

Halo finding in Modified Gravity N-body simulations

by

Simen Tennøe

Thesis

for the degree of

Master of Science in Astronomy

(Master i Astronomi)



*Institute of Theoretical Astrophysics
Faculty of Mathematics and Natural Sciences
University of Oslo*

Oktober 2013

*Institutt for teoretisk astrofysikk
Det matematisk- naturvitenskapelige fakultet
Universitetet i Oslo*

Abstract

Modified gravity theories are a popular research field in the hope that they could explain some of the unanswered questions in cosmology, for example how the universe starts, evolves and ends. One way to test proposed modified gravity models is by analyzing how the model affects the evolution of large scale structures in the universe. In the non-linear regime the only way of doing this is by performing N-body simulations, which simulates how a dynamical system of particles behave and evolve under the influence of physical forces. After performing such a simulation we wish to compare the data from the simulation with the observed universe. This cannot be done directly from the output from N-body simulations. To extract the necessary statistics a halo finding process should be performed, which determine how galaxies are grouped into halos and calculates the properties of these halos. As of yet, no one has taken into account the differences between standard general relativity and modified gravity in their halo finders, so the validity of other halo finders in the regime of modified gravity is therefore unknown.

This is what has been the focus of this thesis. Here, we introduce MORPH, the first halo finder that is completely independent of the gravity model used in the N-body simulations. MORPH can analyze any dataset from a modified gravity N-body simulation. This is performed without the need for code modifications to accommodate for the modified gravity theory. As a part of this work we have examined various unbinding algorithms and their dependence on the gravity model. The main question this thesis set out to answer was whether there is a justified need for a modified gravity adjusted halo finder. The conclusion is that modified gravity must be taken into consideration when we intend to analyze halos in modified gravity datasets. However, only if the halo finders have a high unbinding percentage, making the errors from the unbinding routine larger than the current 10% error bars for halo finding.

Acknowledgements

Firstly, I would like to thank my two supervisors Nicolaas Ervik Groeneboom and David F. Mota for their valuable help through my thesis. I am especially grateful to Nicolaas Groeneboom who has provided me with such an interesting thesis topic and answering my numerous questions. Many thanks also goes to Amir Hammami who first mentioned Nicolaas as a potential supervisor.

I would like to give many thanks to Claudio Llinares, who have provided much needed insight into some of the more theoretical parts of halo finding and modified gravity and Max Grönke for giving me helpful insight into how it could be possible to calculate R_{vir} independently of the gravity model.

In addition I am especially thankful to Katrine H. Olsen who have suffered through my bad grammar several times and helped me improve it and to Jorun Ramstad who have read through the algorithm chapter of this thesis and made sure there are no errors and being a helpful sounding board.

I would also like to thank my family and friends for being supportive and listening to my complaints when the work have taken its toll. Lastly, I would like to thank Realistforeningen for making the years I have worked on my master thesis a pleasant experience, and for being very helpful when I have felt the need to procrastinate.

Contents

1	Introduction	1
1.1	The beginning of modern cosmology	1
2	Modern Cosmology	7
2.1	Cosmic dynamics, the Friedmann equations	8
2.1.1	Equation of state	9
2.1.2	Evolution of the energy density of the universe	11
2.2	The flat Λ CDM universe	13
2.2.1	Friedmann equations for a Λ CDM universe	14
2.3	A shorter history of nearly everything	16
2.3.1	Inflation	16
2.3.2	The Big Bang Nucleosynthesis	18
2.3.3	Recombination	19
2.3.4	Re-ionization	19
2.3.5	Structure formation	19
3	Gravity	21
3.1	Introduction	21
3.1.1	Equivalence principle	21
3.1.2	Preliminaries	22
3.2	Einstein equations	23
3.2.1	Lagrange formalism of general relativity	25
3.2.2	Geodesic equation	27
3.3	Modified Gravity	27
3.3.1	Motivation for modified gravity	27
3.3.2	Modifications of Einstein's equations	28
3.4	Alternative gravity theories	29
3.4.1	Theories of Gravity with Extra Fields	29
3.4.2	Higher Derivative and Non-Local Theories of Gravity	29
3.4.3	Higher Dimensional Theories of Gravity	30
3.5	f(R) gravity	30
3.6	Symmetron model	32
3.7	Tests of general relativity	33
3.7.1	Classical tests	33
3.7.2	Modern tests	33

4	N-body simulations	37
4.1	Introduction	37
4.2	Numerical methods for calculation of the force	38
4.2.1	Direct summation	38
4.2.2	Tree codes	38
4.2.3	Particle Mesh codes	40
4.2.4	Particle-Particle Particle-Mesh codes	40
4.3	Gas dynamics	40
4.4	Numerical methods for moving particles	41
4.5	Ramses	41
4.6	ISIS	42
5	Halo Finders	43
5.1	Introduction	43
5.1.1	What is a Halo?	43
5.1.2	The need for a modified gravity adjusted halo finder	44
5.2	The General Method of Halo Finding	44
5.2.1	Identification of possible halo candidates	45
5.2.2	Particle Collection	46
5.2.3	Calculating position and velocity of halos	46
5.2.4	Unbinding procedure	46
5.2.5	Calculating halo mass and edge	47
5.3	Source of the scatter in the final result	47
5.4	Astrophysical Applications	48
5.4.1	Galaxy formation	48
5.4.2	Dark Matter Detection	49
5.4.3	Gravitational Lensing	49
5.4.4	Modified Gravity Simulations	49
5.5	AHF	50
5.6	Rockstar	50
6	MORPH Algorithms	53
6.1	Loading routines	53
6.2	Parallelization	55
6.3	Friends-of-friends algorithms	56
6.3.1	Friends-of-Friends N^2	56
6.3.2	Friends-of-Friends grid	56
6.3.3	FOF N^2 vs FOF grid runtime	60
6.4	6D phase-space splitting	60
6.4.1	6D phase space friends-of-friends algorithm	66
6.5	Unbinding	72
6.5.1	Calculating ϕ using a spherical approximation	73
6.5.2	Using ϕ using the acceleration of each particle	76
6.5.3	Using ϕ from the N-body simulation	77
6.5.4	Unbounding	77
6.6	Calculating R_{vir}	77

6.6.1	Algorithm	81
6.7	Results of the 6D phase space friends-of-friends algorithm	81
6.8	Mass estimation	81
6.8.1	Mass function	81
6.8.2	Choosing b	85
7	Results	89
7.1	Unbinding routines	89
7.2	R_{vir}	90
8	Conclusions and outlook	99
8.1	Conclusions	99
8.1.1	Do we need a halo finder that takes into account modified gravity models?	101
8.2	Outlook	101
8.2.1	Result improvements	101
8.2.2	Code improvements	102
A	MORPH, Libraries	105
A.1	Libraries	105
B	Code parameters	107

List of Figures

1.1	The Cosmic Microwave Background, image courtesy of wikipedia.	4
2.1	Difference between a positively curved, $k = +1$ universe, a negatively curved universe, $k = -1$ and a flat universe, $k = 0$	8
2.2	Evolution of the scalefactor for universes with $\Omega_{m0} = [0.9, 1.0, 1.1]$	15
2.3	Evolution of the universe, image courtesy of wikipedia.	16
4.1	Example of an octree, image courtesy of wikipedia	39
5.1	The general steps performed by a halo finder.	45
6.1	Illustration of array structure.	54
6.2	Plot of the positions of every 25th particle for the 128^3 particle dataset. . .	55
6.3	The positions of every 25th particle for the 128^3 particle dataset, after the 3D FOF algorithm	60
6.4	Runtime for MORPH using different parameters for the nr of particles needed for the scaling and the scaling itself.	61
6.5	Runtime for the FOF- N^2 scheme for different number of particles.	62
6.6	Runtime for the FOF-Grid scheme for different number of particles	62
6.7	An example of a linking bridge affecting a FOF group.	63
6.8	Plot of the positions of the particles in a single FOF group.	64
6.9	Plot of the velocities of the particles in a single FOF group, after the 3D FOF algorithm.	64
6.10	Plot of the x-position vs x-velocity, y-position vs y-velocity and z-position vs z-velocity for the particles in a single halo, after the 3D FOF algorithm.	65
6.11	Flowchart of the 6D phase space friends-of-friends method.	67
6.12	Mass functions for $f = [0.5, 0.6, 0.7, 0.8, 0.9]$	68
6.13	Potential from the N-body potential for different halos after the 6D phase-space splitting.	78
6.14	Positions of the particles in a single FOF group, after the 3D FOF algorithm (top) and after the 6D splitting (bottom).	83
6.15	Plot of the positions of the particles for each halo in a single FOF group, after the 6D FOF algorithm.	84
6.16	Mass functions for $b = [0.2, 0.21, 0.22, 0.23, 0.24, 0.25, 0.26, 0.27, 0.28, 0.29, 0.3]$	86
6.17	Mass function for $b = 0.21$	87
7.1	Mass function for lcdm1, four different unbinding routines.	91
7.2	Mass function for symm_A, four different unbinding routines.	91

7.3	Mass function for <code>symm_B</code> , four different unbinding routines.	92
7.4	Mass function for <code>symm_C</code> , four different unbinding routines.	92
7.5	Mass function for <code>symm_D</code> , four different unbinding routines.	93
7.6	Mass function for <code>fofr4</code> , four different unbinding routines.	93
7.7	Mass function for <code>fofr5</code> , four different unbinding routines.	94
7.8	Mass function for <code>fofr5</code> , four different unbinding routines.	94
7.9	Fractional difference between the accelerated unbinding for MG and the Newtonian case, smoothed over neighboring bins to remove noise that appear in the binning, for <code>lcdm1</code> and <code>symm_A</code> - <code>symm_B</code>	95
7.10	Fractional difference between the accelerated unbinding for MG and the Newtonian case, smoothed over neighboring bins to remove noise that appear in the binning, for <code>fofr4</code> - <code>fofr6</code>	95
7.11	Fractional difference between the two methods to calculate R_{vir}	97

List of Tables

5.1	Scatter in the main properties computed by halo finders, from Knebe et al. (2013)	48
6.1	FOF group information,. Positions are given as Mpc/h, velocities as Mpc/-Gyr and mass as M_{\odot}	63
6.2	Runtime for MORPH with $f = [0.5, 0.6, 0.7, 0.8, 0.9]$	66
6.3	6D split FOF group halo information,. Positions are in units of Mpc/h, velocities in Mpc/Gyr and mass in M_{\odot}	85
7.1	Analyzed dataset details.	89
7.2	Symmetron model details.	90
7.3	$f(R)$ model details.	90
7.4	The unbinding values for the different unbinding routines for each dataset .	96
7.5	The unbinding values for the different unbinding routines for each dataset, using the standard method to calculate R_{vir}	98
A.1	Code summary, from ohcount.	105

Chapter 1

Introduction

In this thesis we will discuss different halo finding algorithms and their implementation. We will also look at modified gravity theories before implementing our own halo finder that is independent of the gravity model used in the N-body simulation.

The outline of this thesis is as follows: In chapter 1 we will review the beginning of modern cosmology before diving further into the topic in chapter 2. In chapter 3 the topic turns to general relativity and modified gravity theories, elaborating on why it is useful to consider modified gravity models. Chapter 4 gives a short introduction to N-body simulations. In chapter 5 we will discuss halo finders and give a short description of two of the popular halo finders, one from each major type of algorithms. We will take a look at MORPH, the halo finder written in this thesis, and implemented algorithms in chapter 6. Chapter 7 will provide the results obtained from MORPH. Lastly, chapter 8 discloses the conclusions reached as well as giving a short outlook on possible advancements.

Cosmology is the study of the universe as a whole, its content, structure and evolution. It is a fairly new branch of science and began with the evolution of general relativity in the 1920s. Before this, cosmology was considered a branch of metaphysics and/or religion. A science that regards the universe as one big entity might seem far removed from the concerns of humanity, but cosmology tries to answer important questions that have been asked for a long time, such as: what is the universe made of? How did the universe begin and how will it end? Is the universe infinite in size, or does it have boundaries? Cosmology draws heavily from the work within several very different areas of physics, such as theoretical and experimental particle physics, quantum mechanics, general relativity and plasma physics. As such, cosmology spans from general relativity, being on the massive scale to quantum mechanics which is on the minute scale.

1.1 The beginning of modern cosmology

We can learn a lot about the universe from simple observations of the night sky. If the universe is infinite and have existed forever, then why is the night sky dark? No matter in which direction one look we are going to have line of sight to a star, and no matter how far away that star is, the light would have had time to reach us. Therefore the sky would be as bright as the surface of stars. Since this is clearly not the case we can conclude that the universe cannot both be infinite and have existed for an infinite amount of time.

The basis for all of cosmology is the Copernican principle, also known as the cosmological principle, which states that the universe on large scales is homogeneous and isotropic. This means that there is no place in the universe that is special, it does not matter from where or in which direction we look, the universe is going to seem similar.

Light from stars and galaxies typically contain absorption spectra and any one given absorption line is always emitted at the same wavelength. By looking at where the absorption spectra should be compared to where we observe it, we can calculate the Doppler shift of the light from a star and find out if it moves towards, or away from us. This factor z , called the cosmological redshift, is given as:

$$z \equiv \frac{\lambda_{\text{ob}} - \lambda_{\text{em}}}{\lambda_{\text{em}}}, \quad (1.1)$$

with λ_{ob} the wavelength of the observed spectral line and λ_{em} the emitted wavelength. When looking at the light from distant objects we discover that nearly all of them are redshifted¹, no matter in which direction we look. It might seem that our position in the universe is special, we are at the position from which everything moves away, but these observations is exactly what we would expect to see from a expanding universe. Every point is moving away from each other point, so an observer in a distant galaxy would observe the same phenomena. An important distinction, the cosmological redshift does not originate because the observed objects are moving away from us², but because space itself is expanding, resulting in the light waves being stretched. In 1929 astronomer Edwin Hubble plotted the redshift of galaxies against their distance from us and got a linear relation called Hubble's law (Hubble 1929). This law relates distance, d , and redshift z ,

$$z = \frac{H_0}{c}d, \quad (1.2)$$

with c the speed of light and H_0 the Hubble's constant, $H_0 = 70 \pm 7\text{km/s/Mpc}$, which measures how fast space expands. Using Hubble's law we see that we can use redshift as a measure of distance in addition to being used as a measure of time, due to the expansion of the universe. If we interpret the redshift as a Doppler shift we get a better known versions of Hubble's law:

$$v = H_0d. \quad (1.3)$$

This can be used to find the velocity, v , of an object moving away from us when the distance, d , is known. Since we have established that the universe is expanding, we introduce a scale factor, $a(t)$. This factor tells us how much the universe has expanded in a given timespan. The norm is to set it to 1 at our time, $a(t_0) \equiv 1$, where t_0 is the time today. The Hubble parameter, H , which gives Hubble's constant at a time t , is defined using this scalefactor,

$$H(t) \equiv \frac{da(t)}{dt} = \frac{\dot{a}(t)}{a(t)}, \quad (1.4)$$

where $\dot{a}(t)$ is the time derivative of the scalefactor. The Hubble parameter measures how fast the scale factor changes. The scalefactor also gives us the following relation linking

¹This is the reason why z is called the cosmological redshift.

²The traditional Doppler shift.

distance between two objects today and at any another time t :

$$d(t) = a(t)d_0. \tag{1.5}$$

Rearranging this equation and taking the derivative with respect to time gives us Hubble's law. If we discover how the scale factor changes with time we know how the universe evolves. This is why there is such an interest in solving the equations for $a(t)$.

We now know that the universe cannot be both infinite and have existed for an infinite time, or perhaps neither and that the universe today is expanding. From this knowledge, two different cosmological models evolved in the 1920s, the Big Bang theory and the Steady State model. In the big bang theory the universe starts in a extremely dense and hot initial state and as time goes by the universe expands and subsequently the density and temperature is decreased. The average distance between galaxies are increasing as the universe evolves with time. The Steady State model assumed an addition to the cosmological principle, namely that there is no favored moment in time. Therefore the Steady State universe is infinitively old, if there existed a time where the universe came into being that would certainly be a special moment in time. This universe expands following Hubble's law, but for it to be in a steady state, the density of the universe must remain constant. Because of the continuously expanding space, matter must be created at an equal rate.

During the 1950s and 1960s these two models coexisted. The followers of the big Bang theory argued that the continuous creation of matter violates mass-energy conservation, while the followers of the Steady State model retorted with it being no less weird than the creation of the entire universe in one big explosion. This debate lessened in 1965 when Arno Penzias and Robert Wilson discovered that the universe is filled with a isotropic background of microwave radiation (Penzias & Wilson 1965) called the Cosmic Microwave Background (CMB). The temperature of this radiation is fir particularly well by a black-body radiation with a temperature $T = 2.7$ K. An image of the CMB can be seen in figure 1.1, taken by Planck, a satellite designed to detect small perturbations in the CMB. The existence of the CMB is an important cosmological clue and is one of the reasons that the Big Bang theory was favored instead of the Steady State model. In the Big Bang theory CMB is explained as "left over" radiation from a time when the universe was very hot and dense. When the universe expanded it became less dense and the radiation cooled down. The Steady State model on the other hand have no good explanation for the CMB. Today the success of the Big Bang theory mainly rests on three observational pillars.

- The Cosmic Background Radiation.
- The abundances of light elements in the universe.
- The expansion of the universe, as found by Hubble's law.

In 1916 Albert Einstein published a paper containing the theory of general relativity (GR) (Einstein 1916). This was a generalization of the special theory of relativity. It introduces the concept of spacetime, a model that combines space and time into one entity. It is usually interpreted as space being three dimensions with time being the fourth. The effects of gravity is understood as caused by the curvature of spacetime, everything that has a mass curves the spacetime around it. GR is one of the most important cosmological

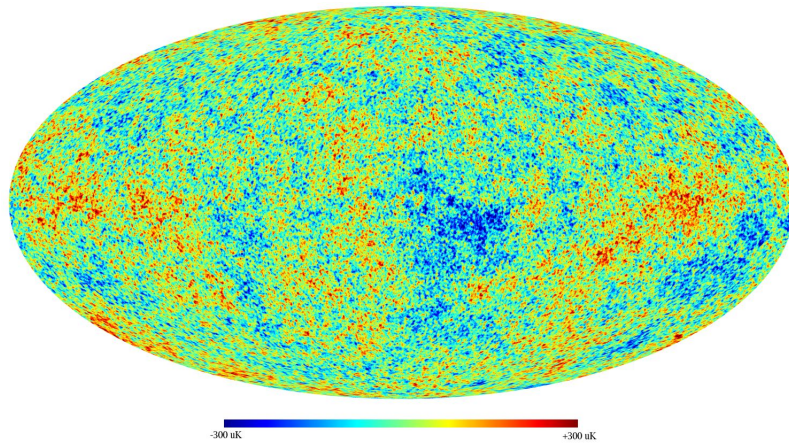


Figure 1.1: The Cosmic Microwave Background, image courtesy of wikipedia.

tools we have since on cosmological scales the only acting force is gravity. The weak nuclear force acts on distances less than 18^{-18} m while the strong nuclear force acts on distances less than 10^{-15} m. The electromagnetic force and gravity are both long range forces, but on large scales the universe is electrically neutral, so there is no electromagnetic force at work. The evolution of the universe is thus dominated by gravity, the weakest of the forces. The current cosmological models are all based on Einsteins field equations and the solutions allow us to model the universe from the early big bang up until today and into the future.

Unsolved problems

Even though we have discovered a lot about the evolution of the universe in the last century, there still are much we do not know. A few of todays biggest unsolved problems are:

Dark matter. Dark matter is a type of matter that neither emits nor absorbs electromagnetic radiation and only interacts through its gravitational effect. It was introduced to explain the missing matter density of the universe. A small portion of Dark Matter is likely baryonic astronomical objects that emit no or close to no electromagnetic radiation. Examples of such objects are non-luminous gas or Massive Astrophysical Compact Halo Objects (MACHO) such as black holes, neutron stars, brown dwarfs and planets. The other possible type is non baryonic dark matter of which there exists 3 different types: Cold Dark Matter (CDM), Warm Dark Matter (WDM) and Hot Dark Matter (HDM) or a combination of these (Davis et al. 1985). Cold dark matter is the currently the most viable candidate, and the energy density of CDM in the universe is measured to be $\Omega_{CDM} = 0.227$.

Dark energy. Dark energy is an even bigger mystery than dark matter and it constitutes $\sim 73\%$ of the matter-energy density of the universe. The two leading theories are a cosmological constant and quintessence. The biggest difference between these is are that the cosmological constant is constant in time and space, while quintessence is dynamic and changes with time. Dark energy is thought to be very homogeneous,

not dense, not know to interact through any of the fundamental forces other than gravity and it has $\omega < -1/3$.

Cosmic inflation. We are not yet sure the phase in the history of the universe called inflation actually happened and if it did happen, what the mechanism behind it was.

In addition to these there is a plethora of other unanswered questions in cosmology, so there still is considerable work left to be done. One avenue being explored in the attempt to answer some of the above questions are modified gravity theories. Modified gravity changes some of the aspects of GR by adding new properties to the spacetime. Some of these models can explain several of the unexplained observed phenomena, but fall short when used to explain others and as such they need thorough testing. Inventing a new modified gravity model is the less challenging part of this process, testing the validity of the model afterwards is the more complicated part.

This is a field where computer simulations play an increasing part in the quest to understand the universe. One of the ways to test proposed modified gravity models is by looking at how they affect the evolution of large scale structures in the universe and how the predicted large scale structures compare to the observed large scale structures. The only way to get the large scale structure evolution is by performing N-body simulations. These are simulations which simulate how a dynamical system of particles behave and evolve under the influence of physical forces. A large number of particles are put into the simulation volume under the effect of gravitational forces and the evolution is computed. Once such a simulation have been performed for a specific modified gravity model, we want to compare the data from such a N-body simulation to the observed universe. This cannot be done directly from the output of a N-body simulation, such a simulation only outputs each particles position, mass, velocity and similar particle properties, and it is not possible to compare such data with the positions and velocity of observed galaxies in the universe.

Because of this we need to do a halo finding process which determine how galaxies are grouped together in halos and then generates relevant properties for these halos. This data can then be compared to observations. Up until this thesis GR have been used in the halo finding process, no one has taken into account the differences between standard GR and different modified gravity theories in their halo finders. The validity of other halo finders in the realms of modified gravity is therefore not know. This is what we would like to improve upon in this thesis, by implementing a halo finder, called MORPH, that is independent of the modified gravity model used in the N-body simulations. By doing this we get a halo finder that can properly be used to test the results from modified gravity n-body simulations to see if the model is correct. As such this thesis improves a small step in the process of determining if modified gravity models can be used to explain some of the unsolved problems in cosmology. We are also going to test the implemented halo finder to make sure it gives correct results in the standard GR regime when compared to other known halo finders, and take a closer look at how different methods behave in modified gravity regime.

Chapter 2

Modern Cosmology

The underlying principle of modern cosmology is the Copernican principle, which states that no place in the universe is special. More explicitly it states that on large scales the universe is isotropic and homogeneous. Isotropic means that there is no preferred direction, the universe looks the same no matter in which direction we look. That the universe is homogeneous means that it looks the same independently of our position. It is more precisely explained by using a co-moving observer which is an observer at rest and unaccelerated with respect to nearby matter. The universe is homogeneous if all co-moving observers see identical properties, and isotropic if all comoving observers see no preferred direction.

We need to introduce the concept of horizons, cosmologists work with two different types of horizons.

- *Event horizon.*

The event horizon is the boundary that divides all events into two groups. Those who have been, are, or at some time will be observed as opposed those that the observer never will be able to see.

- *Particle horizon.*

The particle horizon is the maximum distance that particles could have traveled during the age of the universe, it marks the boundary between the observable and unobservable regions. The particle horizon is important because it gives us the size of regions where causal physics can operate.

Before we try to figure out the evolution of the universe there are a few key concepts that is useful to know, the first being the curvature of the universe. We live on the surface of earth, which is a two dimensional curved surface. The universe can have a curvature in the same way, just expanded to more dimensions. The geometry of a curved space can be described by a constant k , called the curvature constant. There are three distinct shapes our universe can have, a positively curved universe, $k = +1$, a negatively curved universe, $k = -1$ and a flat universe, $k = 0$. Each of these have different properties, as seen in figure 2.1. If we have two lines that starts out as parallel to each other in a positively curved universe, the lines will meet at great distances and the angles of a triangle adds up to more than π . Such a universe is called a open universe. In a universe with negative curvature, two parallel lines are going to diverge, while the angles of a triangle adds up

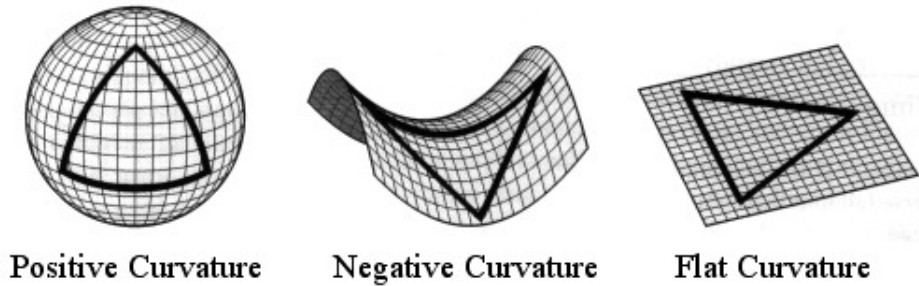


Figure 2.1: Difference between a positively curved, $k = +1$ universe, a negatively curved universe, $k = -1$ and a flat universe, $k = 0$.

to less than π , called a closed universe. Lastly, in a flat universe two parallel lines are going to remain parallel forever and the angles of a triangle adds up to exactly π . It is possible to imagine that combinations of these curvatures existed e.g. the universe has positive curvature in one place, while it is flat elsewhere. But this breaks the cosmological principle, giving a non-homogeneous universe, so we only need to take into account these three distinct curvatures and disregard any combinations between them.

One of the more basic properties of space is the distance between two different points, the line element. For a curved expanding space, which observations suggest is the case with our universe, the line element ds is given by the Friedmann-Lemaitre-Robertson-Walker (FLRW) metric (Bergstrom 2006),

$$ds^2 = c^2 dt^2 - a(t)^2 \left(\frac{dr^2}{1 - kr^2} + r^2 d\Omega^2 \right), \quad \text{with } d\Omega^2 = d\theta^2 + \sin^2 \theta d\phi^2. \quad (2.1)$$

(r, θ, ϕ) are comoving coordinates, which are coordinates fixed with respect to the expansion of the universe, and as such they follow the Hubble flow. Most large collections of matter, such as galaxies, are nearly comoving, their velocity through space is low so they only move due to expansion. The time coordinate t is the cosmic time, which is measured with a clock moving with the same rate as the expansion of the universe.

Much of this chapter have been inspired by Ryden (2002), Øystein Elgarøy (2010) and Dodelson (2004), as well as other various minor sources.

2.1 Cosmic dynamics, the Friedmann equations

As previously mentioned we would like to know the evolution of $a(t)$, which can be obtained from the Friedmann equations. This is a set of equations that govern the expansion of space in homogeneous and isotropic universe models. We start with the FLRW metric and solve Einstein's field equations for a curved universe with curvature k and a cosmological constant Λ . The first Friedmann equation is derived from the 00 component of Einstein's field equations, while the second is derived from the trace, since GR is needed

we simply state the two resulting equations here,

$$H^2(t) + \frac{kc^2}{a^2} = \frac{8\pi G\rho}{3} + \frac{\Lambda c^2}{3} \quad (2.2)$$

$$\frac{\ddot{a}}{a} = -\frac{4\pi G}{3} \left(\rho + \frac{3p}{c^2} \right) + \frac{\Lambda c^2}{3}. \quad (2.3)$$

Here ρ is matter and energy density, p is pressure and Λ is the cosmological constant.

2.1.1 Equation of state

We now want to solve the Friedman equations. The first equation we need is an equations of state, a relation between pressure, p , and matter/energy density, ρ . Fortunately we only need the simplest form of the equation of state possible, given as

$$p = \omega\rho c^2, \quad (2.4)$$

where ω is a constant that depends on the type of matter and c is the speed of light.

Dust, $\omega \approx 0$

Most of the matter in the universe is moving at non-relativistic speeds. In cosmology such matter is called dust, since on cosmological scales even the sun is just a small speck of matter. The equation of state for a ideal, non-relativistic gas is

$$p = \frac{Nk_B T}{V}, \quad (2.5)$$

which can be rewritten

$$p = \frac{k_B T}{mc^2} \rho c^2. \quad (2.6)$$

The relation between the mean square speed of the particles in a gas and temperature is

$$m\langle v^2 \rangle = 3k_B T, \quad (2.7)$$

when inserted into the equation above, it gives

$$p = \frac{\langle v^2 \rangle}{3c^2} \rho c^2. \quad (2.8)$$

So from this we see that dust has

$$\omega = \frac{\langle v^2 \rangle}{3c^2}. \quad (2.9)$$

Since this was a non-relativistic gas, with $v \ll c$ it is a good approximation to use $\omega \approx 0$, so dust is pressureless with an equation of state

$$p = 0. \quad (2.10)$$

Radiation, $\omega = 1/3$

For a gas of massless particles¹, such as photons the equation of state is

$$p = \frac{1}{3}\rho c^2, \quad (2.11)$$

and thus, $\omega = \frac{1}{3}$. Today the universe is filled with relic radiation, the CMB radiation, but its contribution to the current energy density is negligible, meaning the universe is independent of radiation at the present time. In the past however, when the universe was hotter and denser, radiation was the dominant component of the energy density. So while we can ignore the radiation density today, it dominated the evolution of the early universe.

Cosmological constant Λ , $\omega \approx -1$

After Albert Einstein formulated the theory of general relativity he applied it to the universe as a whole. He assumed a static, homogeneous, dust filled and isotropic universe. Such a universe gives $\dot{a} = \ddot{a} = 0$, $\Lambda = 0$ and $a(t) = a_0$ and the Friedmann equations become

$$\frac{kc^2}{a^2} = \frac{8\pi G\rho}{3}, \quad (2.12)$$

$$0 = -\frac{4\pi G}{3} \left(\rho + \frac{3p}{c^2} \right) a. \quad (2.13)$$

From the last equation we see that we must have either $a = a_0 = 0$ or $\rho = p = 0$. Neither of these options work well with our existence. The first corresponds to having no universe while the second corresponds to an empty universe. With this solution one should conclude that a static, dust filled universe is impossible, but all observations at that time pointed to the universe being static, so to make a static solution possible, Einstein introduced the cosmological constant, Λ , as a repulsive parameter. In his way of thinking this corresponds to assigning a curvature to empty space. The second Friedmann equation in such a universe, with $a(t) = a_0 = \text{constant}$ gives

$$\Lambda = 4\pi G\rho_0. \quad (2.14)$$

Inserting this into the first Friedmann equation we observe that the curvature must be positive, since the right hand side of the equation is positive, giving

$$a_0 = \frac{c}{\sqrt{4\pi G\rho_0}} = \frac{c}{\Lambda}. \quad (2.15)$$

This is called an Einstein universe. Einstein himself was never pleased with having to introduce this constant, and referred to it as the "Biggest blunder of his life" (Gamow 1970). One of the problems with this method, that force a static universe, is that it is unstable. Any small perturbation away from a_0 makes it so either the Λ -term or the dust-term will dominate. Today the understanding of the cosmological constant has

¹Neutrinos are treated as radiation, they have a mass though it is so small we approximate them as massless.

changed from Einstein's view, and the cosmological constant is interpreted as a result of Heisenberg's uncertainty principle. Empty space is not empty, but have an energy density due to quantum mechanical processes. Writing the Friedmann equations correspondingly gives

$$H^2(t) + \frac{kc^2}{a^2} = \frac{8\pi G}{3}(\rho + \rho_\Lambda), \quad (2.16)$$

$$\frac{\ddot{a}}{a} = -\frac{4\pi G}{3} \left(\rho + \rho_\Lambda + \frac{3p_\Lambda}{c^2} \right), \quad (2.17)$$

where

$$\rho_\Lambda = \frac{\Lambda}{8\pi G}. \quad (2.18)$$

We insert this relation into Friedmann's second equation, equation 2.17 and compare it with equation 2.2 for a dust filled universe, $p = 0$,

$$-\frac{4\pi G}{3} \left(\frac{\Lambda}{8\pi G} + \frac{3p_\Lambda}{c^2} \right) = \frac{\Lambda}{3}. \quad (2.19)$$

Giving

$$p = -\rho c^2. \quad (2.20)$$

So we see that the equation of state gives $\omega = -1$ for the cosmological constant and for $\Lambda > 0$ we have negative pressure.

