Trabajo de Fin de Máster

# RECONSTRUCTION OF PRIMORDIAL FLUCTUATIONS FROM GALAXY CATALOGS WITH HIGHER ORDER HAMILTONIAN SAMPLING

Mónica Hernández Sánchez

Supervised by

Francisco-Shu Kitaura

and

Claudio Dalla Vecchia

Universidad de La Laguna

Máster en Astrofísica

July 2018

# Resumen

Las diferentes estructuras que observamos en el Universo son el resultado de la evolución de las semillas primordiales, cuyo origen se cree que proviene de fluctuaciones cuánticas. En un período de inflación cósmica, estas fluctuaciones microscópicas fueron amplificadas en una fase de muy rápida expansión apenas $10^{-36}$ segundos después del Big Bang, dando lugar a un Universo prácticamente homogéneo y altamente gausiano. Sin embargo, en algunas regiones, la densidad era ligeramente superior a la media, produciéndose inestabilidades gravitacionales, haciendo que estos picos de densidad atrajesen más materia de sus alrededores. De esta manera, se originó la compleja red cósmica que observamos en surveys galácticos, la cual está compuesta por la agrupación de cúmulos de galaxias en nodos y filamentos, y grandes vacíos cósmicos. La evolución de las fluctuaciones primordiales se puede describir con las ecuaciones de un fluido, haciendo uso de la teoría de perturbaciones en un sistema de referencia Lagrangiano (teoría de perturbaciones Lagrangiana), que resulta ser muy útil para describir la formación de la red cósmica. Dentro de la estructura a gran escala podemos diferenciar dos regiones: aquella a grandes escalas, donde la evolución de las fluctuaciones primordiales sigue aproximadamente un crecimiento lineal, y a escalas intermedias y pequeñas, donde la gravedad es no lineal. En este régimen no lineal, la gravedad acopla distintos modos del espacio de Fourier por ser una interacción de largo alcance y altera la distribución gausiana primordial, introduciendo desviaciones importantes de una distribución simétrica normal. Esto hace muy complejo el análisis de la estructura a gran escala, requiriéndose, para ello, técnicas avanzadas que nos permitan caracterizar la red cósmica.

En este contexto, escogemos una descripción Bayesiana, construyendo la función de probabilidad posterior que se quiere muestrear como el producto de un *prior*, describiendo la distribución continua de la materia oscura, y una *likelihood*, describiendo la distribución discreta de las galaxias. Esto nos permite, en un principio, obtener los campos de materia oscura que son compatibles con un catálogo de galaxias. Para ello, se requiere elegir unas funciones de probabilidad concretas, incluyendo un modelo de formación de estructuras que relacione el campo de densidad primordial con el evolucionado gravitacionalmente y una técnica concreta de muestreo estadístico.

Comenzamos, en primer lugar, considerando un problema simplificado, en el cual asumimos un modelo de formación de estructuras donde el logaritmo del campo de densidad de la materia oscura sigue una distribución gausiana (modelo lognormal). La distribución de galaxias se asume que sigue una distribución poisoniana. En este proycto se ha empleado el código `ARGO`, que utiliza una técnica de muestreo híbrida de cadenas de Markov basada en *Hamiltonian sampling*. Esta técnica tiene la ventaja de ser capaz de muestrear funciones de probabilidad arbitrarias (incluyendo no gausianas) y de forma eficiente, usando los gradientes de los potenciales determinados por la función posterior. Para ello, se define la energía potencial como el negativo del logaritmo de la función posterior que queremos muestrear y la energía cinética referida a unos momentos que son aritificialmente introducidos en el método para posibilitar el muestreo aleatorio. Éstos tienen la ventaja de ser descritos por una simple gausiana. El método consiste en muestrear la probabilidad conjunta de la señal que queremos reconstruir (los campos de materia oscura), que en la analogía hamiltoniana representan las posiciones; y los momentos. La marginalización del muestreo con respecto a los momentos, se logra despreciándolos de la iteración anterior en la cadena de Markov, introduciendo nuevos valores de los mismos en cada iteración de acuerdo a una gausiana con una matriz de covarianza determinada por la masa.

El sistema de posiciones y momentos se evoluciona de acuerdo con las ecuaciones de Hamilton, que se resuelven de manera discreta y se van aceptando o rechazando en un paso de Metropolis-Hastings. Para la conservación del volumen del espacio de las fases se usa un esquema discreto de *Leapfrog*, hasta ahora considerado sólo hasta segundo orden. En este trabajo, hemos implementado por primera vez en el campo de la cosmología estructura a gran escala, esquemas de *Leapfrog* de mayor orden. En concreto, nos enfocamos en el cuarto orden, aunque el marco teórico que introducimos admite órdenes mayores. Esto nos permite evolucionar el sistema hamiltoniano con pasos mucho más amplios que a segundo orden y obtener una mayor aceptación de las iteraciones. Todo esto se traduce en un incremento de la eficiencia del método por un factor 18 en la convergencia de la cadena de Markov, en términos de tiempo computacional, requiriendo dos órdenes de magnitud menos iteraciones que en el caso anterior. Además, obtenemos una longitud de correlación entre los campos de materia muestreados de unas 10 iteraciones frente a las $\sim 300$ requeridas en el método a segundo orden. Esto implica que podemos obtener muestras independientes cada 10 iteraciones una vez se alcanza la convergencia en unas 30 iteraciones.

En un segundo estudio, usamos el código `BIRTH`, que incluye teoría de perturbación lagrangiana para la reconstrucción de las fluctuaciones primordiales, y lo aplicamos a un problema realista con un catálogo de galaxias emulando las galaxias rojas luminosas CMASS de BOSS final data release. Este catálogo incluye evolución cósmica en el cono de luz, velocidades peculiares de las galaxias, geometría del survey y función de selección radial. Para este código, obtenemos los mismos resultados en términos de eficiencia global que en el caso simplificado con el *Leapfrog* de cuarto orden.

A partir de una reconstrucción del campo primordial de fluctuaciones obtenida con el código `BIRTH`, realizamos una simulación de $N$-cuerpos con el código `PKDGRAV3` hasta $z = 0.57$, que corresponde con el redshift medio del catálogo de CMASS. Comprobamos que las estructuras finales de materia oscura guardan una relación espacial muy similar a las condiciones iniciales obtenidas a patir del catálogo. Esto nos muestra una de las aplicaciones interesantes del método desarrollado en este trabajo de fin de máster, dado que nos permitirá una comparación más directa entre los datos y los modelos. La eficiencia obtenida con el método Hamiltoniano desarrollado en este trabajo nos permitirá acometer muchos más estudios en un análisis Bayesiano global, tales como incluir un tratamiento totalmente no lineal de la gravedad en el proceso iterativo de reconstrucción.

