Master's thesis
Astrophysical Sciences

Supermassive black holes and the cosmological formation of massive early-type galaxies (title not final)

Atte Keitaanranta

July 20, 2021

Supervisor(s): Prof. Peter Johansson
M.Sc. Matias Mannerkoski

Censor(s): Prof. Peter Johansson

UNIVERSITY OF HELSINKI
Department of Physics
PL 64 (Gustaf Hällströmin katu 2)
00014 University of Helsinki

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Koulutusohjelma — Utbildningsprogram — Education programme
Faculty of Science	Department of Physics

Tekijä — Författare — Author

Atte Keitaanranta

Työn nimi — Arbetets titel — Title

Supermassive black holes and the cosmological formation of massive early-type galaxies (title not final)

Opintosuunta — Studieinriktning — Study track

Astrophysical Sciences

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Master's thesis	July 20, 2021	0 pages

Tiivistelmä — Referat — Abstract

Abstract goes here.

Avainsanat — Nyckelord — Keywords

Your keywords here

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — övriga uppgifter — Additional information

Contents

1. Introduction

1.1 Information about galaxies, shortly

1.2 Aim of the thesis

2. Background

2.1 Cosmology

2.1.1 Hubble parameter, Friedmann equations and so on

2.1.2 Cosmological perturbations

2.2 Early-type galaxies

2.2.1 Types of ellipticals

2.2.2 Photometric and kinematic profiles

2.3 Evolution of dark matter halos

2.3.1 The Zel'Dovich approximation

2.4 Feedback processes

3. GADGET-3 and KETJU

- Haven't really thought about the contents of this chapter yet

3.1 Overview of GADGET-3

3.2 Smoothed Particle Hydrodynamics

3.3 Gas cooling?

3.4 Feedback?

3.5 KETJU

4. Creating initial conditions for the cosmological simulations

This chapter describes how the initial conditions (ICs) for cosmological simulations are created. The first section describes the so-called 'zoom-in' method, which allows us to have spatially large simulation boxes with high resolution regions. After this, the code MUSIC (MUlti-Scale initial conditions) is introduced, which is used to create a spatial volume with realistic velocity and density perturbations at an early redshift ($z = 50$). Compared to an earlier implementation by Bertschinger (2001), the errors in the velocity and displacement fields are improved by two orders of magnitude (Hahn & Abel, 2011). The created initial ICs are used as a starting point for the performed GADGET-3 simulations that are run in this thesis. The last part of this chaper focuses on the setup of the cosmological setup of the simulations, and on the preliminary low resolution large volume run needed for the higher resolution simulation.

4.1 Zoom-in technique

To study galaxy formation and evolution in a proper cosmological context, the simulation box must have a large volume. The perturbations at multiple fundamental length scales enter the non-linear regime at different redshifts and thus without a large enough volume (spatial dimensions of the box being $\sim 100\,\mathrm{Mpc}$), the matter power spectrum is not resolved correctly on the largest scales. This results in unrealistic gravitational tidal forces from the large scale structures, which in turn create incorrect peculiar velocities in smaller scale structures, as shown in e.g. Mo et al. (2010).

To resolve the gravitational effects of a smaller scale structure, it is also required to have a high resolution, i.e. a large amount of particles with relatively low mass.

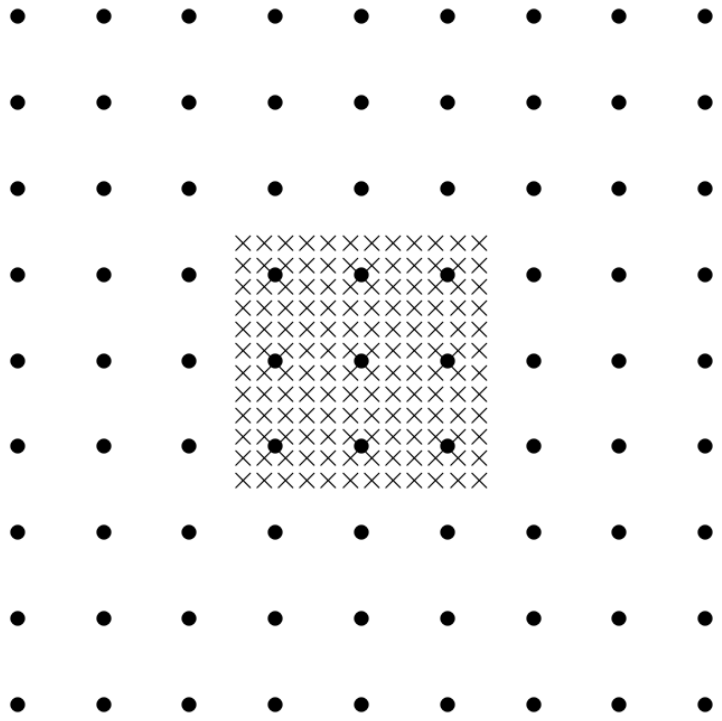

Figure 4.1: An example of a zoom-in region, with one level of mesh refinement. The circles represent the particles on the coarse grid. The crosses represent the refined particles, with each coarse particle inside this region divided into 4^3 particles (Bertschinger, 2001).

Having a sufficient resolution on the whole simulation volume would result in a unreasonably large computational workload. Thus it is more sensible to implement a method, which has a high resolution in a single region of interest, surrounded by a large volume with a smaller resolution. This method, nowadays known as the 'zoom-in' technique, has been in use for multiple decades. One of the first simulations with a region of interest surrounded by low resolution background was performed by Navarro & White (1994), and later implementations have been used by e.g. Power et al. (2003) and Marinacci et al. (2014) to name a few.

An example of a zoom-in grid is shown in Figure 4.1. Here, the dots represent the large particles on a coarse grid, while the crosses represent the refined grid with lower mass particles. In this refinement, each massive particle in the refined region is divided into 4^3 particles. Modern implementations, such as the one used in this thesis, use zoom regions with multiple levels of refinement (see Section 4.2.3). As adding levels of refinement resolves smaller structures, the results of zoom-in

simulations with different maximum resolution change slightly. The simulation with better resolution includes more low-mass satellites, and their locations can also differ. The masses of most massive objects are also affected, and mergers can occur at slighly different times.

