Modelling Warm Dark Matter in Cosmological Simulations

SIGILL: UNIVERSITAT: LUDOVIC: MAXIM:

Jens Oliver Stücker

Modelling Warm Dark Matter in Cosmological Simulations

Master Thesis
at the
Ludwig–Maximilians–Universität (LMU) München

submitted by

Jens Stücker
born on April 22nd 1991 in Hamm

supervised by
PD Dr. Klaus Dolag

Munich, September 17th 2015

Evaluator: PD Dr. Klaus Dolag

Contents

Contents **v**

1 Introduction **3**
- 1.1 The ΛCDM Model . . . 4
- 1.2 The Formation of Structures . . . 6
- 1.3 Cosmological Simulations . . . 8

2 The Warmth of Dark Matter **11**
- 2.1 What is Warm Dark Matter? . . . 11
- 2.2 Thermal Relics . . . 13
- 2.3 Free Streaming . . . 15
- 2.4 The Small Scale Crisis of Cold Dark Matter . . . 16

3 Simulations of Warm Dark Matter **19**
- 3.1 Visualization . . . 19
- 3.2 Zoomed Initial Conditions . . . 19
- 3.3 Structure Formation . . . 21
- 3.4 Artificial Haloes . . . 23
- 3.5 Adaptive Gravitational Softening . . . 24

4 The Dark Matter Sheet **29**
- 4.1 Density Estimates . . . 30
- 4.2 The Geodeosic Deviation Equation . . . 33

5 Anisotropic Softening for Gadget 3 **37**
- 5.1 The Distortions of the Ellipsoids . . . 37
- 5.2 The Potential of an Ellipsoid . . . 40
  - 5.2.1 Ellipsoidal Shell . . . 40
  - 5.2.2 Ellipsoids with Kernel . . . 41
  - 5.2.3 Homogeneous Ellipsoid . . . 44
  - 5.2.4 Epanechnikov Ellipsoid . . . 47
- 5.3 Forces and Tidal Tensor . . . 49
- 5.4 Numerical Evaluation of the Potential . . . 51
  - 5.4.1 Approach 1: Interpolation of the external potential . . . 51
  - 5.4.2 Approach 2: Direct 3d Interpolation of the Coefficients . . . 57

5.4.3 Coordinate Transformations . . . 63
5.4.4 Force Split . . . 64
5.4.5 Time-Steps . . . 66
5.5 Summary and Usage . . . 67

6 Simulations without Artificial Haloes 71
6.1 The Anti-symmetrically Perturbed Plane Wave . . . 71
6.2 Glass Making and PM Force-Anisotropies . . . 72
6.3 Simulations in a full Cosmological Environment . . . 75
6.3.1 Density Fields . . . 76
6.3.2 Halo Mass Function . . . 80
6.3.3 The Distortion Ellipsoids . . . 82

7 Discussion and Conclusions 87

Bibliography 91

Acknowledgements 94

Abstract

The standard ΛCDM model of cosmology postulates that the formation of structures in the universe is driven by a largely unknown component of dark matter. It is one of the most important projects of modern physics to find out what dark matter is. Cosmological simulations are an important tool to predict the effects of different dark matter models, and to constrain properties of dark matter by the comparison with observations of our universe. We attempt to simulate different warm dark matter scenarios in cosmological high-resolution "zoom-in" simulations. However, N-Body simulations of warm dark matter suffer from the artificial fragmentation of filaments into small, spurious haloes. We decide to address this problem by considering new numerical approaches. As a first approach we test Adaptive Gravitational Softening, but find that it does not help out, as it does not follow the anisotropic distortions of the dark matter sheet. Therefore we develop the new numerical technique *Anisotropic Softening* which is based on the potential of ellipsoids that can deform and rotate along all three axes individually. The deformations of the ellipsoid are defined by the Geodesic Deviation Equation, a numerical technique that follows the distortions of an infinitesimal volume element around each particle (Vogelsberger et al., 2008). With Anisotropic Softening we manage to match mass- and force-resolution precisely also in situations of highly anisotropic collapse, and thereby avoid any artificial fragmentation while keeping the force resolution high. As a last step we present warm dark matter simulations in a full cosmological environment that do not suffer from any fragmentation.

Chapter 1

Introduction

In the currently favoured understanding of our universe visible matter makes up only a small fraction of its content. Very different observations, such as galaxy-rotation curves, gravitational lensing effects, and the temperature fluctuations in the Cosmic Microwave Background, indicate that there is a massive component in the universe which has no electromagnetic interaction with light. This invisible component is commonly called *dark matter*. Assuming a dark matter component and a dark energy component which accelerates the expansion of our universe, modern cosmological models are in remarkable quantitative agreement with observations of large scale structures. However, the question still remains: *What is dark matter?* Although there are a large number of well motivated suggestions from particle physics, decisive evidence for any of them is still missing. It is one of the most important projects of modern physics to find out what dark matter is. One approach to address this question is to try to observe dark matter directly by *direct detection experiments*.

The cosmological approach to address this question, is to model our universe theoretically, and compare our expectations with observations of our universe. With this approach properties of dark matter that can lead to observational differences can be investigated . One very fundamental property is *the warmth of dark matter*. A thermal velocity dispersion in the dark matter component leads to a suppression of structure formation below a certain length scale. Therefore a strongly reduced number density of dark matter clumps at low masses could be an indicator for the warmth of dark matter.

The original purpose of this thesis was to use numerical zoom-in simulations to investigate which observables could be used to further quantitatively investigate the warmth of dark matter. However, as a well known problem, simulations of warm dark matter suffer from the formation of small structures which can be clearly identified as numerical artefacts. As we did not want to do simulations that suffer from this uncertainty, we decided to address this problem by testing different numerical techniques. This leads us to the development of what we call *Anisotropic Softening*. We test this technique, and further present a first low-resolution simulation which does not suffer from the formation of artificial structures.

In the remaining part of this introduction we will shortly explain the current cosmological standard model and how numerical simulations of structure formation work. In Chapter 2

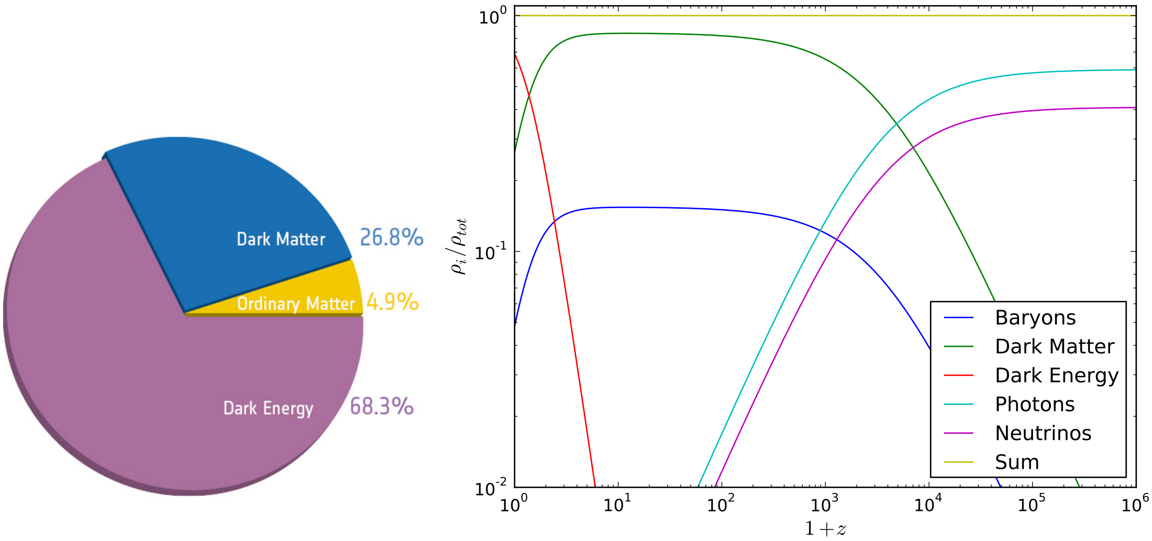


Figure 1.1: *Left: the density parameters* $\Omega_i = \rho_i/\rho_0$ *today. Figure from the website of the Planck Collaboration (Planck Collaboration et al., 2015) accessed on 2015-09-09. Right: the evolution of the density parameters* $\rho_i(a)/\rho_{tot}(a)$ *with redshift. Figure made with the output of the CLASS Code (Blas et al., 2011) for a* Λ*CDM universe.*

we explain in more detail what is meant by warm dark matter and what assumptions are typically made to address the warmth of dark matter in simulations. In Chapter 3 we present simulations of warm dark matter and discuss the problems that arise with artificial haloes. In Chapter 4 we explain the Dark Matter Sheet which we consider as an important notion to understand the behaviour of the dark matter fluid, and to distinguish between physical and numerical effects in simulations. In Chapter 5 we develop the numerical method *Anisotropic Softening* which is based on evaluating the potential of an ellipsoidal mass distribution for every particle in numerical simulations. We test this method in the 6th chapter where we also apply it to a warm dark matter simulation in a full cosmological context. In chapter 7 we discuss arising problems and make some final conclusions.

1.1 The ΛCDM Model

It has been known for a long time that galaxies in our universe, separated by a distance Δx, on average seem to be moving away from each other with a velocity

$$\Delta v = H_0 \Delta x \tag{1.1}$$

where H_0 is the Hubble constant (today) $H_0 = 100\ h\text{km/s/Mpc}$ (with $h \sim 0.7$). Assuming that there is no preferred reference frame in space, this observation can only be interpreted as the expansion of the universe itself. The ΛCDM model - the current standard model of cosmology - is the simplest model that can parametrize the expansion of the universe and most related observations quantitatively correctly. Therefore, it has to assume - in addition to the well known components of light, neutrinos, and baryonic matter - a cold dark matter (CDM) component and a cosmological constant Λ (which is equivalent to a dark energy component):

Assuming Albert Einstein's General Relativity (GR) to be valid, and that the universe is homogeneous and isotropic on large scales (> 100Mpc), the two Friedmann equations can be derived

$$H^2 = \left(\frac{\dot{a}}{a}\right)^2 = \frac{8\pi G}{3}\sum \rho_i - \frac{kc^2}{a^2} + \frac{\Lambda c^2}{3} \tag{1.2}$$

$$\frac{\ddot{a}}{a} = -\frac{4\pi G}{3}\sum \left(\rho_i + \frac{3p_i}{c^2}\right) + \frac{\Lambda c^2}{3} \tag{1.3}$$

where H is the Hubble parameter, a is the scale factor of the universe, ρ_i is the energy density of individual components, p_i is the pressure of those components, k is a constant describing the curvature of the universe, and Λ is the cosmological constant. The Friedmann equations describe how the expansion of the universe relates to its ingredients. Therefore by examining its expansion history, it can also be investigated what its components are.

The density parameters

$$\Omega_i = \frac{\rho_i}{\rho_{cr}} = \frac{8\pi G}{3H_0^2}\rho_i \tag{1.4}$$

$$\Omega_k = \frac{kc^2}{H_0^2 a^2} \tag{1.5}$$

$$\Omega_\Lambda = \frac{\Lambda c^2}{3H_0^2} \tag{1.6}$$

describe the importance of each component. Observationally it has been found that today the universe seems to be flat $\Omega_k \approx 0$, the density parameter associated with the cosmological constant $\Omega_\Lambda \sim 0.7$ is not zero, and that the matter density parameter is $\Omega_m \sim 0.3$, whereas the part of the matter that can interact with light ($\Omega_b \sim 0.05$) makes up only a small fraction of the total density (e.g. Planck Collaboration et al. (2015)). In the left panel of Figure 1.1 we show the pie plot of the density parameters today.

However, as the universe expands, densities change over time, and the importance of each component has been different in the past. For cold matter ($v \ll c$), the energy density decreases like the number density of the particles, as their energy is mostly given by their rest mass

$$\rho_m \propto a^{-3}\,. \tag{1.7}$$

The energy density of highly relativistic particles like photons (or neutrinos) decreases like

$$\rho_r \propto a^{-4}\,, \tag{1.8}$$

since not only their number density decreases, but also the energy associated with every photon decreases

$$E_\gamma = \frac{hc}{\lambda} \propto a^{-1}\,. \tag{1.9}$$

We explain this behaviour in more detail in section 2.1. This is commonly referred to as the red shifting of photons. The redshift z is defined for a photon which was emitted at a scale factor $a = a_{\mathrm{em}}$, and is observed today $a_{\mathrm{obs}} = 1$ with a relative shift in wavelength

$$z =: \frac{\Delta\lambda}{\lambda_{\mathrm{em}}} = \frac{\lambda_{\mathrm{obs}} - \lambda_{\mathrm{em}}}{\lambda_{\mathrm{em}}} = \frac{a_{\mathrm{obs}}}{a_{\mathrm{em}}} - 1 = \frac{1}{a} - 1\,. \tag{1.10}$$

Due to its simple relation to the scale factor, the redshift can be used interchangeable with the scale factor.

$$a = \frac{1}{1+z} \tag{1.11}$$

The last important component is dark energy. The density parameter of the cosmological constant can also be interpreted as the effect of another ingredient with an energy density

$$\rho_{\mathrm{de}} \approx \mathrm{const.} \tag{1.12}$$

which does not decrease with the scale factor. This could, for example, be the case if the vacuum itself contains energy (vacuum energy). However, as the case of a cosmological constant (which would be an extension parameter to GR), and dark energy (being another ingredient of the universe) cannot be discriminated observationally so far, their notions are often used interchangeably, when it comes to parametrizing the expansion of the universe.

In the right panel of Figure 1.1 we show how the density parameters $\rho_i(a)/\rho_{\mathrm{tot}}(a)$ change with the redshift. It can be seen that dark energy only plays a role in the universe since $z \sim 2$. Before this time, there had been a long period of matter domination up to $z \sim 3000$. However, since matter's energy density decreased like a^{-3}, but the energy density of relativistic species like photons and neutrinos decreased like a^{-4}, there was a time where the universe was dominated by those species (radiation domination).

1.2 The Formation of Structures

As the universe was younger ($z \gtrsim 1100$), densities and temperatures ($T \gtrsim 3000\,\mathrm{K}$) were much larger, and hydrogen atoms were still ionized. Interactions between light and baryonic matter were frequent enough to maintain thermal equilibrium locally. However, as the universe cooled down, hydrogen recombined[1], and the universe became transparent for photons. These photons have been moving freely since recombination, and can be observed today as the cosmic microwave background at a nearly uniform temperature of $T_0 \sim 2.7\,\mathrm{K}$ all over the sky. However, the temperature is not perfectly homogeneous, but there exist small anisotropies at the order of 10^{-5} (Figure 1.2). The origin of these fluctuations is still under debate, but the currently favoured theory is the theory of inflation that postulates a short inflationary phase in the early universe, which scaled up initial quantum fluctuations to the observed scales (and even much larger scales). It is believed that these

---

[1] The commonly used notion *recombination* is actually a bit misleading as protons and electrons combined for the first time to hydrogen.

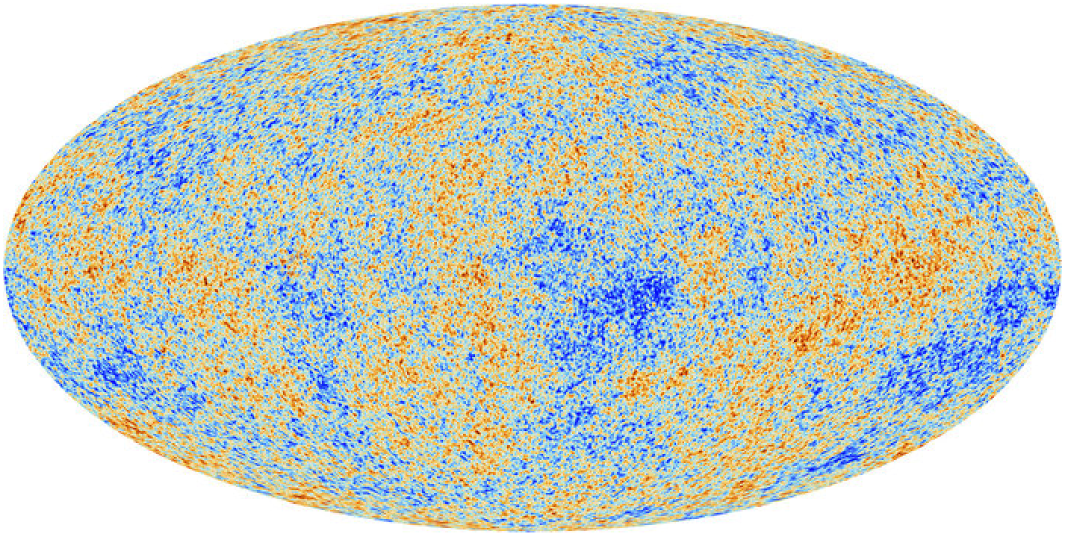


Figure 1.2: *The Cosmic Microwave Background Anisotropies as measured by the Planck Satellite (Planck Collaboration et al., 2015). The colours indicate anisotropies in the temperatures of the observed CMB light at a relative deviation from the mean temperature of* 10^{-5}*. Image from the Website of the Planck Collaboration accessed at 2015-09-09.*

small initial perturbations are the seeds for the growth of all known structures in the universe.

Dark matter most likely also has density perturbations from the same origin like the CMB anisotropies. At the decoupling of the CMB, they, of course, already evolved differently, than the perturbations in baryonic matter and photons. However, combining linear perturbation theory (LPT), the temperature anisotropies from the CMB, and other cosmological probes, the statistics of density fluctuations in the dark matter component can be figured out. They are commonly expressed in terms of a power spectrum

$$\langle \delta(\vec{k})\delta(\vec{k'}^{*})\rangle = (2\pi)^3 \delta_D(\vec{k}-\vec{k'})P(k) \tag{1.13}$$

where $P(k)$ is the power-spectrum, $\delta(\vec{k})$ the Fourier transformation of the density fluctuation field $\delta(\vec{x}) = (\rho(\vec{x}) - \rho_{\mathrm{mean}})/\rho_{\mathrm{mean}}$, and δ_D the 3d Dirac delta function. The statistics of density fluctuations can be completely characterized by the power spectrum and LPT, as long as density fluctuations are small $\delta(k) \ll 1$. This is typically the case up to a redshift of $z \sim 100$ (depending on the scales k, that are being considered - smaller scales leave the regime of LPT earlier). In the era of matter domination ($z \sim 10^4$ to $z \sim 1$), LPT typically predicts the density fluctuations to grow linearly with the scale factor $\delta \propto a$.

The thermal velocity dispersion of dark matter can have a big impact onto the power spectrum on small scales, as we will see in chapter 2.3. If the thermal velocity dispersion is small enough that it can be approximated by zero $v_{\mathrm{therm,\ CDM}} \approx 0$, one speaks of cold dark matter. If it is non-relativistic, but not negligible $0 \ll v_{\mathrm{therm,\ WDM}} \ll c$ one speaks of warm dark matter. If thermal velocities are ultra-relativistic $v_{\mathrm{therm,\ HDM}} \sim c$ one speaks of hot dark matter (HDM). Neutrinos, for example, are a type of HDM. However, they can be excluded as the

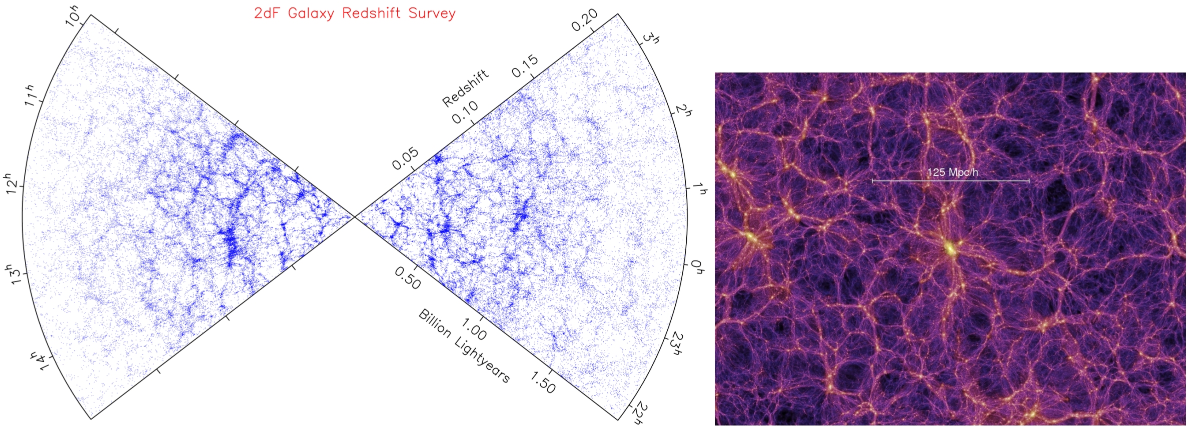


Figure 1.3: *Left: Observations of the cosmic web by the 2dFGRS survey (Colless et al., 2001). Right: The Milenium simulation by Springel et al. (2005). The clustering of large scale structures is in good agreement between observations and simulations. Images from the websites of the two projects accessed on 2015-09-09.*

dominant dark matter component, since they would suppress small scale structure to strongly.

At redshifts $z \ll 100$, structures enter non-linear growth. Structures start growing faster than predicted by LPT, dark matter starts gathering in spherical structures (haloes), and within the dark matter haloes galaxies form, galaxies merge hierarchically and form greater and greater galaxies. As the youngest step of their evolution, they assemble in large clusters of galaxies. At the largest strongly correlated scales, galaxies are distributed along the cosmic web. The cosmic web forms at those scales where density fluctuations just entered non-linear growth $\delta(k) \sim 1$. At scales much larger than the cosmic web ($\sim 100\,\mathrm{Mpc}$), the universe is still relatively homogeneous and has density perturbations $\delta(k) \ll 1$. In Figure 1.3 (left), we show the angular distribution of galaxies versus redshift, as measured by the 2dFGRS survey (Colless et al., 2001).

1.3 Cosmological Simulations

Clearly, non linear structure formation cannot be described by LPT. Cosmological simulations have been developed to investigate the growth of structures in this regime. They follow the evolution of the density field $\delta(\vec{x})$ in real space. The dominant interaction in the growth process of structures is gravity. That is the reason why many simulations discretize the density perturbations, given by LPT at a redshift $z \sim 100$, onto a set of particles, and follow the evolution of the density field only by gravitational interactions between the particles. These simulations are commonly known as N-Body, or dark-matter-only simulations. They can describe the large scale distribution of structures very well (compare Figure 1.3 right). However, on small scales they are not supposed to be correct, since baryons have an important influence here. Much more complex simulations including baryonic physics have been developed, but they involve many uncertainties in the description of unresolved physical pro-

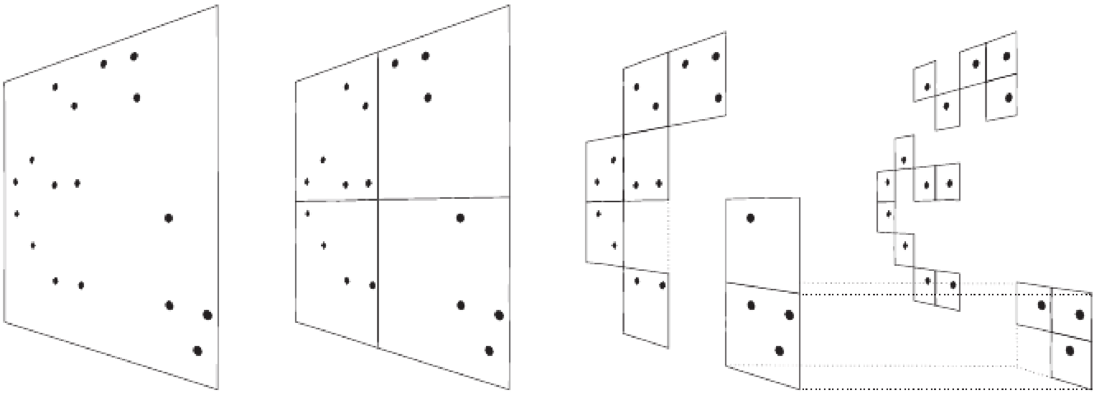


Figure 1.4: *Schematic illustration of the Barnes and Hut oct-tree in two dimensions. Figure from Springel et al. (2001).*

cesses, like cooling, star formation, stellar winds, supernova feedback and black hole feedback.

As we will see, simulations of warm dark matter, already show problems at the level of a dark-matter-only description. As the dark matter distribution is the back bone of all galaxy formation processes, we decided to only address these problems of N-Body simulations, and did not do any simulations with baryonic content.

For the numerical investigations in this thesis we use the code Gadget 3, an updated version of the publicly available code Gadget 2 (Springel, 2005). This is a highly parallelized Tree-PM code for particle based simulations including gravity and optionally Hydrodynamics (SPH) and detailed baryonic physics. The Tree-PM method optimizes the performance of gravitational force calculations at long and short ranges. Long range forces are calculated by the particle mesh (PM). This is a three dimensional grid on which the masses of the particles get deposited. Long range forces are then calculated by solving Poisson's equation on that mesh using Fourier techniques. The PM also allows for simple usage of periodic boundary conditions.

For short-range force-calculations the tree is used. The tree is a hierarchical splitting of the space into nodes. Each node represents a cubic volume element, that consists of eight daughter nodes with half the box size each. This splitting is done recursively up to nodes which only contain one particle in their volume. This is illustrated in Figure 1.4.

The tree is used to summarize the interactions of a big number of clustered particles into a single force evaluation. If a particle interacts with a tree node, either it interacts only with the center of mass of all particles contained in the node, or it opens the node and interacts with all 8 daughter nodes. Whether a tree node is opened, depends on the opening criterion, which can, for example, be an angular criterion where a node is opened if its edge appears under a large angle for the interacting particle.

Chapter 2

The Warmth of Dark Matter

2.1 What is Warm Dark Matter?

So far we do not know a lot about the details of the creation of dark matter particles in the early universe. However, as the early universe was very hot, dark matter particles most likely also carried thermal velocities at their formation. In our current understanding of the universe, dark matter has decoupled very early and remained nearly collisionless since then. As the universe expands, the thermal velocity dispersion of the dark matter particles decreases - dark matter cools down. This seems somewhat surprising on the first glance, as apparently particles loose momentum here in the absence of forces.

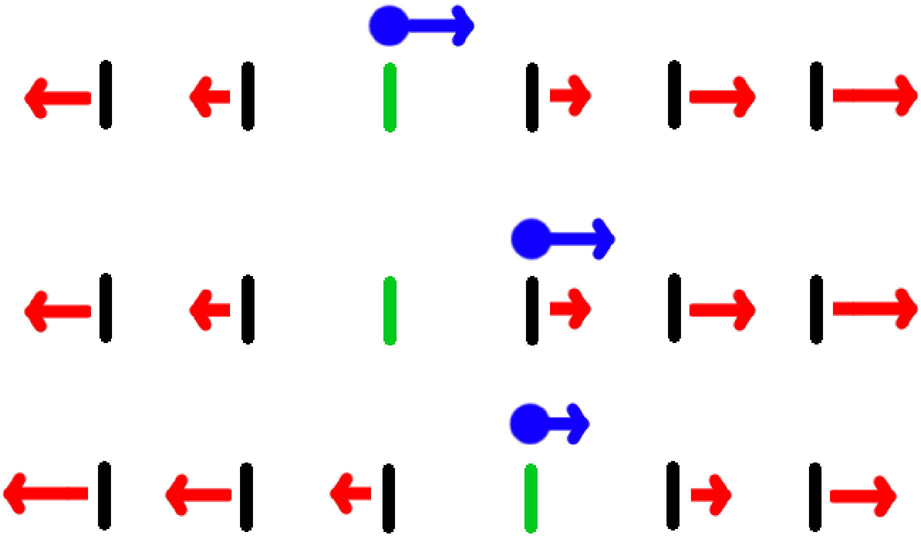


Figure 2.1: *The redshifting of peculiar velocities in an expanding universe. The particle (blue) "looses" velocity just by a change of its reference frame (green).*

This paradox can be dissolved by reconsidering the definition of velocities in an expanding universe. Usually we are talking about peculiar velocities which are the velocities measured from that local coordinate system, which is fixed with respect to the CMB. If a particle moves in an expanding universe (i.e. the rest frames move away from each other), it also continuously changes the rest frame, from which its peculiar velocity has to be measured. Therefore even a free floating particle has a decreasing peculiar velocity in an expanding universe, just caused by the effects of *coordinate transformation*. We illustrate this in Figure 2.1. The expansion of the universe can locally be described by the Hubble flow

$$du = H dx \tag{2.1}$$

where du describes how fast the rest frame at a distance dx appears to be moving away from an observer in a different rest-frame, and H is the Hubble parameter. If we assume to have a free floating particle, moving the distance $dx = v \cdot dt$ in the time dt, then its peculiar velocity decreases by $dv = -du$ due to the transformation to the new coordinate system[2]. Therefore we find

$$\begin{aligned} dv &= -H v dt \\ &= -\frac{1}{a}\frac{da}{dt} v dt \\ &= -v\frac{da}{a} \\ \Leftrightarrow \frac{dv}{v} &= -\frac{da}{a} \\ v &= v_0 \left(\frac{a}{a_0}\right)^{-1} \\ &\propto a^{-1} \end{aligned}$$

which states, that the thermal velocities of non-relativistic particles decrease anti-proportional to the scale factor a. If one uses more generally the Lorentz transformation to calculate the change in velocity

$$v + dv = \frac{v - du}{1 - \frac{v du}{c^2}} \tag{2.2}$$

$$\gamma^2 dv = -du \tag{2.3}$$

$$\frac{dp}{p} = \gamma^2 \frac{dv}{v} \tag{2.4}$$

one can find the general result

$$p \propto a^{-1} \tag{2.5}$$

which includes the energy loss of relativistic particles like neutrinos or photons $E \approx pc$ (redshifting), and the loss in peculiar velocities for non relativistic particles like, e.g., dark matter far past its formation $v \approx p/m$.

---

[2] At this point we assumed a Galilei-transformation, which only holds for the case $v \ll c$.

We now understand that dark matter can have low thermal velocities nowadays, despite having high thermal velocities in the past. However, generally we expect that the dark matter particles still have a (small) thermal velocity. We speak of *cold dark matter* if the thermal velocities are small enough that approximating them by zero makes no notable difference. We speak of *warm dark matter* if the thermal velocities are high enough to influence the formation of structure on observable scales. We want to note here, that this is not a very unambiguous distinction, and what we call cold today, might be something that we call warm tomorrow, as smaller scales might be accessible in the future. We therefore think it is better to think of the *warmth of dark matter* as a property which has to be investigated quantitatively. If we would know how warm dark matter is, we would also know much more about the particle and its formation. Therefore the motivation to investigate the warmth of dark matter, is less to solve dark matter's small scale crisis (see section 2.4), but more to search for clues about what the mysterious dark matter really is.

2.2 Thermal Relics

The mathematically simplest formation scenario for warm dark matter is a dark matter particle which forms in thermal equilibrium (thermal relic), and decouples while being relativistic ($E \gg mc^2$). In the relativistic limit $E \approx pc$ and for formation-annihilation equilibrium (chemical potential $\mu = 0$) the number densities can be simply inferred for bosons n_{BE} from the Bose-Einstein statistics, and for fermions n_{FD} from the Fermi-Dirac statistics:

$$n_{\mathrm{BE}} = \frac{1}{h^3} \int_{R_3} \frac{g_i}{\exp(\frac{pc}{kT}) - 1} d^3p \tag{2.6}$$

$$= \frac{4\pi g_i}{h^3} \int_0^\infty \frac{p^2}{\exp(\frac{pc}{kT}) - 1} dp \tag{2.7}$$

$$x = pc/kT \tag{2.8}$$

$$n_{\mathrm{BE}} = \frac{g_i}{2\pi^2} \left(\frac{kT}{\hbar c}\right)^3 \int_0^\infty \frac{x^2}{\exp(x) - 1} dx \tag{2.9}$$

$$n_{\mathrm{FD}} = \frac{g_i}{2\pi^2} \left(\frac{kT}{\hbar c}\right)^3 \int_0^\infty \frac{x^2}{\exp(x) + 1} dx \tag{2.10}$$

$$\int_0^\infty \frac{x^n}{\mathrm{e}^x - 1} \mathrm{dx} = n!\ \zeta(n+1) \tag{2.11}$$

$$\int_0^\infty \frac{x^n}{\mathrm{e}^x + 1} \mathrm{dx} = n!\ \zeta(n+1)\ (1 - 2^{-n}) \tag{2.12}$$

where $\zeta(m)$ is the Riemann zeta function with $\zeta(3) \approx 1.202$.

$$n_{\mathrm{BE}} = \frac{g_i \zeta(3)}{\pi^2} \left(\frac{kT}{\hbar c}\right)^3 \tag{2.13}$$

$$n_{\mathrm{FD}} = \frac{3}{4} \frac{g_i \zeta(3)}{\pi^2} \left(\frac{kT}{\hbar c}\right)^3 \tag{2.14}$$

Thermal equilibrium is only strictly maintained until interactions become rare and the particle decouples. However, since number densities dilute $n \propto a^{-3}$, and momenta redshift $p \propto a^{-1}$, one can also interpret the statistics at later times by the same distributions with reduced temperature $T(a) = T_D \frac{a_D}{a}$. Then for example for a fermion species like a massive neutrino the density parameter today[3], can be written as

$$\begin{aligned} \Omega_X &= \frac{\rho_X}{\rho_0} && (2.15)\\ &= \frac{8\pi G}{3H^2} n_X m_X && (2.16)\\ &= \frac{8\pi G}{3H^2} m_X \frac{3}{4} \frac{g_X \zeta(3)}{\pi^2} \left(\frac{kT_X}{\hbar c}\right)^3 && (2.17)\\ \Omega_X h^2 &= \left(\frac{g_X}{2}\right) \left(\frac{m_X}{94\,\mathrm{eV}}\right) \left(\frac{T_X}{T_\nu}\right)^3 && (2.18) \end{aligned}$$

where we call T_X the temperature of species X today, $T_\nu = 1.964\,\mathrm{K}$ is the temperature of the neutrino background and m_X is the mass of the particle. Fore example, this formula can be used to constrain the masses of neutrinos from cosmology. For each neutrino species $g_X = 2$, $T_X = T_\nu$, and $\Omega_X \ll 1$, which already excludes a massive neutrino as a warm dark matter candidate, as it cannot lead to a density parameter $\Omega_X \sim 1$ and small enough streaming velocities $v_0 \propto \frac{T_X}{m_X}$ at the same time. A thermal relic WDM particle must have decoupled much earlier than the neutrinos, and the universe must have reheated after its decoupling $T_\nu > T_X$, similar to the temperature increase $T_{\mathrm{CMB}} > T_\nu$ due to the reheating by electron-positron annihilation after the decoupling of the neutrinos. Currently the the mass of a thermal relic is constrained to $m_X > 3\,\mathrm{keV}$ by the observations of Lyman-α forests (Viel et al., 2013). However, we still perform simulations with smaller relic masses, but stay mostly qualitative in our argumentations.