Using a Newtonian point of view, the cosmological constant give rise to the repulsive contribution to the gravitational force. Once Edwin Hubble discovered that the universe was expanding (Hubble 1929), the cosmological constant dropped out of fashion, since it was possible to find expanding solutions of the Friedmann equations without it. Contrary to this, modern observational data indicates that we live in a universe with accelerated expansion, dominated by the cosmological constant.

2.1.2 Evolution of the energy density of the universe

Another thing we need to know is how the energy density of the universe behaves. The first law of thermodynamics is valid in a GR regime, so we have

$$TdS = dE + pdV, \quad (2.21)$$

where T is temperature, S is entropy, p is pressure, E is energy and V is volume. When we apply this to the universe we see that the volume must be proportional to a^3 , $V \propto a^3$. The energy of the universe is given by the energy density multiplied by the volume, $\rho c^2 V \propto \rho c^2 a^3$. The first law of thermodynamics now become

$$dS = dEpdV, \quad (2.22)$$

$$\propto d(\rho c^2 a^3) + p(da^3), \quad (2.23)$$

$$= 3a^2 \dot{a} \rho c^2 + a^3 \dot{\rho} c^2 + 3pa^2 \dot{a}, \quad (2.24)$$

$$= a^3 c^2 \left(\dot{\rho} + 3 \frac{\dot{a}}{a} \left(\rho + \frac{p}{c^2} \right) \right). \quad (2.25)$$

The universe is expanding adiabatically, a diabatic expansion would mean that heat would flow in or out of a given volume, violating homogeneity and isotropy. It is also hard to imagine what would happen if heat were leaving the universe, which violates energy conservation. An adiabatic expansion has $dS = 0$, which means we have,

$$\dot{\rho} = -3\frac{\dot{a}}{a} \left(\rho + \frac{p}{c^2} \right). \quad (2.26)$$

Inserting the equation of state $p = \omega\rho^2$ gives,

$$\dot{\rho} = -3\frac{\dot{a}}{a} \left(\rho + \frac{\omega\rho^2}{c^2} \right), \quad (2.27)$$

$$= -3\frac{\dot{a}}{a} (1 + \omega) \rho. \quad (2.28)$$

This gives us the following differential equation,

$$\frac{d\rho}{dt} = -3\frac{da}{dt} \frac{1}{a} (1 + \omega) \rho, \quad (2.29)$$

$$d\rho = -3\frac{1}{a} (1 + \omega) \rho da. \quad (2.30)$$

We integrate both sides from today, t_0 until t . Using $\rho(t_0) = \rho_0$ and $a(t_0) = a_0$,

$$\int_{\rho_0}^{\rho} \frac{1}{\rho} d\rho = -3(1 + \omega) \int_{a_0}^a \frac{1}{a'} da', \quad (2.31)$$

$$\ln \left(\frac{\rho}{\rho_0} \right) = -3(1 + \omega) \ln \left(\frac{a}{a_0} \right), \quad (2.32)$$

$$\rho = \rho_0 \left(\frac{a_0}{a} \right)^{3(1+\omega)}. \quad (2.33)$$

We now have the general equation for how the energy density evolves and we will take a look at the different possibilities we have for ω . In a dust filled universe, $\omega = 0$, and

$$\rho = \rho_0 \left(\frac{a_0}{a} \right)^3. \quad (2.34)$$

This is expected, the density decreases inversely proportional to the volume, which is proportional with a^3 . Then for radiation, with $\omega = 1/3$ we get,

$$\rho = \rho_0 \left(\frac{a_0}{a} \right)^4. \quad (2.35)$$

Again this is what we would expect, the energy density decreases as the universe expands, which is inversely proportional to a^3 . In addition the energy of each wave decreases as the wave gets redshifted due to the expansion, which is the cause of the last factor of $1/a$. For the cosmological constant, $\omega = -1$, it gets more interesting, since

$$\rho = \rho_0. \quad (2.36)$$

The density of the cosmological constant is constant in time, it does not decrease as the universe expands. This fits with the cosmological constant being tied to the vacuum energy, when more space is created we also get more vacuum energy, thus keeping the energy density constant.

There are several components of the total energy density of our universe. Fortunately for us they are additive, which means the total energy density can be written

$$\rho_{\text{total}} = \sum_i \dot{\rho}_i = -3\frac{\dot{a}}{a} \sum_i \left(\rho_i + \frac{p_i}{c^2} \right), \quad (2.37)$$

$$\sum_i \left[\dot{\rho}_i + 3\frac{\dot{a}}{a} \sum_i \left(\rho_i + \frac{p_i}{c^2} \right) \right] = 0. \quad (2.38)$$

The only way to guarantee that this sum becomes zero is for the individual terms to cancel out,

$$\dot{\rho}_i + 3\frac{\dot{a}}{a} \sum_i \left(\rho_i + \frac{p_i}{c^2} \right) = 0. \quad (2.39)$$

From this we see that the energy densities are independent of each other, and it is possible to solve for each separately and then insert the results into the Friedmann equations.

It is normal to measure the density in terms of the density parameter Ω . It is defined as the ratio of the actual density ρ to the critical density of the universe, ρ_c .

$$\Omega \equiv \frac{\rho}{\rho_c}. \quad (2.40)$$

The critical density gives the mass density of a universe where the expansion stops right after infinite cosmic time has elapsed. We find an expression for the critical density by assuming no cosmological constant, $\Lambda = 0$ in a flat universe, $k = 0$. This gives us

$$H^2 = \frac{8\pi G\rho}{3}. \quad (2.41)$$

Today this become

$$H_0^2 = \frac{8\pi G\rho_0}{3}, \quad (2.42)$$

$$1 = \frac{8\pi G\rho_0}{3H_0^2}. \quad (2.43)$$

We see that $\frac{3H_0^2}{8\pi G}$ must have units of density and it is named the critical density, ρ_c ,

$$\rho_{c0} \equiv \frac{3H_0^2}{8\pi G}, \quad (2.44)$$

2.2 The flat Λ CDM universe

Observations indicate that we live in a universe filled with dust, in the form of dark matter, and a cosmological constant, in the form of dark energy. We want to use our newfound knowledge on such a universe model, to see how it behaves.

2.2.1 Friedmann equations for a Λ CDM universe

In such a universe we get $\Omega_{\Lambda 0} = 1 - \Omega_{m0}$ and the Friedmann equations can be solved analytically. We rewrite them using the density parameter,

$$\frac{H^2}{H_0^2} = \Omega_{m0} \left(\frac{a_0}{a}\right)^3 + (1 - \Omega_{m0}). \quad (2.45)$$

The $\Omega_{m0} \left(\frac{a_0}{a}\right)^3$ term is the contribution from matter and is always positive and the $(1 - \Omega_{m0})$ term is the contribution from the cosmological constant. There exist two different solutions for this universe, the first one being $\Omega_{m0} > 1$ giving $\Omega_{\Lambda 0} < 0$. Then the right hand side of the equation changes sign at a_{\max} and starts to contract. This value of a_{\max} is found by

$$\Omega_{m0} \left(\frac{a_0}{a_{\max}}\right) = \Omega_{m0} - 1, \quad (2.46)$$

$$a_{\max} = a_0 \left(\frac{\Omega_{m0}}{\Omega_{m0} - 1}\right)^{1/3}. \quad (2.47)$$

Rewriting and solving the Friedmann equations gives us the solution

$$H_0 t = \frac{2}{3\sqrt{\Omega_{m0} - 1}} \sin^{-1} \left[\left(\frac{a}{a_{\max}}\right)^{3/2} \right]. \quad (2.48)$$

So this universe will collapse in a Big Crunch after a time

$$t_{\text{crunch}} = \frac{2\pi}{3H_0} \frac{1}{\sqrt{\Omega_{m0} - 1}}. \quad (2.49)$$

If we solve for the scalefactor a instead of t we get

$$a = a_0 \left(\frac{\Omega_{m0}}{\Omega_{m0} - 1}\right)^{1/3} \left[\sin \left(\frac{3}{2} \sqrt{\Omega_{m0} - 1} H_0 t \right) \right]^{2/3}. \quad (2.50)$$

This case corresponds to the universe containing enough matter to overcome the repulsive force from the cosmological constant, so gravity wins and the universe undergoes a Big Crunch at time t_{crunch} .

The other solution comes about if we have $\Omega_{m0} < 1$ giving $\Omega_{\Lambda 0} > 0$. Now the right hand side will always be positive, and the universe will keep on expanding. For this case there exist a value of the scale factor where the contribution from matter becomes equal to the contribution from the cosmological constant. This value is found by

$$\Omega_{m0} \left(\frac{a_0}{a_{m\Lambda}}\right) = 1 - \Omega_{m0} \quad (2.51)$$

$$a_{m\Lambda} = a_0 \left(\frac{\Omega_{m0}}{1 - \Omega_{m0}}\right)^{1/3}. \quad (2.52)$$

For $a < a_{m\Lambda}$ matter dominates while for $a > a_{m\Lambda}$ the cosmological constant dominates. Integrating the Friedmann equation gives

$$H_0 t = \frac{2}{3\sqrt{1 - \Omega_{m0}}} \sinh^{-1} \left[\left(\frac{a}{a_{m\Lambda}}\right)^{3/2} \right]. \quad (2.53)$$

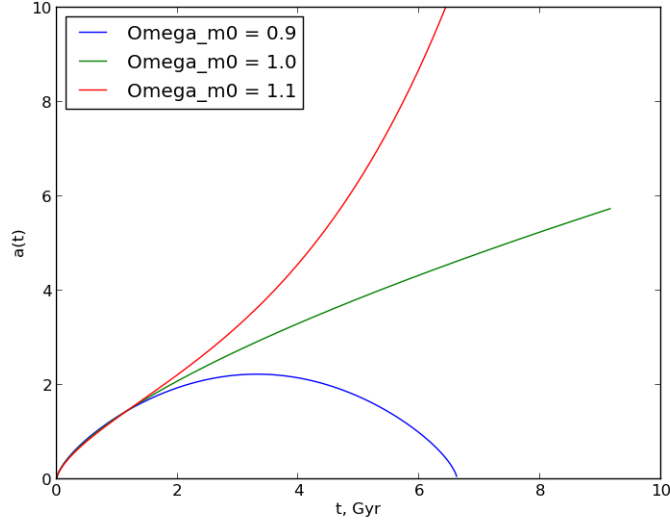


Figure 2.2: Evolution of the scalefactor for universes with $\Omega_{m0} = [0.9, 1.0, 1.1]$

We can solve this for the scalefactor a to get

$$a = a_0 \left(\frac{\Omega_{m0}}{1 - \Omega_{m0}} \right)^{1/3} \left[\sinh \left(\frac{3}{2} \sqrt{1 - \Omega_{m0}} H_0 t \right) \right]^{2/3}. \quad (2.54)$$

When we insert $a = a_0$ into this equation we find the present age of the universe

$$t_0 = \frac{2}{3H_0\sqrt{1 - \Omega_{m0}}} \sinh^{-1} \left(\sqrt{\frac{1 - \Omega_{m0}}{\Omega_{m0}}} \right). \quad (2.55)$$

Using $\Omega_{m0} = 0.3$ and $\Omega_{\Lambda 0} = 0.7$ we get $t_0 = 13.5$ Gyr, which is consistent with the oldest observed objects in the universe. The change from deceleration to acceleration happens at

$$a_{\text{acc}} = a_0 \frac{1}{2} \left(\frac{\Omega_{m0}}{1 - \Omega_{m0}} \right)^{1/3} = \left(\frac{1}{2} \right)^{1/3} a_{m\Lambda}. \quad (2.56)$$

Figure 2.2 is a plot of how the scale factor evolves for different values of Ω_{m0} . One interesting feature of such a universe is the accelerated expansion we get when $\Omega_{m0} = 0.9$ and $\Omega_{\Lambda 0} = 0.1$. A positive cosmological constant is effectively a gravitational repulsion and gives rise to an accelerated expansion of the universe. This means the universe will end in a Big Rip, when the expansion becomes so rapid that everything will be ripped apart into elementary particles. When we have $\Omega_{m0} = 1.1$ and $\Omega_{\Lambda 0} = -0.1$ gravity dominates and the universe will contract and end in a Big Crunch as the scale factor goes towards zero. We also see that a flat universe with $\Omega_{m0} = 1$ and $\Omega_{\Lambda 0} = 0$ will have a continuous expansion with $\ddot{a} = 0$.

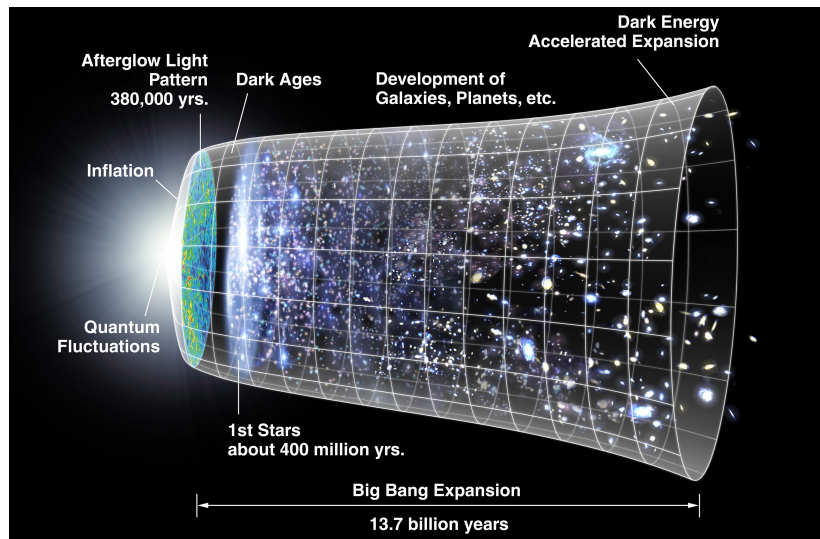


Figure 2.3: Evolution of the universe, image courtesy of wikipedia.

2.3 A shorter history of nearly everything

The Big Bang theory is the prevailing scientific model of how the universe came into being and evolved over time into the shape it has today. We assume it started from a singularity at $t = 0$. In the first time after the big bang, the Planck area, $t = [0, 5 \cdot 10^{-44}]$, the universe was filled with highly relativistic elementary particles, such as quarks, leptons and gauge bosons. What actually happened up until $t = 5 \cdot 10^{-44}$ are purely speculations as our theories are incomplete. Figure 2.3 shows the general evolution of the universe.

2.3.1 Inflation

The Big Bang theory is highly successful, and explains many of the features we observe in the universe. But there are some puzzles that are harder to solve, the three most important being the flatness problem, the horizon problem and the magnetic monopole problem.

The flatness problem

Today the energy density is very close to the critical density, giving us a flat universe with $k = 0$. If we extrapolate backwards in time it means that the energy density close to the big bang must have been extremely fine tuned, the density must have been closer to the critical density than one part in 10^{60} .

The horizon problem

Another even more puzzling problem is the horizon problem. The CMB is isotropic around 2.7K, to a part of one in 10^5 across the whole sky, meaning two different regions of the universe far into the past have been in thermal equilibrium. But the problem arises when we take into account the particle horizon of the photons. At the time of last scattering,

which is the last time the photons could have been in contact, the particle horizon for a photon was $\theta_{PH} \sim 1.8$ degrees. So how is it possible that photons that are separated by as much as 180 degrees have almost exactly the same temperature when these two areas are causally disconnected?

The magnetic monopole problem

The monopole problem is the lack of observed magnetic monopoles. This is not a purely cosmological problem, but arises from the combination of the big bang theory and Grand Unified Theory (GUT) that combines the strong and electroweak force at high temperatures. When the universe cools down it goes through a phase transition where it loses symmetry and the physical forces change behavior. This should give rise to point like topological defects that would act as magnetic monopoles, but there is no evidence that magnetic monopoles exist today. This last problem is not as serious as the first two. Today there does not exist a single definitive Grand Unified Theory, and in some variants magnetic monopoles do not exist. So it might very well be that the problem is in the GUT and not the Big Bang theory.

Inflation

It is now the idea of inflation comes to the rescue. Alan Guth published the idea of inflation in 1981 (Guth 1981), which solves these three problems that otherwise have no satisfactory solution. Inflation is a short period, from $t = 10^{-43}s$ to $t = 10^{-33}s$, right after big bang where the universe underwent a rapid expansion. The universe increased in size by a factor of

$$\frac{a(t_f)}{a(t_i)} = e^N, \quad (2.57)$$

where $a(t_i)$ is the time inflation starts, t_f the time inflation has ended and N is the e-folding number, given as

$$N = H_i(t_f - t_i). \quad (2.58)$$

It depends on the specific model, but it is generally assumed that $N \sim 60$. A note, even though this inflationary epoch seems to contradict that nothing can move faster than the speed of light there is an important difference. It was space itself that expanded, and not the particles in the universe moving away from each other. Thus the particles did not move through space faster than light.

If the universe expands exponentially during the inflationary epoch, the difference between the energy density and (1.) would decrease. If we start out with a curved space, but expand it to a large size, the initial curvature can become infinitesimally small. The horizon problem is solved by the fact that a small area, smaller than the particle horizon, can have been in thermal equilibrium before inflation started. Inflation then expands this area to a size larger than the observable universe, meaning that photons from different sides of the observable universe were causally connected far in the past. The monopole problem is easily solved, if magnetic monopoles were created before or during the inflationary period, the magnetic monopole density might have been reduced to undetectable levels during the inflation. So they exist, but because of inflation they have been dispersed so much we are unable to detect them.

Such an expansion happens in a de Sitter universe, which is a flat universe dominated by the cosmological constant, Λ , where we have something that gives negative pressure, but even though a de Sitter universe provides us with the desired expansion, it has no mechanism that ends the exponential growth. So unfortunately we do need a more complicated model. The cause of inflation is not yet well understood, the most common theory is that the inflation was driven by a scalar field. But one of the largest problem with this is that there exists no know scalar field that can drive inflation. The main thing we should know about a scalar field is that it has a kinetic and a potential energy associated with it, and thus an energy density and pressure. Let us take look at a homogeneous scalar field ϕ , which makes it a function of time only. The energy density is given by

$$\rho_\phi c^2 = \frac{1}{2\hbar c^3} \dot{\phi}^2 + V(\phi), \quad (2.59)$$

and the pressure is given by

$$p_\phi = \frac{1}{2\hbar c^3} \dot{\phi}^2 - V(\phi), \quad (2.60)$$

where $V(\phi)$ is the potential energy of the scalar field. If the field changes slowly with time, such that it has more potential energy than kinetic, we have

$$\frac{\dot{\phi}^2}{2\hbar c^3} \ll V(\phi), \quad (2.61)$$

the equation of state becomes

$$p_\phi = \rho_\phi c^2, \quad (2.62)$$

and it will behave similar to the cosmological constant in the de Sitter universe. It is this fact that is the basis for using a scalar field to drive inflation, as long as the homogeneous scalar field has much the same dynamics as a single particle moving in a potential. Similar to a ball being released from far up the side of a hill, the scalar field will "roll" down the potential until it reaches minimum. If it rolls slowly enough the potential energy can be treated as a constant most of the way down, this is the slow roll approximation. As long the scalar field has not reached the minima, the potential energy is greater than the kinetic, and we will have exponential expansion of the universe. Once the slow roll approximation has broken down, the scalar field will start oscillating around the minima. There is a friction term associated with the scalar field, which means the oscillations will be dampened and lose energy to the surroundings. The energy stored in the field will because of this friction go into creating "normal" particles, a process called reheating. After the reheating the universe enters the radiation dominated era and evolution proceeds as usual. Heisenberg's uncertainty relation limits how homogeneous the scalar field can be. As a consequence of this inflation starts and ends at slightly different times in different regions of space, causing perturbations in the energy density that is the basis for the large scale structure of the universe.

2.3.2 The Big Bang Nucleosynthesis

Shortly after inflation ended, at around $t \approx 1$ s, the Big Bang Nucleosynthesis (BBN) started. Previously when the universe was much hotter and denser, neutral atoms and

bound nuclei did not exist. In the early universe free neutrons combined with protons to form deuterium, but the resulting nuclei was immediately broken up by high energy photons, with a energy equal or greater than the deuterium's binding energy of 2.22 MeV. Due to this deuterium production can not begin before there are less photons with a energy above or equal 2.22 MeV than baryons. And since there are roughly 10^9 times as more photons than baryons, the average photon energy must be much lower than 2.22 MeV. Because of this there is no deuterium production before the temperature falls bellow 0.1 MeV. Almost no elements heavier than helium was formed in the BBN, with the exception of small traces of lithium. This is because reaction rates for the processes necessary to produce heavier elements is too low at temperatures below 0.1 MeV.

2.3.3 Recombination

Recombination² is the epoch where electrons bind with protons and neutrons without being instantly excited. This happens when the temperature is so low that the highest energy photons no longer immediately ionizes the atoms, when the temperature has dropped to 1 eV. This temperature is again lower than the binding energy of neutral hydrogen, 13.6 eV, for the same reasons as the delay in the start of the BBN. Before recombination photons could not move freely over long distances, matter and radiation were tightly coupled through the Compton and Coulomb scattering, so it is in this era that the universe became transparent, the mean-free-path of the photons became equal to the size of the observable universe. The photons from this time is what we observe as the CMB radiation. These photons originally had a temperature of $\sim 3000K$, but due to the expansion of the universe the temperature has dropped to $2.7K$ at the present time. The CMB radiation is a valuable tool as it gives us a look at the universe when it was only 300 000 years old, and it is one of our best sources of information of the early universe.

2.3.4 Re-ionization

Following recombination the universe entered a calm era and did not undergo any major changes for quite some time, it continued to expand and cool down. Clouds of gas interacted through gravity and started to cluster together due to small perturbations from the inflationary epoch. After some time these gas clouds became so dense that they collapsed due to gravity, leading to a massive increase in temperature. When the pressure and temperature became high enough, new nuclear reactions started and the first stars in the universe was born. These stars started to send out photons with high enough energy to ionize the rest of the gas in the universe, returning it to its previous ionized state, but at this time the gas was so diluted, compared to earlier stages, that the universe still remained transparent.

2.3.5 Structure formation

If the universe was completely homogeneous after the big bang it would have stayed so forever. Looking at the universe today we see that this is not the case, so there

²Even though it is the first time electrons and nuclei combined to form neutral atoms, this era is called recombination.

must have been perturbations in the original density. Inflation seems to be one way to introduce fluctuations in the primordial energy density. From here on the idea behind what happens is simple, where there are overdensities in the matter density, more matter gathers, attracting even more matter, creating larger and larger structures. But even though the idea is simple, the calculations are complicated, as they should be done in an expanding universe with several different components on a multitude of objects. Due to this N-body simulations are often used to understand the dynamics behind the structure formation. Observations indicate that the universe is largely composed of voids, with densities as low as 1/10 of the average density while the matter condenses into large filaments with a web like structure of galaxies, galaxy groups, clusters and super clusters, which confirms the results from N-body simulations.

Chapter 3

Gravity

3.1 Introduction

The theory of general relativity was developed by Albert Einstein and published in 1916 (Einstein 1916). GR combines space and time into one entity, called spacetime. Gravity is a geometric property of spacetime, where spacetime curves due to the effects of matter and energy density. The relation between the two is given by Einstein's field equations, which is a system of differential equations. It is important to note that in general relativity gravity is no longer considered a force, but simply the effect of the curvature of the four dimensional spacetime. General relativity is the basis for modern cosmology as none of the other forces work on cosmological distances, as mentioned previously, so to be able to understand how the universe evolves we need to know the basics of general relativity.

Much of this chapter have been inspired by Øvind Grøn (2007), Capozziello et al. (2009), De Felice & Tsujikawa (2010), Hammami (2013), (Clifton et al. 2012) and (Sotiriou & Faraoni 2010).

3.1.1 Equivalence principle

The basis for general relativity is the equivalence principle (Einstein 1908). It states that:

- You cannot perform any experiment that will distinguishing between a uniform gravitational field and being in a uniformly accelerated reference frame.

This is easily explained by a thought experiment, consider an observer in an elevator placed at the surface of Earth. The elevator is in every way sealed from its surroundings. We drop different objects and measure how much time it takes for each to reach the floor and their acceleration is measured to be g . The elevator is then moved into space, far away from the influence of any gravitational sources and is given a constant acceleration equal to g in the upwards direction. If we do the same experiment we will get the exact same measurements. The objects will in both cases fall to the floor with an acceleration equal to g and there is no way we as observers can distinguish between the two cases. An additional effect of the equivalence principle is that light will bend in a gravitational field.

3.1.2 Preliminaries

To be able to do calculations in GR there are several useful relations and conventions we need to know that are of a purely mathematical nature and we will introduce a few of them now.

Conventions

In general relativity the norm is to use Einstein's summation convention, which is useful when working on vectors, matrices and tensors. Instead of writing the summation sign explicitly, it is assumed when we have the same index in the upper and lower subscript. So

$$y = \sum_{\mu=0}^3 g_{\mu} x^{\mu} = g_0 x^0 + g_1 x^1 + g_2 x^2 + g_3 x^3, \quad (3.1)$$

is reduced to

$$y = g_{\mu} x^{\mu}. \quad (3.2)$$

Time is the 0 component and 1, 2 and 3 the spatial components. Another convention is that the Greek alphabet is used for space and time components, where the indices takes the values 0, 1, 2, 3 and the Latin alphabet is used for spatial components only, with values 1, 2, 3.

Friedman-Lemaitre-Robertson-Walker metric

The metric $g_{\mu\nu}$ is a symmetric tensor that is very useful in GR and cosmology. It is non-degenerate, meaning its determinant is not equal zero, $g = |g_{\mu\nu}| \neq 0$, and an inverse metric $g^{\mu\nu}$ exists

$$g^{\mu\nu} g_{\nu\sigma} = \delta_{\sigma}^{\mu}, \quad (3.3)$$

where δ_{σ}^{μ} is the Kronecker delta, defined as

$$\delta_{\sigma}^{\mu} \equiv \begin{cases} 0, & \text{if } \mu \neq \sigma \\ 1, & \text{if } \mu = \sigma \end{cases}. \quad (3.4)$$

The metric gives the connection between values in the coordinate system and the line element ds^2 . The Friedman-Robertson-Walker metric, mentioned in the previous chapter is

$$g_{\mu\nu} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & a^2(t) & 0 & 0 \\ 0 & 0 & a^2(t) & 0 \\ 0 & 0 & 0 & a^2(t) \end{bmatrix}. \quad (3.5)$$

$g^{\mu\nu}$ is the inverse of $g_{\mu\nu}$,

$$g^{\mu\nu} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1/a^2(t) & 0 & 0 \\ 0 & 0 & 1/a^2(t) & 0 \\ 0 & 0 & 0 & 1/a^2(t) \end{bmatrix}. \quad (3.6)$$

In four dimensional space-time, using Einstein's summation notation, the line element is given as

$$ds^2 = g_{\mu\nu} dx^\mu dx^\nu. \quad (3.7)$$

Here dx^μ is an infinitesimal displacement in the direction x^μ .

3.2 Einstein equations

The defining equations of GR, and the basis for modern cosmology is the Einstein field equations (Einstein 1916), which relates the geometry of space-time to the energy-momentum content of space. The equation is given as

$$G_{\mu\nu} \equiv R_{\mu\nu} - \frac{1}{2}g_{\mu\nu}R = \kappa^2 T_{\mu\nu}. \quad (3.8)$$

$G_{\mu\nu}$ is the Einstein tensor and $\kappa^2 = 8\pi G$ is a constant only. The left hand side of this equation is a function of the geometry (metric) of the universe while the right hand side is a function of the energy it contains. This equation might seem simple, but when fully written out it is a system of 10 coupled, nonlinear, hyperbolic-elliptic partial differential equations.

Ricci tensor, $R_{\mu\nu}$, and Ricci scalar, R

The Ricci tensor, $R_{\mu\nu}$, depends on the metric and its derivatives. It is, expressed in terms of Christoffel symbols,

$$R_{\alpha\beta} = \Gamma_{\mu\nu,\alpha}^\alpha - \Gamma_{\mu\nu,\nu}^\alpha + \Gamma_{\beta\alpha}^\alpha \Gamma_{\mu\nu}^\beta - \Gamma_{\beta\nu}^\alpha \Gamma_{\nu\alpha}^\beta. \quad (3.9)$$

If we contract the Ricci tensor we get the Ricci scalar, as seen in (3.8),

$$R = R^\mu_\mu = g^{\mu\nu} R_{\mu\nu}. \quad (3.10)$$

Christoffel symbol Γ

One of the most useful relations in GR is the Christoffel symbol, Γ ,

$$\Gamma_{\alpha\beta}^\mu = \frac{g^{\mu\nu}}{2} \left(\frac{\partial g_{\alpha\nu}}{\partial x^\beta} + \frac{\partial g_{\beta\nu}}{\partial x^\alpha} + \frac{\partial g_{\alpha\beta}}{\partial x^\nu} \right). \quad (3.11)$$

In GR commas in a subscript is used to symbolize the derivative with respect to x , for example

$$\frac{\partial g_{\mu\nu}}{\partial x^\alpha} \equiv g_{\mu\nu,\alpha}. \quad (3.12)$$

We use this to write the Christoffel symbol as

$$\Gamma_{\alpha\beta}^\mu = \frac{g^{\mu\nu}}{2} (g_{\alpha\nu,\beta} + g_{\beta\nu,\alpha} + g_{\alpha\beta,\nu}) \quad (3.13)$$

Another important derivative is the covariant derivative, denoted by a ";" and defined as

$$x^\mu_{;\nu} = x^\mu_{,\nu} + \Gamma_{\alpha\nu}^\mu x^\alpha. \quad (3.14)$$

The stress-energy tensor $T^{\mu\nu}$

The stress-energy tensor $T^{\mu\nu}$ is a symmetric rank 2 tensor that describes material characteristics. It characterizes the matter and energy content of the universe and is thus an important part of the Einstein equations. Since we cannot derive the matter and energy content of the universe, we need observations to get the content of the stress-energy tensor. Because this is a rank 2 tensor it can be displayed in matrix form,

$$T^{\mu\nu} = \begin{bmatrix} T^{00} & T^{01} & T^{02} & T^{03} \\ T^{10} & T^{11} & T^{12} & T^{13} \\ T^{20} & T^{12} & T^{22} & T^{23} \\ T^{30} & T^{31} & T^{32} & T^{33} \end{bmatrix}, \quad (3.15)$$

and the different components are:

- T^{00} represents energy density
- T^{i0} represents momentum density
- T^{ii} represents normal stress
- T^{ij} represents shear forces ($i \neq j$)

As we have seen previously, perfect fluids are used to great effect in cosmology, the energy-momentum tensor of a perfect fluid is

$$T_{\mu\nu} = \left(p + \frac{p}{c^2}\right) \mu_\mu \mu_\nu + p g_{\mu\nu}. \quad (3.16)$$

In a comoving orthonormal basis the four velocity components are $\mu^\mu = (c, 0, 0, 0)$ and the stress-energy tensor for a perfect fluid can be written as

$$T_{\hat{\mu}\hat{\nu}} = \begin{bmatrix} \rho c^2 & 0 & 0 & 0 \\ 0 & p & 0 & 0 \\ 0 & 0 & p & 0 \\ 0 & 0 & 0 & p \end{bmatrix}, \quad (3.17)$$

where $p > 0$ is pressure and $p < 0$ is tension. There are three different types of perfect fluids that is of use in cosmology,

1. Dust or non-relativistic gases, given by $p = 0$ and $T_{\mu\nu} = \rho \mu_\mu \mu_\nu + p g_{\mu\nu}$.
2. Radiation or relativistic gases, $p = \frac{1}{3} \rho c^2$.
3. Vacuum energy, if it is a perfect fluid we have $p = -\rho c^2$.