To locate the regions of interest, we need to first perform a computationally less expensive simulation without a high resolution volume (discussed in 4.3.2) including only dark matter, and then choose the zoom-in regions and perform the simulation again with baryons included. The initial conditions (ICs) must also portray a realistic case, i.e. the fluctuations at a very high redshift must match the expected density structure. from linear theory. As the redshift is till very high ($z > 20$), the evolution of density structures is still follows the linear theory. Fortunately, we can use a single program to create realistic ICs, with zoom-in box included.

4.2 Music

4.2.1 Overview

Music (MUlti-Scale Initial Conditions) is a code written by Hahn & Abel (2011), which can be used to create the ICs for Gadget-3 simulations. The program creates a simulation volume, with velocity and particle displacements calculated using a power spectrum and two-point correlation given as input. To account for fluctuations on smaller scales, the program is also able to create a high resolution zoom-in region for the ICs. The zoom-in region can have multiple levels of refinement, and an example of an IC created using Music is shown in Figure 4.2.

Music improves on the prior work, e.g. the Grafic-2 code made by Bertschinger (2001), which also produces ICs with multiple levels of refinement. One of most notable upgrades in Music compared to Grafic-2 is the way the transfer function is used to calculate the density perturbations, discussed in 4.2.2. Another improvement is the way different refinement levels constrain other levels, which is discussed in 4.2.3.

The initial conditions produced by the code best describe the velocity and density structure when the redshift of the produced ICs lie on the linear perturbation regime. If the initial redshift z_i of the ICs is set to be too low (i.e. on the non-linear regime), the formation of the first haloes occurs at unrealistically late times and the formation of high-mass haloes is suppressed, as shown by Reed et al. (2013). Still, Music gives relatively accurate ICs even in the mildly non-linear regime of $z_i \sim 20$

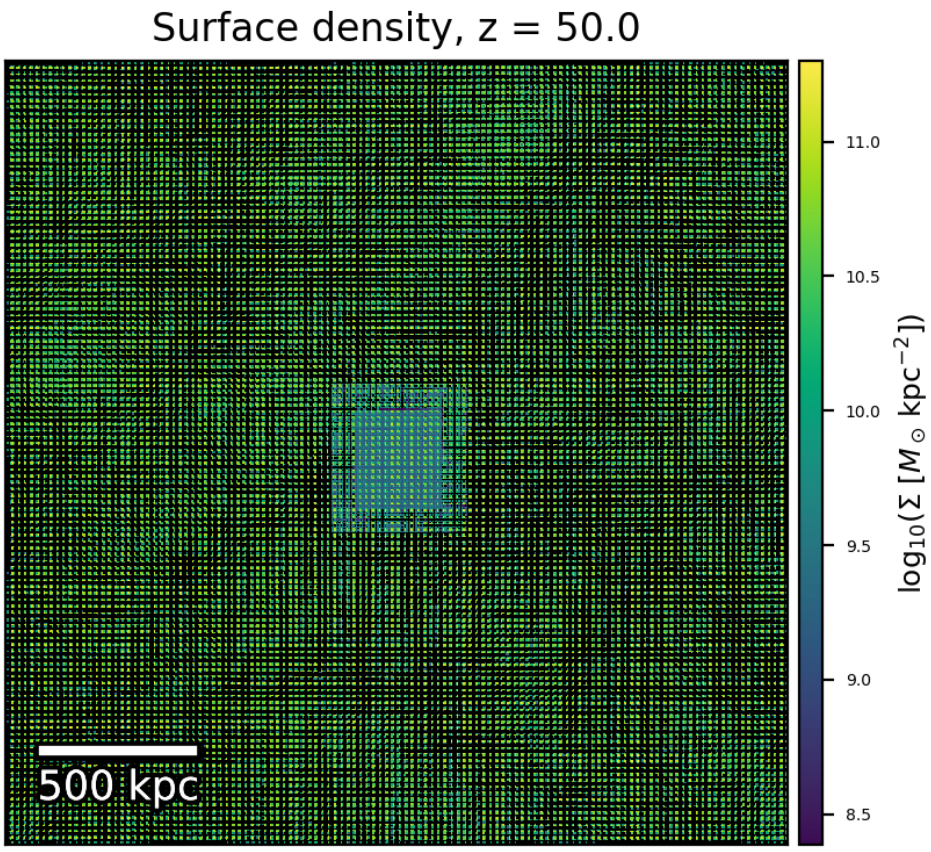

Figure 4.2: An example of a density field created using MUSIC, in physical units. The sidelength of the simulation box is 100 Mpc/h. At the center, a region with four refinement levels is included. Note: changed this from the example in the MUSIC user guide to a plot I made. Not sure if I'm 100% happy with this yet.

(Hahn & Abel, 2011).

4.2.2 Generation of the seed density field

For the generation of particle velocities and positions, MUSIC first needs to create a density field, described by a density perturbation $\delta(\mathbf{r})$. Using an a Power spectrum $P(k)$ given as input, this field can be described completely, as the power spectrum of the field is given by

$$P(k) \equiv \langle \tilde{\delta}(\mathbf{k}) \tilde{\delta}^*(\mathbf{k}) \rangle, \tag{4.1}$$

where * denotes the complex conjugate, and $\tilde{\delta}(\mathbf{k})$ represents the Fourier transform of the density perturbation, i.e.

$$\tilde{\delta}(\mathbf{k}) = \int \delta(\mathbf{x}) e^{-i\mathbf{k}\cdot\mathbf{x}} \mathrm{d}^3\mathbf{x}. \tag{4.2}$$

The transfer function $T(k)$ can also be used to express the power spectrum, defined as

$$P(k) = \alpha k^{n_s} T^2(k), \tag{4.3}$$

where α is a constant used to normalize the power spectrum, and n_s is the spectral index, describing the slope of the spectrum. The value of the spectral index is given to MUSIC in the configuration file, while α is evaluated in using another, observable constant.