A typical velocity scale for the particles at a given redshift z is

$$v_0 = \frac{kT_X}{mc}(1+z)\,. \tag{2.19}$$

Using (2.18) to relate mass and temperature at given cosmological parameters, we find:

$$\frac{v_0}{1+z} \approx 0.01 \left(\frac{\Omega_x}{0.3}\right)^{1/3} \left(\frac{h}{0.65}\right)^{2/3} \left(\frac{1.5}{g_x}\right)^{1/3} \left(\frac{\mathrm{keV}}{m_x}\right)^{4/3} \mathrm{km/s}\,. \tag{2.20}$$

The degeneracy of 1.5 corresponds to the case of a gravitino, which is one possible thermal relic candidate. This formula is in correspondence with Bode et al. (2001). We note that a warm dark matter particle of $m \sim 1\,\mathrm{keV}$ would have very low thermal velocities of $v_0 \sim 10\,\mathrm{m/s}$ today. However, at a red shift of $z \sim 10^4$, this would correspond to a thermal velocity of $v_0(z = 10^4) = 100\,\mathrm{km/s}$, which is relevant for structure formation. A better quantitative account, for what thermal velocities are relevant for structure formation, is given by the free streaming scale.

---

[3] Assuming it is non relativistic today $\rho \approx nm$.

2.3 Free Streaming

The free streaming scale describes roughly how far dark matter particles can travel by their thermal velocities.

$$\lambda_{\mathrm{fs}} = \int \frac{v_{\mathrm{th}}(t)}{a(t)} dt \tag{2.21}$$

$$= \int \frac{v_{\mathrm{th}}(a)}{a^2 H(a)} da \tag{2.22}$$

Note that there are different definitions of the free streaming scale in literature, but we stick for simplicity with this one for our discussion.

Due to the thermal diffusion of the dark matter particles, below the free streaming scale density fluctuations are strongly suppressed with respect to the cold dark matter case ($v_{\mathrm{th}}(a) \sim 0$). Far above the free streaming scale density perturbations remain unaffected by the thermal velocities. The effects of free streaming on the Power Spectrum have been described by Bode et al. (2001) who suggested the fitting formula

$$T(k) = \left[1 + (\alpha k)^{2\nu}\right]^{-5/\nu} \tag{2.23}$$

with free parameters α and ν, to describe the deviations from a corresponding cold dark matter scenario

$$P_{\mathrm{WDM}}(k) = T^2(k) P_{\mathrm{CDM}}(k) \tag{2.24}$$

From Boltzmann code simulations Viel et al. (2005) obtained $\nu = 1.12$ and

$$\alpha = 0.049 \left(\frac{m_x}{1\,\mathrm{keV}}\right)^{-1.11} \left(\frac{\Omega_\nu}{0.25}\right)^{0.11} \left(\frac{h}{0.7}\right)^{1.22} h^{-1}\mathrm{Mpc} \tag{2.25}$$

as a best fit for $k < 5h\mathrm{Mpc}^{-1}$ which are parameters frequently used in literature. We do not use this spectrum in our simulations, but use the Boltzmann Code CLASS (Blas et al., 2011) to calculate a warm dark matter power-spectrum for the cosmology described in Table 3.1. To show that we are consistent with other investigations of warm dark matter, we plot our power spectra in Figure 2.2 together with the corresponding spectra based on the fitting formula from Bode et al. (2001). We plot the dimensionless power-spectrum here

$$\Delta^2(k) = \frac{k^3}{2\pi^2} P(k) \tag{2.26}$$

as it is easier to interpret than $P(k)$: where the dimensionless power spectrum is of order unity $\Delta^2 \sim 1$, perturbations are deeply in the non-linear regime, and we expect structures to form at these scales. Scales with $\Delta^2 \ll 1$, on the other hand, are in the linear regime and have a smaller impact on the dominant structures. By inspecting the power spectra in Figure 2.2 we see, that the CLASS WDM spectra and the fitting formula agree very well. However, on smaller scales $k \gg k_{fs}$ oscillatory deviations appear. These can probably be understood similarly to the acoustic oscillations in the CMB - with the difference of a strongly enhanced

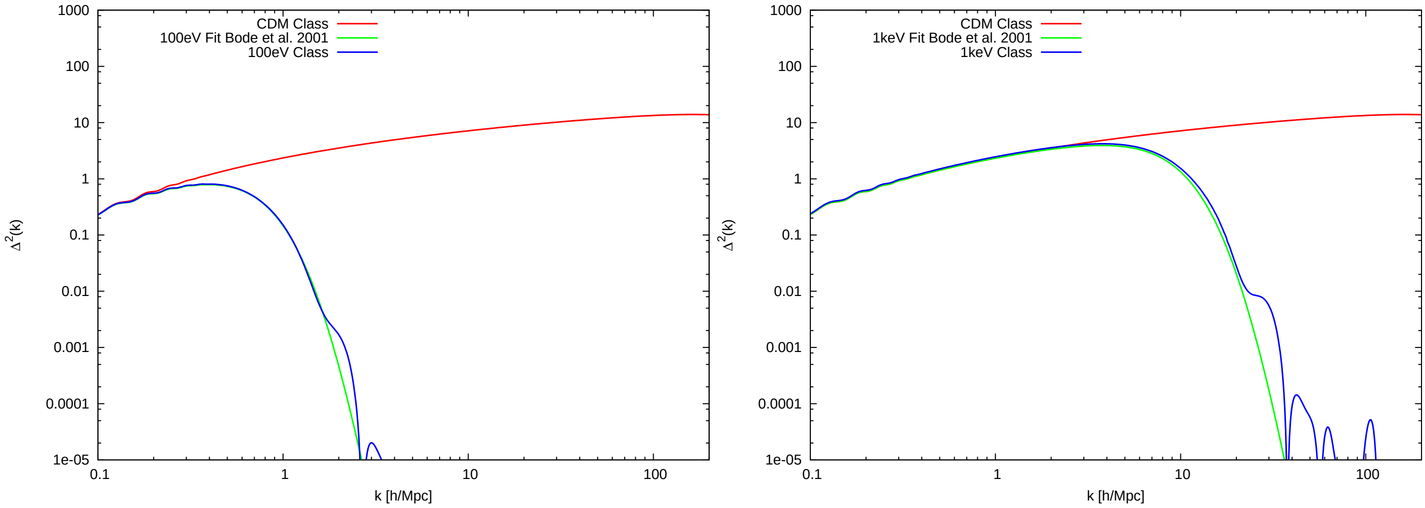


Figure 2.2: *CDM power spectrum in comparison to the WDM powerspectra of a* 100 eV *thermal relic (left) and a* 1 keV *thermal relic (right). The red line corresponds to a CDM power spectrum that we calculated with CLASS, the blue line to a WDM power spectrum we calculated with CLASS, and the green line to a fitting formula frequently used in literature.*

diffusion damping (due to no collisions) and a greatly reduced "sound" horizon (given by the free streaming scale). However, these oscillations are practically irrelevant, as they only appear at strongly suppressed scales $\Delta^2(k) \ll 1$.

From the power spectra we can further anticipate:

- In the CDM scenario the power spectrum Δ^2 is increasing monotonically with k. Therefore smallest scales are expected to collapse first in CDM scenarios, leading to the hierarchical picture of structure formation where larger objects form by the merging and accretion of smaller objects (also known as bottom up structure formation).
- In WDM power spectra there is a maximum in $\Delta^2(k)$. Scales $k \approx k_{\mathrm{max}}$ around the maximum are the first to form structures in a WDM cosmology. The first objects form by a monolithic collapse from a relatively smooth fluid. Larger scales $k < k_{\mathrm{max}}$ form then by hierarchical growth, as usual. However, also some objects form at smaller (slightly suppressed) scales $k \gtrsim k_{\mathrm{max}}$ at later times by the fragmentation of larger objects. This is sometimes referred to as a top down structure formation. Note that modes with $k \gtrsim k_{\mathrm{max}}$ can not be described by LPT anymore, as soon as structures around the free streaming scale are collapsing, even if $\Delta^2(k \gtrsim k_{\mathrm{max}}) \ll 1$. Perturbations of smaller length scales can be amplified during the collapse of larger objects, and lead to the fragmentation of the larger objects (Valinia et al., 1997).

2.4 The Small Scale Crisis of Cold Dark Matter

Although the ΛCDM paradigm seems to work out in remarkable quantitative agreement with observations on large scales, there seem to be several disagreements on small scales (Weinberg

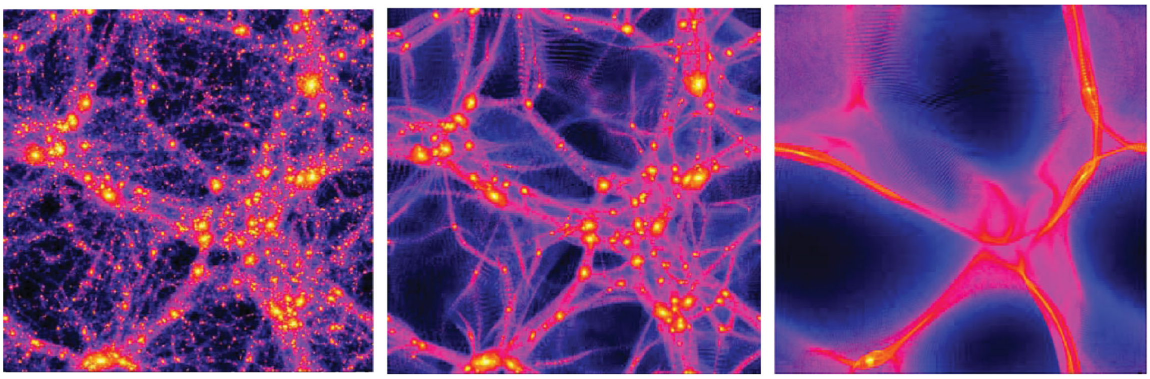


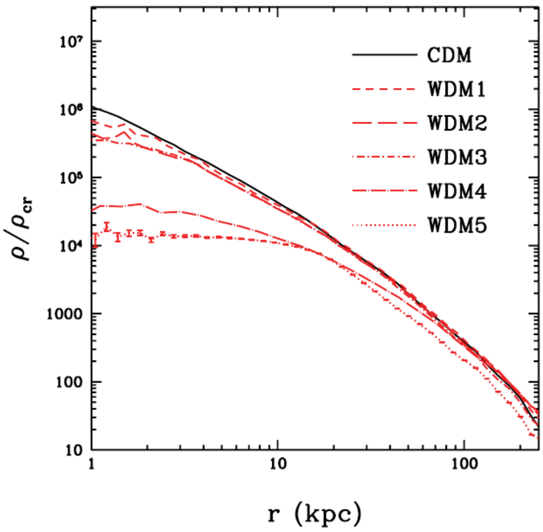
Figure 2.3: *Density maps of* $L = 40\mathrm{Mpc}$ *simulations for CDM, WDM with* $m_X = 0.2\mathrm{keV}$ *and WDM with* $m_X = 0.05\mathrm{keV}$ *- Figure 1 from Macciò et al. (2012). The warmer the dark matter, the larger the scale, at which structures are suppressed.*

et al., 2013):

- The missing satellite problem: Numerical simulations predict far more satellite haloes for a Milky-Way-sized galaxy than satellites are observed.
- The too big to fail problem: Numerical simulations predict the largest sub haloes for a Milky Way type galaxy to be far more massive than the brightest observed satellites of the Milky Way.
- The Core-Cusp problem: Observed Milky Way satellites seem to exhibit a cored central dark matter profile (i.e. it flattens). Numerical dark matter simulations predict cuspy profiles with steep power-laws in the centre.

It is still under debate how to solve this problems properly. They could be related to not properly known baryonic physics in simulations, they could originate from observational difficulties, or they could be an evidence for a deviation from the ΛCDM model.

Warm dark matter has been repeatedly proposed as a solution to the small scale crisis of dark matter. In principle, warm dark matter's suppression of small scale power (compare Figure 2.3) could explain an under-abundance of small haloes, and the thermal velocity dispersion could lead to cores in the centres of dark matter haloes. However, recent research indicates that the thermal cut-off might be too steep to solve for these problems (Weinberg et al., 2013). For example Macciò et al. (2012) did re-simulations from the same 2 keV initial conditions, just varying the thermal velocities of their particles. Indeed they found that thermal velocities can lead to cores in their halo, like can be seen in Figure 2.4. However, also they find that the thermal velocities that are needed to form an appreciable core, imply a relic mass for which the power spectrum is suppressed so far, that the galaxy would not form in the first place. They summarize this as a 'Catch 22' problem for WDM: *If you want a large core you won't get the galaxy, if you get the galaxy it won't have a large core.*



Label	m_X (keV)
CDM	-
WDM1	2.0
WDM2	0.5
WDM3	0.2
WDM4	0.1
WDM5	0.05

Figure 2.4: *Core profiles of WDM Haloes - Figure 2 from Macciò et al. (2012).*

We will not attempt to solve the small scale crisis of dark matter in this thesis. We mention the small scale crisis here, as it has been the motivation of many WDM studies in the past. We are more interested in understanding the principle impact of the warmth of dark matter on the formation of structures. We see the warmth of dark matter more as a property of dark matter, than as a viable solution to the small scale crisis. In the grand project of finding out what dark matter is, its warmth could be an important factor. Note, that even a typical CDM candidate like a weakly interacting massive particle (WIMP) has a thermal velocity dispersion. For a WIMP this might suppress haloes on mass scales like $10^{-6}M_{\odot}$ (Green et al., 2005).

Chapter 3

Simulations of Warm Dark Matter

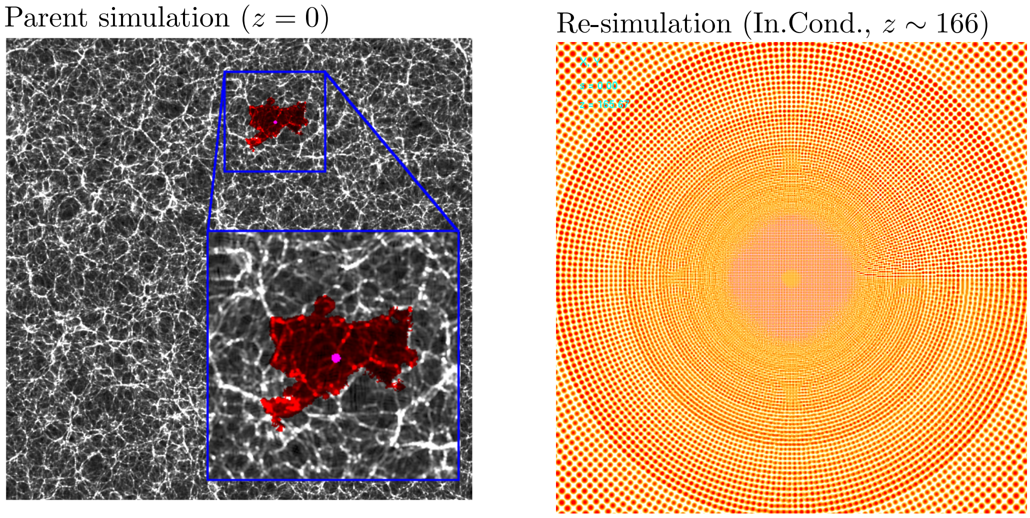
In this section we want to present simulations of warm dark matter with zoomed-initial-conditions. Therefore we explain the concept of zoomed initial conditions, describe the process of structure formation for different relic masses, point out the numerical problem of artificial haloes in WDM simulations, and try to solve for it with Adaptive Gravitational Softening.

3.1 Visualization

We implemented a visualization-code, that can read particle data from Gadget snapshots, and render them very efficiently utilizing the GPU. We do not provide a detailed description of the code, as this would go beyond the scope of this thesis. However, all visualization pictures within this thesis (if not stated differently in an image's caption), have been made with this visualization-code. We want to mention that it can interpolate snapshots, integrate densities along the line of sight, average arbitrary quantities along the line of sight, realize 2d or 3d projections, do phase space projections, and read and utilize subfind output, amongst many other features. This visualization has turned out very useful several times. For example, it helped to find out whether something is going wrong on first sight, gave us often a very intuitive impression of a situation, or it helped to get very quickly a good overview over a large number of different output quantities.

3.2 Zoomed Initial Conditions

We set up a simulation with zoomed initial conditions. This is done by selecting a halo from a cosmological simulation, tracing the halo's particles back into the initial conditions, refining the mass resolution in the region around the proto-halo, adding small-scale perturbations to the high-resolution region that are not present in the parent simulation, and coarsening the mass resolution for regions more distant from the proto-halo. This creates a zoom-in affect like can be seen in Figure 3.1. This whole procedure is done by the ZIC-Code (Tormen et al., 1997). Running the simulation with the new initial conditions (re-simulation), the same halo gets reproduced again - with higher mass resolution, correct long-range forces, and at comparably low costs.


Parent simulation ($z = 0$)

Re-simulation (In.Cond., $z \sim 166$)

Figure 3.1: *Left: The parent simulation Dianoga and the halo for the re-simulation which has been selected to be inside a void. Halo Selection and image from Schlachtberger (2014). Right: Zoomed Initial Conditions at $z \sim 166$. The initial conditions for the re-simulated halo lie in the high resolution region in the centre. Going outwards from the high resolution region, the mass resolution is decreasing, thereby imitating the long range forces from the rest of the box. The same region can be seen in Figure 3.2 (left) in an evolved stage.*

We use as parent simulation the cosmological dark matter only box Dianoga described by Borgani and Viel (2009). The simulation's parameters are listed in table 3.1. We re-simulate the halo "Frog" which was selected by Schlachtberger (2014) to be in a void of the parent simulation. We perturb the high-resolution region by the different spectra of CDM, WDM (1keV) and WDM (100eV) that we showed in Figure 2.2. Using this different spectra for the high resolution region does not introduce any inconsistencies between high- and low-resolution region, as long as the power spectra of the parent simulation and the high-resolution region match at all modes which are resolved in the parent simulation. Only modes with $k < k_{\mathrm{Nyq}}$ are resolved in the parent simulation with the Nyquist frequency

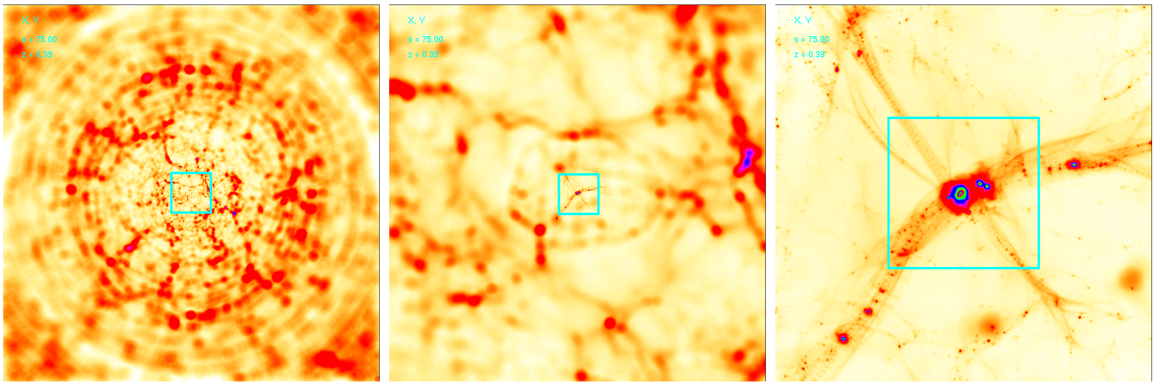
$$k_{\mathrm{Nyq}} = 2 \cdot \frac{1024}{1h^{-1}\mathrm{Gpc}} \approx 2h\mathrm{Mpc}^{-1}\,. \tag{3.1}$$

Inspecting Figure 2.2 once again, we find that the 1 keV scenario perfectly matches the cold dark matter scenario up to that wave number. However, for the 100 eV scenario this is not the case anymore. Therefore, the 100 eV simulation will be somewhat inconsistent. However, we will still include it here. This should be reasonable, as long as we stay qualitative with our argumentations, and keep in mind that a real 100 eV scenario should probably contain even less structure than we see in our simulations. In Figure 3.2 we illustrate the re-simulation by showing a zoom-in into the high -resolution region of the 100 eV simulation at $z \sim 0.4$.

Our re-simulations have $N_{\mathrm{LR}} \approx 2.2 \cdot 10^6$ low-resolution and $N_{\mathrm{HR}} = 1.9 \cdot 10^7$ high-resolution particles. Each high-resolution particle carries a mass of $m = 4 \cdot 10^6 h^{-1} M_{\odot}$.

Table 3.1: *Cosmological and numerical parameters of the parent simulation (Borgani and Viel, 2009).*

Parameter	Ω_m	Ω_Λ	Ω_b	h	σ_8	n_s	Boxsize	N	m_{part}
Value	0.24	0.76	0.04	0.72	0.8	0.96	$1h^{-1}$Gpc	1024^3	$6.2 \cdot 10^{10} h^{-1} M_\odot$



1 Gpc/h 100 Mpc/h 10 Mpc/h

Figure 3.2: *Zoom from the low-resolution regions into the high-resolution region in the centre at $z \sim 0.4$. The rectangle indicates the region shown in the next image. Only the central 4 Mpc which are marked in the right image, are shown in subsequent images. The left image shows the same region that can be seen in Figure 3.1 (right) in the initial conditions.*

3.3 Structure Formation

We evolve the initial conditions with the Gadget 3 code. The thermal velocity dispersion of warm dark matter is neglected from the beginning of the simulation. This can be done, because the thermal velocities are expected to be small during the simulated late stages of structure formation. The impact of the free streaming onto the initial power spectrum is much larger than late time free streaming. Deviations should be quantitatively small, and qualitatively irrelevant.

We visualize the process of structure formation for the three considered scenarios of CDM, WDM1 (1 kev) and WDM2 (100 eV) in Figure 3.3. We summarize all observations we make by the visual inspection of the formation processes:

- In CDM first structures are forming at the visible scales already well before $z = 9$. The first collapse in the CDM scenario will be always at the smallest resolved scales. For WDM1 the first collapse is happening around $z \sim 9$ and for WDM2 around $z \sim 4$.
- Large structures are mostly unaffected by the warmth of dark matter.
- The warmer the dark matter, the less substructure appears.
- In the warm dark matter scenarios it can be nicely seen that haloes are formed at the nodes of the cosmic web.

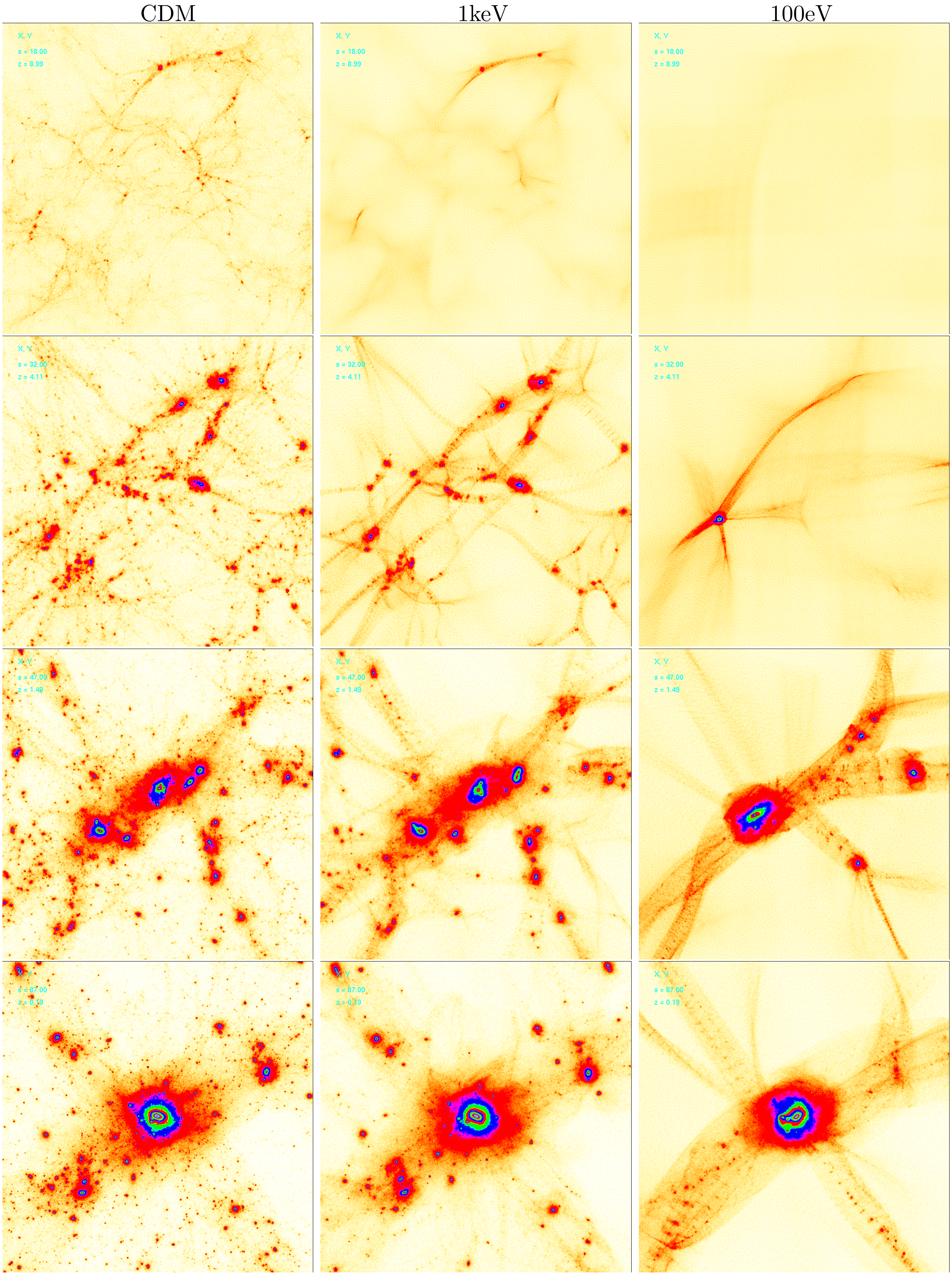
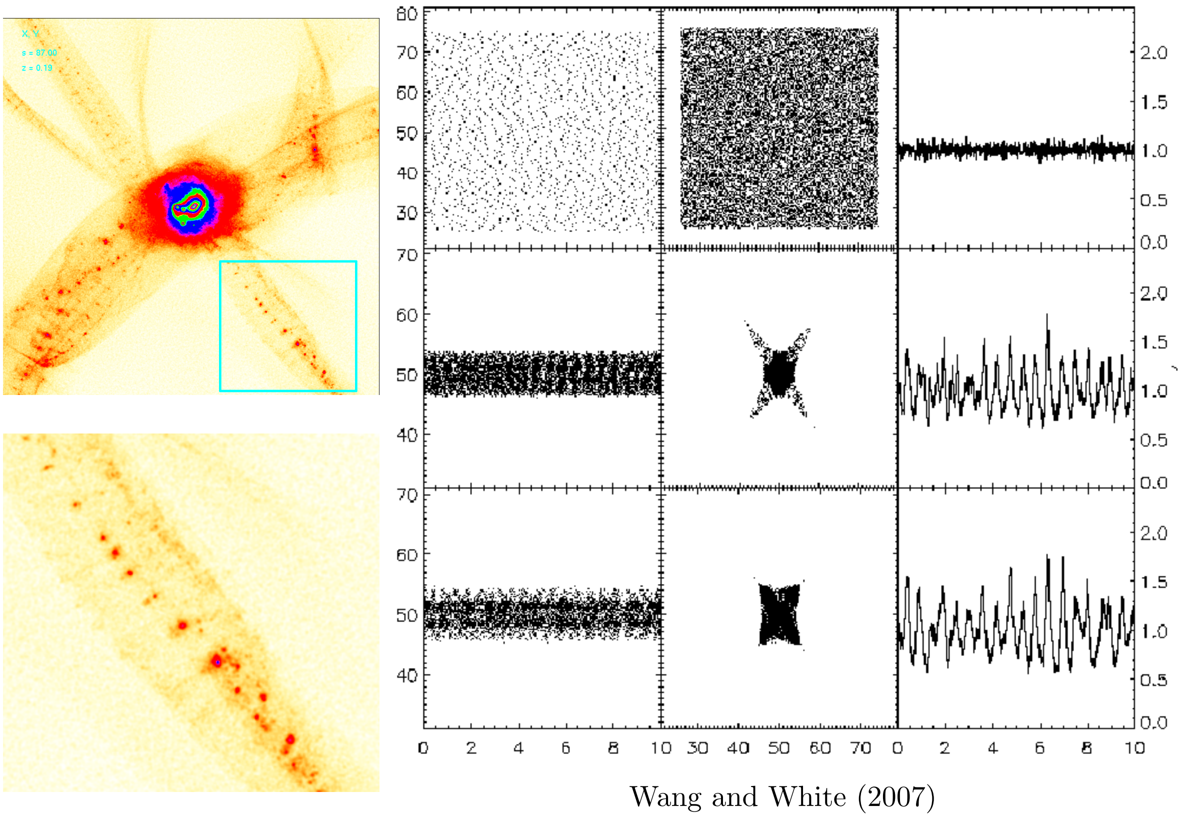


Figure 3.3: *Density projection in a* $4 \times 4 \times 2$ *Mpc volume of the re-simulated halo for three different dark matter scenarios at different redshifts. Left column CDM, centre column WDM with* $m_x = 1\,\mathrm{keV}$*, right column WDM with* $m_x = 100\,\mathrm{eV}$*. Redshifts from top to bottom:* $8.99, 4.11, 1.49, 0.19$.



Wang and White (2007)

Figure 3.4: *Artificial Haloes in WDM simulations: Left a zoom into artificial haloes in a filament from our simulations. Right: The collapse of a homogeneous filament from Wang and White (2007) Figure 6. The left column shows the filament from the top, the central column a cross-section, and the right column the density along the filament. Top to bottom shows different points in time. As even the collapse of a homogeneous filament produces this artificial structures at separations of the mean particle distance, it becomes clear that these structures are not physical, but only an artefact of the simulation methods.*

- The cosmic web and features like caustics ($\sim$ sharp edges) are more visible in the warm dark matter cases, as the dark matter fluid is much smoother in these cases.
- Small evenly spaced clumps can be seen in the filaments of the warm dark matter simulations. These are artificial haloes, as we will discuss in the next section.

3.4 Artificial Haloes

The evenly spaced haloes which we show in more detail in Figure 3.4 (left), have been identified as artificial, numerical structures. This becomes clear, as they, for example, do not appear at the same locations if the resolution is scaled up (which physical objects do), or their particle's positions traced back into the initial conditions, form very unusual oblate shapes (Lovell et al., 2014). In Figure 3.4 (right) we also show the collapse of a homogeneous filament from Wang and White (2007). As regularly spaced clumps seem to be forming in

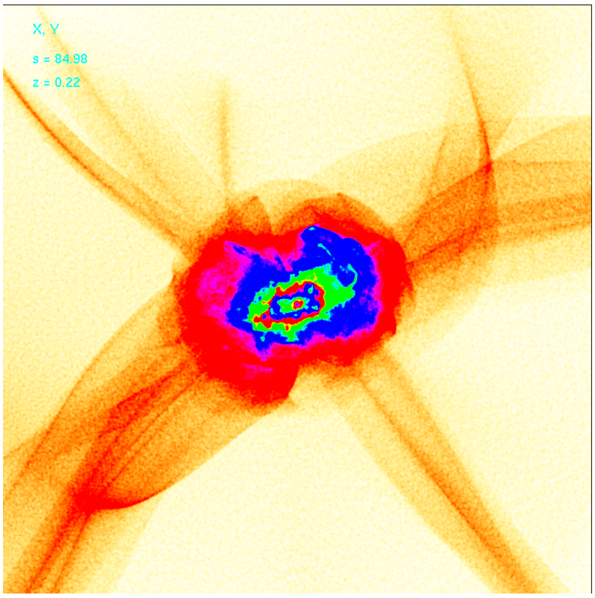


Figure 3.5: *Same setup like in Figure 3.4 left, but with a softening* $\epsilon = 2d = 100\,\mathrm{kpc}$. *A very large softening can eliminate the artificial structures in the filaments. However, also the structure of the halo gets strongly distorted, as forces are greatly underestimated.*

numerical simulations even for the discretized version of a homogeneous filament, N-Body simulations seem to exhibit a numerical deficit here.

There are in principle three different ways of dealing with artificial haloes. (1) Discriminating them from physical haloes in post processing steps. For example, Lovell et al. (2014) discriminated between artificial and physical haloes by the shape of their proto haloes, and obtained a halo mass function without artificial haloes. (2) Developing simulation techniques that do not form artificial haloes. Such techniques have been developed and used in Hahn et al. (2013), Angulo et al. (2013), and Hahn and Angulo (2015). We will discuss this in more detail in the next chapter. (3) Simply ignoring them. As we originally planned to be perform simulations with baryons, hydrodynamics and star formation, we did not consider (1) and (3) as attractive options. E.g., star formation histories could be very wrong if galaxies grow by the merging of small artificial galaxies that are hosting stars, instead of the smooth accretion of gas. Therefore we decided to try approach (2), i.e. to aim for a simulation technique that does not form artificial haloes. Our first idea on this was to use Adaptive Gravitational Softening, as we will discuss in the next section.

3.5 Adaptive Gravitational Softening

In N-Body simulations forces between particles are usually only chosen as point mass interactions $F \propto r^{-2}$ if particles are separated by a distance $r \gg \epsilon$ where ϵ is the softening length. If particles come close to each other $r \lesssim \epsilon$ their interactions are softened. This way non-physical two-body scattering [4] can be reduced. Typically the softening ϵ is chosen much smaller than the mean particle separation $\epsilon \ll d$. However, this way the density field will be granular, and the particle discretization will have an impact on the force fields. In Figure 3.5 we show that a very large softening $\epsilon \sim 2d$ can in principle solve the problem with the artificial haloes. However, such a large softening leads also too far to low force resolution in dense regions like haloes.

Therefore we decided to use adaptive gravitational softening (Iannuzzi and Dolag, 2011) which infers the softening for each particle individually from the local density

$$\epsilon_i \propto (\rho(\vec{x}_i))^{-1/3} \tag{3.2}$$

This way the softening adapts locally to the mass resolution, and can have larger values in low density regions, while having small values in high density regions.