3.2.1 Lagrange formalism of general relativity

General relativity can be seen as a classical field theory, it is therefore possible, as in other field theories¹, to derive the Einstein equations from a given action, S ,

$$S = S_{EH} + S_M, \quad (3.18)$$

where S_{EH} is the Einstein-Hilbert action and S_M is the action for the matter field.

$$S_{EH} = \frac{1}{2\kappa^2} \int (R - 2\Lambda)\sqrt{-g}d^4x, \quad (3.19)$$

$$S_M = \int \mathcal{L}\sqrt{-g}d^4x, \quad (3.20)$$

where R is the Ricci scalar, Λ the cosmological constant and \mathcal{L} the Lagrangian density for matter and energy. Following the standard algorithm we apply the principle of least action,

$$\delta S = \delta S_{EH} + \delta S_M = 0. \quad (3.21)$$

So we have

$$\delta S_{EH} = -\delta S_M. \quad (3.22)$$

We solve for each side separately, first for the Einstein-Hilbert part δS_{EH} , then for the matter Lagrangian part δS_M .

$$\delta S_{EH} = \frac{1}{2\kappa^2} \int \sqrt{-g}\delta R + (R - 2\Lambda)\delta\sqrt{-g}d^4x, \quad (3.23)$$

since $\delta\Lambda = 0$. We now need to find $\delta\sqrt{-g}$ and δR . Lets start with $\delta\sqrt{-g}$. Jacobi's formula for differentiating a determinant gives us,

$$\begin{aligned} \delta g &= \delta \det(g_{\mu\nu}), \\ &= g g^{\mu\nu} \delta g_{\mu\nu}. \end{aligned}$$

Using this we get

$$\delta\sqrt{-g} = \frac{1}{2\sqrt{-g}}\delta g, \quad (3.24)$$

$$= \frac{1}{2}\sqrt{-g}(g^{\mu\nu}\delta g_{\mu\nu}). \quad (3.25)$$

We now use that

$$g_{\mu\nu}g^{\mu\nu} = g^{\mu\nu}g_{\mu\nu}, \quad (3.26)$$

to get

$$\delta\sqrt{-g} = -\frac{1}{2}\sqrt{-g}g_{\mu\nu}\delta g^{\mu\nu}. \quad (3.27)$$

¹An interesting observation, even though we can use variational calculus to derive the laws of physics, it is rare that this principle is used to get these laws initially. Most often the laws are first found by other means and then variational calculus is used to derive the same laws from a given action.

Now if we take a look at δR and use the definition of the Ricci scalar, equation 3.10, it gives

$$\delta R = R_{\mu\nu}\delta g^{\mu\nu} + g^{\mu\nu}\delta R_{\mu\nu}.$$

The first term is already proportional to $\delta g^{\mu\nu}$. Using Stoke's theorem on the second term we see that it vanish when integrated and does not contribute to the variation of the action. We put what we have found into the equation for δS_{EH} to get

$$\delta S_{EH} = \frac{1}{2\kappa^2} \int \sqrt{-g} R_{\mu\nu} \delta g^{\mu\nu} - (R - 2\Lambda) \frac{1}{2} \sqrt{-g} g_{\mu\nu} \delta g^{\mu\nu} d^4x, \quad (3.28)$$

$$\delta S_{EH} = \frac{1}{2\kappa^2} \int \sqrt{-g} \left(R_{\mu\nu} - (R - 2\Lambda) \frac{1}{2} g_{\mu\nu} \right) \delta g^{\mu\nu} d^4x. \quad (3.29)$$

We now take a look at the matter Lagrangian part δS_M ,

$$\delta S_M = \int \sqrt{-g} \delta \mathcal{L} + \mathcal{L} \delta \sqrt{-g} d^4x, \quad (3.30)$$

$$= \int \sqrt{-g} \left(\frac{\delta \mathcal{L}}{\delta g^{\mu\nu}} - \frac{1}{2} \mathcal{L} g_{\mu\nu} \right) \delta g^{\mu\nu} d^4x. \quad (3.31)$$

The stress tensor $T_{\mu\nu}$ is by definition given as,

$$T_{\mu\nu} = -2 \left(\frac{\delta \mathcal{L}}{\delta g^{\mu\nu}} - \frac{1}{2} \mathcal{L} g_{\mu\nu} \right). \quad (3.32)$$

Giving us

$$\delta S_{EH} = \int \sqrt{-g} \left(-\frac{1}{2} T_{\mu\nu} \right) \delta g^{\mu\nu} d^4x. \quad (3.33)$$

Going back to the principle of least action,

$$\delta S_{EH} = -\delta S_M, \quad (3.34)$$

$$\frac{1}{2\kappa^2} \int \sqrt{-g} \left(R_{\mu\nu} - (R - 2\Lambda) \frac{1}{2} g_{\mu\nu} \right) \delta g^{\mu\nu} d^4x = - \int \sqrt{-g} \left(-\frac{1}{2} T_{\mu\nu} \right) \delta g^{\mu\nu} d^4x. \quad (3.35)$$

Which leads to

$$\frac{1}{2\kappa^2} \left(R_{\mu\nu} - (R - 2\Lambda) \frac{1}{2} g_{\mu\nu} \right) = \frac{1}{2} T_{\mu\nu}. \quad (3.36)$$

And this gives us the Einstein equation,

$$R_{\mu\nu} - \frac{1}{2} g_{\mu\nu} R + \Lambda g_{\mu\nu} = \kappa^2 T_{\mu\nu}. \quad (3.37)$$

3.2.2 Geodesic equation

A geodesic describes the motion of a point particle under the influence of gravity alone, it gives the shortest path between two points in curved spacetime. The path taken by a falling rock, a satellite in orbit and the motion of a planet are all geodesics in curved spacetime. This can be found by writing the equation for the distance between two points in curved spacetime and minimizing this length using calculus of variation. For a geodesic curve we then have the following equation

$$u^\mu_{;\nu} u^\nu = 0. \quad (3.38)$$

This is called the geodesic equation, writing it out gives,

$$(u^\mu_{;\nu} + \Gamma^\mu_{\alpha\nu} u^\alpha) u^\nu = 0. \quad (3.39)$$

We now use

$$\frac{d}{d\lambda} = u^\nu \frac{\partial}{\partial x^\nu}. \quad (3.40)$$

and the notation $\dot{} = \frac{d}{d\lambda}$ to get,

$$\ddot{x}^\mu + \Gamma^\mu_{\alpha\nu} \dot{x}^\alpha \dot{x}^\nu = 0, \quad (3.41)$$

which is the better known version of the geodesic equation.

3.3 Modified Gravity

General relativity is a very successful model, and have passed all observational tests, so why would we want to modify something that works this well?

3.3.1 Motivation for modified gravity

The first attempts at modifying gravity was motivated by curiosity and in an effort to further understand the new theory. Today we see that there are several shortcomings with the standard Λ CDM model and therefore we try to modify gravity in the hope that it can explain/solve the shortcomings of the Λ CDM model.

Cosmological constant problem

One of the most serious shortcomings with the Λ CDM model is the cosmological constant problem². Vacuum fluctuations seems to exist in nature and to have normal gravitational properties and it can be argued that it is equal to the cosmological constant. The problem appear when we calculate this value using quantum field theory, we end up with a number heavily contradicting cosmological observations. The difference is extreme, performing the calculations we get a number that is 107 orders of magnitude higher than the upper bounds set by data from the voyager spacecraft, which is largest disagreement between observations and theoretical predictions in the entire history of physics (Dutra 2005).

²Similar to the ultraviolet catastrophe, this have been dubbed the vacuum catastrophe.

Postulations

The Λ CDM model comes at the price of several postulations, such as inflation, dark matter and dark energy. Together dark matter and dark energy amounts to around 96% of the matter-energy of the universe and as previously mentioned, very little is known about either. This means that our most successful model is mostly based on matter and energy that we have no clue to how behaves and of which we have very few direct observations. A model that is this highly dependent on two pieces of undiscovered physics is a serious fundamental problem. Dark energy is one problem that might be explained by modified gravity theories while the inflation of the universe is another, there also exist theories that try to explain both at the same time.

Coincidence problems

The present value of Λ has another problem associated with it, the energy density of Λ is of the same order of magnitude as the average matter density of the universe today, $\rho_{\Lambda 0} \sim \rho_{m0}$. Which leads to a coincidence problem, these two values scale with the size of the universe in very different ways, so to assume that they are similar today purely by coincidence is naive. Another coincidence problem also exists, namely why the dark matter energy density is so close to the baryon energy density. They have very different production mechanisms so once more it is naive to assume it is by chance.

Galaxy structure

We have another potential problem with missing satellites. The Λ CDM model predicts there to be numerous dwarf galaxies around the main galaxies, but to use the Milky way as an example, only 30 dwarf galaxies have been observed. It might be though, that the rest are dark galaxies consisting of dark matter and therefore have not been observed yet, but it is still a noteworthy potential problem.

Solution

Modified gravity might solve several of these problems. A hope is that modified gravity might explain how the universe works and evolves without having to introduce exotic particles such as dark energy or that we can explain inflation. But even if it turns out that the classical understanding of gravity is correct, by questioning its merits and introduce changes to see what happens, we gain an increased understanding of the theory.

3.3.2 Modifications of Einstein's equations

If we want to modify Einstein's equation there are two ways we can do it, either change the right hand side or the left hand side. By changing the right hand side we obtain

$$G_{\mu\nu} = \kappa^2(T_{\mu\nu} + T_{\mu\nu}^{DE}). \quad (3.42)$$

This is equivalent with adding another form of matter-density to our universe model, such as dark energy or dark matter. If we change the left hand side we obtain

$$G_{\mu\nu} + G_{\mu\nu}^{Mod} = \kappa^2(T_{\mu\nu}). \quad (3.43)$$

This is the equivalent of changing our understanding of the spacetime, without having to introduce new exotic particles to the universe.

We have to be careful when modifying GR, it is a highly successful theory and has so far passed all tests performed. This means that any modification of GR must keep intact the success of the original theory, and make sure we get the same results on the scales we have performed experiments and observations on. A "screening"-mechanism is often used to make sure this happens, this is any sort of mechanism that makes sure that gravity behaves as observed, it shields us from the changes in the modified general theory on local scales.

3.4 Alternative gravity theories

There exist many different types of modified gravity theories, but there are three general classes:

- Theories of Gravity with extra fields, such as Scalar-Tensor Theories
- Higher Derivative and Non-Local Theories of Gravity, for example $f(R)$ -theories
- Higher Dimensional Theories of Gravity, such as Kaluza-Klein Theories of Gravity

Each of these are again divided into a multitude of different subgroups of theories.

3.4.1 Theories of Gravity with Extra Fields

In general relativity the gravitational force is mediated through a rank 2 tensor field. While there are good reasons to couple matter fields to gravity in this way, there is nothing that explicitly prevents us from adding additional fields. The simplest form is the addition of a scalar field, but it is also possible to add other fields such as vector, tensor or higher rank fields.

Scalar-Tensor theories

Scalar-Tensor theories are some of the best established and well studied theories of modified gravity and they are very good examples of the methods we can use to modify GR. They are also of particular interest as the field equations they give are simple enough to allow analytical solutions to several interesting physical problems. If we reduce the number of dimensions in higher dimensional theories, such as Kaluza-Klein and string models, we can end up with scalar-tensor theories. One example of a Scalar-Tensor theory is the symmetron model, which will be covered in greater depth in section 3.6.

3.4.2 Higher Derivative and Non-Local Theories of Gravity

The field equations of GR has at the most only second order derivatives. One way to expand GR is then to permit higher order derivatives in the field equations. A problem with this way of modifying gravity is that they are susceptible to introducing instabilities into the theories. One such theory is $f(R)$ -gravity, which will be discussed further in section 3.5.

3.4.3 Higher Dimensional Theories of Gravity

Riemann, Cayley and Grassmann were some of the first to study higher dimensional geometry, called Riemann geometry. In the mid 19th century they developed tools to do calculations on this geometry. It is a central piece of general relativity as spacetime is based on Riemann geometry. Since Riemann geometry is not restricted to 3+1 dimensions, we have the tools needed to formulate GR in higher dimensions. Some serious problems appear by doing this, the most dire being that gravity does not behave as a > 3 dimensional force. The Newtonian potential in N -dimensional spacetime generally follows

$$\phi_N \propto \frac{1}{r^{N-3}}. \quad (3.44)$$

For $N \neq 4$, we cannot have stable planetary orbits and it is clear that gravity cannot appear as a > 3 dimensional force in our solar system. The extra dimensions added are therefore most often hidden in some way. In most theories it is spatial dimensions that are added, the addition of extra time dimensions have been studied, but extra time dimensions might lead to problems with causality as two time dimensions permit closed time-like curves in the form of circles in the plane.

Kaluza-Klein Theories of Gravity

Kaluza-Klein theory grew from an attempt to unify gravity and electrodynamics. The basic idea behind a Kaluza-Klein theory is to look at GR on a 4+1 dimensional manifold, a manifold with N dimensions is a topological space that close to each point resembles normal N -dimensional Euclidean space. One of the spatial dimensions is assumed to be small and compact. We can then perform a harmonic expansion of all fields along that one extra dimension, creating an effective 3+1 dimensional theory. This is the same idea used by 10 dimensional string theories and 11 dimensional M-theory. Each different method used to reduce the theory to four dimensions give a new four dimensional theory, so there is a plethora of different effective theories. But there are complex unanswered question with such a theory, for example why one of the spatial dimensions is much smaller than the others, and what mechanism is behind the expansion of the three extended dimensions?

3.5 $f(R)$ gravity

$f(R)$ gravity is one of the simpler forms of modified gravity and was first proposed by Hans Adolph Buchdahl in 1970 (Buchdahl 1970). These theories generalize the Einstein-Hilbert action by adding additional scalar curvature invariants, done by replacing the linear function of the Ricci scalar with a more general function, $f(R)$. It is a family of theories that is defined by this function of the Ricci scalar³. These theories have been studied extensively and are well known, they can lead to a period of accelerated expansion early in the history of the universe and can thus explain inflation. Recently $f(R)$ gravity have also been of interest as a possible explanation for the observed accelerated expansion of the Universe today. $f(R)$ theories behave in such a way that the mass of the scalar

³It is from here the name originates.

field becomes large near dense bodies, making the force short ranged, thus effectively suppressing the magnitude of the scalar force in denser regions, such as our solar system.

The actual modification from standard general relativity is made by generalizing the Lagrangian of the Einstein-Hilbert action,

$$S_{EH} = \frac{1}{2\kappa} \int \sqrt{-g} R d^4x. \quad (3.45)$$

and R is replaced by a function of R , $f(R)$,

$$S_{EH} = \frac{1}{2\kappa} \int \sqrt{-g} f(R) d^4x. \quad (3.46)$$

The field equations we derive from this action are covariant and Lorentz invariant for the exact same reasons that Einstein's equations are. To get standard gravity from a $f(R)$ -theory we just set

$$f(R) = R. \quad (3.47)$$

There are 2 different variational methods used to reduce the Einstein-Hilbert action to Einstein's field equations. One is the metric (or second order) formalism, here the field equations are derived by variation of the action with respect to the metric tensor $g_{\mu\nu}$. The affine connection $\Gamma_{\beta\gamma}^\alpha$ depends on $g_{\mu\nu}$. The other formalism is Palatini (or first order) formalism, where $\Gamma_{\beta\gamma}^\alpha$ and $g_{\mu\nu}$ are independent variables. Because of this, as we intuitively might expect, there exists several variants of $f(R)$ gravity. By using the metric formalism we get metric $f(R)$ gravity while Palatini formalism gives Palatini $f(R)$ gravity. There also exists a third variant of $f(R)$ gravity: metric-affine $f(R)$ gravity. This is a more general theory, and by making certain assumptions it can be reduced to both metric $f(R)$ gravity and Palatine $f(R)$ gravity. Metric-affine $f(R)$ gravity is derived by using Palatine formalism, but without the assumption that the matter action is independent of the connection.

Various $f(R)$ models

Since $f(R)$ gravity changes with the chosen function, there exists an infinite amount of $f(R)$ models. The changes can also be made arbitrarily small, which makes it hard to test $f(R)$ models against standard general relativity or even against other $f(R)$ models. To reduce this problem it is often assumed that the function $f(R)$ takes the form of a Taylor expansion.

$$f(R) = \sum_{i=0}^{\infty} a_i R^i \quad (3.48)$$

$$= a_0 + a_1 R + a_2 R^2 + \dots \quad (3.49)$$

Here the first term is similar to a cosmological constant and setting the second term to $a_1 = 1$ give us standard general relativity. The most realistic $f(R)$ model to date is found by using the function $f(R) = R + \alpha R^2$, it was the first inflation model introduced by Starobinsky in 1979 (Starobinskiĭ 1979). Generally the function follows $f(R) = R + \alpha R^n$, but as shown by Hwang and Noh (Hwang & Noh 2001), there is only one viable option, with $n \approx 2$. The αR term leads to the accelerated expansion of the universe while the linear R term makes sure that inflation ends smoothly. This is a viable theory to the standard scalar field inflation.

3.6 Symmetron model

Much of recent research have focused on developing screening mechanisms to explain why scalar field have not been detected in local experiments of gravity, such as in the solar system. The symmetron model, a scalar field theory, elegantly provides such a screening mechanism. It is one of the known mechanisms for screening a fifth force, and thus recover general relativity in the dense regions of the universe. It contains a scalar field that permeates the universe and is the cause of a new force that interacts between massive objects, this field is called the symmetron field, ϕ . The key property of this field is that it is symmetric in regions with high density, while in regions of low density the symmetry is broken and the field mediates a new force. When the symmetry is broken the field couples to matter with a given matter coupling energy.

In this theory the vacuum expectation value (VEV) of the scalar field depends upon the local matter density, the VEV of an operator is the operators expected value in vacuum. The VEV becomes large in regions with low matter density and low in regions with high matter density. In vacuum the scalar field gets $\text{VEV} \neq 0$, which breaks symmetry $\phi \rightarrow -\phi$. In high density matter fields ϕ is confined close to zero.

Since the matter density of the universe changes with time, the effective potential of the symmetron field evolves and we get a phase transition when the matter density falls below a critical value. For a different range of initial conditions the symmetron scalar field reaches this symmetry-breaking vacuum at the present time. N-body simulations of the evolution of structure formation shows that there are differences in halo structure for halos formed before and after this phase transition.

The symmetron model was first introduced with a hope that it could explain the accelerated expansion of the universe, but it has been showed that the energy density of the symmetron field is too low to be able to contribute to the dark energy. The symmetron model therefore needs to add a cosmological constant to restore the accelerated expansion, and because of this the background evolution of the symmetron model is indistinguishable from the Λ CDM universe, (Hinterbichler et al. 2011). But since the symmetron model provides an elegant screening mechanism, there have been proposed theories that use the symmetron model as a switch that turns on f(R)-gravity once the symmetron field reaches the symmetry broken phase (Bamba et al. 2012).

The symmetron model is specified by the following action

$$S = \int \sqrt{-g} \left[\frac{R}{2} M_{\text{pl}}^2 - \frac{1}{2} (\partial\phi)^2 - V(\phi) \right] d^4x + S_m(\tilde{g}_{\mu\nu}, \psi_i). \quad (3.50)$$

Here M_{pl} is defined as $M_{\text{pl}} \equiv \frac{1}{\sqrt{8\pi G}}$, ψ_i are the various matter fields. $\tilde{g}_{\mu\nu}$ is a conformal rescaling of $g_{\mu\nu}$ given by

$$\tilde{g}_{\mu\nu} = A^2(\phi) g_{\mu\nu}. \quad (3.51)$$

We can the obtain the Einstein equations from equation 3.50 similar to how we did in section 3.2.1. The cosmology obtained from a coupled scalar field model is strongly constrained by local measurements of gravity putting constrains on the coupling strength and range of the scalar field.

The fifth force is generally stronger for smaller halos than large ones, because small halos are most often located in low density regions of space where there is less suppression of the force. Because the fifth force is dependent on matter density, dark matter halos

in high density regions will generate a different fifth force in comparison to similar halos in low density regions. At large scales the galaxies in the universe is divided into galaxy filaments, these are among the largest known cosmic structures in the universe. They are massive, thread like structures that form boundaries of large voids in the universe. Due to contrast in matter density inside the filaments compared to the voids, halos along these filaments are going to experience a direction dependence in the fifth force. This has an impact on the shape of the halos, they can become elongated.

3.7 Tests of general relativity

3.7.1 Classical tests

Einstein proposed three different tests of general relativity in 1916 (Einstein 1916), called the classical tests of gravity. Shapiro proposed another test, the light travel time delay test (Shapiro 1964), which also is considered one of the classical tests. All modified theories of gravity must be able to explain and reproduce the observed values in these tests. The classical tests are:

Perihelion precession of Mercury's orbit

The perihelion of Mercury's orbit is rotating around the sun. This precession deviates from the predicted precession calculated using Newtonian mechanics, but is fully explained using general relativity.

Gravitational lensing

A matter distribution bends the trajectory of light moving close to it. This was one of the first observational confirmations of general relativity.

Gravitational redshift of light

Light moving out of a gravitational field is redshifted when it is observed in a region with a weaker gravitational field.

Light travel time delay

This is a relativistic delay in the round trip a radar signal takes when bouncing off a planet in the solar system. The measurements are performed so that the signal moves close to the sun, to maximize the effect of the sun's gravitational potential.

3.7.2 Modern tests

In addition to the classical tests, there are several other tests that a modified gravity theory must pass in order to be accepted. As with the classical tests, all modified theories of gravity must be able to explain and reproduce the observed values.

Completeness

To be complete, a theory must be able to analyze every experiment of interest. If a theory cannot predict gravitational redshift or the orbit of planets, it is not complete. It must also mesh with all other physical theories.

Agreement with the Newtonian mechanics

In the limit of a weak gravitational field created by slow moving objects the model must agree with Newtonian mechanics as well as special relativity.

Cosmological tests

Cosmological tests are fairly new, and these are better at discerning between different gravitational theories. They are explained in more detail later in this section.

Frame-dragging tests

In general relativity a massive rotating object will affect the metric of spacetime. This will make the orbit of a nearby test particle precess.

Cosmological tests

Gravitational waves are one of the best methods to test different theories of gravity. A gravitational wave is a ripple in the curvature of spacetime, it propagates as a wave, traveling outwards from the source. All known modifications of general relativity predict the existence of gravitational waves, but not the same properties. The type of the observed gravitational wave will rule out several modified gravity theories. The gravitational waves can take on a lot of different forms and there are several ways we can discern between the gravitational waves predicted by the different theories. One property that varies is the speed the waves propagate with. In GR the speed of the waves are equal to the speed of light in vacuum, while in many other theories the speed is lower. By comparing the arrival time of the electromagnetic radiation with the arrival time of the gravity waves several models can easily be eliminated.

A better, second test is to measure the polarity of the gravitational wave. General relativity is the only theory that predicts radiation with helicity modes ± 2 . How much observations of the polarity can constrain the gravity theories depends upon if we can identify the source of the waves. If the source can be identified it is possible to obtain the polarization mode, if not it might still be constrained to a limited amount of possibilities. There exist six different polarization states giving us six different tests for modified gravity theories, one for each mode. In GR the radiation is always quadrupole radiation that carries away positive energy while many other theories predict dipolar radiation where they sometimes carry away negative energy. This could be used as an additional test.

Gravitational waves have not yet been directly detected, making these tests unable to be performed, but there are indirect evidence for gravity waves existence (Hulse & Taylor 1974).

Parameterized post-Newtonian formalism

Post-Newtonian formalism is a tool that expresses theories of gravity in terms of the lowest order derivations from Newtonian gravity. Parameterized post-Newtonian formalism is a version of this, that in explicitly states how the parameters in the modified theories of gravity differs from Newtonian gravity. This is a useful tool when comparing modified gravity theories to Newtonian gravity in the limit of a weak gravitational field created by a slow moving object. It is also very useful when comparing different modified theories of gravity against each other.

N-body simulations

To be able to observe the evolution of structure formation in the universe in the non-linear regime we are required to perform N-body simulations, these simulations are an important tool to see how the large scale structure of the universe evolves in different schemes of modified gravity. As mentioned, some theories of modified gravity, such as the symmetron model, affects the shape of halos. So when we know the evolution of the structure formation for a given theory, we can compare it with observations to see how well the fit is. To be able to make sense of the output of a N-body simulation we need to perform a halo finding, as it makes no sense to compare the individual particles from the simulations with actual observations. What the halo finding process entails is to determine where there are groups of galaxies bound together by gravity. Once these groups are found we can compare them with observational data.

Chapter 4

N-body simulations

4.1 Introduction

N-body simulations simulate how a dynamical system of N particles behave and evolve under the influence of physical forces. Numerical simulations of N-body systems have a surprisingly long history, the first calculations was done by Sebastian von Hoerner in 1960, who performed simulations with 16 particles (von Hoerner 1960). From then on the number of simulated particles have evolved in accordance with Moore's law (Moore 1965) roughly doubling every 18 months.

The particles treated by an N-body simulation often correspond to actual physical objects, but not necessarily. An example is a N-body simulation of a cloud of gas, it is not feasible to have each molecule in the cloud represented as a particle, instead each particle in the simulation represents a much larger quantity of gas. On the other hand, if we simulate a solar system it is natural to assign each physical object as particle in the simulation. Cosmological N-body simulations are often used to study the evolution of the large scale structures in the universe, in this case each "particle" is often a gathering of dark matter the size of a billion solar masses.

The general idea behind a N-body code is quite simple, the code only consists of two major modules, one to calculate the force on all particles from a given particle configuration at a given time and another to move all particles according to the force affecting them. This general algorithm can be seen in Algorithm 1. The complications come from the sheer number of particles being simulated, both these modules are computationally intensive, especially the force calculation, making optimization of the code vitally important.

Algorithm 1 Pseudocode for a N-body simulation

```
for each time step  $t$  do
  for each particle  $i$  do
    calculate  $F(x_i(t))$ 
    integrate to calculate change in position,  $\Delta x_i$ 
     $x_i(t + \Delta t) = x_i(t) + \Delta x_i$ 
  end for
end for
```

There exist two major types of N-body calculations, collisional and collisionless. Here collisional does not mean actual collisions between objects, but rather close encounters between them. The latest collisional N-body simulations have reached more than 10^6 particles (Harfst et al. 2007), while collisionless calculations have reached over 10^9 particles (Springel et al. 2005; Teyssier et al. 2009; Stadel et al. 2009; Iliev et al. 2010). The difference between collisional and collisionless N-body simulations can be explained using the relaxation time of the system, commonly defined to be the time it takes for an object's velocity to change by a order of itself. Collisional N-body simulations are dynamically old, the dynamic time is short compared to their age and the relaxation time is shorter than the simulated timespan. An example of such a system is globular clusters, which most often orbit deep within the gravitational potential of a host galaxy. The accumulated effect of many small close encounters between objects significantly affects the evolution of the system.

In a collisionless N-body simulation the long term effect of two body encounter are negligible. The most often encountered collisionless system is galaxies and systems of galaxies, here the number density is so low and the time scale so large that stellar encounters no longer play an important role. As such cosmological N-body codes are collisionless. If there are no initial perturbations, a collisionless system settles into a dynamic equilibrium.

The universe is large compared to the area of interest in a cosmological N-body simulation and the simulation volume can not be assumed to exist in isolation. If there are empty space outside the simulation volume, matter tends to collapse towards the center of the volume. The simplest solution to avoid this is to use periodic boundary conditions. Due to this, most simulations are performed inside a cubical volume, as that is the natural geometrical volume to use with periodic boundary conditions.

4.2 Numerical methods for calculation of the force

The computation of the gravitational force is the most time consuming part of any cosmological dark matter N-body simulation, thus a lot of work have been done to develop fast force calculation algorithms.

4.2.1 Direct summation

Direct summation is the easiest and most precise method to compute the force, but by far the slowest. The contributions to the force on a particle is summed up from all other particles, and the process repeated for each particle. This method scales as $O(N^2)$, so direct summation quickly becomes useless as the number of particles increase.

4.2.2 Tree codes

The tree code was developed by Barnes and Hut (Barnes & Hut 1986). The idea behind this algorithm is to approximate the long range interactions between particles by gathering the far away particles into groups. Then the force from each group is computed, instead of from individual particles. There are now less interactions to be calculated, a tree code therefore scales as $O(N \log(N))$ compared to the $O(N^2)$ of the direct summation. The

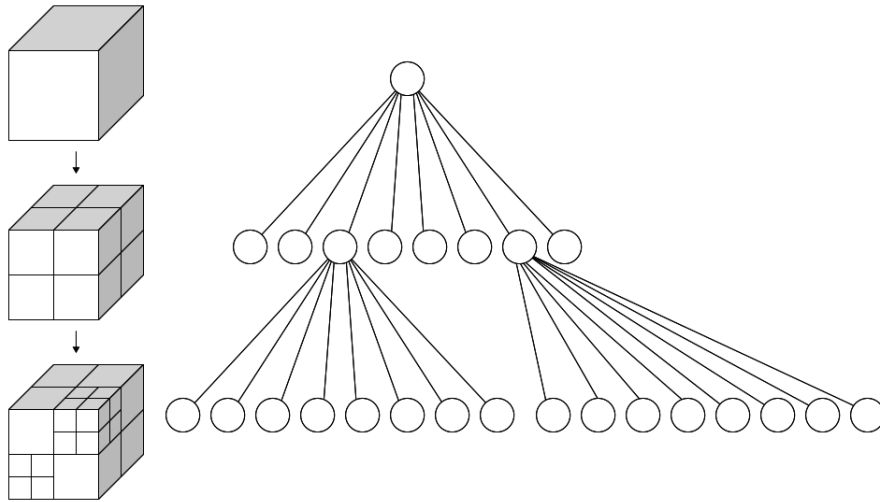


Figure 4.1: Example of an octree, image courtesy of wikipedia

disadvantage with this method is that the approximations introduce small errors in the force calculations.