## *Abstract*

The different structures we observe in the Universe result from the cosmic evolution starting from some primordial seeds. These are believed to have originated from some quantum fluctuations. During the inflationary epoch, they were stretched and frozen just $10^{-36}$ seconds after the Big Bang, giving rise to an almost homogeneous Universe closely Gaussian distributed. However, these fluctuations were slightly larger to the mean in some regions, causing gravitational instabilities, so that density peaks were attractors to their surrounding matter. In this way, a complex non-Gaussian cosmic web emerged, composed by the clustering of galaxies in knots, filaments and sheets; and cosmic voids, where the mean density is very low. This makes the analysis of the large scale structure very complex, so advanced techniques are required to characterize this cosmic web. We choose, in this context, a Bayesian framework, which allows us to approach the problem from a robust statistical perspective. We start considering a simplified problem, assuming a lognormal prior model for the density field and a Poisson likelihood for the galaxy distribution. Using the `ARGO` code we are able to sample the posterior resulting from the product of the prior and likelihood with a hybrid Markov Chain Monte Carlo method: the Hamiltonian sampling technique. In this way, we obtain a dark matter field in each Markov iteration, which is statistically compatible with the input galaxy distribution, given the model. In the Hamiltonian analogy, the dark matter fields represent the positions, and the momenta are artificially introduced to enable sampling arbitrary posterior distribution functions. They are sampled from a Gaussian distribution with a given mass. The Hamiltonian system is evolved through the discretized equations of motions, where the final positions and momenta are accepted or rejected according to a Metropolis-Hastings step. Until now, a second order Leapfrog scheme has been used to solve this problem. We explore in this thesis higher order discretizations, focusing, in particular, on fourth order Leapfrog. The result is a factor 18 faster convergence in terms of computational time, with respect to the original second order Leapfrog scheme. Moreover, we obtain a correlation length of about 10 iterations, as opposed to $\sim 300$ with the old scheme. This implies that we can obtain independent samples each 10th iterations once we reach convergence, which is about 30 iterations, two orders of magnitude less than with the second order algorithm.

In a second study, we used the `BIRTH` code, which includes Lagrangian perturbation theory for the reconstruction of the primordial fluctuations. We apply this code, which also relies on Hamiltonian sampling, to a realistic mock galaxy catalog resembling the BOSS-CMASS final data release. This catalog includes cosmic evolution in the light-cone, peculiar velocities, survey geometry, radial selection function. We obtain the same results in terms of global efficiency, as in the simplified case study, when we implement the fourth order Leapfrog scheme.

Using one of the reconstructed primordial fluctuations with the `BIRTH` code, we make a constrained $N$-body simulation with `PKDGRAV3` to $z = 0.57$, which corresponds to the mean redshift of CMASS galaxies. We observe a high spatial resemblance between the obtained dark matter structures and the initial conditions obtained from the catalog. This shows one of the interesting applications of the method developed in this thesis, as it permits a more direct comparison between data and models. The efficiency of the higher order Hamiltonian method developed here will enable us to tackle many more complex studies within a global Bayesian framework, such as including a full non-linear gravity solver within the iterative reconstruction process.

## Contents

# 1 Introduction

Our local Universe is far from homogeneous and isotropic, showing a variety of structures in different parts of the sky. Clusters of galaxies are grouped in superclusters, which are the knots of a network of dense filaments and walls. These are the largest structures we know, and they form the boundaries of cosmic voids, which have a very low mean density. All these components make up what we call *the Cosmic Web*. However, when we go to large scales ($\geq 100$ Mpc), the Universe starts to look more and more homogeneous.

On such large scales, the universal density distribution is generally assumed to rise from a homogeneous and isotropic spatial Gaussian process. The *Cosmological Principle* states that, as we are not located at any special place in the Universe (the so-called *Copernican Principle*), every fundamental observer at the same cosmic epoch observes the same Hubble expansion of the distribution of galaxies, the same isotropic Cosmic Microwave Background and the same large scale structure. Therefore, on sufficiently large scales the Universe is both homogeneous and isotropic, being the homogeneity defined in an average sense. Isotropy does not necessary imply homogeneity without introducing the additional assumption that the observer is not, as we have said, in a special place. One would observe isotropy in any spherically symmetric distribution of matter, but only if one were in the center of the pattern. So observed isotropy, together with the Copernican Principle let us formulate the Cosmological Principle [2].

Structures have been formed from the gravitational instability of small primordial density inhomogeneities (perturbations). Although we do not know exactly their origin, the most accepted theory is that of inflation, where the microscopic quantum fluctuations were amplified by a period of rapid expansion to macroscopic scale. So the large scale structure of the expanding Universe has grown by inflation of quantum fluctuations, that were stretched and frozen $10^{-36}$ seconds after the Big Bang. Adopting the most recent set of cosmological parameters, which declare a spatial flat space with an accelerated expansion due to the density parameter of the dark energy, $\Omega_\Lambda$; initial conditions of the early Universe are assumed to be closely Gaussian distributed.

At scales of megaparsecs, perturbations start to evolve through the influence of gravity: due to primordial fluctuations, the concentration of matter was slightly higher than the average in some regions of the Universe. These density peaks attract more matter from their surroundings, originating a process of collapse into more dense structures and generating a non-Gaussian distribution of matter. Moreover, while in the *hot dark matter* paradigm structures are formed by fragmentation (top-down), the *cold dark matter* theory, whose predictions are in agreement with the observations of the large scale structure, is described by a hierarchical clustering: the first objects to form out of the primordial fluctuations are those that are generated by the collapse the smallest perturbations. These objects subsequently merge with other objects and form larger ones (bottom-up).

Therefore, we can describe the distribution of matter in the Universe at a given time dividing it into volumes which initially evolve independently of each other. However, as the gravitational forces between closer regions become stronger, this independence would no longer hold, and the different modes of the Fourier space become coupled, generating deviations in the primordial Gaussian distribution. The Cosmic Microwave Background is accurately described by linear perturbation theory, however, the perturbations in the matter density of the Universe at smaller scales start to become non-linear. We can divide the large scale structure in two regimes: large linear scales, where the evolution of fluctuations is close to linear growth and perturbation theory converges to the correct result if we go to sufficiently high order terms; and smaller scales,

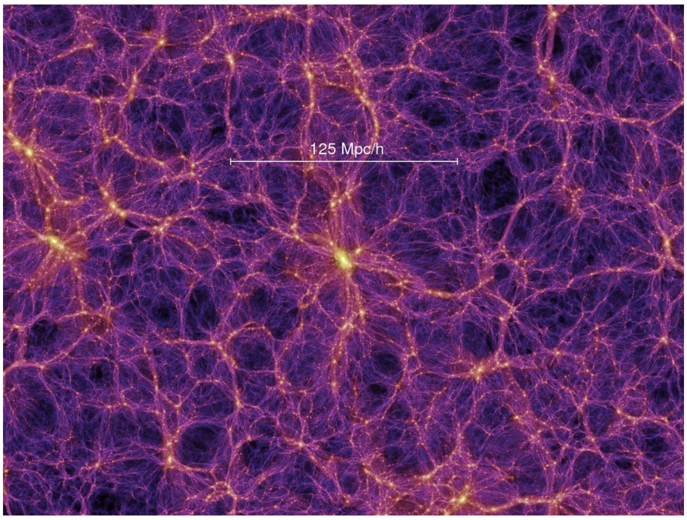


Figure 1: Millennium simulation of Cosmic Web. The figure shows the density distribution of the matter un the Universe at $z = 0$. Credits: The Millennium Simulation (https://wwwmpa.mpa-garching.mpg.de/galform/virgo/millennium/)

where gravitational clustering takes us into a non-linear regime, for which perturbation theory becomes inaccurate. Therefore, due to the non-linear nature of gravity on these scales, N-body simulations gives the best theoretical prediction: the dark matter fluid is sampled in phase space using as many macro-particles as possible, each one representing a set of dark matter particles, evolving without collision under the effect of their mutual gravitational attraction.

