coordinates ranging over the entire set of possible z values on the Cartesian grid as well as $x = x_i$ and $y = y_j$, where x_i and y_j are its own x and y coordinates respectively. This is followed by a bisection along the z-axis and finally each node in *Comm_z*(x_i,y_j) has its own local arrays for position and velocity.

At the end of the process of domain decomposition, every node on the Cartesian grid therefore has information about particle positions and velocities in its own local domain. In addition, it has the full position array of particles, which information it uses to shortlist particles that are not local to its domain, but are needed to complete the short range force calculation. This is actually done by identifying the set of particles within a perpendicular distance r_{cut} of the boundary of the domain that is local to the node. The complete array for particle positions is discarded as soon as non-local particles needed to complete the short range force calculation are shortlisted. It now remains the task of the node to independently compute the tree part of the force on all particles in its domain as well as to move the same particles using the equation of motion.

The PM node *broadcasts* the entire force array at the end of its job. All particles in a domain carry identity tags so that we can trace their trajectories in the simulation. Using the identity tags, each grid node retains the force only for particles within the local domain, adds this to the short range force computed locally, moves the particles and recomputes their velocities. The irrelevant part of the array for the long range force is discarded.

Particle positions and velocities are finally sent back and the origin node *gathers* the arrays. *MPI_Reduce* is used within the Cartesian grid to communicate particle identities to the origin node.

4.2 A Flowchart for the Parallel Algorithm

The parallel algorithm in detail is as follows :

```
Initialisation :

  * PM node reads/generates initial positions, velocities and masses
    of all particles.
  * PM node BROADCASTs particle positions.
  * PM node SENDs particle velocities to the origin node.
  * Cartesian grid of processors is set up for domain decomposition.

Loop :

  * Origin node BROADCASTs full position array to all grid nodes
    and PM node.
  * Origin node initiates domain decomposition.
  * All grid nodes participate in domain decomposition and at the
    end of domain decomposition have own local position and velocity
    arrays.
  * All grid nodes shortlist particles not local to their own domain,
    but are needed to complete the short range force calculation.
  * Grid nodes form tree structure for local particles as well as
    non-local particles.
  * Grid nodes compute short range force for local particles.
  * PM node computes long range force for all particles concurrently.
  * Grid nodes receive long range force via BROADCAST by PM node.
  * Grid nodes compute total force and move local particles.
  * Origin node GATHERs particle positions and velocities from all
    grid nodes.

End Loop
```

In a single timestep of execution of the parallel TreePM code, the following contribute significantly to the CPU time :

- Short range force calculation at each grid node.
- Long range force calculation at the PM node.

- Broadcast of long range force to every grid node.
- Domain decomposition.
- Broadcast of particle positions to every node.
- Shortlisting of non-local particles required for short range force calculation at every grid node.
- Collecting particle positions, velocities and identity tags at the origin node.

The time taken for the long range force calculation is independent of the number of processors used for the short range force computation. All the other tasks listed above vary with the number of processors for a simulation with a given number of particles. The time taken for broadcast of the long range force and that for broadcast of particle positions to every node are expected to be similar, i.e. time taken by these should vary in the same manner with the number of processors. The time taken for these increases rapidly with the number of processors, but this remains the least important communication overhead. In fig.4.3, we have plotted time taken in seconds for the cases itemised above as a function of N_{short} for a simulation with 128^3 particles. The work for domain decomposition can be done concurrently by many processors in later bisections and the computational overhead is dominated by the first few bisections. So the time taken for domain decomposition varies slowly with N_{short} which means that the overhead due to this does not change significantly as we vary the number of processors. The time taken for the short range force calculation decreases rapidly as we increase the number of processors. This variation is faster than $1/N_{short}$ and can be understood as follows. Each processor handles a smaller and smaller number of particles and their velocities as the load is divided amongst larger and larger number of domains and this leads to an improved cache performance. The dot-dot-dashed line shows the time taken for short range force calculation, domain decomposition and shortlisting of non-local particles. This is the total time taken by any grid node for computing in each step of the simulation. The dashed line shows the time taken for the long range force calculation. Comparing these two, we see that we get good load balancing for the maximum number of processors used, i.e $N_{short} = 64$. For a smaller number of processors, the long range force calculation is finished earlier and the processor assigned to this job stays idle for some time.

4.3 Performance of the Parallel Code

Performance of parallel programs is measured in terms of speedup, where speedup is defined as the time taken to run the program on a single processor divided by the time taken to run the same program on N_{proc} processors.

$$S_{N_{proc}} = \frac{T_{seq}}{T_{N_{proc}}} \tag{4.1}$$

For a fully parallelisable problem, this should scale as N_{proc}. However, in problems where load balancing is not perfect and inter-process communication or computational overhead due to parallelisation is significant, speedup is less than N_{proc}. Our aim is to optimise our algorithm to make speedup as close to N_{proc} as possible, especially for a reasonably large value of N_{proc}. The speedup efficiency is the speedup divided by N_{proc}, i.e. $E_{N_{proc}} \equiv (S_{N_{proc}}/N_{proc}) \times 100\%$.

If we use a single processor for the long range force calculation while changing N_{short}, the number of processors computing the short range force, then speedup is not linear in N_{proc}. For small N_{short}, the long range force calculation takes much less time compared to the short range force calculation and the efficiency of parallelisation is low due to poor load balancing. As N_{short} is increased, efficiency of parallelisation improves till load balancing is achieved. In the regime where N_{short} is smaller than the optimum value for load balancing, the code speeds up faster than linear. If N_{short} is larger than the number required for load balancing with the long range force calculation, the performance does not improve any further. In this regime, communication overheads and/or long range force calculation take(s) up more time than short range force calculation and there is no significant change in speedup with increasing N_{short}. The optimum value of N_{short} depends on the size of the simulation and the details of how communications are organ-

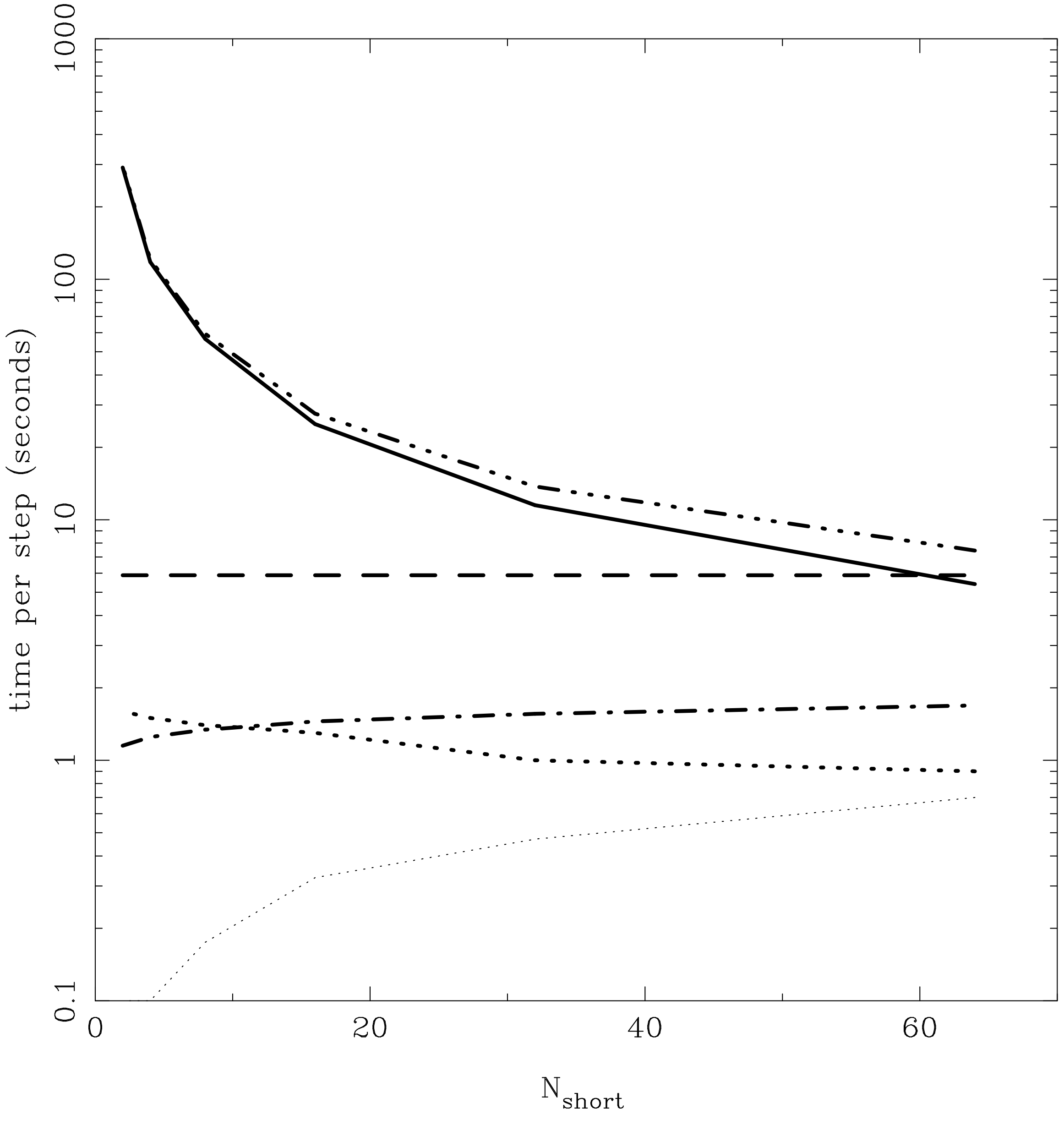

Figure 4.3: This figure shows the time taken in seconds for short range force calculation (full line), long range force calculation (dashed line), domain decomposition (dot-dashed line), shortlisting of non-local particles (dotted line) and communicating the long range force (thin dotted line) as a function of the number of processors for short range force calculation in one timestep of a parallel TreePM simulation with 128^3 particles. The dot-dot-dashed line is the sum of the times taken for short range force calculation, domain decomposition and shortlisting of non-local particles.

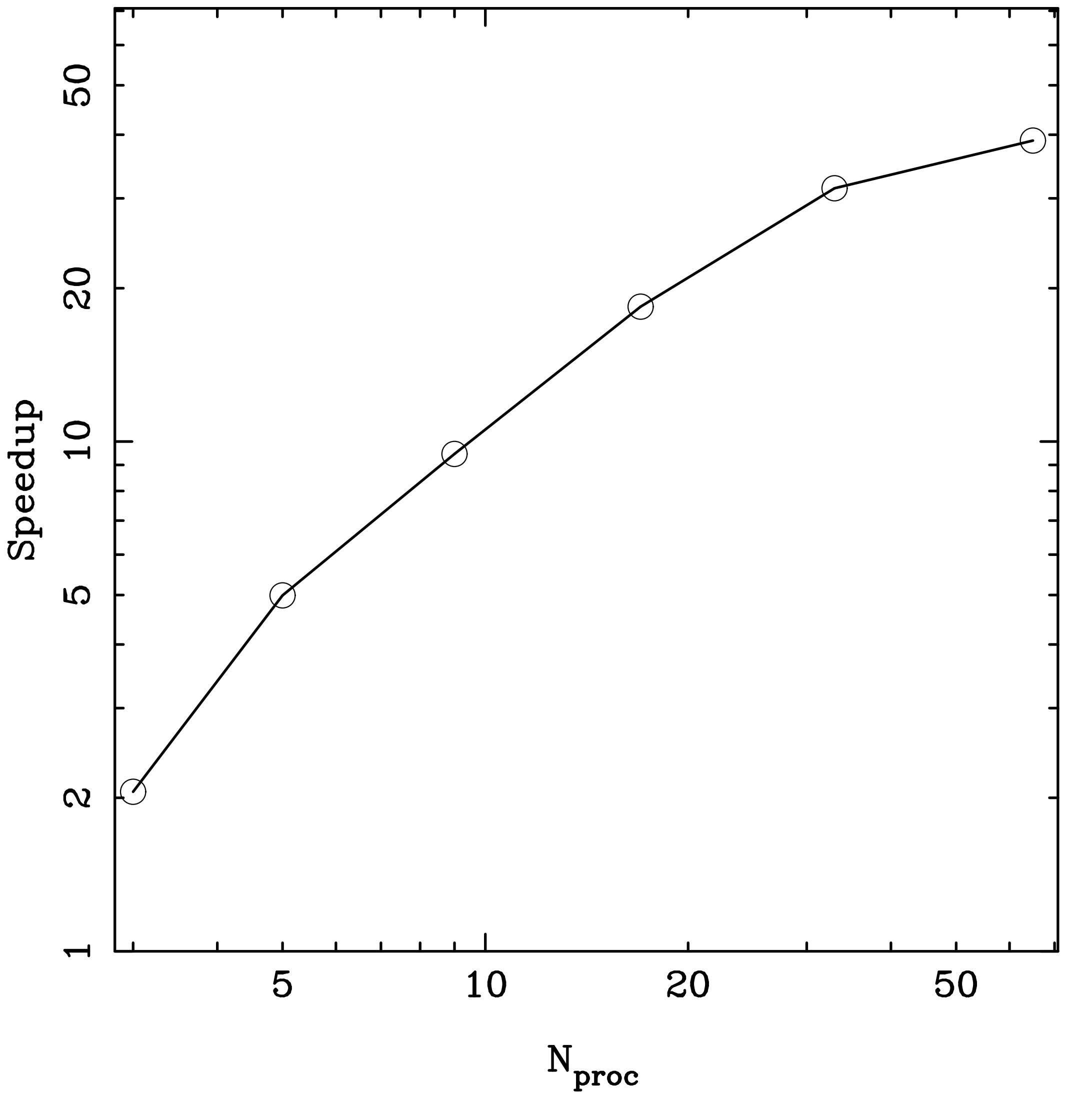

Figure 4.4: The speedup is plotted as a function of N_{proc} for 128^3 simulations (circles). Features expected from the analysis of the algorithm are clearly seen here with the efficiency dropping off at both small and large N_{proc}. At large N_{proc}, the speedup begins to saturate and for small N_{proc} the speedup decreases very rapidly.

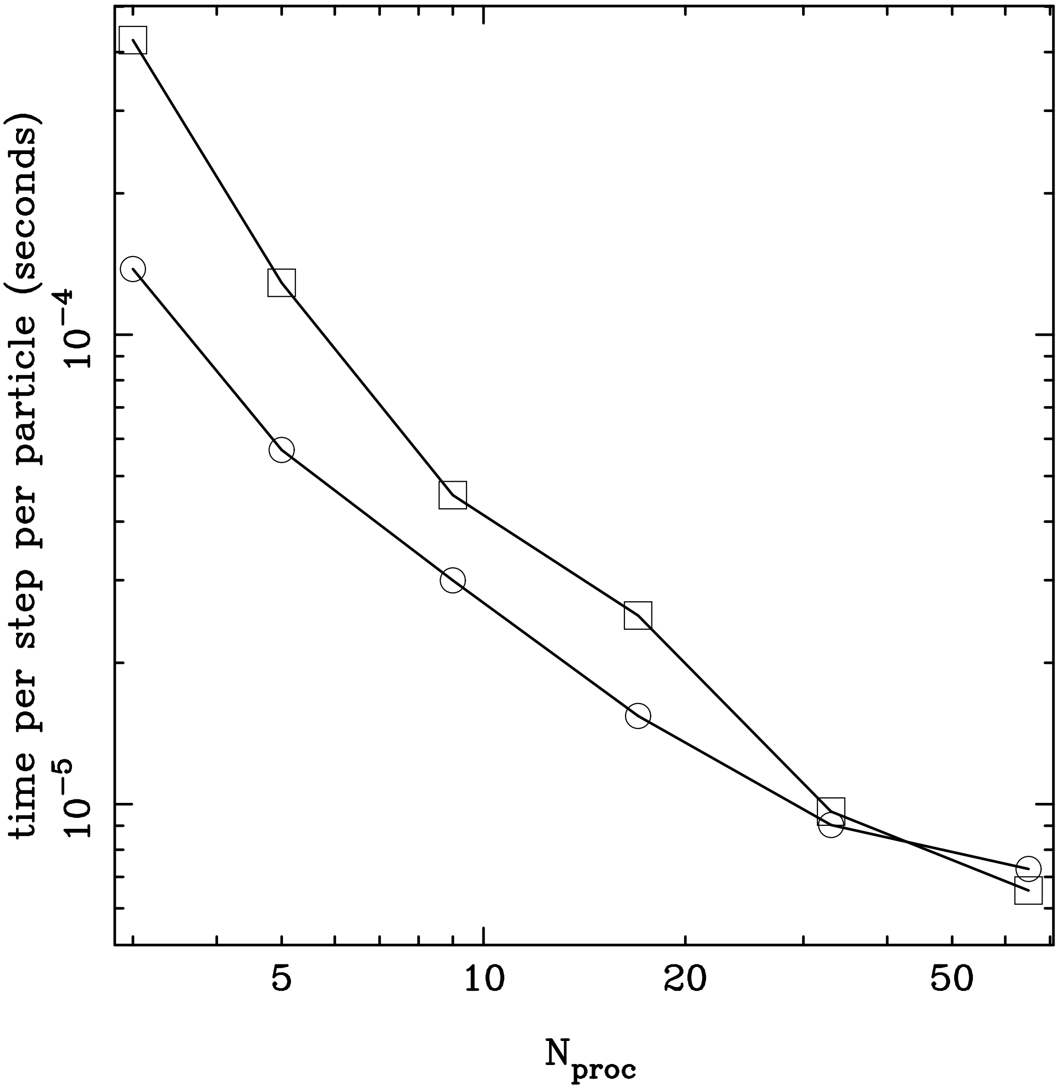

Figure 4.5: The time taken per step per particle is plotted as a function of N_{short} for simulations with 128^3 (circles) and 256^3 (squares) particles. We have used $N_{long} = 1$; thus $N_{proc} = N_{short} + 1$. A 256^3 simulation requires about 6.5μs per particle per timestep for $N_{proc} = 65$.

ised. These features can be seen in fig.4.4 where the speedup is plotted as a function of N_{proc} for 128^3 simulations. The speedup is almost linear beyond $N_{proc} = 5$ for simulations with 128^3 particles and it starts dropping beyond $N_{proc} = 33$ and the speedup efficiency falls below unity. Data for this figure was obtained on the Linux cluster *Kabir* (http://cluster.mri.ernet.in). Each node of *Kabir* is a dual processor Xeon server (2.4GHz, FSB 533MHz) with 2GB or more RAM. These nodes are connected in 2D torus topology with Dolphin-SCI (scalable coherent interconnect) network (http://www.dolphinics.com). Scali's ScaMPI (http://www.scali.com) is used for parallel computing with the Intel Fortran 90 compiler (http://www.intel.org). We obtain a speedup of 31.4 on 33 processors and 39 on 65 processors for the 128^3 simulations. Speedup is greater than the number of processors for $N_{proc} = 9$ and 17. This can be explained in terms of improved cache performance for smaller data sizes and also due to improvement in load balancing between the short range and the long range force computations.

Performance of the parallel code is presented in fig.4.5 where we have plotted time taken per step per particle as a function of N_{proc}. Notice that for $N_{proc} = 65$, the time taken per step per particle is only 6.5μs for 256^3 simulations. Thus we can do a simulation of 4000 timesteps in five days.

We can make further improvements to our method by using more processors for the long range force calculation and by using a larger box for computing the long range force as this will reduce the communication overhead. These changes, however, will be needed for a larger number of processors that we have access to and are not very relevant for parallelisation of the code on a small Linux cluster.

4.4 Discussion

In this chapter, we have presented an algorithm for parallelising the TreePM code on a Beowulf cluster. We have used both functional and domain decompositions. Functional decomposition has been used to separate the computation of long range and short range forces as well as the task of coordinating communications between different components. Short range force calculation is time consuming and is shown to benefit from the use of domain decomposition. We have tested the code on our Linux cluster by comparing the final positions and velocities of particles in some test cases with the output of the sequential code. The error profile of this code is identical to that for the sequential TreePM code (Bagla and Ray, 2003).

We have optimised the CPU time required at the cost of memory requirements. The maximum memory requirement per node is about 80 bytes per particle for double precision variables. We require up to 160 MB per node for the 128^3 simulations and 1.25 GB per node for the 256^3 simulations. These numbers represent the maximum memory requirements and for much of the time the memory requirement is much smaller than this. Memory requirements can be reduced by about 25% by reorganising the code. Such changes are useful mainly for a larger number of processors than used here.

For the 128^3 simulations we have obtained a speedup of 31.4 on 33 processors and 39 on 65 processors. The time taken for one timestep per particle is 6.5μs for a 256^3 particle simulation on 65 processors. Thus a simulation that runs for 4000 timesteps will take 5 days on this cluster. These results are for a simulation with a global timestep and further optimisations in terms of individual timesteps can be carried out.

Chapter 5

The TreePM Method for Two-Dimensional Cosmological Simulations

We know that structures such as galaxies and clusters of galaxies are very overdense compared to the average density of the Universe and it is believed that these were formed by the growth of small perturbations via gravitational instability (Peebles, 1980; Peacock, 1998; Bernardeau et al., 2002; Padmanabhan, 2002a). We also know that equations that describe the growth of density perturbations due to gravitational clustering (Peebles, 1974, 1980) cannot be solved analytically when the overdensities are large and we have to rely on N-Body simulations for detailed predictions. Limited information about non-linear gravitational clustering can be obtained by using approximations or ansatze, e.g. stable clustering (Peebles, 1980). Stable clustering allows us to relate the initial spectrum of fluctuations with the final spectrum of fluctuations (Davis and Peebles, 1977). Generic features in the evolution of gravitational clustering can also be understood using *scaling relations*. Scaling relations are a detailed prescription for relating the initial and final spectra of fluctuations (Hamilton et al., 1991; Nityananda and Padmanabhan, 1994; Peacock and Dodds, 1994; Jain, Mo and White, 1995; Peacock and Dodds, 1996; Padmanabhan, 1996; Padmanabhan et al., 1996; Bagla and Padmanabhan, 1997b; Bagla, Engineer and Padmanabhan, 1998; Smith et al., 2003). The motivation for studying non-linear scaling relations is to understand the key phases in gravitational clustering and to identify the relevant process in each phase (Padmanabhan, 1996). The existence of the scaling relations also implies that gravitational clustering does not erase the memory of initial conditions. Another useful and a more prosaic motivation for studying the scaling relations is to be able to invert the observed galaxy clustering and recover the primordial power spectrum (Hamilton et al., 1991). The scaling relations are valid for hierarchical models where the initial conditions contain fluctuations at all scales and the amplitude of fluctuations increases monotonically as we go from large scales to small scales.

Scaling relations indicate that there are three prominent regimes in the evolution of gravitational clustering. When the amplitude of perturbations is small such that the density contrast δ is very small ($|\delta| \ll 1$), mode coupling is not important and the evolution closely follows the predictions of linear perturbation theory (Peebles, 1974). The power spectrum and correlation function evolve without a change in shape in the *linear regime*. As density contrast grows and becomes comparable to unity, motions induced by gravitational collapse start to dominate over the expansion of the Universe. The *quasi-linear regime* is dominated by infall onto density peaks and the correlation function grows rapidly in this phase (Padmanabhan, 1996). Even later, gravitational collapse leads to formation of structures in or close to dynamical equilibrium and further evolution of the density contrast is dominated by the depletion of the average density of the Universe due to expansion of the Universe as the density of collapsed structures remains almost constant. This is the *asymptotic* or *highly non-linear regime*.

It is obvious that a large dynamic range is required in any N-Body simulation in order to investigate the scaling relations over a range of scales such as would cover all three regimes described above. It is difficult to achieve such a large dynamic range in simultions in three dimensions, particularly for negative values of the index n of the initial power spectrum. Fig.5.1 illustrates this fact. We have used the HKLM fitting function (Hamilton et al., 1991) from Smith et al. (2003) for computing the dimensionless power spectrum $\Delta^2(k)$, where k is the wavenumber of a given Fourier mode. In the figure, we have plotted Δ^2 at $k = k_{nyq}$ as a function of index n. Here k_{nyq} is the Nyquist frequency and n is the index of the initial power spectrum for scale-free models. For every n, $\Delta^2(k_{nyq})$ is plotted at an epoch such that $\Delta^2(k_f) \simeq 0.01$, where $k_f = 2\pi/N_{box}$ is the fundamental mode for a simulation box of size N_{box}. We have taken $N_{box} = 256$. It is obvious from the plot that for $n = -2.0$, one can barely reach the quasi-linear regime even at the smallest scales accessible (corresponding to k_{nyq}) in a simulation if fluctuations are to remain linear at the box scale (corresponding to k_f) if we take a box of size 256.

By simulating a two-dimensional system a much higher dynamic range can be achieved with similar computational resources (Beacom et al., 1991; Bagla, Engineer and Padmanabhan, 1998; Munshi and Coles, 1998). For example, since $2048^2 \approx 160^3$, the computational requirements are similar for a two-dimensional simulation with a boxsize of 2048 and a three-dimensional simulation with a boxsize of 160. This implies that we can achieve a dynamic range in 2d which is ten times larger at the same computational cost. This allows us to probe non-linearities over a larger range of scales in two dimensions as compared to three dimensions. As long as one is interested in generic aspects of clustering that are independent of dimension, studies in two dimensions have this advantage over ones in three dimensions. Higher dynamic range is thus the basic motivation for studying gravitational clustering in two dimensions. We describe the two-dimensional TreePM code in this chapter[1].

5.1 The Gravitational Force in Two Dimensions

When we go from three to two dimensions, we have, in principle, two different ways of modelling the system (Bagla, Engineer and Padmanabhan, 1998) :

- We can consider perturbations restricted to a plane in a three-dimensional expanding Universe. The force between particles varies as $1/r^2$ and we assume that all

[1] This chapter is based on *TreePM Method for Two-Dimensional Cosmological Simulations* (S. Ray), astro-ph/0406009. Accepted for publication in JAA.

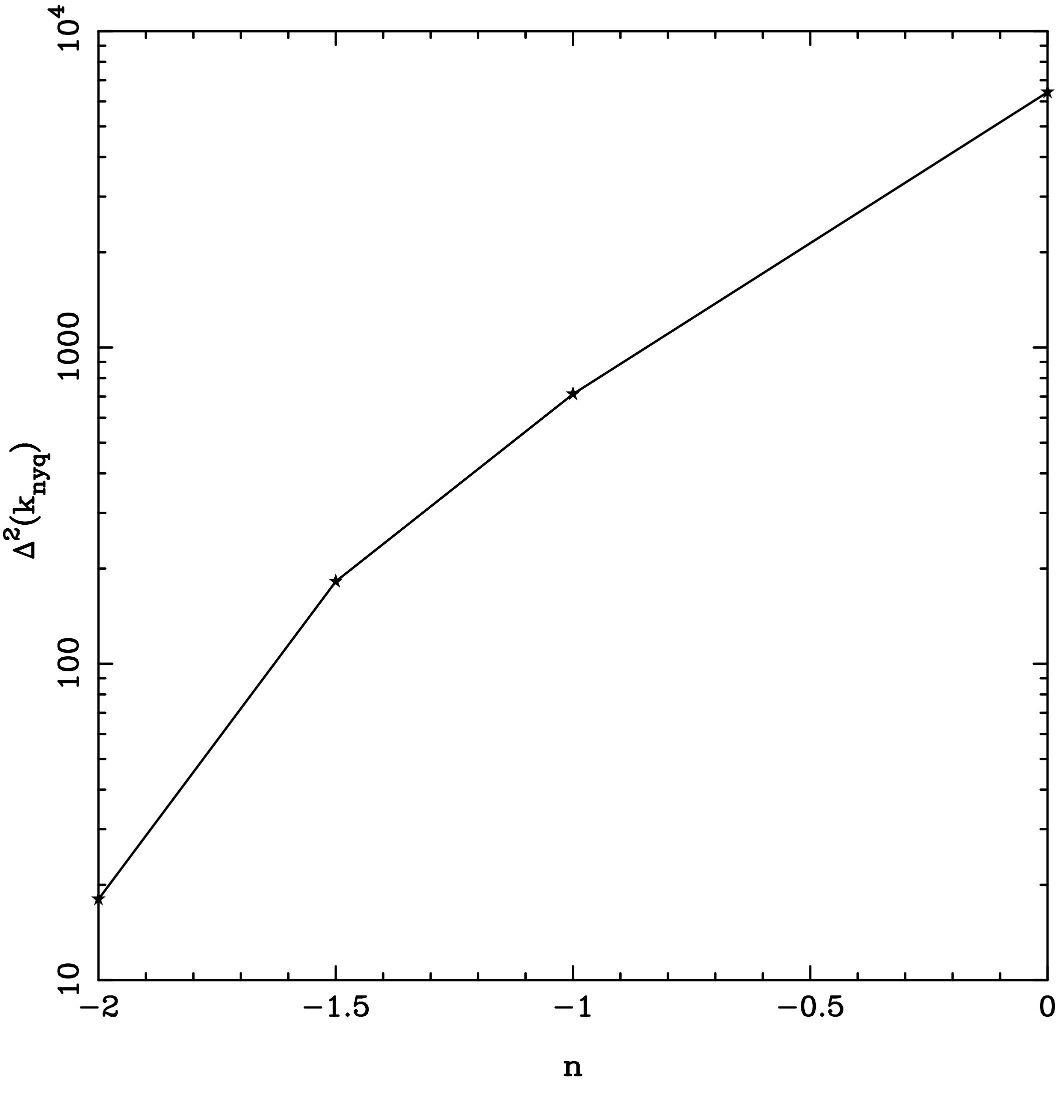

Figure 5.1: The dimensionless power spectrum Δ^2 at the Nyquist frequency k_{nyq} is plotted as a function of the index n of power law models. $\Delta^2(k_{nyq})$ is plotted at an epoch such that $\Delta^2(k_f) \simeq 0.01$, where $k_f = 2\pi/N_{box}$ is the fundamental mode for a simulation box of size N_{box}.

perturbations and peculiar velocities are confined to a single plane at the initial instant.

- We can study perturbations that do not depend on one of the three coordinates, i.e. we start with a set of infinitely long straight "*needles*", all pointing along one axis. The force of interaction then falls as $1/r$. Evolution keeps the needles pointed in the same direction and we study the properties of clustering in an orthogonal plane. Particles in the N-Body simulation represent the intersection of the needles with this plane and the interaction can be described using the Poisson equation in two dimensions.

In both of these approaches, the Universe is three-dimensional and the background is expanding isotropically.

The study of two-dimensional perturbations in a three-dimensional expanding Universe is not of much use : to begin with, we do not gain the desirable dynamic range in computation if we stick to three dimensions, even if perturbations are confined to a plane, because the force between particles has to be still computed by the solution of the Poisson equation in three dimensions. Also, the interaction of matter outside the plane with these perturbations eventually makes it a three-dimensional problem (Bagla, Engineer and Padmanabhan, 1998).

The second option is where the two-dimensional system is represented by the intersection of the needles with an orthogonal plane. The needles are the particles in the system and the force between these particles is given by the solution of the Poisson equation in two dimensions. Such a system is dichotomous with the background Universe expanding isotropically. However, analytic results for the evolution of density profiles around peaks in two dimensions have been derived for systems of this type (Filmore and Goldreich, 1984). Because scaling relations in the asymptotic regime depend on the kind of dynamical equilibrium that is reached in massive haloes (if an equilibrium is ever reached), density profiles of massive haloes are related to the non-linear scaling relations and we therefore choose this second option (Bagla, Engineer and Padmanabhan, 1998; Munshi and Coles, 1998).