One way to normalize the spectrum is to measure the variance of the galaxy distribution σ^2 at a distance R. As shown in e.g. Mo et al. (2010), the predicted variance is defined as

$$\sigma^2(R) = \frac{1}{2\pi^2} \int P(k) \widetilde{W}^2(k) k^2 \mathrm{d}k, \tag{4.4}$$

where $\tilde{W}(k)$ is defined as the Fourier transform of a window function $W(r)$. There are multiple choices for the window function, and the most popular one is the top-hat window function $W(r)$, which is given by

$$W(r) = \begin{cases} \frac{3}{4\pi R^3}, & r \leq R \\ 0, & r > R. \end{cases} \tag{4.5}$$

The Fourier transform of this function is

$$\widetilde{W}(k) = \frac{3}{(kR)^3} \left(\sin(kR) - kR \cos(kR) \right). \tag{4.6}$$

As the variance, as well as the shape of the power spectrum are known, the amplitude of the spectrum can now be normalized. For historical reasons, the distance where the variance is usually measured is $R = 8\,\mathrm{Mpc/h}$, and is also the value MUSIC takes as input to normalize the power spectrum. At this distance, $\sigma(8\,\mathrm{Mpc/h}) \equiv \sigma_8$ is valued close to unity. The value is close to one, since this is approximately the length scale at which the structure evolution becomes non-linear, with structures having higher length scales evolving linearily [SOURCE].

As the shape of the power spectrum are known and the spectrum is normalized, the generation of the density field is now possible. The goal is to create the density field from random noise values $\mu(\mathbf{r})$, while requiring that the amplitudes follow the power spectrum $P(k)$. The sample of random values $\mu(\mathbf{r})$ is called the white noise sample. White noise field is a field which with an constant power spectrum, i.e. $P(k) \propto k^n$ with $n = 0$ (Mo et al., 2010).

MUSIC generates the white noise sample using a Gaussian distribution. Discussion of Gaussian distributions can be found in multiple sources of literature, for

example Mo et al. (2010). For a distribution to be Gaussian, the probability density function ϱ of random value x is

$$\varrho(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\bar{x})^2}{2\sigma^2}\right), \tag{4.7}$$

where $\bar{x}$ and σ^2 are the mean and the variance, respectively. This equation can be generalized to higher dimensions, and for a n-dimensional random field $\delta(\mathbf{x}) = (\delta_1, \delta_2, \ldots, \delta_n)$ the distribution is Gaussian if the probability distribution function can be written as

$$\varrho(\delta_1, \delta_2, \ldots, \delta_n) = \frac{1}{\sqrt{(2\pi)^n \det(\mathcal{M})}} e^{-\mathcal{L}}, \tag{4.8}$$

where $\mathcal{M}$ is the covariance matrix, $\mathcal{M}_{ij} = \langle \delta_i \delta_j \rangle$, and

$$\mathcal{L} \equiv \frac{1}{2} \sum_{i,j} \delta_i \left(\mathcal{M}^{-1}\right)_{ij} \delta_j. \tag{4.9}$$

It is assumed that the Gaussian random field is homogenous and isotropic, meaning that the multivariant Gaussian distribution functions are invariant under spatial translation or rotation. Therefore the two-point correlation function $\xi(x)$ completely determines the distribution functions. Specifically, the field's one-point distribution function is

$$\varrho(\delta)\mathrm{d}\delta = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\delta^2}{2\sigma^2}} \mathrm{d}\delta, \tag{4.10}$$

where the variance of the field is $\sigma^2 = \xi(0)$. Note: Peter, you mentioned that I should explain a bit what $\xi(0)$ means, but I'm struggling with this. Is there a book/paper I could check or could you explain a bit what this means?

Returning to the white noise sample, the amplitudes of the random values $\mu(\mathbf{r})$ can be generated to follow the power spectrum $P(k)$ by considering the Fourier transformed random field $\tilde{\mu}(\mathbf{k})$. With this, the Fourier transformed field $\tilde{\delta}(\mathbf{k})$ can be written as

$$\tilde{\delta}(\mathbf{k}) = \sqrt{P(||\mathbf{k}||)}\tilde{\mu}(\mathbf{k}) = \alpha ||\mathbf{k}||^{n_s/2} T(||\mathbf{k}||)\tilde{\mu}(\mathbf{k}). \tag{4.11}$$

A common procedure to evaluate the real-space density field is to perform an inverse Fourier transform to $\tilde{\delta}(\mathbf{k})$, as is done in e.g. Bertschinger (2001). MUSIC uses a different approach, and calculates the real-space density field as

$$\delta(\mathbf{r}) = \mathcal{T}_R(||\mathbf{r}||) * \mu(\mathbf{r}), \tag{4.12}$$

where $\mathcal{T}_R(r)$ is the real-space counterpart of $\widetilde{\mathcal{T}}(k) \equiv \alpha k^{n_s/2} T(k)$ and "$*$" is denoting a convolution, discussed thoroughly by Salmon (1996). Briefly, the white noise $\mu(\mathbf{x})$ is realised as density perturbations in real space using the convolution operator. This method has previously been used by e.g. Pen (1997) and Sirko (2005). Hahn & Abel (2011) also show that previous implementations that use the inverse Fourier transform of $\delta(\mathbf{k})$ results in too small values for the two-point correlation function.

To calculate the $\delta(\mathbf{r})$ via convolution, it is till required to transform $\widetilde{\mathcal{T}}(k)$ into real space, which MUSIC achieves by performing the calculation (assuming that $\widetilde{\mathcal{T}}(k)$ is spherically symmetric)

$$\mathcal{T}_R(r) = \frac{1}{(2\pi)^3} \int_{\mathbb{R}} \widetilde{\mathcal{T}}(k) e^{i\mathbf{x}\cdot\mathbf{k}} \cdot \mathbf{k} \mathrm{d}^3\mathbf{k} \tag{4.13}$$

$$= \frac{1}{2\pi^2} \int_0^\infty \widetilde{\mathcal{T}}(k) \frac{\sin(kr)}{kr} \mathrm{d}k. \tag{4.14}$$

4.2.3 Creation of nested grids

When creating the refined regions, the white noise of the subgrids must be consistent with the white noise og the coarse domain, meaning that when dividing a particle (parent cell) into smaller particles (children cells), the mass must be conserved. A common way to achieve this has been the Hofmann-Ribak algorithm (Hoffman & Ribak, 1991), which GRAFIC-2 also uses.

The Hofmann-Ribak algorithm first creates a white noise field $w^{\ell+1}$. This field is unconstrained at this point and has a variance 8 times higher than the one level coarses ℓ. 8 times higher variance is equivalent to dividing the parent cells into groups eight children cells. To match the level $\ell + 1$ to ℓ, the mean white noise of the finer level is then matched to the value of the coarser level. Bertschinger (2001) achieves this by subtracting the mean of the unconstrained sample from the sum of the coarse white noise and the unconstrained white noise.