## 1.1 The ΛCDM model

The ΛCDM model, also known as the *the Standard Model*, describes our current cosmological picture of the Universe. It is based in a hot big bang as the origin of space-time and it assumes that our Universe today is dominated by the cosmological constant, Λ, associated with dark energy, that was first introduced by Einstein to obtain a compatible solution with a static Universe; together with the *Cold Dark Matter* (CDM). The main observations which led the present ΛCDM model to be the most accepted one are: the Cosmic Microwave Background (CMB), to be the best observational confirmation of the Big Bang model; and the type Ia supernovae, through distance measurements, which constrain $\Omega_m$ and $\Omega_\Lambda$ by the Supernova Cosmology project [3]. The obtained supernova data imply that there is a significant, positive cosmological constant and, therefore, the Universe today is undergoing an accelerated expansion. Moreover, the Baryonic Acoustic Oscillations (BAO), originated when photons and baryons decoupled in the primordial baryon-photon plasma, are until now also in agreement with this model.

The basis of this model is the hypothesis of a homogeneous and isotropic Universe at large scales, as is it previously described. Therefore, the field equations of General Relativity are reduced to the *Friedmann equations*[4]:

$$\left(\frac{\dot{a}}{a}\right) = \frac{8\pi G}{3}\rho - \frac{k}{a^2} + \frac{\Lambda}{3} \tag{1}$$

$$\frac{\ddot{a}}{a} = -\frac{4\pi G}{3}(\rho + 3P) + \frac{\Lambda}{3}. \tag{2}$$

$a$ is the scale factor, usually express as:

$$\frac{a}{a_0} = \frac{1}{z+1}, \tag{3}$$

where $a_0 = a(t=0)$. $k$ is the curvature parameter, that can take the values: $k = -1, 0, 1$ and indicates an open, flat and closed geometry of space, respectively. $G$ is the gravitational constant and $\rho$ is the energy density, given by the sum of the matter component and the radiation one:

$$\rho = \rho_m + \rho_{rad}. \tag{4}$$

The matter energy density is the sum of the dark matter term and the baryons one, $\rho_m = \rho_{dm} + \rho_b$. The density associated with the Cosmological Constant is:

$$\rho_\Lambda = \frac{\Lambda}{8\pi G}. \tag{5}$$

The Friedmann equations 1 and 2 represent the energy balance and the balance of forces in the Universe, respectively.

The ΛCDM model can be described by the following six parameters:

- Today's value of the Hubble parameter, $H_0$, usually defined as: $h = \frac{H_0}{100} \frac{\text{kms}^{-1}}{\text{Mpc}}$. The Hubble parameter is a measurement of the expansion rate defined as:

$$H = \frac{\dot{a}}{a} \tag{6}$$

- The physical baryon density: $\Omega_b h^2$.
- The physical CDM density: $\Omega_{dm} h^2$.

  These two expressions are defined with the dimensionless energy density, $\Omega$:

$$\Omega = \frac{\rho}{\rho_c}, \tag{7}$$

  where $\rho_c$ is the critical density, which is the density where the curvature parameter is zero: $\rho = \rho_c$ at $k = 0$:

$$\rho_c = \frac{3H^2}{8\pi}. \tag{8}$$

- The spectral index of the primordial power spectrum[1]: $n_s$
- The $\sigma_8$ parameter, which is the r.m.s of the density perturbation when it is smoothed with a top-hat filter[2] of $8h^{-1}$ Mpc [5]:

$$\epsilon(r) = \frac{1}{2\pi^2} \int P(k) k^2 W(k) dk \tag{9}$$

  where $W(k)$ is the top-hat filter function.

- The reionization optical depth: $\tau$. The intergalactic medium evolved from a largely neutral state (post-recombination) to one of being largely ionized. The optical depth to reionization, $\tau$, is a quantity which provides a measure of the line-of-sight free-electron opacity to CMB radiation. It is computed as the integral of the electron density, $n_e$, times the Thomson cross section over the geometrical path length between $z = 0$ and $z$ where the reionization takes part [6].

[1] In section 2.3 we will explain where this parameter comes from.

[2] Also in section 2.3 we will see more information about filtering.

The Standard model defines a Universe with a null curvature, $k = 0$, as a deduction from the anisotropies of the CMB, with the following parameter values[3]: $h = 0.678$, $\Omega_m = 0.307$, $\Omega_b = 0.048$, $\sigma_8 = 0.8228$ and $n_s = 0.96$.

## 1.2 Motivation and objectives

Nowadays, the improvement of galaxy surveys is generating such an amount of data that is leading cosmology into the data science world. Therefore, *big data* methods have to be develop to deal with this challenge. Although observations are a crucial piece in the advance of cosmology, most of the matter in the Universe is dark, however, ΛCDM model predicts that the dynamics of galaxies are driven by the underlying dark matter, existing a difference between them, known as *bias*, that is needed to be modeled. On the other hand, due to the non-linear behavior of gravity at small scales, the large scale structure is difficult to analyze, requiring advanced techniques which allow us to characterize the Cosmic Web. In this context, we choose a statistical Bayesian framework, which permits us to obtain the compatible dark matter field with a galaxy catalog. In this project we use `ARGO` code, where the probability density function (PDF) of the density field is sampled, given a set of galaxy coordinates $(x, y, z)$. To do so, the code uses a lognormal prior model for the density field and a Poisson likelihood for the galaxy distribution[4], through the Hamiltonian Monte Carlo (HMC) sampling. This method defines a Hamiltonian system, where the positions are related to the dark matter field, and the momenta are artificially introduced to sample arbitrary posterior distribution functions; and evolve it through the discretized Hamilton's equations, where the most common used scheme is the Leapfrog algorithm.

This project aims to increase the efficiency of the method. To do so, a higher order discretization of the equation of motion has been implemented, in particular, the fourth order Leapfrog algorithm, until now, being the second order Leapfrog the one used in the field. With this higher order algorithm we will be able to reach the convergence of the Markov chain faster than with the original one. This improvement is going to be study with some tests, such as measurements of the number of iterations required to reach the convergence in the Markov chain, the computational time needed for the code to achieve this convergence, the percentage of acceptance of the iterations, the Gelman-Rubin test, and the correlation length. With this last one we can estimate the number of iterations required by the code to obtain independent samples, after reaching the convergence. Moreover, a study of the optimal *stepsize* to sample the parameter space with the HMC sampling is also included in this work.