Our first simulations with AGS showed rather buggy behaviour. The high-resolution region was torn apart by huge unphysical accelerations of low-resolution particles at its boundaries. We investigated this behaviour, and found that the unphysical accelerations were changing with the number of processors, like it can be seen in Figure 3.6. Clearly there was a problem with the parallelization. We found out that if the pre-compiler-directive PLACEHIGHRES-REGION was used together with the AGS, an exportation of the particle type was missing. When PLACEHIGHRESREGION is active, two different forcecuts between long-range and short-range forces are used for different particles depending on their type. As the type of a particle was not known on a different processor, a wrong force cut was chosen for all interactions from a different processor. We fixed this bug, so that the AGS now also works together with the PLACEHIGHRESREGION directive, which is mostly used in resimulations.

Having fixed the problems with the AGS-Code, we test the AGS on the same setup like in Figure 3.5. We show the line-of-sight averaged softening in Figure 3.7. Clearly artificial haloes also emerge in this situation. An explanation, why the AGS does not help out, is given by Hobbs et al. (2015), as illustrated in Figure 3.8. Hobbs et al. (2015) have implemented an adaptive mesh refinement method which calculates the inertia tensor in each grid cell, to get a measure for the local anisotropy. If the anisotropy is too high, they prohibit any further refinement, even if the particle number is high enough to trigger a refinement for classical refinement criteria. Apparently they can perform simulations that do not suffer from artificial haloes. However, we are not sure whether their way of decreasing the force resolution has any advantages over just using a large constant softening $\epsilon \sim d$. Further, it remains unclear, why the inertia tensor should be able to determine anisotropy consistently. For example a Poisson distribution[5], that is contracted by a factor of two along one direction, can locally not be distinguished from a Poisson distribution with just twice the number of particles. Actually the term anisotropy is nothing that can be defined consistently for an infinitesimal volume element in configuration space, as a smooth density field will always look homogeneous in a small enough volume element. What Hobbs et al.

---

[4] Doing N-Body simulations one should always keep in mind that the massive particles of $m \sim 10^{10} M_\odot$ are representing huge ensembles of much smaller particles. They represent segments of a continuous fluid.

[5] which is homogeneous in the continuum limes $N \to \infty$

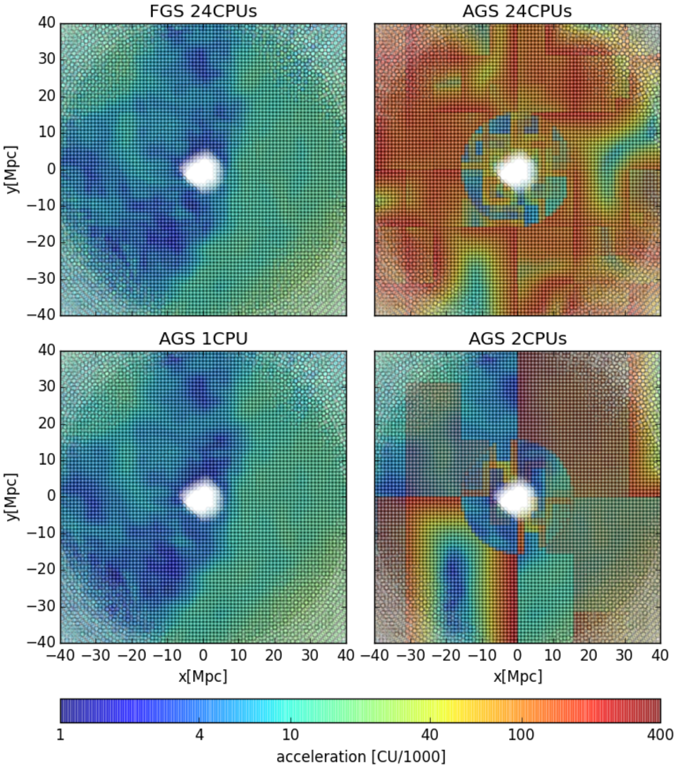


Figure 3.6: *Acceleration ($z = 180$) of the low-resolution particles (high-resolution region in the middle is cut out in this plot) for different scenarios that we tested to debug the AGS-Code. The number of processor changed the shape of the errors, which indicated that it was a parallelization problem. The spherical boundary also seen in this plots is where the high-resolution setup changes from particles of type 2 (constant mass) to particles to type 3 particles (mass increasing outwards). It can be seen that the errors are also correlated with the particle type.*

(2015) implied with the notion anisotropy by showing their Figure 1, is something that can only be defined for arbitrary distributions, if the history of the particles is considered. We start therefore with the same paradigm - i.e. the anisotropy has to be considered - but choose a very different way to define it, estimate it, and to use it to adapt the force resolution.

Therefore we explain the thinking concept of *the dark matter sheet* in phase space in the next chapter, which helps to get a clear notion, of what could be meant by anisotropy. We also explain the distortion tensor (Vogelsberger et al., 2008), which is a consistent way to follow the three dimensional anisotropic distortions of small volume elements in dark matter simulations. Based on a clear notion of anisotropy we then define, describe and implement Anisotropic Softening in chapter 5.

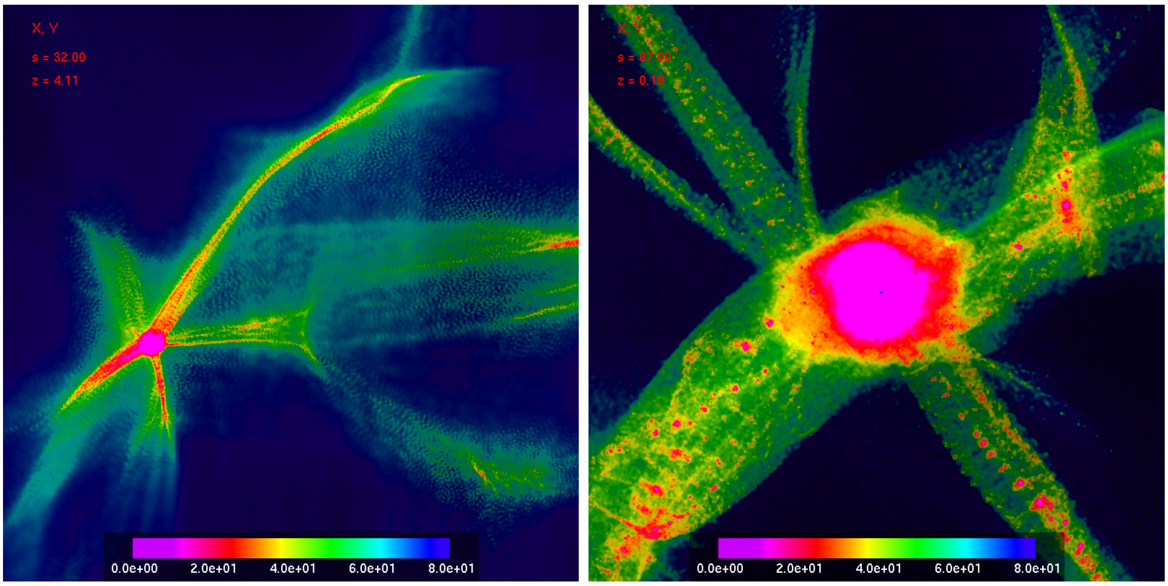


Figure 3.7: *Size of the softening in colour (note the colourbar scale is* 2.8ϵ*) for our 100eV WDM simulation using the AGS. Left at first filament collapse at redshift* $z = 4$ *and right at* $z = 0.2$*. The AGS does not help reducing the amount of artificial haloes, as it becomes far too small during anisotropic collapse. (We draw particles with a size* $\epsilon_{vis} \propto \epsilon$ *here so that high density regions are visually resolved better, and low density regions show less granularity than in prior visualisations with fixed particle size.)*

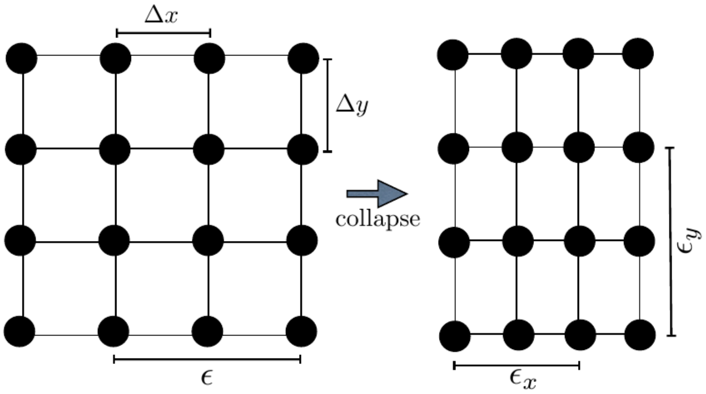
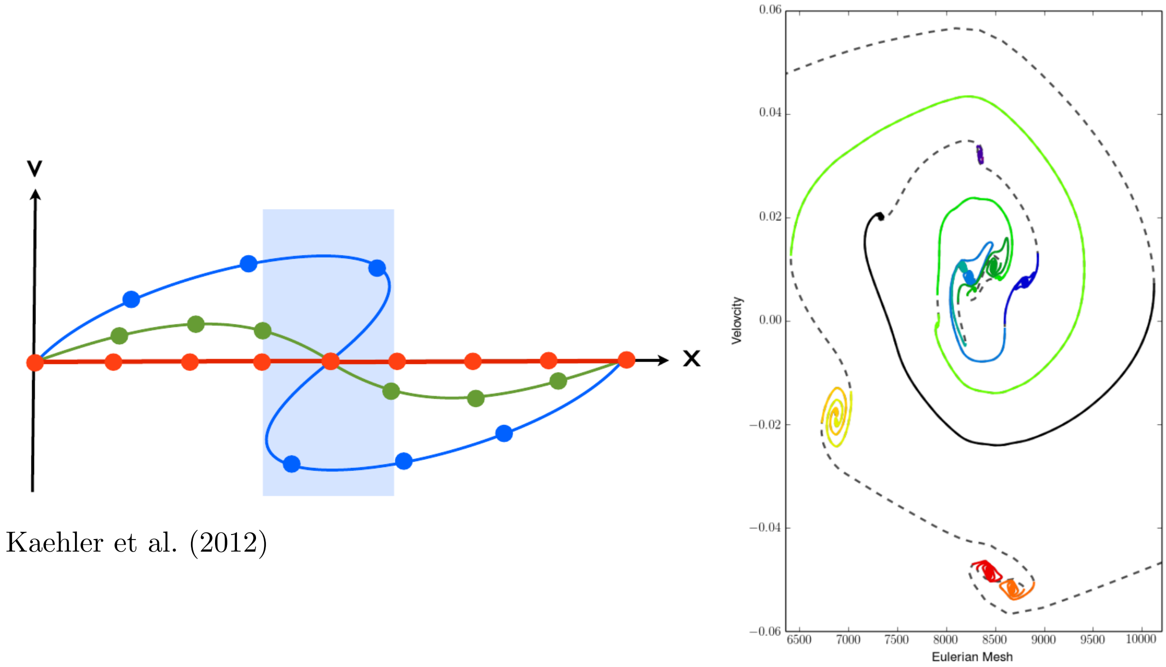


Figure 3.8: *Figure 1 from Hobbs et al. (2015). During anisotropic collapse the particle spacing changes in one direction* Δx*, while it stays the same in the other direction* Δy*. However, AGS is isotropic and inferred from the density and therefore will increase the force resolution in both* x *and* y *direction during this collapse. For an anisotropic collapse the force resolution in adaptive softening does not follow the true mass resolution. To follow the mass resolution correctly a softening would need to be adaptive in each direction individually.*

Chapter 4

The Dark Matter Sheet



Kaehler et al. (2012)

Shandarin and Medvedev (2014)

Figure 4.1: *Two illustrations of the dark matter sheet in phase space. Left (Figure 1 from Kaehler et al. (2012)): The early evolution of a 1D dark matter sheet. For* $z \to \infty$ *(red) the velocities become very small* $v \to 0$ *and dark matter can be clearly identified as a one dimensional sheet in phase space. As time evolves perturbations grow (green) and at some point in time dark matter streams can cross each other (blue), so that particles from different origins can be found at a single point in space* x*. Going to even later stages (right) the dark matter sheet goes on folding and stretching, thereby increasing its complexity in phase space. One might be able to describe it roughly by a coarse grained velocity dispersion and a real space density then. However, if one looks close enough, the dark matter fluid always remains to be a sheet (i.e., something one-dimensional in two-dimensional phase space here).*

In the last chapter we have seen that Adaptive Gravitational Softening fails to match mass and force resolution properly. In this chapter we want to explain the necessary notions to understand, what actually could be meant by *the true mass resolution* of a simulation. For this it is necessary to get a better understanding of the intrinsic nature of the collisionless dark matter gas, i.e. the dark matter sheet. Further we want to motivate here what is needed for a simulation which matches mass and force resolution, and thereby might not suffer by the rise of artificial haloes. Therefore we here develop the motivational basis and the needed notions for what we call *Anisotropic Softening*.

Cold dark matter has no intrinsic thermal velocity dispersion (per definition). Therefore, if we go to high redshifts where linear perturbation theory holds, and everything is very homogeneous, the dark matter fluid has a single valued velocity field (which can be expressed as a function $\vec{v}(\vec{x})$). Therefore *dark matter gas forms a three-dimensional submanifold in a six-dimensional phase space.* In this context one speaks of *the dark matter sheet.* As time evolves, the sheet goes through several phases of collapsing and stretching, thereby greatly increasing the complexity of its structure. However, looking at its fine-grained phase-space structure, it will never loose its three dimensional nature. We illustrate this for the simpler case of an one dimensional dark matter fluid in a two dimensional phase-space in Figure 4.1. Although this argumentation is only strictly valid for cold dark matter, it can also be applied to warm dark matter, by noting that the thermal velocities are very small compared to the bulk velocities. This is valid in the same sense like approximating a real sheet of paper (which has a thickness) as a two dimensional surface, is correct in most situations.

In Figure 4.2 we illustrate how the folding of the dark matter sheet looks in real space. This Figure is based on the anti-symmetrically perturbed wave described by Valinia et al. (1997). We describe this setup in more detail in section 6.1.

4.1 Density Estimates

In numerical simulations the dark matter fluid has to be discretized in some way. In N-Body simulations this is done by a finite set of tracer particles. Based on the tracer particles a density estimate is defined, from which gravitational forces are inferred via Poisson's equation. In the classical N-Body approach each particle is thought of as a δ-function

$$\rho_{\mathrm{NBody}}(\vec{x}) = \sum m\delta(\vec{x} - \vec{x}_i) \tag{4.1}$$

To reduce effects from unphysical collisions when particles come close to each other, the δ functions are replaced by kernel profiles

$$\rho_{\mathrm{NBody}}(\vec{x}) = \sum mW(|\vec{x} - \vec{x}_i|, h) \tag{4.2}$$

where $W(r, h)$ is a kernel function, h is the kernel support radius where the density becomes zero. For a cubic spline $h = 2.8\epsilon$ where ϵ is the softening. The softening ϵ is usually chosen much smaller than the mean particle separation. In Figure 4.3 left we illustrate the N-Body density estimate. The N-Body density estimate is usually poor (e.g. it is exactly zero in the largest part of the volume). However, in N-Body simulations not the density estimate

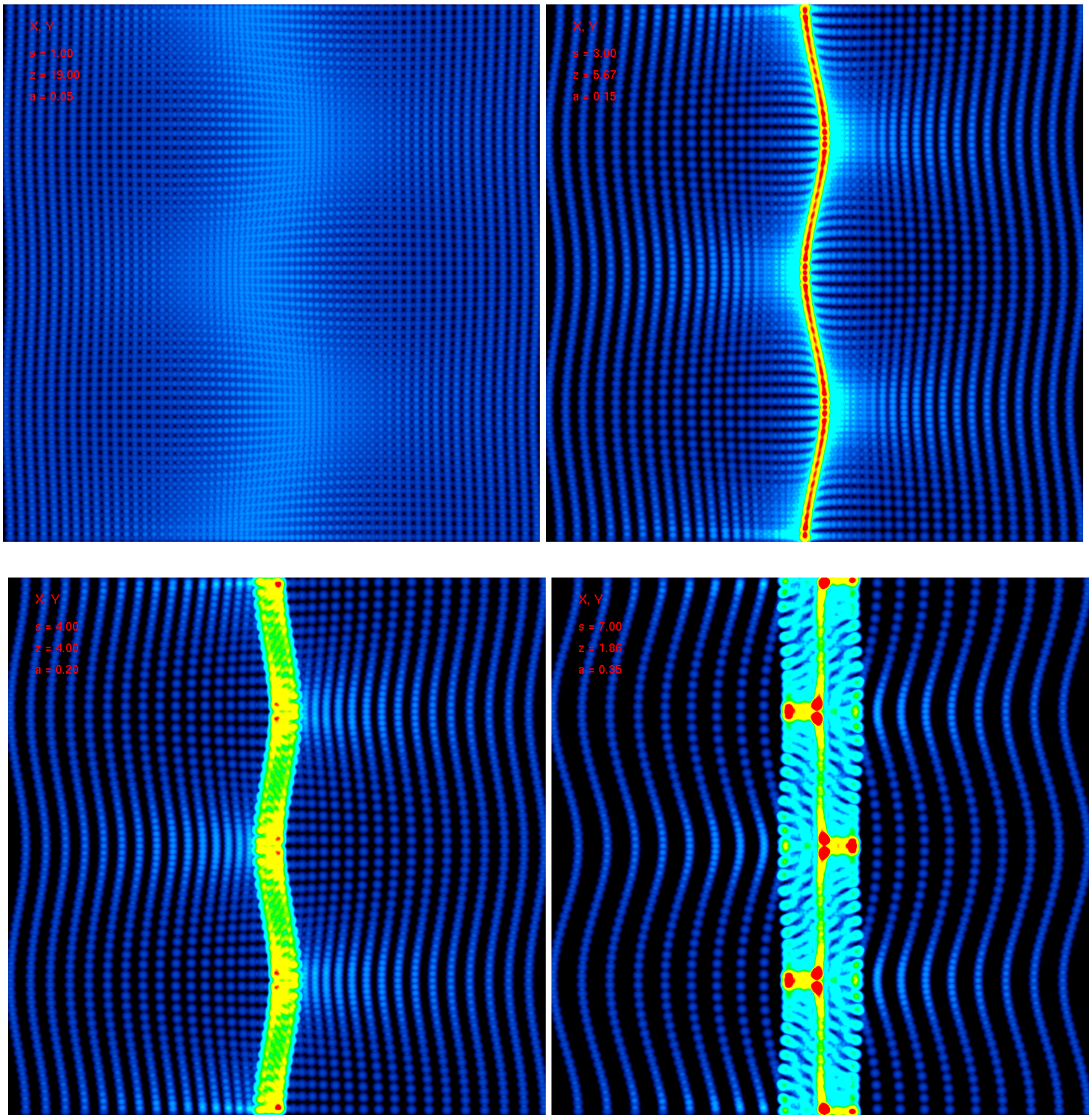
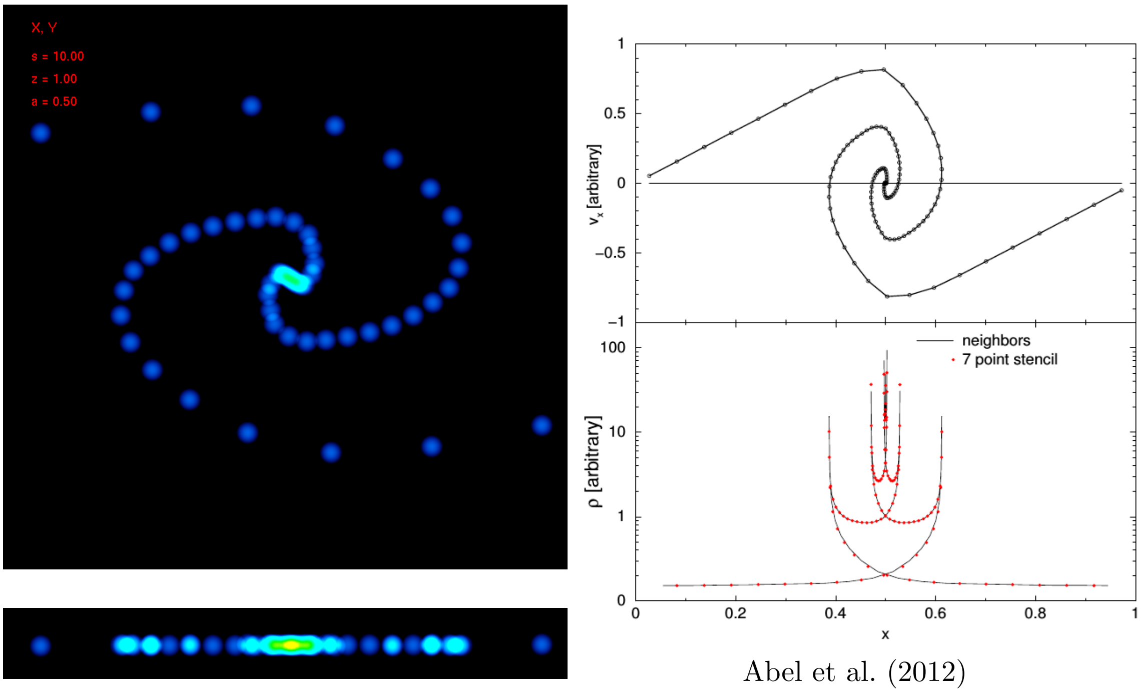


Figure 4.2: *Collapse of an antisymmetrically pertubed plane wave (Valinia et al., 1997) for different expansion factors:* $a = 0.05$ *(top left),* $a = 0.15$ *(top right),* $a = 0.20$ *(bottom left) and* $a = 0.35$ *(bottom right). Here we can see how the folding of the dark matter sheet looks in real space (i.e. the formation of a pancake). At later stages (bottom right) one also sees the collapse of the sheet in the second dimension, thereby forming a filament along the z-direction.*

matters, but the forces which are inferred from it. These are of higher accuracy, as they are of the order of an integrated function of the densities. However, the granularity of the N-Body density field still leads to two-body forces which would not appear in the continuum limit if the softening is smaller than the local mass resolution. Further, problems can arise where the softening is larger than the local mass resolution, i.e. forces get underestimated.



Abel et al. (2012)

Figure 4.3: *Phase space diagrams (x versus vx) of the Zel'dovich Pancake. Left: A typical N-Body density estimate using particles with softenings $\epsilon \ll d$ where d is the mean particle separation. Right: The Zel'dovich Pancake with a density estimate based on the volume elements between the particles. The top panels show the sheet in phase space whereas the bottom panels show the projected real space density (left), and the single stream real space densities (right).*

Clearly, it is desirable to have a method which adapts more precisely to the true local mass resolution.

A density estimate with an adaptive resolution can be obtained by choosing a variable softening length $\epsilon \propto \rho^{-1/3}$. Such a density estimate is commonly used to interpret the results from N-Body simulations or for visualization purposes. It also has been used to calculate forces in N-Body simulations, and is called Adaptive Gravitational Softening in this case (Iannuzzi and Dolag, 2011). However, as early collapses are highly anisotropic, and as it does not use any information about the phase space nature of the fluid, it also fails to match the local mass distribution properly (see e.g. Figure 4.5 left).

A very different way to infer a density estimate from discrete particles has been proposed by Abel et al. (2012). They proposed to think of the particles as massless tracers of the dark matter sheet. The mass of the fluid lies not within the particles, but within the dark matter sheet in between. The particles can be used to reconstruct the dark matter sheet, and to infer a density estimate that follows the deformations of the sheet more accurately. In Figure 4.3 (right) it is illustrated, how this can be done in one dimension: by connecting Lagrangian the dark matter sheet can be reconstructed in phase space. What Lagrangian neighbours are, is defined by the initial conditions for $a \rightarrow 0$ and can, in principle, be inferred from the particle IDs at all times. The connections define small one-dimensional volume elements (lines). The density can then be inferred by the projection of those volume elements onto real space. If the particles are initially on a grid and the fluid is initially homogeneous, then all volume elements contain the same mass. As trajectories can not cross in phase space, the mass in each volume element is conserved. The projected real space density for each volume element is then given at later times by $\rho = m/\Delta x$ where Δx is the distance between the Lagrangian neighbours defining the element. As several Lagrangian volume elements can be at the same location, the total density is then inferred by summing over all single stream densities which are at the same location. Note that this density estimate can, for example, describe caustics ($\Delta x \rightarrow 0$, $\rho \rightarrow \infty$) correctly, whereas in a softening based density estimate, densities stay always limited. This density estimate can easily be generalized to three dimensions. The volume elements can, for example, be replaced by a tessellation of Lagrangian space into tetrahedra, and the single stream densities become $m/\Delta V$.

This tetrahedra-based density estimate has been used in Kaehler et al. (2012) to improve visualizations, in Hahn et al. (2013) to improve simulations and in Angulo et al. (2013) to perform first WDM simulations that do not suffer from the formation of artificial haloes. Further the tetrahedra can be used for interpreting the results from N-Body simulations on a qualitatively new level. E.g. they have been used to define the flip-flop in Lagrangian space which describes how often a volume element has gone through a caustic (Shandarin and Medvedev, 2014). This could be used to identify substructures. Further the multi-stream field can be defined by the number of tetrahedra present at each point in space which can, for example, be used to distinguish between haloes, filaments and pancakes (Ramachandra and Shandarin, 2015). As a state of the art example of what can be done by considering the real nature of the dark matter sheet, we want to refer to Hahn and Angulo (2015) who have developed a code for dark matter simulations that reconstruct the dark matter sheet to even higher orders (than the linear interpolation by tetrahedra), and that can adaptively refine the resolution in Lagrangian space by placing additional flow tracers. The potential of phase space methods is huge.

A density estimate based on volume elements that follow the deformations of the dark matter sheet, seems to be superior in situations where the resolution is high enough to reconstruct the sheet precisely. Also inferred simulation methods seem to be able to avoid the artificial fragmentation known from WDM simulations (Angulo et al., 2013). We want try a conceptually similar approach where we follow the deformations of small ellipsoidal volume elements around each particle by the Geodesic Deviation Equation (Vogelsberger et al., 2008) (Vogelsberger and White, 2011), and infer forces by solving the Poisson equation for these ellipsoids. In the next section we explain the GDE, and why it can be used for an improved density estimate.

4.2 The Geodeosic Deviation Equation

The Geodesic Deviation Equation (GDE) is used to follow the evolution of the displacement vector between two in phase space infinitesimally separated particle trajectories.

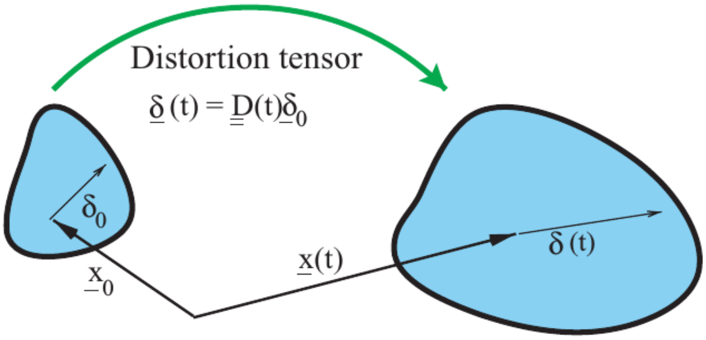


Figure 4.4: *Figure from Vogelsberger et al. (2008). The real space distortion tensor describes the evolution of a small spatial displacement vector. As the displacement can be taken in any direction, it effectively follows the distortion of a small volume element around a particle.*

$$\begin{pmatrix} \Delta\vec{x} \\ \Delta\vec{v} \end{pmatrix}(t) = \overline{\overline{D}}(t) \cdot \begin{pmatrix} \Delta\vec{x}_0 \\ \Delta\vec{v}_0 \end{pmatrix} \tag{4.3}$$

where $\overline{\overline{D}}(t)$ is the 6 × 6 phase space distortion tensor[6]

$$\overline{\overline{D}} = \begin{pmatrix} \partial\vec{x}/\partial\vec{x}_0 & \partial\vec{x}/\partial\vec{v}_0 \\ \partial\vec{v}/\partial\vec{x}_0 & \partial\vec{v}/\vec{v}_0 \end{pmatrix} =: \begin{pmatrix} \underline{\underline{D}}_{xq} & \underline{\underline{D}}_{xp} \\ \underline{\underline{D}}_{vq} & \underline{\underline{D}}_{vp} \end{pmatrix} \tag{4.4}$$

Initially the distortion tensor is (by definition) a unit tensor. The evolution of the distortion tensor is described by the GDE which is a first order 6 × 6 differential equation connecting the temporal change of the distortion tensor to itself and the tidal tensor. We do not want to go into the mathematical details here - these are described in Vogelsberger et al. (2008) for the Newtonian case, and have been generalized in Vogelsberger and White (2011) to an expanding universe. However, it is important to know that the 3 × 3 tidal tensor is defined by the second derivatives of the potential

$$T_{\mathrm{ij}} = -\frac{\partial^2\phi}{\partial x_i \partial x_j} \tag{4.5}$$

and has to be evaluated additionally to follow the evolution of the distortion tensor. We use the implementation of the GDE scheme in Gadget3 from Vogelsberger and White (2011) which integrates the GDE in an expanding universe beneath the usual integration of the particle trajectories to obtain the distortion tensor for each particle at all times.

In this thesis we are only interested in the real space distortion tensor defined by the evolution of a small positional displacement in the dark matter sheet $\vec{\delta}$

$$\vec{\delta}(t) = \underline{\underline{D}}(t)\vec{\delta}_0 \tag{4.6}$$

---

[6] We adopt the notation from Vogelsberger and White (2011) here, where 3 × 3 tensors have two under-lines, and 6 × 6 tensors two over-lines.

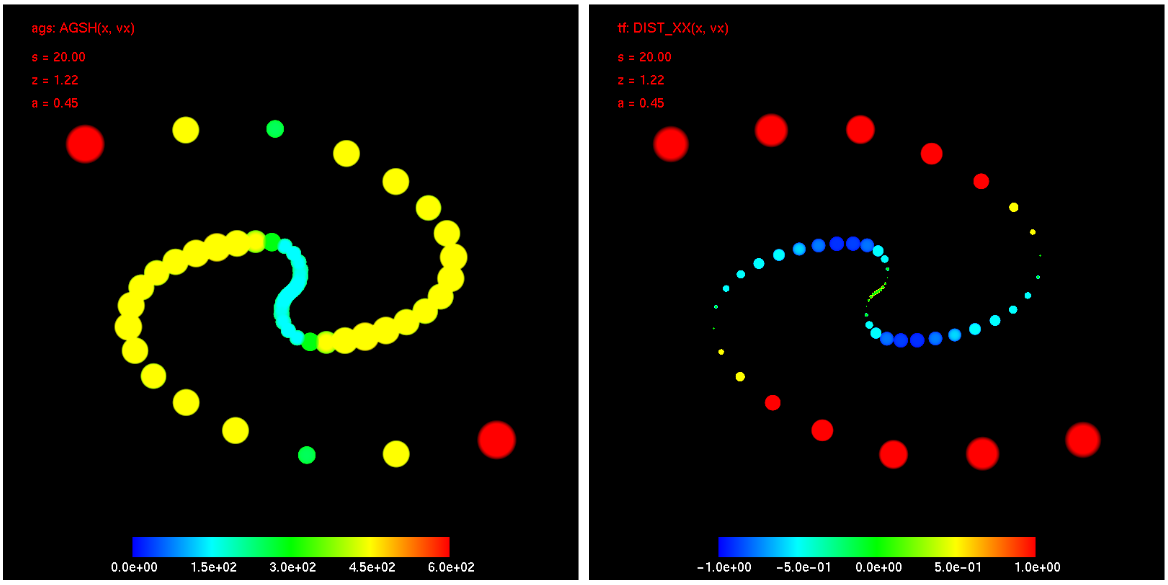


Figure 4.5: *Phase space diagram x-v_x of the Zel'dovich Pancake. Left: color and radii are proportional to the softening inferred from the AGS code. Right: Colours code the D_{xx} component of the distortion tensor, and the sizes of the particles were chosen proportional to the $|D_{xx}|$ component, to illustrate that the distortion tensor follows the deformations of small volume elements. The AGS density estimate makes no differences between particles from different streams at similar positions x. Further, softenings are too big in the x direction as the volume element is reshaped isotropically although only the spacing in x direction changes in the Zel'dovich Pancake. (Note: The particle distributions were explicitly chosen to be the same. They are not the results from different simulations.)*

The real space distortion tensor is equal to $\underline{\underline{D}}_{xq}$ plus a correction which considers that a small displacement in the spatial coordinate $\vec{\delta}$ also implies a small displacement in the velocity coordinate for particles within the dark matter sheet (for an illustration compare 4.1 the green line). However, this correction is small if the initial conditions are taken early, and therefore we always approximate

$$\underline{\underline{D}} \approx \underline{\underline{D}}_{xq} \tag{4.7}$$

for simplicity. As the displacement $\vec{\delta}_0$ can be taken in any direction, the real space distortion tensor effectively follows the deformations of a small volume element of the dark matter sheet (illustrated in Figure 4.4). Generally regular real 3×3 matrices map spheroids onto ellipsoids (in more detail in section 5.1). Therefore, if we think of a small initially spherical volume element around each particle, the real space distortion tensor describes, how it is distorted to an ellipsoidal shape at later times.

To get an intuitive understanding of how this can be used to obtain a density estimate that follows the dark matter sheet more precisely, we show the D_{xx} component of the distortion tensor for the Zel'dovich Pancake in Figure 4.5 (right). If we assign an ellipsoidal mass profile to each particle defined by the distortion tensor, we get a softening that follows deformations in every spatial direction individually. In the next section we describe what is needed to infer forces and tidal forces from a density estimate based on the distortion-ellipsoids. We also implement this in Gadget 3 and call it *Anisotropic Softening*.

Chapter 5

Anisotropic Softening for Gadget 3

We define the density field

$$\rho(\vec{r}) = \sum_i \rho_{el}(\vec{r} - \vec{r}_i, a_i, b_i, c_i, \vec{e}_{a,i}, \vec{e}_{b,i}, \vec{e}_{c,i}) \tag{5.1}$$

where the index i runs over different particles, a, b, and c are the semi-axes of ellipsoidal volume elements around the particles, $\vec{e}_a$, $\vec{e}_b$, $\vec{e}_c$ their orthonormal spatial orientations, and ρ_{el} is the single-ellipsoid density profile which is given in the eigensystem of the ellipsoid

$$\vec{r}_e = \begin{pmatrix} \vec{e}_a^T \\ \vec{e}_b^T \\ \vec{e}_c^T \end{pmatrix} (\vec{r} - \vec{r}_i) \tag{5.2}$$

as

$$\rho_{el}(x_e, y_e, z_e) = \rho\left(u = \sqrt{\frac{x_e^2}{a^2} + \frac{y_e^2}{b^2} + \frac{z_e^2}{c^2}}\right) \tag{5.3}$$

$$\tag{5.4}$$

where $\rho(u)$ is a kernel function which we here choose to be the Epanechnikov kernel

$$\rho(u) = \rho_0(a, b, c) \begin{cases} (1 - u^2) & \text{if } u \leq 1 \\ 0 & \text{if } u > 1 \end{cases} . \tag{5.5}$$

With *Anisotropic Softening* we mean a force calculation inferred from this density field. In the following sections we develop what is needed to implement Anisotropic Softening into Gadget 3. Therefore we (1) explain how we define the ellipsoids $(a, b, c, \vec{e}_{a,i}, \vec{e}_{b,i}, \vec{e}_{c,i})$ by the distortion tensor, (2) solve Poisson's equation for the Epanechnikov-ellipsoid in its eigensystem, (3) derive the corresponding forces and tidal tensors, and (4) discuss details about a numerical efficient implementation.