An octree is used to store the particles; it is a tree data structure in which each internal node has exactly eight children. The octree is generated by dividing the volume cube into eight, one partition for each node in the octree and then loop through all particles. Each particle is put into the appropriate node, if the given node already contains a particle, that node is split into eight, and the original particle is put into the correct node before the new particle is tried to be assigned to a node. This process continues until the new particle is in a node by itself. A visualization of a quadtree can be seen in figure 4.1. This process is repeated for each particle until all have their own node. The quadtree is then traversed from the bottom up and the center of mass and total mass for each node is calculated.

This tree is used when the forces is calculated when the particles are far away from the given particle we do not traverse completely through the tree, but stop once the given threshold is reached and use the total mass and position of that node. The octree does not need to be recomputed for each time step, which saves valuable CPU time. This threshold is called the opening angle, θ , which determine how far away particles must be before they are grouped together. The opening angle is given by

$$\theta = \frac{s}{D}, \quad (4.1)$$

where s is the width of the region represented by the internal node, and D is the distance between the particle and the node's center of mass. The node is deemed sufficiently far away when θ is smaller than a given value. The parameter θ gives the accuracy of the simulation, larger values increase the speed of the simulation but decrease accuracy. When $\theta = 0$ the algorithm degenerates to a direct sum algorithm, as no internal node is treated as a single group.

4.2.3 Particle Mesh codes

The particle mesh (PM) scheme is a different way to speed up the calculations. This scheme is based on representing the gravitational potential on a Cartesian grid and then solve the Poisson equation for this grid. The PM method consist of three steps. The first is to calculate the density field. The simplest method is to assign each particle to the closest grid point (NGP), but this method creates large truncation errors. A better method is to use multi-linear interpolation to assign a particle to the eight closest grid points, called Cloud-in-Cell (CIC). What makes this method so fast is the Fourier space solution of the Poisson equation, given as

$$\hat{\phi}(\vec{k}, t) = -\frac{4\pi G a^2}{k^2} \hat{\rho}(\vec{k}, t). \quad (4.2)$$

Here $\hat{\phi}$ and $\hat{\rho}$ are the discrete Fourier transforms of gravitational potential and the mass density. The last step is then to interpolate the gravitational force back to the particles. The PM method scales as $O(N) + O(N_g \log N_g)$, where N_g is the number of grid points used. For standard grid sizes the PM method is faster and needs less memory than the tree algorithm. The disadvantage with this method is that it does not cover close particle interactions very well since the method has low short range accuracy.

4.2.4 Particle-Particle Particle-Mesh codes

This scheme, called a P³M scheme, is a hybrid algorithm between a Particle Mesh code and a direct particle-particle code and tries to improve the resolution of the PM scheme. The idea is to split the force into a long range and a short range force, the PM method is used to calculate the long range force, while Particle-Particle interaction is used to correct the force at shorter distances, when particles are within two or three grid spacings. In the presence of a dense cluster of particles the Particle-Particle step significantly slows the code down, to avoid this, an adaptive mesh is often chosen (Adaptive Mesh Refinements - AMR), so that we have a finer resolution in regions of high density. Pair summation is still done, but now for particle pairs within two or three grid spacings of the subgrid. Adaptive P³M codes scales as $O(N \log N)$. At the present time parallel adaptive P³M schemes seems to be the best choice for large cosmological N-body simulations.

4.3 Gas dynamics

If baryons are present in the simulations gas dynamics must be taken into consideration. The strong non-linearity of the gas dynamical equations makes it difficult to develop numerical methods whose accuracy can be proven. One of the more popular methods is to use Smooth-particle Hydrodynamics (SPH), in this algorithm the fluid variables, such as baryon density, temperature, velocity, etc., are represented as fixed mass particles. The baryonic mass density is calculated by smoothening each particle by a smoothening function, W . Since SPH makes use of particles it is an extension of N-body methods, making it easy to extend existing N-body codes with this method.

4.4 Numerical methods for moving particles

Once the force have been calculated the particles need to be moved, which is done by integrating the equations of motion. The equations of motion for a set of particles only interacting through gravity can be written as (Bagla & Padmanabhan 1997).

$$\ddot{x} + \frac{\dot{a}}{a}\dot{x} = -\frac{1}{a^2}\nabla_x\varphi, \quad (4.3)$$

$$\nabla_x^2\varphi = \frac{3}{2}H_0^2\Omega_0\frac{\delta}{a}, \quad (4.4)$$

here x is the comoving coordinates, φ is the gravitational potential due to perturbations and δ the density contrast. These equations are then numerically integrated for each particle. Any numerical integration scheme can be chosen, for example the leap-frog method or second/fourth order Runge-Kutta. The fourth order Runge-Kutta method is shown as an example. We have an unknown function $y(t)$, and know how it changes with time

$$\dot{y} = f(t, y), \quad y(t_0) = y_0. \quad (4.5)$$

A step length h is chosen and the notation $y(t + nh) = y_n$ is used, where n is the number of steps taken. The scheme now becomes

$$y_{n+1} = y_n + \frac{1}{6}h(k_1 + 2k_2 + 2k_3 + k_4), \quad (4.6)$$

$$t_{n+1} = t_n + h, \quad (4.7)$$

for $n = 0, 1, 2, \dots$, where

$$k_1 = f(t_n, y_n), \quad (4.8)$$

$$k_2 = f(t_n + \frac{1}{2}h, y_n + \frac{h}{2}k_1), \quad (4.9)$$

$$k_3 = f(t_n + \frac{1}{2}h, y_n + \frac{h}{2}k_2), \quad (4.10)$$

$$k_4 = f(t_n + h, y_n + hk_3). \quad (4.11)$$

The fourth order Runge-Kutta method is as the name implies a fourth-order method, so the local truncation error is on the order of $O(h^5)$ while the total accumulated error is of order $O(h^4)$.

4.5 Ramses

An example of a N-body code is RAMSES (Teyssier 2002). RAMSES is based on Adaptive Mesh Refinements, with a tree data structure that allows recursive grid refinements. These refinements are created and destroyed dynamically from cell to cell, which makes the code more flexible and better able to capture complicated flows. The hydrodynamical solver in RAMSES is based on the Godunov scheme, a scheme for solving partial differential equations, and is used to treat the evolution of baryons. The refinements are found by an algorithm based on a "quasi-Lagrangian" mesh evolution, ensuring that the two body relaxation can be ignored. The numerical integration method is a second-order midpoint scheme. RAMSES outputs a file for each processor, that contains the position and velocity of each particle at a given time, calculated by the given processor.

4.6 ISIS

Another example of a N-body code is ISIS (Llinares et al. 2013), this is the code that provides the datasets our halo finder is tested on later in this thesis. ISIS is based in RAMSES but can additionally solve for a large class of scalar-tensor theories of modified gravity. The large modification is that a non-linear implicit solver have been added to take into account the equation of motion of the scalar field in a static approximation instead of the standard linear Poisson solver implemented in RAMSES.

Chapter 5

Halo Finders

5.1 Introduction

As seen in Chapter 4, a cosmological N-body simulation typically calculates the evolution of a set of particles and output the position and velocity of each particle at a given time.

N-body simulations are stochastic, the initial conditions are random and the evolution chaotic. Comparing the particles positions and velocity directly with observations are optimistic. Due to this there have been developed a series of different statistics that we can use instead. Examples are the N-pint correlation function, the distribution of particles in each cell, the velocity distribution of single particles, etc. For this thesis the most interesting statistics is the internal properties of clusters of galaxies, halos, but as such we need to find the clusters and which particles belong to a cluster. This is the job of a halo finder. The field of halo finding is fast expanding, in the last ten years the number of halo finders have tripled (Knebe et al. 2013).

The first generation of halo finders focused on finding isolated halos, while modern halo finders look for halos within the uniform background density of the universe as well as subhalos contained within larger host halos. There are also new challenges with regard to the size of the simulations. The simulation data have become to large to analyze on a single CPU in a reasonable amount of time and thus the developers have started to parallelize their halo finders to cope with the problem. There also is a starting trend to include halo finders in the N-body simulation codes to reduce the amount of data that needs to be stored. The problem with such a combination of N-body code and halo finder is if the halo finder is incomplete or we want to look at new properties of the halos, it would then be necessary to rerun the entire simulation, as the raw simulation data is lost in the halo finding process and only a reduced dataset is saved.

5.1.1 What is a Halo?

Since it is desirable to write a program that identifies halos in the resulting chaos of a N-body simulation it might be useful to know what a halo is. Unfortunately this is a surprisingly hard question to answer and it has been the focus of several studies (Maccio et al. 2003; Prada et al. 2006). It is commonly agreed that a halo is a gravitationally bound object (Knebe et al. 2011), but even with this simple definition there exists several problems.

The first problem arises from the fact that halos may have substructures. Should the mass of a subhalo be considered as a part of the mass of the host? The answer to this depends on the specific scientific problem investigated, for example studies of gravitational lensing need to know the complete mass of both the host and its subhalos. It is important to note that in some halo finders return the host mass including the mass of substructure, e.g AHF (Knollmann & Knebe 2009), while others do not.

The second problem is how to define the edge of a halo. The most commonly used method assumes spherical symmetry and uses a theoretically calculated factor for the reference density, which varies from halo finder to halo finder. To complicate matters, it is not certain that dark matter halos can be approximated as spherical, especially when the effects of modified gravity theories is taken into account, as mentioned in section 3.6. The above mentioned problems regarding substructure of a halo is also the cause of differences when defining the edge of a halo, should the subhalos be included in the definition of the edge of the host halo or not?

As we can see from this short discussion there are several different definitions of some of the most basic properties of a halo and none are more correct than the others. The author of a halo finder code choose a definition that fits the specific problem to be studied. As such it is important to note that different halo finders will get different results, even when analyzing the same dataset.

5.1.2 The need for a modified gravity adjusted halo finder

Since there already exist several halo finders why is there a need to implement another? The problems with existing halo finders is that none of them is equipped to handle simulation data from N-body simulations that have incorporated modified gravity theories. That is to say, they run on such datasets, but the implemented equations are derived from standard general relativity and as such the validity of the results are unknown. Previously this has not been a big problem, but recently there has been developed a modified gravity N-body code at the University of Oslo, ISIS, (Llinares et al. 2013). Because of this we want to write a halo finder, MORPH, that takes into account modified gravity theories and examine the effects, as there is a need for such a code to complement ISIS.

5.2 The General Method of Halo Finding

The first two halo finders that showed up in literature are the foundation for most halo finders today. These are the spherical overdensity (SO) method (Press & Schechter 1974) and the friends-of-friends (FOF) algorithm (Davis et al. 1985). These two methods can be generalized as Density Peak Locators and Particle Collectors:

Density Peak Locators

Density Peak Locators, such as the SO method, works by finding peaks in the matter density field by different means. One example is to use a particle simulation to detect the centers particles gravitate towards. Once the peaks in the matter density field are located, particles are added in spherical shells of increasing size until the enclosed mass falls below a predetermined density threshold, derived from a spherical top-hat

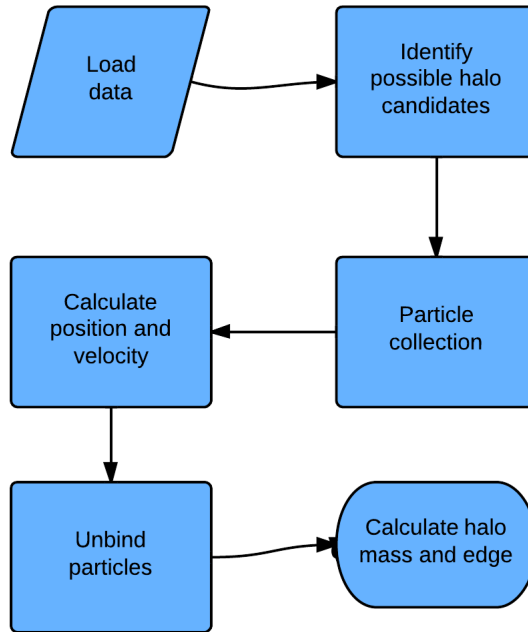


Figure 5.1: The general steps performed by a halo finder.

collapse. AHF, Amiga’s Halo Finder (Knollmann & Knebe 2009), an example of a Density Peak Locator, is described in section 5.5.

Particle Collectors

Particle Collectors, mainly FOF algorithms, link together particles that are close to another in either 3D or 6D phase-space. It is a purely geometric method and connects particles that are closer than a prescribed length, called the linking length. There exist a few problems that may arise with this algorithm, the foremost being the risk of two halos linked together via a ”linking bridge”, figure 6.7. If there is one particle that is within linking range of two halos, that particle cause the two separate halos to be linked together as one big halo. It is also impossible to find substructures within a halo if a fixed linking length is used. Rockstar (Behroozi et al. 2013), an example of a Particle Collector, is described in section 5.6.

An overview of the general steps in a halo finder can be seen in figure 5.1.

5.2.1 Identification of possible halo candidates

The first step in almost all non-FOF halo finders is to generate a list of potential halo centers. This is done by either locating peaks in the density field or minimums in the gravitational potential field. One method is to simulate water particles flowing in the gravitational potential field and see where they gather, which are at the positions of the minimum points in the potential. There exists techniques that avoid this step by using velocity (e.g. STF (Elahi et al. 2011)) or phase-space (e.g. Rockstar (Behroozi et al. 2013)) based methods.

5.2.2 Particle Collection

After the list of potential halo centers have been generated particles belonging to these objects must be gathered, how this collection is accomplished varies from halo finder to halo finder and the differences affect the final results. One difference stems from the choice whether a particle can belong to more than one object at any given time. If it can not, the mass of a subhalo is not considered a part of the mass of the host halo. If a particle can, it may be assigned as both a member of a subhalo and the subsequent host halos.

5.2.3 Calculating position and velocity of halos

Once a list of possible particle candidates have been created for each potential halo their position needs to be determined. There is a wide array of different ways to do this, some halo finders use the center of mass for a given fraction of the particles while others use the position of the density peak or the minimum in the gravitational potential used in the identification of possible halo candidates step. The same must to be done for the velocity of each halo, which can be calculated by a simple average of all particles, a central subset or using the velocity of the most bound particle.

5.2.4 Unbinding procedure

The purpose of the unbinding procedure is to remove particles that are not gravitationally bound to a halo. Such a particle would, after a given time have moved away from the halo and it would be wrong to consider it a part of the halo. The unbinding criterion is given by the escape velocity of a particle

$$v_{\text{esc}} = \sqrt{2\phi}, \quad (5.1)$$

where ϕ is the gravitational potential. Particles are removed from a halo if their velocity $v > v_{\text{esc}}$. The unbinding procedure can be linked to the calculation of the position and velocity of halos, since it removes particles from a halo, and may thus introduce a need for the velocity and position to be recalculated. The unbinding procedure can also be considered partly linked to the particle collection algorithm. Some halo finders rely on the fact that particles that do not belong to a halo are removed in this step. They can therefore use collection methods that are less conservative, as particles can be added to a halo without fear of it not being an actual member of the halo.

All halo finders treat halos as if they are in isolation in the unbinding procedure, it is not a part of a host halo and there are no subhalos around. There are several variations in how different halo finders calculate the gravitational potential, the order in which they remove particles and when to stop the unbinding procedure. This leads to some discrepancies between the halo finders in their final results. Some codes calculate ϕ using a tree structure, while others assume spherical symmetry. Furthermore there are codes that remove one particle at a time before recalculating ϕ , while others remove all particles found to be unbound in one iteration before reiterating the entire unbinding process, others again restart the process once a fraction of the particles have been unbound. In addition there also exists different termination criteria, some codes stop when there are no more particles to be removed while others stop once a fraction of the particles have

been removed. It also has to be mentioned that there exists codes that do not have any unbinding procedures, for example ADAPTAHOP (Aubert et al. 2004).

All configuration space halo finders need an unbinding procedure no matter how conservative the particle collection, to make up for false positives (Knebe et al. 2013). 3D space halo finders are essentially converted into mock 6D phase-space halo finders by the inclusion of an unbinding procedure, since they then take into account a particle's velocity when deciding if the particle is to stay as a member of the halo. Phase-space halo finders are more reliable when creating the particle list, as they take into account both velocity and position of each particle. Thus there is a lesser need for an unbinding procedure and a smaller percentage of particles needs to be unbound.

It should be mentioned that the unbinding procedure is not a physically motivated method, particles can become momentarily unbound to later become bound again. If we remove these particles because they are unbound, we remove particles that should belong to the halo. One way to get around this is to track the particles over several time steps. The need for an unbinding procedure is also dependent on the physical problem being addressed, and not only the algorithm used in the particle collection. If we only want to look at the mass of halos, the mass of a few particles that should be removed only give a small error in the final result. On the other hand parameters such as the spin of a halo, are sensitive to such interlopers.

5.2.5 Calculating halo mass and edge

This is one of the more problematic topics, as discussed in section 5.1.1, and have given rise to several different definitions, below are a few taken from Knebe et al. (2013).

- The halo edge is the distance to the farthest bound particle.
- The halo edge is defined via the spherical top-hat collapse model.
- The halo edge is the 'zero-velocity' radius.
- Do not define an edge, just give the bound particle list.
- It should be defined dynamically, only particles that stay with the halo for several time steps are considered a part of the halo.

As we can see from this, it depends on the actual scientific problem which method we should choose when calculating halo mass and edge.

5.3 Source of the scatter in the final result

As we have seen, halo finding is not as easy as applying a filter to the results from a N-body simulation. It requires several steps, and many of these steps can be done very differently, changing the results depending on the method used. Because of this we will get a scatter in the physical properties of a halo depending on the halo finder used, this is unfortunate as we would like to get the same result independently of the code.

There are two major reasons for this scatter:

Quantity	Scatter
<u>Set of common objects:</u>	
Position	$< 1\%R_{200}$
Bulk velocity	$< 1\%V_{\max}$
M_{200c}	3%
V_{\max}	$< 1\%$
R_{\max}	2%
Shape	5%
Spin	18%
<u>Full catalogues:</u>	
Mass function	10 %
R_{\max} function	20 - 30 %

Table 5.1: Scatter in the main properties computed by halo finders, from Knebe et al. (2013)

- Halo finder methodology.
- Definition of halo properties.

The scatter from halo definitions can be taken into account as we know which definitions are used, and can change them as needed. Thus the scatter must originate from the halo finder methodology. Here there are two possibilities, the halo finding scheme or the unbinding method. Knebe et al. (2013) examined these sources closer and found that most of the scatter in the halo finder methodology comes from the initial particle collection. They tested the scatter between halo finders for a set of common objects versus the full halo catalog, their results are in table 5.1. As can be seen from the set of common objects the scatter in $V_{\max} < 1\%$ and M_{200c} 3%

One of the current goals of halo finders are to get more consistent results, at the present halo finders can give 10% error bars on halo mass and 20-30% error bars on V_{\max} (Knebe et al. 2013).

5.4 Astrophysical Applications

The applications of halo and sub-halo information are spread over several different fields of astrophysics. Galaxy formation, dark matter detection, gravitational lensing and modified gravity simulations are described below. Other possible applications are examining halo streams and tidal debris, large-scale structure formation and near-field cosmology.

5.4.1 Galaxy formation

Simulating the evolution of the visible universe is a harder task than to just track dark matter. Visible matter have several types of dynamics that dark matter does not have. This creates additional challenges for halo finders, as we need a well defined treatment of gas particles.

5.4.2 Dark Matter Detection

There are two distinct ways to detect dark matter, direct and indirect detection. Indirect detection tries to observe secondary particles from either the decay or self-annihilation of dark matter while direct detection searches for the nuclear signature recoil of dark matter collisions. Both of these methods are sensitive to dark matter substructure in dark matter halos. The emission for indirect detection scales with the density, therefore subhalos can enhance the signal compared to the signal from a smooth dark matter halo (Stoehr et al. 2003). Direct dark matter detection is sensitive to the local velocity distribution and because of this subhalos can distort the signal. To be able to theoretically model the received signal we need an accurate model of the complete structure of the dark matter halo along with the velocity distribution, so we know what the signal we are searching for looks like.

5.4.3 Gravitational Lensing

A gravitational lens is a distribution of matter, such as a halo, that lies between us and a distant object, e.g. a galaxy. This matter distribution bends the light from the distant object and this can be used to find the mass distribution for the object the light is lensed around (Einstein 1936). There exist three distinct types of gravitational lensing, strong-, weak- and micro-lensing. Halo finders are not of immediate relevance for micro lensing and it is therefore ignored.

Strong Lensing

In strong lensing we get multiple images of the distant object. We can use the particular features of the multi-image to determine the matter distribution of the lens. To be able to do this we need reliable models for how the matter could be distributed. These models are primarily acquired from simulations of the cosmic structure formation, which in turn use one or more halo finders to find the dark matter halos.

Weak Lensing

Most gravitational lenses are weak, which differs from strong lensing in that we do not get separate images, but only a distortion of the original image. Weak gravitational lensing is one of the key methods used to test cosmological models, and dark matter and dark energy distributions. As with strong lensing we need reliable models of the matter distribution of the lens, and therefore need halo finders for the same reasons.

5.4.4 Modified Gravity Simulations

This is the field most relevant for this thesis. There have been developed N-body codes at the University of Oslo that simulate the evolution of particles according to a specific modified gravity theory (Llinares et al. 2013). We can use these codes to test MG theories to see which might be plausible and which can be discarded. Many of the standard halo properties, such as virial radius, rotation curve, spin, etc., are calculated using GR and therefore must be changed to take into account modified gravity models. This also means

that the results of the unbinding procedure are affected when using modified gravity, and as such MG must be taken into consideration when the unbinding is performed.

5.5 AHF

AHF (Knollmann & Knebe 2009) is a Density Peak Locator that is parallelized with MPI and OpenMP. The halo finding algorithm is fully recursive and automatically identifies halos, sub-halos, sub-sub-halos, etc. AHF starts by covering the simulation area in a grid of user defined size and in each cell the particle density is calculated using a triangular shaped cloud weighing scheme. If the particle density is above a given threshold the cell is divided into a finer grid with half the cell size and the density is recalculated. If the density still is above the threshold the process is repeated until no further refinements are needed. By doing these refinements we get a grid that follows the density field.

We now start with the finest cells and mark isolated regions as potential halos. Then the same is done with the slightly coarser grid, but it is also linked together with the inner halo. Following this procedure, a tree of nested halos is created, that follow the halos from high density to low density. If two isolated patches are linked together, the two branches of the two trees are joined. This creates a tree structure of nested grids that follows the density contour levels. Once this is done substructure can be determined. The tree is traversed from top to bottom, if the tree splits up, the number of particles in each branch are counted and the branch with the most particles is considered the host halo while the other branches are subhalos. This is done for each level where the tree branches. Now AHF runs the unbinding procedure, the halos are treated as if they are in isolation and the gravitational potential is calculated using standard general relativity, see section 6.5.1. Particles that have a velocity greater than the local escape velocity are removed and the removed particles from the subhalos are tested against the host to see if they instead are bound there. It is important to note that subhalos are included in their host halos.

5.6 Rockstar

Rockstar (Behroozi et al. 2013) is a phase-space based halo finder. It is designed to maximize halo consistency across several time steps, in addition to a focus on efficiency and parallelization. This halo finder MORPH most closely resembles.

The first step Rockstar performs is to run a fast 3D FOF method and the found groups are used to divide the simulation up into separate work units. Rockstar uses a modified FOF algorithm, as usual neighboring particles are assumed to be a part of the same halo if they are within the linking length, but if a particle has more than 16 neighbors, the neighbor-finding process is skipped for these particles and Rockstar checks for the particles up to twice the linking length. If any of those particles are within this new linking length and belong to another FOF group, the two groups are linked together. This algorithm makes it so neighbors must be calculated over a larger distance, but it needs far fewer of these calculations. It also makes it so that Rockstar can use a larger linking length, with $b > 0.2$, than usual in FOF-based halo finders. The algorithm becomes faster when a bigger linking length is chosen.

For each FOF group, Rockstar creates a hierarchy of FOF subgroups in phase space and adaptively calculates a phase space linking length. The phase-space distance between two particles, p_1 and p_2 , is given as:

$$d(p_1, p_2) = \left(\frac{|\vec{x}_1 - \vec{x}_2|^2}{\sigma_x^2} + \frac{|\vec{v}_1 - \vec{v}_2|^2}{\sigma_v^2} \right)^{1/2}, \quad (5.2)$$

here σ_x^2 and σ_v^2 are the particle and velocity dispersion for this specific FOF group. For each particle the distance to the nearest particle is calculated and the linking length is chosen such that a fraction, $f = 0.7$, of the particles have a neighbor closer than the linking length. If there are more than 10,000 particles in the halo, a random subset of 10,000 particles are used to calculate the linking length, this is to speed up the code since this algorithm scales as $O(N^2)$. Once subgroups are found this process is repeated for each subgroup until the minimum number of particles is reached for each subhalo. A seed halo is generated for each such subgroup at the lowest level and particles are recursively assigned to the closest seed halo. This distance is computed by

$$d(h, p) = \left(\frac{|\vec{x}_h - \vec{x}_p|^2}{r_{vir}^2} + \frac{|\vec{v}_h - \vec{v}_p|^2}{\sigma_v^2} \right)^{1/2}, \quad (5.3)$$

$$r_{vir} = \left(\frac{M_h}{\frac{4}{3}\pi\rho_{vir}} \right)^{1/3}. \quad (5.4)$$

Here h denotes the halo and p the particle. σ_v the velocity dispersion of the halo, M_h the mass of the seed halo and vir the virial overdensity. The authors use the halo radius r_{vir} to make sure that the particles are close to the main density peak, even if they are close in velocity space. r_{vir} makes it so that if two halos are within each other virial radius, velocity space becomes the dominant method of distinguishing between two halos. For all seed halos the Poisson uncertainty are calculated for the velocity and position, if this uncertainty is within 10σ the two seed halos are merged.

The halo position and velocity is found by averaging the positions and velocity for the lowest level seed halo that best minimizes the Poisson error. For the halo masses Rockstar calculates spherical overdensities, using all particles in all substructures. Rockstar unbinds particles using a single pass modified Barnes-Hut method to calculate the gravitational potential and then removes particles that are not gravitationally bound to the halo. The typical boundness value for a Rockstar run is 98%.

The master process divides the simulation into separate rectangular regions, and then sends each particle to the appropriate processor, along with particles in the corresponding ghost zones. Each processor then does 3D FOF calculations and each of the groups found this way are distributed further for phase-space analysis. A single FOF group is not split up into several tasks, so the minimum analysis time needed is equal to the analysis time of the largest FOF group, independently of the number of processors.

Chapter 6

MORPH Algorithms

The goal of this thesis have been to implement a halo finder that takes into account the effects of modified gravity. We will now take a look at the methods and algorithms used in MORPH and describe the new algorithms used to gain independence from the general relativity method used while performing the N-body simulation. The steps performed by MORPH are the following:

- Load the N-body data.
- Perform a friends-of-friends (FOF) grid method to split the dataset into FOF groups.
- Send the FOF groups to different slave processors for further refinement.
- Perform a 6D phase space FOF method.
- Calculate the desired statistics.
- Gather the results in the master processor.

6.1 Loading routines

The first step in any halo finder is to read the data from a N-body simulation into memory. The output of a N-body simulation are doubles describing each particle's position, velocity and acceleration, and if applicable other properties such as mass, potential energy and similar attributes, the exact output differs from code to code. A halo finders task is to gain meaningful information from such a dataset. Most N-body codes give the output in binary form, both to increase the speed and to reduce the size of the data files. Each N-body code uses its own format for storing the data and we must most likely write a new loading routine for MORPH when working on data from a new N-body code. MORPH uses 1D arrays to initialize objects, they are on the form in figure 6.1, where the numbers of the Halo arrays and Particle arrays are put into the appropriate array. The loading routine simply needs to generate such an array to be able to initialize the data. A typical N-body dataset looks like figure 6.2, where the position of each particle have been plotted.

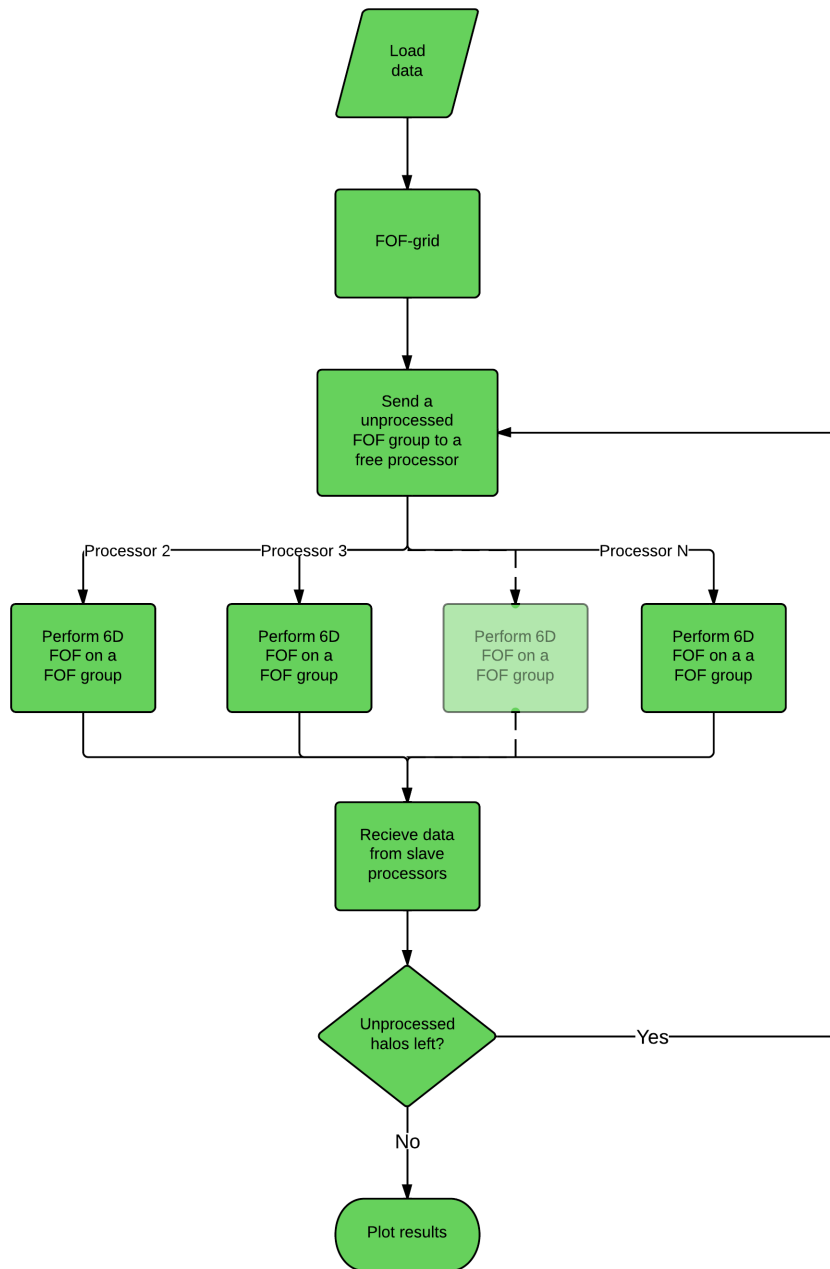


Figure 6.1: Illustration of array structure.