For this project, we use a realistic mock catalog resembling the BOSS-CMASS distribution of galaxies. This catalog was built from observations of Luminous Red Galaxies (LRGs) and gives a three dimensional spatial view of the large scale structure in the Universe. With the main goal of obtaining the reconstruction of the large scale structure, we are going to obtain the initial conditions from this catalog at $z = 60$ using the `BIRTH` code, which has a more advanced method in the reconstruction of the primordial fluctuations, and where the fourth order Leapfrog algorithm has also been implemented. Then, we are going to generate the *whitenoise* of the initial conditions and, with the implementation of the N-body code `PKDGRAV3`, we are going to read it. This code generates again the initial conditions and evolve them through the N-body simulation until $z = 0, 57$, which is the mean redshift at where the BOSS galaxy samples are located.

[3] Taken from the BigMD simulation: https://www.cosmosim.org/cms/simulations/bigmdpl/

[4] We will see this in more detail in section 3

# 2 Primordial Fluctuations and perturbation theory

This chapter shows some important meaningful concepts for the development of this project. Moreover, a review of the formalism to analyze the evolution of the primordial fluctuations is also given, starting from the equations of a fluid and solving them, focusing on the linear approximation of the Perturbation Theory and the Zel'Dovich approximation. Finally, a structure formation model beyond the linear theory is discussed: the Lognormal model.

## 2.1 Primordial fluctuations

We can define the primordial fluctuations through the *overdensity field* or *density contrast*:

$$\delta(\vec{x}) = \frac{\rho(\vec{x}) - \overline{\rho}}{\overline{\rho}}, \tag{10}$$

where $\rho(\vec{x})$ is the density distribution of matter at location $\vec{x}$ and $\overline{\rho}$ is the mean density of the Universe.

Equation 10 is often described in Fourier space:

$$\delta(\vec{x}) = \sum_k \delta(\vec{k}) e^{i\vec{k}\cdot\vec{x}} \tag{11}$$

We represent the Universe dividing it into cubic cells of volume $V = L^3$ with periodic boundary conditions: $\delta(L, y, z) = \delta(0, y, z)$, so the wavevector has the components:

$$k_x = n_x \frac{2\pi}{L},\ k_y = n_y \frac{2\pi}{L},\ k_z = n_z \frac{2\pi}{L} \tag{12}$$

where, $n_x,\ n_y,\ n_z$ are integers.

The Fourier coefficients $\delta(\vec{k})$ are complex quantities given by:

$$\delta(\vec{k}) = \frac{(2\pi)^3}{L^3} \int_V \delta(\vec{x}) e^{-i\vec{k}\cdot\vec{x}} d^3\vec{x} \tag{13}$$

As it is described before, if we take averages over larger scales the inhomegeneities becomes less. This can be formalized using the 2-point correlation function, which expresses the probability of finding a galaxy at a distance $\vec{r}$ from a galaxy selected in a uniform, random distribution. It can be defined in terms of the distribution of galaxies in space, with the probability of finding pairs of galaxies separated by distance $\vec{r}$, or in terms of the density contrast as it follows [7]:

$$\epsilon(\vec{x}_1, \vec{x}_2) = \langle \delta(\vec{x}_1) \delta(\vec{x}_2) \rangle. \tag{14}$$

Equation 14 has a positive correlation if the density perturbation has the same sign at $\vec{x}_1$ and $\vec{x}_2$; and negative when there is overdensity at one and underdensity at the other. It proves how density perturbations at different locations are correlated with each other, if galaxies are clustered they must be correlated.

We can define $\vec{r} = \vec{x}_1 - \vec{x}_2$, so due to statistical homogeneity the 2-point correlation function depends only on $\vec{r}$:

$$\epsilon(\vec{r}) = \langle \delta(\vec{x})\delta(\vec{x}+\vec{r}) \rangle \tag{15}$$

and it is independent of direction due to statistical isotropy, so $\epsilon(\vec{r}) = \epsilon(r)$.

Taking the inverse Fourier transform of equation 13 and the previous equation 15, we obtain:

$$\begin{aligned} \epsilon(r) &= \langle \delta(\vec{x})\delta(\vec{x}+\vec{r}) \rangle = \left\langle \int \frac{L^3}{2\pi^3} \int \frac{L^3}{2\pi^3} \delta(\vec{k})\delta^*(\vec{k}')e^{-i\vec{k}'\vec{r}}e^{-i\vec{k}'(\vec{r}+\vec{x})}d^3\vec{k}d^3\vec{k}' \right\rangle \\ &= 2\pi \left\langle \int \frac{L^3}{2\pi^3} \int \frac{L^3}{2\pi^3} \delta^D(\vec{k}-\vec{k}')\langle|\delta(\vec{k})|^2\rangle e^{-i(\vec{k}'-\vec{k})\vec{r}-i\vec{k}\vec{x}}d^3\vec{k}d^3\vec{k}' \right\rangle \\ &= \frac{L^3}{2\pi^3} \int \langle|\delta(\vec{k})|^2\rangle e^{-i\vec{k}\vec{x}}d^3\vec{k} \end{aligned} \tag{16}$$

Moreover, the correlation function at zero separation gives the variance of the density perturbation:

$$\sigma^2 = \langle \delta^2 \rangle = \langle \delta(\vec{x})\delta(\vec{x}) \rangle \equiv \epsilon(0) \tag{17}$$

## 2.2 Perturbation theory

What we need to know now is the equation that describes the evolution of $\delta(\vec{x})$. Before recombination, photons and baryons were tightly coupled as a fluid. When photons decoupled from baryons, these last ones behaved as an ideal gas. On the other hand, dark matter is assumed to be a collisionless fluid, and, for these reasons, we can describe $\delta(\vec{x})$ using fluid equations.

The time evolution of a fluid is given by:

- The continuity equation, which results from mass conservation:

$$\frac{\partial \rho}{\partial t} + \vec{\nabla}_r \cdot (\rho \vec{v}) = 0 \tag{18}$$

- The Euler equation, which express the force acting on a fluid due to the gradient of the pressure and a gravitational potential, describing moment conservation:

$$\frac{\partial \vec{v}}{\partial t} + (\vec{v} \cdot \vec{\nabla}_r)\vec{v} = -\frac{\vec{\nabla}_r P}{\rho} - \vec{\nabla}_r \Phi \tag{19}$$

- The Poisson equation, describing the potential induced by the mass inhomogenity:

$$\vec{\nabla}_r^2 \Phi = 4\pi G \rho \tag{20}$$

During the Matter-Dominated epoch, at $k > aH$, the General Relativity equations were reduced to Newtonian ones, that are going to be developed then [8].

We are going to take $\vec{x}$ as the comoving coordinates, such that:

$$\vec{r} = a(t)\vec{x} \tag{21}$$

where $\vec{r}$ are the physical coordinates.