While this algorithm does conserve the mass of the coarse grid on the finer subgrids, the approach does not preserve the Fourier modes of the coarse white noise. Thus MUSIC uses a modified method, which still utilizes the Hofmann-Ribak algorithm while retaining the Fourier modes.

When creating the refined subgrids, the method still divides each parent cell into eight children cells. A two-dimensional example of this is shown in Figure 4.3. After this, the Fourier modes of the coarse grid are preserved by using a Fast Fourier Transform (FFT) on two regions: on the fine grid, and on the coarse grid region which is overlapping with the finer grid. For the finer grid, all of the modes up to

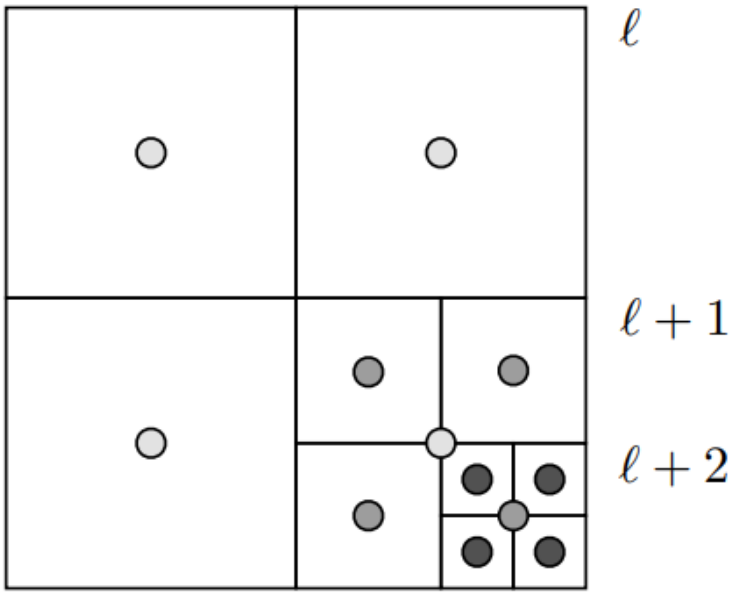

Figure 4.3: A two dimensional example layout of multiple scale nested grid (Hahn & Abel, 2011). The figure shows parent cells on level ℓ, and children cells on two higher levels. The center of the children cells is not in the same position as the center of the parent cell.

the so-called Nyquist wave of the coarse grid, $\mathbf{k} \leq \mathbf{k}_{\mathrm{Ny}}$, are then replaced by with the modes of the coarse level. The value of the Nyquist wave number is $\mathrm{k}_{\mathrm{Ny}} = \pi/\Delta x$, where Δx is the spacing of the particles on the coarse grid. This value acts as a cutoff value for the FFT, as numerical calculations of Fourier transformations cannot be continued to infinity.

The Fourier modes are now preserved, and an inverse Fourier transform can be performed to create the a refined subgrid with perserved Fourier modes. Finally, a reverse version of the Holfmann-Ribak algorithm is used. With this, average of the children cells on the finer level define and replace the the white noise values of the coarser level on the overlapping regions.

With multiple levels of refined subgrids present, this method is started from the finest subgrid, and moving to coarser grids one level at a time. In the regions where levels ℓ and $\ell + 1$ overlap, the parent cells are replaced by the average of the eight children cells and thus, in addition to conserving the Fourier modes, the mass is conserved.

To calculate the over-density fields on each level of the nested grids using FFTs, the convolution kernels $T(r)$ are still needed for all refined levels. For this, the levels of the nested grid are divided into domains Ω_ℓ and each domain is surrounded by

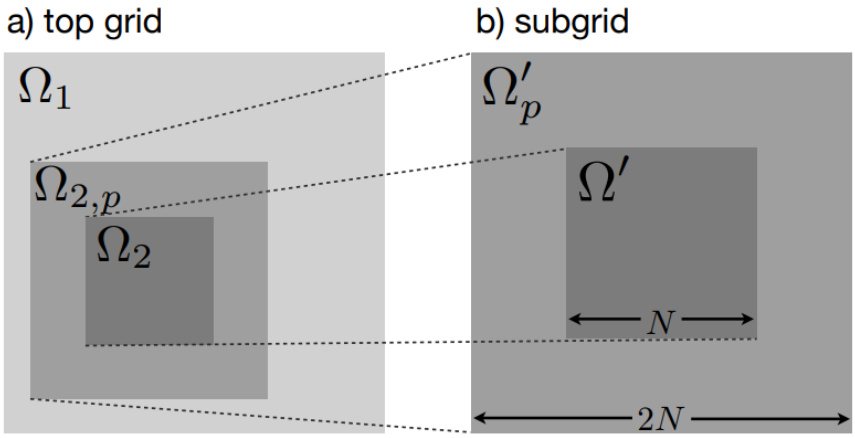

Figure 4.4: A two-dimensional representation of domains of two levels of refinement, including the padded domain for the subgrid. The domain of the higher resolution level has a sidelength of N, and the padded domain has a sidelength of $2N$ (Hahn & Abel, 2011).

a padded domain $\Omega_{\ell,\mathrm{p}}$. The sides of the padded domain are double the length compared to the domain Ω_ℓ, as shown in Figure 4.4. The padded domain is needed to keep the Fourier transforms isolated, which leads to conservation of the wave modes when performing the convolution.

The creation of convolution kernels starts from the finest level ℓ. At this level, the convolution kernel in the domain and the padded domain is simply equivalent to the real-space transfer function,

$$T^\ell(\mathbf{x}) = \mathcal{T}_R(||\mathbf{x}||), \tag{4.15}$$

where $\mathcal{T}_R$ is evaluated as shown in eq. 4.14. The origin is placed at the center of the domain. For the next level, the calculation of the kernel is performed in two different ways, depending on the location for which the kernel is evaluated at. If the location on the level does not have a refined subgrid over it, the calculation for the kernel is performed in the same way as for the most refined level, as shown in eq. . As mentioned before, for the overlapping regions, the kernels all averaged over the eight children cells. This is achieved with the calculation

$$T^{\ell-1}(\mathbf{x}) = T^{\ell-1}(x,y,z) = \frac{1}{8} \sum_{i,j,k\in\{-\frac{\Delta x^\ell}{2},\frac{\Delta x^\ell}{2}\}} T^\ell(x+i, y+j, z+k), \tag{4.16}$$

where Δx^ℓ is the spacing of the particles on level ℓ. The relation of the spacing of particles in refinement levels ℓ and $\ell+1$ is $\Delta x^{\ell+1}/\Delta x^\ell = 2$. After level $\ell-1$, the calculation of kernels moves on to level $\ell-2$, and so on. For these, the kernels are calculated the same way as for level $\ell-1$.