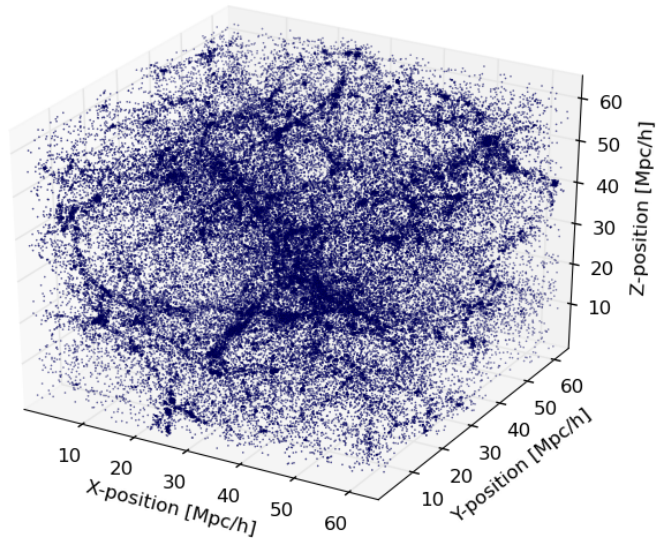


Figure 6.2: Plot of the positions of every 25th particle for the 128^3 particle dataset.

6.2 Parallelization

MORPH is designed with parallelization in mind, which impacts the algorithms chosen and introduces additional steps into the computations. A combination of a 3D FOF and a 6D FOF method have mainly been used. The naive way to perform the halo finding would be to directly run the 6D phase-space splitting, but the problem is that this algorithm scales as $O(N^2)$, so it would be horribly slow. This is where the idea for the parallelization enters. If we first run a 3D friends-of-friends algorithm to split the particles up into a set of independent 3D friends-of-friends groups (FOF groups) we can then send these FOF groups to different processors to execute the slow 6D phase-space splitting.

A single FOF group is never split up for analysis between different processors. The reason for this is that we already have done a 3D space friends-of-friends method and all particles might be bound to a single halo. Therefore all particles must be tested against all other particles. The analysis time for the largest FOF group thus sets a lower limit for the wall time of the 6D phase space splitting, this lower time limit is independent of the number of processors. Because of this we sort all friends-of-friends groups from highest number of particles to lowest and perform the phase space splitting for the FOF groups with the highest number of particles first. This ensures that we avoid the case where the phase space splitting is finished for all FOF groups except the largest, having one processor working while others are idle. This method helps to minimize the analysis time, but when the number of halos \gg number of processors this will generally not be a problem.

6.3 Friends-of-friends algorithms

Recall that the idea behind the friends-of-friends algorithm, as mentioned in section 5.2, is to connect particles that are close to another in either 3D space or 6D phase-space. It is a purely geometric method that links particles that is closer than a prescribed length, the linking length. There are two different methods to do this, first FOF separation, a non-grid routine¹ that scales as $O(N^2)$ and a much faster grid version that scales as $O(N \log N)$. The first routine was implemented as a test function, to make sure that both schemes provide the same answer. It is also easier to grasp the general idea behind the FOF algorithm by looking at the most basic version.

6.3.1 Friends-of-Friends N^2

The non-grid version is the simplest form of the friend-of-friend methods and have existed for a long time (Davis et al. 1985), but unfortunately it is also one of the slowest. The method is quite simple, for each particle we check whether all other particles are within a pre-specified linking length. If a particle is closer to the original particle than the linking length it is considered to belong to the same group. Then for all particles found in the first pass a new search is done, finding particles that are within the linking length of the newly added particles and attaching them to the same group. This process is repeated until no new particles are found. If the number of particles is larger than a given threshold, set to 20 particles, we save the linked particles as a friends-of-friends group, if not they are discarded. Once this is done the next particle that have not been tested or assigned to a FOF group is found and the process is repeated until all particles have been tested.

The linking length is chosen as a fraction b of the mean inter-particle distance, in MORPH this is calculated as the cube root of the mean particle volume:

$$\text{Linking length} = \frac{1}{\text{NrParticles}^3}. \quad (6.1)$$

The value for b is chosen to $b = 0.21$, see section 6.8.2 for how this was selected.

Pseudocode for the basic FOF algorithm can be found in Algorithm 2 and the method used to find the neighboring particles and add them to the halo in Algorithm 3. Since this FOF scheme scales as $O(N^2)$ it is horribly slow for larger simulations and is therefore nearly useless for any practical purposes.

6.3.2 Friends-of-Friends grid

The FOF grid algorithm is much faster than the standard FOF method. The idea is that we divide the particles into a grid and if the size of a single grid cell is larger than the linking length we only need to check all the 26 neighboring grid points as we are assured that particles in any other cell is further away than the linking length. This negates most of the unnecessary testing done in the N^2 method. We then do almost the same as in the basic algorithm, but with a few additional steps, this pseudo code can be found in Algorithm 4. In addition to creating the linked list over all particles at the start, we need to create a grid over all particles, with a grid size larger than the linking length. This

¹This algorithm is interchangeably called both non-grid and N^2

Algorithm 2 Pseudocode FOF-algorithm

Create linked list over all particles

while True **do**

 Find the next particle not already in a halo from the linked list

if No particles left in the linked list **then**

 Break loop

else

 //Find all the neighboring particles and add them to the halo

 findNeighbors(Particle, Halo)

if Particles in Halo > minimum number of particles **then**

 Save halo

end if

end if

end while

Algorithm 3 Pseudocode for findNeighbors(Particle, Halo)

function FINDNEIGHBORS(Particle, Halo)

 Add Particle to Halo

 Remove Particle from the linked list

for each particle in the linked list, TestParticle **do**

 CalculateDistance(Particle, TestParticle)

if Distance < Linking length **then**

 Add TestParticle to Halo

 Remove TestParticle from the linked list

end if

end for

for Each particle found as a neighbor in the previous process **do**

 findNeighbors(Particle, Halo)

end for

end function

grid goes from (-1,-1,-1) to (1,1,1) and all particle positions are scaled to this scale. The width of each cell is given by

$$\text{Width} = \frac{2}{\text{LinkingLength}}. \quad (6.2)$$

The grid is then populated, and the particles are inserted into their corresponding grid cell according to the particles position. Each grid cell contains a linked list of all particles belonging to that cell, making it fast to access and easy to remove and add particles. We have a new neighbor finding algorithm, see Algorithm 5. Since the FOF grid method is only used to divide the simulation into FOF groups that is going to be processed further, we use a slight modification of the original FOF algorithm. If a particle is found to have more than a given number of neighbors, set to 10 particles, we immediately drop the neighbor-finding process and instead calculate the neighboring particles for up to thrice the linking length, see section 6.3.2. All particles found within this distance are linked together as before. This means that neighbors must be calculated over a larger distance, but fewer of these calculations must be made. A comparison between different values of the scalefactor and the linking length was done to select these values, see below.

This also solves an additional problem that showed up for larger datasets, number of particles $\geq 512^3$. The neighbor-finding routine uses recursion to find the neighboring particles, but for large datasets it takes so many passes through the routine that the stack gets filled up with function calls, resulting in a segmentation fault. Because of this we choose a scaling factor > 1 . Since this routine now checks for an increased linking length in dense areas, fewer function calls are made in the recursive routine.

Algorithm 4 Pseudocode for FOF grid algorithm

```

Create linked list over all particles
Create a grid
Populate the grid with all particles
while True do
    Find the next particle not already in a halo from the linked list
    if No particles left in the linked list then
        Break loop
    else
        //Find all the neighboring particles and add them to the halo
        findNeighborsGrid(Particle, Halo)
        if Particles in Halo  $>$  minimum number of particles then
            Save halo
        end if
    end if
end while

```

We ran the FOF-grid algorithm on the test dataset with 128^3 particles and plotted every 25th particle, see figure 6.3. When compared to figure 6.2 it can be seen that many particles have been removed and that reasonable FOF groups have been found.

Algorithm 5 Pseudocode for findNeighborsGrid(Particle, Halo)

```
function FINDNEIGHBORSGRID(Particle, Halo)
  Add Particle to Halo
  Remove Particle from the linked list
  Get Particle position in Grid
  for Each neighboring Grid point do
    if Number of neighbors > 12 then
      Break
      Flag = True;
    end if
    for All particles in each Grid point, TestParticle do
      if Number of neighbors > 12 then
        Break
      end if
      CalculateDistance(Particle, TestParticle)
      if Distance < Linking length then
        Add TestParticle to Halo
      end if
    end for
  end for
  if Flag == True then
    for Each grid point within 2 cells do
      for All particles in each Grid point, TestParticle do
        CalculateDistance(Particle, TestParticle)
        if Distance < Linking length then
          Add TestParticle to Halo
        end if
      end for
    end for
  end if
  for Each particle found as a neighbor in the previous process do
    findNeighborsGrid(Particle, Halo)
  end for
end function
```

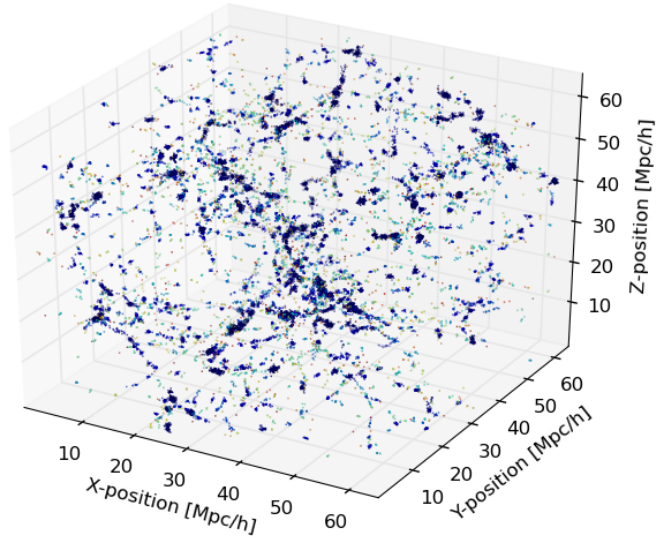


Figure 6.3: The positions of every 25th particle for the 128^3 particle dataset, after the 3D FOF algorithm

Scalefactor of the linking length and number of particles needed before scaling

We did a series of runs, using only the FOF grid algorithm, for the 128^3 particle dataset with different values for the scalefactor and number of particles needed before the linking length was scaled, see section 6.3.2, and got the wall time for each run. The results can be seen in figure 6.4. As seen, we obtain the best performance when the linking length is tripled if more than 10 particles are found as neighbors.

6.3.3 FOF N^2 vs FOF grid runtime

A comparison of the runtime for both the FOF- N^2 and FOF-grid method for different number of particles have been performed. A plot of the FOF- N^2 runtimes can be found in figure 6.5 and for the FOF-grid in figure 6.6. As can be seen from these plots the grid method is vastly superior to the N^2 method, as expected, and will therefore be the one used in MORPH.

6.4 6D phase-space splitting

As we saw from figure 6.3, we have already split our dataset into what seems like reasonable halos, so why do we need to perform an additional splitting routine? One problem that arises with only a 3D splitting is that we can get linking bridges between neighboring FOF groups. An example of such a configuration can be seen in figure 6.7. These are clearly

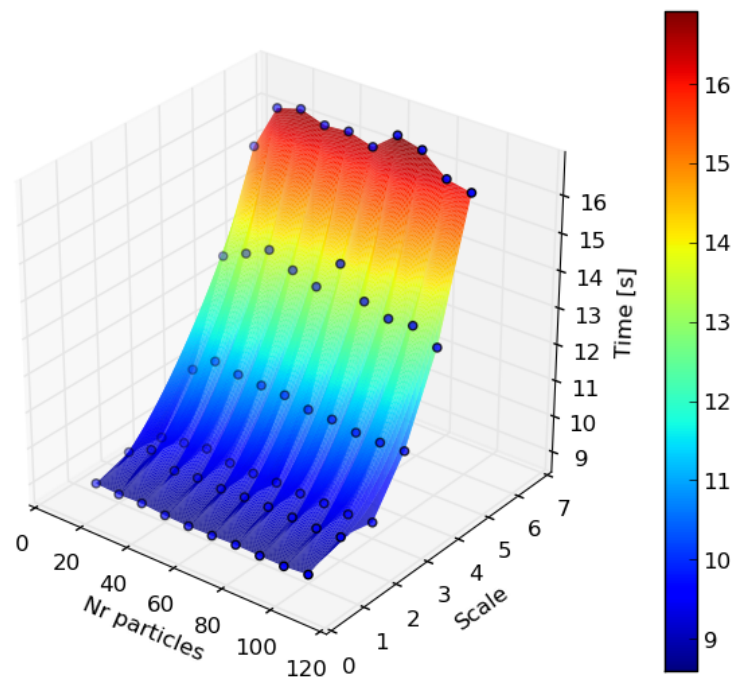


Figure 6.4: Runtime for MORPH using different parameters for the nr of particles needed for the scaling and the scaling itself.

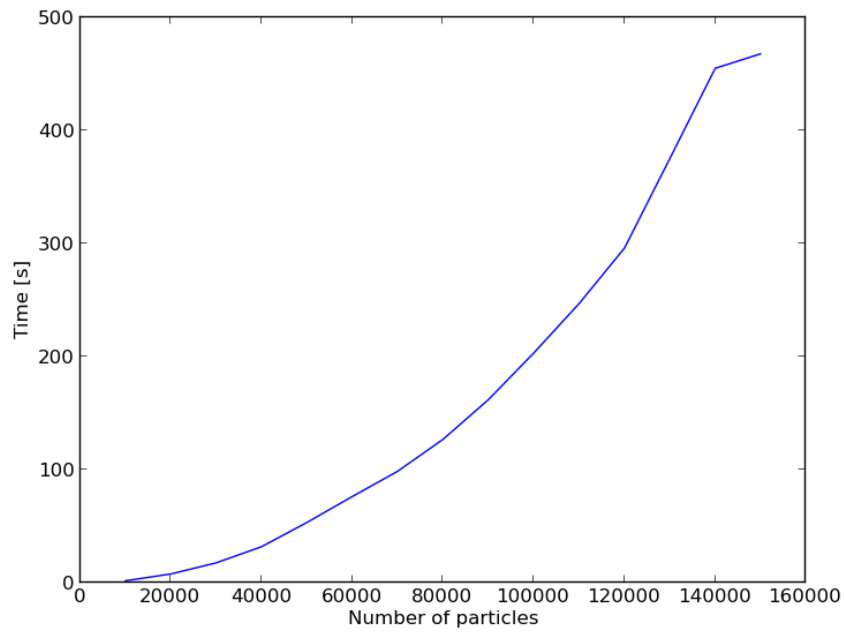


Figure 6.5: Runtime for the FOF-N² scheme for different number of particles.

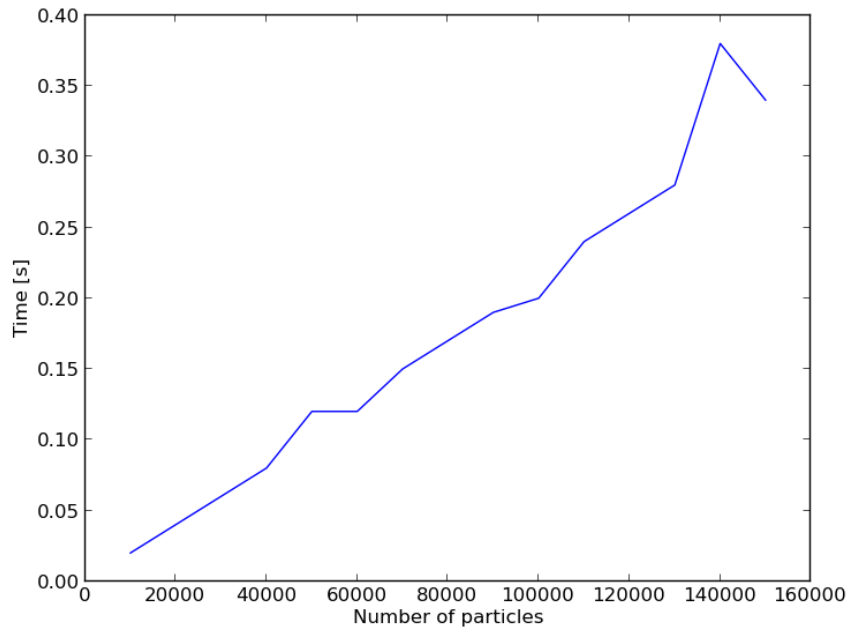


Figure 6.6: Runtime for the FOF-Grid scheme for different number of particles

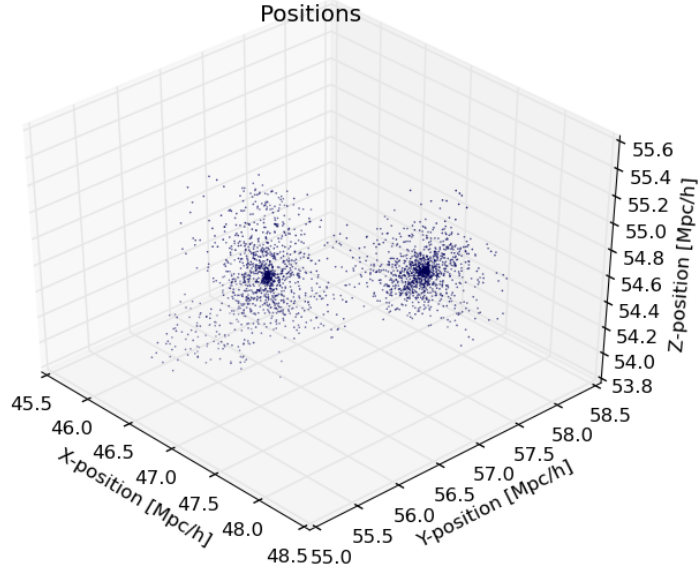


Figure 6.7: An example of a linking bridge affecting a FOF group.

Particles	Mass	x	y	z	V_x	V_y	V_z
16206	$1.5205 \cdot 10^{14}$	16.623	56.225	18.231	0.000989	0.017172	0.011939

Table 6.1: FOF group information,. Positions are given as Mpc/h, velocities as Mpc/Gyr and mass as M_\odot .

two distinct halos that have been connected through a linking bridge of a few particle and they need to be separated.

We have only considered the positional information in the 3D splitting, but we also need take into account the velocity of the halos. There might exist two independent halos that are moving through each other, where they would occupy the same place in position space, but not in velocity space. The 3D splitting would find such a configuration of halos to be a single halo. Such an example can be seen in figure 6.8, the positions do not indicate anything special, but looking at the velocity of the halo, figure 6.9, we see that there are indications of several halos. We see this spread in figure 6.10, which is three plots of position vs velocity in each direction (x,y,z). Especially the y vs v_y plot show signs of a differences in velocity space. Another characteristics left out by only doing a 3D splitting is the information on halo substructures.

The solution to this issue is to perform a 6D phase space FOF routine in order to separate possible halos in each 3D FOF group. If two halos are moving through each other they would be separated in phase space, since they have different velocities. The problem we have now is embarrassingly parallel, each FOF group found is independent of the others, so we can distribute each FOF group to separate processors for the 6D phase space splitting.

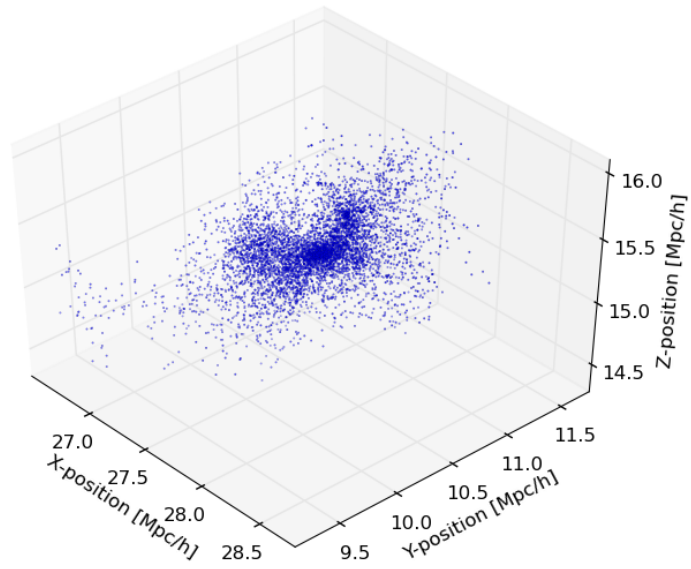


Figure 6.8: Plot of the positions of the particles in a single FOF group.

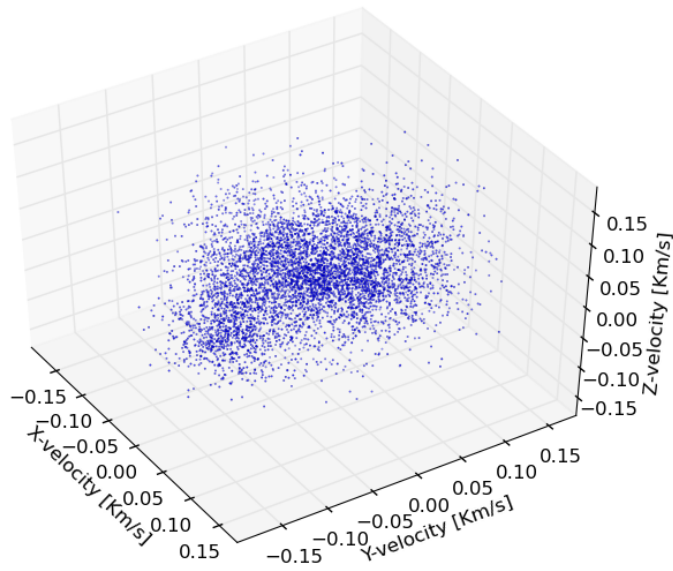


Figure 6.9: Plot of the velocities of the particles in a single FOF group, after the 3D FOF algorithm.

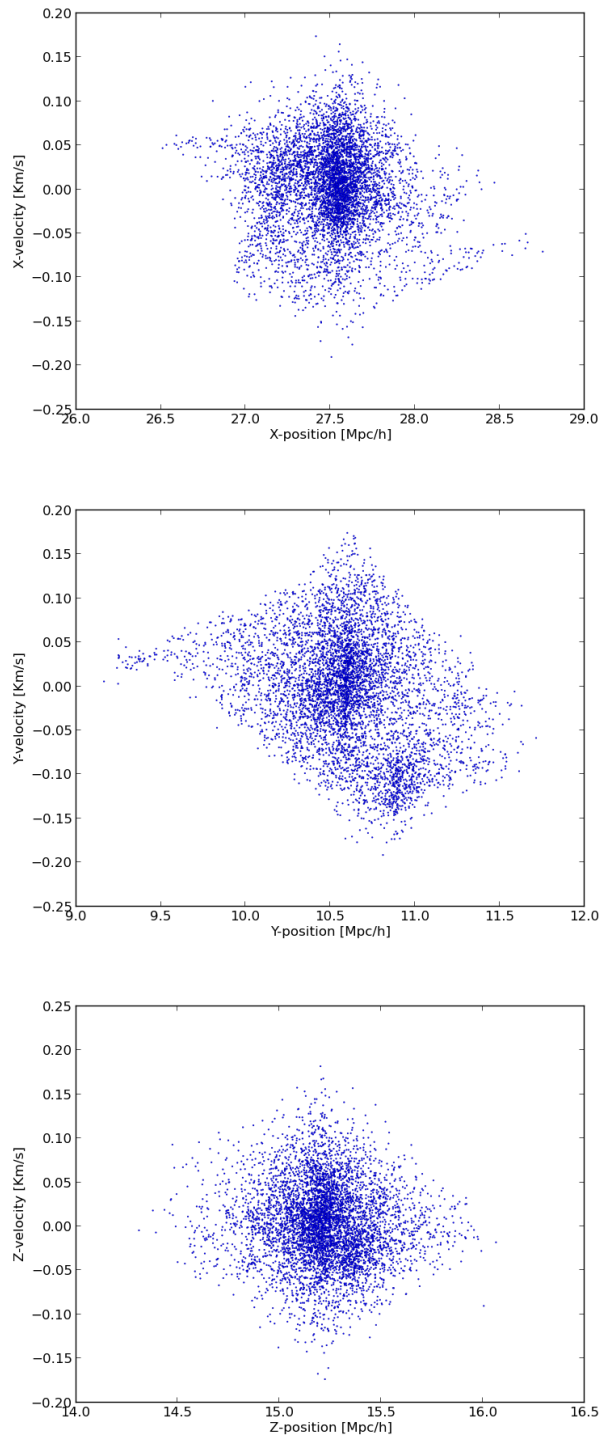


Figure 6.10: Plot of the x-position vs x-velocity, y-position vs y-velocity and z-position vs z-velocity for the particles in a single halo, after the 3D FOF algorithm.

f	Time
0.5	78.258
0.6	82.74
0.7	80.24
0.8	85.11
0.9	105.87

Table 6.2: Runtime for MORPH with $f = [0.5, 0.6, 0.7, 0.8, 0.9]$

6.4.1 6D phase space friends-of-friends algorithm

When the FOF group is received on a slave processor, we only need to process one FOF group at the time. A flowchart for the 6D phase space friends-of-friends method is depicted in figure 6.11.

Creating the substructure

The first order of business when wanting to split a FOF group into separate halos and subhalos is to create the substructure hierarchy. In this step MORPH uses an adaptive phase space linking length and creates a hierarchy of substructures with decreasing linking length, corresponding to tighter isodensity contours in the phase space density field. This gives us a way to distinguish between different levels of substructures in the main FOF group. First the phase-space distance to the closest neighbor is calculated for each particle. The phase-space distance between two particles, p_1 and p_2 , is defined as

$$d(p_1, p_2) = \left(\frac{|\vec{x}_1 - \vec{x}_2|^2}{\sigma_x^2} + \frac{|\vec{v}_1 - \vec{v}_2|^2}{\sigma_v^2} \right)^{1/2}, \quad (6.3)$$

here σ_x^2 and σ_v^2 are the particle and velocity dispersion for this specific FOF group. The phase space linking length is chosen such that a fraction f of all particles are linked together with at minimum one other particle, they have a closest neighbor. f is basically the fraction of particles kept from host group to subgroup. For large FOF groups, number of particles > 10000 , this operation can be very costly. This problem is mitigated by only calculating this linking length for a random 10 000 particle subset of the FOF group. This value can be changed if needed, but 10 000 particles are enough to determine the linking length to a reasonable precision without a too high computational cost. How to choose f depends upon two considerations. If f is too large, $f > 0.9$, the algorithm will run slower and might not find significant particle subgroups. If it on the other hand is too small, $f < 0.5$, it runs much faster, but the algorithm might not recognize smaller substructures that are of interest. We timed MORPH for $f = [0.5, 0.6, 0.7, 0.8, 0.9]$, the results can be found in table 6.2 while the results from the code are in figure 6.12. As seen from figure 6.12, changing f has no effect on the final results. From table 6.2 we see that we get the best runtime when $f = 0.5$, such that 50% of the particles have a closest neighbor.

For the host FOF group we first calculate the linking length using the above mentioned method. For each particle in the FOF group we check if all other particles are closer in phase space than the adaptively chosen linking length.

If a particle is closer, it is attached to the halo. Then for all particles found in the first pass we do a new search, finding particles that are within the linking length of the

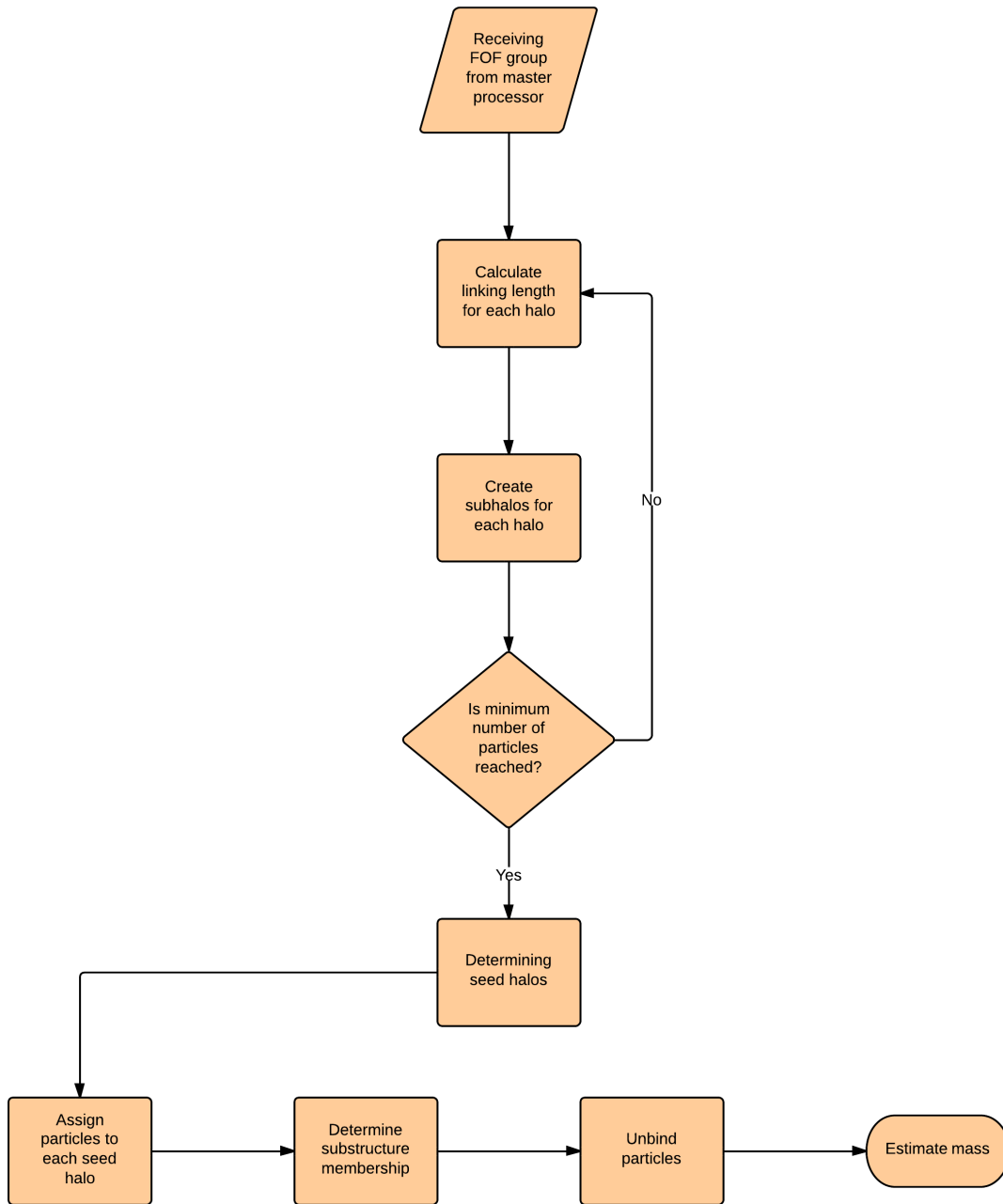


Figure 6.11: Flowchart of the 6D phase space friends-of-friends method.