In order to compute the force, we solve the Poisson equation for the perturbed part of the gravitational potential in two dimensions, whereas the unperturbed background is still the three-dimensional spherically symmetric Friedmann Universe. The perturbations are described by the mass per unit length of the needles representing the particles and the gravitational force due to such a particle situated at the origin has the form :

$$\mathbf{f}(\mathbf{r}) = -\left[\frac{Gm}{r^2}\right]\mathbf{r} \tag{5.1}$$

Here G is the gravitational coupling constant and m is the mass (per unit length).

5.2 The Mathematical Model

In the TreePM method, we explicitly divide the potential and force into a long range component and a short range component. The PM method is then used to compute only the long range component and the tree method is used to calculate the short range component. The gravitational potential (corresponding to the $1/r$ force) for our two-dimensional system can be split into two parts in Fourier space in a manner similar to that used in the three-dimensional TreePM method (Bagla, 2002) :

$$\begin{aligned}
\phi_k &= -\frac{2\pi G\rho_k}{k^2} && (5.2)\\
&= -\frac{2\pi G\rho_k}{k^2}\exp\left(-k^2r_s^2\right) - \frac{2\pi G\rho_k}{k^2}\left[1-\exp\left(-k^2r_s^2\right)\right] &&\\
&= \phi_k^l + \phi_k^s &&\\
\phi_k^l &= -\frac{2\pi G\rho_k}{k^2}\exp\left(-k^2r_s^2\right) && (5.3)\\
\phi_k^s &= -\frac{2\pi G\rho_k}{k^2}\left[1-\exp\left(-k^2r_s^2\right)\right] && (5.4)
\end{aligned}$$

where ϕ^l and ϕ^s are the long range and the short range potentials respectively. r_s is the scale of transition from the long range to the short range force or vice versa.

Expressions for the 2d long range and short range forces in real space are

$$\begin{aligned} \mathbf{f}^l(\mathbf{r}) &= -\left[1 - \exp\left(-\frac{r^2}{4{r_s}^2}\right)\right] \frac{Gm}{r^2}\mathbf{r} \\ \mathbf{f}^s(\mathbf{r}) &= -\exp\left[-\frac{r^2}{4{r_s}^2}\right] \frac{Gm}{r^2}\mathbf{r} \end{aligned} \qquad (5.5)$$

These equations describe the mathematical model for the force in a 2d TreePM code. The long range potential is computed in Fourier space using the PM method using eqn.5.3 instead of eqn.5.2. This potential is then used to compute the long range force. The short range force is computed directly in real space using eqn.5.5. This is done using the tree approximation. The short range force falls rapidly at scales $r \gg r_s$ and hence we need to take this into account only in a small region around each particle. The scale upto which we add the small-scale force is r_{cut}.

We have plotted the 2d long range and the short range forces (eqn.5.5) as a function of r in fig.5.2 to show their dependence on scale. The short range force closely follows the total force up to r_s and then falls rapidly at larger scales. Its magnitude falls below 0.1% of the total force by $5.25r_s$. Therefore, r_{cut} can be safely chosen to be of the order of $5r_s$ or smaller. The long range force reaches a peak around $2.5r_s$. It makes up most of the total force beyond $3r_s$. It falls rapidly below $2.5r_s$, becoming negligible below $r_s/2$.

We soften the gravitational force at small scales in order to ensure collisionless evolution. Each particle is given a finite size to soften the force. To find the force due to such a particle, we solve the Poisson equation in two dimensions with a source term given by an extended mass distribution represented by

$$\rho(r) = mW(r, \epsilon) \qquad (5.6)$$

Here $W(r, \epsilon)$ is the normalised spline kernel used in the SPH formalism (Monaghan, 1992), with ϵ the smoothing length. Here it also represents the size of the particle. $W(r, \epsilon)$ has the following form in two dimensions :

$$W(r, \epsilon) = \left(\frac{40}{7\pi\epsilon^2}\right) \begin{cases} 1 - 6\left(\frac{r}{\epsilon}\right)^2 + 6\left(\frac{r}{\epsilon}\right)^3, & 0 \geq \frac{r}{\epsilon} \leq 0.5 \\ 2\left(1 - \frac{r}{\epsilon}\right)^3, & 0.5 < \frac{r}{\epsilon} \leq 1.0 \\ 0\ , & \frac{r}{\epsilon} > 1.0 \end{cases} \qquad (5.7)$$

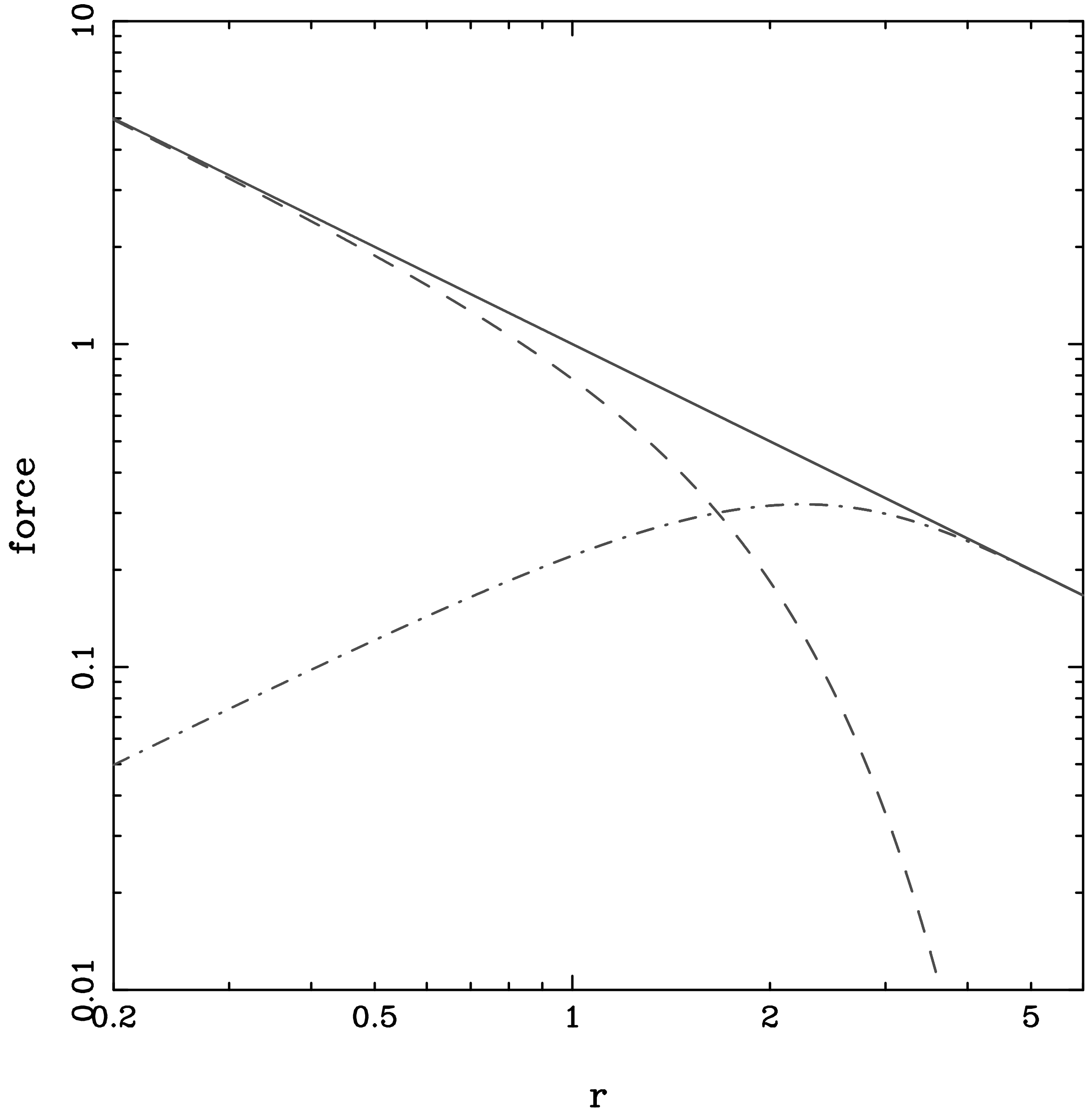

Figure 5.2: This figure shows the 2d long range force, the short range force as well as the total force as a function of scale. The $1/r$ force is shown by the solid line, the long range force by the dot-dashed line and the short range force by the dashed line. We have taken $r_s = 1$ here.

Solving the Poisson equation and matching the potential and its first derivative (i.e. the force) to obtain the constants of integration, we get the cubic spline softened potential :

$$\phi(r) = \begin{cases} \frac{Gmr^2}{7\epsilon^2}\left[20 - \frac{30r^2}{\epsilon^2} + \frac{96r^3}{\epsilon^3}\right] + C_2 + C_1\log(r), & 0 \geq \frac{r}{\epsilon} \leq 0.5; \\ \frac{Gmr^2}{7\epsilon^2}\left[40 - \frac{160r}{3\epsilon} + \frac{30r^2}{\epsilon^2} - \frac{32r^3}{5\epsilon^3}\right] + C_4 + C_3\log(r), & 0.5 < \frac{r}{\epsilon} \leq 1.0; \\ C_5\log(r) + C_6, & \frac{r}{\epsilon} > 1.0 \end{cases} \tag{5.8}$$

with the constants of integration C_1, C_2, C_3, C_4, C_5 and C_6 given by the following :

$$\begin{aligned} C_1 &= 0 = C_6 \\ C_2 &= \frac{Gm}{7}\left[8\log(\epsilon) - \log\left(\frac{\epsilon}{2}\right) - \frac{827}{60}\right] \\ C_3 &= -\frac{Gm}{7} \\ C_4 &= \frac{Gm}{7}\left[8\log(\epsilon) - \frac{154}{15}\right] \\ C_5 &= Gm \end{aligned}$$

We obtain the force $\mathbf{f}$ as a gradient of the potential :

$$\mathbf{f}(\mathbf{r}) = \begin{cases} -\left\{10 - \frac{30r^2}{\epsilon^2} + \frac{96r^3}{5\epsilon^3}\right\}\frac{4Gm}{7\epsilon^2}\mathbf{r}\ , & 0 \geq \frac{r}{\epsilon} \leq 0.5; \\ -\left\{\frac{80}{\epsilon^2} - \frac{160r}{\epsilon^3} + \frac{120r^2}{\epsilon^4} - \frac{32r^3}{\epsilon^5} - \frac{1}{r^2}\right\}\frac{Gm}{7}\mathbf{r}\ , & 0.5 < \frac{r}{\epsilon} \leq 1.0; \\ -\frac{Gm}{r^2}\mathbf{r}\ , & \frac{r}{\epsilon} > 1.0 \end{cases} \tag{5.9}$$

5.3 Error Estimation

Errors in force for various components in the 3d TreePM method have been studied in Chapter 3. The key features of the analysis carry over to the 2d TreePM method. Here we briefly discuss errors in the 2d long range force, the 2d short range force and the 2d TreePM force.

5.3.1 Error in the Long Range Force

The long range force, is computed by the Particle-Mesh method. In this scheme, the mesh and the interpolating function are the main sources of anisotropy. We use the Cloud-in-Cell (CIC) interpolating function (Hockney and Eastwood, 1988). To estimate errors in the long range force, we follow the method used in the error analysis for the 3d TreePM method in Chapter 3 of this thesis. We place a particle at a random position in a mesh cell and use our code to find the long range force at another random point. The force is evaluated at a large number of such random points scattered within some distance from the particle. This process is repeated for a number of times with a different position of the particle. We use the force calculated in this manner to compute the average long range force as a function of distance.

Fig.5.3 shows the expected long range force (solid curve), the average long range force for a particle (dot-dashed curve) as well as the average long range force obtained after removing the effect of the interpolating function (dashed curve). All three are plotted as a function of distance r. The offset in the force due to the interpolating function is removed by *de-convolving* the interpolating function (Bagla and Ray, 2003). The scheme is identical to that followed for the 3d TreePM method (see §3 of Chapter 3). The fractional error (scaled by the total $1/r$ force) in the long range force with deconvolution of the interpolating function is below 0.4% over the entire range of scales under consideration. We use the long range force with deconvolution of the interpolating function in our implementation of the 2d TreePM method.

5.3.2 Error in the Short Range Force

The short range force is calculated using the tree approximation (Barnes and Hut, 1986). Here the entire mass in a distant cell is assumed to be concentrated at the centre of mass of the cell. The error in force introduced by this approximation increases with $\theta : \theta \equiv \frac{d}{r}$, where d is the size of the cell and r is the distance from the point where the force is to be calculated to the centre of mass of the cell. We would not like to use cells that are too close or too large as that can lead to large errors. This is ensured by introducing the cell acceptance criterion, i.e. by comparing θ with a threshold $\theta_c : \theta \leq \theta_c$. To estimate the error due to the tree approximation, we compare the short range force due to particles in a given cell in the tree approximation ($\mathbf{f}^s_{cm}$) with the vector sum of the short range force

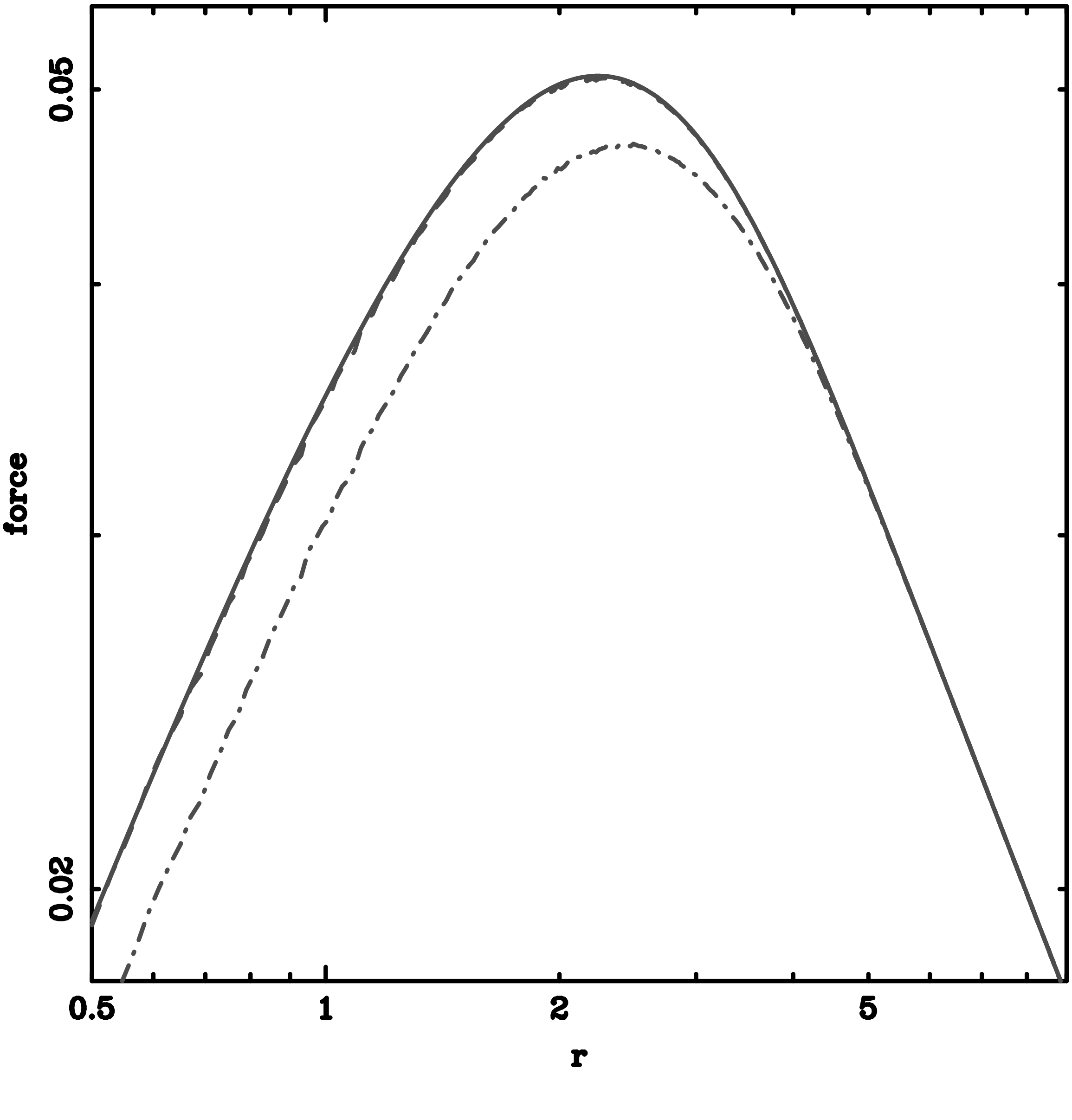

Figure 5.3: The long range force for a single particle is shown as a function of distance in this figure. The solid curve shows the expected long range force and the dot-dashed curve shows the long range force obtained in the simulation. The dashed curve represents the long range force obtained in the simulation after removing the effect of the interpolating function. $r_s = 1$ for all the three cases. The dashed curve follows the theoretical curve remarkably well and peak fractional error is less than 0.4% over the entire range of scales under consideration.

due to all the particles in the cell ($\mathbf{f}^s$). We also have the total $1/r$ reference force $\mathbf{f}_{cm}$. Then the relative error in the force contributed by the tree approximation is :

$$\epsilon_s = \frac{|\mathbf{f}^s_{cm} - \mathbf{f}^s|}{|\mathbf{f}_{cm}|} \tag{5.10}$$

Fig.5.4 shows the fractional error in the short range force ϵ_s as a function of distance r. Here we have taken $\theta_c = 0.5$ and $r_s = 1$. The error has been plotted for $N_p = 10$ and $N_p = 30$, where N_p is the total number of particles in a cell. To obtain the error, particles were distributed randomly in a cell and the force due to these particles was computed at an external point. The error is smaller for larger N_p as chances of an extreme distribution of particles is smaller in such a case. The error is constant at small scales, increases around the scales where the contribution of the short range force is decreasing rapidly, then plummets to very small values at larger scales. The error is large around the transition scale, i.e. around $r \sim r_s$, where the short range force deviates significantly from the $1/r$ form.

5.3.3 Error in the TreePM Force

We turn to the question of errors in the TreePM force calculated by adding the short range and the long range forces. It is important to estimate the errors in numerical evaluation of the force in a realistic situation, even though we do not expect errors to add up coherently. We compute errors for two distributions of particles : a homogeneous distribution and a clumpy distribution. For the homogeneous distribution, we use randomly distributed particles in a box. We use 1024^2 particles for the estimation and we use a 1024^2 grid for the long range force calculation. We compute the force for a reference setup ($r_s = 4$, $\theta_c = 0.01$, $r_{cut} = 6r_s$) and the setup we wish to test ($r_s = 1$, $\theta_c = 0.5$, $r_{cut} = 5r_s$). We compute the fractional error in force acting on each particle. This is defined as

$$\epsilon = \frac{|\mathbf{f} - \mathbf{f}_{ref}|}{|\mathbf{f}_{ref}|} \tag{5.11}$$

Fig.5.5 shows the cumulative distribution of fractional errors. The curves show the fraction of particles with error greater than a threshold error ϵ. The solid line shows this for the

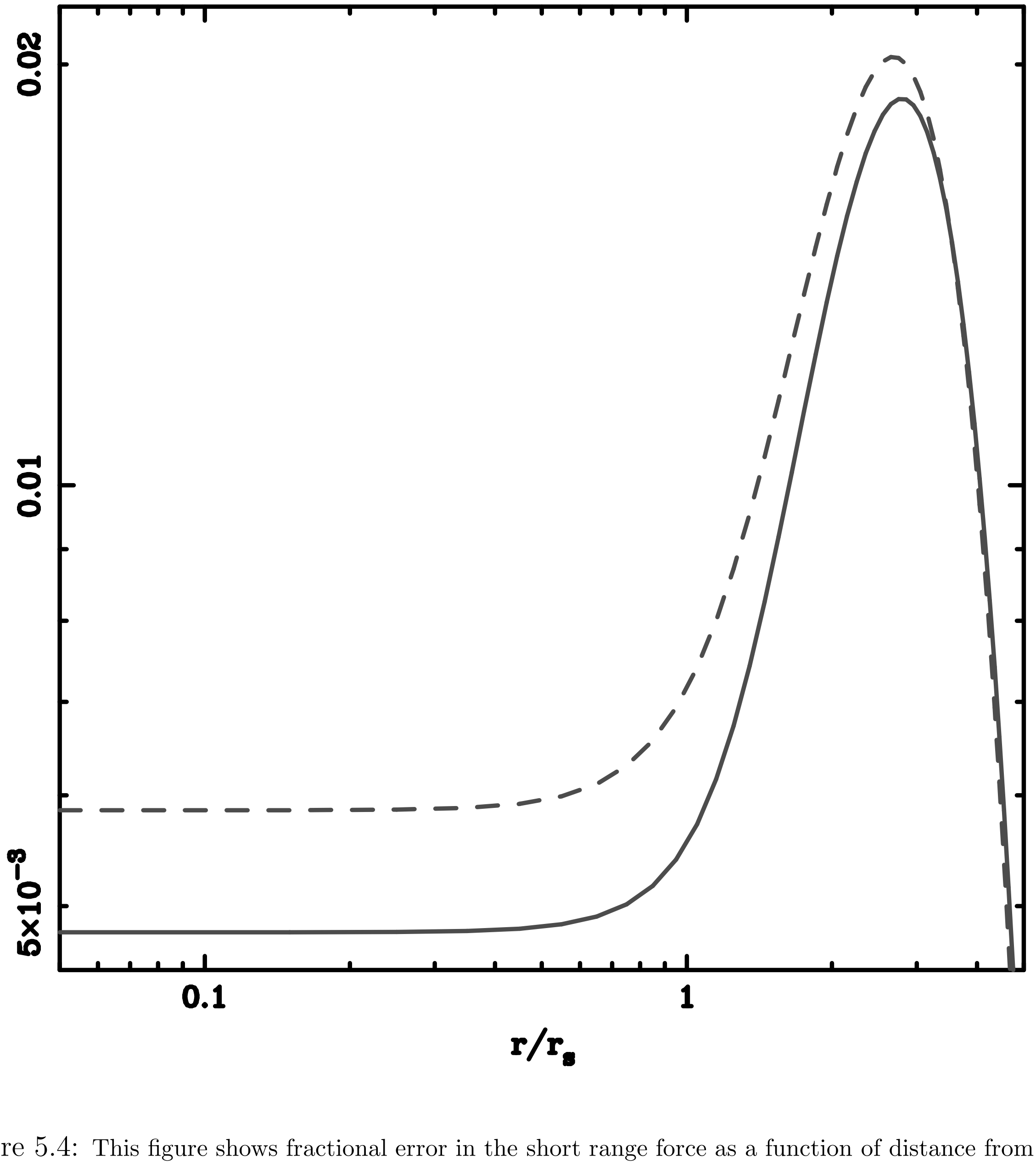

Figure 5.4: This figure shows fractional error in the short range force as a function of distance from a cell containing N_p particles distributed at random. The solid curve corresponds to $\theta_c = 0.5$ and $N_p = 30$ and the dashed curve is for the same θ_c and $N_p = 10$.

homogeneous distribution; error ϵ for 99% of the particles is less than 4%. Results for the clustered distribution of particles are shown by the dashed line and to generate this distribution we used the output of a 2d power law ($n = 1$) simulation run with a Particle-Mesh code in an Einstein deSitter background Universe. Errors in this case are somewhat smaller as compared to the homogeneous distribution for much the same reason as that for a 3d tree code (Hernquist, Bouchet and Suto, 1991) or the 3d TreePM code (Bagla, 2002; Bagla and Ray, 2003). Error ϵ for 99% of the particles is less than 2% for the clustered distribution.

5.4 Integrating the Equation of Motion

Our discussion so far has dealt only with the evaluation of force. This is our main focus here as the key difference between the TreePM and other methods is in the scheme used for evaluation of force. We use the Einstein deSitter background for all the 2d TreePM simulations. The equation of motion (eqns.1.47) is identical to that in three dimensions apart from the obvious differences due to the fact that the system is two-dimensional. The functional form of the gravitational force ($\vec{\nabla}\phi$) for a point source is, of course, different.

We choose the timestep for evolution of positions and velocities to be a small fraction of the smallest dynamical time in the system. The fraction chosen is fixed by requiring scale invariance in evolution of power law spectra - a simulation is repeated with different choices of the fraction until we converge to the largest timestep for which we can reach the non-linear regime ($\bar{\xi} \gg 10$) and retain scale invariance. We then use a timestep that is half of this largest timestep.

Fig.5.6 shows $\bar{\xi}$ as a function of $r/r_{nl}(t)$ for several epochs obtained from a 2d TreePM simulation of a power law model with index $n = -0.4$. $r_{nl}(t)$ is the scale which is going non-linear at time t and is defined by eqn.2.11. In the figure, we have plotted $\bar{\xi}$ for scales which are larger than two times the artificial softening length used in the simulation. We can see that the evolution is scale invariant. Thus the TreePM code evolves density

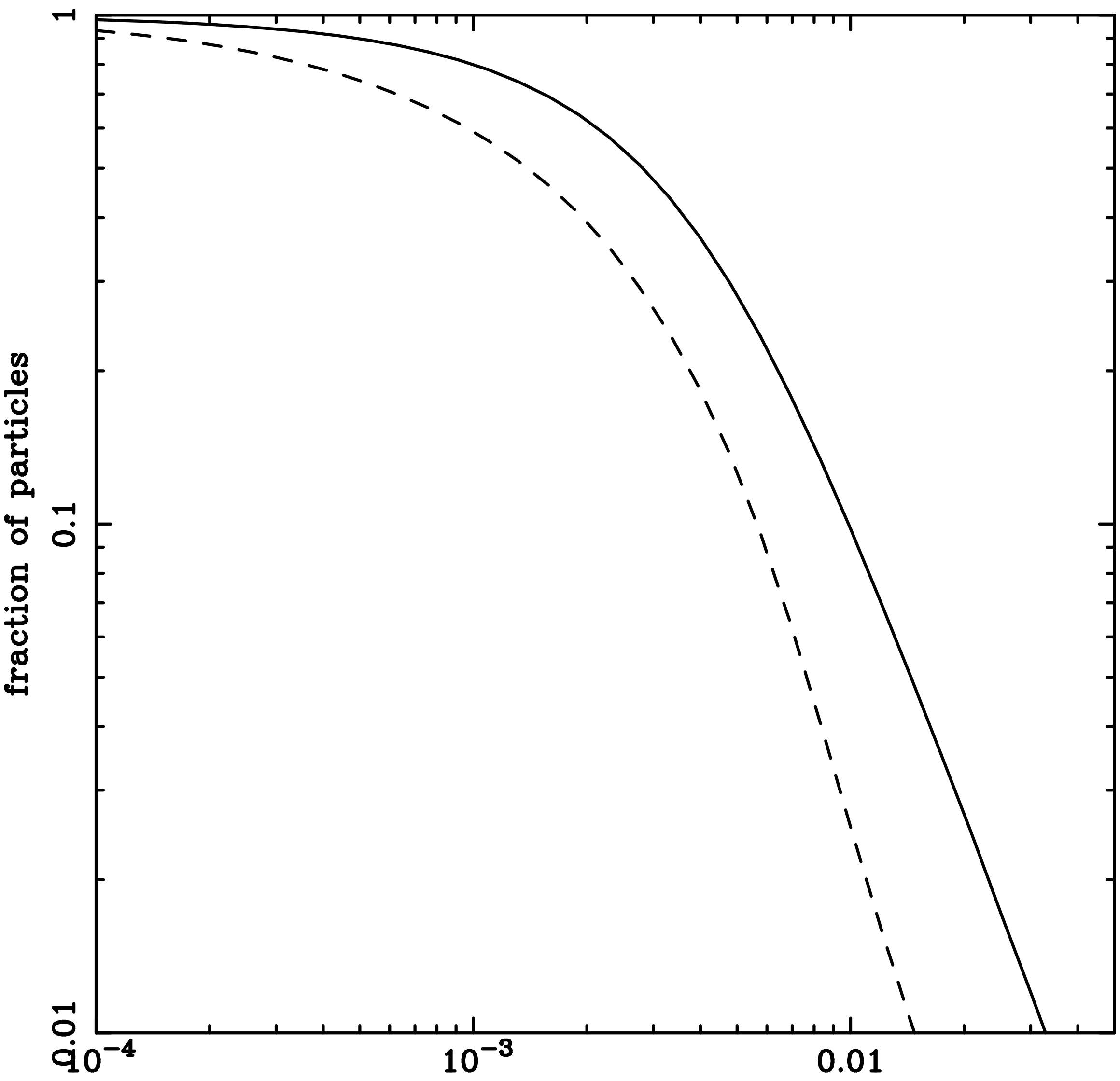

Figure 5.5: This figure shows the distribution of errors for a setup with $r_s = 1$, $\theta_c = 0.5$, $r_{cut} = 5r_s$. The variation of the fraction of particles with error greater than a threshold, as a function of the threshold error, is plotted. The solid line marks the error for a homogeneous distribution of particles and the dashed line shows the same for a clustered distribution. These errors were measured with respect to a reference force determined with very conservative values of r_s and θ_c. The panel shows that 99% of the particles have a fractional error in force that is less than 4% for the homogeneous distribution and less than 2% for the clustered distribution.

fluctuations correctly and can be used to study the strongly non-linear regime.

5.5 Comparison of the TreePM and the Particle-Mesh Codes

We ran a 2d simulation of a power law model with index $n = -0.4$ with 1024^2 particles on a 1024^2 grid in an Einstein deSitter background Universe with a Particle-Mesh code as well as with the TreePM code. For the TreePM run we used $r_s = 1$ and $\theta_c = 0.5$. We used cubic spline softening for the force with softening parameter $\epsilon = 0.2$.

The upper panel of fig.5.7 shows a slice each from the TreePM and PM simulations with the same initial conditions. The upper left panel shows the slice from the TreePM simulation and the upper right panel the same from the PM simulation. The large scale structures are indistinguishable in these panels. $\bar{\xi}(r)$ is plotted as a function of scale r in the lower left panel of fig.5.7. The thick line shows the correlation function for the TreePM simulation and the dashed line shows the same for the PM simulation. There are significant differences at small scales in the two-point correlation function from the two simulations. We have plotted $\bar{\xi}$ at scales larger than twice the softening length used in the TreePM simulation. The correlation function in the TreePM simulation matches with that from the PM simulation at large scales, but at scales of the order of the grid spacing and below, the TreePM simulation has a higher correlation function. This is an indication of the better force resolution of the TreePM method. The scale of softening for the PM code is marked by an arrow in the figure. This figure also shows that the PM method gives correct results almost upto the softening length in a PM code.