The physical velocity can be written as:

$$\frac{d\vec{r}}{dt} \equiv \vec{v} = \dot{a}\vec{x} + a\frac{d\vec{x}}{dt} = H\vec{r} + \vec{u} \tag{22}$$

where $H = \frac{\dot{a}}{a}$, which is the conformal expansion rate as we saw in equation 6; and $\vec{u}$ is the peculiar velocity.

We can rewrite the two first equations (18 and 19) defining the time derivative (lagrangian):

$$\frac{d}{dt} = \frac{\partial}{\partial t} + \vec{v} \cdot \vec{\nabla}_r, \tag{23}$$

which for the Continuity equation implies:

$$\frac{d\rho(\vec{x},t)}{dt} = -\rho(\vec{x},t)\vec{\nabla}_r\vec{v} = -3H(\vec{x},t)\rho(\vec{x},t), \tag{24}$$

where

$$H(x,t) = \frac{1}{3}\vec{\nabla}_r(H\vec{r} + \vec{u}) = H(t) + \frac{1}{3}\vec{\nabla}_r\vec{u} \tag{25}$$

is the locally defined Hubble parameter.

For the Euler equation we have:

$$\frac{d\vec{v}}{dt} = -\frac{\vec{\nabla}_r P}{\rho} - \vec{\nabla}_r\Phi \tag{26}$$

The equations of motions for $\rho, P$ and $\Phi$, perturbed about their background values, are the following ones:

$$\begin{aligned} \rho(\vec{x},t) &= \overline{\rho} + \delta\rho(\vec{x},t) = \overline{\rho}[1 + \delta(\vec{x},t)] && (27)\\ P(\vec{x},t) &= \overline{P} + \delta P(\vec{x},t) && (28)\\ \Phi(\vec{x},t) &= \overline{\Phi}(\vec{x},t) + \phi && (29) \end{aligned}$$

Here, $\delta$ is the fractional overdensity in the fluid, and $\phi$ is the perturbed gravitational potential.

Converting the partial derivatives $(\vec{r},t)$ into $(\vec{x},\tau)$, where $\tau$ is the conformal time expressed as $\tau = \frac{1}{a}t$; we have:

$$\frac{\partial}{\partial t} = \frac{\partial\tau}{\partial t}\frac{\partial}{\partial\tau} + \frac{\partial\vec{x}}{\partial t}\frac{\partial}{\partial\vec{x}} = \frac{1}{a}\frac{\partial}{\partial\tau} - H\vec{x}\cdot\vec{\nabla}_x \tag{30}$$

$$\frac{\partial}{\partial\vec{r}} = \frac{\partial\tau}{\partial\vec{r}}\frac{\partial}{\partial\tau} + \frac{\partial\vec{x}}{\partial\vec{r}}\frac{\partial}{\partial\vec{x}} = \frac{1}{a}\vec{\nabla}_x \tag{31}$$

The Continuity equation is, therefore:

$$\begin{aligned}\frac{\partial\rho}{\partial t} &= (1+\delta)\frac{\partial\overline{\rho}}{\partial t}+\overline{\rho}\frac{\partial\delta}{\partial t} = -3(1+\delta)H(t)\overline{\rho}+\overline{\rho}\frac{\partial\delta}{\partial t} = -3H(t)\rho+\overline{\rho}\frac{\partial\delta}{\partial t} && (32)\\ \vec{\nabla}_r\cdot(\rho\vec{v}) &= \rho\vec{\nabla}_r\cdot\vec{v}+\vec{v}\cdot\vec{\nabla}_r[\overline{\rho}(1+\delta)] = 3H(t)\rho+\rho\vec{\nabla}_r\cdot\vec{u}+\overline{\rho}(\vec{v}\cdot\vec{\nabla}_r)(1+\delta) && (33)\end{aligned}$$

So it yields:

$$\frac{\partial\rho}{\partial t}+\vec{\nabla}_r(\rho\vec{v}) = \overline{\rho}\frac{\partial\delta}{\partial t}+\rho\vec{\nabla}_r\cdot\vec{u}+\overline{\rho}(\vec{v}\cdot\vec{\nabla}_r)(1+\delta) = 0 \tag{34}$$

From the partial derivatives in $(\vec{x},\tau)$ (equation 30), we get:

$$\frac{\partial\delta}{\partial t} = \frac{1}{a}\frac{\partial\delta}{\partial\tau}-H(\vec{x}\cdot\vec{\nabla}_x)\delta \tag{35}$$

and

$$(\vec{v}\cdot\vec{\nabla}_r)(1+\delta) = \vec{v}\cdot\vec{\nabla}_r\delta = H(\vec{x}\cdot\vec{\nabla}_x)\delta+\frac{1}{a}(\vec{u}\cdot\vec{\nabla}_x)\delta \tag{36}$$

So we finally have for the Continuity equation:

$$\frac{\partial\delta}{\partial\tau}+\vec{\nabla}_x[(1+\delta)\vec{u}] = 0 \tag{37}$$

For the Euler equation, taking equation 30:

$$\frac{\partial\vec{v}}{\partial t} = \frac{1}{a}\frac{\partial\vec{v}}{\partial\tau}-H(\vec{x}\cdot\vec{\nabla}_x)\vec{v} = \frac{\partial H}{\partial\tau}\vec{x}+\frac{1}{a}\frac{\partial\vec{u}}{\partial\tau}-aH^2\vec{x}-H(\vec{x}\cdot\vec{\nabla}_x)\vec{u}, \tag{38}$$

and the second term:

$$(\vec{v}\cdot\vec{\nabla}_r)\vec{v} = \frac{1}{a}\,(Ha\vec{x}+\vec{u})\cdot\vec{\nabla}_x(Ha\vec{x}+\vec{u}) = H^2\vec{x}a+H(\vec{x}\cdot\vec{\nabla}_x)\vec{u}+H\vec{u}+\frac{1}{a}(\vec{u}\cdot\vec{\nabla}_x)\vec{u} \tag{39}$$

Finally, we obtain:

$$\frac{\partial\vec{v}}{\partial t}+(\vec{v}\cdot\vec{\nabla}_r)\vec{v} = \frac{\partial H}{\partial\tau}\vec{x}+\frac{1}{a}\frac{\partial\vec{u}}{\partial\tau}+H\vec{u}+\frac{1}{a}(\vec{u}\cdot\vec{\nabla}_x)\vec{u} = -\frac{1}{a}\vec{\nabla}_x\Phi-\frac{1}{a}\frac{\vec{\nabla}_xP}{\rho}. \tag{40}$$

Therefore:

$$\frac{\partial\vec{u}}{\partial\tau}+\frac{H}{a}\vec{u}+(\vec{u}\cdot\nabla_x)\vec{u} = -\vec{\nabla}_x\Phi-\frac{\vec{\nabla}_xP}{\rho}. \tag{41}$$

### 2.2.1 Linear Perturbation Theory

In the lineal regime we assume that the perturbations are small enough: $|\delta| \ll 1$ and $\frac{\vec{\nabla}_x \vec{u}}{\dot{a}} \ll 1$.