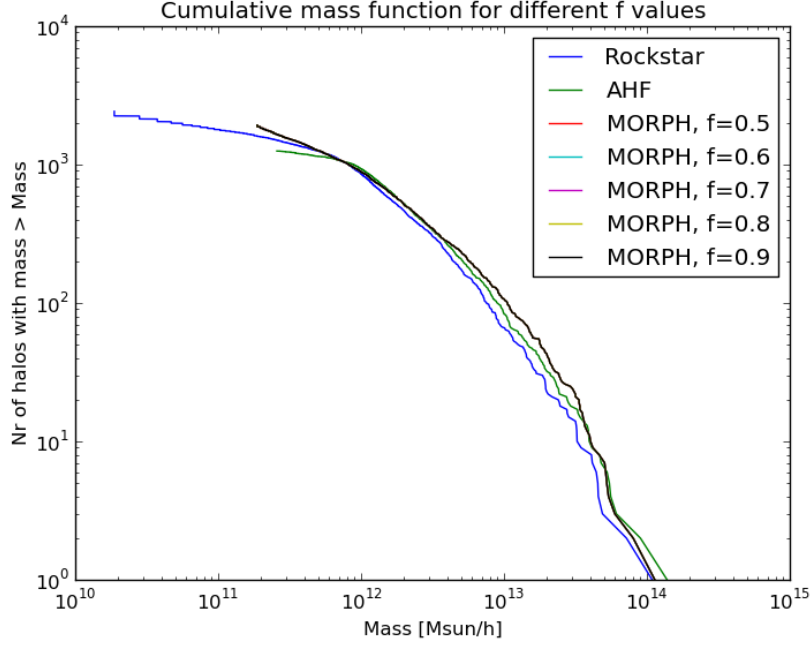


Figure 6.12: Mass functions for $f = [0.5, 0.6, 0.7, 0.8, 0.9]$.

newly added particles and attach them to the same halo. This process is repeated until no new particles are found closer than the linking length. If the number of particles is above a given threshold, set to 20 particles, we save the linked particles as a halo, if not, we discard them. Once this is done we find the next particle in the linked list not attached to a halo and repeat the procedure. When this method is finished we repeat this process for each subgroup. A new linking length is calculated, and the group finding process is repeated. This continues until we reach the minimum limit of particles in the deepest level of substructure hierarchy, which is set to 20 particles. Increasing this value would speed up the phase space finding as fewer recursions are needed before reaching the innermost halo. This might be interesting in the case where we only wish to look at larger halos or if a detailed view of the subhalo structure is not needed. After the substructure hierarchy is complete we calculate the statistics for the structures found, most importantly position and velocity. The pseudo code for these steps can be found in Algorithm 6 and 7.

Determining seed halos

The halos at the deepest levels in the FOF structure are considered seed halos, which are the spawn point of each "real" halo. We go through the hierarchy and create a list of all these seed halos and calculate the Poisson uncertainties in position and velocity. If two seed halos have position and velocity within 10σ of the uncertainties they are merged.

Algorithm 6 Pseudocode for SplitHalo()

```
function SPLITHALO
  if Number of particles < lower seed limit then
    Return
  end if
  FriendOfFriendPhaseSpace()
  Clean up all data
  for Each SubHalo found do
    SplitHalo()
  end for
end function
```

Algorithm 7 Pseudocode for FriendOfFriendPhaseSpace()

```
function FRIENDOFFRIENDPHASESPACE
  Calculate the statistics of the halo
  Create a linked list of all particles
  Calculate the new Linking Length for this halo
  while linked list != empty do
    Find the next Particle in the linked list
    Remove this Particle from the linked list
    //Find all neighboring Particles and add them to the halo
    findNeighborsPhaseSpace(Particle, Halo, LinkingLength)
    if Particles in Halo > minimum number of particles then
      Save halo
    end if
  end while
end function
```

Algorithm 8 Pseudocode for findNeighborsPhaseSpace(Particle, Halo, LinkingLength)

```
function FINDNEIGHBORSPHASESPACE(Particle, Halo, LinkingLength)
  Add Particle to Halo
  for each particle in the linked list, TestParticle do
    CalculateDistance(Particle, TestParticle)
    if Distance < Linking length then
      Add TestParticle to Halo
      Remove TestParticle from the linked list
    end if
  end for
  for Each particle found as a neighbor in the previous process do
    findNeighbors(Particle, Halo, LinkingLength)
  end for
end function
```

The uncertainties are calculated as

$$\mu_x = \frac{\sigma_x}{\sqrt{N}}, \quad (6.4)$$

$$\mu_v = \frac{\sigma_v}{\sqrt{N}}, \quad (6.5)$$

with σ_x and σ_v the position and velocity dispersion and N is the number of particles, all calculated from the smallest of the two halos. The criteria for merging is

$$\sqrt{\frac{(x_1 - x_2)^2}{\mu_x^2} + \frac{(v_1 - v_2)^2}{\mu_v^2}} < 10\sqrt{2}, \quad (6.6)$$

or, for numerical ease

$$\frac{(x_1 - x_2)^2}{\mu_x^2} + \frac{(v_1 - v_2)^2}{\mu_v^2} < 200. \quad (6.7)$$

If two halos are merged the process is stopped and rerun from start. This step is repeated until no halos are merged. By doing this we are ensured that several halos can be merged together, while we also prevent mergers that should not happen. It is possible to imagine a case were three halos would be merged, but restarting the algorithm might prevent a third merging.

Assigning particles to the seed halos

We have now found all seed halos in the FOF group, and all the other halos from the previous step can be discarded. The next part in determining the phase space halos is to remove all particles from the seed halos while keeping halo information intact, such as position, velocity, number of particles assigned to the seed halo and so on. Each particle is then reassigned to the closest seed halo in phase space, as given by equation 6.8. This distance is given by

$$d(\mathcal{H}, p) = \left(\frac{|\vec{x}_{\mathcal{H}} - \vec{x}_p|^2}{r_{vir}^2} + \frac{|\vec{v}_{\mathcal{H}} - \vec{v}_p|^2}{\sigma_v^2} \right)^{1/2}, \quad (6.8)$$

Here \mathcal{H} denotes the halo and p the particle. σ_v is the velocity dispersion of the halo, $M_{\mathcal{H}}$ is the mass of the seed halo and vir is the virial overdensity. For $M_{\mathcal{H}}$ we use the total mass of the particles originally found to be a part of the seed halo. Other halo finders, such as Rockstar (Behroozi et al. 2013) use the definition of vir from Bryan & Norman (1998), but this assumes standard gravity and could therefore provide the wrong results when considering modified gravity simulations. To avoid this potential problem we calculate R_{vir} as shown in section: 6.6. Using the halo radius R_{vir} ensures that particles assigned to subhalos cannot be too far from the main density peak even if they are close in velocity space. Intuitively, the largest effect is that velocity information becomes the controlling factor when distinguishing membership to halos when two halos are within each others virial radius. By doing this it is ensured that we can calculate the mass of the substructures accurately and independently of the choice of f . We also make sure that

particles belonging to a host halo are not mis-assigned to a subhalo, because the particles in the center of the host will be closer in phase space to the true host center than to any of the subhalo centers clustered around it. The pseudo code for assigning particles to a seed halo can be seen in Algorithm 9. After all particles have been assigned we loop over seed halos and remove those which have fewer particles than the lower limit, and reassign the particles belonging to these halos to the new closest seed halo.

Algorithm 9 Pseudocode for assignParticles()

```

function ASSIGNPARTICLES
  Remove all particles from each seed halo
  for All each Particle do
    for All SeedHalos do
      Calculate phase space distance between SeedHalo and Particle
    end for
    Find the minimum phase space distance
    Assign Particle to the closest SeedHalo
  end for
  Remove halos with fewer particles than the halo limit.
  Add the particles that were removed to the now closest SeedHalo
end function

```

Determining substructure membership

We are now required to determine the full subhalo structure to be able to obtain the correct mass. The standard definition of a subhalo is as follows:

- a halo contained within another, larger halo.

However, what "contained" entails is not well defined. The mass of a halo often includes the mass of substructures, so if mass is used as a deciding factor for which halo is the largest an interesting problem is encountered. We need to know the mass before determining the substructure, but before the mass can be calculated we need to know the substructures. This problem is inter dependable is omitted by using instead the number of particles assigned to each halo. Each halo is assigned as a subhalo of the closest largest halo, in phase space. The phase space distance is calculated using equation 6.3, treating the halo centers as particles. The pseudo code for doing this can be found in Algorithm 10. However, there is a possibility for refinement here that have not been implemented. If the N-body simulation contains data from previous time steps it is possible to perform the same process for this step. Then it could be determined which halo in the previous time steps has the largest amount of particles contributing to a given halo's particles, at the present time step. Host-subhalo relationships could then be rearranged, if necessary, to correspond with the host-subhalo relationships in the previous step. However we have not been able to obtain such a dataset.

Algorithm 10 Pseudocode for determining substructure membership

Sort all seed halos from smallest to largest number of assigned particles

if Number of seed halos == 1 **then**

 Copy the seed halo into the halo structure

 Return

end if

for All seed halos, SeedHalo, in the sorted list **do**

for All seed halos in the sorted list, SeedHalo2, beginning at SeedHalo **do**

if SeedHalo < SeedHalo2 **then**

 Calculate the phase space distance between SeedHalo and SeedHalo2

end if

end for

 Find the seed halo with minimum phase space distance

if Not the largest seed halo **then**

 Assign SeedHalo as a subhalo of the seed halo with minimum phase space distance

end if

end for

Copy the largest seed halo into the halo structure

Remove all seed halos

6.5 Unbinding

An important step in all halo finders is the unbinding procedure (Knebe et al. 2013). In this step particles that are not gravitationally bound to a halo are removed. If a particle has a velocity larger than the escape velocity of the halo, it should be considered unbound since it will eventually escape the halo. The escape velocity of an object in isolation in a gravitational potential is given by

$$v_{esc} = \sqrt{2|\phi|}, \quad (6.9)$$

where ϕ is the gravitational potential of the object.

The need for an unbinding procedure is tightly linked to the particle collection method. 3D space halo finders need to perform an unbinding procedure since they find several false positives, particles that are within the borders of a halo, but have a velocity much too high to be gravitationally bound to the object. They need to do this no matter how conservative their particle collection methods are. By including the unbinding step, every 3D space halo finder is converted to a mock phase space halo finder. 6D phase space halo finders do not have the same need for an unbinding process, as they already take into account the velocity of each particle in their initial particle collection, but it is still recommended (Knebe et al. 2013). The need for unbinding also depends on the physical problem studied. Some properties of halos, such as the halo mass, are unresponsive to a few particles that should not be there, while others, for example halo spin, are highly sensitive to these few extra particles.

There are many different algorithms on how to both unbind particles and calculate ϕ . Most halo finders use a tree for this calculation while others simplify the problem by assuming for instance spherical symmetry (Knollmann & Knebe 2009). In the unbinding

process there are codes that remove a single particle before reiterating the process, others eliminate all particles in one go while some remove particles until a certain fraction of particles have been eliminated. There are also differences in the termination criteria, some stop when no more particles are to be removed and others when a certain fraction of particles have been unbound.

6.5.1 Calculating ϕ using a spherical approximation

Here we use the same method as Knollmann & Knebe (2009) to calculate ϕ , we also assume each halo is in isolation, which is an assumption all halo finders make. To get ϕ we integrate Poisson's equation, assuming standard general relativity,

$$\nabla^2\phi = 4\pi G\rho. \quad (6.10)$$

We assume spherical symmetry, which gives $\rho(\phi, \theta, r) = \rho(r)$ and $\phi(\phi, \theta, r) = \phi(r)$.

$$\Delta\phi = \frac{1}{r} \frac{d}{dr} \left(r^2 \frac{d\phi}{dr} \right), \quad (6.11)$$

$$\frac{1}{r} \frac{d}{dr} \left(r^2 \frac{d\phi}{dr} \right) = 4\pi G\rho(r), \quad (6.12)$$

$$\frac{d}{dr} \left(r^2 \frac{d\phi}{dr} \right) = 4\pi r G\rho(r), \quad (6.13)$$

integrating both sides

$$r^2 \frac{d\phi(r)}{dr} - \left[r^2 \frac{d\phi(r)}{dr} \right]_{r=0} = 4\pi G \int_0^r \rho(r') r'^2 dr', \quad (6.14)$$

$$r^2 \frac{d\phi(r)}{dr} - \left[r^2 \frac{d\phi(r)}{dr} \right]_{r=0} = GM(< r), \quad (6.15)$$

$$r^2 \frac{d\phi(r)}{dr} \propto M(< r). \quad (6.16)$$

The last equation shows that for $r \rightarrow 0$ then $\frac{d\phi(r)}{dr} \rightarrow 0$, which gives

$$r^2 \frac{d\phi(r)}{dr} = GM(< r), \quad (6.17)$$

$$\frac{d\phi(r)}{dr} = \frac{GM(< r)}{r^2}. \quad (6.18)$$

Integrating both sides again

$$\phi(r) = \int_0^r \frac{GM(< r')}{r'^2} dr' + \phi_0. \quad (6.19)$$

The integration constant, ϕ_0 , is found by requiring that $\phi(\infty) = 0$, using the assumption that the halo is in isolation,

$$\phi(\infty) = \int_0^\infty \frac{GM(< r')}{r'^2} dr' + \phi_0. \quad (6.20)$$

We split this integral into two separate parts, at R_{vir}

$$= \int_0^{R_{vir}} \frac{GM(< r')}{r'^2} dr' + \int_{R_{vir}}^{\infty} \frac{GM(< r')}{r'^2} dr' + \phi_0, \quad (6.21)$$

we assume that the halo is truncated at R_{vir}

$$= \int_0^{R_{vir}} \frac{GM(< r')}{r'^2} dr' + GM_{vir} \int_{R_{vir}}^{\infty} \frac{1}{r'^2} dr' + \phi_0, \quad (6.22)$$

$$= \int_0^{R_{vir}} \frac{GM(< r')}{r'^2} dr' + G \frac{M_{vir}}{R_{vir}} + \phi_0, \quad (6.23)$$

which gives

$$\phi_0 = - \int_0^{R_{vir}} \frac{GM(< r')}{r'^2} dr' - G \frac{M_{vir}}{R_{vir}}, \quad (6.24)$$

We now need to solve these integrations numerically. One problem is that the distance between two neighboring particles is random. Using a simple trapezoidal rule we get an integral of a discrete function $f(r)$, henceforth using the notation $f(r_i) = f_i$, which with irregular spacing becomes:

$$I = \sum_{i=1}^N \left(f_i (r_i - r_{i-1}) + \frac{1}{2} (f_i - f_{i-1}) (r_i - r_{i-1}) \right), \quad (6.25)$$

which gives us

$$I = \sum_{i=1}^N \frac{1}{2} (3f_i - f_{i-1}) (r_i - r_{i-1}). \quad (6.26)$$

In this case the function is given by

$$f_i = \frac{M(< r_i)}{r_i^2}, \quad (6.27)$$

and the integral from 0 to R_{vir} becomes

$$I = \sum_{i=1}^{r_i < R_{vir}} \frac{1}{2} \left(3 \frac{M(< r_i)}{r_i^2} - \frac{M(< r_{i-1})}{r_{i-1}^2} \right) (r_i - r_{i-1}). \quad (6.28)$$

We implemented this spherical approximation in our code, with pseudo code listed in Algorithm 11.

The disadvantages with this method is that we assume the halo is spherical, that we have standard general relativity and that the halos are in isolation. These assumptions are not expected to work well with simulation data from modified gravity theories.

Algorithm 11 Pseudocode for calculating ϕ , using a spherical approximation.

```
Calculate the virialization radius
Sort all particles with regards to distance from center of halo, stored in r
Mass = 0
resPhi0 = Mvir/Rvir
prevPhi0 = resPhi0
for All particles do
    if distance from center of halo > Rvir then
        break
    end if
    Mass += MassOfParticle
    Phi0 = Mass/(r[i]*r[i])
    resPhi0 += 0.5*(3*Phi0 - prevPhi0)*(r[i]-r[i-1])
    prevPhi0 = Phi0;
end for
resPhi0 *= -GravitationalConstant
Mass = 0
Phi[0] = resPhi0
previousPhi = resPhi0
for All Particles do
    Mass += MassOfParticle
    currentPhi = GravitationalConstant*Mass/(r[i]*r[i])
    Phi[i] = 0.5*(3*currentPhi - previousPhi)*(r[i]-r[i-1]) + Phi[i-1]
    previousPhi = currentPhi
end for
```

6.5.2 Using ϕ using the acceleration of each particle

Another method for unbinding the particles is to utilize the individual particle accelerations. A collisionless system, such as a cosmological N-body simulation, is described by the collisionless Boltzmann equation (CBE)

$$\frac{df}{dt} = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial \mathbf{x}} \cdot \dot{\mathbf{x}} + \frac{\partial f}{\partial \dot{\mathbf{x}}} \cdot \ddot{\mathbf{x}} = \mathbf{0}, \quad (6.29)$$

with $f(\mathbf{x}, \dot{\mathbf{x}}, t)$ being a probability density function defined:

$$dN = f(\mathbf{x}, \dot{\mathbf{x}}, t) d^3\mathbf{x} d^3\dot{\mathbf{x}}. \quad (6.30)$$

The potential energy can be derived from the CBE, as shown by Max Grönke (2013),

$$W = \int_V d^3\mathbf{x} \rho \mathbf{x} \cdot \ddot{\mathbf{x}}. \quad (6.31)$$

When we have a set of discrete particles the potential for a particle p_i becomes

$$W_i = m_i (\mathbf{x}_i - \mathbf{x}_H) \cdot \ddot{\mathbf{x}}_i. \quad (6.32)$$

The escape velocity is obtained when the total energy at ∞ is 0.

$$E_i = \frac{1}{2} m_i v_i^2 + W_i > 0, \quad (6.33)$$

$$\frac{1}{2} v_i^2 > |(\mathbf{x}_i - \mathbf{x}_H) \cdot \ddot{\mathbf{x}}_i|. \quad (6.34)$$

This gives the following escape velocity

$$v_i > \sqrt{2 |(\mathbf{x}_i - \mathbf{x}_H) \cdot \ddot{\mathbf{x}}_i|}, \quad (6.35)$$

for numerical ease, the requirement becomes

$$v_i^2 > 2 |(\mathbf{x}_i - \mathbf{x}_H) \cdot \ddot{\mathbf{x}}_i| \quad (6.36)$$

In the modified gravity regime we have an additional acceleration, the fifth force acceleration, which is included in

$$\ddot{\mathbf{x}} = \ddot{\mathbf{x}}_{\text{Newtonian}} + \ddot{\mathbf{x}}_{\text{fifth force}}. \quad (6.37)$$

We use the relative position of each particle since we wish to compare the escape velocity to the halo center. As long as the N-body code has implemented modified gravity, the halo finder will be independent of the specific modified gravity model. The only thing requires from the N-body code would is to output acceleration along with the other data. This method has a couple of advantages compared to other methods. The first being that it does not assume spherical symmetry, it is completely independent of the shape of the halos and all that is needed is the position of the center of the halo. Second, and more important, the method does not assume standard general relativity, and there is thus no need to make changes in the code to accommodate a specific modified gravity model.

6.5.3 Using ϕ from the N-body simulation

The output from the N-body code contains the potential at the position of each particle, so it would be possible to use this to perform the unbinding. However, the problem with this approach is that there is a problem with the scaling of the potential, the potential no longer approach zero at large distances since it does not exist in isolation. When we approach larger larger radii we expect to enter a new halo, because halos exist within each other's potential wells. The potential of several halos numbered [50, 150, 200, 300, 500, 1000, 5000, 10000, 15000], sorted from largest to smallest mass using lcdm1 dataset without unbinding, was plotted against the radius in figure 6.13. From this, different potential wells can be seen to coexist within the same halo. The escape velocity in the case of using the potential from the N-body code can be found by

$$E_i = \frac{1}{2}m_i v_i^2 + W_i > \mathcal{C}. \quad (6.38)$$

The challenge is to determine \mathcal{C} , especially since the environment is non symmetric, so \mathcal{C} would be dependent of the direction. One way to calculate the normalization is to assume that the central particles of the halo are independent of the environment and then use direct summation to calculate the potential of these particles. The N-body potential is then normalized using the direct summation values. But even after the normalization, the N-body potential differs from the direct summation potential, see Max Grönke (2013).

Since we have better methods to calculate the potential, see section 6.5.2, this method is not fully implemented².

6.5.4 Unbounding

The unbounding in itself is straight forward once ϕ or the criteria for the unbinding has been calculated. We loop through each particle in the halo and remove it if

$$(v_{\text{Particle}} - v_{\text{Halo}})^2 > \text{chosen criteria}. \quad (6.39)$$

If more than a given amount of particles are removed, currently set to 0, either ϕ or the criteria is recalculated and the unbinding procedure repeated. If the number of particles removed is below this limit, the unbinding procedure is considered finished. Here MORPH differs from Rockstar, where they only perform a single pass of the unbinding routine.

6.6 Calculating R_{vir}

We introduce a new way to calculate the virial radius (Shaw et al. 2006). The standard method (Bryan & Norman 1998),

$$R_{vir} = \left(\frac{M_{\mathcal{H}}}{\frac{4}{3}\pi\rho_{vir}} \right)^{1/3}, \quad (6.40)$$

²On the other hand, a simple unbinding method using the potential directly from the N-body code, without any normalization, have been implemented.

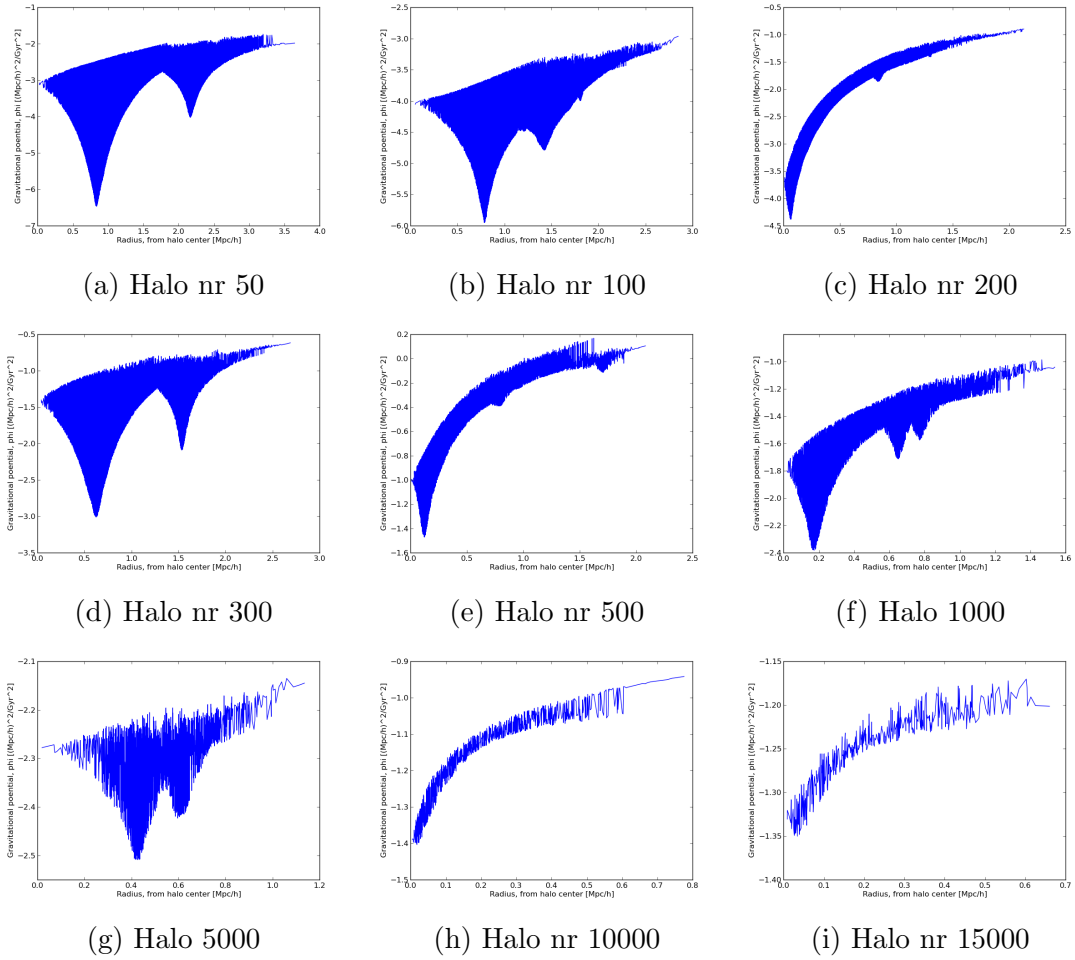


Figure 6.13: Potential from the N-body potential for different halos after the 6D phase-space splitting.

does not take into account effects of modified gravity, and is therefore incorrect when it comes to modified gravity N-body simulations. The new method use properties of the virialization constant β_{vir} . We make use of the well known virial equation

$$\frac{1}{2} \frac{d^2 I}{dt^2} = 2T + W - E_S, \quad (6.41)$$

where T is the kinetic energy, E_S the surface pressure term and W the potential energy. For a halo positioned in empty space the surface pressure term is zero (Shaw et al. 2006).

An object is defined to be fully virialized when

$$\frac{1}{2} \frac{d^2 I}{dt^2} = 0, \quad (6.42)$$

giving us

$$2T - E_S = -W, \quad (6.43)$$

which is the well known virial theorem.

We introduce a virialization constant β_{vir} , similar to how Shaw et al. (2006) does. This constant is given by

$$\beta_{vir} = \frac{2T - E_S}{W} + 1. \quad (6.44)$$

β_{vir} measures the relaxation level of an object. Stable gravitational objects are virialized and β_{vir} helps us find the border between bound objects and unbound objects. A halo is defined to be sufficiently virialized as soon as $|\beta_{vir}|$ falls below a given threshold, which is set to 0.2 (Shaw et al. 2006), and this method is used to obtain the virialization radius. One advantage with this method is that the virialization radius by definition is where the halo changes from virialized to unvirialized. We are now required to calculate the different terms of equation 6.44.

Kinetic energy, T

Calculating the kinetic energy of a halo is straightforward,

$$T = \frac{1}{2} \sum_{i=1}^N m_i |\mathbf{v}_i - \mathbf{v}_H|^2, \quad (6.45)$$

where the velocity of a particle i is \mathbf{v}_i , relative to the halo velocity \mathbf{v}_H .

Potential energy, W

There exists several different ways to calculate the potential energy of a halo

1. Direct summation.
2. Summing up the gravitational potentials at each particle position.
3. Using the particle acceleration.

The first method is computationally very expensive, scaling as $O(N^2)$. As such, it is not a good way to calculate the potential energy as long as there exist other valid methods.

Some N-body codes provide the gravitational potential at the position of each particle. This can be used to find the total potential energy of a halo,

$$W = \sum_{i=1}^N m_i \phi_i. \quad (6.46)$$

As previously mentioned, this method differs from the direct summation (Max Grönke 2013).

The method using the particles acceleration has a couple of advantages over the other two. First, it is computationally inexpensive to calculate and secondly, it takes into account any fifth forces from modified gravity theories, as long as they are incorporated into the acceleration in the N-body simulation. In addition, acceleration is an output we can obtain from N-body codes. The potential energy is, as previously shown,

$$W = \sum_{i=1}^N m_i (\mathbf{x}_i - \mathbf{x}_H) \cdot \ddot{\mathbf{x}}_i. \quad (6.47)$$

This method is the one used in MORPH.

Surface pressure term, E_S

Any particle that is bound to a halo outside the virial radius is by standard conventions not included when we calculate the kinetic and gravitational potential energy for the halo, but they still contribute significantly to the pressure at the boundary. Taking this into account we incorporate a surface pressure term in the virial theorem (Chandrasekhar 1961).

$$E_S = \int P_S(r) \mathbf{r} \cdot d\mathbf{S}. \quad (6.48)$$

The surface pressure is often ignored, since this term is zero for halos in empty space. But for halos with particles clustered around it, or where the border of the halo is arbitrarily set to be at a given radius, it is not the case. The pressure term, P_S can be approximated using the ideal gas law

$$P_s = \frac{1}{3} \frac{\sum_{i=0}^N m_i v_i^2}{V_{R_{vir}-0.8}}. \quad (6.49)$$

Here the volume $V_{R_{vir}-0.8}$ is the volume occupied by the outermost 20% of the particles

$$V = \frac{4\pi R_{vir}^3}{3} - \frac{4\pi R_{0.8}^3}{3} = \frac{4\pi R_{vir}^3}{3} (1 - 0.8^3). \quad (6.50)$$

The radius of the innermost particle in this shell is labeled as $R_{0.8}$ and the outermost as R_{vir} . The surface pressure term can now be approximated as

$$E_s \approx 4\pi R_{0.9}^3 P_s. \quad (6.51)$$

$R_{0.9}$ is the median radius between $R_{0.8}$ and R_{vir} . The effect of including this surface pressure term is to shift the distribution of β_{vir} towards smaller values (Shaw et al. 2006).

6.6.1 Algorithm

We proceed to determine at which radius the halo transitions from virialized to unvirialized. This is achieved by calculating β_{vir} for a number of different radii and observe for which radius β_{vir} falls below the threshold. We divide the halo into a number of spherical shells, set to 200, and calculate β_{vir} for each shell. There is a minimum number of particles needed per shell, set to three and if there are fewer particles than this, the total number of shells is reduced in order to retain the lower limit. We now calculate β_{vir} for each shell and if it falls below the threshold this is defined to be the virialization radius. If β_{vir} never falls below this threshold we set the virialization radius to be equal to the radius of the furthest particle. R_{vir} must be larger, but since there are no other particles bound to the halo we cannot have a larger radius.

There are two different ways to split up the radii for the different shells. The first is to split the distance into equal parts, which leads to a large amount of empty shells, again leading to numerous wasted computations. The other is to go by number of particles per shell, making the increase in particles constant. The advantage with this method is that it is adaptive as there are more shells in denser areas of the halo. Because of this the second option is chosen where a pseudo code for calculating β_{vir} is shown in Algorithm 12.

6.7 Results of the 6D phase space friends-of-friends algorithm

Figure 6.14 is the plot of a halo before and after the 6D phase-space splitting, the different colors corresponding to different subhalos. As we can see the phase-space FOF algorithm successfully isolates different subhalos. A plot of each separated halo are depicted in figure 6.15, and a table with statistics can be found in table 6.3. We note there are a three larger halos, Halo 0, 4 and 9, which corresponds to the three main bulges. As such this algorithm proved the result we intuitively would expect.

6.8 Mass estimation

The mass estimation MORPH uses is the total mass of all particles assigned to one halo in addition to the mass of all subhalos. The mass of particles linked together by the FOF algorithm is not very well interpreted theoretically, but extremely easy algorithmically. An advantage is that it is independent of the cosmological model used in the N-body simulation.

6.8.1 Mass function

The cumulative mass function is a plot of the number of halos with a mass larger than a given mass, most often plotted as a log-log plot. This is the main statistics used to verify MORPH and to test whether modified gravity models should be accounted for in halo finders when working on modified gravity datasets.