4.2.4 Particle displacements and velocity fields

Note: I'll explain more about the Zeldovich approximation on chapter 2

The generated density field from the white noise sample is used as a starting point for the evolution of particle position and velocity fields. The evolution is calculated using the Lagrangian perturbation theory. Time evolution of position and velocity of each fluid element (i.e. each particle) is followed until the initial redshift z_i is reached. The time evolution of the density field "displaces" particles, turning the initial Gaussian field into a non-Gaussian.

In Lagrangian perturbation theory, the time evolution of position $\mathbf{x}(t)$ and velocity $\mathbf{v}(t)$ is determined by a so-called "displacement field" $\mathbf{L}(\mathbf{q}, t)$, where $\mathbf{q}$ is the initial position of a particle. The time evolution of position and velocity is written as

$$\mathbf{x}(t) = \mathbf{q} + \mathbf{L}(\mathbf{q}, t), \tag{4.17}$$

$$\mathbf{v}(t) = \frac{\mathrm{d}\mathbf{L}(\mathbf{q}, t)}{\mathrm{d}t}. \tag{4.18}$$

For the numerical calculation of the displacement field, MUSIC uses the first order approximation, the so-called Zel'Dovich approximation (Zel'Dovich (1970), see also 2.3.1). With this, the displacement field can be calculated as

$$\mathbf{L}(\mathbf{q}, t) = -\frac{2}{3H_0^2 a^2 D(t)} \nabla\phi(\mathbf{q}, t) \equiv D^{-1}(t) \nabla\Phi(\mathbf{q}, t), \tag{4.19}$$

where $D(t)$ is the growth factor of linear density perturbations and Φ is the potential, whose gradient is proportional to the gradient of the gravitational potential ϕ (Hahn & Abel, 2011). The evolution of the gravitational potential is calculated by solving the Poisson's equation, which is written as

$$\Delta\phi(\mathbf{q}, \mathrm{t}) = \frac{3}{2} H_0^2 a^2 \delta(\mathbf{q}, t). \tag{4.20}$$

With this first order approximation, the velocity of each particle is acquired via a gradient of the potential. Thus the velocity field with this approximation results in $\nabla \times \mathbf{v}(t) = 0$. MUSIC is also capable of calculating the fields using a more compicated second order approximation, but it was not used when generating ICs for the simulations in this thesis.

To achieve a numerical solution for the Poisson's equation, MUSIC uses a hybrid of two methods, FFT and a multi-grid solution called Full Approximation Scheme (FAS, Brandt (1977)) solving the equation on the finest grid and proceeding from

that to coarser levels. With this hybrid method, the FFT is only used for the finest level.

For the levels in which the Poisson's equation is solved using FAS, the equation is written in the form of

$$\Delta\phi(\mathbf{x}) = f(\mathbf{x}). \tag{4.21}$$

The implementation of FAS in MUSIC is thoroughly expained in the paper by Hahn & Abel (2011). In summary, MUSIC approximates the value of ϕ on grid level ℓ to be u^ℓ, with the operator Δ switched to Laplacian value L, taken from a table. These can be used to calculate the residual term $r^\ell = f^\ell - L^\ell u^\ell$. This generates two terms affecting level $\ell - 1$. The second term generates an additional correction term for the grid at level $\ell - 1$. In addition, a restrictive term is present due to the first source term, f^ℓ. To account for these terms, a so-called smoothing scheme $S(u^\ell, f^\ell)$ is performed. This method is appplied using the Gauss-Seidel sweep. This method is applied recursively on each level to decrease the value of the residual r^ℓ until a treshold value is reached, resulting in a excellent approximation for the value of ϕ.

4.2.5 Creating IC files

- Step by step explanation of creating the IC file with a zoom-in region

4.3 GADGET-3 setup for the zoom-in -simulations

4.3.1 Cosmological setup

4.3.2 Low-resolution run

4.3.3 Choosing the zoom-in regions

- FoF -algorithm
- Conditions of the chosen halos
- Figure showing the zoom-in regions from the low res run

4.3.4 Initial conditions

- Information from GADGET3 config files

h_0	Ω_m	Ω_b	Ω_Λ	σ_8	ρ_{crit}
70.3	0.276	0.045	0.724	0.811	$9.28 \times 10^{-27}\ \mathrm{kg/m^3}$

Table 4.1: Cosmological parameters used for the simulations. If a simulation doesn't include baryons, the dark matter density parameter Ω_{DM} is equal to the matter density parameter Ω_{m}. If baryons are included, $\Omega_{\mathrm{DM}} = \Omega_{\mathrm{m}} - \Omega_{\mathrm{b}}$.

5. Cosmological GADGET-3 simulations

5.1 Computational load of the simulations

- Quick overview: CPUs used, time elapsed, where simulations were run

5.2 Locating galaxy centers: the shrinking sphere -method

5.3 Properties of the galaxies

Simulation	r_{vir} (kpc)	M_* ($M_\odot$)	M_{V} (mag)	$M_{*,\mathrm{gal}}/M_{\mathrm{vir}}$	M_{bh} ($M_\odot$)
Med res, A	517	3.98×10^{11}	-22.4	0.025	5.82×10^9
Med res, B	574	5.16×10^{11}	-22.7	0.024	6.23×10^9
Med res, C	400	2.56×10^{11}	-22.0	0.035	3.96×10^9
High res, A	526	5.31×10^{11}	-22.6	0.032	3.64×10^9
High res, B	578	6.38×10^{11}	-22.8	0.029	4.54×10^9
High res, C	400	3.46×10^{11}	-22.2	0.047	2.80×10^9

Table 5.1: Properties of the zoomed-in galaxies at redshift $z = 0$.