We also study the density profiles of haloes. A halo has been marked by a circle in the upper panel of fig.5.7 for the two simulations. This particular halo has been chosen as representative for the analysis because it is reasonably large and isolated. The lower right panel of fig.5.7 shows the average density $\bar{\rho}$ within a sphere of radius r with respect to the halo centre plotted as a function of r for the two haloes. The centre of a halo is defined

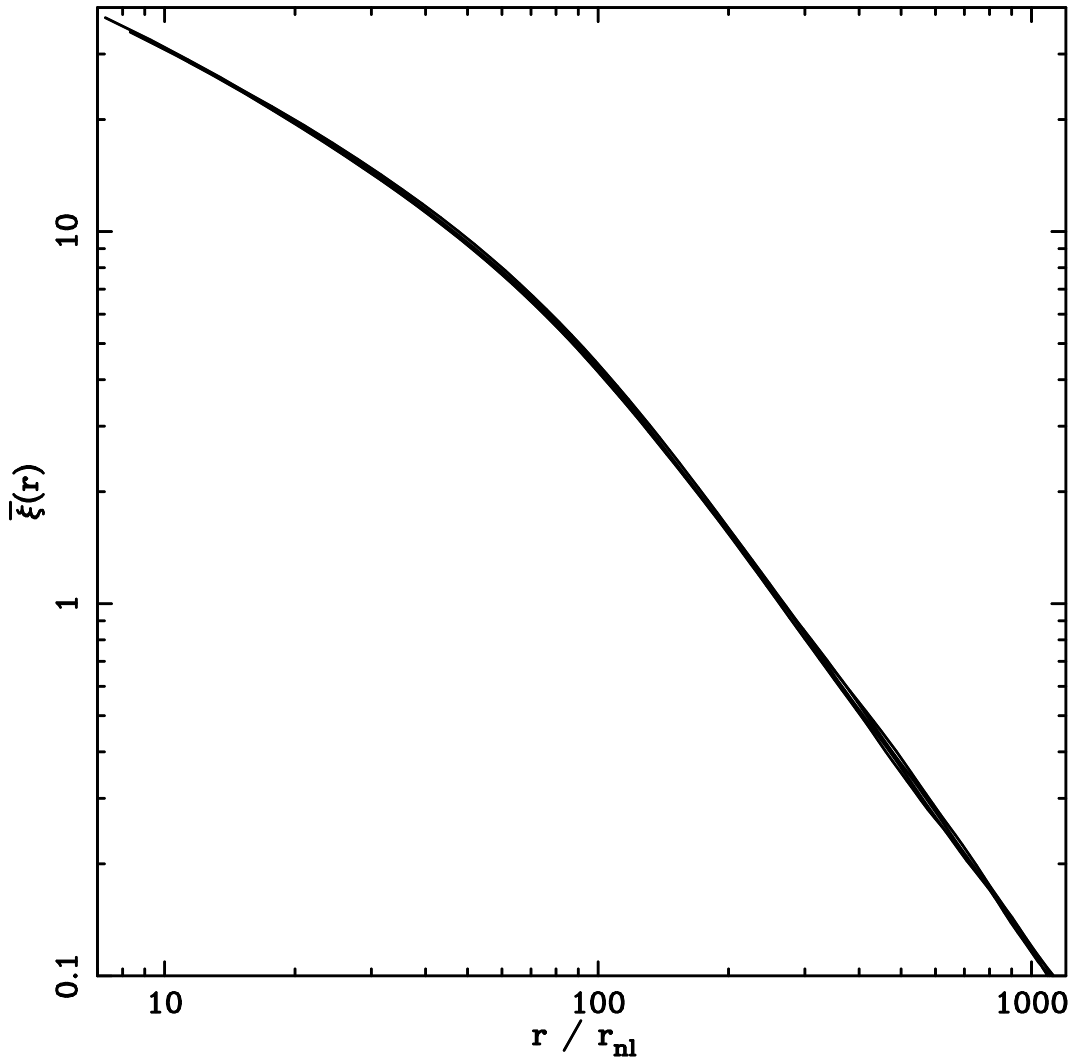

Figure 5.6: This figure shows $\bar{\xi}$ as a function of $r/r_{nl}(t)$ for several epochs. Here $r_{nl}(t)$ is the scale which is going non-linear at time t and it varies in proportion with $a^{2/(n+2)}$ in the Einstein deSitter model. The index of the power spectrum is $n = -0.4$. We have only plotted $\bar{\xi}$ at scales larger than twice the artificial softening length used in the simulation.

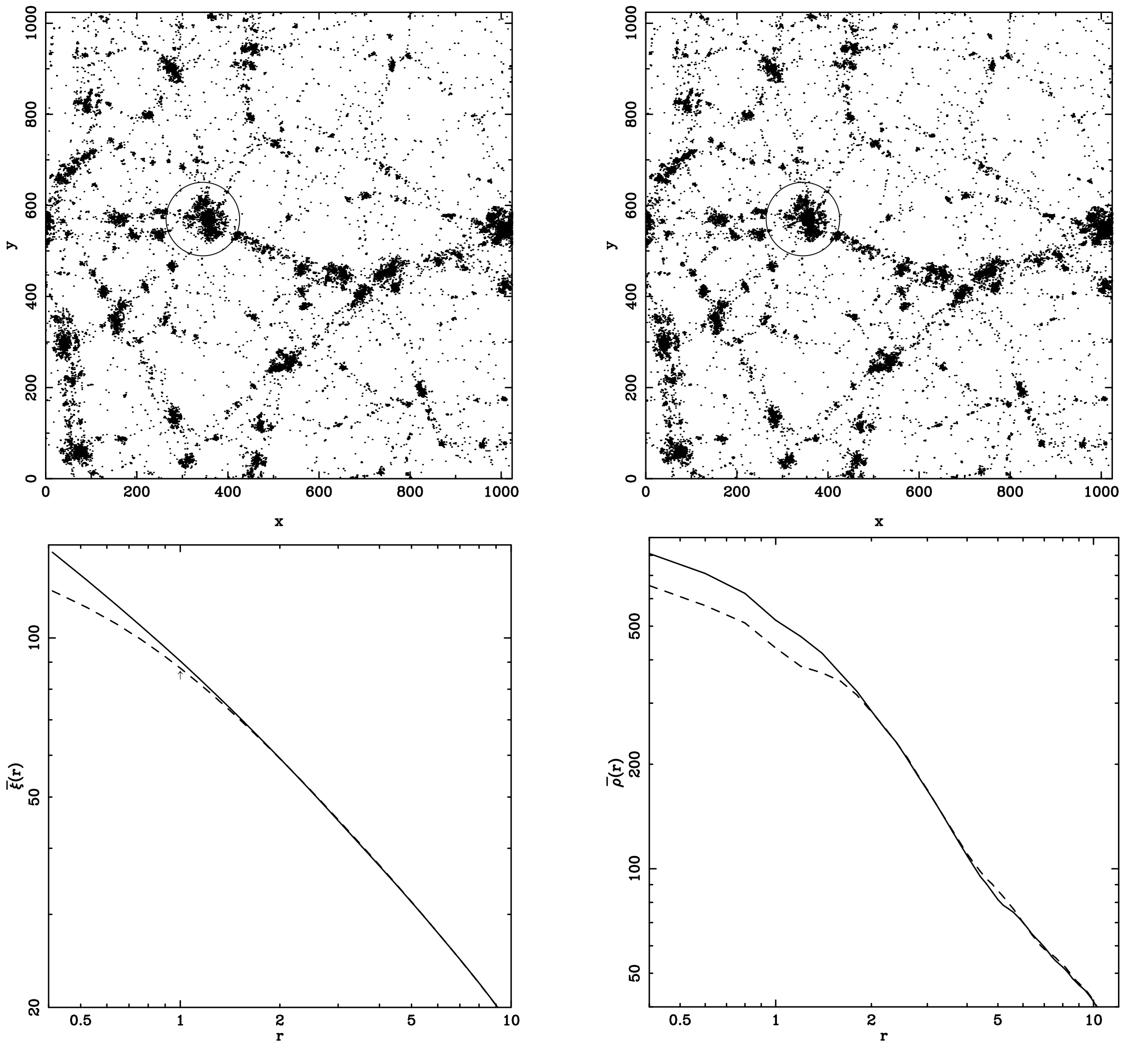

Figure 5.7: The upper panel of this figure shows a slice from a simulation of a power law model with index $n = -0.4$ for two cases. The upper left panel shows a slice from a TreePM simulation. and the upper right the same slice from a PM simulation with the same initial conditions. The large scale structures appear to be the same in the two. The lower left panel of this figure shows the averaged correlation function $\bar{\xi}(r)$ as a function of scale in grid units. The solid line shows this quantity for the TreePM simulation and the dashed line the same for the PM simulation. The scale marked by an arrow in the figure represents the larger softening scale for the PM code. The correlation functions match at large scales but the PM simulation underestimates the clustering at small scales. The lower right panel is a plot which shows the average density $\bar{\rho}$ within a sphere of radius r as a function of r (in grid units) for the two halos circled in the upper panel of the same figure. Again, the thick line shows the average density for the TreePM simulation and the dashed line shows the same for the PM simulation. The density profiles match at large scales as expected, but one can see that the TreePM simulation gives rise to haloes with higher central densities. We have plotted both $\bar{\xi}$ and $\bar{\rho}$ at scales which are at least two times larger than the artificial softening length used in the TreePM simulation.

as outlined in the discussion on the method of density peaks in §6.2 of Chapter 1 of the thesis. The solid line shows the density profile for the halo from the TreePM simulation and the dashed line the same from the PM simulation. We can see that, though not visibly obvious from the simulation slices in fig.5.7, the halo from the TreePM simulation is denser in the central region compared to the halo from the Particle-Mesh simulation. The density profiles converge at some distance from the halo centres. This distance is comparable to the softening length in the PM code.

5.6 Computational Requirements

In this section, we describe the computational resources required for our implementation of the 2d TreePM code. Given that we have combined the tree and the PM codes, the memory requirement is greater than that of either code. We need some variables per grid cell for the PM part, i.e. for the potential and the force. As regards the rest, the requirement is identical to that for a standard Barnes and Hut (2d) tree code. With efficient memory management, we need less than 75MB of RAM for a simulation with 1024^2 particles on a 1024^2 grid. The number mentioned is for floating point variables. We have used double precision variables for the test cases in this study. In that case the requirement goes up by a factor of two. The time taken (per timestep per particle) by the 2d TreePM code ($r_s = 1$, $\theta_c = 0.5$, $r_{cut} = 4.5r_s$, $N_p = 1024^2$, $N_{grid} = 1024^2$) is about 240 microseconds on a 2.4GHz Xeon workstation. The code has been compiled with the Intel F90 compiler.

5.7 Summary

In this chapter, we have described the two-dimensional TreePM method. This method offers greater dynamic range and better resolution compared to the 2d Particle-Mesh method and can therefore probe the non-linear regime in two-dimensional cosmological simulations more effectively. The 2d TreePM code will allow us to explore a higher dynamic range in densities for studying scaling relations in two-dimensions as compared to earlier work done using Particle-Mesh codes (Bagla, Engineer and Padmanabhan, 1998; Munshi and Coles, 1998). Work in this direction has been reported in a later chapter of the thesis.

Chapter 6

Comments on the Size of the Simulation Box in Cosmological N-Body Simulations

N-Body simulations are an important tool in the study of formation of large scale structures. All of the work presented in this thesis is centred around algorithms for N-Body simulations, particularly the TreePM method, and applications of N-Body simulations to problems in gravitational clustering in the Universe. Much of the progress in understanding the physics of the high redshift Universe and comparison with observations would not have been possible without N-Body simulations. Given the importance of this tool it is essential to understand its limitations as ignoring these can easily lead to interesting, but unreliable results. In this chapter we analyse the limitations arising out of the finite size of the simulation volume[1]. A finite size implies that modes larger than the size of the simulation volume are ignored in the simulation and a truncated power spectrum is simulated. In N-Body simulations, a representative region of the Universe is simulated with periodic boundary conditions. The effect of perturbations at scales smaller than the mass resolution of the simulation and that of perturbations at scales larger than the size of the box are ignored. Indeed, even perturbations at scales comparable to the size of the box are undersampled as the power spectrum is sampled at a discrete set of wave

[1] This chapter is based on *Comments on the Size of the Simulation Box in Cosmological N-Body Simulations* (J.S. Bagla and S. Ray), astro-ph/0410373. Accepted for publication in MNRAS.

numbers. It has been shown that, for gravitational dynamics in an expanding Universe, perturbations at small scales do not significantly influence collapse of large scale perturbations (Peebles, 1974a, 1985; Little, Weinberg and Park, 1991; Bagla and Padmanabhan, 1997b; Couchman and Peebles, 1998; Bagla, Prasad and Ray, 2004). Therefore, ignoring perturbations at scales much smaller than the scales of interest does not affect the results of N-Body simulations in a significant manner.

Use of periodic boundary conditions implies that the average density in the simulation box equals the average density in the Universe. Therefore the size of the simulation volume has to be chosen such that the amplitude of typical fluctuations at that scale and at larger scales is ignorable. If the amplitude of perturbations at larger scales is not ignorable and we do not take the contribution of these scales into account, then clearly the simulation cannot be a faithful representation of the model being simulated. It is not obvious as to when fluctuations at larger scales can be considered ignorable. We propose one method of quantifying this in the work here.

It is known from earlier studies that if the amplitude of density perturbations at the box scale is small but not ignorable, simulations underestimate the correlation function, though the number density of low mass haloes does not change by much (Gelb and Bertschinger, 1994a). In other words, the formation of small haloes is not affected, but their distribution is affected by the non-inclusion of long wavelength modes. The number of massive haloes changes significantly (Gelb and Bertschinger, 1994a).

Methods have been devised for incorporating the effects of perturbations at scales larger than the simulation volume (Tormen and Bertschinger, 1996; Cole, 1997). These methods assume that if, in any N-Body simulation, the boxsize is chosen to be large enough, the contribution of larger scales can be incorporated by adding displacements and velocities due to the larger scales independent of the evolution of the system. Essentially, mode coupling between larger and smaller modes is ignored. The primary motivation for developing tools of this nature is to expand the dynamic range over which results of N-Body simulations are valid by adding corrections that change the distribution of matter

Name	N_p	L_{box}	ϵ	Cutoff	Scale
T_300_C_0	256^3	300h^{-1}Mpc	0.47h^{-1}Mpc	None	300h^{-1}Mpc
T_300_C_2	256^3	300h^{-1}Mpc	0.47h^{-1}Mpc	$k \leq 2k_f$	150h^{-1}Mpc
T_300_C_3	256^3	300h^{-1}Mpc	0.47h^{-1}Mpc	$k \leq 3k_f$	100h^{-1}Mpc
T_300_C_4	256^3	300h^{-1}Mpc	0.47h^{-1}Mpc	$k \leq 4k_f$	75h^{-1}Mpc

Table 6.1: This table lists the parameters for the N-Body simulations we have used. All the simulations were done using the TreePM code with a configuration described in Chapter 3 of the thesis. We simulated the ΛCDM model with $\Omega_b = 0.05$, $\Omega_{dm} = 0.25$, $\Omega_{nr} = 0.3$, $k = 0$, $h = 0.7$ and $n = 1$. The first column lists the name of the simulation, the second column the number of particles in the simulation, the third column the size of the box in physical units, the fourth column the softening length for force, the fifth column the cutoff used in units of the fundamental mode k_f of the simulation box and the last column the corresponding cutoff scale in h^{-1}Mpc.

and velocities at scales comparable to the simulation volume. But this invariably brings up the issue of what is a large-enough scale in any given model such that these methods can be used to add the effect of larger scales without introducing errors. Our goal in this work is to understand the effect of large scales on scales that are much smaller than the simulation volume. In this chapter of the thesis we propose a measure based on mass functions to quantify the effect of large scales.

6.1 Numerical Experiments

We carried out a series of N-Body experiments in order to study the effect of perturbations at large scales on perturbations at small scales. The simulations were carried out using the TreePM method (Bagla, 2002; Bagla and Ray, 2003; Ray and Bagla, 2004). We simulated the ΛCDM model with $\Omega_b = 0.05$, $\Omega_{dm} = 0.25$, $\Omega_{nr} = 0.3$, $\Omega_{tot} = 1$, $h = 0.7$ and $n = 1$. We ran a simulation without any truncation of the power spectrum save those imposed by the finite size and resolution of the simulation box. This simulation serves as a reference for other simulations where we truncated the power spectrum. We next used a sharp cutoff such that the power spectrum at $k \leq k_c$ was taken to be zero. In the initial conditions for all the simulations run, the phases as well as the amplitudes of corresponding waves were taken to be the same (except where the cutoffs were applied). A comparison of these simulations allows us to estimate the effect of density perturbations

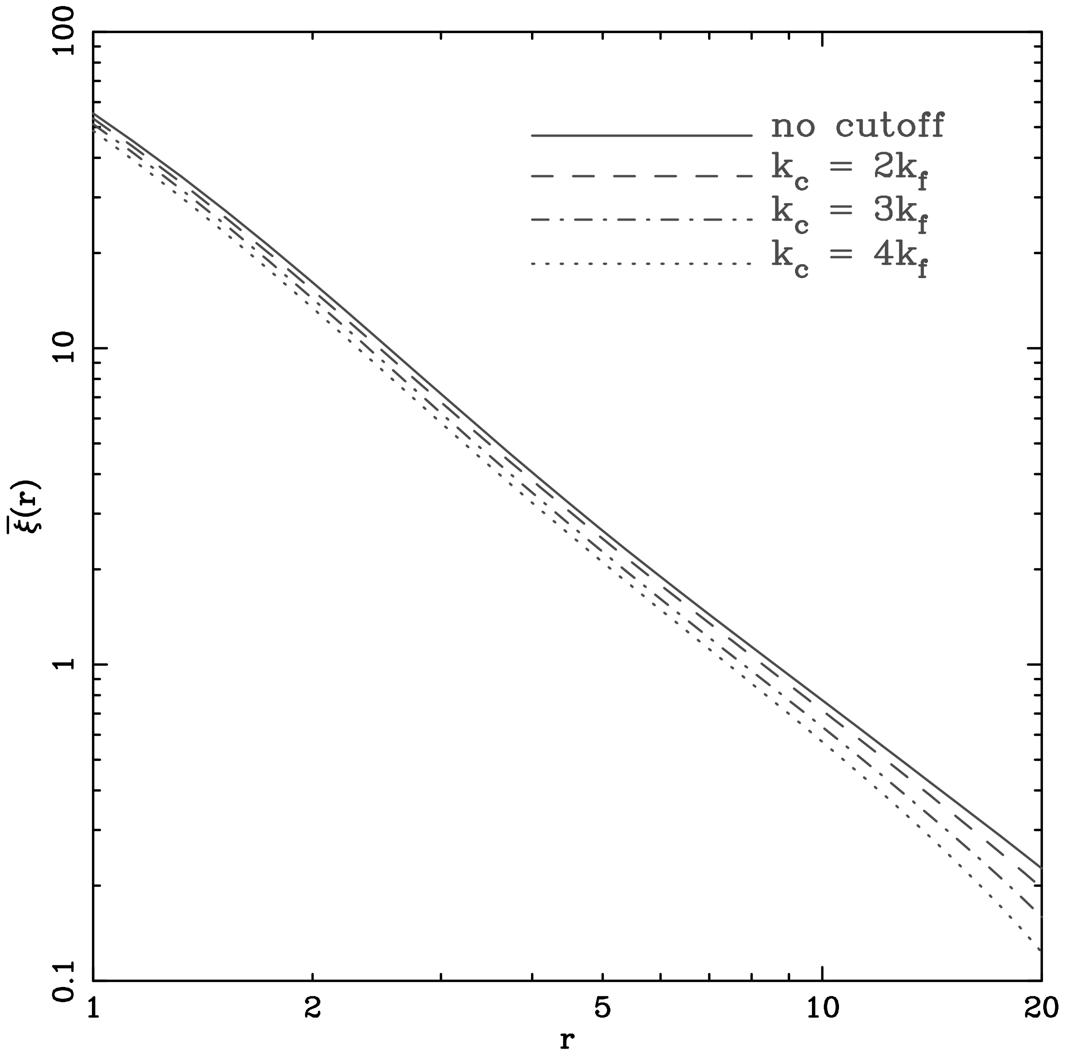

Figure 6.1: This figure shows the averaged correlation function $\bar{\xi}$ as a function of r for the models listed in table 6.1. The amplitude of the correlation function decreases as the cutoff becomes smaller. The correlation function is underestimated by 10% at $r = 2\mathrm{h}^{-1}$Mpc for a cutoff of $100\mathrm{h}^{-1}$Mpc.

at large scales on growth of perturbations at small scales. Detailed specifications of these simulations are listed in table 6.1 where the cutoff wave number is listed in units of the fundamental mode $k_f = 2\pi/L_{box}$ of the simulation box.

We compare the output of these simulations to see whether retaining or dropping of long wave modes affects quantities of interest at smaller scales. In all these comparisons we concentrate on scales smaller than $20\mathrm{h}^{-1}$Mpc whereas the smallest cutoff we have used is $75\mathrm{h}^{-1}$Mpc, so that the two sets of scales are well separated. The amplitude of root

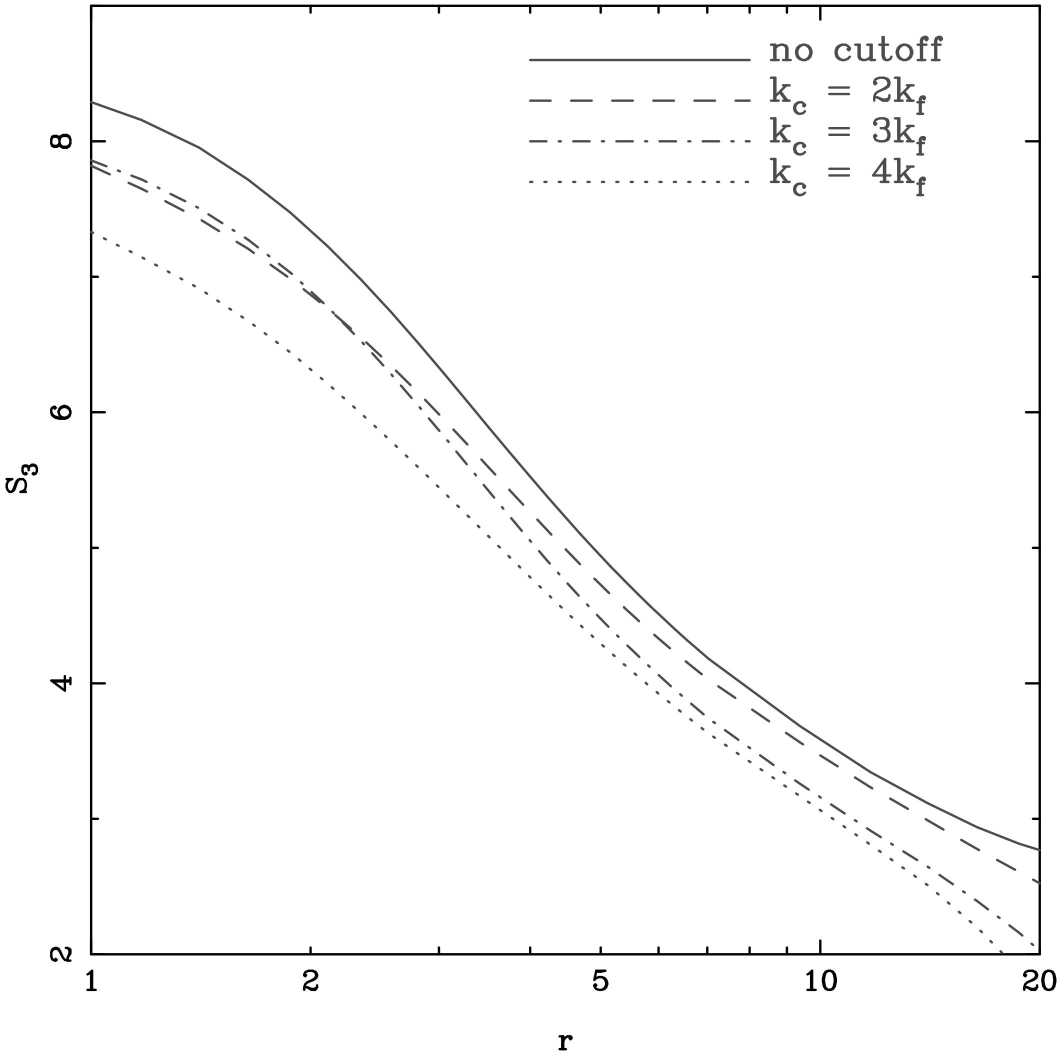

Figure 6.2: This figure shows the skewness S_3 as a function of r for the models listed in table 6.1. The skewness decreases as the cutoff becomes smaller.

mean square fluctuations in mass at $r = 75\mathrm{h}^{-1}\mathrm{Mpc}$ and $z = 0$ is $\sigma \sim 0.1$ in the model considered here and we can regard this to be small.

Fig.6.1 shows the averaged correlation function $\bar{\xi}$ as a function of r at $z = 0$ for the models listed in table 6.1. The amplitude of the correlation function decreases as cutoff becomes smaller. The shape of the correlation function does not change at small scales. This result follows expectations and is indeed similar to fig.2 in Gelb and Bertschinger (1994). Thus the overall effect of perturbations at very large scales is to enhance the amplitude of fluctuations at smaller scales. Ignoring larger scales leads to an underestimation of correlations at small scales. The correlation function is underestimated by 10% at $r = 2\mathrm{h}^{-1}\mathrm{Mpc}$ for a cutoff of $100\mathrm{h}^{-1}\mathrm{Mpc}$. At this scale ($r = 100\mathrm{h}^{-1}\mathrm{Mpc}$), $\sigma = 0.07$ at $z = 0$. We have plotted the skewness S_3 as a function of scale in fig.6.2 for the same set of models. The differences between models here is pronounced. It is clear from these two figures that fairly large scales ($l \leq 100\mathrm{h}^{-1}\mathrm{Mpc}$) make a significant contribution to clustering at small scales ($l \leq 10\mathrm{h}^{-1}\mathrm{Mpc}$).

It is possible to correct for the contribution of larger scales for the moments of a particle distribution (Colombi, Bouchet and Schaeffer, 1994). Thus if we are interested only in the moments then we can take scales larger than the simulation volume into account and compute the correct answer. There are many quantities of interest other than the moments of a distribution and there is no generic method for computing the corrections.

A method has been devised to incorporate the effect of displacements and velocities contributed by large scales (Tormen and Bertschinger, 1996). Here displacements and velocities contributed by larger scales are computed using the Zel'dovich approximation (Zeldovich, 1970) and added to the numbers computed in an N-Body simulation. Essentially this assumes that there is no coupling in contribution of modes within the box and modes larger than the box. With this assumption other methods for computing the displacements due to modes outside the box can also be used in place of the Zel'dovich approximation. The assumption of no mode coupling is true only in the linear regime and

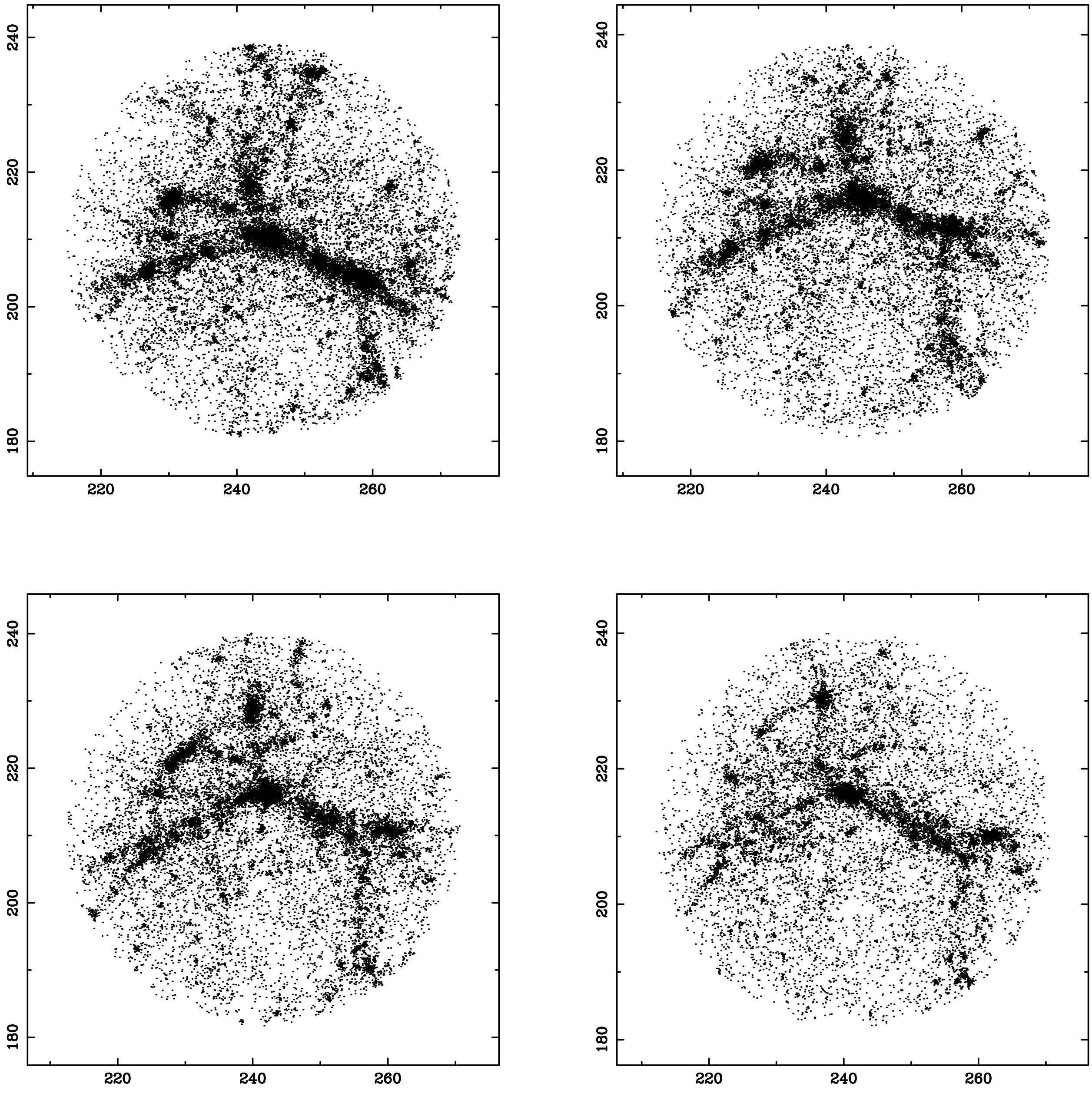

Figure 6.3: Each panel of this figure shows the projected particle distribution in the neighbourhood of a massive cluster from a simulation of the ΛCDM model. The top left panel is for the first simulation in table 6.1, the top right panel for the second simulation, the lower left panel for the third simulation and the lower right panel for the fourth simulation in the table. The total mass in the central massive halo decreases as the large scale cutoff is introduced and then decreased. The number, masses and the distribution of smaller clumps around the central clump also change significantly as the cutoff is reduced below 150h^{-1}Mpc.

hence we still require the amplitude of perturbations at the scale of the simulation box to be much smaller than unity. How small the amplitude should be can only be determined by trial and error. It has been pointed out that the effect of mode coupling can be emulated by enhancing the displacements without modifying the velocities (Cole, 1997). The amount of enhancement needed cannot be derived from first principles. These tools essentially enhance the dynamic range of an N-Body simulation by making these corrections - displacements due to large scales correct the velocity fields and the distribution of particles at scales comparable to the simulation volume.