5.1 The Distortions of the Ellipsoids

We are given the real and regular 3×3 matrix D (the real space distortion tensor) that generally maps a sphere with radius r_0 onto an ellipsoid with semi-axes a, b, c. We want to

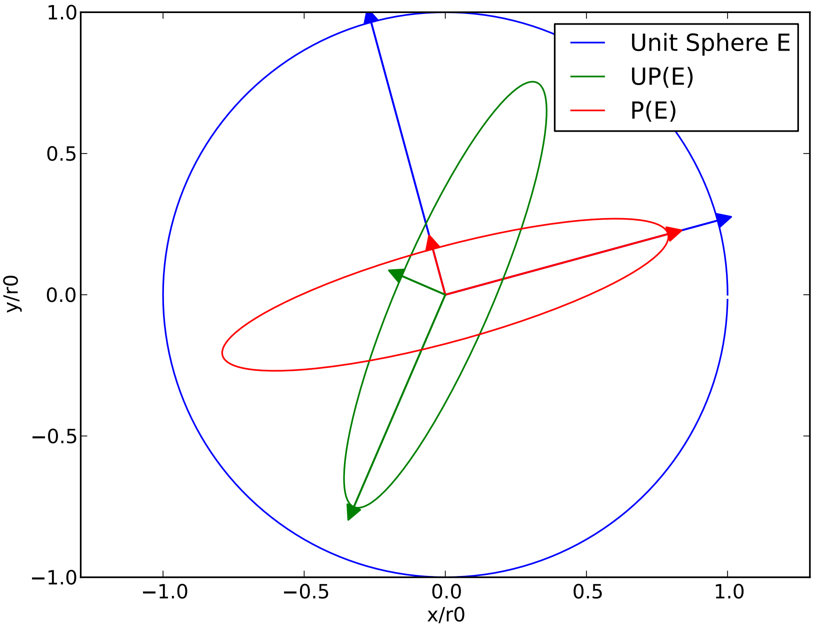


Figure 5.1: *Illustration of the polar decomposition. The matrix P stretches the unit sphere (blue) to an ellipsoidal shape (red) along the directions of its eigenvectors $\vec{e}_{pi}$ by the factor of its eigenvalues λ_{pi}. The blue arrows indicate $\vec{e}_{pi}$ and the red $\lambda_{pi}\vec{e}_{pi}$. The matrix U rotates and mirrors the ellipsoid to the final product of the mapping $D = UP$. The green arrows indicate $U \cdot \lambda_{pi}\vec{e}_{pi}$.*

determine the semi-axes and their spatial orientation.

To any real, regular matrix D the polar decomposition

$$D = U \cdot P \tag{5.6}$$

can be applied, where U is an orthogonal and P is a symmetric matrix. Therefore we can understand the acting of the matrix D as two special successive mappings. The symmetric tensor P acts as a stretching along the directions of the eigenvectors $\vec{e}_{pi}$ by a factor given by the eigenvalues λ_{pi} of P. The acting of the orthogonal tensor can be generally a rotation and/or mirroring. We illustrate this in Figure 5.1. Thus the semi-axes of the ellipsoids are given by the eigenvalues λ_{pi} of P

$$a_i = r_0|\lambda_{pi}| \tag{5.7}$$

and the spatial orientations of the axes are given by the mapped eigenvectors of P

$$\vec{e}_i = U \cdot \vec{e}_{pi}\,. \tag{5.8}$$

Note that $\vec{e}_i$ are mathematically not the eigenvectors of the distortion tensor and λ_{pi} not its eigenvalues (they are its singular values). However, we will often refer to them as the eigenvectors and the eigenvalues of the distortion-ellipsoid. A transformation of a displacement vector $\vec{r} = \vec{x} - \vec{x}_0$ to the eigensystem of the ellipsoid is given by

$$\vec{r}_e =: V^T \cdot \vec{r} \tag{5.9}$$

$$= \begin{pmatrix} \vec{e}_0^T \\ \vec{e}_1^T \\ \vec{e}_2^T \end{pmatrix} \cdot \vec{r} \tag{5.10}$$

We do not have to do the polar decomposition explicitly to obtain λ_{pi} and $\vec{e}_i$. They can be obtained by a singular value decomposition

$$D = V \Sigma W^* \tag{5.11}$$

into the orthogonal matrices V and W^* and a diagonal matrix Σ. The λ_i are then the diagonal elements of Σ, and V corresponds to the same matrix like in (5.10). In our implementation into Gadget we use the routine *gsl_linalg_SV_decomp()* from the GSL library to obtain the singular value decomposition of the real space distortion tensor, and the thereby defined ellipsoid $(a_i, \vec{e}_i)$.

To get a limited parameter space, we limit the axes of the ellipsoids into the range

$$a, b, c \in [r_0 \cdot f_{\min}, r_0 \cdot f_{\max}] \tag{5.12}$$

Which ensures that the axis ratios of the ellipsoids do not exceed $q_{\max} = f_{\max}/f_{\min}$. The limiting rescaling factors can be defined via the pre-compiler directives $f_{\min}$ = AS_MINRESCALE and $f_{\max}$ = AS_MAXRESCALE. Typically we will choose them in a way that $q_{\max} \sim 10^2$.

r_0 is the kernel support radius for the initial sphere. We define this parameter in terms of the softening

$$r_0 = 1.171324 \cdot \epsilon \tag{5.13}$$

which is an input parameter of the code. Note that the relation between r_0 and ϵ is chosen in a way that for a sphere the standard deviation of the Epanechnikov kernel matches the standard deviation of a cubic-spline kernel with the softening ϵ:

$$\sigma^2 = \frac{1}{N} \int r^2 \rho(r, H) d^3 r \tag{5.14}$$

$$\sigma_{\text{Ep}} = r_0 \sqrt{3/7} \tag{5.15}$$

$$= \sigma_{\text{Cub. Spline}} = 2.8\epsilon \cdot \sqrt{4/40} \tag{5.16}$$

Where we mean by H the kernel support radius which is r_0 for our initial sphere and 2.8ϵ for the cubic spline. This way the softening can be used similar to the traditional way despite the different kernel.

5.2 The Potential of an Ellipsoid

To define an anisotropic (ellipsoidal) softening, we want to solve Poisson's equation (setting $G = 1$) for a single ellipsoid in its eigensystem:

$$\Delta\phi(x, y, z) = 4\pi\rho(x, y, z) \tag{5.17}$$

where $\rho(u)$ is a mass distribution with ellipsoidal symmetry

$$\begin{aligned} \rho(x, y, z) &= \rho(u(x, y, z)) \\ u(x, y, z) &= \sqrt{\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 + \left(\frac{z}{c}\right)^2} \end{aligned} \tag{5.18}$$

and we call $\rho(u)$ the kernel function. There is a wide range of literature on the potential of a homogeneous ellipsoid:

$$\rho(u) = \begin{cases} \rho_0 & \text{if } u \leq 1 \\ 0 & \text{if } u > 1 \end{cases}$$

However, we want to derive the potential of an ellipsoid for the more general case of an arbitrary kernel function here. Therefore we start from the potential of a uniformly charged ellipsoidal shell given in Cai (2007) and derive the kernel-ellipsoid potential. After that we apply this to the two cases of a homogeneous ellipsoid and an ellipsoid with an Epanechnikov kernel. Note that our derivations are largely inspired by Cai (2007) and can be seen as generalization of their derivation of a homogeneous ellipsoid potential.

5.2.1 Ellipsoidal Shell

The potential of an ellipsoidal shell (Cai, 2007) with semi-axes (a, b, c), constant density ρ, and thickness Δu in the u coordinate is given by

$$\psi(\vec{x}) = \psi(\lambda(\vec{x})) = 2\pi abc\rho\Delta u \int_\lambda^\infty \frac{ds}{\sqrt{(a^2 + s)(b^2 + s)(c^2 + s)}} \tag{5.19}$$

where λ is an ellipsoidal coordinate and defined as the greatest root of the equation

$$\frac{x^2}{a^2 + \lambda} + \frac{y^2}{b^2 + \lambda} + \frac{z^2}{c^2 + \lambda} = 1 \tag{5.20}$$

for points outside the ellipsoidal shell $(u > 1)$, whereas for points inside the ellipsoidal shell $(u \leq 1)$ $\lambda = 0$. We do not need to understand any details about ellipsoidal coordinates here, but note that ellipsoidal coordinates form an orthogonal curvilinear coordinate system and parametrize the whole space outside the ellipsoid by three real coordinates, all given as roots of (5.20). The largest root λ can be understood as a distance measure from the ellipsoidal shell and it reflects the symmetry of the problem, that the other two ellipsoidal coordinates do not appear in (5.19). Thus the equipotential lines are also lines of constant λ. The integral in (5.19) can not be evaluated analytically. Therefore our plots of potentials are usually based on numeric integrations of high accuracy, using the python routine *scipy.integrate.quad()*.

We illustrate the potential and the forcefield $\vec{a} = -\nabla(\phi)$ (evaluated by finite differences) for an ellipsoidal shell with $M = 4\pi abc\rho\Delta u = 1$ in Figure 5.2 for two different ellipsoids $\vec{A}_1 := (a, b, c) = (2, 0.5, 1.0)$ and $\vec{A}_2 := (2, 0.02, 1.0)$. Note that large axis ratios like in $\vec{A}_2$ are also expected to appear in collisonless dark matter simulations. For example in caustics one axis approaches zero, while the other axes stay finite. Like can be seen in the right panel of Figure 5.2 the potential has nearly a planar symmetry close to the flat side of the ellipsoid then. It can easily be seen that for large radii:

$$\begin{aligned}
&r \gg a > b > c \\
&\lambda \to r^2 \\
&\phi \to 2\pi abc\rho\Delta u \int_{r^2}^{\infty} \frac{ds}{\sqrt{s^3}} \\
&\quad = -2\pi abc\rho\Delta u \frac{2}{r} \\
&\quad = -\frac{M}{r}
\end{aligned}$$

which is the potential of a point mass. This behaviour can also be seen qualitatively in Figure 5.2: Inside the shell the potential is constant ($\lambda = 0$). Outside but close to the shell the potential imitates the shape of the ellipsoid. Far away from the shell it becomes more and more spherical and imitates the potential of a point mass. Note however, that, unlike a spherical mass distribution, the potential of an ellipsoid corresponds at no radius *exactly* to the potential of a point mass.

5.2.2 Ellipsoids with Kernel

The potential of the ellipsoidal shell (5.19) can be generalized to a shell with rescaled semi-axes

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} \to \begin{pmatrix} u \cdot a \\ u \cdot b \\ u \cdot c \end{pmatrix}$$

$$\Delta u \to \frac{\Delta u}{u}$$

$$\psi(\vec{x}, u) = 2\pi abc\rho u^2 \Delta u \int_{\lambda(u)}^{\infty} \frac{ds}{\sqrt{(u^2a^2 + s)(u^2b^2 + s)(u^2c^2 + s)}} \tag{5.21}$$

where we defined

$$\lambda(u): \qquad 1 = \frac{x^2}{u^2a^2 + \lambda(u)} + \frac{y^2}{u^2b^2 + \lambda(u)} + \frac{z^2}{u^2c^2 + \lambda(u)} \tag{5.22}$$

We can substitute $s = u^2 v$, $ds = u^2 dv$ to obtain:

$$\psi(\vec{x}, u) = 2\pi abc\rho u \Delta u \int_{\lambda(u)/u^2}^{\infty} \frac{dv}{\sqrt{(a^2 + v)(b^2 + v)(c^2 + v)}} \tag{5.23}$$

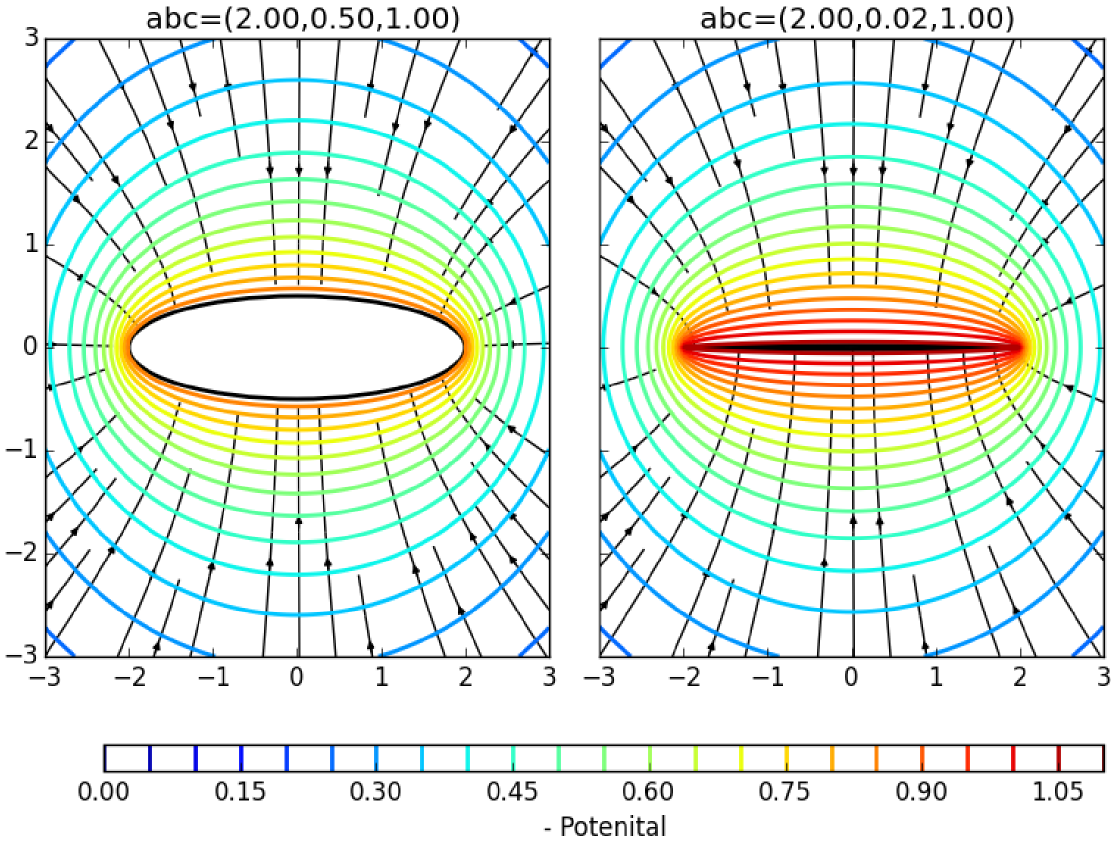


Figure 5.2: *Potential (contours) and Forces (black lines with arrows) for two uniformly charged ellipsoidal shells with different semi-axes. The internal potential is constant, close to the shell the equipotential lines imitate the shape of the shell, and far from the ellipsoid the potential becomes similar to the spherical symmetric potential of a point mass.*

Given the potential of an ellipsoidal shell, it is straightforward to derive the potential of more general mass distributions. Since Poisson's equation is linear, the potential of an ellipsoidal mass distribution can be taken as the sum of a set of ellipsoidal shells with different densities $\rho(u)$ and rescaled semiaxes (ua, ub, uc). However, a distinction must be made between shells which see $\vec{x}$ as an internal point ($\lambda = 0$), and shells which see it as an external point ($\lambda > 0$, given by (5.22)). We define u_0 as the shell which discriminates these two types - i.e. the shell that passes through $\vec{x}$:

$$u_0^2 = \frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} \tag{5.24}$$

Then we can write for the potential of an extended kernel-ellipsoid:

$$\begin{aligned}
\phi(\vec{x}) &= \int_0^\infty \frac{\psi(\vec{x},u)}{\Delta u} du \\
&= \int_0^{u_0} \frac{\psi_{\text{ext}}(\vec{x},u)}{\Delta u} du + \int_{u_0}^\infty \frac{\psi_{\text{int}}(\vec{x},u)}{\Delta u} du \\
\phi(\vec{x}) &= 2\pi abc \left(\int_0^{u_0} u\rho(u) \int_{\lambda(u)/u^2}^\infty \frac{dv}{\sqrt{f(v)}} du + \int_{u_0}^\infty u\rho(u) \int_0^\infty \frac{dv}{\sqrt{f(v)}} du \right) \qquad (5.25)
\end{aligned}$$

where we abbreviated

$$f(v) := (a^2+v)(b^2+v)(c^2+v) \qquad (5.26)$$

We can generally not perform the v integration, as this integral has no analytic solution. However, we can perform the u integration on the term $u\rho(u)$. We call its parent function $G(u) := \int_0^u u\rho(u)$. Separating the integrals for the contribution of the shells with $u > u_0$ we get:

$$\int_{u_0}^\infty u\rho(u) \int_0^\infty \frac{dv}{\sqrt{f(v)}} du = (G(\infty) - G(u_0)) \int_0^\infty \frac{dv}{\sqrt{f(v)}} \qquad (5.27)$$

For the shells with $u < u_0$ we get, integrating by parts and calling $z(u) := \lambda(u)/u^2$:

$$\int_0^{u_0} u\rho(u) \int_{z(u)}^\infty \frac{dv}{\sqrt{f(v)}} du = \left[G(u) \cdot \int_{z(u)}^\infty \frac{dv}{\sqrt{f(v)}} \right]_0^{u_0} - \int_0^{u_0} G(u) \left(\frac{d}{du} \int_{z(u)}^\infty \frac{dv}{\sqrt{f(v)}} \right) du$$

We note that the definitions of z, λ and u_0 imply:

$$\begin{aligned}
z(u) &= \lambda(u)/u^2 \\
z(u_0) &= 0 \qquad \text{(to see this compare (5.22) and (5.24))} \\
z(0) &\to \infty
\end{aligned}$$

Therefore the first term evaluated at $u = 0$ contains an integration from ∞ to ∞ and is thus zero (and also we defined $G(0) = 0$). Further we can perform the derivative of the inner integral in the right term by using

$$\frac{d}{du} = \frac{dz}{du}\frac{d}{dz}$$

$$\begin{aligned}
\int_0^{u_0} u\rho(u) \int_{z(u)}^\infty \frac{dv}{\sqrt{f(v)}} du &= G(u_0) \cdot \int_0^\infty \frac{dv}{\sqrt{f(v)}} + \int_0^{u_0} G(u) \frac{dz}{du} \frac{1}{\sqrt{f(z(u))}} du \\
&= G(u_0) \cdot \int_0^\infty \frac{dv}{\sqrt{f(v)}} + \int_{z(0)}^{z(u_0)} G(u(z)) \frac{1}{\sqrt{f(z)}} dz \\
&= G(u_0) \cdot \int_0^\infty \frac{dv}{\sqrt{f(v)}} + \int_\infty^0 G(u(z)) \frac{1}{\sqrt{f(z)}} dz \\
&= \int_0^\infty \frac{G(u_0) - G(u(v))}{\sqrt{f(v)}} dv \qquad (5.28)
\end{aligned}$$

where we substituted from u to z and renamed the silent variable z to v in the end, so that we could summarize the integrals again. Putting the terms together we find

$$\begin{aligned}\phi(\vec{x}) &= 2\pi abc\left((G(\infty)-G(u_0))\int_0^\infty \frac{dv}{\sqrt{f(v)}}+\int_0^\infty \frac{G(u_0)-G(u(v))}{\sqrt{f(v)}}dv\right)\\ &= 2\pi abc\int_0^\infty \frac{G(\infty)-G(u(v))}{\sqrt{f(v)}}dv\end{aligned}$$

Which is the main result of this section. We summarize:

$$\phi(\vec{x}) = 2\pi abc\int_0^\infty \frac{G(\infty)-G(u(v,x,y,z))}{\sqrt{(a^2+v)(b^2+v)(c^2+v)}}dv \tag{5.29}$$

$$G(u) = \int_0^u u\rho(u)du \tag{5.30}$$

$$u(v,x,y,z) = \sqrt{\frac{x^2}{a^2+v}+\frac{y^2}{b^2+v}+\frac{z^2}{c^2+v}} \tag{5.31}$$

We see that the potential of an ellipsoid can be derived for an arbitrary kernel function $\rho(u)$. However, the integral above can take quite complicated forms. It is a function of six variables (x, y, z, a, b, c) in the general case. Clearly in the general case one cannot evaluate this efficiently numerically for each pair-interaction in dark matter simulations. However, for a certain choice of kernels the (x, y, z) dependence can be separated from the integration and the integrals only remain as a function of (a, b, c). This is the case for kernels where $G(u)$ is a polynomial only containing even powers of u. We consider in the following the two simplest choices for which is the case: a compact homogeneous kernel and an Epanechnikov kernel $(\rho(u) \propto 1-u^2)$. Note that we do not see any possibility to evaluate the potential efficiently for popular kernel choices like the cubic spline or Wendland kernels, because they contain uneven powers of u.

5.2.3 Homogeneous Ellipsoid

We consider the homogeneous ellipsoid (also known as the uniformly charged ellipsoid) to present how a concrete potential can be obtained from (5.29), and to verify (5.29) by reproducing the literature results in this case. For a homogeneous ellipsoid we have:

$$\rho(u) = \begin{cases}\rho_0 & \text{if } u \le 1\\ 0 & \text{if } u > 1\end{cases} \tag{5.32}$$

$$\begin{aligned}G(u) &= \int_0^u u\rho(u)du\\ &= \rho_0\begin{cases}\int_0^u udu & \text{if } u\le 1\\ \int_0^1 udu & \text{if } u>1\end{cases}\\ &= \rho_0\begin{cases}\frac{u^2}{2} & \text{if } u\le 1\\ \frac{1}{2} & \text{if } u>1\end{cases}\end{aligned} \tag{5.33}$$

$$G(\infty) = \frac{1}{2} \tag{5.34}$$

Due to the piecewise definition of the kernel, we also have to separate the case $u(v) < 1$ and $u(v) > 1$ in the integration. This introduces a problematic dependence into the integration, because we are integrating over v, which is only implicitly defined by u. We recall $\lambda = v(u = 1)$ and split the integrals from (5.29):

$$\begin{aligned}\phi(\vec{x}) &= 2\pi abc \left(\int_0^\lambda \frac{G(\infty) - G(u > 1)}{\sqrt{(a^2+v)(b^2+v)(c^2+v)}} dv + \int_\lambda^\infty \frac{G(\infty) - G(u < 1)}{\sqrt{(a^2+v)(b^2+v)(c^2+v)}} dv \right) \\ &= 2\pi abc \int_\lambda^\infty \frac{G(\infty) - G(u < 1)}{\sqrt{(a^2+v)(b^2+v)(c^2+v)}} dv \qquad (5.35)\end{aligned}$$

where we dropped the first integral, as for any compact kernel $G(u > 1) = const = G(\infty)$.

$$\begin{aligned}\phi(\vec{x}) &= \rho_0 \pi abc \int_\lambda^\infty \frac{1 - u^2(v)}{\sqrt{(a^2+v)(b^2+v)(c^2+v)}} dv \\ &= \rho_0 \pi abc \left(\int_\lambda^\infty \frac{dv}{\sqrt{(a^2+v)(b^2+v)(c^2+v)}} - \int_\lambda^\infty \frac{x^2 dv}{\sqrt{(a^2+v)^3(b^2+v)(c^2+v)}} \right. \\ &\qquad \left. - \int_\lambda^\infty \frac{y^2 dv}{\sqrt{(a^2+v)(b^2+v)^3(c^2+v)}} - \int_\lambda^\infty \frac{z^2 dv}{\sqrt{(a^2+v)(b^2+v)(c^2+v)^3}} \right)\end{aligned}$$

Using the mass of a homogeneous ellipsoid $M = \frac{4\pi}{3}\rho_0 abc$ and giving names to appearing terms we find

$$\phi(\vec{x}) =: \frac{3}{4} M \left(I_0(\lambda) - x^2 I_{1a}(\lambda) - y^2 I_{1b}(\lambda) - z^2 I_{1c}(\lambda) \right) \qquad (5.36)$$

$$I_0(\lambda) = \int_\lambda^\infty \frac{dv}{\sqrt{(a^2+v)(b^2+v)(c^2+v)}} \qquad (5.37)$$

$$I_{1a_i}(\lambda) = \int_\lambda^\infty \frac{dv}{(a_i^2+v)\sqrt{(a^2+v)(b^2+v)(c^2+v)}} \qquad (5.38)$$

This result is in agreement with the literature (compare e.g. Cai (2007)). We note that there are no integrands containing an (x, y, z) dependence any more. However, we have to recall that for external points the integration boundary $\lambda(x, y, z)$ depends on the coordinates by the implicit definition (5.20) whereas for all internal points $\lambda = 0$. This means that the external potential has a rather complicated functional dependence on (x, y, z), whereas *the internal potential of a homogeneous is simply a quadratic function*:

$$\begin{aligned}\phi_{\text{int}} &= \rho_0 \pi abc \left(I_0(0) + x^2 I_{1a}(0) + y^2 I_{1b}(0) + z^2 I_{1c}(0) \right) \\ &= \frac{3}{4} M \left(I_0 + x^2 I_{1a} + y^2 I_{1b} + z^2 I_{1c} \right) \qquad (5.39)\end{aligned}$$

where we do not write the argument of the integral-functions if $\lambda = 0$. We show the potential of the homogeneous ellipsoid in Figure 5.3 top.

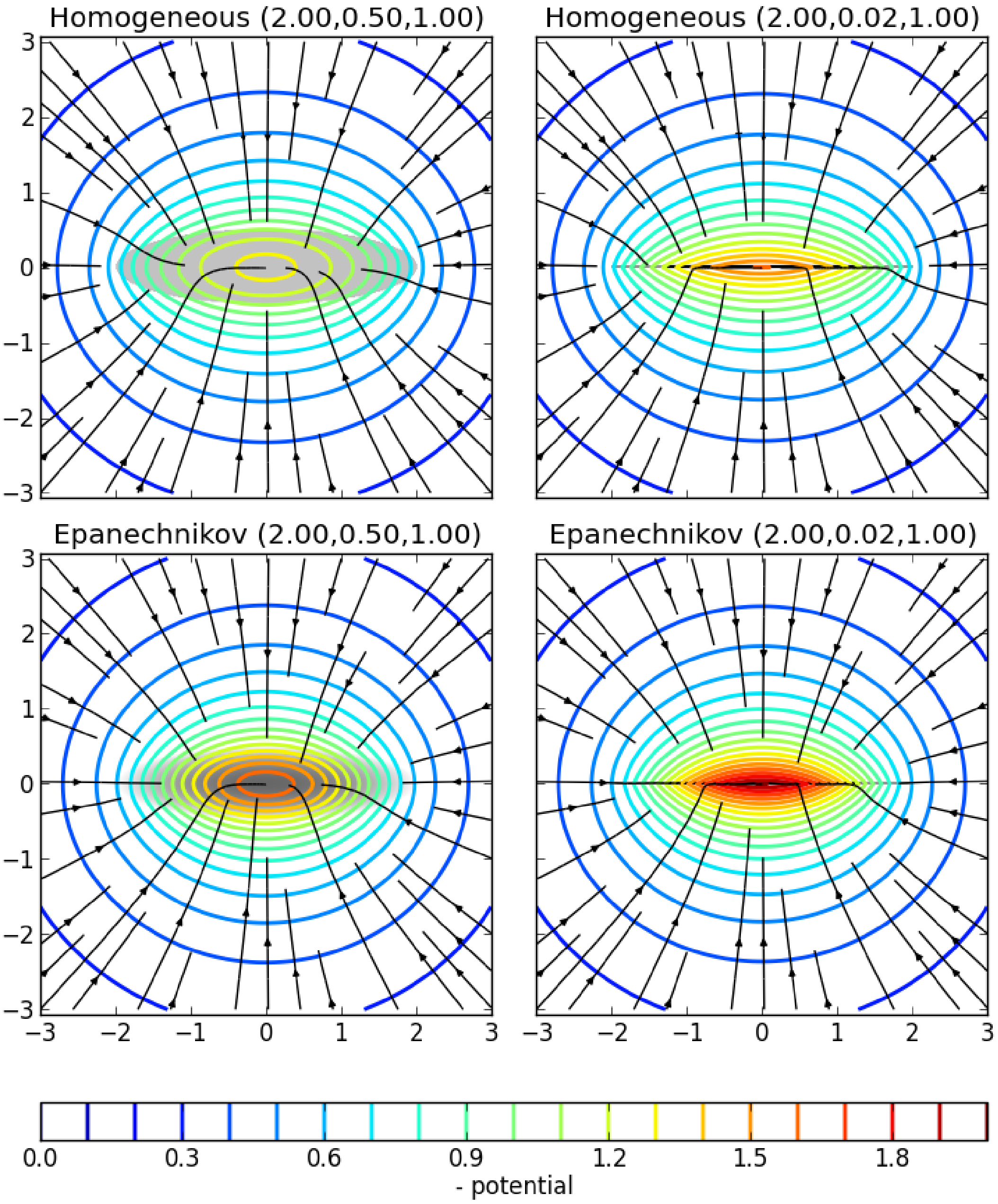


Figure 5.3: *Potential (contours) and forces (black lines with arrows) for a homogeneous ellipsoid (top) and an ellipsoid with Epanechnikov kernel (bottom) for two different settings of semi-axes (left and right).*

5.2.4 Epanechnikov Ellipsoid

For the Epanechnikov kernel we have:

$$\rho(u) = \rho_0 \begin{cases} (1-u^2) & \text{if } u \leq 1 \\ 0 & \text{if } u > 1 \end{cases} \tag{5.40}$$

$$\begin{aligned} G(u) &= \int_0^u u\rho(u)du \\ &= \rho_0 \begin{cases} \int_0^u (u-u^3)du & \text{if } u \leq 1 \\ \int_0^1 (u-u^3)du & \text{if } u > 1 \end{cases} \\ &= \rho_0 \begin{cases} \frac{u^2}{2} - \frac{u^4}{4} & \text{if } u \leq 1 \\ \frac{1}{2} & \text{if } u > 1 \end{cases} \end{aligned} \tag{5.41}$$

$$G(\infty) = \frac{1}{4} \tag{5.42}$$

$$\begin{aligned} M &= 4\pi abc \int_0^1 u^2\rho(u)du = 4\pi abc\rho_0 \int_0^1 u^2 - u^4 du = 4\pi abc\rho_0 \left(\frac{1}{3} - \frac{1}{5}\right) \\ &= \frac{8}{15}\pi abc\rho_0 \,. \end{aligned} \tag{5.43}$$

Analogue to the homogeneous ellipsoid we have to split up the integral from (5.29), but can drop the part from 0 to λ leaving us with

$$\begin{aligned} \phi &= 2\pi abc \int_\lambda^\infty \frac{G(\infty) - G(u<1)}{\sqrt{(a^2+v)(b^2+v)(c^2+v)}} dv \\ &= \frac{1}{2}\pi abc\rho_0 \int_\lambda^\infty \frac{1 - 2u^2 + u^4}{\sqrt{(a^2+v)(b^2+v)(c^2+v)}} dv \end{aligned} \tag{5.44}$$

Using

$$\begin{aligned} u^4(v) = &\frac{x^4}{(a^2+v)^2} + \frac{y^4}{(b^2+v)^2} + \frac{z^4}{(c^2+v)^2} \\ &+ \frac{2x^2y^2}{(a^2+v)(b^2+v)} + \frac{2x^2z^2}{(a^2+v)(c^2+v)} + \frac{2y^2z^2}{(b^2+v)(c^2+v)} \,. \end{aligned}$$

we can again separate the coordinates x, y, z from the integration and obtain

$$\begin{aligned} \phi(\vec{x}) = -\frac{1}{2}\pi\rho_0 abc \, [&I_0(\lambda) \\ &- 2(x^2 I_{1a}(\lambda) + y^2 I_{1a}(\lambda) + z^2 I_{1c}(\lambda)) \\ &+ \left(x^4 I_{2aa}(\lambda) + y^4 I_{2bb}(\lambda) + z^4 I_{2cc}(\lambda)\right) \\ &+2\left(x^2y^2 I_{2ab}(\lambda) + y^2z^2 I_{2bc}(\lambda) + z^2x^2 I_{2ca}(\lambda)\right)\big] \,. \end{aligned} \tag{5.45}$$

with the integrals

$$I_0(\lambda) = \int_\lambda^\infty \frac{dv}{\sqrt{(a^2+v)(b^2+v)(c^2+v)}} \tag{5.46}$$

$$I_{1a_i}(\lambda) = \int_\lambda^\infty \frac{dv}{(a_i^2+v)\sqrt{(a^2+v)(b^2+v)(c^2+v)}} \tag{5.47}$$

$$I_{2a_ia_j}(\lambda) = \int_\lambda^\infty \frac{dv}{(a_i^2+v)(a_j^2+v)\sqrt{(a^2+v)(b^2+v)(c^2+v)}} \,. \tag{5.48}$$

For the internal potential ($\lambda = 0$) we can again write due to constant coefficients:

$$\begin{aligned} \phi_{\mathrm{int}}(\vec{x}) = -\frac{15}{16} M \big[& I_0 - 2(x^2 I_{1a} + y^2 I_{1b} + z^2 I_{1c}) \\ & + x^4 I_{2aa} + y^4 I_{2bb} + z^4 I_{2cc} + 2(x^2y^2 I_{2ab} + y^2z^2 I_{2bc} + z^2x^2 I_{2ca}) \big] \end{aligned} \tag{5.49}$$

which is a polynomial function of fourth order. Note that this expression could also be written in a systematic way using tensors:

$$\phi \propto I_0 - 2 \begin{pmatrix} x^2 \\ y^2 \\ z^2 \end{pmatrix} \cdot \begin{pmatrix} I_{1a} \\ I_{1b} \\ I_{1c} \end{pmatrix} + \begin{pmatrix} x^2 & y^2 & z^2 \end{pmatrix} \begin{pmatrix} I_{2aa} & I_{2ab} & I_{2ac} \\ I_{2ab} & I_{2bb} & I_{2bc} \\ I_{2ac}^2 & I_{2bc} & I_{2cc} \end{pmatrix} \begin{pmatrix} x^2 \\ y^2 \\ z^2 \end{pmatrix} \tag{5.50}$$

We can easily foresee how higher order kernels would increase the number of terms present. For example a 4th order kernel (with only even powers of u) would additionally introduce a symmetric $3\times3\times3$ Tensor leading to additional 10 terms, thereby doubling the computational effort needed to compute the potential.