Therefore, the result of linearizing the continuity equation is:

$$\frac{\partial \delta}{\partial t} + \frac{1}{a}\vec{\nabla}_x \vec{u} = 0 \tag{42}$$

and for the Euler equation:

$$\frac{\partial \vec{u}}{\partial \tau} + \frac{H}{a}\vec{u} = -\vec{\nabla}_x \Phi - \frac{\vec{\nabla}_x P}{\rho} \tag{43}$$

Writing the perturbed Euler equation with the expressions 27, 28 and 29:

$$\frac{\partial \vec{u}}{\partial \tau} + \frac{H}{a}\vec{u} = -\vec{\nabla}_x \phi - \frac{\vec{\nabla}_x \delta P}{\overline{\rho}(1+\delta)}. \tag{44}$$

As it is the linealized equation, with the assumption of $|\delta| \ll 1$ we can write $1 + \delta \approx 1$.

Taking the time derivative of equation 42, we obtain:

$$\frac{\partial^2 \delta}{\partial t^2} - \frac{1}{a} H \vec{\nabla}_x \vec{u} + \frac{1}{a}\vec{\nabla}_x \frac{\partial \vec{u}}{\partial t} = 0 \tag{45}$$

Combining the previous equation with equation 43 and the Poisson equation (equation 20):

$$\frac{\partial^2 \delta}{\partial t^2} - 2H\frac{\partial \delta}{\partial t} - 4\pi G\overline{\rho}\delta - \frac{1}{a^2\overline{\rho}}\vec{\nabla}_x^2 \delta P = 0, \tag{46}$$

Which is the fundamental equation for the growth of structures in Newtonian theory [9]. The second term is the Hubble drag term, which defines how expansion suppresses perturbation growth. The third term is gravitational, and expresses how gravity promotes perturbation growth. Finally, the last one is the pressure term, showing how the pressure gradient affects the perturbation growth.

Considering a bariotropic fluid, $P = P(\rho)$ [10],

$$\delta P = \frac{\partial P}{\partial \rho}\overline{\rho}\delta = c_s^2 \overline{\rho}\delta, \tag{47}$$

where $c_s$ is the sound speed.

Using this expression in equation 46 and Fourier expanding ($\vec{\nabla}^2 \to -k^2$):

$$\frac{\partial^2 \delta}{\partial t^2} + 2H\frac{\partial \delta}{\partial t} + \left(\frac{c_s^2 k^2}{a^2} - 4\pi G\overline{\rho}\right)\delta = 0 \tag{48}$$

- For $\frac{c_s^2 k^2}{a^2} > 4\pi G\overline{\rho}$: the pressure term domains and it gives rise to acoustic oscillations (sound waves) in the fluid.
- For $\frac{c_s^2 k^2}{a^2} < 4\pi G\overline{\rho}$: this system is unstable to gravitational accretion.

The Jeans length is defined as:

$$\lambda_J = c_s \sqrt{\frac{\pi}{G\overline{\rho}}} \tag{49}$$

Perturbations with a proper length exceeding the Jeans length are gravitational unstable, they will grow exponentially. On the other hand, on smaller scales than the Jeans scale, pressure supports oscillations as equation 46 takes the form of a harmonic oscillator.

For equation 48, we look for solutions of the type of:

$$\delta_{k,\tau} = D(\tau)A_k + C(\tau)B_k, \tag{50}$$

Where $D(\tau)$ is the growth factor and $C(\tau)$ the decaying one, but we will only consider the growing solution. A common expression for $D(\tau)$ can be given by the normalization of $D(z) = 1$ at $z = 0$. So the solution for a flat space is:

$$D(z) = \frac{H(z)}{H_0} \int_z^\infty \frac{dz'}{H^3(z')} \left[ \int_0^\infty \frac{dz'}{H^3(z')} \right]^{-1} \tag{51}$$

The density contrast at time $t$ can be calculated from the initial one as:

$$\delta(k,\tau) = \delta(k,\tau_i)\frac{D(\tau)}{D(\tau_i)} \tag{52}$$

Taking equation 42 we can write:

$$\vec{\nabla}_x \cdot \vec{u} = -\frac{\partial \delta}{\partial t} = -\dot{a}\frac{\partial \delta}{\partial a} = -H\frac{\partial \delta}{\partial \ln a} \tag{53}$$

we can define the *growth rate* as:

$$f_\Omega = \frac{1}{H}\frac{\dot{D}}{D} = \frac{d\ln D}{d\ln a}, \tag{54}$$

so equation 53 becomes:

$$\vec{\nabla}_x \vec{u} = -aHf_\Omega \delta \tag{55}$$

### 2.2.2 The Zel'Dovich Approximation

This is one of the various structure formation models beyond the linear theory, the linear order Lagrangian model. The Zel'dovich approximation remains one of our most powerful analytic models of the large scale structure because it is a simple approximation to describe the non-linear regime. For a generic triaxial perturbation, the collapse is expected to occur not to a point, but to a flattened structure of quasi-two-dimensional nature. This is usually defined as *pancake*. This approximation describes the fluid element's trajectory by the initial Lagrangian position $\vec{q}$, the comoving Eulerian position $\vec{x}(\vec{q},t)$ and the displacement field $\vec{\Psi}(\vec{q},t)$.

The mapping between the initial position in Lagrangian coordinates of each element of the fluid, $\vec{q}(0)$ and its final position in Eulerian coordinates, $\vec{x}(\vec{q},t)$ is expressed as [11]

$$\vec{x}(\vec{q},t) = \vec{q}(t_0) + \vec{\Psi}(\vec{q},t), \tag{56}$$

The perturbative solution of the Lagrangian Perturbation Theory (LPT) for the displacement field is:

$$\vec{\Psi}(\vec{q},t) = \vec{\Psi}(\vec{q},t)^{(1)} + \vec{\Psi}(\vec{q},t)^{(2)} + ... + \vec{\Psi}(\vec{q},t)^{(N)} \tag{57}$$

The first order solution is the Zel'dovich approximation, so it assumes that the series can be truncated at linear order: $\vec{\Psi}(\vec{q},t)^{(1)} \simeq D(t)\vec{\nabla}\vec{\Phi}(\vec{q})$.

The conservation of mass let us express:

$$\rho(\vec{x},t)d\vec{x} = \rho(\vec{q})d\vec{q}. \tag{58}$$

We can neglect the density fluctuations in Lagrangian space, obtaining:

$$1 + \delta(\vec{x},t) = J^{-1}(\vec{q},t), \tag{59}$$

$$\left|\frac{\partial\vec{x}}{\partial\vec{q}}\right| = J^{-1}, \tag{60}$$

where $J$ is the Jacobian of the coordinate transformation. The solution of equation 59 in terms of the eigenvalues $\lambda_{1,2,3}$ of the Jacobian is:

$$1 + \delta(\vec{q},t) = \frac{1}{(1-D(t)\lambda_1(\vec{q}))(1-D(t)\lambda_2(\vec{q}))(1-D(t)\lambda_3(\vec{q}))}, \tag{61}$$

with $\lambda_1 > \lambda_2 > \lambda_3$.