Corrections to displacements and velocities can be made without worrying about mode coupling if their effect is small in some absolute sense. None of these methods can be effective if the displacements contributed by large scale modes move two collapsed objects to the same location or move matter that has fallen into a collapsed structure out of it. If the contribution of modes that can affect collapsed structures is not taken into account then the properties of these objects, e.g. mass, angular momentum, density profile, etc., may differ significantly from their asymptotic values. This is illustrated in fig.6.3. This figure shows the projected particle distribution in the neighbourhood of a massive cluster from simulations of the ΛCDM model. The four panels correspond to the models listed in table 6.1. The total mass in the central massive halo decreases as the large scale cutoff is introduced and then decreased. The number, masses and the distribution of smaller clumps around the central clump also change significantly as the cutoff is reduced below $150\text{h}^{-1}\text{Mpc}$.

Fig.6.4 shows the fraction of mass in collapsed structures with mass greater than M for the models listed in table 6.1. This fraction is plotted as a function of mass M. The solid curve shows $F(> M)$ for the simulation without an explicit cutoff (of course, there is an implicit cutoff at the boxsize of $300\text{h}^{-1}\text{Mpc}$). Other curves show the same function for different values of the cutoff and, as the cutoff scale becomes smaller, $F(> M)$ decreases at large M. At smaller masses ($M \leq 10^{14}\text{M}_{\odot}$), the difference in models is negligible. Differences between different curves at the high mass ($M \simeq 10^{15}\text{M}_{\odot}$) end are significant -

more than a factor of two between the extreme curves. Thus ignoring large scale modes results in an underestimation of the number of massive haloes. $F(> M)$ varies rapidly up to a cutoff of 150h^{-1}Mpc and changes very little as the cutoff moves to larger scales. We may conclude from here that scales larger than 150h^{-1}Mpc do not contribute significantly to collapsed structures in the currently favoured ΛCDM models.

6.2 The Proposed Criterion

In the previous section we have argued that the non-trivial contribution of large scale modes is the one that leads to the collapse of haloes - other effects like displacements can be incorporated, in principle, using algorithms like MAP (Tormen and Bertschinger, 1996; Cole, 1997). We can use this fact to devise a criterion to decide whether the boxsize of a simulation is sufficiently large or not.

The mass fraction in collapsed haloes of mass M or larger is given by the Press-Schechter mass function (Press and Schechter, 1974; Bond et al., 1991) :

$$F(M, z) = \mathrm{erfc}\left[\frac{\delta_c}{\sqrt{2}\sigma_L\left(M, z\right)}\right] \tag{6.1}$$

The parameter δ_c indicates the linearly extrapolated density contrast at which the perturbation is expected to collapse and virialise in non-linear collapse. Its value is taken to be 1.686 here - the precise value is not very relevant in this context. Here $\sigma_L(M, z)$ is the linearly extrapolated root mean square (rms) amplitude of fluctuations at the mass scale M and redshift z.

$$\begin{aligned}
\sigma_L^2(M, z) &= 9D_+^2(z)\int\limits_0^\infty \frac{dk}{k}\Delta^2(k)\left[\frac{\sin kr - kr\cos kr}{k^3r^3}\right]^2 \\
M &= \frac{4\pi}{3}\bar{\rho}_{nr}r^3
\end{aligned} \tag{6.2}$$

Here $P(k)$ is the power spectrum of density fluctuations, linearly extrapolated to $z = 0$ and $D_+(z)$ is the growing mode in the linear perturbation theory normalised so that $D_+(z = 0) = 1$. $\bar{\rho}_{nr}$ is the average density of matter in the Universe.

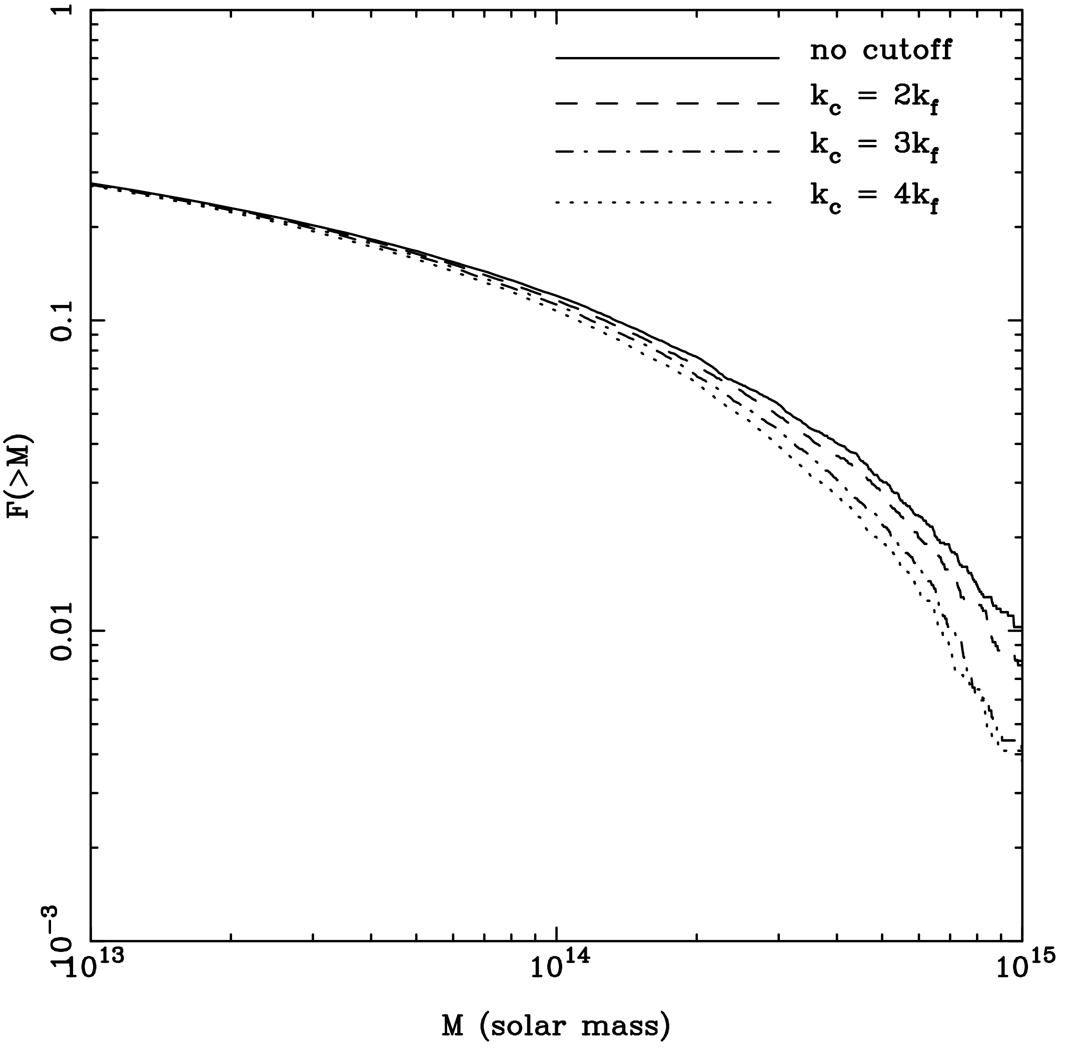

Figure 6.4: This figure shows the fraction of mass in collapsed structures with mass greater than M for the models listed in table 6.1. The solid curve is for the simulation with no cutoff, the dashed curve for a cutoff of 150h^{-1}Mpc, the dot-dashed curve for a cutoff of 100h^{-1}Mpc and the dotted curve for a cutoff of 75h^{-1}Mpc. While the change in cutoff does not change the collapsed mass fraction for $M \leq 10^{14}\mathrm{M}_{\odot}$, the collapsed fraction is underestimated at larger masses when we remove the large scale modes. Suppression of mass function around $M = 10^{15}\mathrm{M}_{\odot}$ is significant when the cutoff is at scales smaller than 150h^{-1}Mpc.

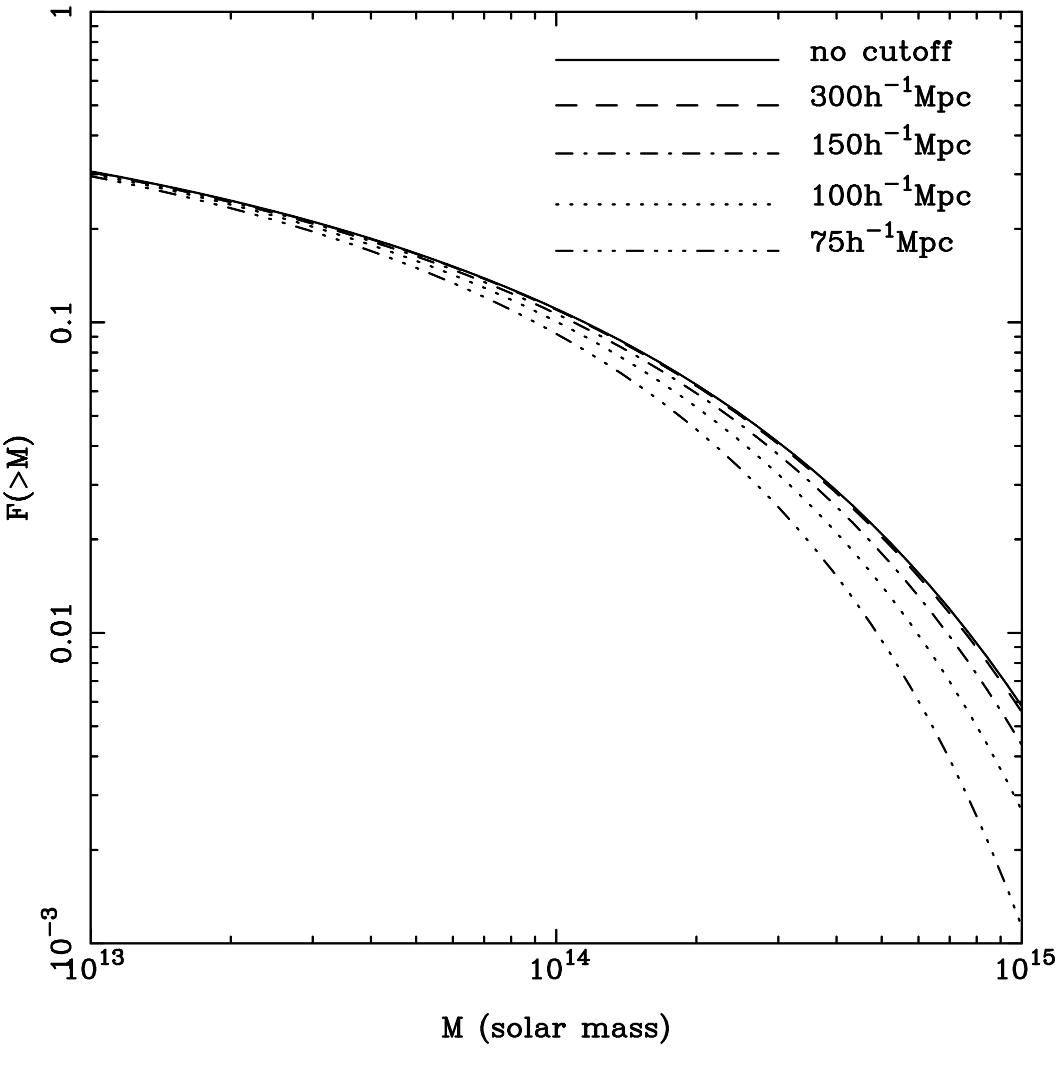

Figure 6.5: This figure shows the fraction of mass in collapsed structures with mass greater than M for the ΛCDM model used here. This was computed using the Press-Schechter mass function (Press and Schechter, 1974). A cutoff was used to remove contributions from large scales (see §3 for details). Note that the values of the cutoff used here coincide with those used in the simulations of the models listed in table 6.1 and plotted in fig.6.4.

In an N-Body simulation, the initial conditions sample a range of values of the wave number k. The amplitude of *rms* fluctuations in a simulation will be different for different simulations due to the partial sampling of modes in any simulation. The lower end of the range of modes sampled corresponds to the fundamental mode in the simulation volume and the upper end (k_{max}) corresponds to the smallest scales sampled in the simulation. From the form of the integral (eqn.6.2) it is clear that at a given scale r, wave modes with $k \leq 2\pi/r$ contribute more significantly than the rest. Given this and the fact that most modern N-Body simulations have sufficient dynamic range, we can concentrate on the lower limit of the range of wave numbers sampled in an N-Body simulation as modes with $k \geq k_{max}$ do not influence scales resolved in the simulation in any significant manner. We can then estimate the *rms* fluctuations in an N-Body simulation by changing the lower limit of the integral in eqn.(6.2) from 0 to $2\pi/L_{box}$ while leaving the upper limit unchanged. The fluctuations will now be a function of the cutoff as well :

$$\begin{aligned}\sigma_L^2(M, z, L_{box}) &= 9D_+^2(z) \\ &\times \int\limits_{2\pi/L_{box}}^{\infty} \frac{dk}{k}\Delta^2(k)\left[\frac{\sin kr - kr\cos kr}{k^3r^3}\right]^2\end{aligned} \tag{6.3}$$

We can use this in eqn.(6.1) and obtain $F(M, z, L_{box})$, the expected collapsed mass fraction in an N-Body simulation.

In the previous section we found that in an N-Body simulation the collapsed mass fraction does not change by much if the cutoff is larger than 150h^{-1}Mpc. This conclusion is reaffirmed by the theoretical calculation of the mass function and collapsed mass fraction. Fig.6.5 shows $F(M, z = 0, L_{box})$ as a function of mass M for different values of the cutoff scale L_{box}. The masses in the most massive collapsed structures change rapidly with the cutoff, implying that the number densities of the most massive structures depend strongly on the large scale modes. Comparison with fig.6.4 shows that the theoretical calculation and the simulations give comparable results.

The criterion we propose here is this : the physical scale L_{box} corresponding to the size of the simulation volume can be considered to be large enough if the expected fraction of mass in haloes $F(M, z, L_{box})$ is comparable to $F(M, z)$, the fraction of mass in haloes when the full spectrum is taken into account. In other words, we require convergence of expected mass in haloes for the simulation volume to be considered large enough so that all the relevant scales are contained within it. As before, we are interested in the effect of large scales on scales much smaller than L_{box}. Therefore we wish to see this convergence at the mass scales of typical haloes. We define the mass scale of non-linearity as $\sigma_L(M_*, z) = 1$ and study the convergence of the mass function at this scale and at neighbouring mass scales.

We look at two mass scales for the ΛCDM model : M_* and $10M_*$. As we shall see, these are more relevant than smaller mass scales in most situations. We require $F(M, z, L_{box}) = (1 - \epsilon)F(M, z)$ to find the threshold length scale L_{box}, with $\epsilon = 0.05$ and a less conservative limit of $\epsilon = 0.1$. These criteria allow us to develop a feel for our approach. We certainly require reasonable convergence at the scale of non-linearity in simulations. For some applications, as when studying rich clusters of galaxies, we require good convergence for very massive haloes. Fig.6.6 shows L_{box} as a function of redshift z according to these criteria for the ΛCDM model (see §2 for the values of the cosmological parameters). The solid curve ($\epsilon = 0.1$) and the dot-dashed curve ($\epsilon = 0.05$) are for $M = M_*$, the dashed curve ($\epsilon = 0.1$) and the dotted curve ($\epsilon = 0.05$) for $M = 10M_*$. For the ΛCDM model, $M_*(z = 0) = 1.2 \times 10^{14}\mathrm{M}_\odot$ - therefore the criteria used here refer to mass scales of typical clusters of galaxies and rich clusters of galaxies respectively at redshift $z = 0$. At $z = 0$, it is clear that a box larger than $100\mathrm{h}^{-1}\mathrm{Mpc}$ is required even if we are interested in haloes with mass M_* and can tolerate an offset of 10% in the collapsed mass. Using a smaller boxsize leads to a greater underestimation of the collapsed mass in haloes. The requirement for the minimum boxsize becomes more stringent if we wish to study rich clusters of galaxies or use a tolerance level of 5% ($\epsilon = 0.05$).

At $z = 3$, $M_* = 1.2 \times 10^{11}\mathrm{M}_\odot$. This is the mass of a typical galaxy halo at that redshift and to get the statistics of these objects right we should have a boxsize of at least $30\mathrm{h}^{-1}\mathrm{Mpc}$ ($\epsilon = 0.1$); the required boxsize increases to $50\mathrm{h}^{-1}\mathrm{Mpc}$ if we demand an

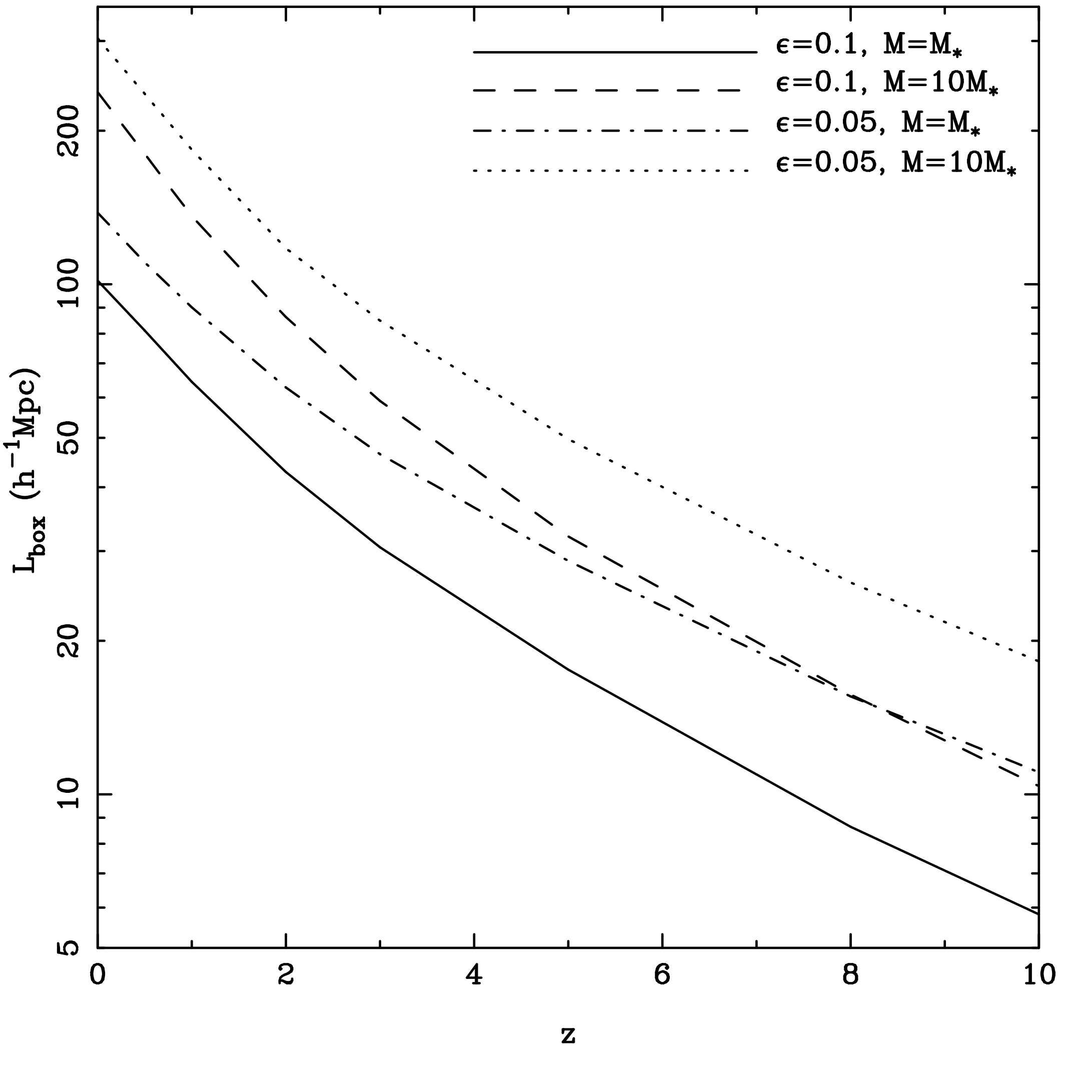

Figure 6.6: We have plotted L_{box} as a function of redshift z for the ΛCDM model (see §2 for the values of the cosmological parameters). Using a smaller boxsize than that prescribed here leads to a greater underestimation of the collapsed mass in haloes than that given by the conditions used to generate these curves.

$\epsilon = 0.05$ instead. A box of this size or an even larger simulation box should be used if we wish to study the inter-galactic medium as there will be more matter left in uncollapsed regions if we use a smaller simulation box. Bright Lyman break galaxies are likely to be in more massive haloes than typical ones and a boxsize of $60\mathrm{h}^{-1}$Mpc or larger is needed to study these in an N-Body simulation.

At higher redshifts, simulations are often used for studying the reionisation of the Universe. At $z = 10$, $M_* = 2 \times 10^6 \mathrm{M}_\odot$. Sources of ionising radiation are likely to reside in much more massive haloes and hence we should use a simulation box that is at least $20\mathrm{h}^{-1}$Mpc across ($M = 10M_*$ and $\epsilon = 0.05$). If we relax the requirement to $\epsilon = 0.1$, then the simulation volume should be more than $10\mathrm{h}^{-1}$Mpc across.

Clearly, the requirement that the mass in collapsed haloes should not depend significantly on scales larger than the simulation box is fairly stringent and in some cases it may make it difficult to address the physical problem of interest. This restriction is less stringent than requiring that the correlation function of haloes converge as it has been shown that the mass function converges well before the correlation function of haloes does (Gelb and Bertschinger, 1994a). However, if the mass function has converged, tools like MAP (Tormen and Bertschinger, 1996) can be used to obtain the correct distribution of haloes. Our criterion can be used with different requirements on the convergence of collapsed mass as long as these requirements are in consonance with the physical application for which the simulation is being run.

The curves for $\epsilon = 0.1$ are not parallel to the curves for $\epsilon = 0.05$. This is due to a dependence on the shape of the power spectrum. This dependence is brought out very clearly in fig.6.7. Here we show L_{box} as a function of the index of the power spectrum for power law models with $P(k) = Ak^n$. The power spectrum is normalised such that $\sigma_L(r = 1) = 1$. We have assumed an Einstein-de Sitter background for these models. The solid curve ($\epsilon = 0.1$) and the dot-dashed curve ($\epsilon = 0.05$) are for $M = M_*$; the dashed curve ($\epsilon = 0.1$) and the dotted curve ($\epsilon = 0.05$) are for $M = 5M_*$. It is clear from this figure that it is difficult to explore the highly non-linear regime for smaller values of n

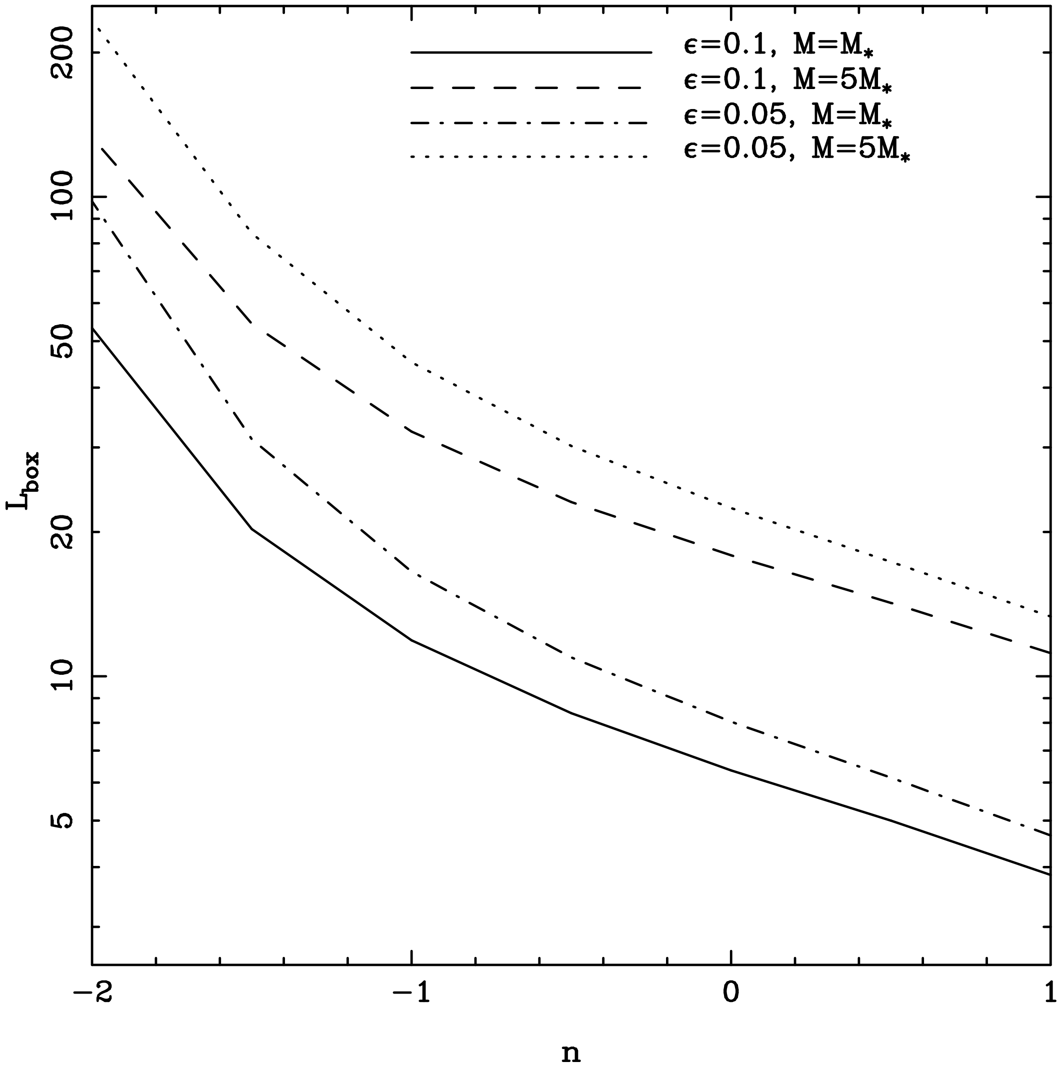

Figure 6.7: L_{box} is shown as a function of the index of the power spectrum for power law models with $P(k) = Ak^n$. The power spectrum is normalised such that $\sigma_L(r = 1) = 1$. We have assumed an Einstein-de Sitter background for these models. It is clear from this figure that it is difficult to explore the highly non-linear regime for smaller values of n.

because the required boxsize increases rapidly as $(n+3) \rightarrow 0$.

These figures show that the convergence of collapsed mass in haloes happens slowly for most models of interest. The offset in collapsed mass for the the simulated model (with a finite sampling of scales) from the collapsed mass for the model that we wish to simulate is a useful indicator of the relevance of large scales. This fact allows us to use our criterion to decide the appropriate size of the simulation volume; the criterion supplies the lower bound on the size.

6.3 Summary and Conclusions

We have studied the influence of long wave modes on gravitational clustering at small scales. We find that for the ΛCDM model scales larger than $100\mathrm{h}^{-1}$Mpc affect the mass function of haloes and the distribution of matter at scales as small as a few Mpc. The effect of long wave modes not present in an N-Body simulation can be incorporated independently of the evolution at small scales for some quantities, but making such corrections is not possible in general. In particular, it is not possible to make corrections if the contribution of large scales changes the mass of collapsed haloes by a significant amount. We can turn this argument around and check whether a given size of the simulation volume can give (close to) correct results for the total collapsed mass in haloes or not. This can be done using the simple analytical formulae outlined in §3 of this chapter. The fractional deviation of collapsed mass from its expected value, if density fluctuations at all scales are taken into account, is a good indicator of the influence of large scales. The fractional deviation (ϵ) can be checked at a given mass scale M of physical interest. Given a choice of M, ϵ and the redshift z at which the output of the simulation is to be studied, we can compute the recommended minimum size for the simulation volume. It is important to note that this scale is the minimum required and a larger simulation may be warranted by other requirements.

The mass in collapsed haloes converges faster than the correlation function of haloes implying that an even larger simulation volume may be required. Corrections to positions and velocities due to large scales can be made using tools such as MAP (Tormen and Bertschinger, 1996; Cole, 1997). Thus we can choose the boxsize by requiring convergence of mass in haloes (unless one can make a convincing case that the relevant physical quantities under study are not as sensitive to the size of the simulation box as the mass in collapsed haloes) and then use MAP to get the correct distribution and velocity field at all scales. Of course, if MAP is not being used, a larger boxsize than the minimum indicated by the convergence of the mass function may be required. Though using a large boxsize can make it difficult to address several interesting questions using N-Body simulations, we believe that it is better to be cautious.

Chapter 7

Scaling Relations for Two-Dimensional Collapse

Non-linear scaling relations indicate that there are three distinct regimes in the evolution of gravitational clustering. The asymptotic or the highly non-linear regime is the primary focus of our attention in this chapter of the thesis. Several earlier studies have analysed the evolution of clustering in the asymptotic regime, mainly with the help of N-Body simulations (Peacock and Dodds, 1994; Jain, Mo and White, 1995; Colombi, Bouchet and Hernquist, 1996; Padmanabhan et al., 1996; Peacock and Dodds, 1996; Jain, 1997; Kanekar, 2000; Smith et al., 2003). Key conclusions about the asymptotic regime can be summarised as follows :

- Stable clustering is not reached in the range of non-linearities explored by N-Body simulations. However, the departure from stable clustering for most models is small.
- Gravitational clustering does not erase the memory of initial conditions, i.e. the non-linear power spectrum is not independent of the linear power spectrum upto the extent non-linearities have been probed. There is no universal asymptotic regime.

One of the main reasons behind the inability to resolve the issue of the asymptotic regime has been the limited dynamic range of N-Body simulations. As discussed in Chapter 5, the problem of dynamic range can be circumvented by simulating a two-dimensional system (Bouchet, Adam and Pellat, 1985; Engineer, Srinivasan and Padmanabhan, 1999; Padmanabhan and Kanekar, 2000). Non-linear scaling relations are likely to have features independent of dimension. In fact, the scaling relations in the quasi-linear regime were predicted (Padmanabhan, 1996) for two-dimensional clustering well before these could be tested in simulations (Bagla, Engineer and Padmanabhan, 1998). It is therefore reasonable to expect that, if we can understand the nature of the asymptotic regime in two dimensions, it will help us solve the problem in three dimensions, even if we cannot map the two-dimensional solution directly to the full problem in three dimensions.

Previous studies of gravitational clustering in two dimensions concluded that there is no stable clustering (Bagla, Engineer and Padmanabhan, 1998; Munshi et al., 1998). Both the studies referred to here were limited to $\bar{\xi} \ll 100$ and the dynamic range in the non-linear regime was limited. In this work[1] we revisit the same issues using the 2d TreePM code (Ray, 2004). The TreePM code has a better force resolution compared to a PM code (Bouchet, Adam and Pellat, 1985; Ray, 2004); therefore our simulations have a significantly larger dynamic range over which we can study the asymptotic regime.