In Figure 5.3 (bottom) we show the potential of the Epanechnikov kernel in comparison to the homogeneous ellipsoid. We note that the Epanechnikov kernel exhibits a deeper potential in the central regions, and the potential seems to be more spherical at similar radii than for the homogeneous case. Comparing the potentials with the ellipsoidal shell in Figure 5.2 there is the remarkable difference that the potential is much more spherical close to the boundary for the extended ellipsoids than for the ellipsoidal shell. This can be understood as these exhibit also contributions from smaller shells which have already a more spherical potential at those distances. Note that this is a profound difference to the spherical case where only the internal potential depends on the kernel. For an ellipsoid the kernel also has an influence onto the external potential.

From here on we only consider the Epanechnikov ellipsoid, because a kernel density estimate using it can represent an arbitrary density field much preciser than the homogeneous ellipsoid. For a sum of different homogeneous ellipsoids the density field would be unsteadily, whereas for Epanechnikov ellipsoids it is continuous.

5.3 Forces and Tidal Tensor

To derive the potential (5.45) for a spatial coordinate x_i we also have to consider the implicit dependence of λ on x_i

$$\frac{\partial\phi}{\partial x_i} = \left.\frac{\partial\phi}{\partial x_i}\right|_{\lambda\ \mathrm{const}} + \frac{d\lambda}{dx}\left.\frac{\partial\phi}{\partial\lambda}\right|_{x_i\ \mathrm{const}} . \tag{5.51}$$

From (5.44) we know

$$\left.\frac{\partial\phi}{\partial\lambda}\right|_{x_i\ \mathrm{const}} = -\frac{15}{16}M\frac{1-2u^2+u^4}{\sqrt{(a^2+\lambda)(b^2+\lambda)(c^2+\lambda)}} \tag{5.52}$$

$$u^2 = \frac{x^2}{a^2+\lambda} + \frac{y^2}{b^2+\lambda} + \frac{z^2}{c^2+\lambda} . \tag{5.53}$$

Per definition of λ it is $u(\lambda) = 1$ and therefore

$$\left.\frac{\partial\phi}{\partial\lambda}\right|_{x_i\ \mathrm{const}} = 0 \tag{5.54}$$

Thus we are left with the surprisingly simple result

$$\frac{\partial\phi}{\partial x_i} = \left.\frac{\partial\phi}{\partial x_i}\right|_{\lambda\ \mathrm{const}} \tag{5.55}$$

which tells us that we do not have to derive the coefficients to obtain forces and tidal forces. Note that this does not mean that the individual coefficients are spatially constant, but only that this special combination of their derivatives is zero. We derive (5.45) and find the forces

$$a_x = -\frac{\partial\phi}{\partial x} = -x\frac{15}{4}M(I_{1a}(\lambda) - x^2 I_{2aa}(\lambda) - y^2 I_{2ab}(\lambda) - z^2 I_{2ac}(\lambda)) \tag{5.56}$$

$$a_y = -\frac{\partial\phi}{\partial x} = -y\frac{15}{4}M(I_{1b}(\lambda) - x^2 I_{2ab}(\lambda) - y^2 I_{2bb}(\lambda) - z^2 I_{2bc}(\lambda)) \tag{5.57}$$

$$a_z = -\frac{\partial\phi}{\partial x} = -z\frac{15}{4}M(I_{1c}(\lambda) - x^2 I_{2ac}(\lambda) - y^2 I_{2bc}(\lambda) - z^2 I_{2cc}(\lambda)) \tag{5.58}$$

and the tidal tensor

$$T_{xx} = -\frac{\partial^2\phi}{\partial x^2} = -\frac{15}{4}M(I_{1a}(\lambda) - 3x^2 I_{2aa}(\lambda) - y^2 I_{2ab}(\lambda) - z^2 I_{2ac}(\lambda)) \tag{5.59}$$

$$T_{yy} = -\frac{\partial^2\phi}{\partial z^2} = -\frac{15}{4}M(I_{1b}(\lambda) - x^2 I_{2ab}(\lambda) - 3y^2 I_{2bb}(\lambda) - z^2 I_{2bc}(\lambda)) \tag{5.60}$$

$$T_{zz} = -\frac{\partial^2\phi}{\partial z^2} = -\frac{15}{4}M(I_{1c}(\lambda) - x^2 I_{2ac}(\lambda) - y^2 I_{2bc}(\lambda) - 3z^2 I_{2cc}(\lambda)) \tag{5.61}$$

$$T_{xy} = T_{yx} = -\frac{\partial^2\phi}{\partial x\partial y} = xy\frac{15}{2}MI_{2ab}(\lambda) \tag{5.62}$$

$$T_{xz} = T_{zx} = -\frac{\partial^2\phi}{\partial x\partial z} = xz\frac{15}{2}MI_{2ac}(\lambda) \tag{5.63}$$

$$T_{yz} = T_{zy} = -\frac{\partial^2\phi}{\partial y\partial z} = yz\frac{15}{2}MI_{2bc}(\lambda) \tag{5.64}$$

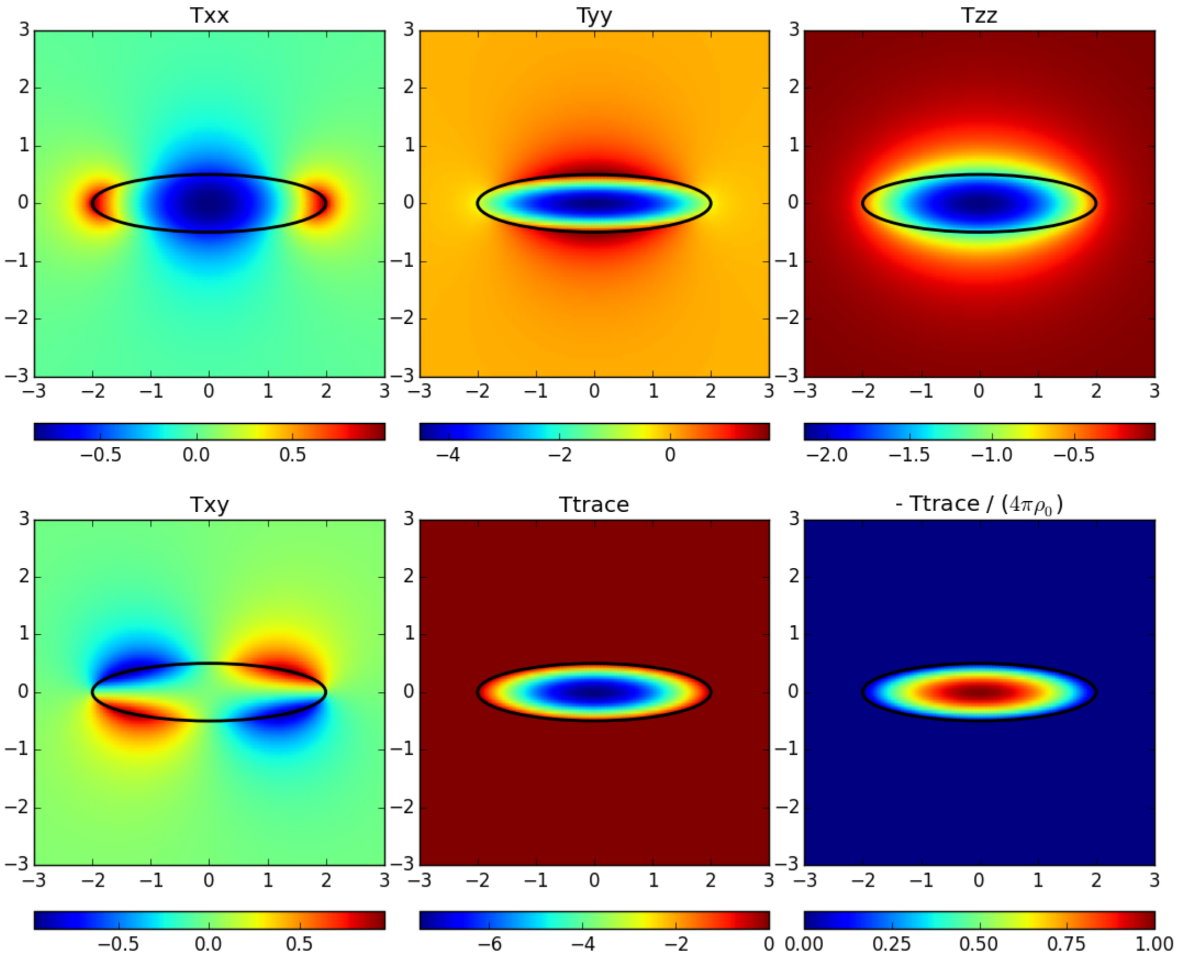


Figure 5.4: *Non-zero tidal tensor components for an Epanechnikov ellipsoid with semi-axes (2, 0.5, 1) in the* x, y *plane. The color represents the value for* $M = G = 1$ *and dimensionless coordinates. The bottom right panel shows the rescaled trace of the tidal tensor which correctly corresponds to the density profile* ρ/ρ_0.

To ensure that we did no errors when we obtained this surprisingly simple derivatives, we checked all results by comparing them for 1000 randomly drawn test cases (a, b, c, x, y, z), with what would be obtained by deriving the potential (5.45) by finite differences. They matched indeed - these simple results seem to be correct.

We display the tidal tensor of an Epanechnikov ellipsoid in Figure 5.4. The components T_{xz}, T_{yz} are zero in the x, y plane, so we do not show them. We also show the trace of the tidal tensor

$$\begin{aligned} T_{\text{trace}} &= T_{xx} + T_{yy} + T_{zz} && (5.65) \\ &= -\left(\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} + \frac{\partial^2\phi}{\partial z^2}\right) && (5.66) \\ &= -\Delta\phi && (5.67) \end{aligned}$$

We can use Poisson's equation

$$\Delta\phi = 4\pi\rho \tag{5.68}$$

to check the consistency of our results. Therefore we also display $-T_{\text{trace}}/(4\pi\rho_0)$, which should match the density in units of the central density. This is indeed the case.

5.4 Numerical Evaluation of the Potential

To evaluate the external potential of the Epanechnikov-ellipsoid one needs ten integral coefficients $I_*(a, b, c, \lambda)$. As we will show in section 5.4.2, we can always separate the largest axis a from the integration so that the coefficients can be expressed in terms of integrals $J_*(\alpha, \beta, \lambda)$ which only depend on λ and the axis ratios $\alpha = b/a$ and $\beta = c/a$. To evaluate the internal potential one only needs the coefficients without coordinate dependence $J_*(\alpha, \beta, \lambda = 0)$.

For a given (α, β, λ) the coefficients can be evaluated numerically by integration. However, as such an integration is costly, this can not be done for each individual pair interaction. We tested two different strategies to make the numerical evaluation of the potential efficient enough that it can be used in N-Body simulations:

1. The coefficients of the internal potential have no coordinate dependence $\lambda = 0$. Therefore they can be evaluated and stored once per particle, every time its distortion tensor changes. Using the knowledge about the internal potential and its derivatives at the boundary of the shell, an approximation to the external potential can be found. This way the coefficients only have to be evaluated once per particle, and not once per interaction.

2. The J_* coefficients can be stored at a discrete set of $(\alpha_i, \beta_j, \lambda_k)$ and interpolated each time they are required. If the interpolation is accurate and fast enough, the coefficients can be obtained for each pair interaction individually.

Initially we used approach (1), as we did not expect that the 3d interpolation in (2) could be made fast and accurate enough. However, after already having a working implementation of (1) we found a way to make (2) faster and also more accurate than (1). The major performance problem of approach (1) is that we could not find a simple explicit expression for the derivatives of the approximated external potential, and thus we had to use finite differences requiring 12 potential evaluations for each interaction. Therefore our newer implementation of (2) outperforms (1) in accuracy and performance. The final version of our code only contains (2), but we still want to present (1) here, as a lot of work flowed into this, and also we can learn something about the potential we are trying to evaluate.

5.4.1 Approach 1: Interpolation of the external potential

Given the exact solution to the internal potential ϕ_{int}, we want to find an approximation to the external potential ϕ_{ext}. Therefore we interpolate the external potential on radial lines from the exact solution at the boundary of the ellipsoid out to a distant radius R, at

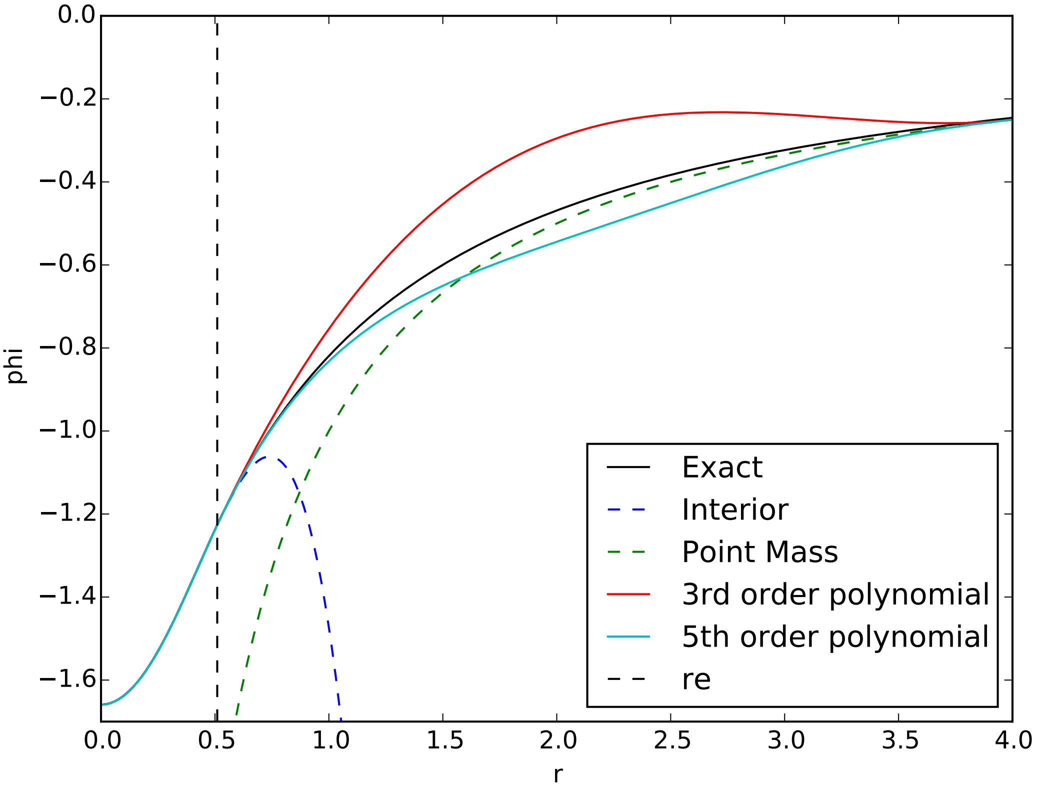


Figure 5.5: *Exact potential and polynomial interpolation along a radial line in the "bad case direction"* $(0.189, 0.978, 0.084)$. *The polynomial interpolation leads not to satisfactory results in this case.*

which we assume the potential to be the spherical symmetric potential of a point mass. The most important advantage of this approach is, that the integral coefficients have only to be evaluated once per particle (a, b, c), and do not depent on $\lambda(x, y, z)$. Further it is a nice property that it introduces a smooth transition to the point mass potential so that we can use the tree and the particle mesh for force-calculations beyond this radius, without introducing any inconsistencies into the potential.

However, as a big drawback it shall be mentioned that with this approach the external potential will not be an exact vacuum solution $\Delta\phi = 0$ any more (only the exact solution is). Furthermore when the axis ratios get large, this radial interpolation method can deviate at some points strongly from the exact potential. Performance-wise it is a problem that it is far too complicated to obtain explicit expressions for the derivatives of the potential. Therefore forces and the tidal tensor have to be evaluated by finite differences which is very inefficient.

We want to radially interpolate the external potential to match boundary conditions given by the internal potential at $r_e(\theta, \phi)$ and the point mass potential at a radius R. r_e is the intersection of the outer shell of the ellipsoid and a radial line through the point (x, y, z) that is to be evaluated. If we want to match the the 0th and 1st derivatives at these boundaries, the problem can be stated as finding a smooth function $\phi_{\mathrm{ext}}(R, \phi, \theta)$ with the conditions

$$\phi_{\mathrm{ext}}(r_e, \phi, \theta) = \phi_{\mathrm{int}}(r_e, \phi, \theta) =: \phi_i(\theta, \phi) \tag{5.69}$$

$$\partial_r \phi_{\mathrm{ext}}(r_e, \phi, \theta) = \partial_r \phi_{\mathrm{int}}(r_e, \phi, \theta) =: \phi_i'(\theta, \phi) \tag{5.70}$$

$$\phi_{\mathrm{ext}}(R, \phi, \theta) = -\frac{M}{R} \tag{5.71}$$

$$\partial_r \phi_{\mathrm{ext}}(R, \phi, \theta) = \frac{M}{R^2} \tag{5.72}$$

$$r_e = r \cdot \left(\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 + \left(\frac{z}{c}\right)^2 \right)^{-1/2} . \tag{5.73}$$

We usually choose $R = 2 \cdot a_{\mathrm{max}}$ where a_{max} is the largest half-axis of the ellipsoid. Note that one could consider to also match the second derivatives of the potentials. However, it would make no sense to use higher orders than the second order, as e.g. the third would have the same order like the first derivatives of the density field, which are not continuous for an Epanechnikov kernel (i.e. the transition between internal and external potential is only smooth up to the second order).

Our first Ansatz for the interpolation was to use a 3rd order polynomial $P_3(r)$ which matches up to the first derivatives at the two boundaries. To qualitatively see, whether this approach leads to reasonable results, we used an ellipsoid with semi-axes $(2.0, 0.5, 1.0)$ and plotted the interpolated potential along radial lines with a random direction in comparison to the correct potential. In Figure 5.5 we show one direction $(0.189, 0.978, 0.084)$ where this leads to particularly bad behaviour. It can be seen that for this interpolation method it even cannot be assured that the potential is a monothonical increasing function of r along radial lines. We also tried a 5th order interpolation matching the boundaries up to the 2nd derivatives, also shown in Figure 5.5. This gives only little improvement, and may still produce very unusual behaviour in the potential. Therefore we discard a polynomial interpolation of the potential at this point.

Inspecting Figure 5.5 we notice, that the exact potential looks similar to the point mass potential, but with a radial varying shift along the r coordinate. This leads us to the Ansatz

$$\phi_{ext} = -\frac{MG}{r + g(r; \theta, \phi)} \tag{5.74}$$

where we allow $g(r)$ to be a third order polynomial so that it can match all four boundary conditions. It is easy to see that a function of the form

$$\phi_e = -\frac{MG}{r + A(\theta, \phi)(R - r)^2 + B(\theta, \phi)(R - r)^3} \tag{5.75}$$

will already match the outer boundary conditions so that we only have to to match the inner

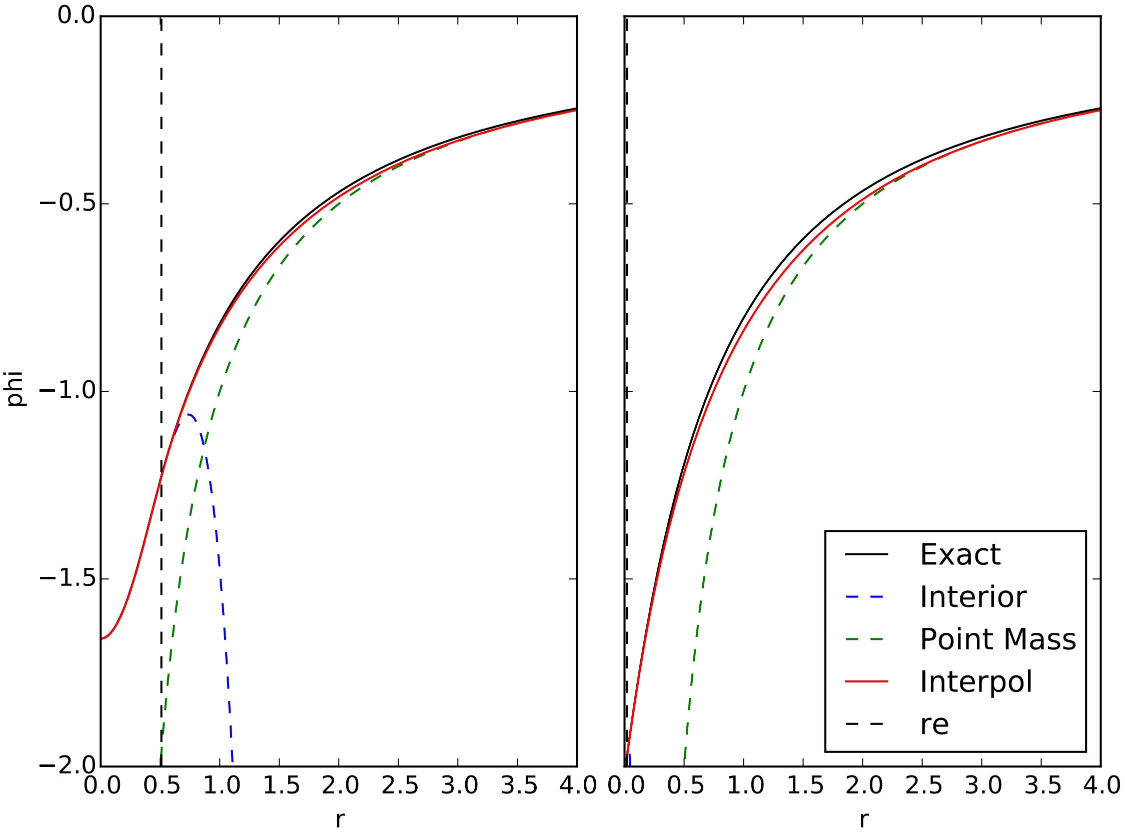


Figure 5.6: *Left: Like Figure 5.5 (direction (0.189 0.978 0.084)), but using the new interpolation technique. Right: Using the same radial line, but for a strongly distorted ellipsoid with axes (2.0, 0.02, 1.0).*

conditions with the coefficients A and B. By evaluating the boundary conditions we find

$$A = (R - r_e)^{-1} \left(\frac{\phi_i'}{\phi_i^2} - 1 \right) - 3(R - r_e)^{-2} \left(\frac{1}{\phi_i} + r_e \right) \tag{5.76}$$

$$B = -(R - r_e)^{-2} \left(\frac{\phi_i'}{\phi_i^2} - 1 \right) + 2(R - r_e)^{-3} \left(\frac{1}{\phi_i} + r_e \right) . \tag{5.77}$$

We can rewrite this as

$$\phi_{ext} = -MG \left[r + \left(\frac{\phi_i'}{\phi_i^2} - 1 \right) (R - r_e) u^2 (1 - u) + \left(\frac{1}{\phi_i} + r_e \right) u^2 (-3 + 2u) \right]^{-1} \tag{5.78}$$

$$u := \frac{R - r}{R - r_e} .$$

Where ϕ_i, ϕ_i' and r_e are all functions of the angular coordinates (ϕ, θ). The radial derivative

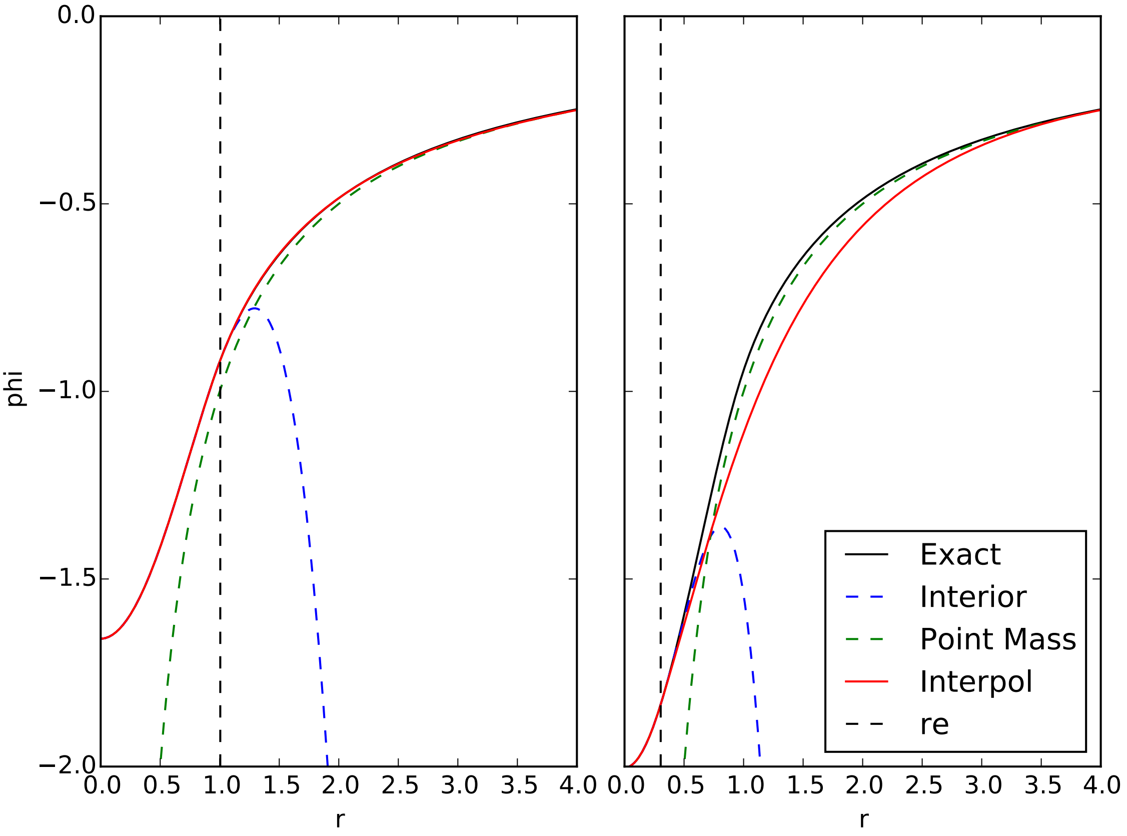


Figure 5.7: *Like Figure 5.6, but using a bad-case direction for the strongly distorted ellipsoid (right).*

of the interior potential can be simply obtained by deriving (5.49):

$$\phi_i' = \frac{\partial \phi_{\text{int}}}{\partial r} = \frac{15}{4} M \left[x^2 I_{1a} + y^2 I_{1a} + z^2 I_{1c} + x^4 I_{2aa} + y^4 I_{2bb} + z^4 I_{2cc} \right. \\ \left. +2(x^2 y^2 I_{2ab} + y^2 z^2 I_{2bc} + z^2 x^2 I_{2ca}) \right] . \tag{5.79}$$

We show in the left panel of Figure 5.6 how this performs in the same situation like 5.5. We try again different random directions, and find that this interpolation seems to perform similar well for this ellipsoid (2.0, 0.5, 1.0) in any direction. We also test this for the case of a more extremely distorted ellipsoid (2.0, 0.02, 1.0). In Figure 5.6 right we plot it for the same direction like before (0.189 0.978 0.084), where it seems to perform well too. However, trying different random directions again in this extremer case we find as a bad-case-scenario the direction (0.16, 0.06, 0.98) which we display in Figure 5.7. This case could possibly be improved by also matching the second derivatives at the boundaries. However, at this point we stick with the first derivatives to not further complicate the mathematical structure of the potential evaluation. It is worth mentioning, that it is probably not so important that the potential matches quantitatively exactly, but more that the potential and its derivatives resemble the correct principle behaviour. However, a problem might be that by not matching

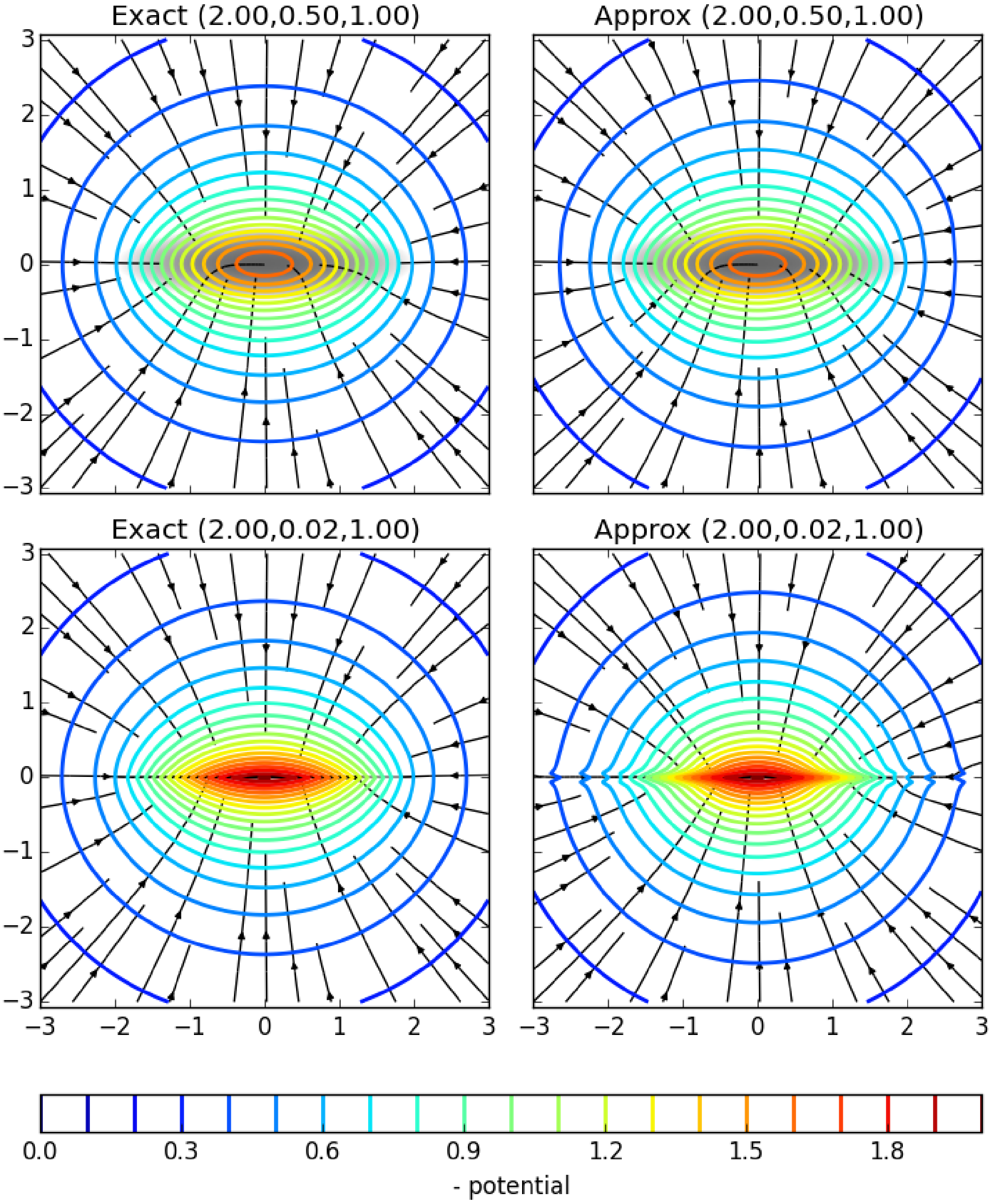


Figure 5.8: *Left exact potential and right our interpolation approximation. Top: typical case for the semi-axes (2.00, 0.50, 1.00) and bottom for a strongly distorted (2.0, 0.02, 1.0) ellipsoid. Generally, the interpolation approximation seems to perform well. However, close to the largest axis deviations occur.*

the second derivatives the tidal tensor can get unsteady.

To get a 2d impression we plot in Figure 5.8 the exact potential versus our approximation for the slightly distorted ellipsoid (2.0, 0.5, 1.0) and a strongly distorted ellipsoid (2.0, 0.02, 1.0). We note that the approximation works very well at any position for the slightly distorted ellipsoid. For the strongly distorted ellipsoid the approximation seems to perform well at most positions, but deviations appear close to the major axis.

Originally we implemented this method into Gadget. Therefore we evaluated the integral coefficients by bi-cubic interpolation of an initially calculated 2 dimensional $\log J_*(\log\alpha_i, \log\beta_i, \lambda = 0)$ grid up to precisions of 10^{-8}. This interpolation was done once per particle each time its distortion tensor changed. We obtained forces and the tidal tensor by finite differences which required 12 potential evaluations in total. However, as we do not use this method any more, we will not describe this in more detail here.

5.4.2 Approach 2: Direct 3d Interpolation of the Coefficients

We want to evaluate the Integral coefficients by a 3d interpolation in (α, β, λ) for every pair interaction. Therefore we describe in the following sections, (1) how to obtain lambda for given (a, b, c, x, y, z), (2) how we cut the direct force summation, and (3) how to set up interpolation grids that can be interpolated at high enough performance and accuracy, and how to obtain the implicitly defined $\lambda(x, y, z)$ fast and accurately.

Lambda Evaluation

The λ equation (5.20) can be rewritten into the numerically more convenient form of a cubic equation

$$0 = \lambda^3 + p\lambda^2 + q\lambda + s =: f(\lambda) \tag{5.80}$$

$$p = a^2 + b^2 + c^2 - x^2 - y^2 - z^2 \tag{5.81}$$

$$q = a^2b^2 + a^2c^2 + b^2c^2 - (b^2 + c^2)x^2 - (a^2 + c^2)y^2 - (a^2 + b^2)z^2 \tag{5.82}$$

$$s = a^2b^2c^2 - b^2c^2x^2 - a^2c^2y^2 - a^2b^2z^2 \tag{5.83}$$

The solutions to a cubic equation are known analytically. However, they require the evaluation of several expensive functions. Therefore we choose the cheaper way of evaluating the λ equation numerically. Therefore we use the implementation of the Newton-method

$$\lambda_{n+1} = \lambda_n - \frac{f(\lambda_n)}{f'(\lambda_n)} \tag{5.84}$$

from the GSL-library *gsl_root_fdfsolver_newton* to evaluate λ numerically. We are only interested in the largest root of the cubic equation. Thus we choose as a starting value

$$\lambda_0 = x^2 + y^2 + z^2 \tag{5.85}$$

which is always larger than the largest root $\lambda \leq r^2$. The Newton method usually converges very quickly, and we stop the iteration if $|\lambda_{n+1} - \lambda_n| < 10^{-8}$.