The Zel'dovich approximation predicts that the density in certain regions, called *caustics*, should become infinite, whereas the gravitational acceleration in these regions remains finite. We cannot justify ignoring pressure when the density is very high because shock waves form and compress infalling material. At a certain point, the process of accretion onto the caustic stops. The condensed matter is contained by gravity within the final structure, while the matter which has not passed through the shock wave is help up by pressure. It has been calculated that approximately half of the material inside the original fluctuation is reheated and compressed by the shock wave. Structures are strongly unstable to fragmentation, so one can generate structure on smaller scales than the pancake [12].

### 2.2.3 The Lognormal model

As a structure formation model, the code `ARGO` (used for the development of chapter 4) assumes the *Lognormal model* [13], derived from the continuity equation (equation 13). If we write this expression in terms of equation 32 and the convective derivative:

$$\frac{d}{d\tau} = \frac{\partial}{\partial\tau} + \vec{u}\cdot\vec{\nabla} \tag{62}$$

which can be obtain from equation 23 and 32, we have:

$$\frac{1}{\rho}\frac{d\rho}{d\tau} = -\vec{\nabla}\cdot\vec{u} \tag{63}$$

$$\int\frac{1}{\rho} = -\int d\tau\vec{\nabla}\cdot\vec{u} \tag{64}$$

This equations gives us a lognormal solution for the density contrast if the divergence of the velocity is Gaussian distributed.

The linear Poisson equation,

$$\delta_L = D\vec{\nabla}^2\Phi, \tag{65}$$

and substituting the linear displacement field from the Zel'dovich approximation:

$$\delta_L = -\vec{\nabla}\cdot\vec{\Psi}. \tag{66}$$

The velocity in the comoving frame $\vec{x}$ can be expressed through the displacement field as

$$\vec{u} = \frac{d\vec{x}}{d\tau} = \frac{d\vec{\Psi}}{d\tau} \tag{67}$$

and:

$$\vec{\nabla}\cdot\vec{u} = \vec{\nabla}\cdot\frac{d\vec{\Psi}}{d\tau}. \tag{68}$$

With equations 64 and 68 we can obtain:

$$\vec{\nabla}\cdot\vec{\Psi} = -\log(1+\delta) + C \tag{69}$$

where $C$ is an integration constant.

Therefore, taking equation 66 we have:

$$\log(1+\delta) = \delta_L + C \tag{70}$$

## 2.3 The Power Spectrum

A generic perturbation can be represented as a superposition of plane waves, by the Fourier representation theorem, and while they are evolving linearly, they are independent of each other.

$$\delta(\vec{x}) = \sum_k \delta(\vec{k})e^{i\vec{k}\cdot\vec{x}} \tag{71}$$

In the continuous limit, as we saw in equation 13, we have:

$$\delta(\vec{x}) = \frac{1}{(2\pi)^3}\int \delta(\vec{k})e^{i\vec{k}\cdot\vec{r}}d^3\vec{k} \tag{72}$$

$$\delta(\vec{k}) = \frac{1}{(2\pi)^3}\int \delta(\vec{x})e^{-i\vec{k}\cdot\vec{r}}d^3\vec{x} \tag{73}$$

Where $k$ is the wave vector of a particular plan wave, also known as *mode*.

The power spectrum is a measurement of the amplitude of fluctuations as a function of distance scales in Fourier space, so it is defined as

$$P(\vec{k}) = \langle|\ \delta(\vec{k})\ |^2\rangle. \tag{74}$$

Therefore, it has no information about phases.

Using equation 16 and 74 we obtain:

$$\epsilon(r) = \langle \delta(\vec{r})\delta(\vec{x}+\vec{r}) \rangle = \frac{L^3}{2\pi^3} \int P(\vec{k}) e^{-i\vec{k}\vec{x}} d^3\vec{k} \tag{75}$$

This is known as the Wiener-Khinchin theorem, which let us express the power spectrum as the Fourier transform of the correlation function.

Using the condition of isotropy and expanding $d^3\vec{k} = k^2 \sin\theta dk d\theta d\Phi$, we have that the power spectrum only depends on the module at $k$, and therefore:

$$\epsilon(r) = \frac{L^3}{2\pi^2} \int P(k) k^2 \frac{\sin(kr)}{kr} dk \tag{76}$$

The function $\frac{\sin(kr)}{kr}$ acts as a *window function.* We are not interested in integrate over all scales, the study of individual galaxies is not consider when we are doing large scale structure simulations. So the smallest scale of cosmological interest is that of a typical separation between neighboring galaxies of the order or 1 Mpc.

We can exclude scales smaller than a certain $R$ ($r < R$ *or* $k > R^{-1}$), it is possible to filter the density field with a *window function.* In x-space this filtering is done by convolution. We introduce a window function, which is usually spherically symmetric [7]. The commonly used is the *top-hat window function*:

$$W(k) = \frac{3(\sin(kr) - kr\cos(kr))}{(kr)^2} \tag{77}$$

So the resulting density will be smoothed according to: $\delta_{smooth}(k) = \delta(k)W(k)$. And therefore, the power spectrum becomes: $P_{smooth}(k) = P(k)W^2(k)$.

The initial power spectrum from inflation is described in terms of the linear growth factor, $D(t)$:

$$P(k) = D^2(t)P_0(k), \tag{78}$$

Observations suggested that the initial power spectrum must have been very broad with no preferred scales so it is natural to begin with a power-law form:

$$P_0(k) = Ak^{n_s} \tag{79}$$

where $A$ is a normalization constant and it is a free parameter, and $n_s$ is the spectral index, which describes how density fluctuations change with the scale $k$. For the Harrison-Zel'dovich power spectrum $n = 1$.

It is also common to define the dimensionless power spectrum as

$$\triangle^2(k) = \frac{1}{2\pi^2} k^3 P(k), \tag{80}$$

which represents the contribution to the variance per unit logarithmic interval in $k$.

The transfer function, $T^2(k)$, describes how the shape of initial power spectrum of the dark matter is modified due to radiation dominated era of the Universe. We can relate the power spectrum and the growth factor through the transfer factor as it follows [14]:

$$P(k,t) \propto k^n T^2(k) D^2(t). \tag{81}$$

So taking equation 78, the previous expression can also be written as

$$P_0 = k^n T^2(k) \tag{82}$$

## 2.4 Bias

Galaxy surveys do not measure the matter density field itself, but the distribution of galaxies or other tracer, that is, highly non-linear objects which are the result of a complex formation process. The bias describes the relation between the distribution of these tracers and that of matter, and it is a result of the varied physics of galaxy formation, which can cause the spatial distribution of baryons to differ from that of dark matter.