A related issue is that of the density profiles of massive haloes. Non-linear scaling relations in the asymptotic regime depend on the kind of dynamical equilibrium that is reached in massive haloes, if an equilibrium is ever reached. Density profiles of massive haloes can be related to the initial conditions and also to the dynamics within these haloes (Filmore and Goldreich, 1984; Subramanian, 2000). It has been claimed that gravitational clustering in hierarchical models leads to a universal density profile (Navarro, Frenk and White, 1996) which is independent of the initial conditions. This density profile, often referred to as the NFW profile, is given by

$$\rho_{_{NFW}}(r) \propto \frac{1}{r(r^2 + r_{200}^2)} \tag{7.1}$$

Here r_{200} is a scale radius. This profile is characterised by a r^{-3} decline at large radii and a cuspy inner profile of the form $\rho(r) \propto 1/r$. The claim of universality has been tested

[1] This chapter is based on *Gravitational Collapse in an Expanding Universe : Scaling Relations for Two-Dimensional Collapse Revisited* (S. Ray, J.S. Bagla and T. Padmanabhan), astro-ph/0410041.

(Navarro, Frenk and White, 1996; Cole and Lacey, 1996; Fukushige and Makino, 1997; Tormen, Bouchet and White, 1997; Moore et al., 1998; Kravtsov et al., 1998; Ghigna et al., 2000; Subramanian, Cen and Ostriker, 2000; Klypin et al., 2001; Power et al., 2003; Fukushige, Kawai and Makino, 2004) and the NFW profile is found to be consistent with the density profiles of massive haloes in N-Body simulations, even though it may not be the best fit. This essentially implies that there is a scatter in the slopes of density profiles obtained and one can fit different functional forms to the N-Body data. The possibility of existence of NFW-like profiles implies that there are some universal aspects of gravitational clustering in an expanding Universe. There is no *ab initio* derivation of the NFW profile, but one can argue that, if there is a universal profile, it should have same asymptotes as the form shown above (Syer and White, 1998).

7.1 The Scaling Relations

In this section, we review the scaling relations in gravitational clustering in both two and three dimensions.

7.1.1 Scaling Relations in Three Dimensions

Given an initial power spectrum, one can compute the linearly extrapolated two-point correlation function at any epoch. Scaling relations are a prescription for relating the non-linear correlation function and the linear correlation function and are therefore an important tool in understanding non-linear gravitational clustering.

The scaling relations are written for the averaged correlation function given by

$$\bar{\xi}(t,x) = \frac{3}{x^3}\int_0^x \xi(t,y)y^2 dy \tag{7.2}$$

The average number of neighbours within a distance x of a particle is

$$N(x,t) = (na^3)\int_o^x 4\pi y^2 dy[1+\xi(y,t)] \tag{7.3}$$

where n is the comoving number density. The conservation law for pairs (Peebles, 1980; Nityananda and Padmanabhan, 1994) is

$$\frac{\partial \xi}{\partial t} + \frac{1}{ax^2}\frac{\partial}{\partial x}[x^2(1+\xi){v_p}^{rad}] = 0 \tag{7.4}$$

where ${v_p}^{rad}(t,x) = \mathbf{v}_p \cdot \hat{\mathbf{x}}$ denotes the radial component of the average relative velocity of pairs $\mathbf{v}_p$ at separation x and epoch t. Using

$$(1+\xi) = \frac{1}{3x^2}\frac{\partial}{\partial x}[x^3(1+\bar{\xi})] \tag{7.5}$$

in (7.4), we get

$$\frac{1}{3x^2}\frac{\partial}{\partial x}[x^3\frac{\partial}{\partial t}(1+\bar{\xi})] = -\frac{1}{ax^2}\frac{\partial}{\partial x}\left[\frac{{v_p}^{rad}}{3}\frac{\partial}{\partial x}[x^2(1+\bar{\xi})]\right]. \tag{7.6}$$

Integrating, we find that

$$x^3\frac{\partial}{\partial t}(1+\bar{\xi}) = -\frac{{v_p}^{rad}}{a}\frac{\partial}{\partial x}[x^3(1+\bar{\xi})] \tag{7.7}$$

(The integration allows the addition of an arbitrary function of t on the right hand side. This function is set to zero so as to reproduce the correct limiting behaviour at the linear end ($\bar{\xi} \ll 1$), where ${v_p}^{rad}/\dot{a}x = -\,(2/3)\,\bar{\xi}$ for an Einstein de-Sitter Universe.) We change variables from t to a :

$$a\frac{\partial}{\partial a}[1+\bar{\xi}(a,x)] = \left(\frac{{v_p}^{rad}}{-\dot{a}x}\right)\frac{1}{x^2}\frac{\partial}{\partial x}[x^3(1+\bar{\xi}(a,x))] \tag{7.8}$$

Defining $h(a,x) = -({v_p}^{rad}/\dot{a}x)$, we get

$$\left(\frac{\partial}{\partial \ln a} - h\frac{\partial}{\partial \ln x}\right)\,(1+\bar{\xi}) = 3h\left(1+\bar{\xi}\right) \tag{7.9}$$

This equation shows that the evolution of $\bar{\xi}(a,x)$ is essentially decided by the function h.

In the extreme non-linear limit $\left(\bar{\xi} \gg 1\right)$, it is expected that bound structures will form and will have constant proper sizes (Peebles, 1980; Nityananda and Padmanabhan, 1994; Padmanabhan, 2000). To maintain a stable structure, the radial component of the relative pair velocity ${v_p}^{rad}(a,x)$ of particles separated by x should balance the Hubble velocity $Hr = \dot{a}x$ at the same scale; hence ${v_p}^{rad} = -\dot{a}x$, or $h(a,x) = 1$ (the stable clustering hypothesis). The behaviour of $h(a,x)$ for an Einstein de-Sitter Universe for $\bar{\xi} \ll 1$ follows from linear theory and $h = (2/3)\bar{\xi}$ in this limit. Thus $h(a,x)$ depends on (a,x) only through $\bar{\xi}(a,x)$ in the linear limit, while $h \cong 1$ is the non-linear limit. If we *assume* that h depends on a and x only through $\bar{\xi}(a,x)$, i.e. $h(a,x) = h[\bar{\xi}(a,x)]$, then we can solve eqn.7.9 :

$$\bar{\xi}_L(a,l) = \exp\left(\frac{2}{3}\int^{\bar{\xi}(a,x)} \frac{d\mu}{h(\mu)(1+\mu)}\right); \quad l = x[1+\bar{\xi}(a,x)]^{1/3} \tag{7.10}$$

Given the function $h(\bar{\xi})$, this relates $\bar{\xi}_L$ and $\bar{\xi}$. Equivalently, it gives a mapping $\bar{\xi}(a,x) = U[\bar{\xi}_L(a,l)]$ between the non-linear and linear correlation functions evaluated at scales x and l respectively. Here $\bar{\xi}_L$ is the linearly extrapolated mean correlation function. The $(2/3)$ factor in the exponent is $(2/D)$ in D-dimensions and the relation between the linear and the non-linear scales transforms to $l = x[1+\bar{\xi}(a,x)]^{1/D}$. The non-local nature of this relation represents the transfer of power in gravitational clustering and cannot be ignored or approximated by a local relation between $\bar{\xi}_{NL}(a,x)$ and $\bar{\xi}_L(a,x)$. In the linear regime $\left(\bar{\xi} \ll 1, \bar{\xi}_L \ll 1\right)$ $U(\bar{\xi}_L) \simeq \bar{\xi}_L$. The non-linear phases of evolution can be conveniently divided into two parts, the quasi-linear $(1 \leq \bar{\xi} \leq 200)$ and the highly non-linear or asymptotic $(200 \leq \bar{\xi})$ regimes. The quasi-linear regime can be called the regime of radial infall. In the non-linear or asymptotic phase virialized systems dominate. The scaling relations in three dimensions are :

$$\bar{\xi}(a,x) \propto \begin{cases} \bar{\xi}_L(a,l) & \text{(Linear)} \\ \bar{\xi}_L(a,l)^3 & \text{(Radial Infall)} \\ \bar{\xi}_L(a,l)^{3h/2} & \text{(Asymptotic limit)} \end{cases} \tag{7.11}$$

Here h is the asymptotic value of the dimensionless radial pair velocity.

7.1.2 Scaling Relations in Two Dimensions

It is possible to obtain similar relations between $\xi(a,x)$ and $\xi_L(a,l)$ in two dimensions and the relation can be obtained, as before, from the pair conservation equation (Peebles, 1980; Nityananda and Padmanabhan, 1994) :

$$\frac{\partial D}{\partial A} - h(A,x)\frac{\partial D}{\partial X} = 2h(A,X) \tag{7.12}$$

Here $D = \log(1+\bar{\xi})$, $X = \log(x)$ and $A = \log(a)$. The characteristics of this equation are

$$x^2 \left[1 + \bar{\xi}(x,a)\right] = l^2 \tag{7.13}$$

Self-similar radial collapse models (Filmore and Goldreich, 1984) for collapse of cylindrical perturbations show that the turn-around radius for a shell and the initial density contrast inside the shell are related as $x_{\rm ta} \propto l/\bar{\delta}_i \propto l/\bar{\xi}_L(l)$. Noting that in two dimensions mass enclosed in a shell $M \propto x^2$, we find $\bar{\xi}(x) \propto \left[\bar{\xi}_L(l)\right]^2$ in the regime dominated by infall. In 2d, the scaling relations have the form

$$\bar{\xi}(a,x) \propto \begin{cases} \bar{\xi}_L(a,l) & \text{(Linear)} \\ \bar{\xi}_L(a,l)^2 & \text{(Radial Infall)} \\ \bar{\xi}_L(a,l)^h & \text{(Asymptotic limit)} \end{cases} \tag{7.14}$$

h is again the asymptotic value of the radial pair velocity. The usual stable clustering limit is $h = 1$. In such a case, or for any value of h that does not depend on the initial conditions, slope of the non-linear correlation function is a unique function of the slope of the initial correlation function. For a general h, we can relate the slope of the correlation function in the asymptotic regime to the slope of the initial linear correlation function :

$$\bar{\xi}(a,x) \propto x^{2h(n+2)/(2+h(n+2))} \tag{7.15}$$

If, however, $h(n+2) =$ constant, then the non-linear correlation function has the same slope, independent of the initial correlation function. This will happen if gravitational clustering erases all memory of initial conditions. Note that this will make h and hence the velocity fields a function of the initial conditions.

In an earlier work, Bagla, Engineer and Padmanabhan (1998) had concluded that $h \simeq 0.75$ in the asymptotic regime. In the same work as well as in investigations by others (Munshi et al., 1998), the prediction for the infall-dominated quasi-linear regime had been confirmed. In this work, we would like to address the following two questions :

- does the asymptotic value of h scale with n such that $h(n+2) =$ constant ? or
- does h reach a universal value independent of n ? If so, then what is this value ?

7.2 Numerical Experiments

We carried out a series of numerical experiments with power law models with indices $n = -0.4$, 0.0 and 1.0. It is difficult to reach the asymptotic regime for $(n+2) \rightarrow 0$ before the perturbations at the box scale become important and hence we have not used models with a more negative index. Models were normalised so that $\Delta^2(k = 8k_f, a = 1) = 1$ where k_f is the wave number of the fundamental mode; essentially the fluctuations at 1/8 of the box size were normalised to unity at $a = 1$. Simulations were run upto a sufficiently late epoch while requiring that the fluctuations at half the box scale were well within the linear regime at the final epoch, i.e. $\Delta^2(k = 2k_f, a_{fin}) \ll 1$.

Simulations discussed here had 2048^2 particles on a 2048^2 grid in an Einstein deSitter background Universe. The 2d TreePM code was used for all the simulations, though we have checked many of the results in 4096^2 Particle-Mesh simulations as well.

7.2.1 The Two-Point Correlation Function

We find that the evolution of the system is self similar - all the relevant quantities have the same form when scaled by $r_{nl} \propto a^{2/(n+2)}$, where r_{nl} is the scale at which the amplitude of density fluctuations is unity. Fig.7.1 shows the averaged correlation function $\bar{\xi}(r)$ as a function of r/r_{nl} for simulations with $n = -0.4$ (upper panel) and $n = 1.0$ (lower panel). Curves were plotted for scales larger than twice the softening length used in the simulations in order to keep out features that depend on the choice of the softening length. The evolution is self-similar upto and beyond $\bar{\xi} = 100$.

The shape of the correlation function $\bar{\xi}(r)$ for the quasi-linear regime is consistent with the results of earlier studies (Bagla, Engineer and Padmanabhan, 1998; Munshi et al., 1998). There was disagreement between the above-mentioned studies for the asymptotic regime : one study (Bagla, Engineer and Padmanabhan, 1998) favoured $h \simeq 0.75$ and the other found $h(n+2) = 1.3$ (Munshi et al., 1998). The latter behaviour erases the memory

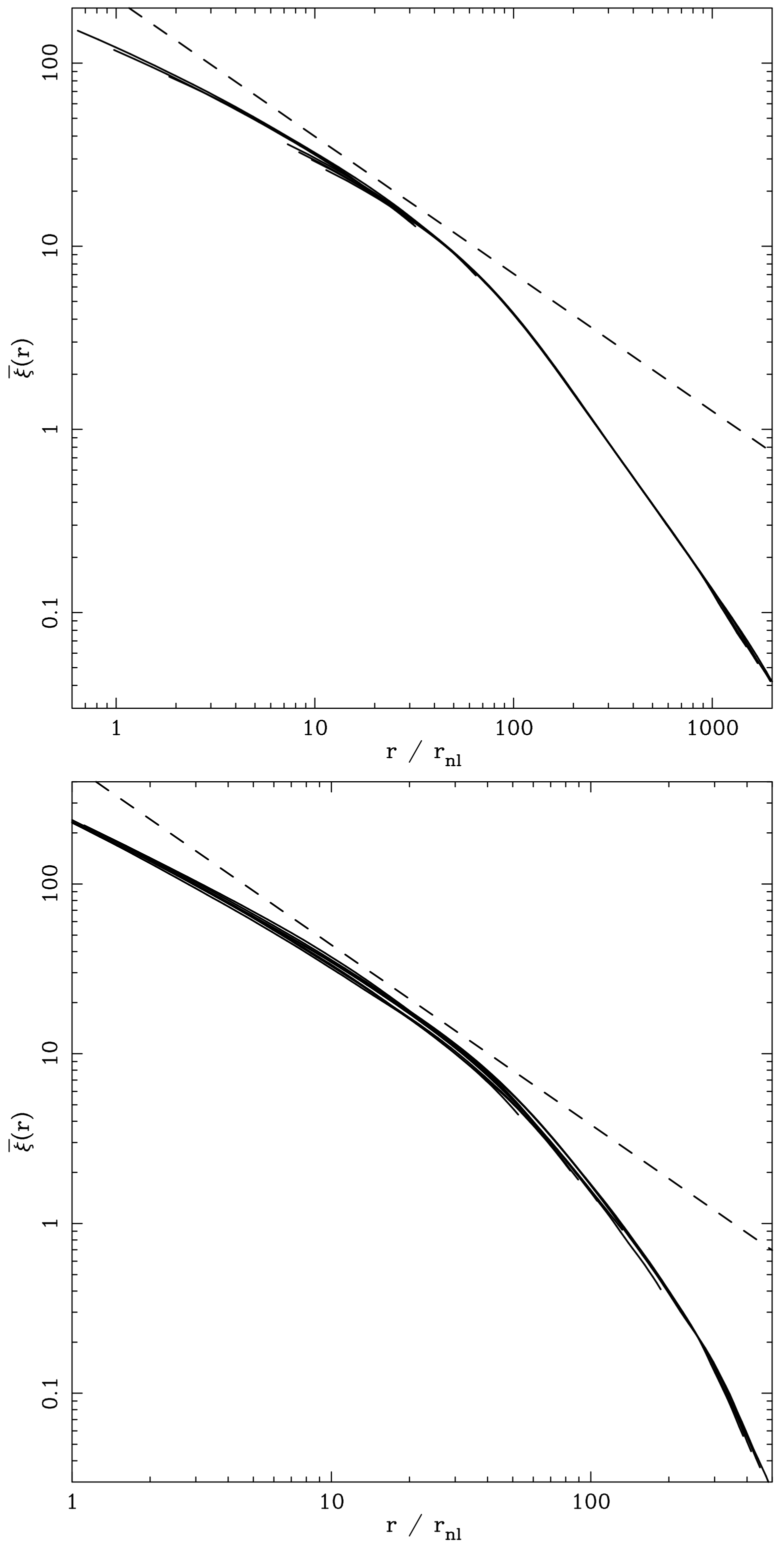

Figure 7.1: $\bar{\xi}$ is plotted as a function of r/r_{nl} for simulations with spectral index $n = -0.4$ (upper panel) and $n = 1.0$ (lower panel). The curves span more than a decade in scale factor. Clearly, over the entire range of clustering, the evolution is self-similar. Dashed lines in these panels shows the asymptotic slope for $h = 0.75$, a result suggested by an earlier study (Bagla, Engineer and Padmanabhan, 1998).

of initial conditions and the slope of the correlation function in the asymptotic regime is then the same for all models. The dynamic range in both the studies was narrow, being limited to $\bar{\xi} \leq 40$.

The current work agrees with a slope of the correlation function corresponding to a value of $h = 0.75$ in the region of overlap with the earlier work by Bagla, Engineer and Padmanabhan (1998) and it rules out $h(n+2) = 1.3$. A dashed line corresponding to the expected slope of $h = 0.75$ is marked in both panels of fig.7.1 and it runs parallel to the curve upto about $\bar{\xi} \simeq 40$, the largest non-linearity studied earlier (Bagla, Engineer and Padmanabhan, 1998). As the clustering amplitude increases, the slope of the correlation function decreases below the slope expected for $h = 0.75$. The asymptotic slope of the correlation function is $-0.53 \leq \gamma \leq -0.50$ for $n = -0.4$ and $-0.80 \leq \gamma \leq -0.77$ for $n = 1$, where $\gamma = \partial \log \bar{\xi} / \partial \log r$ and is evaluated at $\bar{\xi} \geq 100$. We have given the 95% confidence limits on the slope and the limits were derived using a χ^2 fit to the correlation function in the asymptotic regime. Different values of γ imply departure from the asymptote with $h(n+2) =$ constant. The range of values of γ listed here are consistent with $0.416 \leq h \leq 0.451$ for $n = -0.4$ and $0.417 \leq h \leq 0.444$ for $n = 1$. Thus we recover a similar range of asymptotic values for h from the correlation function. Slope of the correlation function is a more stable estimator of h than a direct determination.

Fig.7.2 shows the scaling relations for two-dimensional gravitational clustering. $\bar{\xi}(x, a)$ is plotted as a function of $\bar{\xi}_L(l, a)$ for $n = -0.4$, $n = 0$ and $n = 1$. Data from multiple epochs have been used here. Curves marking the linear, quasi-linear and the asymptotic regimes are shown in the figure. We have also marked a line showing the stable clustering limit. The equations for the piecewise fit are :

$$\bar{\xi}(a,x) = \begin{cases} \bar{\xi}_L(a,l) & \bar{\xi}_L(l) \leq 0.5;\ \bar{\xi}(x) \leq 0.5 \\ 2\ \bar{\xi}_L(a,l)^2 & 0.5 \leq \bar{\xi}_L(l) \leq 2;\ 0.5 \leq \bar{\xi}(x) \leq 8 \\ 10.5\ \bar{\xi}_L(a,l)^{0.45} & 10 \leq \bar{\xi}_L(l);\ 22 \leq \bar{\xi}(x) \end{cases} \qquad (7.16)$$

It is remarkable that the points for all the three models follow the same scaling relations in all the three regimes. The asymptotic slope implied by the scaling relation lies near $h = 0.45$, but it is difficult to say whether this is the final value or if it would continue

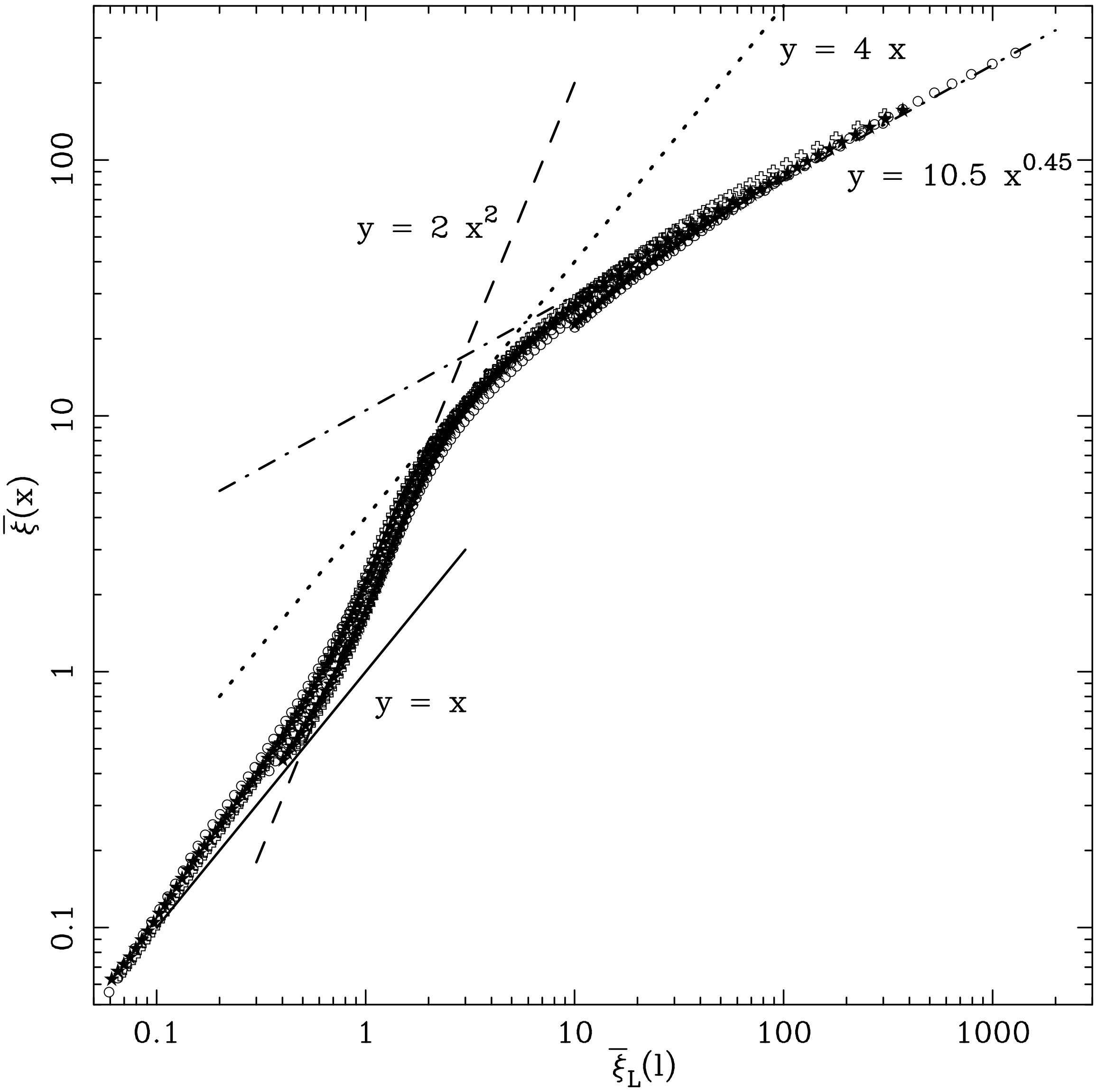

Figure 7.2: This figure shows the scaling relations for two-dimensional gravitational clustering. $\bar{\xi}(x, a)$ is plotted as a function of $\bar{\xi}_L(l, a)$ for $n = -0.4$, $n = 0$ and $n = 1$. Data from multiple epochs has been used here. Curves marking the linear (solid curve), quasi-linear (dashed curve) and the asymptotic regime (dot-dashed curve) are shown. We have also marked a dotted line showing the stable clustering limit. A remarkable feature of clustering in two dimensions is that the non-linear correlation function in the asymptotic regime is *smaller* than the linear correlation function.

to decrease as we go to higher non-linearities. Note that the value of h favoured by the data at $\bar{\xi} \geq 100$ is slightly lower, as is clear from the values obtained from the slope of the correlation function. This figure is clearly inconsistent with an asymptote of $h(n+2) =$ constant as the scaling relation for different models would have been different in that case.

The results shown here were obtained with the TreePM code and they match with the correlation function obtained in 4096^2 Particle-Mesh simulations. The comparison was carried out up to $\bar{\xi} = 70$.

7.2.2 The Radial Pair Velocity

Fig.7.3 shows the pair velocity $h(r, a)$ as a function of $\bar{\xi}(r, a)$ for several epochs for $n = -0.4$ (upper panel) and $n = 1.0$ (lower panel). The dashed line shows the expected value of h in the linear limit. For the linear and quasi-linear ($\bar{\xi} \leq 10$) regimes, h is a single-valued function of $\bar{\xi}$, irrespective of the epoch for a given model. Indeed, we find that there is no significant difference between the curves for different models at $\bar{\xi} \leq 10$. At higher non-linearities, there is scatter in the curves so that it is difficult to test any claims using the figure. Thus the ansatz that h depends on epoch and scale *only through* $\bar{\xi}$ (Nityananda and Padmanabhan, 1994) is consistent with our simulation data and this allows us to find the form of the scaling relations. Conversely, non-linear scaling relations can be used to find $h(\bar{\xi})$. We have checked that the scaling relations plotted in fig.7.2 are consistent with the $h(\bar{\xi})$ curves plotted in fig.7.3.

The value of h fluctuates a lot at late epochs and, as we mentioned in the previous section, the slope of the correlation function gives a more stable estimate of h. The key feature in this regime is that $h < 1$ and it does not show any sign of heading towards the stable clustering limit of $h = 1$. This is consistent with our results for the slope of correlation function that we obtained in the previous section.

It is important to understand what $h < 1$ means in terms of dynamics. One possibility

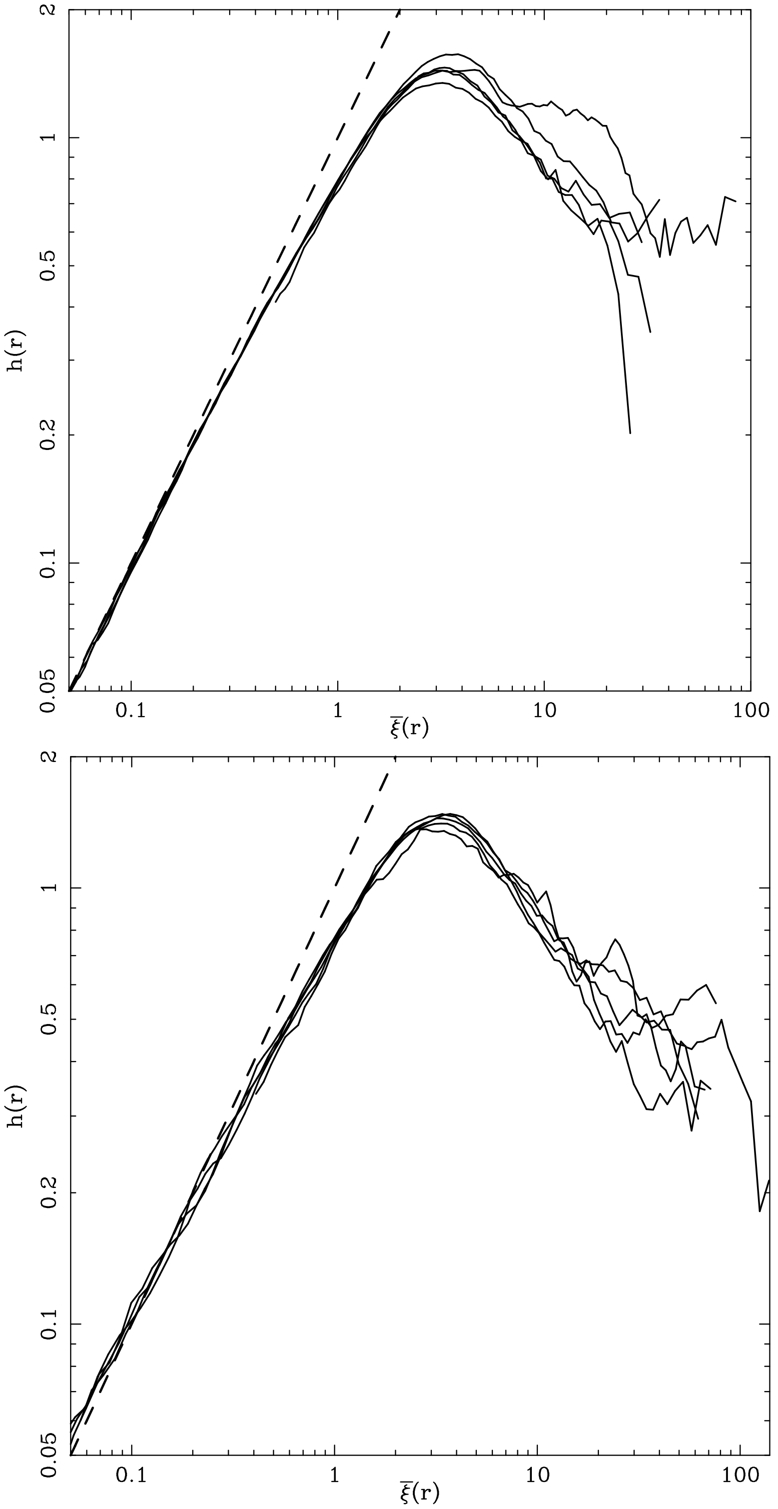

Figure 7.3: The pair velocity $h(r, a)$ is plotted as a function of $\bar{\xi}(r, a)$ for a large number of epochs a for models with $n = -0.4$ (upper panel) and $n = 1.0$ (lower panel). The dashed line shows the expected value of h in the linear limit. The $h - \bar{\xi}$ curves are the same across all epochs and for all models within the scatter.