5.3.1 Rotation curves

- Med res galaxy A simulation is performed with single precision, I'm currently doing the simulation in double precision
- Rotation curves do not have the same limits on the y-axis

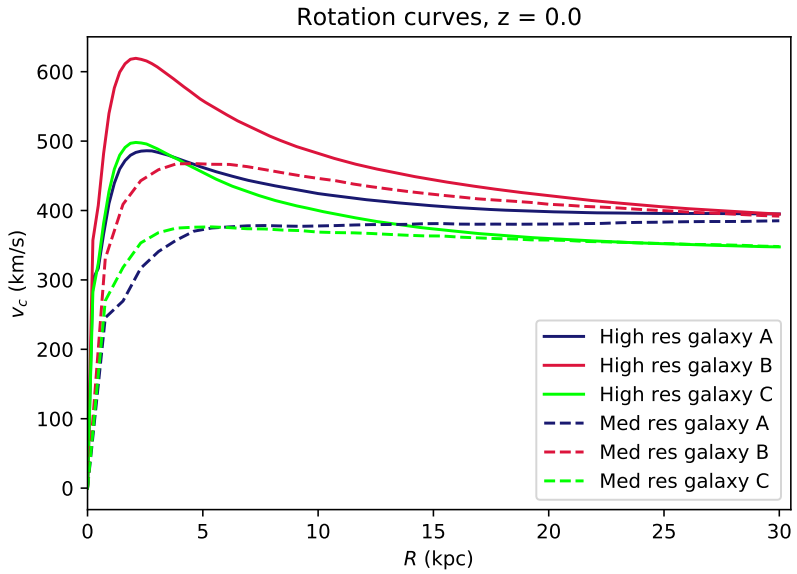

Figure 5.1: Rotation curves for each galaxy including baryons, at redshift 0. The continuous lines represent the high resolution simulations and the dasheld lines represent the medium resolution simulations.

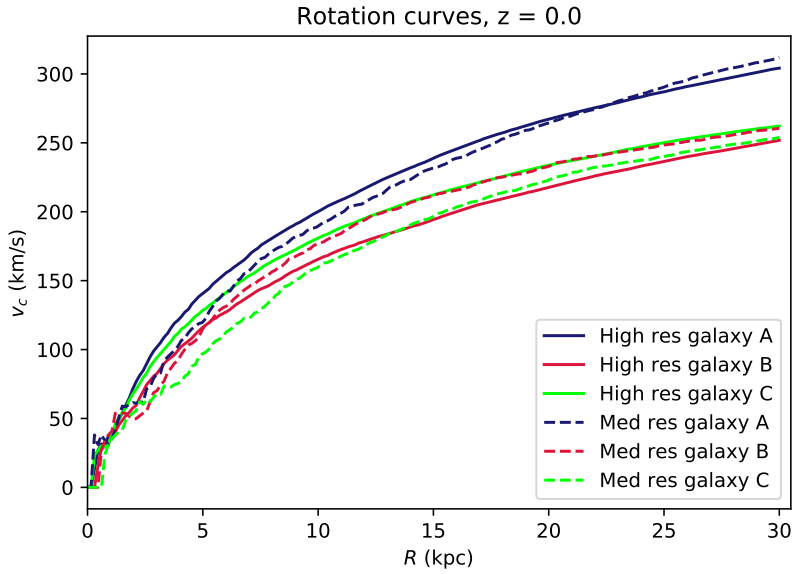

Figure 5.2: Rotation curves for each galaxy including only dark matter, at redshift 0. The continuous lines represent the high resolution simulations and the dasheld lines represent the medium resolution simulations.

5.3.2 Star formation history

- The stellar mass evolution plot is not yet done for high res simulations.

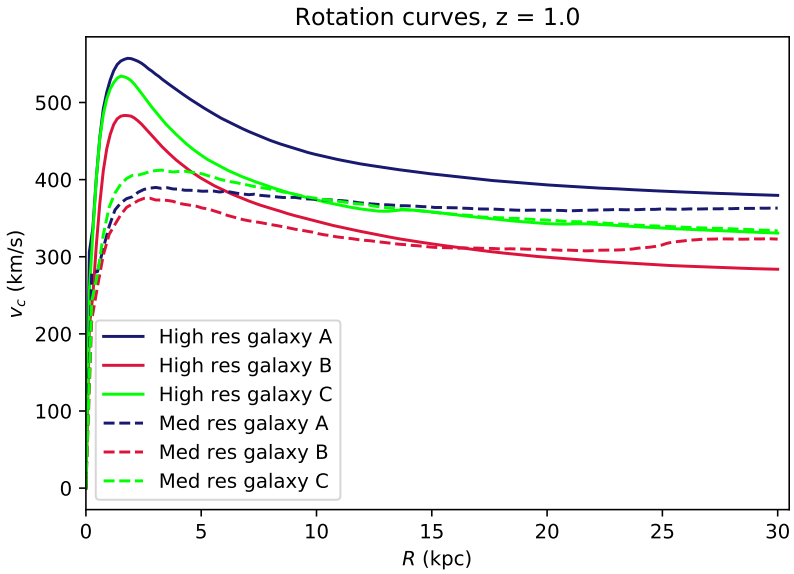

Figure 5.3: Rotation curves for each galaxy including baryons, at redshift 1. The continuous lines represent the high resolution simulations and the dasheld lines represent the medium resolution simulations.

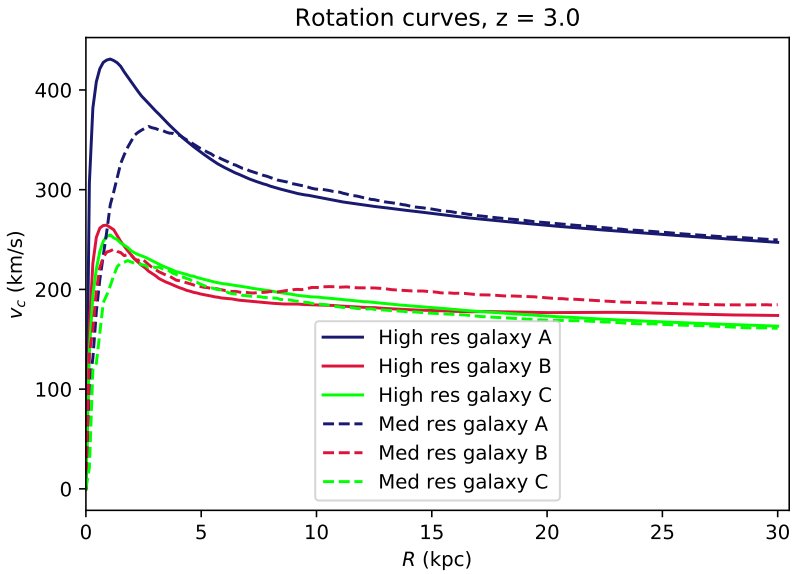

Figure 5.4: Rotation curves for each galaxy including baryons, at redshift 3. The continuous lines represent the high resolution simulations and the dasheld lines represent the medium resolution simulations.