Algorithm 12 Pseudocode for calculating β_{vir} .

```
if Number of shells > NrParticles/minimum particles per shell then
    Shells = NrParticles/minimum particles per shell
else
    Shells = Number of shells;
end if
if Shells == 0 then
    Particles per shell = NrParticles;
else
    Particles per shell = NrParticles/Shells;
end if
for Each shell, n do
    R = r[n*Particles per shell]
    Calculate  $\beta_{vir}(R)$ 
    if  $\beta_{vir}(R) \leq 0$  then
        Rvir = R
        break
    end if
end for
if Rvir not found then
    Rvir = radius for the furthest away particle
end if
Mvir = 0;
for each Particle, i do
    if r[i] > Rvir then
        break
    end if
    Mvir += Mass of Particle;
end for
```

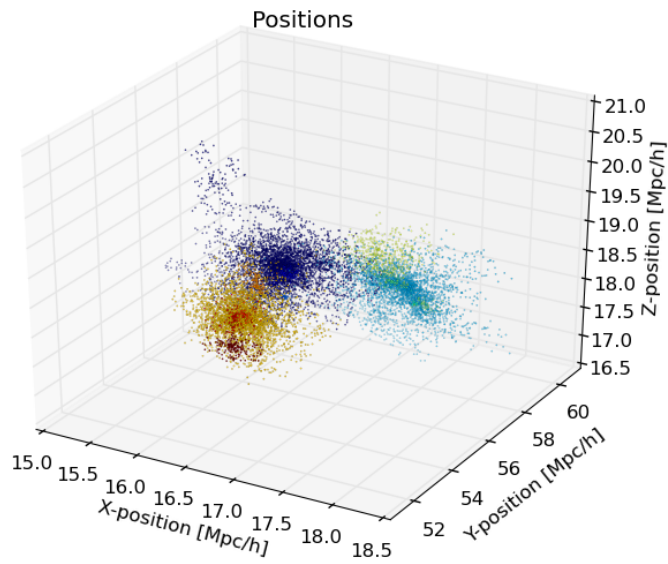
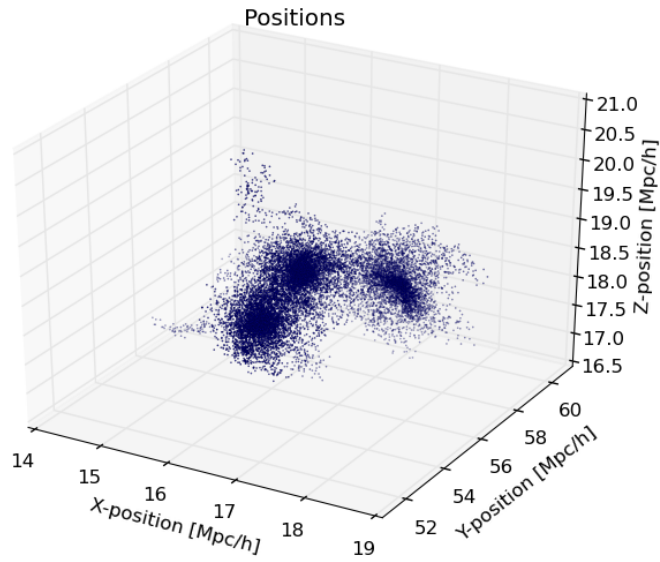


Figure 6.14: Positions of the particles in a single FOF group, after the 3D FOF algorithm (top) and after the 6D splitting (bottom).

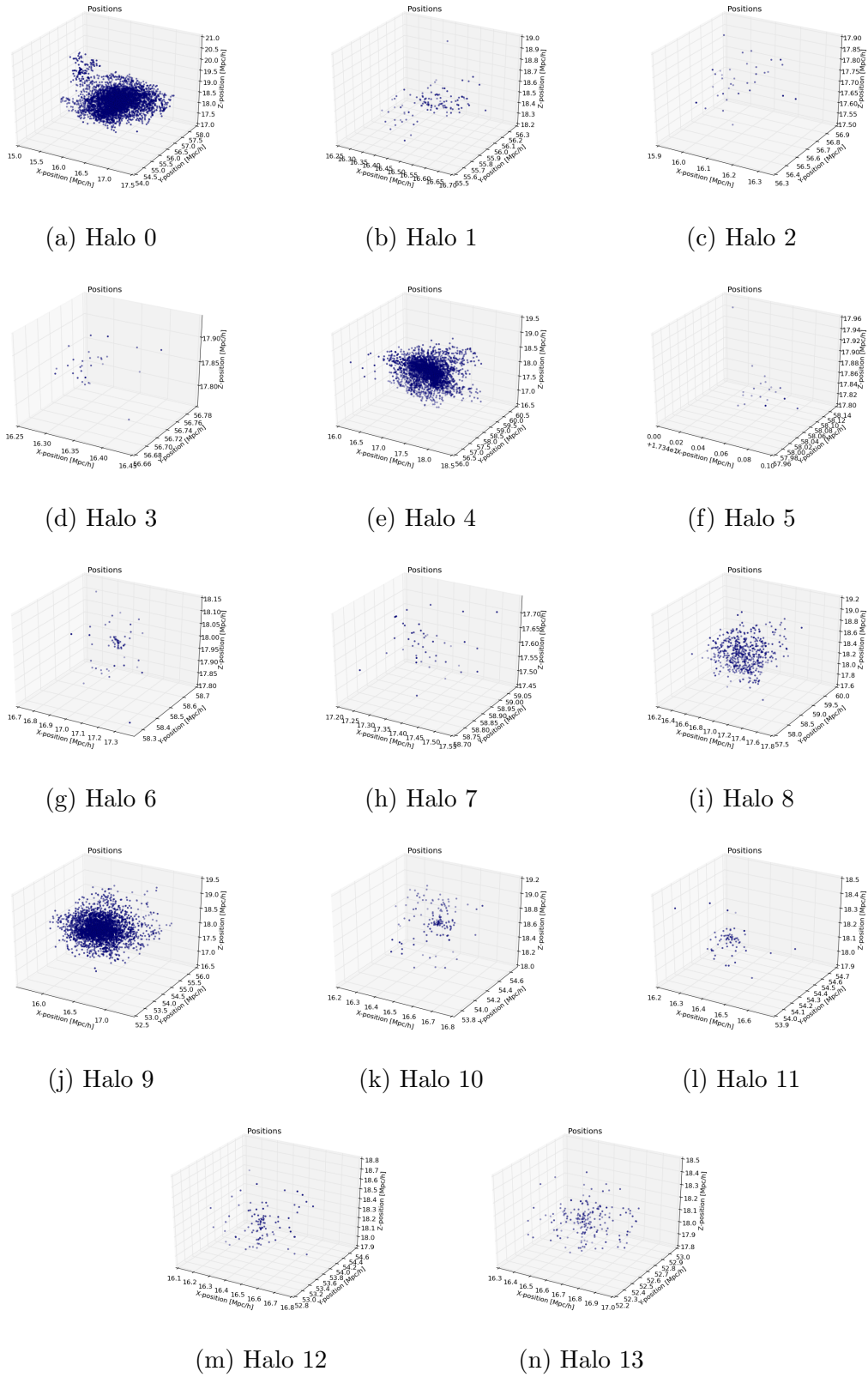


Figure 6.15: Plot of the positions of the particles for each halo in a single FOF group, after the 6D FOF algorithm.

Halo	Particles	Mass	x	y	z	V_x	V_y	V_z
0	4784	$1.5200 \cdot 10^{14}$	16.105	56.191	18.3858	0.0172	0.0021	-0.0024
1	113	$1.0602 \cdot 10^{12}$	16.491	55.900	18.4842	0.0282	0.0440	0.0164
2	35	$6.0986 \cdot 10^{11}$	16.064	56.645	17.6798	0.0157	0.0450	-0.0592
3	30	$2.8148 \cdot 10^{11}$	16.313	56.711	17.8574	0.0337	0.0427	-0.0594
4	3498	$4.4558 \cdot 10^{13}$	17.259	58.223	17.9371	-0.0208	-0.0107	0.0347
5	20	$1.8765 \cdot 10^{11}$	17.395	58.079	17.8372	-0.0280	-0.0468	0.0661
6	62	$5.8173 \cdot 10^{11}$	17.070	58.509	18.0020	0.0503	-0.0484	0.0061
7	46	$4.3160 \cdot 10^{11}$	17.345	58.903	17.6021	0.0044	-0.0460	0.1043
8	532	$7.4872 \cdot 10^{12}$	16.869	58.774	18.4347	-0.0387	-0.0143	0.0164
9	3578	$4.4043 \cdot 10^{13}$	16.304	53.710	18.1195	-0.0206	0.0478	0.0297
10	136	$1.8953 \cdot 10^{12}$	16.523	54.304	18.6643	-0.0123	-0.0101	0.0146
11	66	$6.1925 \cdot 10^{11}$	16.396	54.210	18.1742	0.0136	0.0186	0.0177
12	99	$3.7249 \cdot 10^{12}$	16.481	53.326	18.3501	0.0075	0.0865	-0.0496
13	184	$1.7264 \cdot 10^{12}$	16.627	52.600	18.1007	-0.0016	0.0955	-0.0043

Table 6.3: 6D split FOF group halo information,. Positions are in units of Mpc/h, velocities in Mpc/Gyr and mass in M_\odot .

6.8.2 Choosing b

Lastly we need to determine the best choice for parameter b , introduced in section 6.3.1. The most common is to set $b = 0.2$, while Rockstar uses a larger value of $b = 0.28$. Because of this we want to test MORPH for $b = [0.2 - 0.3]$, in 0.01 increments, on the lcdm1 dataset to see which value for b gives the best fit with Rockstar. The results are in figure 6.16 and the best fit is for $b = 0.21$, see figure 6.17. This analysis was done using the accelerated unbinding as this is the unbinding method that is independent of the gravity model, see section 6.5.

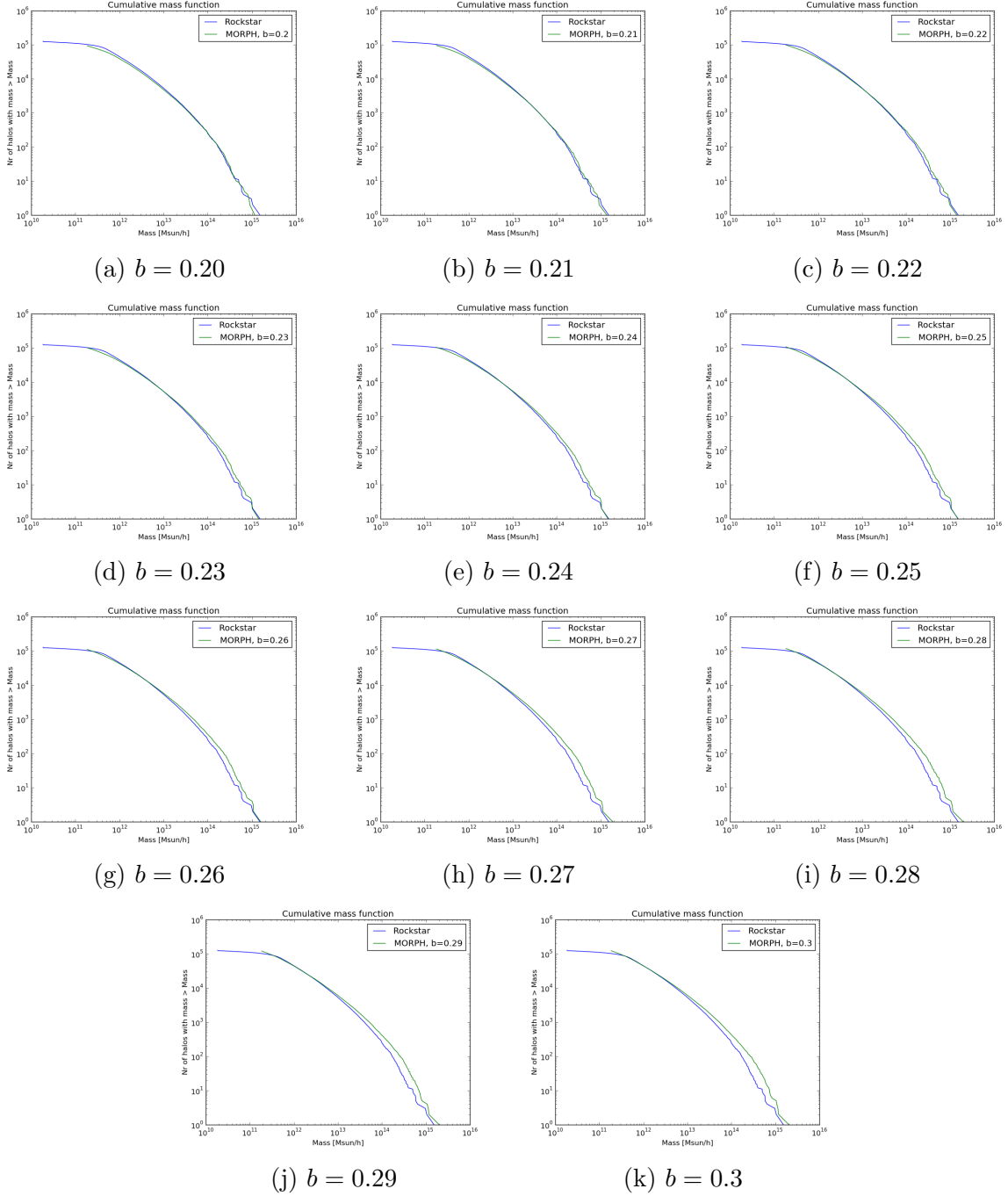


Figure 6.16: Mass functions for $b = [0.2, 0.21, 0.22, 0.23, 0.24, 0.25, 0.26, 0.27, 0.28, 0.29, 0.3]$.

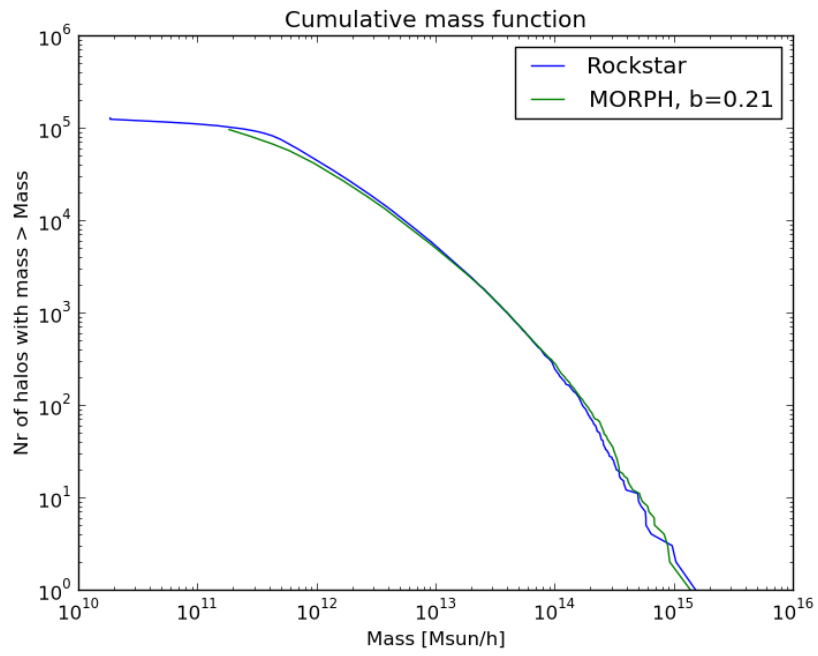


Figure 6.17: Mass function for $b = 0.21$.

Chapter 7

Results

We have developed MORPH, a halo finder that is independent of the gravity model used in N-body simulations. The simulation datasets that have been analyzed are summarized in table 7.1. The lcdm128 simulation was mainly used as a debugging dataset, since it only contains 128^3 particles. This has provided an invaluable tool to quickly test MORPH after making changes or implementing new methods. The 512^3 datasets are from a series of simulations: one Λ CDM run, four runs with different symmetron models and three with $f(R)$ models, all using the same initial conditions. The details of each symmetron model is listed in table 7.2, where L is the cosmological range of the fifth force in Mpc/h, z_{ssb} the redshift for when the spontaneous symmetry breaking occurs and β the strength of the fifth force relative to gravity. The details of the $f(R)$ models are listed in table 7.3, with the parameters being the value of df/dR today and n . The particles have a mass of $9.26138 \cdot 10^9 M_\odot/h$, a matter content of 26.7%, with $\Omega_{\text{CDM},0} = 0.222$, $\Omega_{b,0} = 0.045$ and $\Omega_{\Lambda,0} = 0.733$. The size of the simulations are 256Mpc/h with periodic boundary conditions. The snapshots are taken at $z = 0$.

7.1 Unbinding routines

One of two new methods that takes into account modified gravity is the unbinding routine. Three different unbinding routines have been implemented, accelerated unbinding (section

Dataset	Nr Particles	Mass [M_\odot/h]	Box size [Mpc/h]	Acceleration	Φ
lcdm128	128^3	$9.38 \cdot 10^9$	64	No	No
lcdm1	512^3	$9.26138 \cdot 10^9$	256	Yes	Yes
symm_A	512^3	$9.26138 \cdot 10^9$	256	Yes	Yes
symm_B	512^3	$9.26138 \cdot 10^9$	256	Yes	Yes
symm_C	512^3	$9.26138 \cdot 10^9$	256	Yes	Yes
symm_D	512^3	$9.26138 \cdot 10^9$	256	Yes	Yes
fofr4	512^3	$9.26138 \cdot 10^9$	256	Yes	Yes
fofr5	512^3	$9.26138 \cdot 10^9$	256	Yes	Yes
fofr6	512^3	$9.26138 \cdot 10^9$	256	Yes	Yes

Table 7.1: Analyzed dataset details.

Dataset	z_{ssb}	β	L
symm_A	1	1	1
symm_B	2	1	1
symm_C	1	2	1
symm_D	3	1	1

Table 7.2: Symmetron model details.

Name	$ f_{R0} - 1 $	n
fofr4	10^{-4}	1
fofr5	10^{-5}	1
fofr6	10^{-6}	1

Table 7.3: $f(R)$ model details.

6.5.2), spherical unbinding (section 6.5.1) and a simplified unbinding using the potential from the N-body simulation (section 6.5.3), which promptly will be ignored. We wished to compare the first two models with the case of no unbinding and the results from Rockstar. In addition, we examined the accelerated unbinding method in the Newtonian case. In order to do this we ignored the acceleration for the fifth force, $\ddot{\mathbf{x}}_{\text{fifth force}} = 0$, and only $\ddot{\mathbf{x}}_{\text{Newtonian}}$ is used in equation 6.37. A series of runs using the chosen unbinding routines along with no unbinding was performed on each of the 512 datasets. The cumulative mass function was plotted for each method alongside the Rockstar results.

The plots for the corresponding datasets are in the following figure: lcdm1 - figure 7.1, symm_A - figure 7.2, symm_B - figure 7.3, symm_C - figure 7.4, symm_D - figure 7.5, fofr4 - figure 7.6, fofr5 - figure 7.7 and fofr6 - figure 7.8.

The fractional difference was calculated between the accelerated method for MG and for the Newtonian case. The plot for lcdm1 and symm_A - symm_B are in figure 7.9 and the plot for fofr4 - fofr6 are in figure 7.9.

We also calculated the unbinding percentage for each unbinding routine for all datasets, table 7.4.

7.2 R_{vir}

The other routine that have been made independent of the gravity model is the calculation of R_{vir} through the use of the virialization constant β_{vir} (section 6.6). To test the effects of this routine we performed the previously mentioned runs, but used equation 6.40 calculate R_{vir} , and not the primary method implemented in MORPH. The fractional difference between these two methods was calculated for each dataset and are depicted in figure 7.11.

We also calculated the unbinding percentage for each unbinding routine for all datasets, which is in table 7.5.

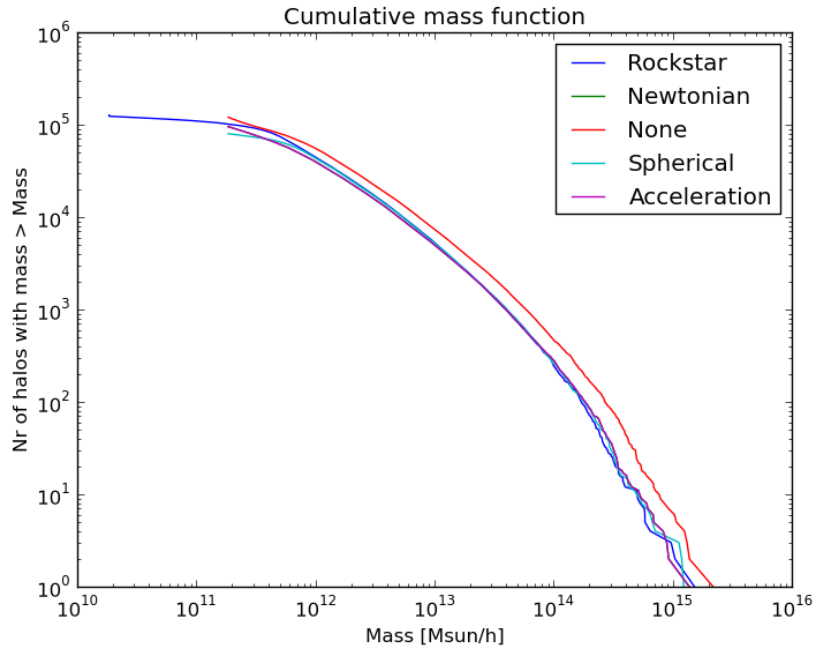


Figure 7.1: Mass function for lcdm1, four different unbinding routines.

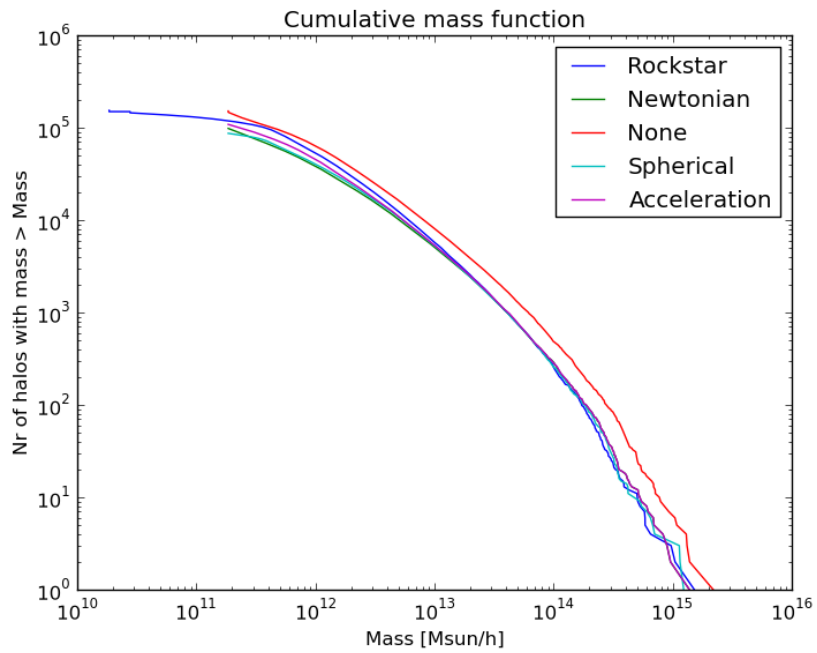


Figure 7.2: Mass function for symm_A, four different unbinding routines.

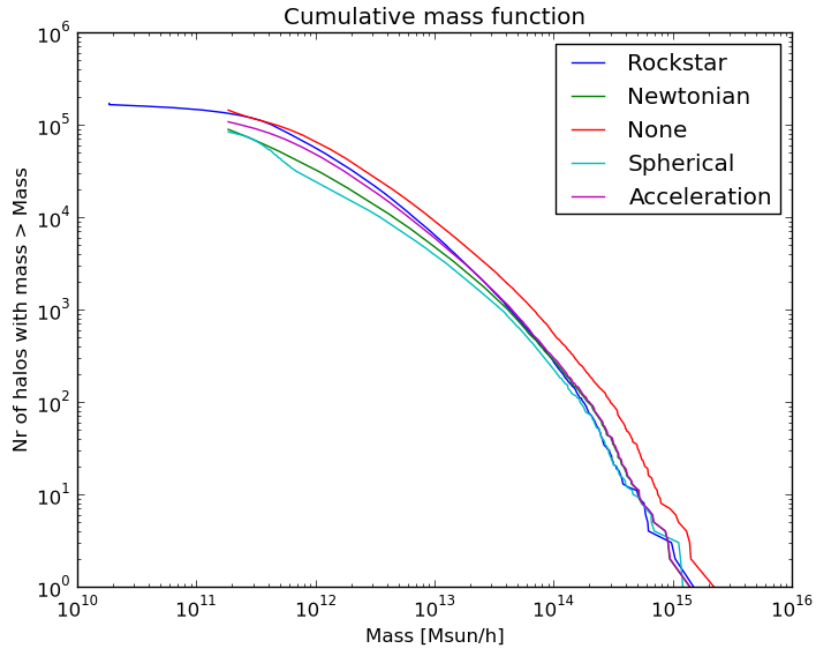


Figure 7.3: Mass function for symm_B, four different unbinding routines.

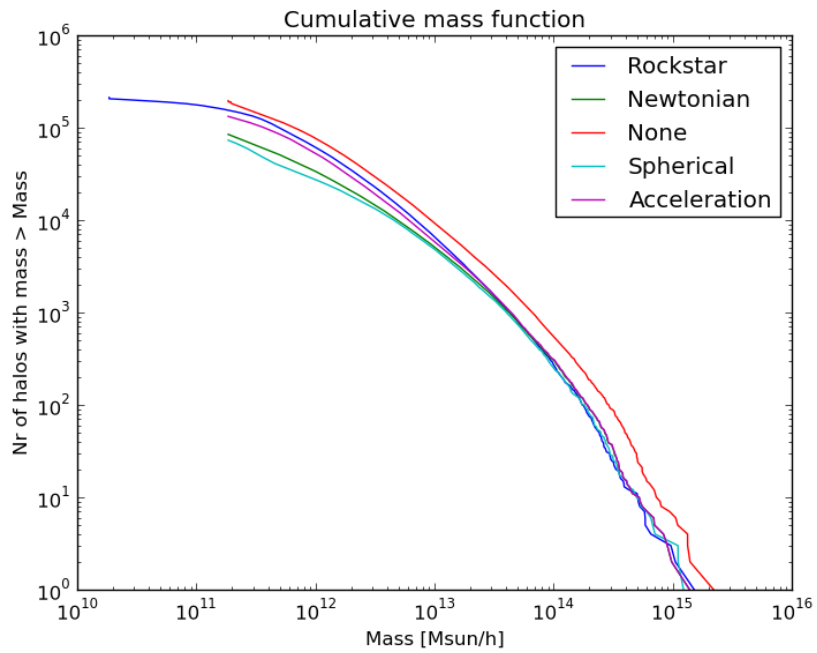


Figure 7.4: Mass function for symm_C, four different unbinding routines.

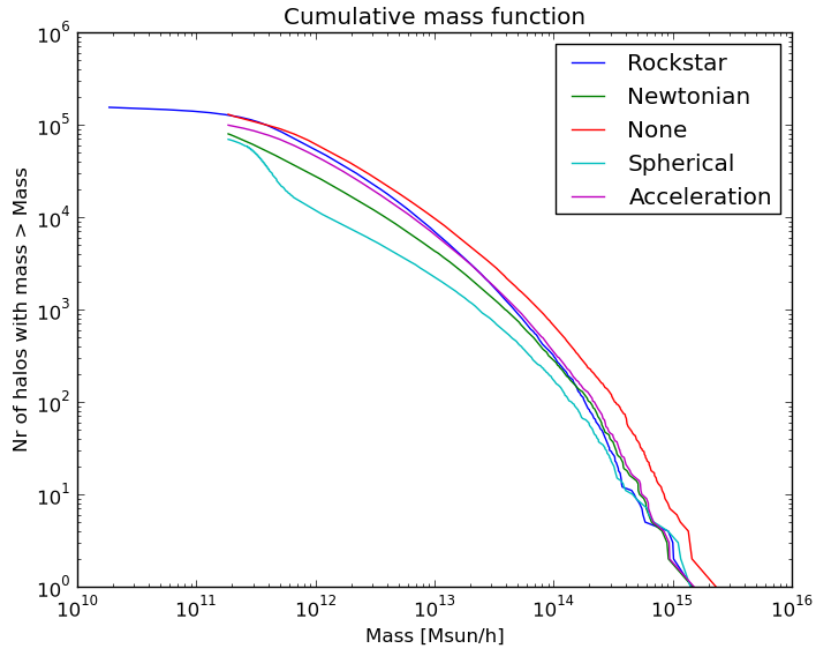


Figure 7.5: Mass function for symm.D, four different unbinding routines.

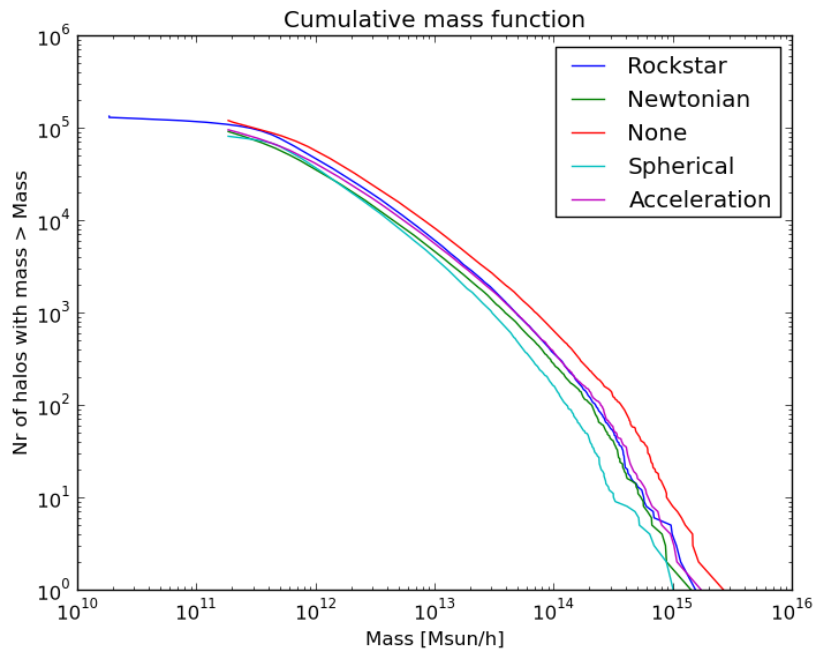


Figure 7.6: Mass function for fofr4, four different unbinding routines.

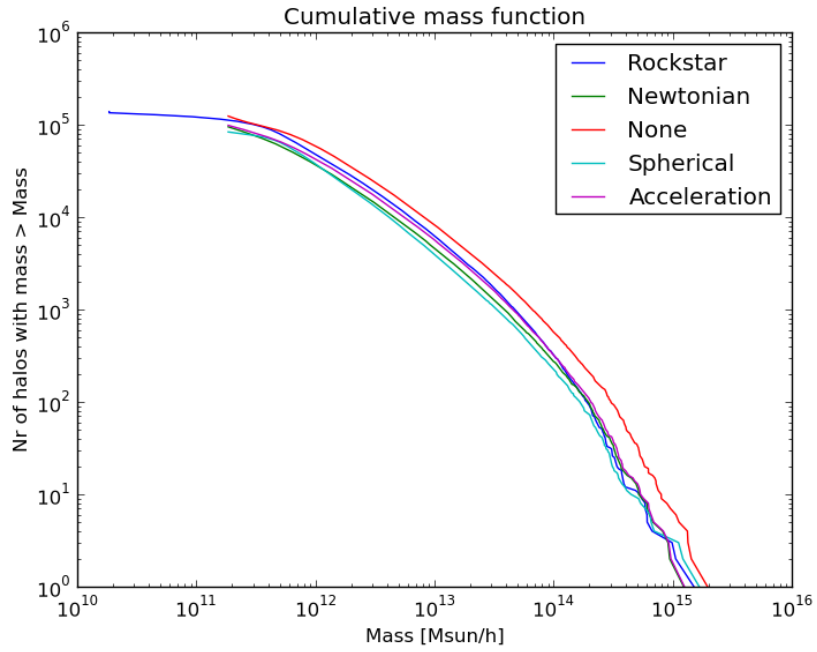


Figure 7.7: Mass function for fofr5, four different unbinding routines.