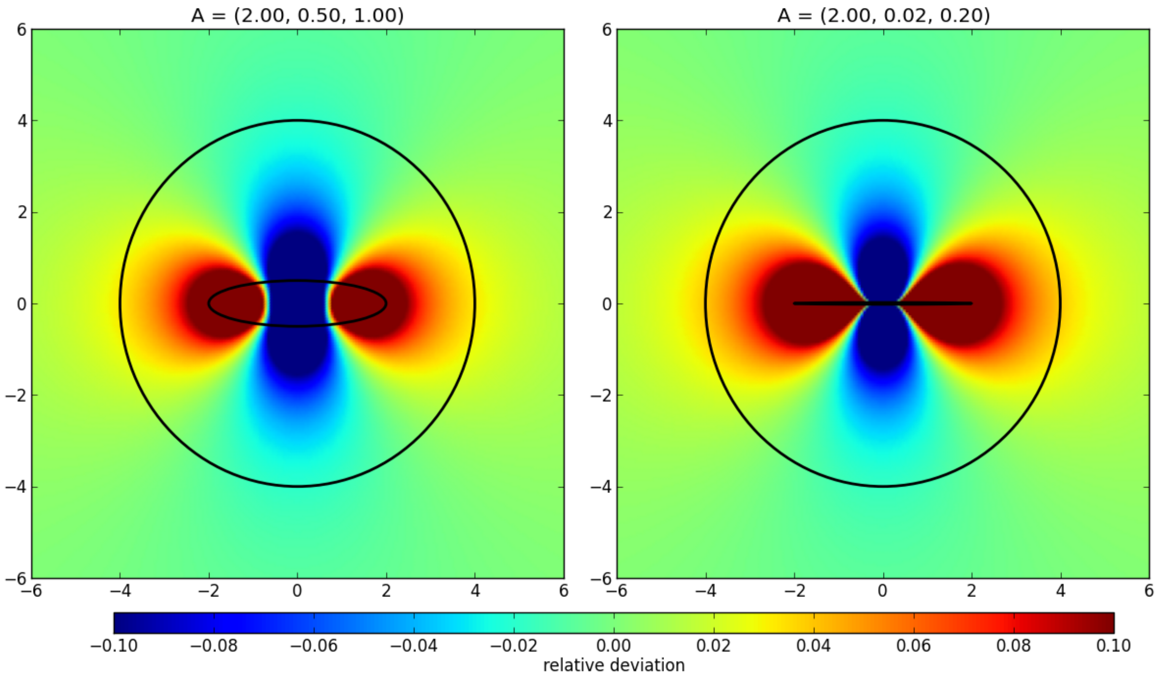


Figure 5.9: *Relative deviation from a point mass potential* $\epsilon = (\phi - \phi_{pm})/\phi_{pm}$ *for a typical ellipsoid (2, 0.5, 1.0) and a distorted ellipsoid (2, 0.02, 0.2) in the* x, y*-plane. Points with* $|\epsilon| > 0.1$ *received the color of the next boundaries* $\epsilon = \pm 0.1$*. A red color refers to regions where the ellipsoidal potential is deeper than the point mass potential, blue color to regions where it is flatter. Also drawn are the shells of the ellipsoids and the radii at which the potentials are cut to point mass potentials for* $q_{cut} = 2$*. The relative errors at the cuts are between* 10^{-2} *and* 10^{-1}*. They seem not to depend strongly on the distortion of the ellipsoids.*

Potential cut

To be able to use the tree summation for long range forces, we assume the potential to be the point mass potential beyond a radius r_{cut}

$$\phi(\vec{x}) = \begin{cases} \phi_{el}(\vec{x}) & \text{if } r \leq r_{\mathrm{cut}} \\ -\frac{M}{r} & \text{if } r > r_{\mathrm{cut}} \end{cases} \tag{5.86}$$

We choose r_{cut} in units of the largest semi-axis a of the potential-sourcing ellipsoid

$$r_{\mathrm{cut}} = q_{\mathrm{cut}} \cdot a \tag{5.87}$$

where q_{cut} is a parameter that can be defined via the pre-compiler directive $q_{\mathrm{cut}} = \mathrm{AS\_RCUT}$. We usually use $q_{\mathrm{cut}} = 2$. In Figure 5.9 we plot the relative deviation of the ellipsoidal potential from a point mass potential $\epsilon = |\phi - \phi_{pm}|/\phi_{pm}$.

Integrals

We want to make the evaluation of the integrals

$$
\begin{aligned}
I_0 &= \int_\lambda^\infty \frac{dv}{\sqrt{(a^2+v)(b^2+v)(c^2+v)}} \\
I_{1\rho} &= \int_\lambda^\infty \frac{dv}{(\rho^2+v)\sqrt{(a^2+v)(b^2+v)(c^2+v)}} \\
I_{2\rho\sigma} &= \int_\lambda^\infty \frac{dv}{(\rho^2+v)(\sigma^2+v)\sqrt{(a^2+v)(b^2+v)(c^2+v)}}
\end{aligned}
$$

numerical efficient. Therefore we want to integrate them in the beginning of the simulation, tabulate them, and interpolate these tables when needed. We notice that for all integrals one variable can be separated from the integration, so that we only need to store tables of three parameters: The I_0 integral can be rewritten as

$$
\begin{aligned}
I_0 &= \int_\lambda^\infty \frac{dv}{\sqrt{(a^2+v)(b^2+v)(c^2+v)}} \\
&= \int_\lambda^\infty \frac{a^{-3}dv}{\sqrt{(1+v/a^2)(b^2/a^2+v/a^2)(c^2/a^2+v/a^2)}}
\end{aligned}
$$

We substitute

$$
\begin{aligned}
u &= v/a^2 \\
du &= a^{-2}dv \\
I_0 &= \int_{\lambda/a^2}^\infty \frac{a^{-1}du}{\sqrt{(1+u)(b^2/a^2+u)(c^2/a^2+u)}} \\
&= a^{-1}\int_{\lambda/a^2}^\infty \frac{du}{\sqrt{(1+u)(\beta^2+u)(\gamma^2+u)}} \\
&=: a^{-1}J_0(\alpha,\beta,\lambda_r)
\end{aligned}
$$

where we defined the axis ratios

$$\beta = b/a \leq 1 \tag{5.88}$$

$$\gamma = c/a \leq 1, \tag{5.89}$$

the reduced λ_r parameter

$$\lambda_r = \frac{\lambda}{a^2}, \tag{5.90}$$

and the J_0 integral. Note that from $\lambda \leq r^2 \leq r_{\text{cut}}^2$ it follows $\lambda_r \leq q_{\text{cut}}^2$ which limits the range of λ_r values that have to be tabulated. For the other integrals one can substitute analogously, and further use the symmetry in α and β to obtain a catalogue of seven integrals that actually have to be tabulated:

$$aI_0 =: J_o(\alpha, \beta, \lambda_r) = \int_{\lambda_r}^{\infty} \frac{du}{\sqrt{(1+u)(\beta^2+u)(\gamma^2+u)}} \tag{5.91}$$

$$a^3 I_{1a} =: J_{1a}(\alpha, \beta, \lambda_r) = \int_{\lambda_r}^{\infty} \frac{dv}{\sqrt{(1+u)^3(\alpha+u)(\beta+u)}} \tag{5.92}$$

$$a^3 I_{1b} =: J_{1b}(\alpha, \beta, \lambda_r) = \int_{\lambda_r}^{\infty} \frac{dv}{\sqrt{(1+u)(\alpha+u)^3(\beta+u)}} \tag{5.93}$$

$$a^5 I_{2aa} =: J_{2aa}(\alpha, \beta, \lambda) = \int_{\lambda_r}^{\infty} \frac{dv}{\sqrt{(1+u)^5(\alpha+u)(\beta+u)}} \tag{5.94}$$

$$a^5 I_{2bb} =: J_{2bb}(\alpha, \beta, \lambda_r) = \int_{\lambda_r}^{\infty} \frac{dv}{\sqrt{(1+u)(\alpha+u)^5(\beta+u)}} \tag{5.95}$$

$$a^5 I_{2ab} =: J_{2ab}(\alpha, \beta, \lambda_r) = \int_{\lambda_r}^{\infty} \frac{dv}{\sqrt{(1+u)^3(\alpha+u)^3(\beta+u)}} \tag{5.96}$$

$$a^5 I_{2bc} =: J_{2bc}(\alpha, \beta, \lambda_r) = \int_{\lambda_r}^{\infty} \frac{dv}{\sqrt{(1+u)(\alpha+u)^3(\beta+u)^3}} \tag{5.97}$$

The remaining three integrals can be computed by using the symmetry in α and β, without an extra interpolation table:

$$\begin{aligned} J_{1c}(\alpha, \beta, \lambda_r) &= J_{1b}(\beta, \alpha, \lambda_r) \\ J_{2ac}(\alpha, \beta, \lambda_r) &= J_{2ab}(\beta, \alpha, \lambda_r) \\ J_{2cc}(\alpha, \beta, \lambda_r) &= J_{2bb}(\beta, \alpha, \lambda_r) \end{aligned}$$

However, to avoid additional bookkeeping, we also compute and store those three integrals along with the others in our implementation. Having defined this, we now have to find ways to calculate, store, and interpolate the seven J-integrals from Equations (5.91) - (5.97). In Figure 5.10 we once plot these integrals for $\lambda = 0$ to get an impression of how strongly they vary with α and β. We find that these integrals can be strongly varying functions of α and β and therefore an equally spaced tabulation of J_* in α, β would be to imprecise in most cases. We therefore tabulate the integrals equally spaced in $\log \alpha$ and $\log \beta$. It might be tempting to tabulate the integrals also in logarithmic form $\log J_*$. However, this would additionally require ten calls to an exponential function per interaction which is not desirable performance-wise. The parameter space of α and β is limited by the minimal axis ratio q_{min} =AS_MINRESCALE/AS_MAXRESCALE. For our simulations this is typically $q_{\text{min}} \sim 10^{-2}$. For the α, β dependence we then tabulate the integrals in the intervals

$$\begin{aligned} \log(\alpha) &\in [\log(q_{\text{min}}), 0] \\ \log(\beta) &\in [\log(q_{\text{min}}), 0] \end{aligned}$$

For the interpolation in the λ_r direction we can not use an equal spacing in logarithmic space, as λ can be zero. As can be seen in Figure 5.11 for the J_{2bc} integral, the integrals can be varying very strongly at small values of λ. Further we note, that the slope of the integrals can get as steep as $\lambda^{-2.5}$ (as the integrands have slopes up to $\lambda^{-3.5}$ for $\lambda \gg 1 \geq \alpha \geq \beta$). To overcome

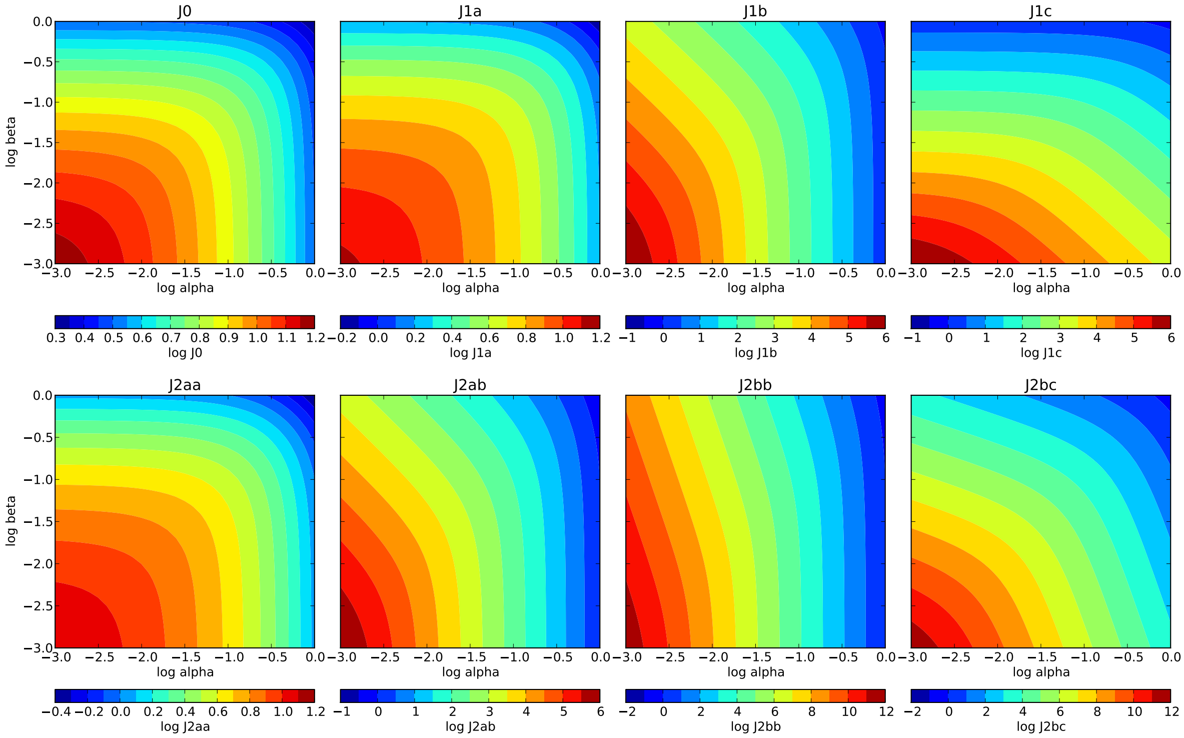


Figure 5.10: *All J-Integrals for* $\lambda = 0$ *in the form* $\log J(\log\alpha, \log\beta)$ *with* $\log = \log_{10}$. *Colour-bars are logarithmic and scaled individually for each image. We do not show* J_{cc} *and* J_{ac}*, as they are mirrored versions of* J_{bb} *and* J_{ab}*, just like it is the case for* J_{1c} *and* J_{1b}.

this two problems, (1) we choose the grid points for λ distributed as $\lambda_i = \lambda_{\max} \cdot f_i^4$, where f_i is equally spaced in $[0, 1]$, (2) divide each integral by its integrand $F_{\text{tab},*} = J_*/i_*$, which limits the slope to something of the order 1. In Figure 5.11 we show how this interpolation performs for 64 supporting points in λ. This is already a case where it was particularly hard to achieve acceptable results for the precision of the interpolation. We summarize our method to obtain the J-Integrals:

- All J-Integrals are evaluated numerically at $64 \times 64 \times 64$ grid, with α and β equally distributed in log-space and λ_i distributed like f_i^4. For the integration we use the routine *gsl_integration_qag* from the GSL library. The integrals are integrated to a relative precision of 10^{-10} and then stored as J_*/i_*. The grids require in total 20MB of working space.
- When needed, the integrands are evaluated by a trilinear interpolation.

To show that the accuracy of the interpolation is sufficient, we test the interpolators for 10^6 randomly drawn values, equally distributed in $\log\alpha$, $\log\beta$ and λ. We compare their results with the direct integration result from *gsl_integration_qag* (relative accuracy 10^{-10}). In Table 5.1 we display the mean and maximum relative errors we obtain this way. The maximum relative errors are of order 10^{-2} and the mean relative errors are of the order 10^{-4}.

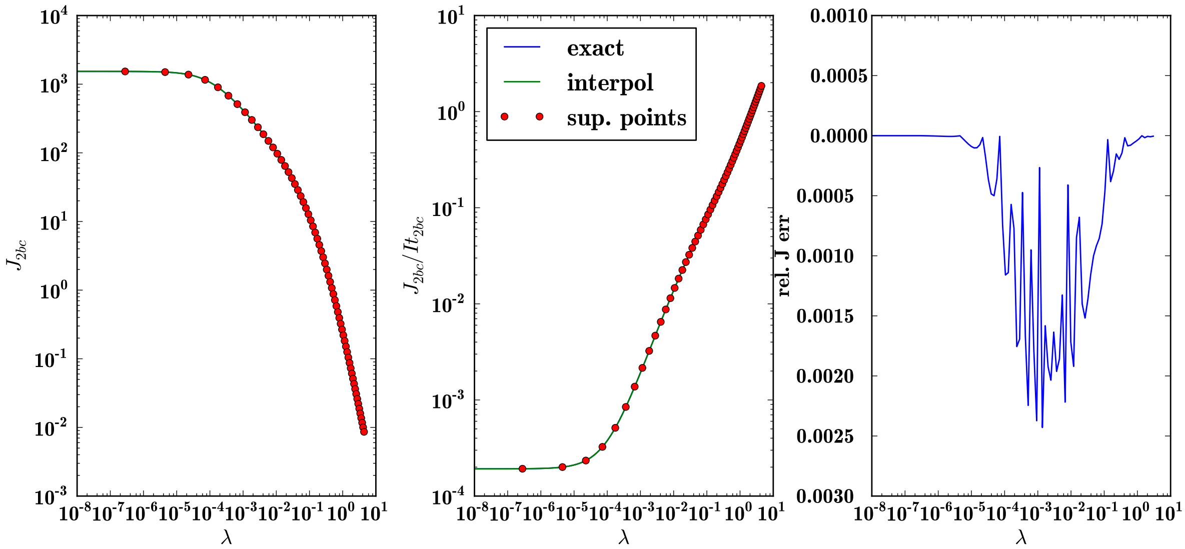


Figure 5.11: *Left: Exact and interpolated* J_{2bc} *integral. Centre:* J_{2bc}/i_{2bc} *where we mean by* i_{2bc} *the integrand of the* J_{2bc} *integral. Right: relative error of the interpolation. As red points we show the supporting points of the interpolation. Not seen in this plot is the supporting point at* $\lambda = 0$.

Table 5.1: *Maximal and mean relative errors for the* J*-integral interpolation for* 10^6 *randomly drawn scenarios.*

	J_0	J_{1a}	J_{1b}	J_{1c}	J_{2aa}	J_{2ab}	J_{2ac}	J_{2bb}	J_{2bc}	J_{2cc}
$\epsilon_{\max}/10^{-2}$	0.961	0.982	0.825	0.994	0.993	0.872	1.004	0.822	0.932	1.014
$\epsilon_{\text{mean}}/10^{-4}$	1.203	0.951	1.414	1.414	0.875	1.153	1.153	1.586	1.298	1.585

One might wonder at this point, whether the evaluation of the integrals up to this accuracy also guaranties a corresponding accuracy in the potential evaluation. In principle, the relative deviations in the potential could be much larger, as it consists of sums of these terms which could erase each other. Thus we also test the potential evaluation for 10^6 randomly selected settings $(a = 1, b, c, \vec{x})$ where $b < 1$ and $c < 1$ are uniformly distributed in logarithmic space and $\vec{x}$ is distributed uniformly in a sphere with radius q_{cut}. We calculate the potential (5.45), the forces (5.59) - (5.64) and the tidal tensors (5.59)-(5.64) once for interpolated coefficients (no subscript) and once for directly integrated coefficients (subscript e). As relative precisions we define

$$\epsilon_\phi := \frac{|\phi - \phi_e|}{|\phi_e|} \tag{5.98}$$

$$\epsilon_{a_i} := \frac{|a_i - a_{e,i}|}{|\vec{a}_e|} \tag{5.99}$$

$$\epsilon_{T_{ij}} := \frac{|T_{ij} - T_{e,ij}|}{||T||_\infty} \tag{5.100}$$

where we mean by $||T||_\infty$ the infinity norm of the matrix T, i.e. the absolute value of its largest element. We define the errors in the forces and the tidal tensor in this way, to not overestimate errors that only lead to small rotational deviations. We display the maximum and the relative errors obtained this way in Table 5.2. Note that there is a symmetry between y and z, but not in x, because in our setup x is always the coordinate aligned with the largest axis $a = 1$. The maximal errors are again of order 10^{-2} and the mean errors are below 10^{-4}. Note that the mean error in the potential evaluation of $7 \cdot 10^{-9}$ is much smaller than the mean error in the integral coefficients from the tests before. The reason for this is probably that a uniform spherical distribution in $\vec{x}$ leads to higher values of $\lambda(\vec{x})$ than the uniform distribution in λ on average. The interpolations seem to be better for higher values of $\lambda \sim 1$ (compare Figure 5.11 right).

Table 5.2: *Maximal and mean relative errors in the potential ϕ, the forces $\vec{a}$ and the tidal tensor T_{ij} for 10^6 randomly drawn scenarios.*

	ϕ	a_x	a_y	a_z	T_{xx}	T_{xy}	T_{xz}	T_{yy}	T_{yz}	T_{zz}
max / 10^{-2}	0.772	0.653	0.661	0.527	0.692	0.363	0.633	2.16	0.569	2.160
mean / 10^{-5}	0.0007	2.68	0.0007	0.0005	2.41	3.12	3.12	8.22	4.85	8.20

As our second tests for the potential evaluations correspond to a realistic case in N-Body simulations, we conclude that we are able to evaluate potentials/forces/tidal-tensors for the Epanechnikov ellipsoid on average more precisely than 10^{-4}, and in bad, but rare cases at least to precisions around 10^{-2}. We think that this precision is high enough, and therefore do not use a higher order interpolation scheme than the trilinear, as this would come at a significant performance cost.

5.4.3 Coordinate Transformations

So far we have discussed, how the potential can be evaluated in the eigensystem of the ellipsoid (i.e. the axes of the ellipsoid are aligned with the coordinate axes). Generally, all ellipsoids in our simulation can have an arbitrary alignment. We therefore transform for each pair interaction the distance vector $\vec{r} = \vec{r_i} - \vec{r_j}$ (where i names the particle that moves in the ellipsoidal potential of particle j) to the eigensystem of the ellipsoid $\vec{x}$, calculate the forces and tidal forces in the eigensystem, and transform the forces and tidalforces then back to the world system.

To transfer a displacement vector $\vec{r}$ to the eigensystem of the Ellipsoid $\vec{x}$, we have to use the transformation:

$$x_i = \sum_j \frac{\partial x_i}{\partial r_j} r^j \tag{5.101}$$

$$= \sum_j V_{ji} r^j \tag{5.102}$$

$$\vec{x} = V^T \cdot \vec{r} \tag{5.103}$$

where V is the matrix already defined in (5.10). After calculating the accelerations $\vec{a}'$ in the eigensystem of the ellipsoid, they transform back as:

$$\begin{aligned} a_i &= -\frac{\partial \phi}{\partial r_i} \\ &= -\sum_j \frac{\partial x_j}{\partial r_i} \frac{\partial \phi}{\partial x_j} \\ &= \sum_j V_{ij} \cdot a'_j && (5.104) \\ \vec{a} &= V \vec{a}' \,. && (5.105) \end{aligned}$$

Similarly, the tidal forces transform as:

$$\begin{aligned} T_{ij} &= -\frac{\partial^2 \phi}{\partial r_i \partial r_j} \\ &= -\frac{\partial}{\partial r_j} \sum_k \frac{\partial x_k}{\partial r_i} \frac{\partial \phi}{\partial x_k} \\ &= -\sum_l \frac{\partial x_l}{\partial r_j} \frac{\partial}{\partial r_l} \sum_k \frac{\partial \phi}{\partial x_k} \frac{\partial x_k}{\partial r_i} \\ &= -\sum_{kl} \frac{\partial x_l}{\partial r_j} \frac{\partial x_k}{\partial r_i} \frac{\partial^2 \phi}{\partial x_k \partial x_l} \\ &= \sum_{kl} V_{ik} T'_{kl} V_{jl} && (5.106) \\ T &= V \cdot T' \cdot V^T && (5.107) \end{aligned}$$

Where we have used, that we have a linear transformation $\frac{\partial}{\partial r_l} \frac{\partial x_k}{\partial r_i} = 0$.

5.4.4 Force Split

In the TreePM Method used in Gadget 2 (Springel, 2005) and Gadget 3 the potential is split into a long-range and a short-range part

$$\phi = \phi_S + \phi_L \tag{5.108}$$

where the long-range potential

$$\phi_{L,k} = \phi_{pm,k} \exp(-k^2 r_s^2) \tag{5.109}$$

is calculated by the particle-mesh in Fourier space, and the short range force is calculated in real space using the tree. We denoted $\phi_{pm,k}$ the Fourier-Transformation of the point mass potential $-mG/r$. The long-range part is given in real space by

$$\phi_L = -\frac{Gm}{r} \mathrm{erf}\left(\frac{r}{2r_s}\right) \tag{5.110}$$

Given an arbitrary potential ϕ which we want to realize in the TreePM method, the short-range-force will have to satisfy

$$\begin{aligned}\phi_S &= \phi - \phi_L && (5.111)\\ &= \phi + \frac{Gm}{r}\mathrm{erf}\left(\frac{r}{2r_s}\right) && (5.112)\\ &= \phi - \phi_{pm}\mathrm{erf}\left(\frac{r}{2r_s}\right) && (5.113)\end{aligned}$$

If the wanted potential is identical to the potential of a point mass $\phi = -mG/r$ this can be summarized to

$$\begin{aligned}\phi_S &= \left(1 - \mathrm{erf}\left(\frac{r}{2r_s}\right)\right) && (5.114)\\ &= \phi \cdot \mathrm{erfc}\left(\frac{r}{2r_s}\right) && (5.115)\end{aligned}$$

This is how the short-range potential is calculated in Gadget. One important subtlety at this point is, that this splitting is only correct if the potential is a point mass potential at those scales where the long-range forces matter. Or to put it in another way: This force split is only consistent if the softening is much smaller than the smoothing scale $\epsilon \ll r_s$.

This is usually satisfied in typical simulations where the softening is much smaller than the particle separation. However, in our case we want to choose a potential which might still be not point-mass-like at scales larger than r_s. We therefore do not use the force split like in (5.115), but instead we subtract the long-range potential explicitly like in (5.112). Note that in this way we can calculate ellipsoidal potentials up to the radius $r_{\mathrm{S,cut}} = 4.5 r_s$ at which the short-range summation is stopped. We always put an upper limit to the axes of the ellipsoids in a way that $r_{\mathrm{cut}} < r_{\mathrm{S,cut}}$, where r_{cut} was the radius at which we assume the potential to be equivalent to a point mass potential. This way the axes are limited by

$$a \cdot q_{\mathrm{cut}} < r_{\mathrm{S,\ cut}} \qquad (5.116)$$

This leads to the limitation for the maximum rescale factor in terms of code parameters:

$$q_{\max} = \min\left(\mathrm{AS\_MAXRESCALE}, \frac{\mathrm{boxsize}\cdot\mathrm{ASMTH}\cdot\mathrm{RCUT}}{1.17\epsilon_0\cdot\mathrm{PMGRID}\cdot\mathrm{AS\_RCUT}}\right) \qquad (5.117)$$

To get the correct force-split we derive (5.112) and find

$$\begin{aligned}F_{is} &= -\frac{\partial\phi_s}{\partial x_i} && (5.118)\\ &= F_i + \frac{MG}{r}\left(\mathrm{erf}\left(\frac{r}{2r_s}\right) - \frac{r}{r_s\sqrt{\pi}}\exp\left(-\frac{r^2}{4r_s^2}\right)\right) && (5.119)\\ &=: F_i - F_{i,pm}\cdot\alpha(r/2r_s) && (5.120)\end{aligned}$$

which becomes the forcesplit that is in the standard implementation of Gadget if one sets $F_i = F_{i,pm}$ (compare, e.g. Bagla (2002) Equation 7):

$$F_{is} = F_i \cdot \left(\mathrm{erfc}\left(\frac{r}{2r_s}\right) + \frac{r}{r_s\sqrt{\pi}}\exp\left(-\frac{r^2}{4r_s^2}\right)\right) \tag{5.121}$$

We derive the Forces another time to obtain also the force split for the tidal tensor:

$$\begin{aligned}
T_{ij,s} &= \frac{\partial F_{is}}{\partial x_j} && (5.122)\\
&= T_{ij} + \left(\delta_{i,j}\frac{m}{r^3} - 3\frac{m x_i x_j}{r^5}\right)\left(\mathrm{erf}\left(\frac{r}{2r_s}\right) - \frac{r}{r_s\sqrt{\pi}}\exp\left(-\frac{r^2}{4r_s^2}\right)\right) && (5.123)\\
&\quad + \frac{m x_j x_i}{r^5}\frac{r^3}{2r_s^3\sqrt{\pi}}\exp\left(-\frac{r^2}{4r_s^2}\right) && (5.124)\\
&= T_{ij} - T_{ij,pm}\left(\mathrm{erf}\left(\frac{r}{2r_s}\right) - \frac{r}{r_s\sqrt{\pi}}\exp\left(-\frac{r^2}{4r_s^2}\right)\right) && (5.125)\\
&\quad + \frac{m x_j x_i}{r^5}\frac{r^3}{2r_s^3\sqrt{\pi}}\exp\left(-\frac{r^2}{4r_s^2}\right) && (5.126)\\
&=: T_{ij} - T_{ij,pm}\alpha(r/2r_s) + \frac{m x_j x_i}{r^5}\beta(r/2r_s) && (5.127)
\end{aligned}$$

which is consistent with the tidal-tensor cut in Vogelsberger et al. (2008) (the equation between 40 and 41) if we use the point mass potential again.

$$T_{ij,s} =: T_{ij}\left(\mathrm{erfc}\left(\frac{r}{2r_s}\right) - \frac{r}{r_s\sqrt{\pi}}\exp\left(-\frac{r^2}{4r_s^2}\right)\right) + \frac{F_i x_j r}{2\sqrt{\pi}r_s^3}\exp\left(-\frac{r^2}{4r_s^2}\right) \tag{5.128}$$

We defined in between the functions α and β which are tabulated at the beginning of the simulation. The relevant equations we use for our force cuts now are equations (5.113), (5.120), and (5.127). Note that we do not use exactly the point mass potential for ϕ_{pm}, $F_{i,pm}$, and $T_{ij,pm}$, but we use a potential softened on the scale 1/10 of the grid-spacing. Mathematically, this should make no difference, because the PM force is already suppressed by several orders of magnitude at such small distances. However, we do not want to run into any trouble with divisions by small numbers, if particles come very close to each other.

5.4.5 Time-Steps

In Gadget3 the time-step criterion for only gravitationally interacting particles is

$$\Delta t_{\mathrm{grav}} = \min\left[\Delta t_{\mathrm{max}}, \left(\frac{2\eta\epsilon}{|\vec{a}|}\right)^{1/2}\right] \tag{5.129}$$

where Δt_{max} is an upper limit for the time steps, η is an accuracy parameter, ϵ is the softening and $\vec{a}$ is the acceleration of a particle. As we changed the softening which defines the length-scale on which forces change, we also have to adapt the time-step criterion. Therefore we employ two different possibilities:

$$\epsilon \to \epsilon_0 \cdot q_{\mathrm{min}} \,. \tag{5.130}$$

This chooses the absolutely minimal reasonable softening for the time-stepping criterion of all particles. This is a very conservative choice and can be activated with the directive AS_TIMESTEP_MINIMAL. As second possibility we define

$$\epsilon \to \epsilon_c \tag{5.131}$$

where ϵ_c is the softening along the smallest semi-axis c of the corresponding particle $\epsilon_c \approx 0.85c$. We tried both choices in different situations and found that the results did not differ significantly. Both choices seem to be conservative enough. Therefore we usually use the second choice, as it is faster. It is activated if AS_TIMESTEP_MINIMAL is not defined as a directive.

5.5 Summary and Usage

We implemented Anisotropic Softening into Gadget 3 which can evaluate forces and the tidal tensors for the potential of ellipsoids with Epanechnikov kernel. The ellipsoids are initially spheroidal with an effective radius that is defined as usual by the softening parameter ϵ_0. For later times the ellipsoids are distorted according to the distortion tensor that is followed by the Geodesic Deviation Equation. We list all precompiler-directives with a suggested value in Table 5.3. The only other value changing the behaviour of the Anisotropic Softening is the softening from the parameter file. The softening should be chosen at least slightly larger than the mean particle separation. We also list switches that add output quantities to the snapshot-files in Table 5.4. Some further comments on some switches:

The switch of AGS has to be turned on, as we use a similar direct summation criterion like the AGS-Code, just with a different direct summation radius r_{cut}. Therefore we store our direct summation radius r_{cut} =AS_RCUT$\cdot\, a$ in the code-variable AGS_Hsml, and use the part of the AGS-Code that updates the maximal direct summation radius in the tree. To output the direct summation radius use the switch AGS_OUTPUTGRAVSOFT.