- Redshifts missing from the stellar mass evolution plot
- SFRs, also histograms?

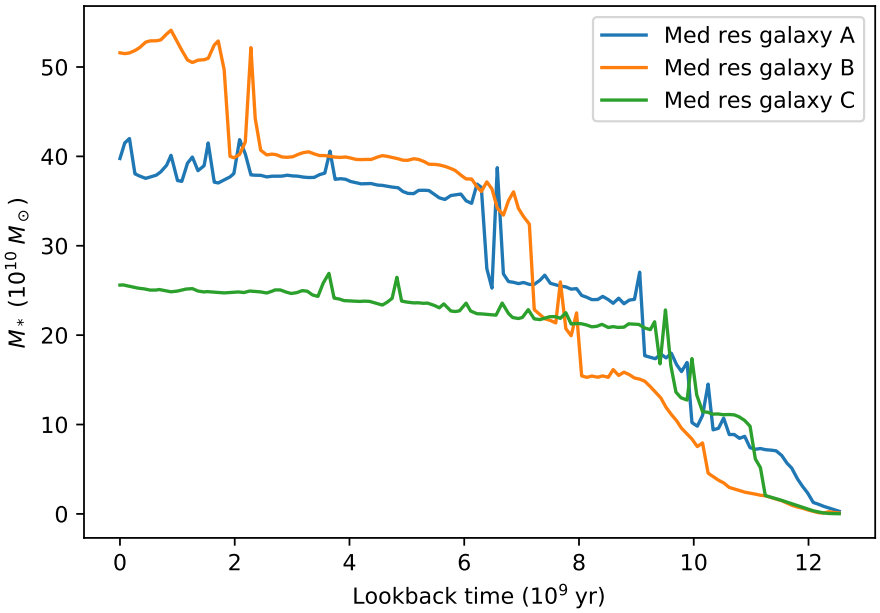

Figure 5.5: Stellar mass evolution for the medium resolution galaxies. The calculated stellar mass is the stellar mass within $r_{\text{gal}} = r_{\text{vir}}/10$.

- Again, med res A results will probably change a bit when double precision run is finished.

- Formation efficiencies, comparing to the cosmological parameter

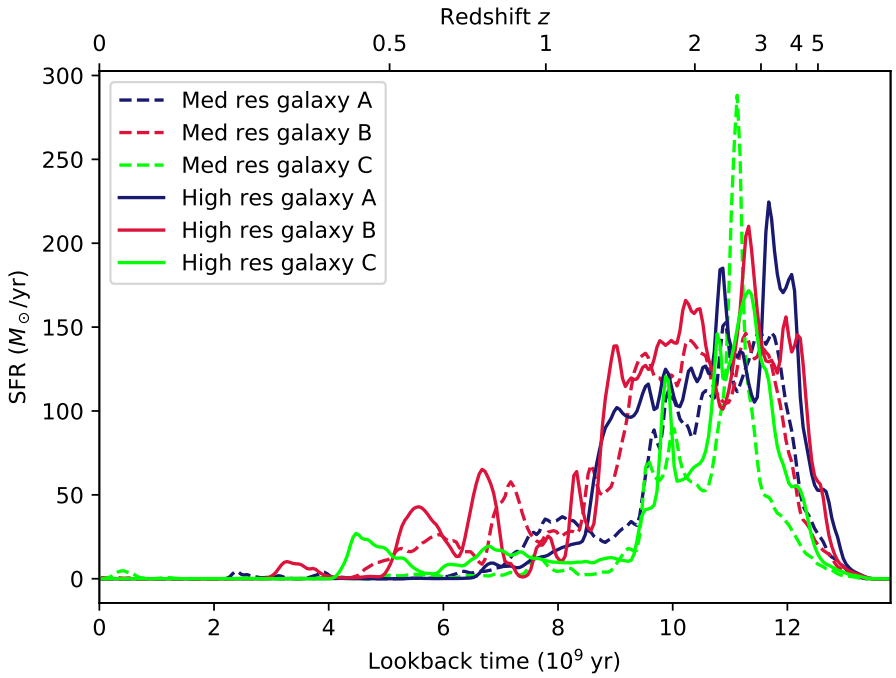

Figure 5.6: Stellar for mation rates for each zoomed-in galaxy, plottes ad a function of lookback time. The lines are created from histograms having a length of 5 Myr, which are then smoothed. The continuous lines represent the high resolution simulations and the dasheld lines represent the medium resolution simulations.

5.3.3 Colors and magnitudes

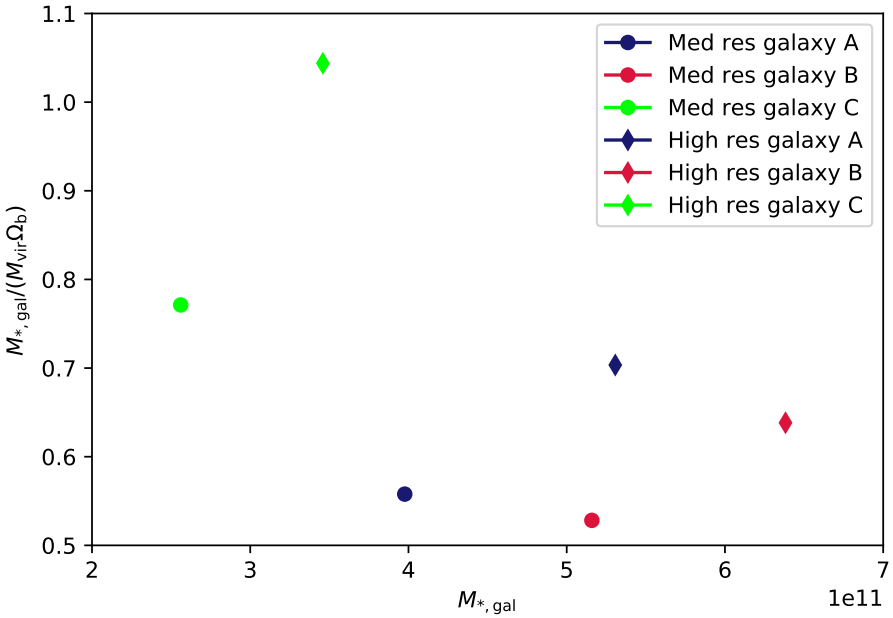

Figure 5.7: Galaxy formation efficiencies for each galaxy, plotted with their stellar masses. The cosmological baryon density Ω_{b} is set to 0.045 in the simulations. The diamond and circular markers show the results of the high resolution and the medium resolution zoom-in simulations, respectively.