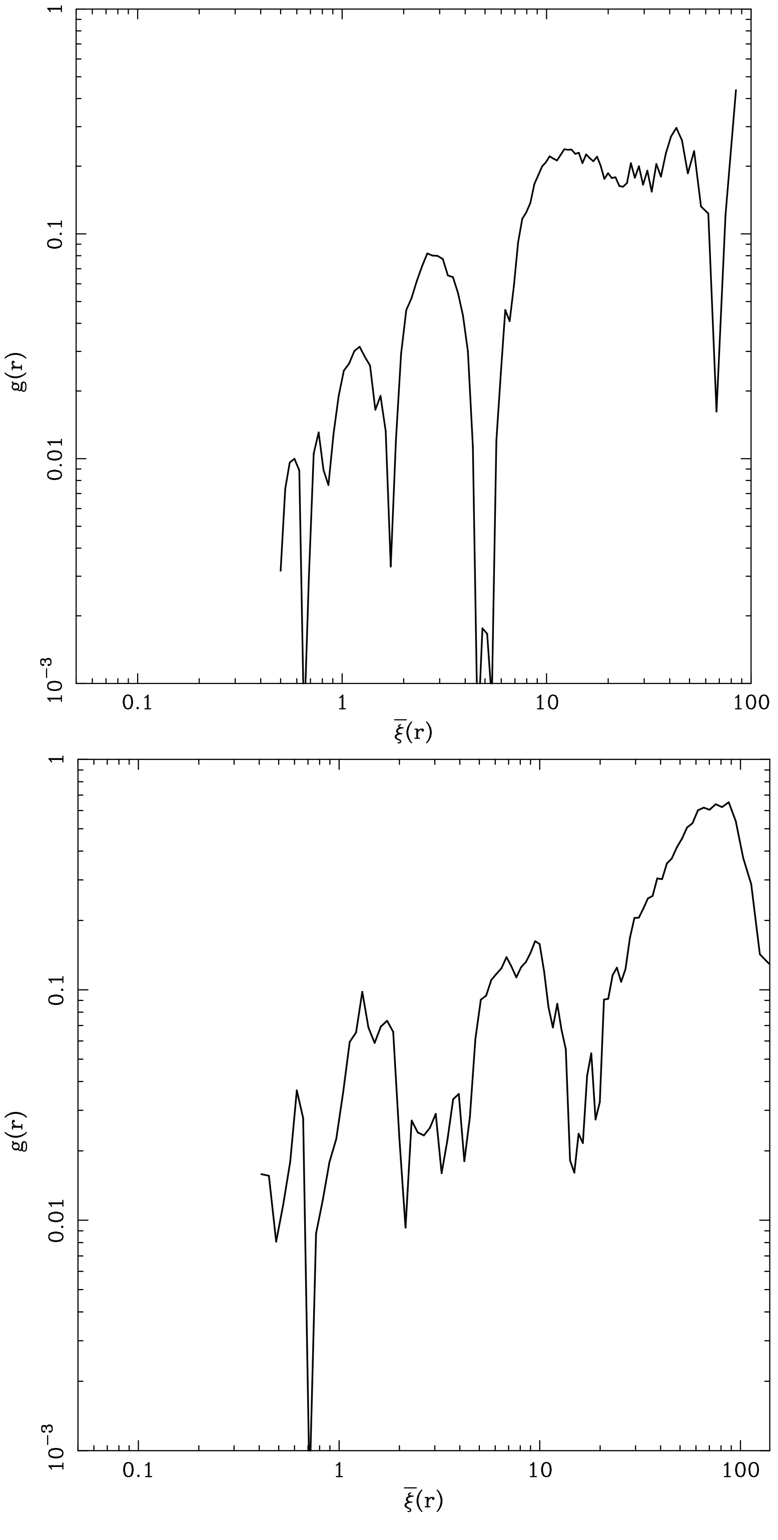

Figure 7.4: The transverse component of the pair velocity $g(r, a)$ is plotted as a function of $\bar{\xi}(r, a)$ for a late epoch for models with $n = -0.4$ (upper panel) and $n = 1.0$ (lower panel). The magnitude $|g|$ is plotted here; a remarkable fact apparent from this figure is that the transverse component is more important for $n = 1$. At $\bar{\xi} \gg 1$, the magnitude of g is comparable with that of h.

is that haloes are evaporating. This can happen if two-body relaxation is important; but we have used the softening length $\epsilon \geq 0.2$ grid lengths in all our simulations and the number density of particles in most haloes in our simulations is very high. Hence two-body scattering should not be important. A poor integrator for the equation of motion can also lead to evaporation from haloes. We have tested the integrator that we use in three-body problems and with highly eccentric binary orbits and this possibility can also be ruled out.

7.2.3 Non-Radial Motions

Now we study the transverse counterpart of h. Analogous to h, we can define the transverse pair velocity function as

$$\mathbf{g}(\mathbf{r}, a) = -\left(\mathbf{v}_p \times \mathbf{r}\right)/Hr^2 \tag{7.17}$$

In two dimensions, it takes the following simple form :

$$g(r, a) = -\frac{v_{p_x} y - v_{p_y} x}{Hr^2} \tag{7.18}$$

Here $\mathbf{v}_p$ is the mean velocity of pairs, $\mathbf{r} \equiv (x, y)$ the separation of pairs of particles and H, as usual, the Hubble parameter. In two dimensions, the "cross" product of two vectors is just a number. Fig.7.4 shows $|g|$ as a function of $\bar{\xi}$. In the figure, we have plotted the magnitude of $\mathbf{g}(\mathbf{r}, a)$ as its value oscillates about zero at large scales. The curve is plotted for only one epoch, but the relative value of $|g|$ compared to h shows that this is a significant component in the non-linear region. Note that we require coherent transverse motions in order to detect anything here as random transverse motions will cancel out otherwise. We have also studied velocity dispersions and we do not find any systematic excess in the pairwise transverse velocity dispersion in comparison to the pairwise collinear velocity dispersion. In dynamical equilibrium these should have the same magnitude in two dimensions and at $\bar{\xi} \geq 10$ we find this to be true.

To investigate this further, we have studied motions within individual haloes by plotting the angular momentum profile of a halo as a function of distance from the centre

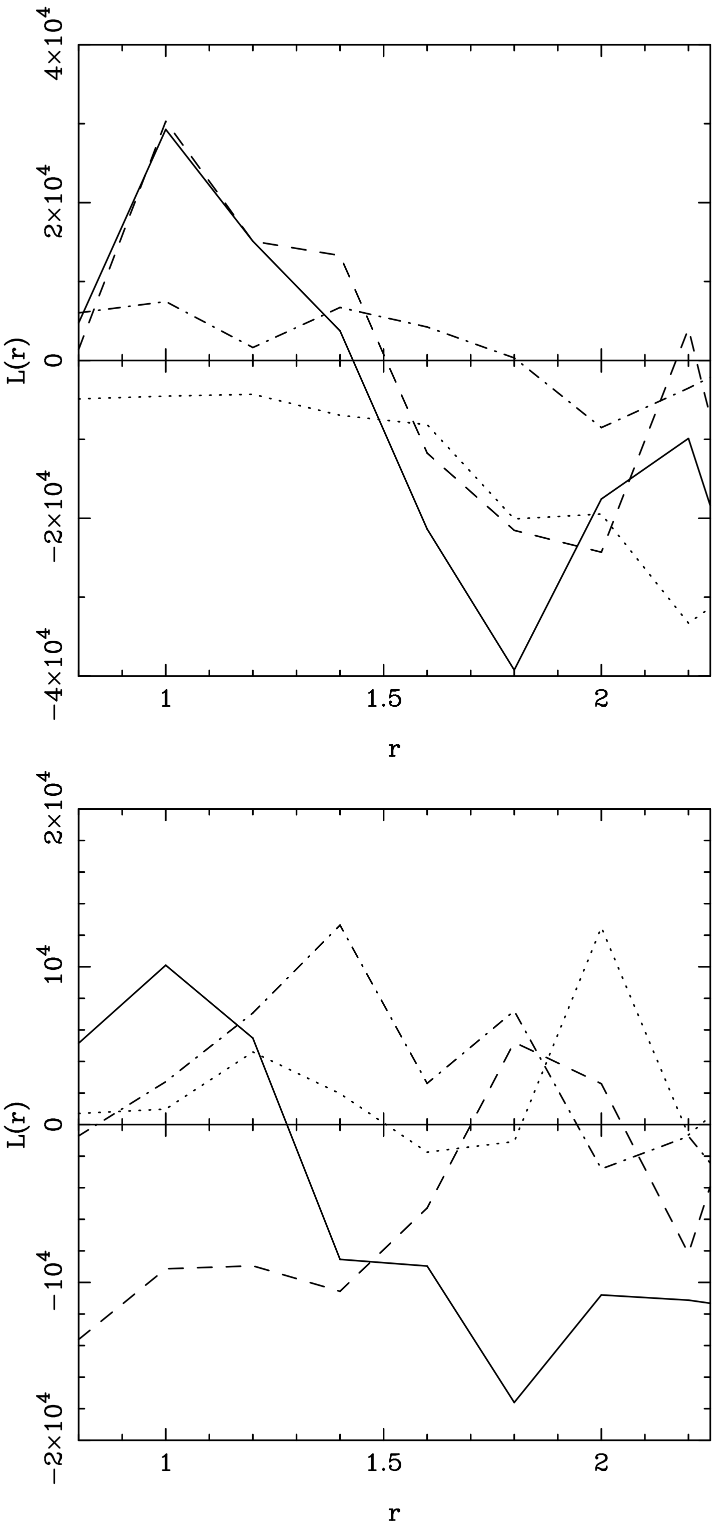

Figure 7.5: In this figure we plot the circularly averaged angular momenta of particles in a halo within a distance r from the centre of mass of the halo for some haloes. The upper panel is for index $n = -0.4$ and the lower one for $n = 1.0$. Clearly all the haloes in both panels have annuli with coherent rotations. Different line styles in the figure correspond to different haloes.

of the halo. There are no systematic features worth commenting on, except that there are annuli with coherent rotations for most haloes. Fig.7.5 shows the circularly averaged angular momenta of particles in a halo within a distance r from the centre of mass of the halo for some (randomly selected) haloes. The upper panel is for index $n = -0.4$ and the lower one for $n = 1.0$. Clearly, all the haloes in both panels have annuli with coherent rotation. (A discussion of the selection criteria adopted for haloes used here is given in the next subsection.)

It is important to understand the origin of coherent transverse motions in massive haloes. Comparing the $g - \bar{\xi}$ plots for $n = -0.4$ and $n = 1$ offers us an important clue; transverse motions are much stronger for $n = 1$ compared to the other model. Collisions between substructure falling into haloes in the early phases of evolution can generate transverse motions of this nature (Bagla, Prasad and Ray, 2004) and there is certainly more substructure in the $n = 1$ model. We put forward the hypothesis that collisions between substructure are responsible for generation of coherent transverse motions. We are testing this hypothesis in a series of numerical experiments. Results of these studies will be reported elsewhere.

7.2.4 Density Profiles of Massive Haloes

Density profiles of massive haloes and the non-linear scaling relation are closely related. Massive haloes can be studied here as they extend over a large region - large compared to the softening length. We identified haloes in our simulations using the method described in Chapter 1, §6 (Tormen, Bouchet and White, 1997). In this method, the density field is smoothed at a length scale R (we take $R = 5$ grid lengths) and peaks in the smoothed density field are identified. Particles within a distance R of a peak are selected and we find the centre of mass for these particles. We now select a smaller set of particles from this group, i.e. those that are within $R - \Delta R$ from the centre of mass and compute the centre of mass. The iterative process continues till we are left with only about a hundred particles. The centre of mass of these particles is taken to be the centre of the halo and

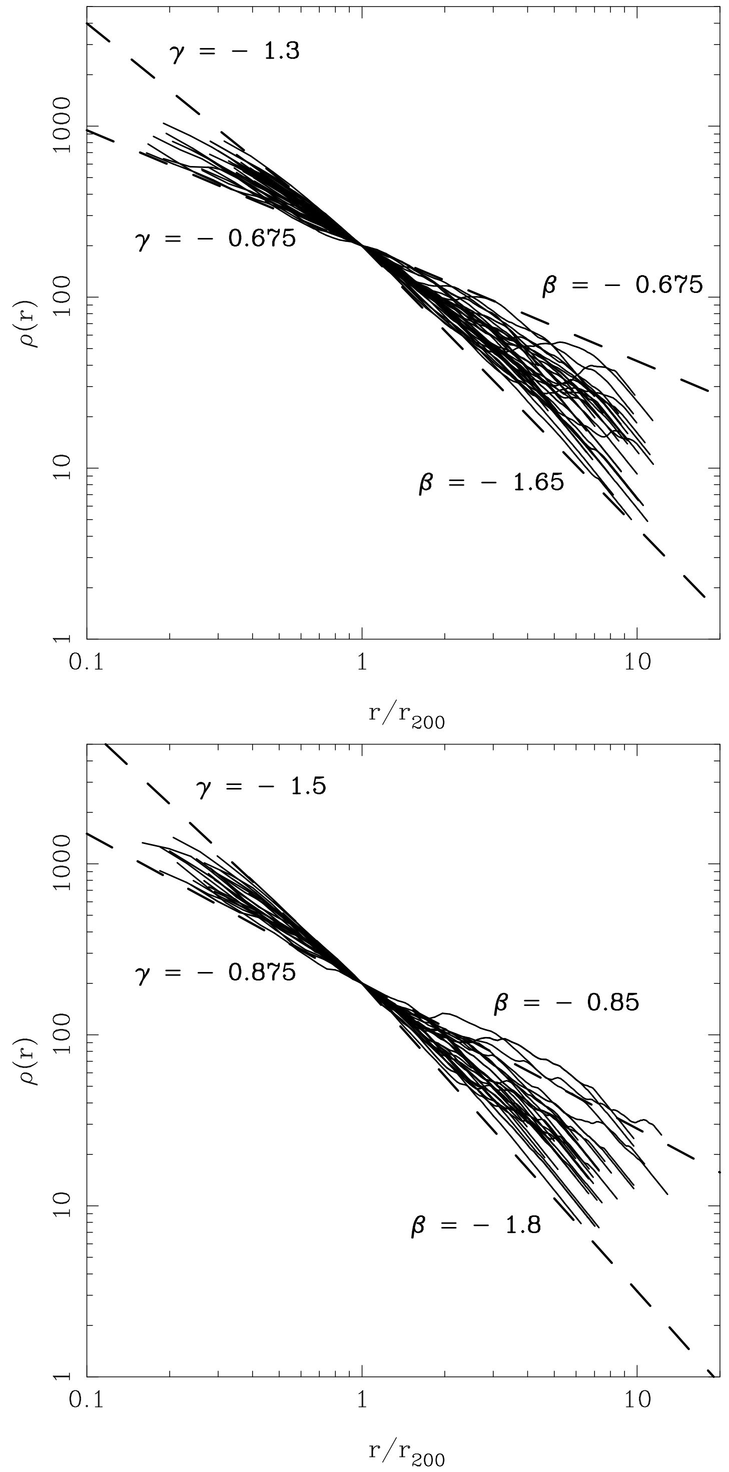

Figure 7.6: This figure shows spherically averaged density profiles of a number of spherically symmetric virialised haloes from simulations with indices $n = -0.4$ (upper panel) and $n = 1.0$ (lower panel). The density is plotted as a function of r/r_{200}, where r_{200} is the scale at which density is 200 times the average density in the simulation. The four dashed lines in each panel mark the approximate extremes of inner (γ) and outer slopes (β) for the density profiles. All the haloes in the figure were taken from TreePM simulations with 2048^2 particles.

density profiles are plotted in terms of the distance from this centre. We applied two further criteria :

- Central density contrast of the halo must be large : $\delta \geq 500$.
- Haloes and their neighbourhood should be smooth and there should be no mergers going on. To implement this in an objective manner we compute the ratio $\chi = \langle \cos^2 \phi \rangle / \langle \sin^2 \phi \rangle$ where the average is over all particles within the central region of the halo and ϕ is the position angle of a particle from the centre of mass of the halo. We require $0.9 \geq \chi \geq 1.1$ for a halo for its density profile to be used in the analysis.

Fig.7.6 shows the density profiles of haloes that satisfy these criteria. Density profiles of a large number of haloes have been plotted; for each halo we have drawn the density ρ as a function of r/r_{200}, where r_{200} is the scale at which the density is 200 times the average density, average density being unity in the simulations. The top panel shows the density profiles for $n = -0.4$ and the lower panel that for $n = 1.0$. It is clear that there is a large scatter in density profiles. The inner slope varies from -0.4 to -1.4 for $n = -0.4$, whereas it is in the range -0.6 to -1.6 for $n = 1$. The inner slope γ is obtained by fitting a power law that passes through $\rho = 200$ and $r/r_{200} = 1.0$ and only points inside of this radius are used. The distribution of the inner slopes of the density profiles is very broad. The median γ is approximately -0.9 for $n = -0.4$ and -1.1 for $n = 1.0$. Self-similar, spherical collapse predicts an inner slope of -0.89 and -1.2 for $n = -0.4$ and $n = 1$ respectively. These values are well within the scatter of the inner slopes.

Transverse motions can affect the density profiles of massive haloes in a significant manner (Subramanian, 2000). Such effects will be stronger if the transverse motions are stronger and, as is obvious from fig.7.3, transverse motions are stronger for larger n. Rotation is playing a more important role in models with more substructure and is leading to flatter density profiles. This reduces the difference between density profiles of massive haloes for these models.

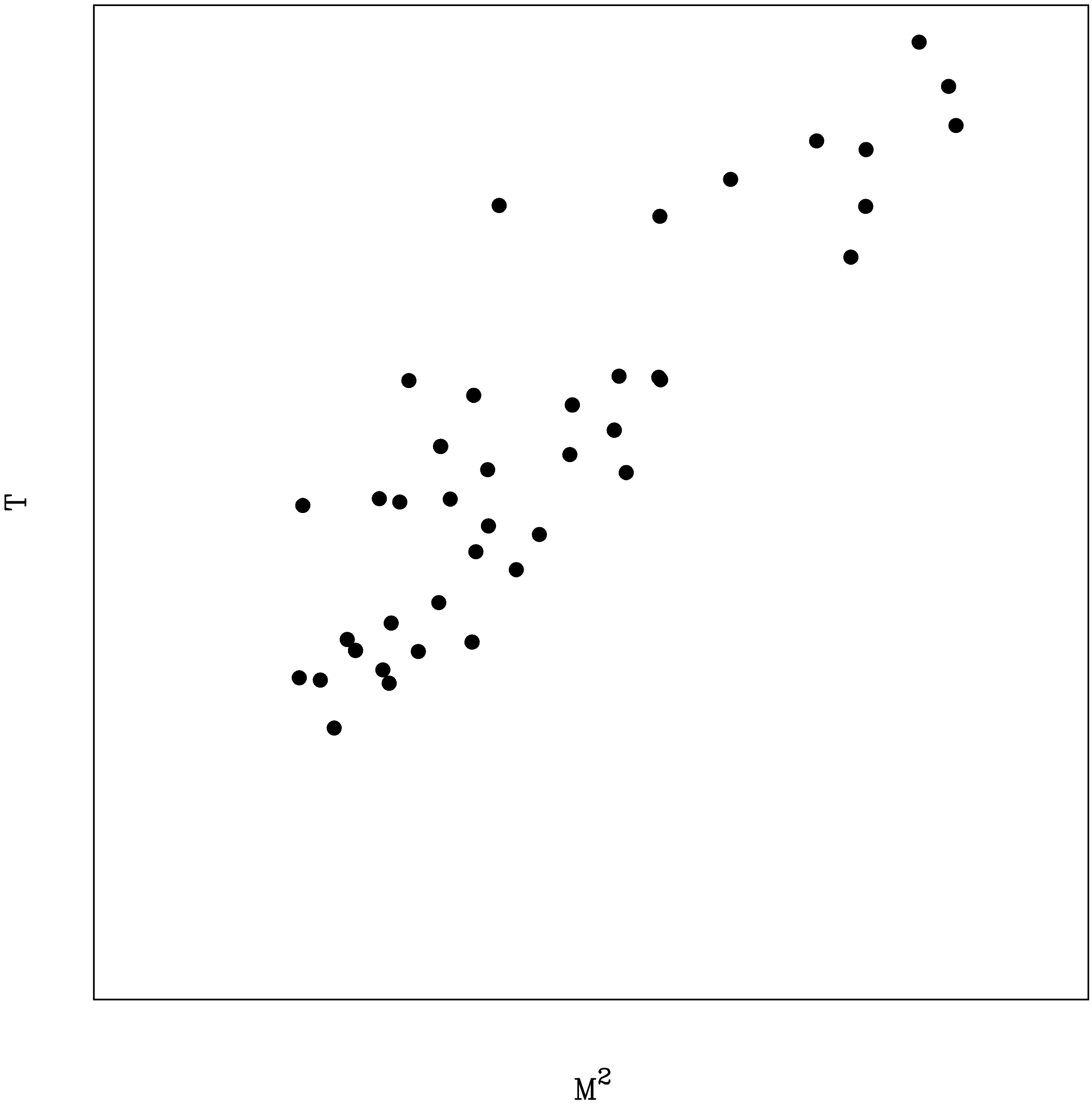

Figure 7.7: The average kinetic energy T of massive haloes is plotted against M^2 on a log-log scale. Each point on this graph represents one halo and we have used the same set of haloes that have been used for studying the density profiles. The plot shows that $T \propto M^2$ and hence the haloes are close to dynamical equilibrium.

7.2.5 Virial Equilibrium

The stable clustering limit requires the formation of objects that are in dynamical equilibrium. Whether these objects exist or not can be tested using the virial theorem in proper coordinates. In D-dimensions, the equation of motion for the ith particle is

$$m_i \ddot{\mathbf{r}}_i = -\sum_{j \neq i} \frac{G m_i m_j \left(\mathbf{r}_i - \mathbf{r}_j\right)}{\left|\left(\mathbf{r}_i - \mathbf{r}_j\right)\right|^D} \tag{7.19}$$

Taking the dot product of this equation with $\mathbf{r}_i$ and using the antisymmetry of the right hand side on (i, j), we can obtain the following :

$$\frac{1}{2}\ddot{I} = 2\sum_i \frac{1}{2} m_i \dot{\mathbf{r}}_i^2 - \frac{1}{2}\sum_i \sum_{j \neq i} \frac{G m_i m_j}{\left|\left(\mathbf{r}_i - \mathbf{r}_j\right)\right|^{D-2}} \tag{7.20}$$

where $I = \sum m_i \mathbf{r}_i^2$ is the trace of the moment of inertia tensor. Note that in $D = 3$, the second term leads to the potential energy while in $D = 2$, it is a constant unrelated to the potential energy of the configuration. In steady state, in proper coordinates, we can set the second time derivative of I to zero for a cluster of particles in dynamical equilibrium. In D=3 this gives the usual $2T + U = 0$ result, while in $D = 2$ we get :

$$\begin{aligned} 2T &= 2\sum_i \frac{1}{2} m_i \dot{\mathbf{r}}_i^2 \\ &= \frac{1}{2}\sum_i \sum_{j \neq i} G m_i m_j \\ &= \frac{1}{2} G M^2 \left(1 - \frac{1}{N}\right) \simeq \frac{1}{2} G M^2 \end{aligned} \tag{7.21}$$

where the sums are over all particles. Particles are located at $\mathbf{r}_i$ and have velocities $\dot{\mathbf{r}}_i$ and m_i is the mass of the ith particle. We assume that all the particles have the same mass to get the penultimate expression. Here N is the number of particles in a halo. For sufficiently large N, we can ignore $1/N$ in comparison with unity. Thus the kinetic energy depends *only on the total mass of the halo.* If the massive haloes that we are studying here are in virial equilibrium, then the kinetic energy T for these haloes should be proportional to M^2. We use all the particles within the radius r_{200} in a halo to compute the mass of the halo and fig.7.7 shows the two quantities (T and M^2) plotted against each other for the massive haloes that we used in our study of density profiles. It is clear that these quantities are correlated even though there is some scatter about the average relation. We may conclude that the massive haloes identified and studied in our simulations are close to dynamical equilibrium.

7.3 Summary

The basic motivation for the work presented in this chapter was to improve our understanding of the non-linear scaling relation for two-dimensional gravitational clustering. Previous studies (Bagla, Engineer and Padmanabhan, 1998; Munshi et al., 1998) had established the behaviour for the quasi-linear regime, but could not probe the nature of the asymptotic regime in detail. We used the 2d TreePM code (Ray, 2004) which allowed us to study the non-linear regime in far greater detail compared to the PM code used in earlier studies. Our results can be summarised as follows :

- The evolution of the correlation function for power law models is scale- invariant up to the highest non-linearities studied in our simulations.

- We reproduce the results of earlier studies in the quasi-linear regime. We confirm that the slope of the scaling relation in this regime is close to 2, as predicted on the basis of the infall-dominated growth model (Padmanabhan, 1996).

- We do not find any difference between the scaling relations for different power law models. Thus the information about initial conditions is retained even in the extremely non-linear regime.

- The stable clustering limit ($h \rightarrow 1$ as $\bar{\xi} \rightarrow \infty$) is not reached, but there is a model independent non-linear scaling relation in the asymptotic regime. (Though it is generally claimed that stable clustering with $h \rightarrow 1$ is a *natural* asymptotic state, it has been shown that the Davis-Peebles scale-invariant solution (Davis and Peebles, 1977) and the hierarchical model for the three-point correlation function are inconsistent with the standard stable clustering picture (Kanekar, 2000). This, of course, is for gravitational collapse in three dimensions.)

- The correlation function in the extremely non-linear regime is a less steep function of scale than reported in earlier studies. Pair velocities in the asymptotic regime are smaller than expected from the stable clustering model. We show that massive haloes have coherent rotation in annuli. This is further confirmed by a study of the transverse component of the pair velocity - the average over all particles for this quantity is non-zero and comparable in magnitude to the radial component.

- The transverse component of the pair velocity is more significant for models with a larger spectral index, i.e. for models with more substructure. This suggests that gravitational collisions between substructure might be responsible for generating coherent transverse motions.

- We find that there is no universal density profile for massive haloes in two-dimensional gravitational clustering. There is a large scatter in inner as well as outer slopes of density profiles of haloes. Median value of the inner slope of density profiles is different for different models, though the difference is much smaller than the scatter in values. Tests show that these clusters are close to dynamical equilibrium.

- The difference between the median inner slopes for the different models studied is smaller than expected from self-similar spherical collapse. We argue that the difference in significance of transverse motions for different models is a likely reason for this tendency towards convergence in the direction of a universal profile.

- We find that, in two dimensions, stable clustering implies a virial equilibrium where the kinetic energy of particles in a halo depends only on the total mass.

Further work is required to test our hypothesis that gravitational collisions between substructure are responsible for generating coherent transverse motions. Work is in progress in this direction.

Chapter 8

Patterns in Clustering of Overdense Regions

In Chapter 1 of this thesis we had introduced the concept of bias in galaxy clustering. We had defined bias as $b^2 = \bar{\xi}_{gal}/\bar{\xi}_{dm}$, where the symbols carry the same meaning as before. In early studies, the focus was on linear scales and it was believed for simplicity that b is a constant. Bias only goes to a constant at scales much larger than the inter-object separation. At small scales it is likely to be a complicated function of scale, redshift, galaxy type and cosmological parameters. Time evolution of bias can be understood in a qualitative manner (Bagla, 1998a). In this study[1] we would like to understand the relationship of galaxy clustering with the initial power spectrum of density fluctuations and our aim is to relate the underlying cosmological model to observables that quantify clustering of overdense regions (that can be mapped to collapsed structures like galaxies as a simple first approximation) through systematic N-Body experiments. The idea of bias is only an useful intermediate construct in this exercise. Given this, we need not worry about the differences in deterministic and stochastic bias (Dekel A. and Lahav O., 1999).

Several models have been constructed to understand the clustering of galaxies in terms of the distribution of collapsed objects like haloes and their properties. One of

[1] This chapter is based on *Patterns in Clustering of Overdense Regions* (J.S. Bagla and S. Ray), manuscript in preparation.

the first steps in this direction was the work by Neyman and Scott (1952). A similar approach was later used by McClelland and Silk (1977) and Peebles (1980) in developing the cluster model for galaxies. The more modern version of the same, the halo model (Mo and White, 1996; Mo, Jing and White, 1997; Sheth and Tormen, 1999; Cooray and Sheth, 2002; Smith et al., 2003), combines the extended Press-Schechter mass function with spherical / elliptical collapse (Mo and White, 1996; Sheth and Lemson, 1999; Sheth, Mo and Tormen, 2001). In this model, the large scale clustering of the mass arises through the correlations between different haloes. On small scales the correlations are derived purely from the substructure in a typical halo.

The hierarchical correlation amplitudes or the reduced moments $S_Q(r) = \bar{\xi}_Q(r)/\bar{\xi}_2(r)^{Q-1}$ (defined in Chapter 1) can be computed within the framework of the halo model for both density peaks and dark matter haloes in the quasi-linear regime (Mo, Jing and White, 1997). To calculate the higher order moments of peaks in the halo model, the general formalism of Fry and Gaztanaga (1993), where the galaxy overdensity δ_g is expanded in a Taylor series as a function of the mass overdensity $[\delta_g = F(\delta)]$, is adopted.

The predictions of the halo model have been compared with N-Body simulations (Casas-Miranda, Mo and Boerner, 2002; Smith et al., 2003). The relation between the halo autocorrelation function and the mass autocorrelation function can be calculated simply within the halo model and the results are consistent with the same obtained from N-Body simulations (Mo and White, 1996; Sheth and Lemson, 1999; Sheth, Mo and Tormen, 2001).

It has been pointed out that the degeneracy between the density parameter Ω_{nr} and b is an intrinsic feature of the linear perturbation theory and the use of second order perturbation theory and higher order statistics allows the degeneracy to be lifted (Fry and Gaztanaga, 1993; Matarrese, Verde and Heavens, 1997). Second order perturbation theory has been used to compute bias using the bispectrum in the quasi-linear regime (Matarrese, Verde and Heavens, 1997; Buchaltar and Kamionkowski, 1999). Simulations have been used to examine the effect of a bias on the power spectrum and the bispectrum in the non-linear regime. It has been found that results follow predicted trends far beyond the regime where perturbation theory is expected to be valid. The biased bispectrum continues to follow the so-called hierarchical form in the strongly non-linear regime, independent of the initial spectrum (Fry, Melott and Shandarin, 1995).

All the models that we have referred to so far concentrate on computing moments of galaxy distribution in real space, whereas observations of galaxy clustering are largely carried out in redshift space (Colless, 1998). The halo model can be adapted to include the effect of redshift space distortions (White, 2000b; Kang et al., 2002). The analytical model for the non-linear redshift space power spectra of dark matter and haloes is based on a halo prescription and has three important ingredients : the halo mass function, the density profile of haloes and the halo-halo redshift power spectrum. The predicted redshift power spectrum for haloes is found to be insensitive to the details of the halo density profile. Simulations show that the halo model works well for real space clustering, but not for the redshift space power spectrum (Kang et al., 2002). The analytical model for redshift space has not been extended to compute the higher moments of halo distribution in redshift space.

8.1 Our Methodology

There are no ab initio models for the relation between galaxy clustering and the initial power spectrum of density fluctuations. In this work we try to systematically identify generic features in the problem by studying the distributions of overdense regions and mass in a series of N-Body simulations. We compute moments for mass and overdense regions in real space and redshift space. We look for patterns in the distributions and test various hypotheses. Our focus is the non-linear regime and redshift space.

We have focused on simulations of scale invariant power spectra in an Einstein deSitter background Universe ($\Omega_m = 1.0$). Scale-invariant models do not have any intrinsic scale except for the scale of non-linearity introduced by gravity. These are ideal models for such studies as self-similar evolution is a good check for the accuracy of simulations. We identify high density regions in simulations. These are assumed to host galaxies or collapsed objects in proportion to mass. We study the distribution of mass as well as the distribution of overdense regions obtained using high density cutoffs on the distribution of mass. We also study the effect of redshift space distortions on the same. Moments of counts of objects in cells are used to compute statistics and study clustering properties. In particular, we compute the second moment $S_2 = \bar{\xi}_2$ and the third moment or the skewness $S_3 = \bar{\xi}_3/\bar{\xi}_2^2$ for various distributions.

We use the parallel TreePM code for all the simulations in this study (Bagla, 2002; Ray and Bagla, 2004). We used a cubic spline softened force with softening length $\epsilon = 0.4$. We compute all relevant statistics at scales larger than 2ϵ to avoid errors due to the softening of the force. We study clustering with initial power spectra of the form :

$$P(k) = Ak^n \tag{8.1}$$

These were normalised such that the linear fluctuations at the scale of 8 grid lengths were unity at present ($a = 1.0$). We ran simulations of these models with index $n = -1.25$, 1.0, -0.5, 0.0 and 1.0 on a 256^3 grid containing 256^3 particles. We analysed the data for epochs where $r_{nl} \sim 8$ grid lengths for all the models.