AS_VERBOSE=1 adds some very useful output in the beginning of the simulation. Infos are displayed about the axis limits and the tests described in section 5.4.2 are done for 1000 random test cases each. This way it can easily be checked whether the accuracy is still acceptable when changing the minimal allowed axis ratios to a value different than 10^{-2}. Also the real maximum rescaling parameter q_{max} will be stated. All output that comes from the Anisotropic Softening code begins with "AS: [...]" so that one can easily find all corresponding output by the shell command "grep ˆAS: outfile":

```
AS: Allocating 20.00 MB for the interpolation grids
AS: BaseSoftening, BaseAxis: 500, 585.662
AS: qmin, qmax: [0.05, 2.10099]
AS: qmax is given by MIN(AS_MAXRESCALE, qmax-pm)
AS: qmax-pm = boxsize * ASMTH * RCUT/(1.17 * soft * PMGRID *AS_RCUT)
AS: Min Axis Ratio: 5.00e-02 / 5.00e+00 = 1.00e-02
AS: Setting up 64x64x64 interpoaltion grid with
```

Table 5.3: *Precompiler-directives, their effect and a suggest example-value. The bottom part designates foreign components that are needed by the code.*

directive	effect	suggested value
ANISOTROPIC_SOFTENING	main switch	defined
AS_ANISOTROPIC_FORCE	Use ellipsoids for force evaluation	defined
AS_ANISOTROPIC_TIDAL	Use ellipsoids for tidal tensor evaluation	defined
AS_RCUT	Transition r_{cut}/a to point mass potential	2.0
AS_MINRESCALE	Minimal rescaling factor	e.g. 0.05
AS_MAXRESCALE	Maximal rescaling factor	e.g. 5.0
AS_TIMESTEP_MINIMAL	Use conservative time-step criterion	undefined
AS_VERBOSE	Degree of output (0, 1 or 2)	1
ADAPTGRAVSOFT	AGS main switch - is required	defined
AGS_NOCORRECTION	No need for correction terms	defined
DISTORTIONTENSORPS	Main switch for GDE	defined
GDE_TYPES	Particle types that use the GDE	2
UNEQUAL_SOFTENINGS	Is in conflict with AGS switch	undefined

```
AS: log alpha, logbeta in [-2.20, 0.20], lambda in [0.00, 4.00]
AS: Testing Ellipsoidal Integral Interpolation with 1000 3d-randomvariables for
AS: logalpha/beta in (-2.00, 0) and lambda in (0, 4.00)
AS: Errors for      J0   J1a   J1b  J2aa  J2ab  J2bb  J2bc   J1c  J2ac  J2cc
AS: Mean/1e-3    0.127 0.097 0.153 0.088 0.124 0.173 0.134 0.142 0.116 0.158
AS: Max /1e-3    2.853 3.049 2.744 3.131 2.406 3.564 1.870 2.919 3.103 2.952
AS: Testing Ellipsoidal Potentials/Forces/Tidaltensors for 1000 settings
AS: axisratios logalpha/beta in (-2.00, 0) and r (0, 2.00)
AS: Maximum relative errors for phi, acc, tid:
AS:  1.92e-03    1.06e-03   ( 5.44e-04  5.34e-04  9.82e-04)
AS:              1.67e-03   (           3.88e-03  1.86e-03)
AS:              1.15e-03   (                     2.63e-03)
AS: Mean relative errors for phi, acc, tid:
AS:  1.92e-06    2.75e-05   ( 2.58e-05  3.35e-05  3.29e-05)
AS:              1.67e-06   (           8.88e-05  5.30e-05)
AS:              1.15e-06   (                     8.68e-05)
```

Table 5.4: *Directives that add additional output to the snapshot files. "#" denominates the number of floats used for the output, and "block" the block-name that is used if snapshots are written in the type 2 output format. The central part of the table mentions interesting related quantities from other parts of the code. In the bottom part we list quantities that might additionally be found in the snapshot files, as they are standard output quantities of the GDE, but that are not of interest for this Thesis.*

directive	quantity	#	block
AS_OUTPUT_SEMI_AXES	a, b, c	3	AXES
AS_OUTPUT_DISTORTION_EIGENSYSTEM	V-Matrix	9	DIES
.	λ_{ip} see (5.7)	3	DIEV
AS_OUTPUT_DISTORTION_ANISOTROPY	$\lvert\lambda_{\mathrm{p,max}}\rvert / \sqrt[3]{\lvert\lambda_{0p}\lambda_{1p}\lambda_{2p}\rvert}$	1	ANIS
AS_OUTPUT_ANGLES	Angles between $\vec{e}_{p,i}$ and $\vec{e}_i$	3	ANIS
AGS_OUTPUTGRAVSOFT	r_{cut}	1	AGSH
OUTPUT_DISTORTIONTENSORPS	6x6 Distortion tensor	36	DIPS
OUTPUT_TIDALTENSOR	Tidal tensor	9	TIPS
DISTORTIONTENSORPS	Stream Density	1	STDE
.	Caustic Counts	1	CACO
.	Flow Determinant	1	FLDE
.	Phase Space Determinant	1	PSDE
.	Annihilation Radiation	3	ANRA
GDE_READIC	Init Density	1	INDE
.	Sheet Orientation	9	SHOR

Chapter 6

Simulations without Artificial Haloes

In this chapter, we apply the Anisotropic Softening to two different numerical setups. The first one - the anti-symmetrically perturbed plane wave - can be understood as the formation of a homogeneous filament, and serves as a simple test problem under idealized conditions. As a second set-up we present simulations in a cosmological environment.

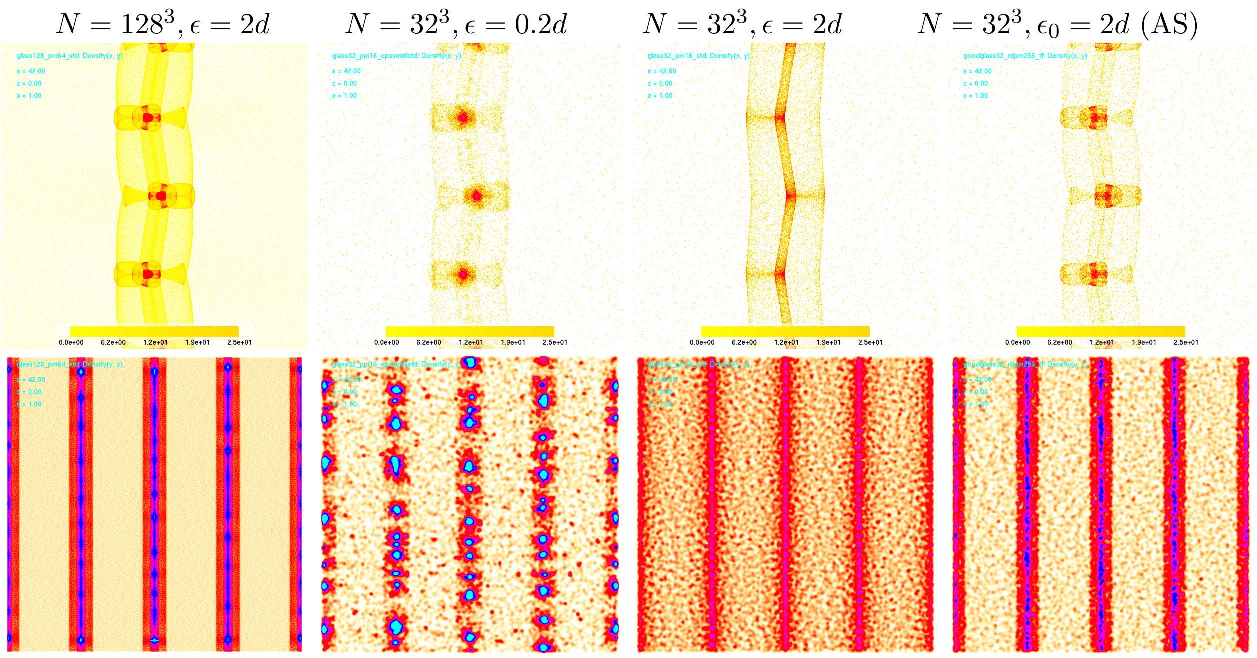
6.1 The Anti-symmetrically Perturbed Plane Wave

We set up initial conditions for the anti-symmetrically perturbed wave described in Valinia et al. (1997). This is a two dimensional problem that uses the plane-wave perturbation from the Zeldovich-Pancake along the x-dimension, and adds a small sinusoidal phase perturbation along the y dimension. The initial gravitational potential of this problem is given by

$$\phi(\vec{x}) = \overline{\phi} \cos \left(k_p \left[x + \epsilon_a \frac{k_p}{k_a^2} \cos k_a y \right] \right) . \tag{6.1}$$

We choose the same parameters like Hahn et al. (2013) $k_p = 2\pi/L$, $k_a = 4\pi/L$ and $\epsilon_a = 0.2$. $\overline{\phi}$ is chosen so that first shell crossing occurs at an expansion factor of $a_c = 1/7.7 \approx 0.13$.

This problem is particularly useful to study the formation of artificial haloes in filaments, since it can be understood as the formation of a homogeneous filament. There is first the collapse along the x-Dimension to a pancake, followed by a collapse along the y-dimension forming a homogeneous filament along the z-direction. As physically there should be perfect symmetry along the z-direction, any collapse along the z-direction will be clearly an artificial numerical effect. We set up our initial conditions on a glass. By using a glass instead of a grid, we can ensure that there is no stabilization due to perfect symmetries which will also not be the case in a typical Cosmological simulation. We had some trouble with creating initial conditions which do not suffer from large scale anisotropies. We already use the best initial conditions that we could create in this section, but we will describe them more detailed in the next section 6.2. Note that in Figure 4.2 we this problem showed already for a grid in the x, y plane, as visually it was more intuitive to follow the folding of the sheet for the grid like distribution.



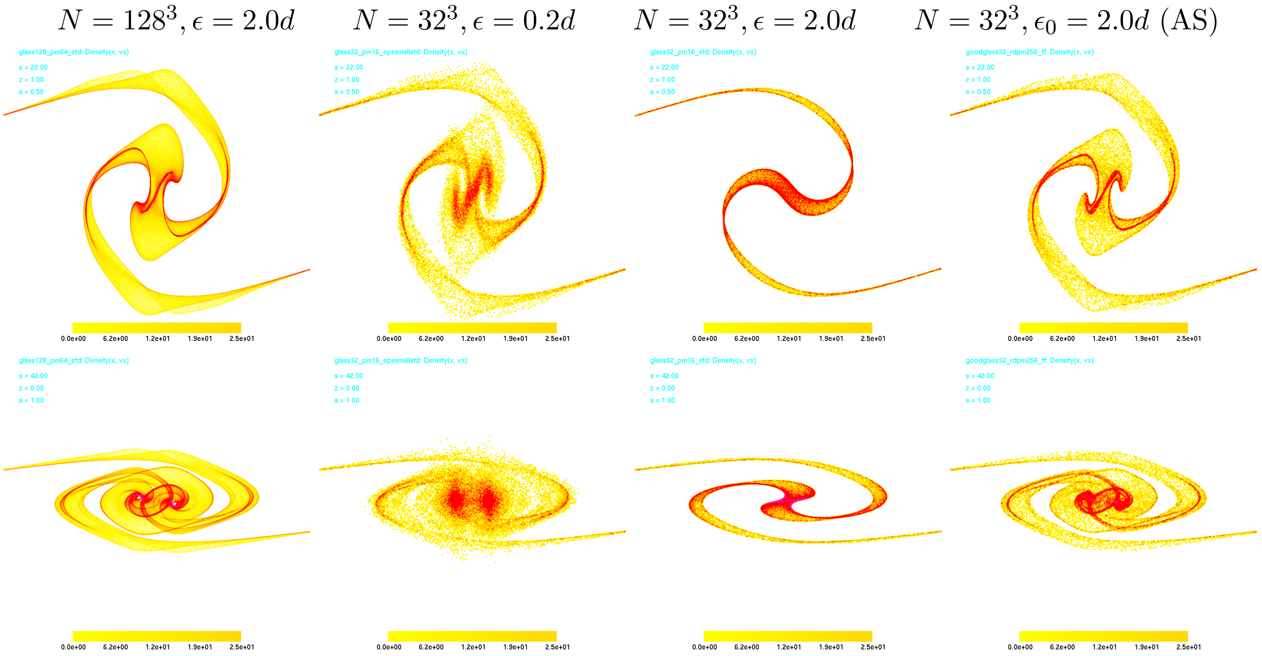
$N = 128^3, \epsilon = 2d$ $N = 32^3, \epsilon = 0.2d$ $N = 32^3, \epsilon = 2d$ $N = 32^3, \epsilon_0 = 2d$ (AS)

Figure 6.1: *Real Space projected densities of the ASPW for different numerical setups. Top:* (x, y)*, bottom:* (y, z)*. There is only physical dynamics in the* x *and* y *dimension, inhomogeneities in the* z *direction are of unphysical origin. First column: high resolution N-Body with high softening for comparison. Second column: N-Body with softening smaller than the particle separation* $\epsilon < d$*, artificial haloes are forming. Third column: N-Body with softening larger than the particle separation* $\epsilon > d$ *- no artificial haloes, but also physical structure is missing, as forces have not been resolved properly. Fourth column: Anisotropic Softening. As the force resolution is dynamically matched with the mass resolution, no artificial haloes form, while all physical structure is reproduced correctly.*

We visualize the density distributions in real space in Figure 6.1 and in phase space in Figure 6.2. It can be seen that with Anisotropic Softening neither artificial haloes are created, nor gets the force resolution underestimated. Beneath the problem of the fragmentation of filaments, here it can also be seen that the particle distribution between the filaments shows a much more granular behaviour for N-Body with $\epsilon \ll d$ than the smooth distributions from the anisotropic softening. Although it is usually thought that only filaments are subject to artificial fragmentation, it seems that the particle distribution in pancakes might also be contaminated by artificial effects of a too small softening.

6.2 Glass Making and PM Force-Anisotropies

Glass files are used to represent a homogeneous distribution discretized to particles. Other possibilities to represent a homogeneous distribution are for example cubic grids, a densest sphere packing, or just a random Poisson distribution. In cosmological simulations a Poisson distribution is not a good choice, as the Poisson noise is often much larger than the physical density perturbations that are to be followed. Grid-like representations are already more



$N = 128^3, \epsilon = 2.0d$ $N = 32^3, \epsilon = 0.2d$ $N = 32^3, \epsilon = 2.0d$ $N = 32^3, \epsilon_0 = 2.0d$ (AS)

Figure 6.2: *Phasespace diagramms (x, v_x) of the ASPPW for $a = 0.5$ (top) and $a = 1.0$ (bottom). First column: high resolution N-Body with high softening for comparison. Second column: N-Body with softening smaller than the particle separation $\epsilon < d$, two-body collisions destroy the fine-grained nature of the sheet. Third column: N-Body with softening larger than the particle separation $\epsilon > d$ there is no scattering, but the sheet is under-evolved, as forces have not been resolved properly. Fourth column: Anisotropic Softening. As the force resolution is dynamically matched with the mass resolution, the structure of the sheet is reproduced precisely whereas there have been no collisions.*

suitable, as forces cancel out mostly so that perturbations can be represented well. However, they also have the disadvantage of introducing anisotropies into the simulation. A glass is a homogeneous, amorphous distribution that also minimizes forces between particles. A glass for N-Body simulations can be produced by starting from a Poisson distribution, evolving the system with negative time steps (forces are repulsive this way), and taking away all particle-velocities each step. This way the system converges into a minimum energy distribution where all forces nearly vanish.

However, like described in Wang and White (2007), this is not the whole story of making a good glass. As the Gadget-Code does not evaluate all forces per direct summation, but uses the particle mesh and the tree to efficiently approximate long-range forces, some long-range anisotropies are introduced into all simulations. To make a good glass, the effects of those anisotropies have to be minimized. In Figure 6.3 we show that even on a grid where forces should cancel out exactly, the particle mesh can introduce unphysical forces. These forces are usually small compared with the physical forces, but still they can make a big difference, especially if the simulation starts at very high redshift. We use two methods to decrease the unphysical forces from the particle mesh. (1) We choose an extraordinary large smoothing parameter of asmth=3.5, (2) we introduce a random displacements technique for the particle

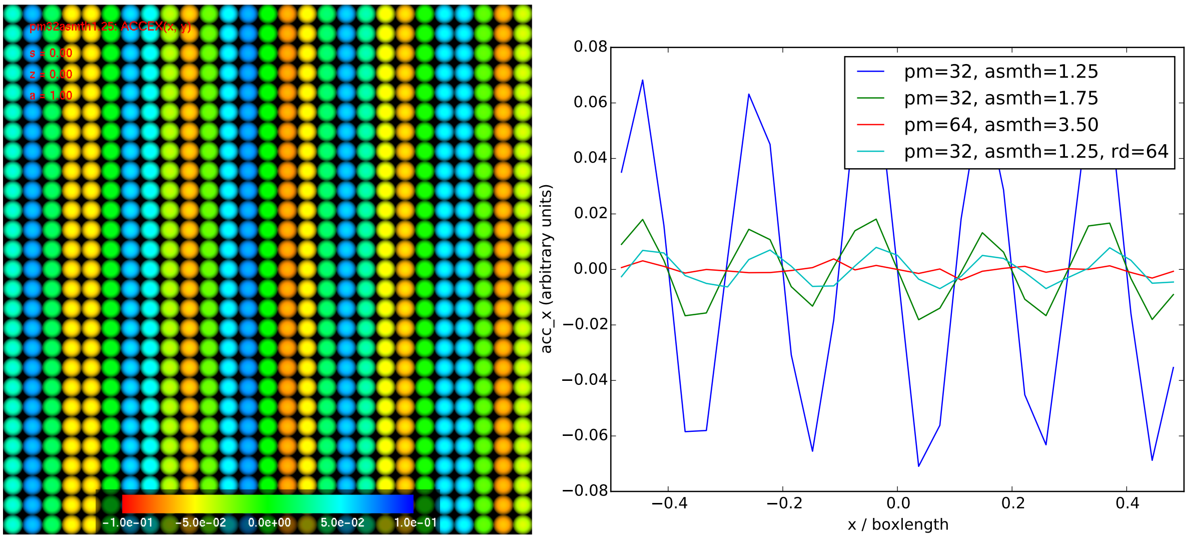


Figure 6.3: *Unphyisical numerical forces a_x on a uniform grid with $N = 27^3$. Left: Forces a_x as color versus (x, y) position of the particles for $PM = 32$ and $asmth = 1.25$. a_x forces along the x direction for different choices of code parameters. The force errors get smaller with larger smoothing radii asmth. Note that 1.25 is a common choice for asmth. Also we can see that the visible force errors come from the particle mesh and not from the tree, as the cases $(pm = 32, asmth = 1.75$ and $pm = 64, asmth = 3.5$ have the same forcecut (and therefore same tree forces), but a different amplitude of the errors. Using random displacements (see text) can also reduce the problem.*

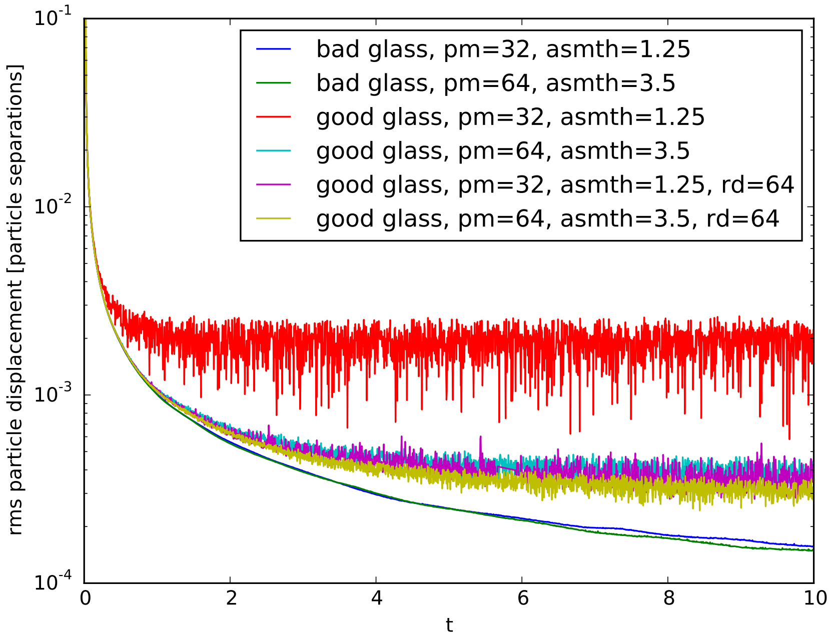


Figure 6.4: *Convergence for different glass making procedures (rms particle displacement for each timestep).*

mesh. Therefore we repeatedly add a random displacement to all particles, calculate the PM force on that particle distribution, save the result, and average later over all force-realisations. If there were no errors in the force calculation, the forces would be the same each time, as the system should be Galilei-invariant. However, since there is no exact Galilei-invariance due to the force-anisotropies, this procedure can reduce the effect of those. This can be seen in Figure 6.3 right for the case where we used $rd = 64$ random displacements. This procedure is inspired by Wang and White (2007) who added such a random displacement to all particles each time step, but did not average over different realizations. We adopt their name giving of a "bad glass" for a glass that was made without any random displacements, a "good glass" for a glass that uses one displacement each time step, and a "good glass" with a rd-value for a glass that was made averaging pm-forces over several realizations each time step.

In Figure 6.4 we show the convergence of the glass making procedure for different glass-making techniques. It can be seen that bad glasses converge very well. However, this is not a good thing, as the converged distribution contains the effects of the large scale anisotropies. The good glasses stop converging at a point where the noise that is introduced from the (now random) force anisotropies, dominates the particle displacements. However, a better convergence for good glass can be achieved by reducing the force anisotropies by a larger smoothing parameter or by averaging over several realizations. We made a glass with $N = 32^3, pm = 64, asmth = 3.5$ and $rd = 64$ which we used for the simulations from the last section, and which we also use for all subsequent simulations.

6.3 Simulations in a full Cosmological Environment

We set up initial conditions for a dark matter only simulation in a cosmological box with the initial conditions generator S-GenIC, an updated version of Volker Springel's IC-Generator N-GenIC. We use all the same cosmological parameters like Angulo et al. (2013) so that a future comparison might be easy $\Omega_m = 0.276$, $\Omega_\Lambda = 0.724$, $\Omega_b = 0.045$, $h = 0.703$, $\sigma_8 = 0.811$, $n_s = 0.96$. We adopt their choice of a WDM mass $m_X = 250\,\mathrm{eV}$, and to get an equivalent power spectrum we calculate a CLASS (Blas et al., 2011) CDM spectrum and rescale it like Angulo et al. (2013) with their version of the formula from Bode et al. (2001) :

$$T_{\mathrm{WDM}} = T_{\mathrm{CDM}}[1 + (\alpha k)^2]^{-5.0} \tag{6.2}$$

$$\alpha = 0.05 \left(\frac{\Omega_m}{0.4}\right)^{0.15} \left(\frac{h}{0.65}\right)^{1.3} \left(\frac{m_X}{1\,\mathrm{keV}}\right)^{-1.15} \left(\frac{1.5}{g_x}\right)^{0.29} h^{-1}\mathrm{Mpc} \tag{6.3}$$

Angulo et al. (2013) have chosen a box with a width of 80 Mpc, at a particle number $N = 1024^3$ leading to a mass resolution of $3.65 \times 10^7 h^{-1} M_\odot$. However, as our Code has big performance issues yet, and we do not have the time left to do any further conceptional changes, for the Anisotropic Softening we just perform a simulation of much lower resolution $N = 128^3$, boxsize=40 Mpc, $M_{\mathrm{part}} = 2.34 \times 10^9 h^{-1} M_\odot$, as a first test case. We might be doing more sophisticated simulations in the future, but they will not become part of this thesis anymore. The simulation ran with the softening parameters $\epsilon_0 = 1.6d$, $q_{\mathrm{min}} = 0.05$, and $q_{\mathrm{max}} = 1.05$. The upper limit q_{max} for the rescale factors was in this case given by the limit for a consistent

Table 6.1: *List of simulation parameters. D: Boxsize, ϵ: softening, d: particle separation, M_p: particle mass. Last column denotes the figures where density projections of the corresponding simulations appear. N-Body simulations with the incommon large softening are marked by an asterisk.*

Method	Spectrum	D/(Mpc/h)	N	ϵ/(kpc/h)	ϵ/d	$M_\mathrm{p}/M_\odot$	Figures
AS	WDM	40	128^3	500	1.6	$2.34 \cdot 10^9$	6.5, 6.6, 6.9
N-Body	WDM	40	128^3	20	0.064	$2.34 \cdot 10^9$	6.5
N-Body*	WDM	40	128^3	500	1.6	$2.34 \cdot 10^9$	6.5, 6.6
N-Body	CDM	40	128^3	20	0.064	$2.34 \cdot 10^9$	-
N-Body	WDM	40	256^3	10	0.064	$2.29 \cdot 10^8$	6.9
N-Body*	WDM	40	256^3	250	1.6	$2.29 \cdot 10^8$	6.6, 6.9
N-Body*	WDM	40	512^3	125	1.6	$3.65 \cdot 10^7$	6.6, 6.7
N-Body*	WDM	80	512^3	250	1.6	$2.29 \cdot 10^8$	6.7
N-Body	CDM	40	512^3	5	0.064	$3.65 \cdot 10^7$	6.7
N-Body	CDM	80	512^3	10	0.064	$2.29 \cdot 10^8$	6.7

forcecut from (5.117). We also do a set of N-Body simulations for comparisons. We list the parameters of these simulations in table 6.1.

6.3.1 Density Fields

We show projections of the density for the three WDM simulations with resolution $N = 128^3$ in Figure 6.5. While the standard N-Body case suffers from artificial fragmentation, the AS and the high softening N-Body cases do not show any artificial fragmentation. However, the N-Body case with high softening has also a big lack of physical structure. Due to too small force resolution it suppresses a lot of structure. Contrary the Anisotropic Softening reproduces all physical structures that can be found in the small softening case. Therefore it unites the benefits from both approaches.

However, also it shall be mentioned that the performance of the Anisotropic Softening is much slower than of the two other approaches. Compared with the N-Body case with large softening, it has a similar direct summation radius, but needs about ten times as long for every force evaluation, and has a much smaller time-stepping. Therefore one run took about a factor of 100 longer in total. Therefore the question has to be asked, whether just using N-Body simulations with $\epsilon \sim d$ at much higher resolution might be a more effective alternative.

In figure 6.6 we compare the density field of the Anisotropic Softening simulation with N-Body runs using a large softening $\epsilon = 1.6d$ at increasing resolutions $N = 128^3$, $N = 256^3$, and $N = 512^3$. From the visual inspection of this figure we infer, that the effective resolution of the Anisotropic Softening lies somewhere between the 2^3 times and 4^3 times higher resolved N-Body cases with large softening. This figure also demonstrates that N-Body simulations with large softening do not suffer from artificial fragmentation, and that their large-scale structure can be brought to convergence. Such simulations could be used for obtaining halo mass functions in principle. We discuss this in more detail in the next section.

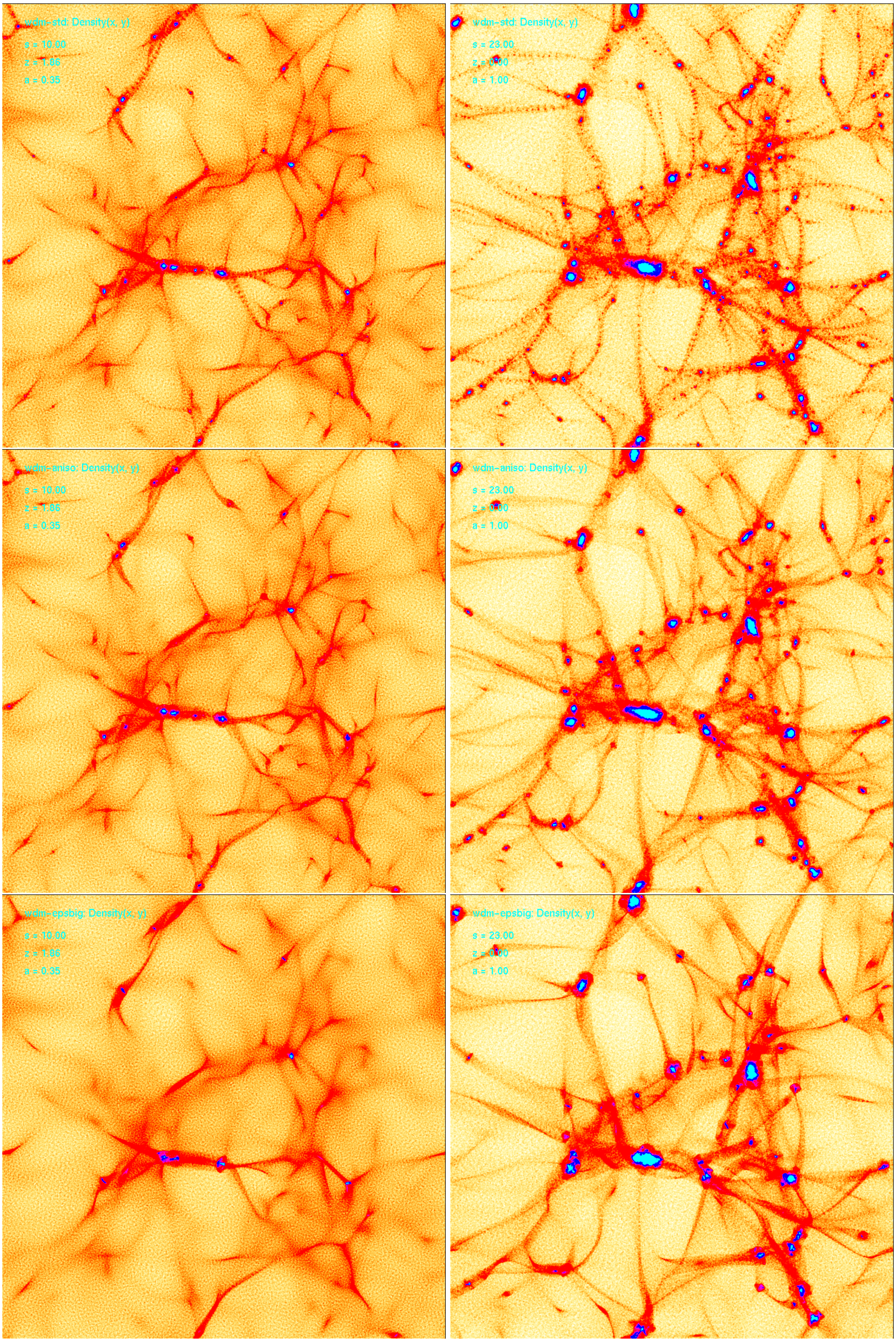


Figure 6.5: *Density projection of a* $(40\,\mathrm{Mpc/h})^3$ *box, left:* $a = 0.35$ *right:* $a = 1.0$*. Top: N-Body with* $\epsilon = 0.064d$*, centre: Anisotropic Softening with* $\epsilon_0 = 1.6d$*, bottom: N-Body with* $\epsilon = 1.6d$*.*

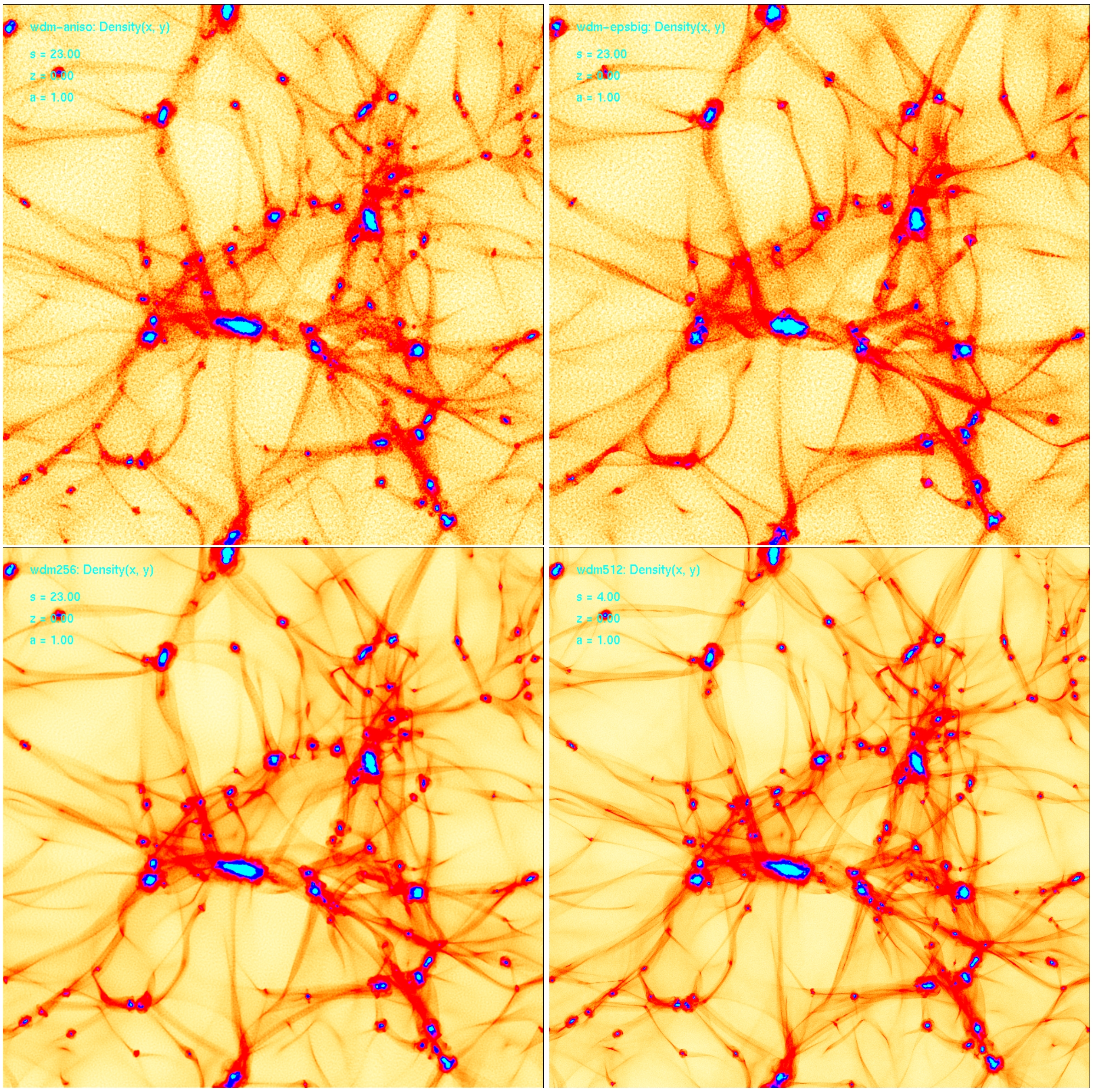


Figure 6.6: *Density projection of the* $(40\,\mathrm{Mpc/h})^3$ *box for different set-ups. Top left: anisotropic softening with* $N = 128^3$, $\epsilon_0 = 1.6d$. *The remaining panels N-Body with* $\epsilon = 1.6d$ *at resolutions* $N = 128^3$ *(top left),* $N = 256^3$ *(bottom left), and* $N = 512^3$ *(bottom right).*