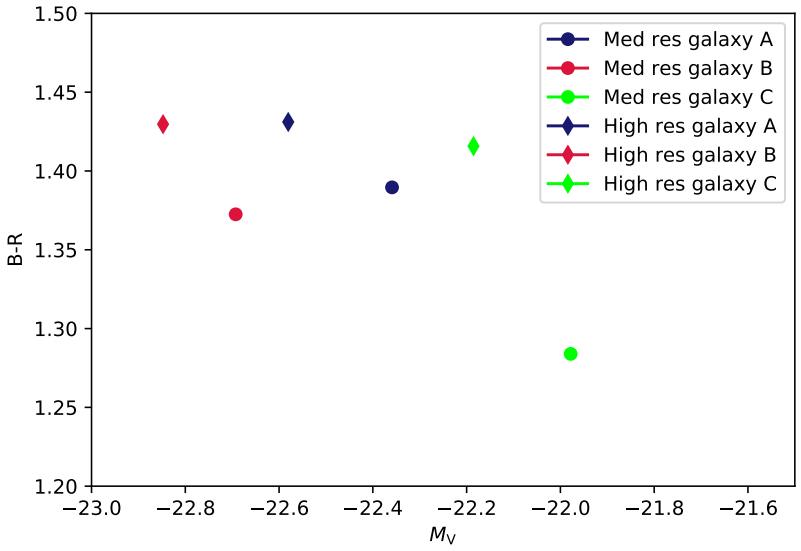

Figure 5.8: B-R colors for each simulated galaxy, plotted with each galaxy's absolute magnitude in the V-band. The diamond and circular markers show the results of the high resolution and the medium resolution zoom-in simulations, respectively.

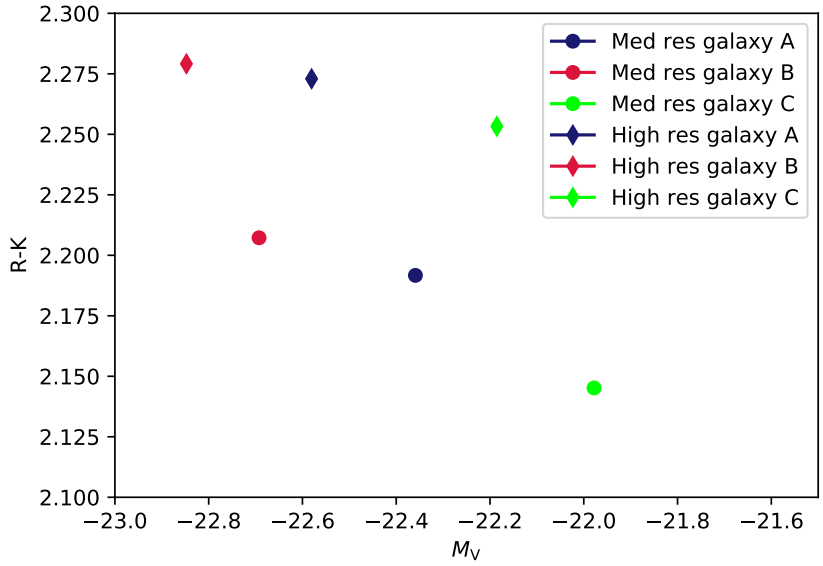

Figure 5.9: R-K colors for each simulated galaxy, plotted with each galaxy's absolute magnitude in the V-band. The diamond and circular markers show the results of the high resolution and the medium resolution zoom-in simulations, respectively.

6. Simulations with KETJU

7. Conclusions

- recap on what was written/studied
- more own thoughts on results
- future missions
- how could the simulations be more realistic (higher resolution, more feedback stuff?)

Bibliography

Bertschinger, E. (2001). Multiscale Gaussian Random Fields and Their Application to Cosmological Simulations. *Astrophysical Journal Supplement Series*, 137(1):1–20.

Brandt, A. (1977). Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, 31(138):333–390.

Hahn, O. & Abel, T. (2011). Multi-scale initial conditions for cosmological simulations. *Monthly Notices of the Royal Astronomical Society*, 415(3):2101–2121.

Hoffman, Y. & Ribak, E. (1991). Constrained Realizations of Gaussian Fields: A Simple Algorithm. *Astrophysical Journal Letters*, 380:L5.

Marinacci, F., Pakmor, R., & Springel, V. (2014). The formation of disc galaxies in high-resolution moving-mesh cosmological simulations. *Monthly Notices of the Royal Astronomical Soviety*, 437(2):1750–1775.

Mo, H., van den Bosch, F. C., & White, S. (2010). *Galaxy Formation and Evolution*.

Navarro, J. F. & White, S. D. M. (1994). Simulations of dissipative galaxy formation in hierarchically clustering universes-2. Dynamics of the baryonic component in galactic haloes. *Monthly Notices of the Royal Astronomical Society*, 267(2):401–412.

Pen, U.-L. (1997). Generating Cosmological Gaussian Random Fields. *Astrophysical Journal Letters*, 490(2):L127–L130.

Power, C., Navarro, J. F., Jenkins, A., Frenk, C. S., White, S. D. M., Springel, V., Stadel, J., & Quinn, T. (2003). The inner structure of ΛCDM haloes - I. A numerical convergence study. *Monthly Notices of the Royal Astronomical Society*, 338(1):14–34.

Reed, D. S., Smith, R. E., Potter, D., Schneider, A., Stadel, J., & Moore, B. (2013). Towards an accurate mass function for precision cosmology. *Monthly Notices of the Royal Astronomical Society*, 431(2):1866–1882.

Salmon, J. (1996). Generation of Correlated and Constrained Gaussian Stochastic Processes for N-Body Simulations. *Astrophysical Journal*, 460:59.

Sirko, E. (2005). Initial Conditions to Cosmological N-Body Simulations, or, How to Run an Ensemble of Simulations. *Astrophysical Journal*, 634(2):728–743.

Zel'Dovich, Y. B. (1970). Gravitational instability: an approximate theory for large density perturbations. *Astronomy & Astrophysics*, 500:13–18.