We used moments of counts of objects in cells (Peebles, 1980) to compute the volume-averaged n-point correlation functions (de Vaucouleurs, 1971; Peebles, 1971, 1980) used for the study. We computed these for the distributions of overdense regions and the full mass distribution obtained from the N-Body experiments. Fig.8.1 is a plot of S_3 vs. S_2 for the mass distribution for index $n = -1.0$. The horizontal lines on the plot correspond to the prediction from second order perturbation theory (eqns.1.74) in the weakly non-linear regime (dashed line) and the phenomenological fit to data from earlier simulations (eqn.1.75) for the strongly non-linear regime (dot-dashed line). It is seen that the distribution of mass in our N-Body simulations compares well with expectations from second order perturbation theory and also that our simulations are consistent with earlier N-Body experiments.

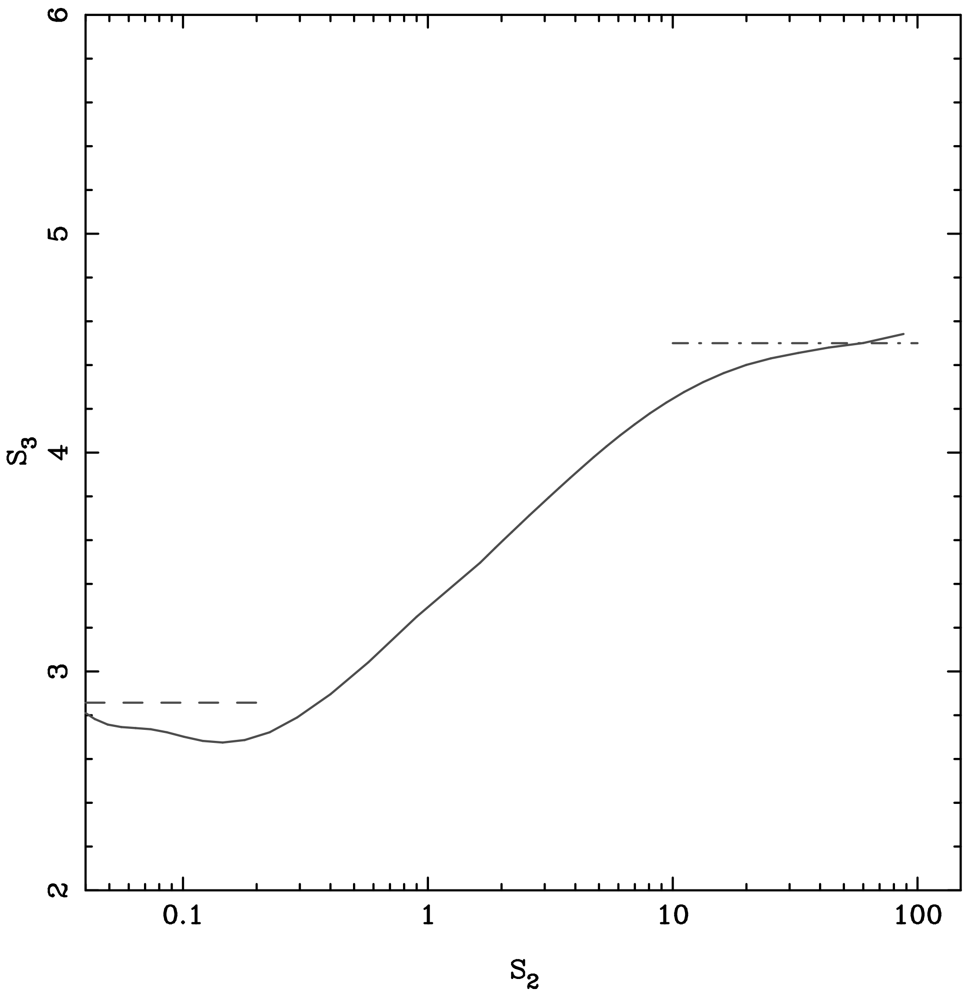

Figure 8.1: S_3 vs. S_2 for the mass distribution has been plotted for a simulation with power law index $n = -1.0$. The dashed line is the second-order perturbation theory prediction for the weakly non-linear regime, while the dot-dashed line represents a fit to data from earlier N-Body simulations in the strongly non-linear regime.

Overdense regions in the simulations are found by computing the Lagrangian density and then imposing a cutoff on the density. Fig.8.2 shows two slices (thickness 15 grid lengths) from a simulation of a power law model with index $n = 0.0$. The left panel shows all particles in the simulation in the given slice, whereas the right panel shows only particles within the slice which have Lagrangian densities at least 100 times higher than the average density ($\bar{\rho}$) in the simulation. The absence of a uniform low density background in the right panel leads to a higher degree of clustering.

We have also studied the nature of clustering of matter and overdense regions in redshift space. In redshift space, observations of clustering are distorted by peculiar motions (Kaiser, 1987) along the line of sight (see Chapter 1) and we have to transform the coordinates of each particle in a simulation following the prescription given in eqn.1.84. We assume that the size of the simulation volume at any redshift z is much smaller than the distance to the simulation volume, i.e. we use the plane-parallel approximation.

8.2 Results

We first study the variation of S_2 for mass and overdense regions with scale in real as well as redshift space for various values of the initial power spectrum n in the simulations. All the figures in this section as well as subsequent sections, unless explicitly stated, correspond to epochs where $r_{nl} \sim 8$ for all the models that we have considered. In fig.8.3, we have plotted S_2 as a function of scale r for mass and overdense regions for various values of the index n in real space. We applied cutoffs $\rho_{cut} = 100\bar{\rho}$ and $1000\bar{\rho}$ to generate the respective distributions of overdense regions. A solid line represents the mass distribution, while a dashed line represents a distribution with $\rho_{cut} = 100\bar{\rho}$ and a dot-dashed line one with $\rho_{cut} = 1000\bar{\rho}$ in all the figures. Fig.8.3 shows that high density regions cluster more strongly than the underlying mass distribution for all the models and S_2 is larger for more overdense regions. Visualising bias as the difference between the mass distribution and the distribution of overdense regions, we see that bias decreases monotonically as we increase the value of n for the same value of ρ_{cut}. Therefore, a model with $n = -1.0$ is more biased than one with $n = 0.0$. The biased distributions (i.e. the distributions of overdense regions) and the unbiased distributions (i.e. the distributions for the underlying mass) have approximately the same slope only for scales for which $S_2 \leq 1$ i.e the linear bias model holds for only large linear scales. Bias is thus scale-dependent at smaller scales.

For $n = 0.0$, we can see from fig.8.3 that the distributions for overdense regions and mass are coincident over a wide range of linear scales. This is because, for this case, for the particular epoch that we have considered, most of the mass has already collapsed into overdense regions and therefore the distributions of overdense regions are almost the same as the mass distribution (Bagla, 1998a). We have plotted S_2 as a function of scale at an earlier epoch ($r_{nl} \sim 4$) for mass and overdense regions ($\rho_{cut} = 100\bar{\rho}$) in fig.8.4 for index $n = 0.0$. We can see that at this epoch mass and overdense regions have different levels of clustering over the entire range of scales of interest.

In fig.8.5, we have plotted S_2 as a function of r for both mass and overdense regions for various values of the index n in redshift space. We applied cutoffs of $\rho_{cut} = 100\bar{\rho}$, $1000\bar{\rho}$ as before for the distributions of overdense regions. This figure is qualitatively similar to fig.8.3. This means that as far as the second moment is concerned, redshift space distortions play a less serious role compared to bias in modifying the nature of clustering.

In fig.8.6, we have plotted the second moment of overdense regions as a function of the second moment of the matter distribution in real and redshift space for index $n = -1.25$. A dashed line represents a distribution with $\rho_{cut} = 100\bar{\rho}$ and a dot-dashed line one with $\rho_{cut} = 1000\bar{\rho}$ in both panels of the figure. The upper panel is for real space, while the lower panel is for redshift space. The figure clearly shows that there is a scale dependence in bias that cannot be accounted for by a constant factor in the non-linear regime. Further, this dependence is different in real and redshift space.

Next, we study the variation of S_3 for mass and overdense regions with scale in real as well as redshift space for the same models as above. Fig.8.7 shows the dependence of S_3 on scale, bias and the index n of the initial power spectrum in real space. We applied the same density cutoffs on the matter distribution as before to obtain the distributions of overdense regions. A feature common with the second moment is that models with more negative indices are more biased for the same value of ρ_{cut}. Within the same model, S_3 is increasingly small for larger values of ρ_{cut} used to generate the distributions of overdense regions, i.e. larger the bias, smaller is the value of S_3.

In fig.8.8, we show the dependence of S_3 on scale, the initial power spectrum and bias in redshift space. We find that in redshift space S_3 for overdense regions with $\rho_{cut} \geq 100\bar{\rho}$

does not depend on the initial power spectrum for $S_2 \geq 1$. Redshift space effects also make S_3 for the distribution of mass as well as the distributions of overdense regions scale-independent at scales at which $S_2 \geq 1$. The *finger-of-god* effect (Davis and Peebles, 1983; Hamilton, 1997; Peacock et al., 2001) appears to override all other influences at small non-linear scales in redshift space.

We also simulated the ΛCDM model with $\Omega_b = 0.05$, $\Omega_{dm} = 0.25$ and $\Omega_{nr} = 0.3$ and we have shown some results for the epoch $a = 1$ in fig.8.9. The upper left panel of fig.8.9 shows S_2 as a function of r for mass and overdense regions in real space, while the upper right panel shows the same in redshift space. The lower set of panels shows S_3 as the function of S_2 in real (lower left panel) and redshift (lower right panel) space for mass and overdense regions. These plots show the same features as the earlier cases with power law initial spectra, particularly the striking nature of S_3 in redshift space. Our conclusions therefore appear to be true across different initial conditions and different cosmological models. Also the features visible in the lower right panel of fig.8.9 are consistent with observations of the third moment of galaxy clustering in redshift space (Croton et al., 2004). We expect that halo-based models should be able to explain the behaviour of S_3 that we are seeing in redshift space using N-Body simulations. It should also be possible to use our analysis as a discriminant between different models for bias. This is work in progress.

8.3 Discussion

In this chapter we have studied the clustering properties of mass and overdense regions in real space as well as redshift space in simulations of power law or scale invariant models. The distribution of mass in our N-Body simulations compares well with expectations from second order perturbation theory (fig.8.1). The second moment of the distribution of particles in overdense regions shows expected behaviour, i.e. overdense regions show a greater tendency to cluster as compared to the matter distribution given by all particles in a simulation.

The following features emerge clearly from the analysis. We summarise our results in real space first. Bias affects clustering in a significant manner. Changes in the second as well as the third moment of distributions due to the influence of bias depend significantly on the index of the power spectrum. Models with more negative indices tend to have a larger bias. The effect of bias for a given initial power spectrum is also monotonic, i.e. S_2 and S_3 for a distribution vary monotonically with the amount of bias.

Redshift space distortions play a more dominant role than bias and significantly modify the nature of clustering. The skewness S_3 for distributions with $\rho_{cut} \geq 100\bar{\rho}$ is independent of the model at non-linear scales where $S_2 \geq 1$. We find that redshift space effects make S_3 scale-independent for all models and for all distributions (matter and overdense regions) at scales where $S_2 \geq 1$. All of this is probably due to the role played by the fingers-of-god effect (Hamilton, 1997). Our results are consistent with simulation studies which have shown that the redshift space halo correlation function is insensitive to the details of the chosen halo density profile in the halo model (Kang et al., 2002). This analysis suggests that large peculiar motions within virialised haloes at small scales, that add a Gaussian spread to the redshift space radial coordinate (White, 2000b), efficiently remove signatures of initial conditions.

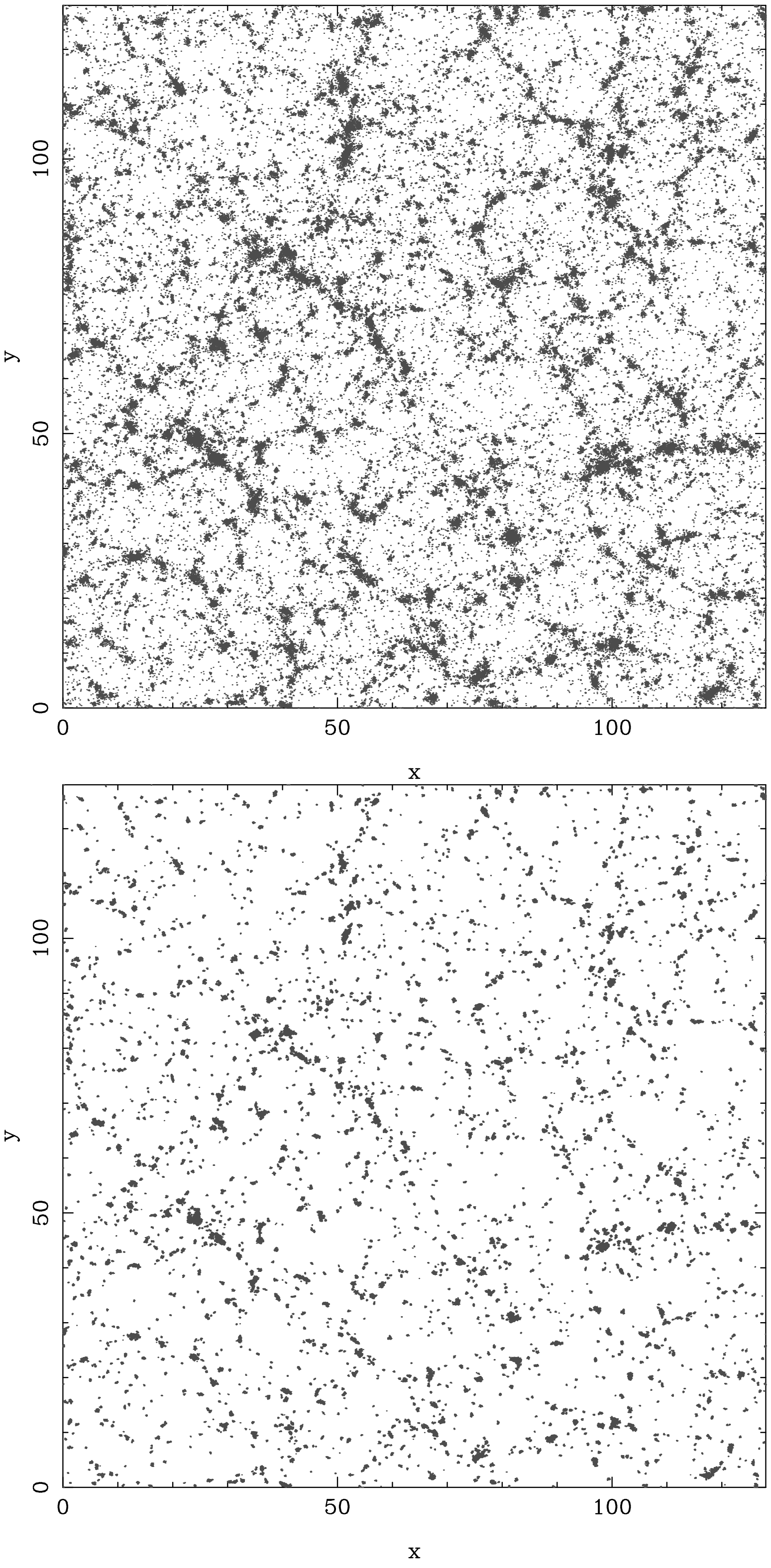

Figure 8.2: This figure shows two slices from a TreePM simulation of a power law model with index $n = 0.0$. The left panel shows all particles in the simulation in the given slice, whereas the right panel shows only particles within the slice which have Lagrangian densities at least 100 times higher than the average density in the simulation.

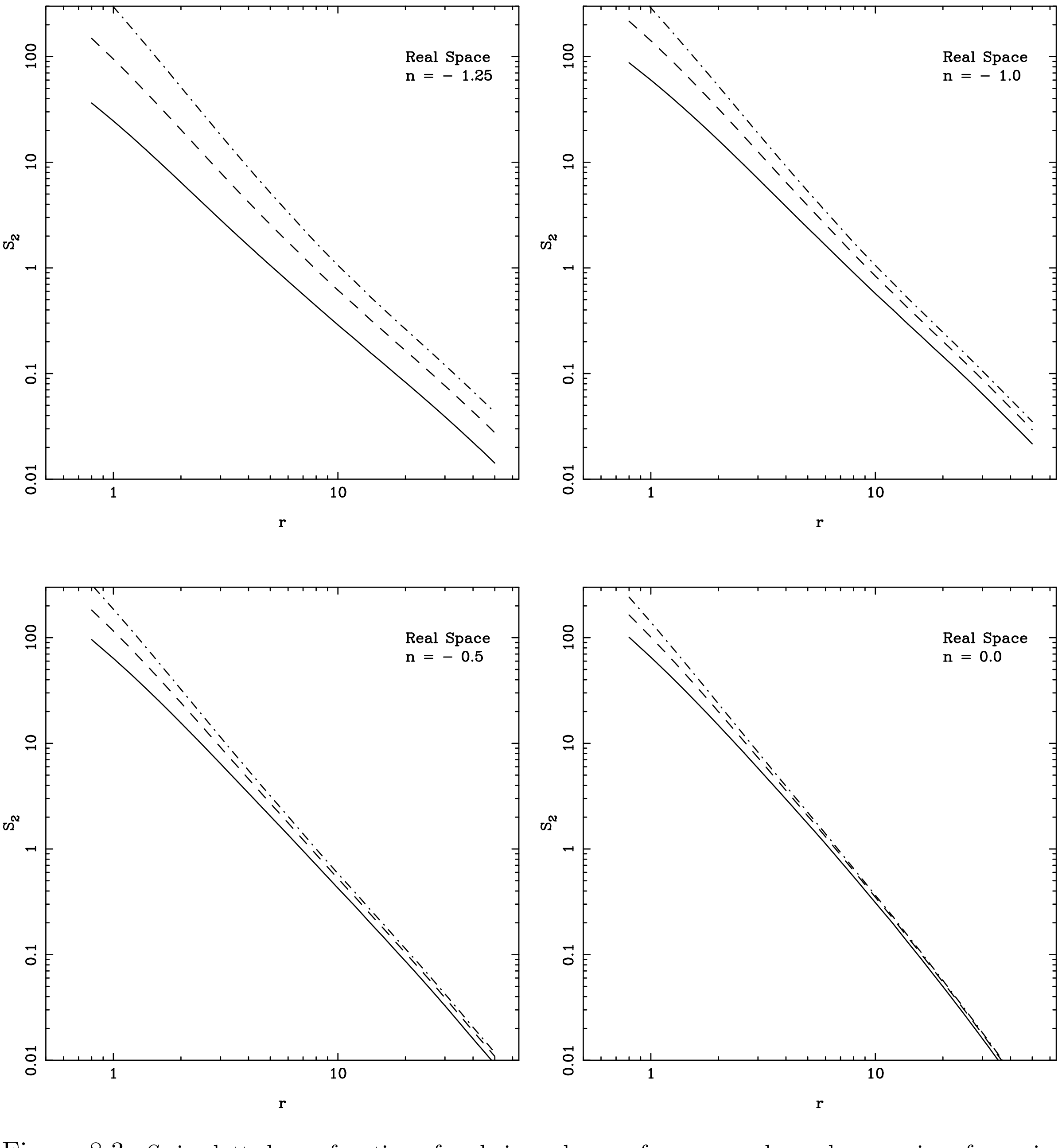

Figure 8.3: S_2 is plotted as a function of scale in real space for mass and overdense regions for various power law models. A solid line represents the mass distribution, a dashed line a distribution with $\rho_{cut} = 100\bar{\rho}$ and a dot-dashed line one with $\rho_{cut} = 1000\bar{\rho}$ in all the panels. We can see that bias is decreasing monotonically as we increase the value of the index n for the same value of the density cutoff used to generate a specific distribution of overdense regions.

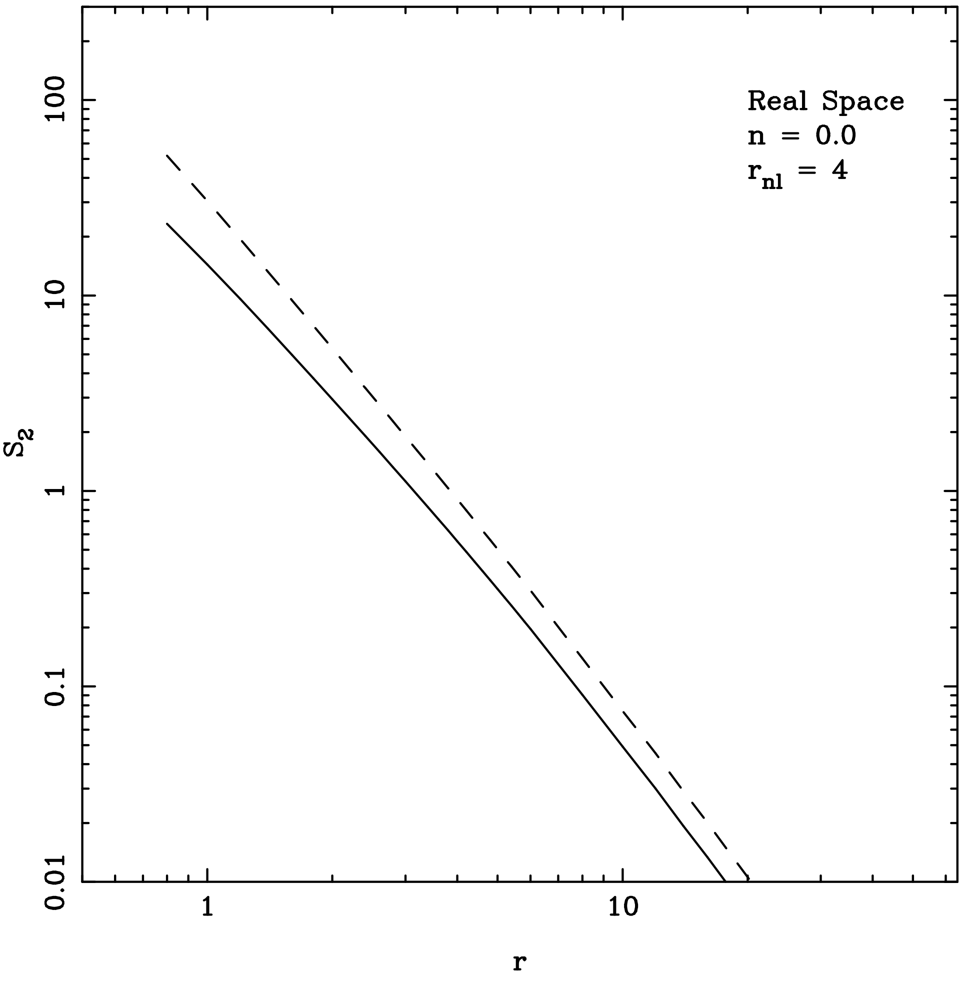

Figure 8.4: S_2 is plotted as a function of scale in real space for mass and overdense regions ($\rho_{cut} = 100\bar{\rho}$) for index $n = 0.0$ at an earlier epoch ($r_{nl} \sim 4$) with respect to that used for fig.8.3. At this epoch, much of the mass is still in the underlying matter distribution and overdense regions are evolving independent of the matter distribution.

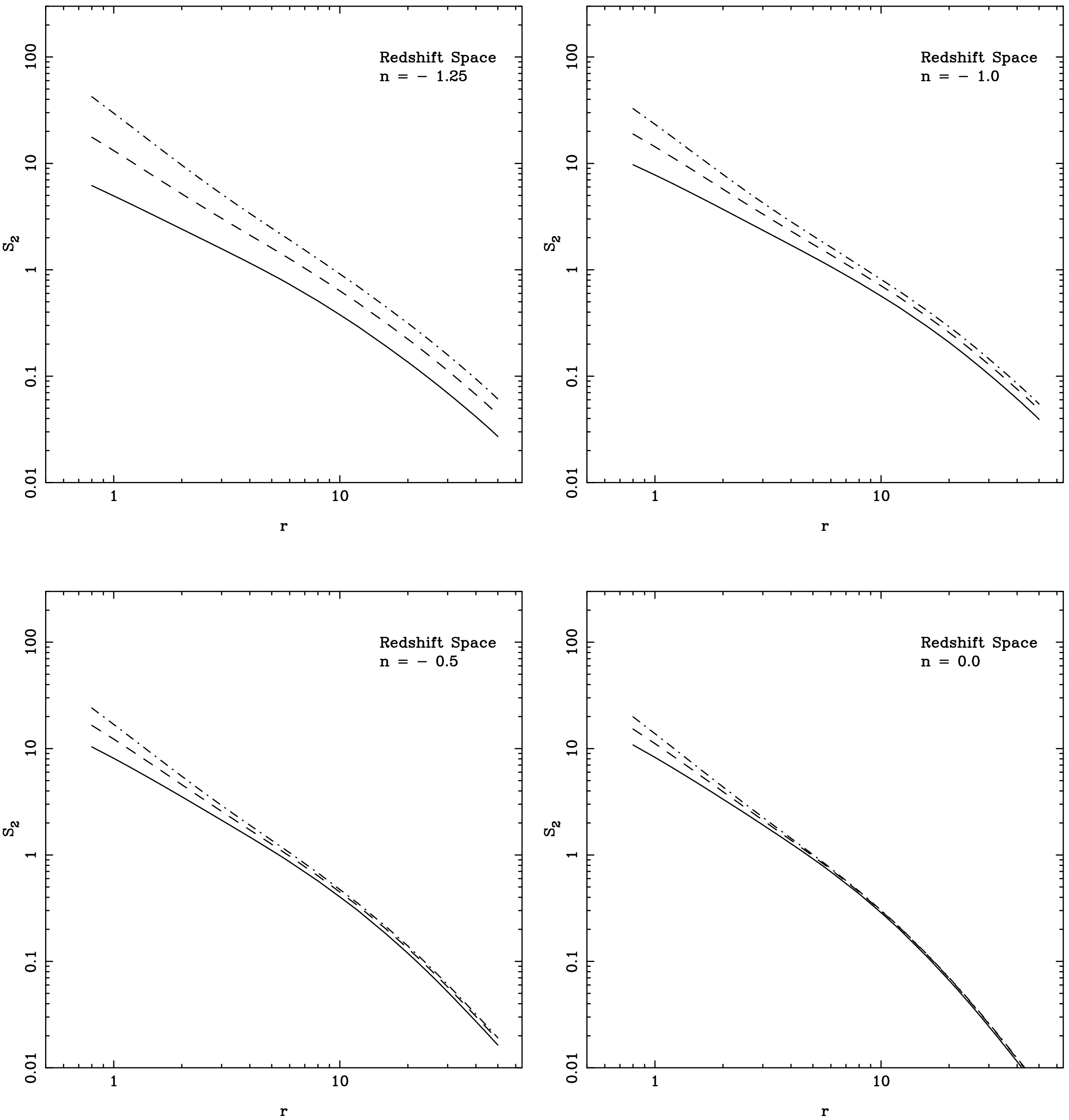

Figure 8.5: S_2 is plotted as a function of scale in redshift space for mass and overdense regions for various power law models. A solid line represents the mass distribution, a dashed line a distribution with $\rho_{cut} = 100\bar{\rho}$ and a dot-dashed line one with $\rho_{cut} = 1000\bar{\rho}$ in all the panels. We can see that bias is decreasing monotonically as we increase the value of index n for the same value of the density cutoff used to generate a specific distribution of overdense regions.

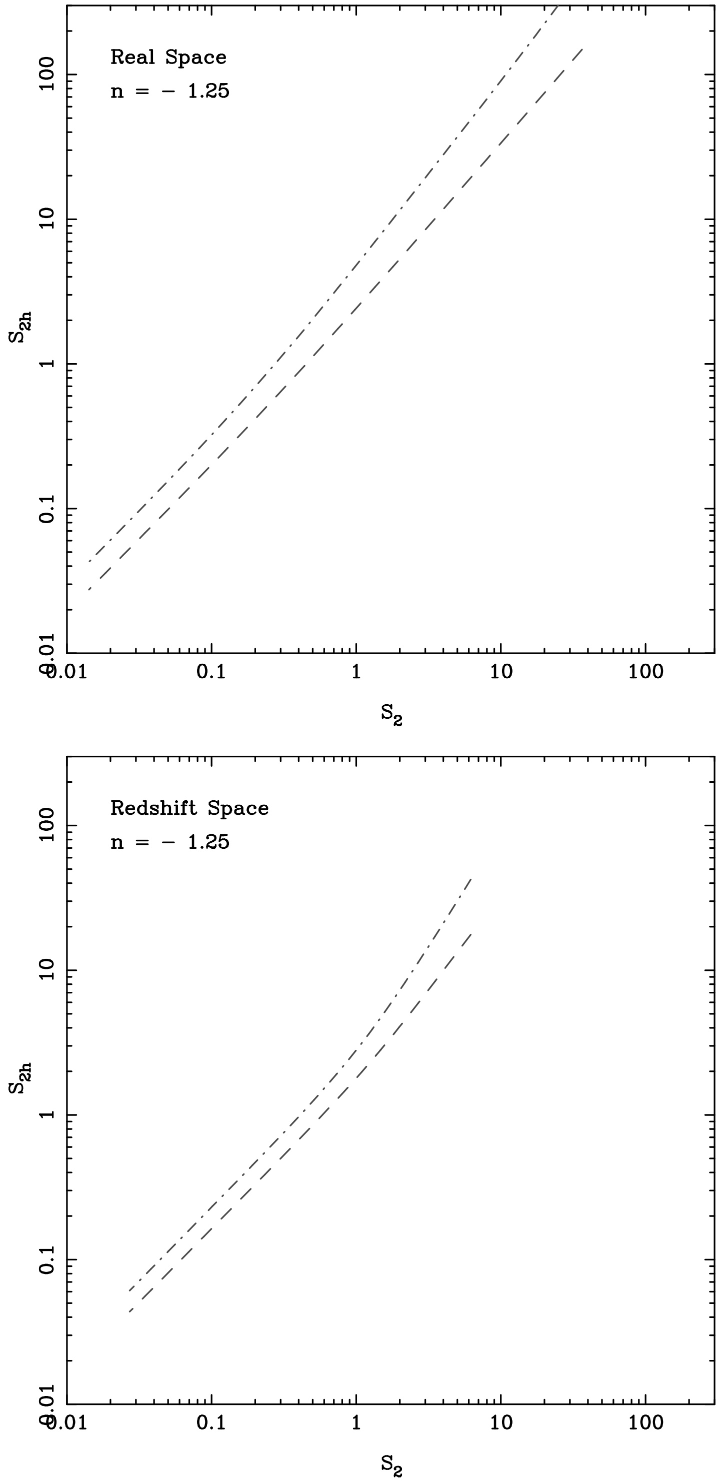

Figure 8.6: The second moment of overdense regions is plotted as a function of the second moment of the matter distribution in real and redshift space for index $n = -1.25$. The upper panel is for real space, while the lower panel is for redshift space. A dashed line represents a distribution with $\rho_{cut} = 100\bar{\rho}$ and a dot-dashed line one with $\rho_{cut} = 1000\bar{\rho}$ in both panels.