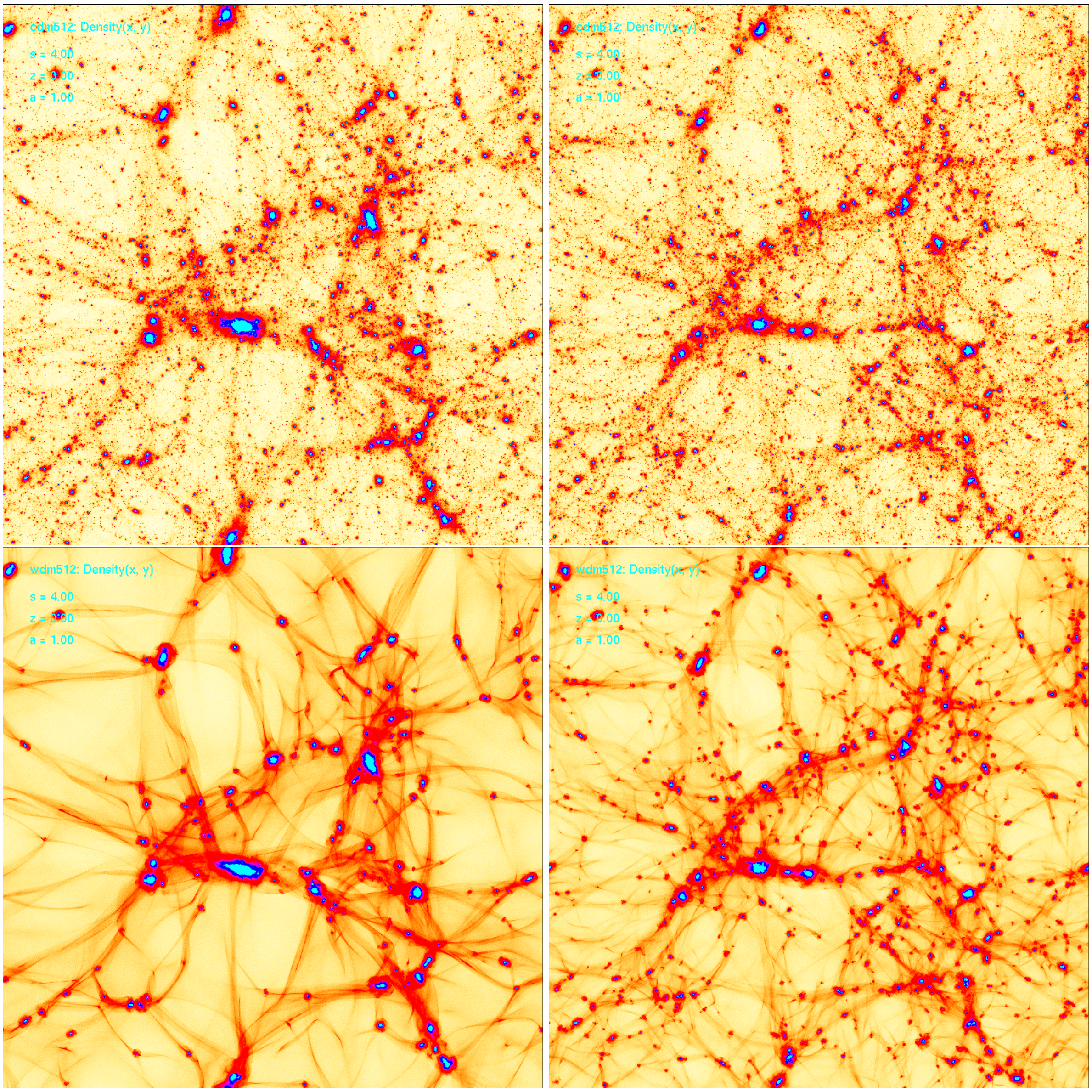


Figure 6.7: *Density projection of the simulation boxes of* $(40\,\mathrm{Mpc/h})^3$ *(left) and* $(80\,\mathrm{Mpc/h})^3$ *(right) for CDM (top) and WDM (bottom). Note, the* $80\,\mathrm{Mpc/h}$ *boxes show similar structure to the* $40\,\mathrm{Mpc/h}$ *at twice the length scale, because the same seeds were used when generating the initial conditions, and therefore the phases of the initial pertubations are the same, shifted to large scales.*

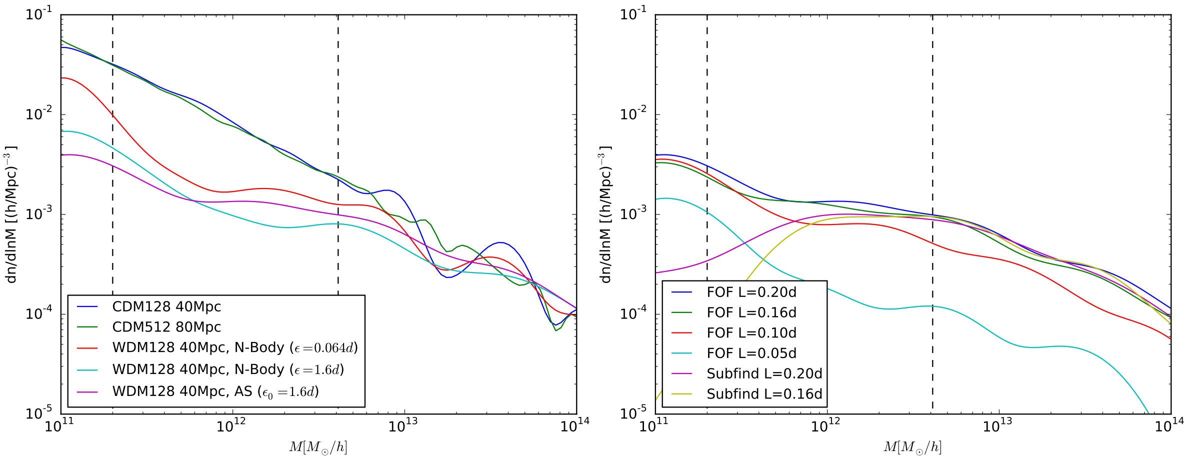


Figure 6.8: *Left: Halo mass functions for different simulations with a linking length of* $L = 0.20d$*. The HMF changes strongly with the numerical techniques for the WDM. However, these are not confident, as they are subject to problems with the Halo Finding. Right: Halo mass functions for the WDM simulation with Anisotropic Softening and* $N = 128^3$ *for different halo finding techniques, i.e. FOF with differnt linking lengths and Subfind with two different linking lengths. With Subfind the number of low mass haloes is strongly decreased, as it rejects haloes that are not selfbound. Dashed lines indicate the boundaries in which the CDM scenarios match.*

In figure 6.7 a comparison between our highest resolution (512^3) WDM and CDM cases is shown for boxes with width 40 Mpc and 80 Mpc. As expected the small scale structure is strongly suppressed in the WDM case. However, it is worth noting that there is a lot of small scale structure appearing in the WDM cases that just started to fragment into smaller haloe-like structures. These structures originate probably from modes which lie at scales smaller than the maximum scale of the power spectrum $k > k_{\mathrm{max}}$ (compare figure 2.2), but get strongly amplified during the collapse of larger structures. For many of these objects it is hard to say whether they might be called a halo or not. For example, Angulo et al. (2013) distinguished between haloes and "proto-haloes" and obtained two different halo mass functions - one with and one without proto-halos.

6.3.2 Halo Mass Function

In N-Body simulations, haloes are typically detected by a friends-of-friends (FOF) algorithm. That is, particles are linked by a length L to groups of particles. After the linking all particles which are in groups of less than 32 particles are rejected. All other groups form haloes with masses given by the sum of their particle's masses. Typically the linking length is chosen to be $L = 0.2d$. However, Angulo et al. (2013) reported that this linking length led to perculating structures in their simulations. In CDM simulations everything is fragmented on all scales, but in WDM simulations the particle distribution remains smooth on smaller scales. This enhances the linking strongly. To address this problem they chose a small linking length of $L = 0.05d$.

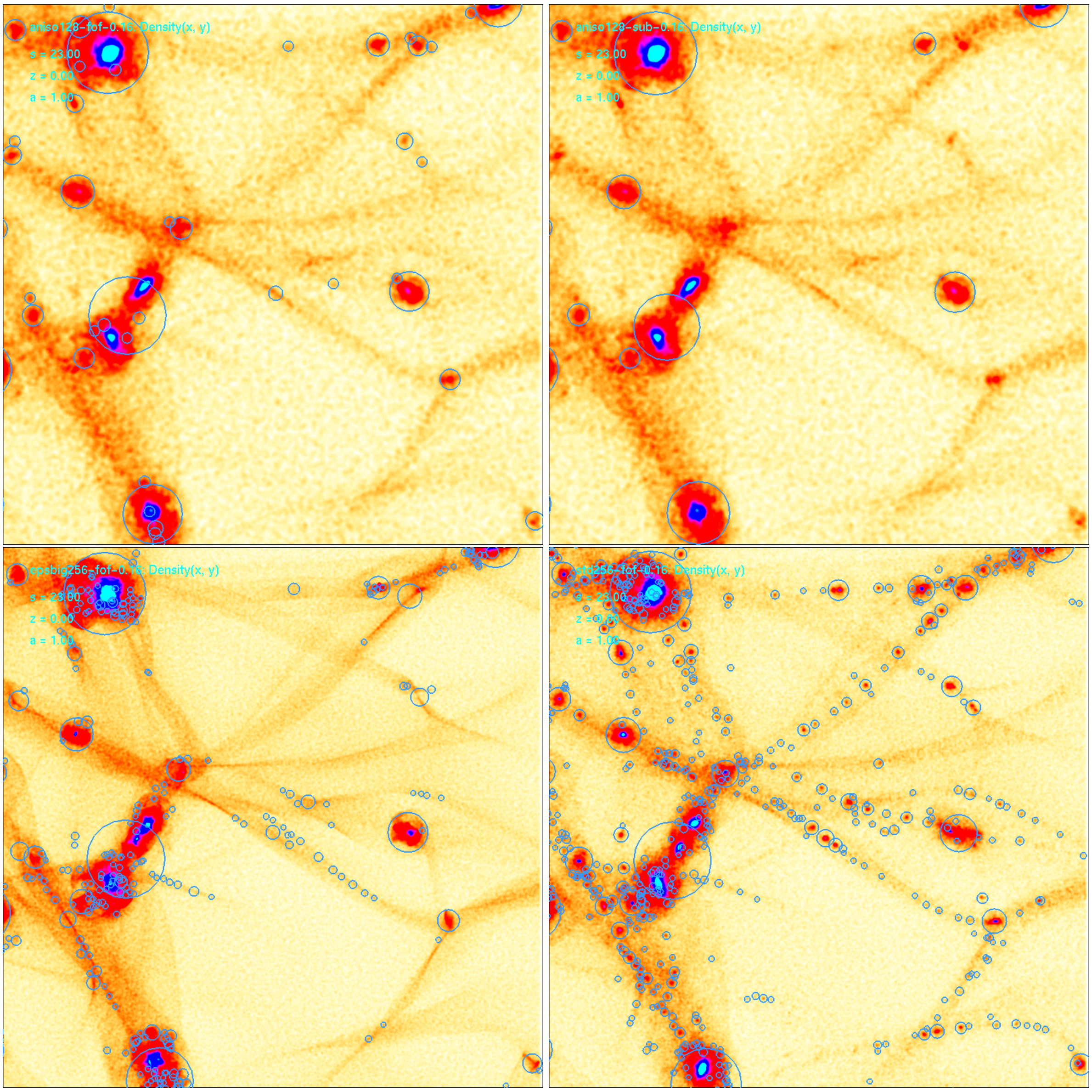


Figure 6.9: *Density maps with detected haloes marked ($r = R_{200}$) for different scnearios: Top-Left: FOF with $L = 0.16d$ in the AS simulation ($N = 128^3$) the FOF finds all physical structures, but also some incorrect structures close to haloes. Top right: Same simulation, but using Subfind. No false detections can be seen, but also some physical structures have not been detected. Bottom-Left: Simulation with Large softening and $N = 256^3$ - the number of FOF-false detections become worse with higher resolution. Bottom Right: Simulation with small softening - here many detections correspond to artificial haloes. It becomes clear that the FOF false detections are something different than artificial haloes.*

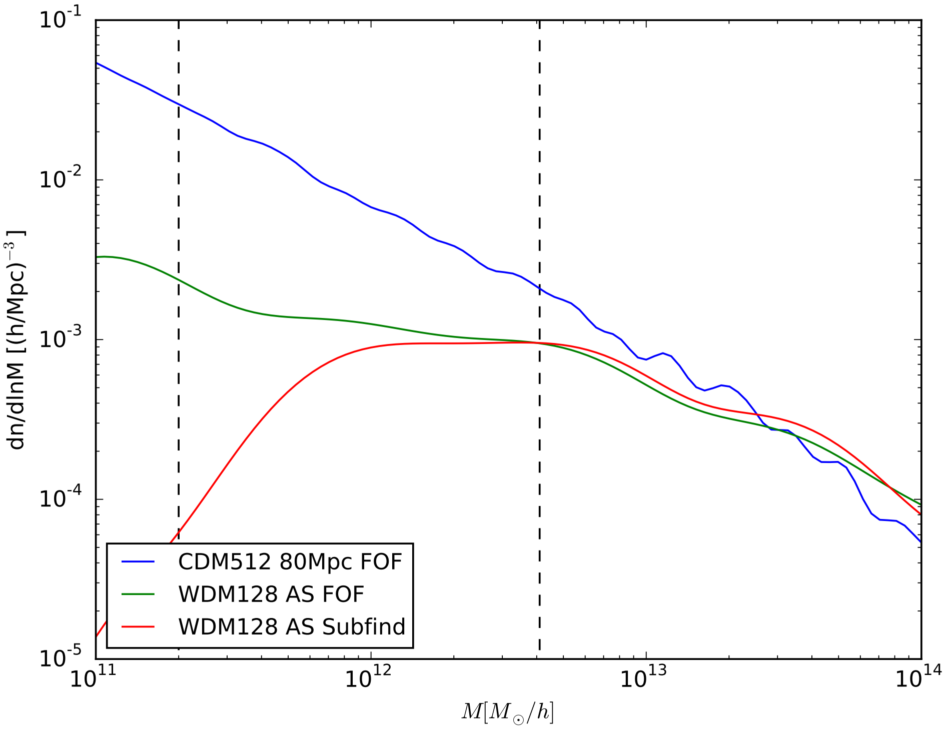


Figure 6.10: *Limits on the halo mass function of WDM, and the halo mass function for CDM. The functions originate from FOF runs with* $L = 0.16$ *and a Subfind run with* $L = 0.16$*. The Subfind result can server as an upper limit and the FOF result as a lower limit to the real halo mass function of the Anisotropic Softening simulation at small scales.*

In figure 6.8 we show the HMF from our simulations with a standard linking length (left) and the results of tests of different linking lengths (right). By visual inspection of our simulations we find no percolating structures like Angulo et al. (2013)[7], but we find as a problem that the FOF algorithm finds a big number of small structures where we only find a very smooth density distribution by visual inspection. We illustrate this in figure 6.9. Therefore we also tested Subfind which rejects structures that are not self bound after the FOF linking. Subfind seems to reject all wrong detections, but seemingly also rejects a significant number of physical structures.

We can not present a confident halo mass function here, because the uncertainties from the halo finding algorithms are too big. However, since we have seen that the FOF algorithms overestimate and Subfind understimates the number of small haloes, we can at least show an upper and lower boundary in figure 6.10.

6.3.3 The Distortion Ellipsoids

In the remaining Figures we show some interesting output quantities from the AS-simulation. In Figure 6.11 we show the caustic count. The caustic count is a discrete number that

---

[7] probably because our resolution is much lower

describes how often a particle has gone through a caustic where its volume element inverted. It is evaluated in the code from Vogelsberger and White (2011) by counting the sign changes of the determinant of the real space distortion tensor. Initially it is zero, and in voids it always stays zero. At first collapse it gets a value of one and is increasing from there on. Using it for visualisation one can much easier distinguish collapsed structures from voids. For example the pancakes become visible in Figure 6.11 which is not the case for the usual density projections like Figure 6.5.

*** Rotation angles
In Figure 6.12 we show the eigenvalues of the distortion-ellipsoid. It can be seen that the largest eigenvalue is nearly allways bigger than 1. That means that the largest axis a of the ellipsoids will be larger than its initial value. In our simulation the rescale factors were limited by $q_{\max} = 1.05$. Therefore effectively all interactions within a radius of $2 \cdot 1.05 \cdot r_0$ have been evaluated by direct summation[8]. This is the major reason for the bad performance of the code. Further, the smallest eigenvalue λ_3 determines the size of the time steps. From figure 6.12 we anticipate that this can become very small especially in high density regions. Therfore the timesteps become the smallest where the highest amount of direct-summation is needed.

---

[8] The factor 2 comes from the direct summation radius being AS_RCUT · a.

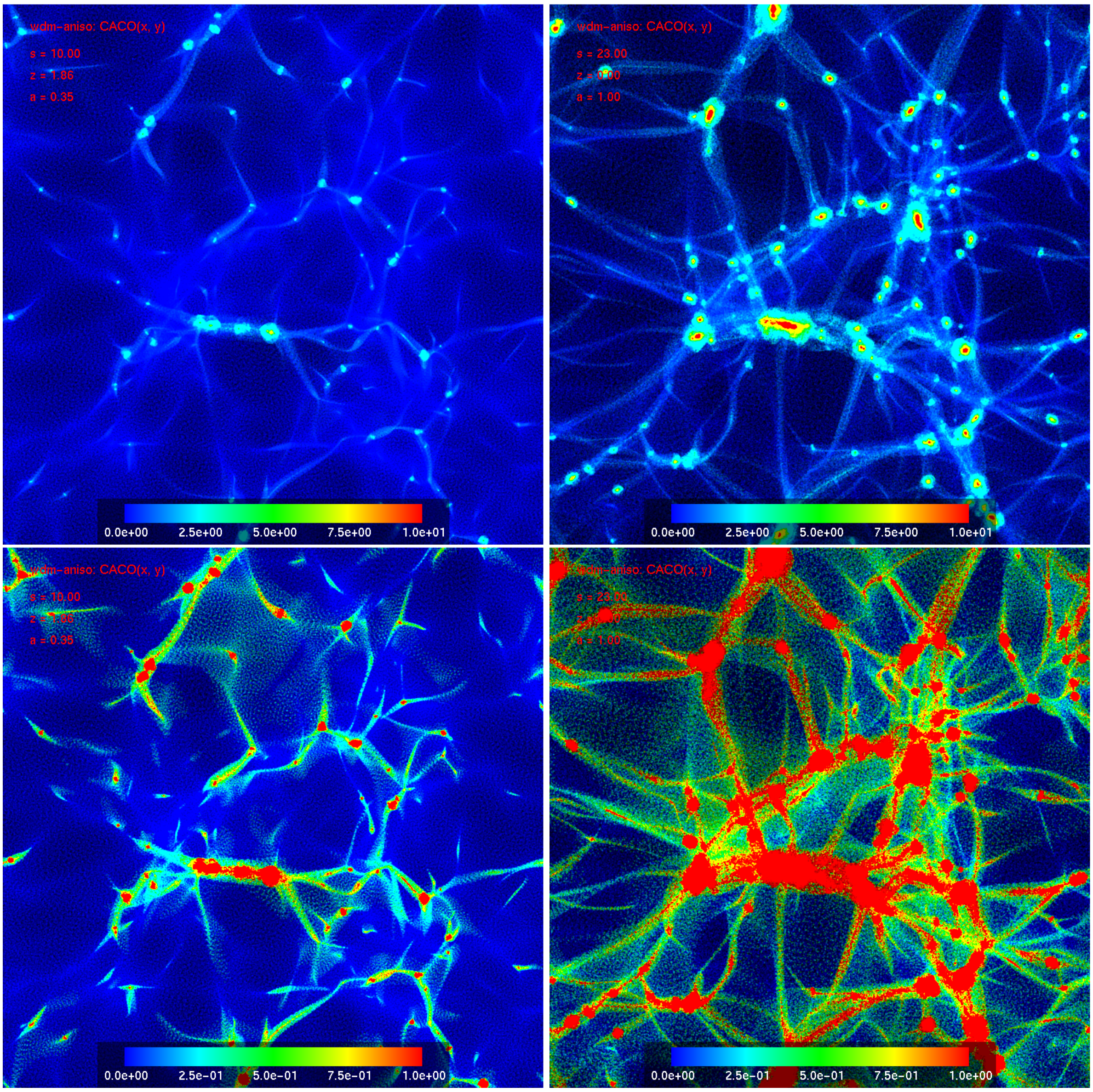


Figure 6.11: *Caustic counts for the simulation with Anisotropic Softening at* $a = 0.35$*, and* $a = 1.0$*. Top and bottom just with color scales different by a factor of ten. The caustics count is usually an integer-valued quantity, but as we average it along the line of sight, it can also take non-integer values here. Note how pancakes (greenish 2 dimensional structures between filaments) become nicely visible in the bottom figures. In density projections (compare Figure 6.5) they usually cannot be seen, as a projected void and a pancake are hardly distinguishable. The caustic counts are* $>= 1$ *in collapsed structures, but* 0 *for voids, and can be used for qualitative discriminations.*

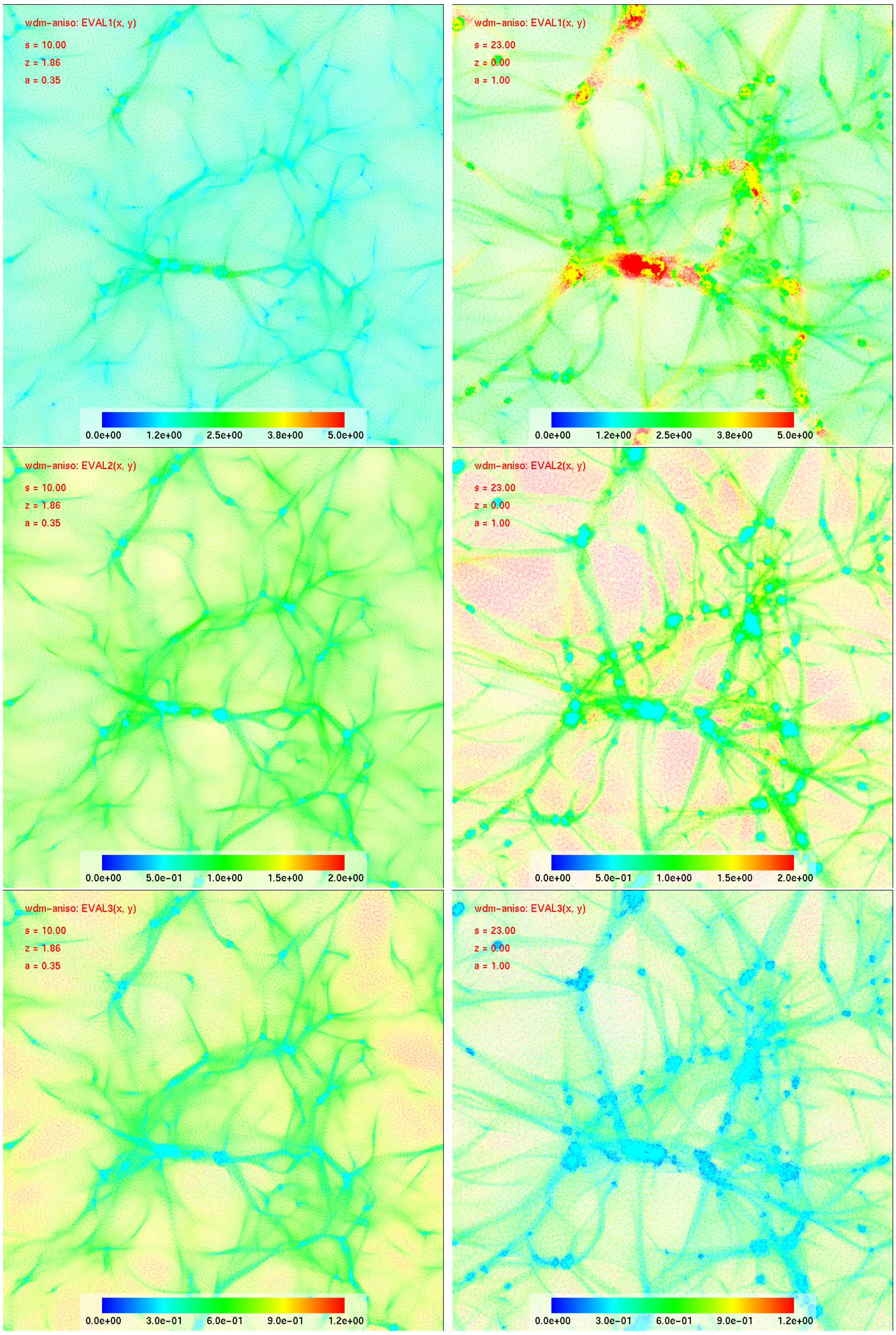


Figure 6.12: *Eigenvalues of the distortion ellipsoid at* $a = 0.35$ *(left) and* $a = 1.0$ *(right), from top to bottom* $|\lambda_1| > |\lambda_2| > |\lambda_3|$*. Color scales are chosen individually in every row. At* $a = 1$*,* λ_1 *seems to be larger than one nearly everywhere,* λ_2 *is larger or smaller than one, depending on the type of structure, and* λ_2 *seems to be smaller than one, everywhere except voids. Note, how strongly* λ_1 *increases in the central halo in the upper right panel. In high density regions the complexity of the dark matter sheet grows rapidly.*

Chapter 7

Discussion and Conclusions

The numerical modelling of structure formation in warm dark matter universes suffers from the formation of artificial haloes at small scales. This seems to be a problem that appears if the softening is chosen much smaller than the mean particle separation, i.e. if the force resolution is chosen much larger than the mass resolution. We found that Adaptive Gravitational Softening can not help out here, as it cannot follow the real mass resolution in situations of highly anisotropic collapse. Unfortunately these situations are the common case in cosmological simulations.

Therefore we decided to develop a new numerical technique which we call *Anisotropic Softening*. Using the distortion tensor from the Geodesic Deviation Equation (Vogelsberger and White, 2011), Anisotropic Softening follows the distortions of small ellipsoidal volume elements around each particle. We solved the potential of an ellipsoid with Epanechnikov-kernel to derive a consistent potential from a density estimate based on the distortion-ellipsoids. This way we can choose an initial softening similar to the mean particle separation that consistently follows the mass resolution during phases of highly anisotropic collapse, as it can get rotated and distorted in all three dimensions independently.

We tested our code with the anti-symmetrically perturbed plane wave (Valinia et al., 1997) which can be understood as the formation of a homogeneous filament. At low resolution we found already remarkable agreement between Anisotropic Softening and the high resolution N-Body case. Further there has not been an artificial fragmentation of the filament, like for the N-Body simulations with small softening. Last we tested the code in a full cosmological context, and found that it could indeed overcome the formation of artificial haloes while not suppressing any physical structures.

However, while performing very well at the same resolution in a physical sense, Anisotropic Softening also suffers from much larger computational costs compared to same resolution N-Body simulations. It is desirable to do a more sophisticated performance study in the future, but for the moment we find about a factor of 100 in computational time at same resolution. This is due to the more complicated force evaluation, a large number of direct summation interactions and a small time stepping. The biggest issue when scaling it to larger simulations is the direct summation radius. We see two possibilities to improve on the performance.

i One could try to deposit the ellipsoids onto the particle mesh. Then one would not need to limit the ellipsoid-sizes by the scale of the force split, and therefore use much higher resolved particle meshes, and save lots of effort with the short range interactions.

ii One could implement a more sophisticated treatment of large softening in the tree. Currently, if only one softening in a tree node is larger than its separation to an interacting particle, the whole node is opened. As in most nodes there will be at least one particle with one large axis present, nearly all particles within a radius $\sim r_0$ will be summed over via direct summation. From Figure 6.12 we infer that the ellipsoids generally seem to have one axis, that is larger than its initial size, and two axes that are smaller or similar to their initial sizes. One could split the tree into several trees, that order the ellipsoids according to their alignment, and their extensions. Thereby one could distinguish better between interactions that really have to be evaluated via direct summation, and interactions that can still be approximated by the tree node.

It seems like phase space methods like Anisotropic Softening, the Hahn et al. (2013) scheme, and the Hahn and Angulo (2015) scheme can overcome the problem of artificial haloes. However, up to the point where they are fully operational in high-density regions, and publicly accessible for everyone, we suggest to use N-Body simulations with a softening slightly larger than the mean particle separation with glass-based initial conditions for investigations of warm dark matter. It might take a very high resolution to bring them to a converged stage, but at that stage, we would not expect artificial structures.

Bibliography

Abel, T., Hahn, O., and Kaehler, R. (2012). Tracing the dark matter sheet in phase space. *Monthly Notices of the Royal Astronomical Society*, 427:61–76.

Angulo, R. E., Hahn, O., and Abel, T. (2013). The warm dark matter halo mass function below the cut-off scale. *Monthly Notices of the Royal Astronomical Society*, 434:3337–3347.

Bagla, J. S. (2002). TreePM: A Code for Cosmological N-Body Simulations. *Journal of Astrophysics and Astronomy*, 23:185–196.

Blas, D., Lesgourgues, J., and Tram, T. (2011). The Cosmic Linear Anisotropy Solving System (CLASS). Part II: Approximation schemes. *Journal of Cosmology and Astroparticle Physics*, 7:34.

Bode, P., Ostriker, J. P., and Turok, N. (2001). Halo Formation in Warm Dark Matter Models. *The Astrophysical Journal*, 556:93–107.

Borgani, S. and Viel, M. (2009). The evolution of a pre-heated intergalactic medium. *Monthly Notices of the Royal Astronomical Society*, 392:L26–L30.

Cai, W. (2007). Potential field of a uniformly charged ellipsoid. http://micro.stanford.edu/ caiwei/me340/A_Ellipsoid_Potential.pdf (2015-07-28).

Colless, M., Dalton, G., Maddox, S., Sutherland, W., Norberg, P., Cole, S., Bland-Hawthorn, J., Bridges, T., Cannon, R., Collins, C., Couch, W., Cross, N., Deeley, K., De Propris, R., Driver, S. P., Efstathiou, G., Ellis, R. S., Frenk, C. S., Glazebrook, K., Jackson, C., Lahav, O., Lewis, I., Lumsden, S., Madgwick, D., Peacock, J. A., Peterson, B. A., Price, I., Seaborne, M., and Taylor, K. (2001). The 2dF Galaxy Redshift Survey: spectra and redshifts. *Monthly Notices of the Royal Astronomical Society*, 328:1039–1063.

Green, A. M., Hofmann, S., and Schwarz, D. J. (2005). The first WIMPy halos. *Journal of Cosmology and Astroparticle Physics*, 8:3.

Hahn, O., Abel, T., and Kaehler, R. (2013). A new approach to simulating collisionless dark matter fluids. *Monthly Notices of the Royal Astronomical Society*, 434:1171–1191.

Hahn, O. and Angulo, R. E. (2015). An adaptively refined phase-space element method for cosmological simulations and collisionless dynamics. *ArXiv e-prints*.

Hobbs, A., Read, J., Agertz, O., Iannuzzi, F., and Power, C. (2015). Novel Adaptive softening for collisionless N-body simulations: Eliminating spurious halos. *ArXiv e-prints*.

Iannuzzi, F. and Dolag, K. (2011). Adaptive gravitational softening in GADGET. *Monthly Notices of the Royal Astronomical Society*, 417:2846–2859.

Kaehler, R., Hahn, O., and Abel, T. (2012). A Novel Approach to Visualizing Dark Matter Simulations. *ArXiv e-prints*.

Lovell, M. R., Frenk, C. S., Eke, V. R., Jenkins, A., Gao, L., and Theuns, T. (2014). The properties of warm dark matter haloes. *Monthly Notices of the Royal Astronomical Society*, 439:300–317.

Macciò, A. V., Paduroiu, S., Anderhalden, D., Schneider, A., and Moore, B. (2012). Cores in warm dark matter haloes: a Catch 22 problem. *Monthly Notices of the Royal Astronomical Society*, 424:1105–1112.

Planck Collaboration, Ade, P. A. R., Aghanim, N., Arnaud, M., Ashdown, M., Aumont, J., Baccigalupi, C., Banday, A. J., Barreiro, R. B., Bartlett, J. G., and et al. (2015). Planck 2015 results. XIII. Cosmological parameters. *ArXiv e-prints*.

Ramachandra, N. S. and Shandarin, S. F. (2015). Multi-stream portrait of the cosmic web. *Monthly Notices of the Royal Astronomical Society*, 452:1643–1653.

Shandarin, S. F. and Medvedev, M. V. (2014). Tracing the Cosmic Web substructure with Lagrangian submanifold. *ArXiv e-prints*.

Springel, V. (2005). The cosmological simulation code GADGET-2. *Monthly Notices of the Royal Astronomical Society*, 364:1105–1134.

Springel, V., White, S. D. M., Jenkins, A., Frenk, C. S., Yoshida, N., Gao, L., Navarro, J., Thacker, R., Croton, D., Helly, J., Peacock, J. A., Cole, S., Thomas, P., Couchman, H., Evrard, A., Colberg, J., and Pearce, F. (2005). Simulations of the formation, evolution and clustering of galaxies and quasars. *Nature*, 435:629–636.

Springel, V., Yoshida, N., and White, S. D. M. (2001). GADGET: a code for collisionless and gasdynamical cosmological simulations. *New Astronomy*, 6:79–117.

Tormen, G., Bouchet, F. R., and White, S. D. M. (1997). The structure and dynamical evolution of dark matter haloes. *Monthly Notices of the Royal Astronomical Society*, 286:865–884.

Valinia, A., Shapiro, P. R., Martel, H., and Vishniac, E. T. (1997). Gravitational Instability in Collisionless Cosmological Pancakes. *The Astrophysical Journal*, 479:46–69.

Viel, M., Becker, G. D., Bolton, J. S., and Haehnelt, M. G. (2013). Warm dark matter as a solution to the small scale crisis: New constraints from high redshift Lyman-α forest data. *Phys. Rev. D*, 88(4):043502.

Viel, M., Lesgourgues, J., Haehnelt, M. G., Matarrese, S., and Riotto, A. (2005). Constraining warm dark matter candidates including sterile neutrinos and light gravitinos with WMAP and the Lyman-α forest. *Phys. Rev. D*, 71(6):063534.

Vogelsberger, M. and White, S. D. M. (2011). Streams and caustics: the fine-grained structure of Λ cold dark matter haloes. *Monthly Notices of the Royal Astronomical Society*, 413:1419–1438.

Vogelsberger, M., White, S. D. M., Helmi, A., and Springel, V. (2008). The fine-grained phase-space structure of cold dark matter haloes. *Monthly Notices of the Royal Astronomical Society*, 385:236–254.

Wang, J. and White, S. D. M. (2007). Discreteness effects in simulations of hot/warm dark matter. *Monthly Notices of the Royal Astronomical Society*, 380:93–103.

Weinberg, D. H., Bullock, J. S., Governato, F., Kuzio de Naray, R., and Peter, A. H. G. (2013). Cold dark matter: controversies on small scales. *ArXiv e-prints*.

Acknowledgements

I wish to thank all people that supported me during the work of this thesis. First of all, I am very grateful to my supervisor PD Dr. Klaus Dolag for giving me the opportunity to work within the CAST group on such an interesting topic. He introduced me to simulation techniques, helped me with daily problems, always found time for interesting discussions, and never lost his optimistic, kind, and supportive attitude. It was really enjoyable for me to have a large degree of freedom to follow own ideas and suggestions during my thesis, guided by his foresight and reassured by his patience.

Further I want to thank all members of the CAST group, particularly Manuel Behrendt, Alessandro Ballone, Martin Zintl, Rhea-Silvia Remus, and Madhura Killedar for all the good advice and interesting discussions during the last year. Also a great thanks to my fellow students Zsolt, Martha, Patrice, Matthias, Daniel, Viktor, Tamás, Riccardo, Emilio, and many more. It have been two great years together with you, and I hope we will stay in contact wherever each of us is going. Of course, also I want to thank my family. It really keeps me going that always a warm and cosy bed will be free for me when I come home.

Last I want to express how much I appreciate the work of Volker Springel, for writing this great code Gadget, and Mark Vogelsberger and Simon White, for extending it by the Geodesic Deviation Equation. This is really a wonderful piece of mathematics, and I am looking forward to further on work with it during my PhD at the MPA in Garching.

Selbstständigkeitserklärung

Hiermit versichere ich, die vorliegende Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel benutzt. Mir ist bekannt, dass Zuwiderhandlung auch nachträglich zur Aberkennung des Abschluss führen kann.

München, 17.09.2015