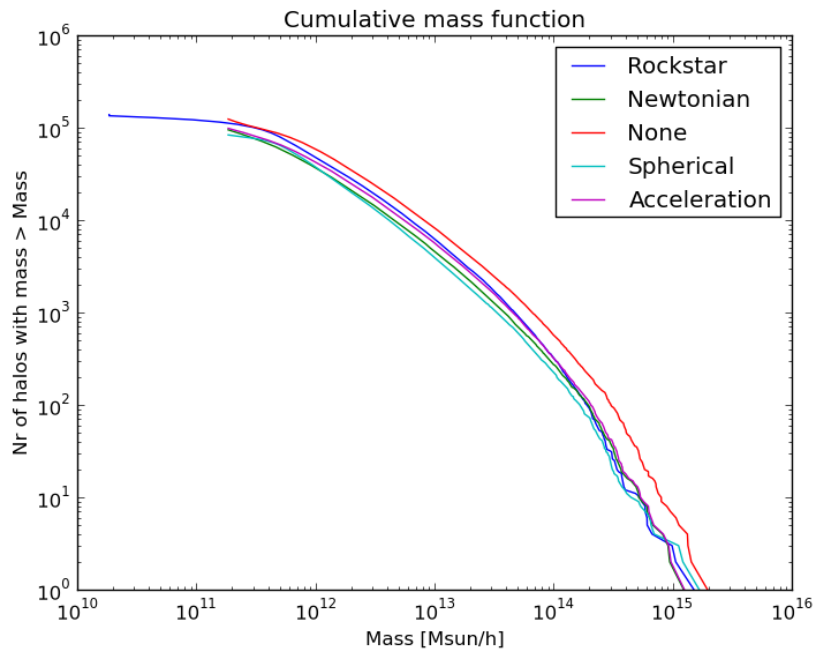


Figure 7.8: Mass function for fofr5, four different unbinding routines.

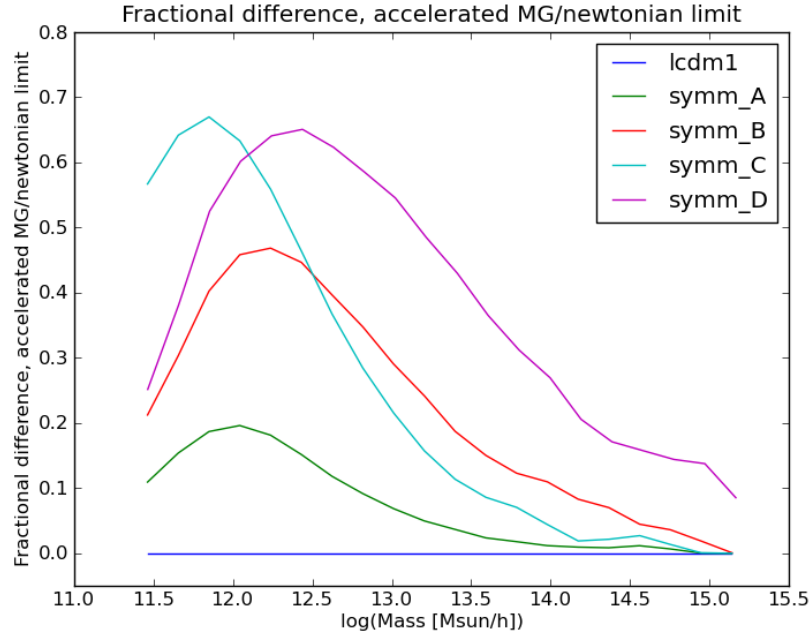


Figure 7.9: Fractional difference between the accelerated unbinding for MG and the Newtonian case, smoothed over neighboring bins to remove noise that appear in the binning, for lcdm1 and symm_A - symm_B.

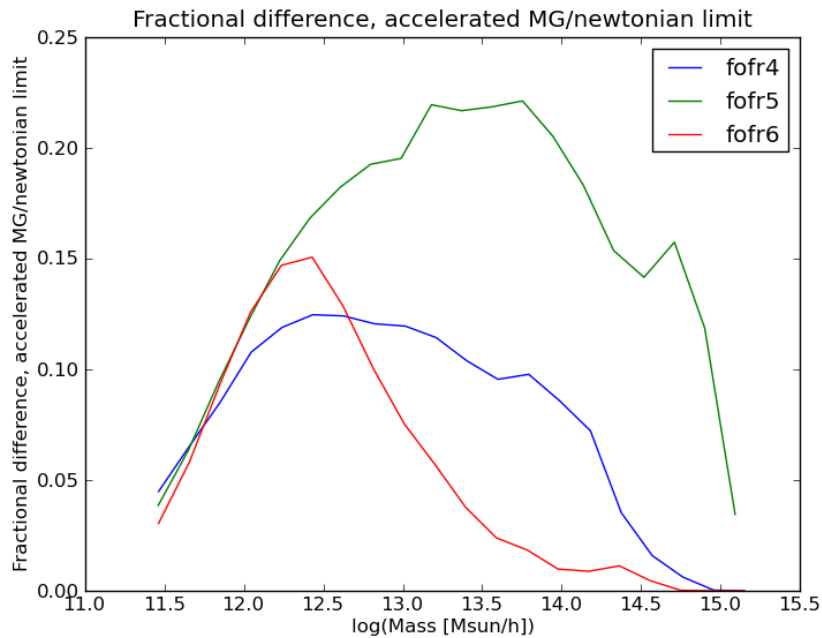


Figure 7.10: Fractional difference between the accelerated unbinding for MG and the Newtonian case, smoothed over neighboring bins to remove noise that appear in the binning, for fofr4 - fofr6.

Dataset	Unbinding method	% of particles unbound
lcdm1	None	0
lcdm1	Spherical	31.48
lcdm1	Accelerated	33.6
lcdm1	Newtonian	33.55
symm_A	None	0
symm_A	Spherical	38.27
symm_A	Accelerated	34.30
symm_A	Newtonian	39.00
symm_B	None	0
symm_B	Spherical	56.79
symm_B	Accelerated	34.91
symm_B	Newtonian	47.72
symm_C	None	0
symm_C	Spherical	54.10
symm_C	Accelerated	36.33
symm_C	Newtonian	49.25
symm_D	None	0
symm_D	Spherical	73.78
symm_D	Accelerated	34.90
symm_D	Newtonian	54.50
fofr4	None	0
fofr4	Spherical	54.53
fofr4	Accelerated	33.78
fofr4	Newtonian	45.11
fofr5	None	0
fofr5	Spherical	48.90
fofr5	Accelerated	33.50
fofr4	Newtonian	43.16
fofr6	None	0
fofr6	Spherical	37.47
fofr6	Accelerated	33.87
fofr6	Newtonian	37.94

Table 7.4: The unbinding values for the different unbinding routines for each dataset

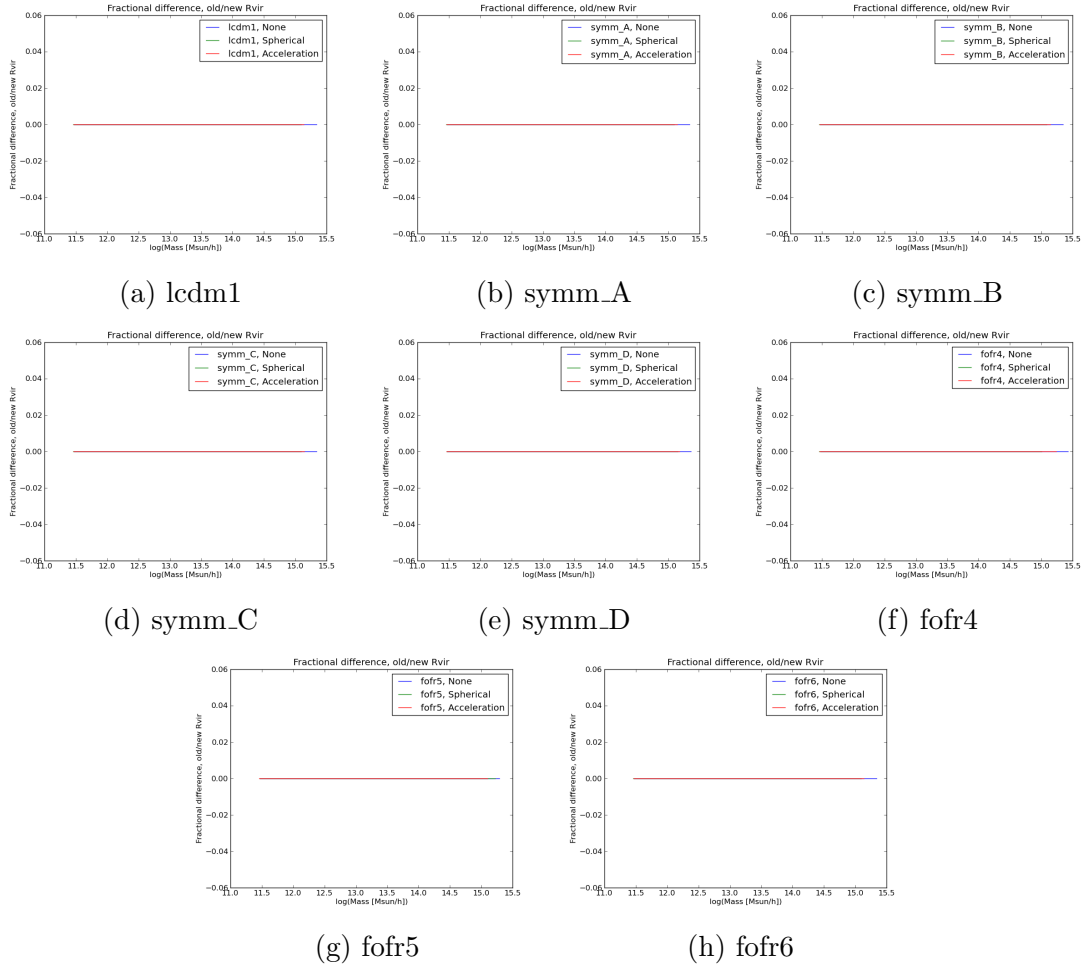


Figure 7.11: Fractional difference between the two methods to calculate R_{vir} .

Dataset	Unbinding method	% of particles unbound
lcdm1	None	0
lcdm1	Spherical	31.48
lcdm1	Accelerated	33.6
symm_A	None	0
symm_A	Spherical	38.27
symm_A	Accelerated	34.30
symm_B	None	0
symm_B	Spherical	56.79
symm_B	Accelerated	34.91
symm_C	None	0
symm_C	Spherical	54.10
symm_C	Accelerated	36.33
symm_C	Loaded	4.95
symm_D	None	0
symm_D	Spherical	73.78
symm_D	Accelerated	34.90
fofr4	None	0
fofr4	Spherical	54.53
fofr4	Accelerated	33.78
fofr5	None	0
fofr5	Spherical	48.90
fofr5	Accelerated	33.50
fofr6	None	0
fofr6	Spherical	37.47
fofr6	Accelerated	33.87

Table 7.5: The unbinding values for the different unbinding routines for each dataset, using the standard method to calculate R_{vir} .

Chapter 8

Conclusions and outlook

In this chapter we draw conclusions from the provided results and give a brief outlook on what future work will focus on.

8.1 Conclusions

We now have a working halo finder that is independent of the modified gravity model. From figure 7.1 it can be seen that Rockstar, AHF, MORPH with spherical unbinding and MORPH with accelerated unbinding all are in good agreement for the standard gravity case.

Calculation of R_{vir}

One of the two new methods implemented to gain independence from the gravity model is the calculation of R_{vir} . In section 6.6 we showed that this algorithm is an improvement upon previous algorithms with regard to modified gravity runs. As can be seen from figure 7.11, together with comparing table 7.4 with table 7.5, there is no difference between using β_{vir} to calculate R_{vir} compared to equation 6.40 in the final results. This result is not unexpected, as seen in section 6.4.1, MORPH uses only R_{vir} in the 6D FOF-method to calculate the phase space distance between particles and the seed halos. The seed halos are small, close to 20 particles, and are located in the densest part of the halos. This means that all the particles are well within the virialization radius for both methods. It also implies that if the two methods obtain different virial radii, the difference between these numbers will be small, since the particles are clustered tightly together. In addition, the virial radius is only used to calculate assignment of particles to seed halo where a difference in R_{vir} does not change much when assigning particles. All of the above points to the fact that the choice of R_{vir} has little or no effect on the final results.

Unbinding

The unbinding procedure is the second method implemented to gain independence of the gravity model. From figure 7.1 it can be seen that Rockstar, AHF, MORPH with spherical unbinding and MORPH with accelerated unbinding are in good agreement for the Λ CDM model. However, there are discrepancies when no unbinding is performed. An unbinding

routine is needed to obtain reasonable results (Knebe et al. 2013), as we clearly see, and we will disregard the no unbinding case. The Newtonian case for the accelerated unbinding is identical to the full accelerated unbinding method, which is exactly as expected in the Λ CDM model. Any other result would suggest there are bugs in the code. Compared with the results from previously verified halo finders, Rockstar (Behroozi et al. 2013) and AHF (Knollmann & Knebe 2009), it can be concluded that MORPH provides correct results for both the spherical unbinding and the accelerated unbinding in the Λ CDM case.

We now look at the four symmetron models analyzed, as shown in figure 7.2-7.5. It can be observed that there are increasing discrepancies between Rockstar and the accelerated method compared to the spherical unbinding and the accelerated method in the Newtonian case. We also note that there are differences between the spherical unbinding and the accelerated method in the Newtonian limit. The spherical unbinding also has a weird inverted s-shape when observing lower mass halos, and there should not be any physical effects that could lead to this behavior. This suggests that there are some bugs in the spherical unbinding method that show up in the more extreme modified gravity simulations. Until this issue is addressed, the spherical unbinding is not trusted and will from here on be ignored.

The more telling result is the difference between the accelerated method for modified gravity and in the Newtonian case. The only difference between them is that one takes modified gravity into account while the other does not. This shows the main effect modified gravity has on the unbinding procedure, where figure 7.9 depicts the fractional difference between these two methods. Note that there is a large difference between them, much more than the 10% error bar that current halo finders have. We observe the same behavior in figure 7.6-7.8 and figure 7.10. We therefore conclude that modified gravity must be taken into consideration when performing the unbinding routine for modified gravity N-body data.

Rockstar makes no specific corrections for modified gravity models in its unbinding routine, so why does Rockstar provide such accurate results in the modified gravity regime? The main reason for this is most likely the implementation of the 3D and 6D FOF methods employed by Rockstar. As previously shown in this thesis, how R_{vir} is calculated has no visible effect on the final results. In Rockstar, the only method left that is dependent on the modified gravity model is the unbinding routine. Rockstar typically has a boundness value of 98% (Behroozi et al. 2013), so even if the unbinding method is inaccurate in the modified gravity regime, it will not have a great impact on the results. Comparing this number with the unbinding percentages of MORPH for the acceleration method, table 7.4, we see that MORPH unbinds a much higher percentage of particles, $\sim 34 - 35\%$. As such the unbinding procedure has a greater impact on the final results in MORPH than in Rockstar. To summarize, while Rockstar gives close to correct results for the tested models this is due to the fact that Rockstar's unbinding routine does not do much work, and the errors are thus less visible. To improve their results for modified gravity simulations Rockstar should implement a modified gravity independent unbinding routine.

8.1.1 Do we need a halo finder that takes into account modified gravity models?

The main question this thesis set out to answer was whether there is a need for a modified gravity adjusted halo finder. What can be concluded from the above is that the choice of unbinding procedure affects the halo finding on modified gravity datasets. When working in the modified gravity regime we need to take the unbinding procedure into consideration, preferentially making the halo finder completely independent of the gravity model in order to perform precision halo finding.

On the other hand, the problem is that the errors in the subhalo mass is of the order 10% and for V_{max} of the order 20-30% (Knebe et al. 2013), all resulting from the intrinsic particle gathering algorithms. For the worst case in the Newtonian limit, ~ 1.65 times too many particles are unbound. Even if the same happens with Rockstar this would give an error of $\sim 1.3\%$, which is much less than the current error bars for halo finders. As such we do not yet need to take modified gravity into account for Rockstar, as there are other steps in the halo finding algorithms that introduces greater errors. But once they have been eliminated, modified gravity must be incorporated into the halo finders. The conclusion is that modified gravity must be taken into consideration when we wish to analyze halos in modified gravity datasets. However, only if the halo finders have a high unbinding percentage, making the errors from the unbinding routine larger than the current error bars for halo finding.

8.2 Outlook

This thesis have just started to scratch the surface on what can be done with MORPH and there is a much work left to do that this thesis have not had the time to focus on. This future work is divided into two distinct paths, the first is improve the results obtained from MORPH. The other is to focus on improving the code, adding features, fixing bugs and optimizing the code for speed and memory consumption.

8.2.1 Result improvements

There are a few obvious results that we would like to generate in future works, listed below in no particular order.

Examine why spherical unbinding does not work

One thing to examine is why the spherical unbinding fails in the modified gravity regime. There are three possible sources for the error, the spherical approximation, assuming standard gravity when solving the Poisson equation and numerical errors in the integration. The contribution from each source would also be interesting to investigate.

Calculate and test relevant halo statistics for dependency on the modified gravity model

There are several halo properties of interest currently not calculated, such as the spin of a halo and total energy. The routines that calculate these properties should be implemented

in MORPH. As previously mentioned some halos properties, such as the halo mass, are unresponsive to a few particles that should not be there, while others, for example halo spin, are highly sensitive to these few extra particles (Knebe et al. 2011). Because of this it would be of interest to examine the dependency on the modified gravity model for these halo properties.

Run on several modified gravity models

One relation that would be interesting to examine is the relation between the deviations and the models. This can be done by performing analysis on a series of modified gravity N-body simulations with identical initial conditions and changing variables for modified gravity model and then record the deviations from the accelerated unbinding procedure in MORPH. A variation of this is to run on more extreme gravity models, to see if we get visible deviations from Rockstar, further confirming our results from the previous section.

Run on several datasets

One test that should be done to confirm our results is to get a hold of a large number of N-body simulations, preferably > 100 , using the same modified gravity model with different initial conditions and then run MORPH on each. This would give a large number of runs that a average and standard deviation could be calculated for. Then we could see if the observed deviations are statistical outliers or if the average of the spherical unbounding is outside the standard deviation from the accelerated unbinding. The problem with this is that we need a large amount of simulations, which cost an enormous amount of CPU hours to compute, unfortunately this is at the current time not feasible due to the resources available.

Test R_{vir}

The effects of using the new method to calculate β_{vir} should be examined closer. There is no difference in the final results, but R_{vir} for all the found halos should be looked at.

Test on Halos gone MAD datasets and compare more thorough with other halo finders

One obvious test is to compare MORPH with the halo finders tested in Knebe et al. (2011), on the same datasets. These tests have not been prioritized as they are not representative of the halos expected to be found in real simulations, they only considered spherical NFW or Plummer halos with little to no substructure. Instead of this, we have focused on running MORPH on actual simulations. This have been more important than usual, as we have been working on modified gravity halos that might deviate more from these test halos than what standard gravity halos would. But doing this test would give additional confirmation that MORPH are correct in the standard gravity regime.

8.2.2 Code improvements

There are several aspects of the code that can be improved upon in future works, some of the more important are listed below in no particular order.

Add unbound particles to other possible halos

At the moment particles found to be unbound in the unbinding routine are permanently removed. Instead what should be done is to test if the unbounded particles might be bound to one of the other halos found in that FOF-group. If they are gravitationally bound to several halos, the one closest in phase space should be chosen.

Remove memory leaks

Currently there are several memory leaks in MORPH. These are not serious as the worst have been fixed, but there should optimally be none. This gets more important the larger the dataset analyzed are.

Improve usability

There is much work left to be done to improve the usability of MORPH. More methods to read output from standard N-body codes should be added and a method to choose which from the .ini file should be implemented. Optimally the user should not need to change anything in the source code. Another thing needed is to clean up the code and add more comments. There are also code segments and functions that are outdated and no longer needed which should be removed.

Optimizing MORPH

There are several steps that can be done to optimize the code of MORPH.

Speed

A lot of work has already been put into increasing the speed of MORPH, but there is still more that can be done. One of the main limiting factors is that the 6D phase space splitting algorithm has a lower limit on the wall time set by the analysis time for the largest halo. To increase the effect of multiple processors we need to split the 3D FOF groups into smaller FOF subgroups and implement a stitching algorithm that correctly combines the results back into the halos. Much can also be done by optimizing the speed of the current implemented algorithms.

Use the GPU instead of the CPU

Utilizing the GPU instead of the CPU is one way to speed up the calculations. A GPU consist of thousands of cores optimized for high parallel efficiency, and we can utilize this to do our 6D phase space splitting. To be fully able to utilize the advantages of thousands of cores we need to implement a better 6D phase space algorithm, as discussed above. The GPU support would be implemented using CUDA, a parallel computing platform and programming model created by NVIDIA.

Memory

The larger the datasets the more important memory optimization is. The 512 datasets analyses in this thesis contain 15 doubles per particles, which means that at a bare minimum we need $512^3 * 15 * 8$ bytes = 16.11 GB of memory to hold the data and this is without any of the data computed by MORPH. We want to run MORPH on datasets larger than 512^3 particles, and memory consumption then becomes a concern.

HaloFinding@Home

This is mostly a curiosity, but would be an interesting concept to do with a halo finder when working on extremely large datasets. The @Home suffix is used for distributed computing services like SETI@Home and Folding@Home. The idea behind these services is to utilize the thousand of unused home computers around the world instead of using large expensive computer clusters to do the calculations. In our case this would be used to do the 6D phase space splitting, replacing the MPI part of the code with a code that sends and gathers tasks to and from home computers that have the appropriate software installed.

Appendix A

MORPH, Libraries

A short summary of the source code is in table A.1.

A.1 Libraries

MORPH use only a few standard libraries, making it easier to compile on new machines. The libraries are:

- `<boost/algorithm/string.hpp>` for easier handling of strings.
- `<vector>` to get better handling of arrays.
- `<fstream>` for file handling.
- `<iostream>` to get standard input/output.
- `<sstream>` to be able to do string manipulations, this will be removed from the code as this only is used for testing purposes.
- "MPI.h" to get MPI support.

Language	Files	Code	Comment	Comment %	Blank	Total
cpp	26	3750	1247	25.0%	1553	6550
python	23	550	154	21.9%	217	921
c	3	154	119	36.4%	179	406
make	1	45	45	57.7%	28	106
Total	53	4541	1565	25.6%	1877	7983

Table A.1: Code summary, from ohcount.

Appendix B

Code parameters

MORPH comes with a MORPH.ini file. This file contains a number of parameters that can/must be changed, depending on the physical problem examined. Most of the parameters should not be changed without a good reason, as the most appropriate values already have been chosen.

BoxSize = **256.0** Size of the simulation box, in Mpc/h.

Mass = $9.26 \cdot 10^9$ Mass of the simulation particles, in M_{\odot}/h .

source = `./source/` Path of the source directory.

inData = `./inData/` Path of input data folder.

outData = `./outData/` Path of output data folder.

convMass = $1 \cdot 10^{10}$ Conversion factor to convert mass from input file to M_{\odot} .

convDistance = 1 Conversion factor to convert distances from input file to Mpc/h.

convVelocity = $1.02 \cdot 10^{-3}$ Conversion factor to convert velocities from input file to Mpc/Gyr.

convAcceleration = $3.223 \cdot 10^{10}$ Conversion factor to convert acceleration from input file to Mpc/Gyr².

convPhi = $1.045 \cdot 10^{-12}$ Conversion factor to convert Φ from input file to Mpc²/Gyr².

G = $4.49 \cdot 10^{-15}$ The gravitational constant in units of Mpc³ $M_{\odot}^{-1}h$ Gyr⁻².

ScaleDensity = **360** Virial overdensity.

b = **0.21** Scaling factor for the linking length we calculate.

HaloLimit = **20** All halos with less particles than this is discarded, this give the lower limit for the size of a halo.

HaloSeed = **20** Numbers of particles needed to create a halo seed.

f = 0.5 f is the fraction of particles kept from parent group to subgroup, it is used to calculate the adaptive linking length in the 6D phase-space splitting.

NrLinking = 10000 Number of random particles used to calculate the above linking length. A higher number is more accurate, but makes the calculations slower.

MaxHalos = 1000 This is the maximum number of halos expected to be found in the 6D phase-space splitting. This value is used to allocate the array that the results from the slave processor is stored in. If the number of halos found in the slave is larger we get a segmentation fault, as the mpi routine then tries to write to unallocated memory. A larger value could always be chosen, but that increases memory consumption.

NrParticlesDouble = 10 Number of particles that the 3D FOF routine needs to find belonging to the same halo before the linking length is increased.

LinkingLenghtScale = 3 When the above number of particles is reached this gives the scale that the linking length is multiplied by.

minUnbind = 0 This is the ending criteria for the unbinding process. When we reach this number of particles unbound in one iteration, the unbinding process for that halo is ended.

NrShells = 200 The number of shells used to calculate β_{vir} .

minParticlesShell = 3 The minimum number of particles per shell when calculating β_{vir} . If there are less than $\text{minParticlesShell} \times \text{NrShells}$ particles the number of shells used is reduced to follow this criteria.

BetaVirThreshold = 0.2 Threshold for when a halo is considered virialized, when $\beta_{vir} < \text{BetaVirThreshold}$.

Warning, the following parameters can, but should not in almost any circumstance, be changed. These are used by the lower level routines and changing any of these will result in MORPH crashing spectacularly. If these are changed the code needs to be rewritten to work, this should not be attempted without intimate knowledge of how MORPH works.

ParticleSize = 11 Number of numbers in a CParticle, used among other things to create particle arrays.

HaloSize = 14 Number of numbers specific to a CHalo, used among other things to create halo arrays.

ArrayExtraSize = 10 Numbers of extra spaces allocated in a CArray. The higher the value the more memory is allocated extra for each array, and fewer allocations need to be made when one want to increase the number of numbers stored in a CArray. This can be increased, but that would lead to an increased memory consumption. The current value is chosen such that it is low enough to reduce memory usage, but still high enough so that in most cases we do not need to allocate more memory.

Bibliography

- Aubert, D., Pichon, C., & Colombi, S. 2004, *mnras*, 352, 376
- Bagla, J. S. & Padmanabhan, T. 1997, *Pramana*, 49, 161
- Bamba, K., Gannouji, R., Kamijo, M., Nojiri, S., & Sami, M. 2012, ArXiv e-prints
- Barnes, J. & Hut, P. 1986
- Behroozi, P. S., Wechsler, R. H., & Wu, H.-Y. 2013, *Astrophys.J.*, 762, 109
- Bergstrom, L. 2006, *Cosmology and particle astrophysics* (Berlin New York: Springer)
- Bryan, G. L. & Norman, M. L. 1998, *apj*, 495, 80
- Buchdahl, H. A. 1970, *mnras*, 150, 1
- Capozziello, S., De Laurentis, M., & Faraoni, V. 2009
- Chandrasekhar, S. 1961
- Clifton, T., Ferreira, P. G., Padilla, A., & Skordis, C. 2012, *Phys.Rept.*, 513, 1
- Davis, M., Efstathiou, G., Frenk, C. S., & White, S. D. M. 1985, *apj*, 292, 371
- De Felice, A. & Tsujikawa, S. 2010, *Living Rev.Rel.*, 13, 3
- Dodelson, S. 2004, *Modern Cosmology* (Academic Press)
- Dutra, S. M. 2005, *Cavity quantum electrodynamics the strange theory of light in a box* (New York: J. Wiley)
- Einstein, A. 1908, *Jahrbuch der Radioaktivität und Elektronik*, 4, 411
- Einstein, A. 1916, *Annalen der Physik*, 354, 769
- Einstein, A. 1936, *Science*, 84, 506
- Elahi, P. J., Thacker, R. J., & Widrow, L. M. 2011, *mnras*, 418, 320
- Gamow, G. 1970, *My world line : an informal autobiography* (New York: Viking Press)
- Guth, A. H. 1981, *Phys.Rev.*, D23, 347
- Hammami, A. 2013, Master's thesis, University of Oslo

Harfst, S., Gualandris, A., Merritt, D., et al. 2007, *na*, 12, 357

Hinterbichler, K., Khoury, J., Levy, A., & Matas, A. 2011, 84, 103521

Hubble, E. 1929, *Proceedings of the National Academy of Science*, 15, 168

Hulse, R. A. & Taylor, J. H. 1974, *apjl*, 191, L59

Hwang, J.-c. & Noh, H. 2001, *Physics Letters B*, 506, 13

Iliev, I. T., Moore, B., Gottloeber, S., et al. 2010

Knebe, A., Knollmann, S. R., Muldrew, S. I., et al. 2011

Knebe, A., Pearce, F. R., Lux, H., et al. 2013

Knollmann, S. R. & Knebe, A. 2009, *Astrophys.J.Suppl.*, 182, 608

Llinares, C., Mota, D. F., & Winther, H. A. 2013

Maccio, A. V., Murante, G., & Bonometto, S. A. 2003, *Astrophys.J.*, 588, 35

Max Grönke. 2013, Master's thesis, University of Oslo

Moore, G. E. 1965, *Electronics*, 38, 114

Øvind Grøn. 2007, *Lecture Notes on the General Theory of Relativity* (Springer)

Øystein elgarøy. 2010, *AST4220: Cosmology I*

Penzias, A. A. & Wilson, R. W. 1965, *apj*, 142, 419

Prada, F., Klypin, A. A., Simonneau, E., et al. 2006, *Astrophys.J.*, 645, 1001

Press, W. H. & Schechter, P. 1974, *apj*, 187, 425

Ryden, B. 2002, *Introduction to Cosmology* (Benjamin Cummings)

Shapiro, I. I. 1964, *Physical Review Letters*, 13, 789

Shaw, L. D., Weller, J., Ostriker, J. P., & Bode, P. 2006, *apj*, 646, 815

Sotiriou, T. P. & Faraoni, V. 2010, *Rev.Mod.Phys.*, 82, 451

Springel, V., White, S. D. M., Jenkins, A., et al. 2005, *nat*, 435, 629

Stadel, J., Potter, D., Moore, B., et al. 2009, *mnras*, 398, L21

Starobinskii, A. A. 1979, *Soviet Journal of Experimental and Theoretical Physics Letters*, 30, 682

Stoehr, F., White, S. D. M., Springel, V., Tormen, G., & Yoshida, N. 2003, *mnras*, 345, 1313

Teyssier, R. 2002, *Astron.Astrophys.*, 385, 337

Teyssier, R., Pires, S., Prunet, S., et al. 2009, aap, 497, 335

von Hoerner, S. 1960, zap, 50, 184