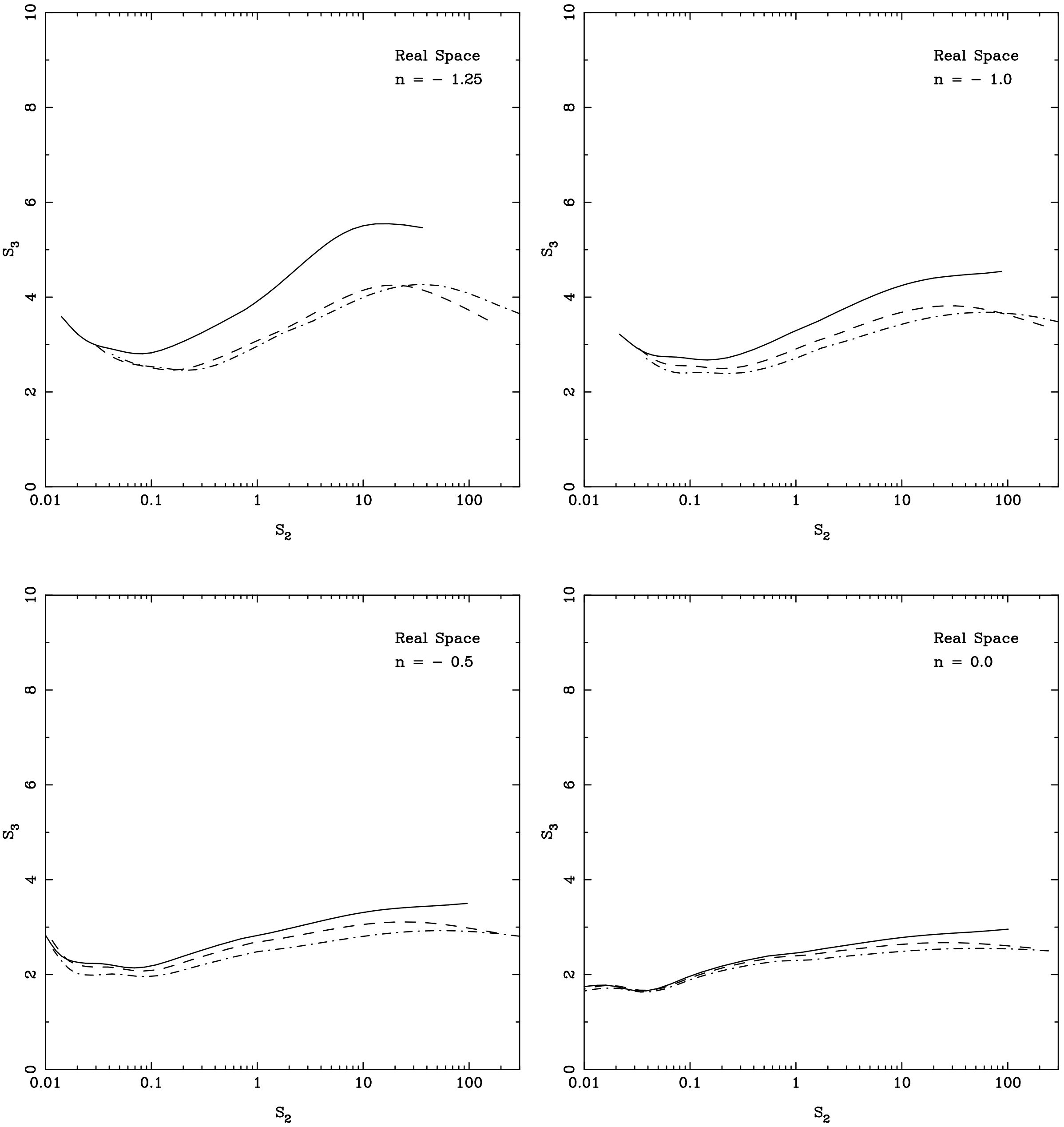

Figure 8.7: S_3 is plotted as a function of scale in real space for mass and overdense regions for various power law models. A solid line represents the mass distribution, a dashed line a biased distribution with $\rho_{cut} = 100\bar{\rho}$ and a dot-dashed line one with $\rho_{cut} = 1000\bar{\rho}$ in all the panels. Within the same model, S_3 is decreasing monotonically with increase in density threshold.

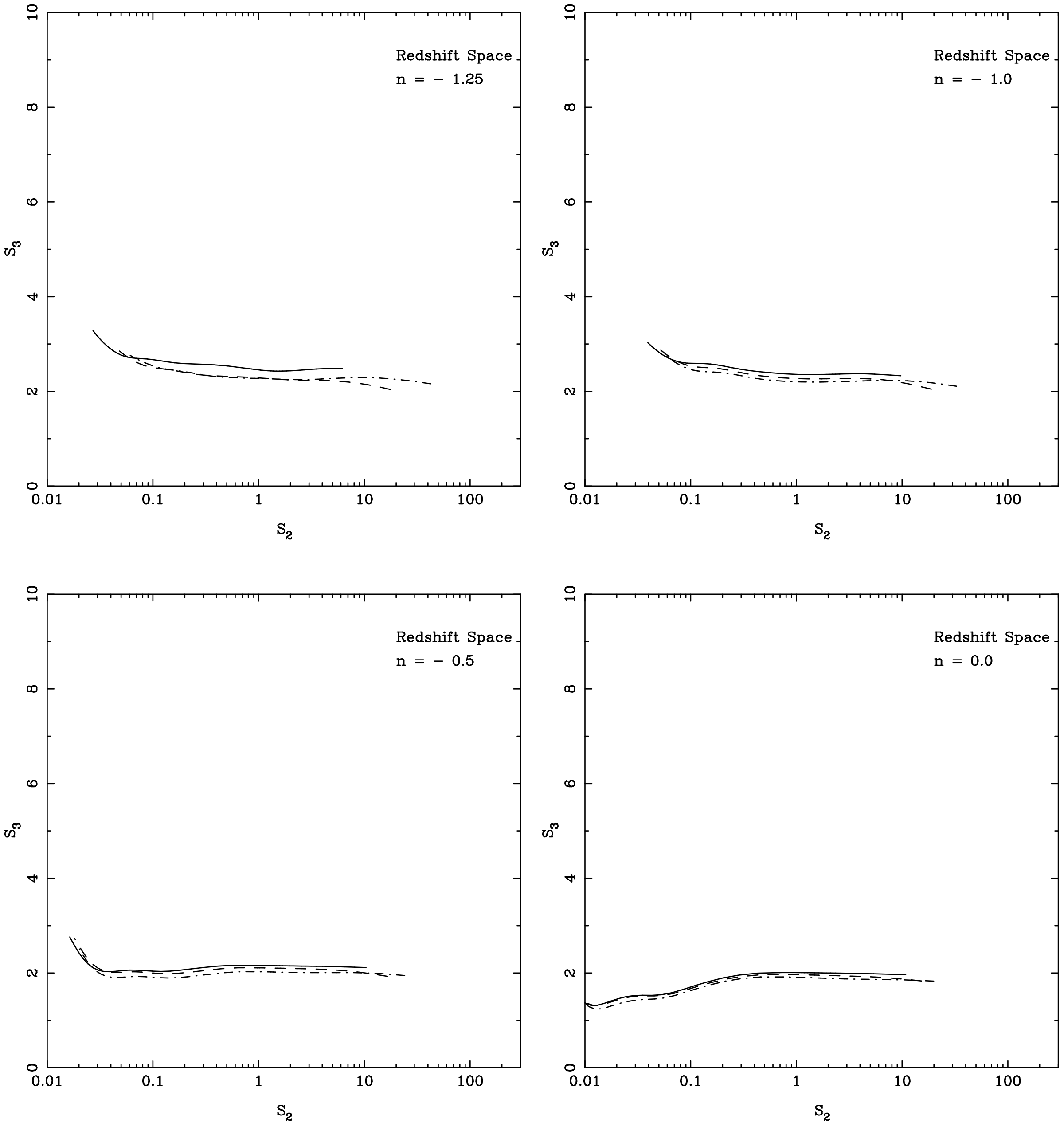

Figure 8.8: S_3 is plotted as a function of scale in redshift space for mass and overdense regions for various power law models. A solid line represents the mass distribution, a dashed line a distribution with $\rho_{cut} = 100\bar{\rho}$ and a dot-dashed line one with $\rho_{cut} = 1000\bar{\rho}$ in all the panels. Redshift space distortions make S_3 almost independent of bias and the power law index for scales at which $S_2 \geq 1$.

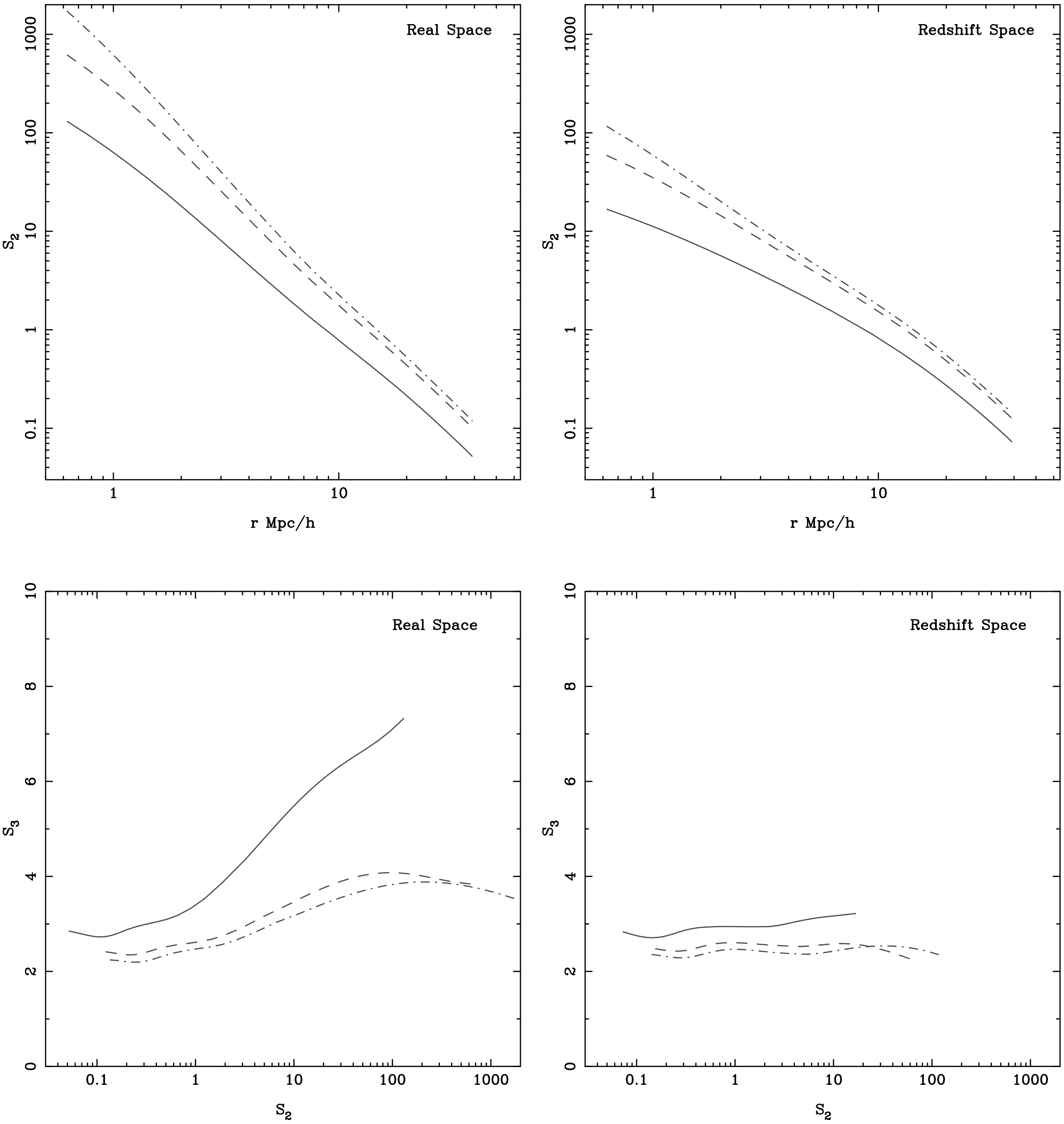

Figure 8.9: The upper left panel of this figure shows S_2 as a function of r for mass and overdense regions in real space, while the upper right panel shows the same in redshift space for a simulation of the ΛCDM model with $\Omega_b = 0.05$, $\Omega_{dm} = 0.25$ and $\Omega_{nr} = 0.3$ at epoch $a = 1$. The lower set of panels shows S_3 as the function of S_2 in real (lower left panel) and redshift (lower right panel) space for mass and overdense regions. A solid line represents the mass distribution, a dashed line a distribution with $\rho_{cut} = 100\bar{\rho}$ and a dot-dashed line one with $\rho_{cut} = 1000\bar{\rho}$ in all the panels.

Bibliography

Appel A.W. 1985, SIAM, J. Sci. Stat. Comp. 6, 85

Aarseth S.J. 1963, MNRAS 126, 223

Aarseth S.J., Gott J.R. and Turner E.L. 1979, ApJ 228, 664

Allen B. and Shellard E.P.S. 1990, Phys. Rev. Lett. 64, 119

Bagla J.S. and Padmanabhan T. 1994, MNRAS 266, 227

Bagla J.S. 1996, PhD Thesis, *Gravitational Clustering in an Expanding Universe*, Inter-University Centre for Astronomy and Astrophysics

Bagla J.S. and Padmanabhan T. 1997a, Pramana – Journal of Physics 49, 161

Bagla J.S. and Padmanabhan T. 1997b, MNRAS 286, 1023

Bagla J.S. and Padmanabhan T. 1997c, MNRAS 289, 671

Bagla J.S. 1998a, MNRAS 297, 251

Bagla J.S. 1998b, MNRAS 299, 417

Bagla J.S., Engineer S. and Padmanabhan T. 1998, ApJ 495, 25

Bagla J.S. 2002, Journal of Astrophysics and Astronomy 23, 185

Bagla J.S. 2003, *Numerical Simulations in Astronomy*, ed. K. Tomisaka and T. Hanawa, p.32

Bagla J.S. and Ray S. 2003, New Astronomy 8, 665

Bagla J.S. 2004a, *Cluster Computing in Astrophysics*

Bagla J.S. 2004b, *Cosmological N-Body Simulations : Techniques, Scope and Status*, to appear in Current Science (special issue on Cosmology, guest editor T. Padmanabhan)

Bagla J.S., Prasad J. and Ray S. 2004, astro-ph/0408429

Bagla J.S. and Ray S. 2004, astro-ph/0410373, accepted for publication in MNRAS

Bagla J.S., Ray S. and Padmanabhan T. 2005, Work in Progress

Bardeen J.M., Bond J.R., Kaiser N. and Szalay A.S. 1986, ApJ 304, 15

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

Barnes J. and Hut P. 1989, ApJS 70, 389

Barnes J.E. 1990, J. Comp. Phys. 87, 161

Beacom J.F., Dominik K.G., Melott A.L., Perkins S.P. and Shandarin S.F. 1991, ApJ 372, 351

Bennett D.P. and Bouchet F.R. 1990, Phys. Rev. D 41, 2408

Bennett C.L. et al. 2003, ApJS 148, 1

Bernardeau F. 1994, A&A 291, 697

Bernardeau F., Colombi S., Gaztanaga E. and Scoccimarro R. 2002, Physics Reports 367, 1

Bertschinger E. 1998, ARA&A 36, 599

Bertschinger E. 2001, ApJS 137, 1

Bharadwaj S., Mittal A.K. and Seshadri T.R. 1999, A&A 351, 405

Binney J. and Knebe A. 2002, MNRAS 333, 378

Bode P., Ostriker J. and Xu G. 2000, ApJS 128, 561

Bode P. and Ostriker J.P. 2003, ApJS 145, 1

Bond J.R., Efstathiou G. and Silk J. 1980, Phys. Rev. Lett. 45, 84

Bond J.R. and Szalay A. 1983, ApJ 274, 443

Bond J.R., Cole S., Efstathiou G. and Kaiser N. 1991, ApJ 379, 440

Bouchet F.R., Adam J.C. and Pellat R. 1985, A&A 144, 413

Bouchet F.R. and Kandrup H.E. 1985, ApJ 299, 1

Bouchet F.R. and Hernquist L. 1988, ApJS 68, 521

Brainerd T.G., Scherrer R.J. and Villumsen J.V. 1993, ApJ 418, 570

Brainerd T.G. and Villumsen J.V. 1994, ApJ 431, 477

Buchaltar A. and Kamionkowski M. 1999, ApJ 521, 1

Casas-Miranda R., Mo H.J. and Boerner G. 2002, astro-ph/0203280

Cen R.Y., Jameson A., Liu F. and Ostriker J.P 1990, ApJL 362, L41

Chitre S.M. et al. 2001, Current Science 80, 10

Cole S. and Lacey C. 1996, MNRAS 281, 716

Cole S. 1997, MNRAS 286, 38

Colless M. 1998, astro-ph/9804079, To appear in Phil. Trans. R. Soc. Lond. A

Colombi S., Bouchet F.R. and Schaeffer R. 1994, A&A 281, 301

Colombi S., Bouchet F.R. and Hernquist L. 1996, ApJ 465, 14

Cooley J.W. and Tukey J.W. 1965, Math. Comp. 19, 297

Cooray A. and Sheth R. 2002, Physics Reports 372, 1

Couchman H.M.P. 1991, ApJL 368, L23

Couchman H.M.P. and Peebles P.J.E. 1998, ApJ 497, 499

Cowsik R. and McClelland J. 1972, Phys. Rev. Lett. 29, 669

Croton D.J. et al. 2004, MNRAS 352, 1232

Davis M. and Peebles P.J.E. 1977, ApJS 34, 425

Davis M. and Peebles P.J.E. 1983, ApJ 267, 465

Davis M., Efstathiou G., Frenk C. and White S.D.M. 1985, ApJ 292, 371

Dehnen W. 2001, MNRAS 324, 273

Dekel A. 1982, ApJL 261, L13

Dekel A. and Lahav O. 1999, ApJ 520, 24

de Vaucouleurs G. 1971, PASP 83, 113

Dicke R.H., Peebles P.J.E., Roll P.G. and Wilkinson D.T. 1965, ApJ 142, 414

Doroshkevich A.G. and Zeldovich Ya.B. 1964, SvA 7, 615

Doroshkevich A.G., Zeldovich Ya.B. and Novikov I.D. 1967, SvA 11, 233

Doroshkevich A.G. 1970, Astrophysica 6:320, 30

Doroshkevich A.G. et al. 1980, MNRAS 192, 321

Dubinski J. 1996, New Astronomy 1, 133

Dubinski J. 2004, New Astronomy 9, 111

Engineer S., Srinivasan K. and Padmanabhan T. 1999, ApJ 512, 1

Efstathiou G. 1979, MNRAS 189, 203

Efstathiou G. and Eastwood J.W. 1981, MNRAS 194, 503

Efstathiou G., Davis M., Frenk C.S. and White S.D.M. 1985, ApJS 57, 241

Eisenstein D.J. and Hut P. 1998, ApJ 498, 137

Ewald P.P. 1921, Ann. Physik 64, 253

Ferrell R. and Bertschinger E. 1994, Int. J. Mod. Phys. C 5, 933

Filmore J.A. and Goldreich P. 1984, ApJ 281, 1

Freedman W.L. et al. 2001, ApJ 553, 47

Friedmann A. 1922a, Z.Phys. 10, 377

Friedmann A. 1922b, Z.Phys. 21, 326

Fry J.N., Melott A.L. and Shandarin S.F. 1993, ApJ 412, 504

Fry J.N. and Gaztanaga E. 1993, ApJ 413, 447

Fry J.N., Melott A.L. and Shandarin S.F. 1995, MNRAS 274, 745

Fukushige T. and Makino J. 1997, ApJ 477, L9

Fukushige T., Kawai A. and Makino J. 2004, ApJ 606, 625

Gelb J. 1992, PhD Thesis, MIT

Gelb J.M. and Bertschinger E. 1994, ApJ 436, 467

Gelb J.M. and Bertschinger E. 1994a, ApJ 436, 491

Ghigna S., Moore B., Governato F., Lake G., Quinn T. and Stadel J. 2000, ApJ 544, 616

Goodman J., Heggie D.C. and Hut P. 1993, ApJ 415, 715 33

Gott J.R., Turner E.L. and Aarseth S.J. 1979, ApJ 234, 13

Groth E.J. and Peebles P.J.E. 1976, A&A 53, 131

Gurbatov S.N., Saichev A.I. and Shandarin S.F. 1989, MNRAS 236, 385

Guth A.H. 1981, Phys. Rev. D 23, 347

Guth A.H. and Pi S.Y. 1985, Phys. Rev. D 32, 1899

Hamilton A.J.S., Kumar P., Edward L. and Matthews A. 1991, ApJ 374, L1

Hamilton A.J.S. 1997, astro-ph/9708102

Harrison E.R. 1970, Phys. Rev. D 1, 2726

Heath D.J. 1977, MNRAS 179, 351

Hennawi J.F. and Ostriker J.P. 2002, ApJ 572, 41

Hernquist L. 1987, ApJS 64, 715

Hernquist L. and Katz N. 1989, ApJS 70, 419

Hernquist L., Bouchet F.R. and Suto Y. 1991, ApJS 75, 231

Hernquist L., Hut P. and Makino J. 1993, ApJL 402, L85

Hernquist L. 1993, ApJ 404, 717

Hernquist L., Katz N., Weinberg D. and Jordi M.-E. 1996, ApJL 457, L51

Hockney R.W. and Eastwood J.W. 1988, *Computer Simulation using Particles*, McGraw Hill

Hoffman Y. and Ribak E. 1991, ApJL 380, 5

Hoyle F. 1953, ApJ 118, 513

Huang S., Dubinski J. and Carlberg R.G. 1993, ApJ 404, 73

Hubble E. 1929, Proceedings of the National Academy of Sciences 15, 3

Hui L. and Bertschinger E. 1996, ApJ 471, 1

Hut P., Makino J. and McMillan S. 1995, ApJL 443, 93

Jain B., Mo H.J. and White S.D.M. 1995, MNRAS 276, L25

Jain B. 1997, MNRAS 287, 687

Jeans J. 1902, Phil. Trans. 199A, 49

Juszkiewicz R., Bouchet F.R. and Colombi S. 1993, ApJ 412, L9

Kaiser N. 1987, MNRAS 227, 1

Kanekar N. 2000, ApJ 531, 17

Kang X. et al. 2002, astro-ph/0201124

Katz N. and Gunn J.E. 1991, ApJ 377, 365

Klypin A.A. and Shandarin S.F. 1983, MNRAS 204, 891

Klypin A. 2000, astro-ph/0005502

Klypin A., Kravtsov A.V., Bullock J.S. and Primack J.R. 2001, ApJ 554, 903

Knebe A., Green A. and Binney J. 2001, MNRAS 325, 845

Kochanek C.S. and White M. 2000, ApJ 543, 514

Kravtsov A.V., Klypin A.A., Bullock J.S. and Primack J.R. 1998, ApJ 502, 48

Lahav O. et al. 1991, MNRAS 251, 128

Landau L.D. and Lifshitz E.M. 1975, *The Classical Theory of Fields*, Pergamon Press

Larson R.B. 1969, MNRAS 145, 405

Layzer D. 1963, ApJ 138, 174

Lee B.W. and Weinberg S. 1977, Phys. Rev. Lett. 39, 165

Lemaitre A.G. 1931, MNRAS 91, 483L

Lemaitre A.G. 1931, MNRAS 91, 490

Lightman A.P. and Schechter P.L. 1990, ApJS 74, 831

Little B., Weinberg D.H. and Park C. 1991, MNRAS 253, 295

Ma C. 1998, ApJL 508, L5

Matarrese, Lucchin, Moscardini and Saez 1992, MNRAS 259, 437

Matarrese S., Verde L. and Heavens A.F. 1997, MNRAS 290, 651

Matsubara T. 1994, ApJ 424, 30

McClelland J. and Silk J. 1977, ApJ 217, 331

McMillan S.L.W. and Aarseth S.J. 1993, ApJ 414, 200

Meiksin A. and White M. 2001, MNRAS 324, 141

Melott A.L., Shandarin S.F., Splinter R.J. and Suto Y. 1997, ApJL 479, 79

Merz H., Pen U.L. and Trac H. 2004, astro-ph/0402443

Miller R.H. 1983, ApJ 270, 390

Mo H.J. and White S.D.M. 1996, MNRAS 282, 347

Mo H.J., Jing Y.P. and White S.D.M. 1997, MNRAS 284, 189

Monaghan J.J. 1992, Annu. Rev. Astron. Astrophys. 30, 543

Moore B., Governato F., Quinn T., Stadel J. and Lake G. 1998, ApJ 499, L5

Munshi D., Chiang L., Coles P. and Melott A.L. 1998, MNRAS 293, L68

Munshi D. and Coles P. 1998, MNRAS 293, L68

Narlikar J.V. 1993, *Introduction to Cosmology*, Cambridge University Press

Navarro J.F., Frenk C.S. and White S.D.M. 1996, ApJ 462, 563

Navarro J.F., Frenk C.S. and White S.D.M. 1997, ApJ 490, 493

Neyman J. and Scott E.L. 1952, ApJ 116, 144

Nityananda R. and Padmanabhan T. 1994, MNRAS 271, 976

Norberg P. et al. 2001, MNRAS 328, 64

Norberg P. et al. 2002, MNRAS 332, 827

Norman M.L. 2000, astro-ph/0005109

Padmanabhan T. 1993, *Structure Formation in the Universe*, Cambridge University Press

Padmanabhan T. 1996, MNRAS 278, L29

Padmanabhan T., Cen R., Ostriker J.P. and Summers F.J. 1996, ApJ 466, 604

Padmanabhan T. 2000, Aspects of Gravitational Clustering, *Large Scale Structure Formation*, ed. by R. Mansouri and R. Brandenberger, Astrophysics and Space Science Library, volume 247, Kluwer Academic, astro-ph/9911374

Padmanabhan T. and Kanekar N. 2000, PhRvD 61, 023515

Padmanabhan T. 2002a, Theoretical Astrophysics, Volume III : Galaxies and Cosmology, 638 pp, Cambridge University Press

Padmanabhan T. 2002b, Statistical Mechanics of Gravitating Systems in Static and Cosmological Backgrounds in *Dynamics and Thermodynamics of Systems with Long Range Interactions*, Eds : T. Dauxois, S. Ruffo, E. Arimondo and M. Wilkens; Lecture Notes in Physics, Springer (2002), astro-ph/0206131

Peacock J.A. and Heavens A.F. 1985, MNRAS 217, 805

Peacock J.A. and Dodds S.J. 1994, MNRAS 267, 1020

Peacock J.A. and Dodds S.J. 1996, MNRAS 280, L19

Peacock J.A. 1998, *Cosmological Physics*, Cambridge University Press

Peacock J.A. et al. 2001, Nature 410, 169

Pearce F.R. and Couchman H.M.P. 1997, New Astronomy 2, 411

Peebles P.J.E. 1970, Astron. J. 75, 13

Peebles P.J.E. 1971, *Physical Cosmology*, Princeton University Press

Peebles P.J.E. 1974, ApJ 189, L51

Peebles P.J.E. 1974a, A&A 32, 391

Peebles P.J.E. 1980, *The Large-Scale Structure of the Universe*, Princeton University Press

Peebles P.J.E. 1985, ApJ 297, 350

Peebles P.J.E. 1993, *An Introduction to Physical Cosmology*, Princeton University Press

Perlmutter S. et al. 1999, ApJ 517, 565

Power C., Navarro J.F., Jenkins A., Frenk C.S., White S.D.M., Springel V., Stadel J. and Quinn T. 2003, MNRAS 338, 14

Press W.H. and Schechter P.L. 1974, ApJ 187, 425

Quinn T., Katz N., Staedel J. and Lake G. 1997, astro-ph/9710043

Ray S. and Bagla J.S. 2004, astro-ph/0405220

Ray S. 2004, astro-ph/0406009, accepted for publication in JAA

Ray S., Bagla J.S. and Padmanabhan T. 2004, astro-ph/0410041

Rees M.J. and Ostriker J.P. 1977, MNRAS 179, 541

Robertson H.P. 1935, Proc. Nat. Acd. Sci. 15, 822

Rybicki G.B. 1986, *The Use of Supercomputers in Stellar Dynamics*, ed. P. Hut and S. McMillan, Springer

Sahni V. and Coles P. 1995, Physics Reports 262, 1

Salmon J.K. 1991, PhD Thesis, *Parallel Hierarchical N-Body Methods*, California Institute of Technology

Salmon J.K. and Warren M.S. 1994, J. Comp. Phys. 111, 136

Schneider P., Ehlers J. and Falco E.E. 1992, *Gravitational Lenses*, Springer-Verlag

Shectman S.A. et al. 1996, ApJ 470, 172

Silk J. 1977, ApJ 211, 638

Sheth R.K. and Jain B. 1997, MNRAS 285, 231

Sheth R.K. and Lemson G. 1999, MNRAS 304, 767

Sheth R.K. and Tormen G. 1999, MNRAS 308, 119

Sheth R.K., Mo H.J. and Tormen G. 2001, MNRAS 323, 1

Smith R.E. et al. 2003, MNRAS 341, 1311

Smoot G.F. et al. 1992, ApJ 396, L1

Snir M., Otto S., Huss-Lederman S., Walker D. and Dongarra J. 1999, *MPI–The Complete Reference. Volume 1, The MPI Core (second edition)*, MIT Press

Spergel D.A. et al. 2003, ApJS 148, 175

Springel V., Yoshida N. and White S.D.M. 2001, New Astronomy 6, 79

Steinmetz M. and Mueller E. 1993, A&A 268, 391

Strauss M.A., Yahil A., Davis M., Huchra J.P. and Fisher K.B. 1992, ApJ 397, 395

Strauss M.A. and Willick J.A. 1995, Phys. Rep. 261, 271

Subramanian K. 2000, ApJ 538, 517

Subramanian K., Cen R. and Ostriker J.P. 2000, ApJ 538, 528

Syer D. and White S.D.M. 1998, MNRAS 293, 337

Tegmark M. and Peebles P.J.E. 1998, ApJ 500, 79

Tegmark M. et al. 2003, astro-ph/0310723

Tormen G. and Bertschinger E. 1996, ApJ 472, 14

Tormen G., Bouchet F.R. and White S.D.M. 1997, MNRAS 286, 865

Toshikazu E. et al. 1993, PASJ 45, 269

Trimble V. 1987, Annu. Rev. Astron. Astrophys. 25, 425

Van Waerbeke L. et al 2000, A&A. 358, 30

Verde L. et al. 2002, MNRAS 335, 432

Vilenkin A. and Shellard E.P.S. 1994, *Cosmic Strings and Other Topological Defects*, Cambridge University Press

Villumsen J.V. 1989, ApJS 71, 407

Walker A.G. 1936, Proc. London Math. Soc. 42, 90

White S.D.M. 1976, MNRAS 177, 717

White S.D.M. and Rees M.J. 1978, MNRAS 183, 341

White M. 2000a, private communication

White M. 2000b, astro-ph/0005085

Wright E.L., Bennet C.L., Gorski K., Hinshaw G. and Smoot G.F. 1996, ApJ 464, L21

Xu G. 1995, ApJS 98, 355

Zeldovich Ya.B. and Podurets M.A. 1966, SvA 9, 742

Zeldovich Ya.B. and Novikov I.D. 1967, SvA 11, 526

Zeldovich Ya.B. 1970, A&A 5, 84

Zeldovich Ya.B. 1972, MNRAS 160, 1