In a simple Gaussian model it is shown that the 2-point correlation functions in the underlying dark matter and the galaxies are related by [2]

$$\epsilon_g(r) = b^2 \epsilon_{dm}(r), \tag{83}$$

where $b$ is the bias factor. From the relation between the 2-point correlation function and the density contrast

$$\left(\frac{\delta\rho}{\rho}\right)_g = b\left(\frac{\delta\rho}{\rho}\right)_{dm} \tag{84}$$

or, what is the same,

$$\delta_g = b\ \delta_{dm} \tag{85}$$

# 3 Bayesian Statistics

In this chapter we will introduce the Bayesian Statistics and the Hamiltonian Monte Carlo Markov Chain, which is the algorithm implemented in `ARGO` and `BIRTH` code, used for this project.

The frequentist statistics tests whether an event occurs or not. It calculates the probability of an event in the long run of the experiment, i.e, how frequently something happens in an infinite number of trials, so it is based on many repetitions of the experiment. On the other hand, the Bayesian framework needs a prior probability distribution which embodies, before seeing any data, how plausible it is that the parameters could have values in the different regions of parameter space. So it works with degrees of belief or credences.

The Bayesian approach reduces statistical inference to probabilistic inference by defining a joint distribution for both parameters and the observable data.

When we combine a prior distribution for the parameters with the conditional distribution for the observed data, we get a joint distribution for all quantities related to the problem. We can derive the Bayes' rule for the posterior distribution of the parameters as it follows [15]:

$$\mathscr{P}(\theta|x_1,...,x_c) = \frac{\pi(\theta)\mathscr{L}(x_1,...,x_n|\theta)}{p(x_1,...,x_n)}, \tag{86}$$

where $\theta$ is the parameter vector and $x_1,...,x_n$ is the data vector.

- $\mathscr{P}(\theta|x_1,...,x_n)$ is the posterior function, the conditional probability of the parameter vector $\theta$ given the data vector $x$;
- $\pi(\theta)$ is the prior probability distribution of the parameters $\theta$, and is already known without knowledge of any data;
- $\mathscr{L}(x_1,...,x_n|\theta)$ is the likelihood and it is the probability of the data $x$ given the parameter vector $\theta$, or the model of the data;
- $p(x_1,...,x_n)$ is the evidence and it is expressed as

  $$\int \pi(\theta|x_1,...,x_n)\mathscr{L}(x_1,...,x_n|\theta)d\theta. \tag{87}$$

  So it is the distribution of the observed data marginalized over the parameters, but it is only important to take into account for model comparison, otherwise, we can express equation 86 as a proportionality of the likelihood:

$$\mathscr{P}(\theta|x_1,...,x_c) \propto \pi(\theta)\mathscr{L}(x_1,...,x_x|\theta) \tag{88}$$

## 3.1 Hamiltonian Monte Carlo Markov Chain

In this section we follow [1] to do the derivation of the Hamiltonian Monte Carlo algorithm. To sample the posterior we use the Hamiltonian dynamics. It operates on a n-dimensional position vector, $q_i$ and a n-dimensional momenta vector, $p_i$, for $i = 1, ..., n$. So the full state space has 2n-dimensions. The combination of position an momenta variables is known as phase space, and the total energy function for point in phase space is the Hamiltonian:

$$\mathscr{H}(q,p) = U(q) + K(p) = U(q) + \frac{1}{2}\sum_i \frac{p_i^2}{m_i} \tag{89}$$

The kinetic energy is usually defined as

$$K(p) = \frac{1}{2}p^T M^{-1} p, \tag{90}$$

where $M$ is the mass matrix, symmetric, positive-defined and typically diagonal.

To relate the Hamiltonian dynamics with a probabilistic measure we resort to the canonical distribution definition:

$$\mathscr{P}(q,p) = \frac{1}{Z_K} e^{\frac{-H(q,p)}{K_b T}}. \tag{91}$$

$Z$ is the normalization of the distribution function.

Equation 91 can also be expressed as

$$\mathscr{P}(q,p) = \mathscr{P}(q)\mathscr{P}(p) = \frac{1}{Z} e^{\frac{-U(q)}{K_b T}} e^{\frac{-K(p)}{K_b T}}. \tag{92}$$

$U(q)$ and $K(p)$ are factorizing into two separated probabilities $\mathscr{P}(q)$ and $\mathscr{P}(p)$. The Hamiltonian Monte Carlo method defines the positions, $q$, as the variable to sample, in our case, the primordial fluctuations $\delta(x)$; and the momenta, $p$, are artificially introduced in the kinetic term just to allow us to explore the phase-space. Therefore, the momenta are introduced to evolve the system and get $q$. The marginalization is done to avoid the dependence of momenta when obtaining the posterior. In `ARGO` code this is done by introducing in each iteration new values for momenta. We can write equation 91 as

$$U(q) = -\ln \mathscr{P}(q) \tag{93}$$

Here, we can easily see that the posterior, $\mathscr{P}(q)$, can be obtained from the term $e^{-U(q)}$. We have set $k_b T = 1$ and the constant will vanish due to the HMC method.

However, as we are interested in evolving the system, what we are going to sample is:

$$e^{-H} = e^{-K} e^{-U}. \tag{94}$$

If we take the exponential of equation 90, we get

$$e^{-K} = e^{-\frac{1}{2}p^T M^{-1} p}, \tag{95}$$

and it is the expression of a multivariate Gaussian distribution with $M$ as the covariance matrix of the momenta. The second term of equation 94 is just the posterior, as it is expressed in equation 93.

The partial derivatives of the Hamiltonian determine how $q$ and $p$ change with time, $t$, according to the Hamilton's equations:

$$\frac{dq_i}{dt} = \frac{\partial \mathscr{H}}{\partial p_i} \tag{96}$$

$$\frac{dp_i}{dt} = -\frac{\partial \mathscr{H}}{\partial q_i} \tag{97}$$

Substituting equation 89 and 90, the Hamilton's equations can then be written as

$$\frac{dq_i}{dt} = M^{-1} p_i \tag{98}$$

$$\frac{dp_i}{dt} = -\frac{\partial U}{\partial q_i} \tag{99}$$

Moreover, the Hamiltonian dynamics has to fulfill some properties explained below:

- H is conserved as $q_i$ and $p_i$ evolve through time: $\frac{dH}{dt} = 0$.
- The dynamics also preserves the volumes of regions of phase space: Liouville's theorem.
- Hamiltonian dynamics is reversible: the mapping form the state $t, (q(t), p(t))$ to the next state $t+s, (q(t+s), p(t+s))$ is one-to-one, and therefore, the inverse mapping is obtained by negating the times derivatives in equations 96 and 97.

Together, these properties imply that the canonical distribution is invariant with respect to any transformation.

However, to evolve the system and get the positions, $q$, we must discretize the Hamilton's equations using some non-zero time step, and introducing, thus, an inevitable error. The Leapfrog discretization is the commonly used scheme, through which we can obtain better results than with other methods. It preserves space volume and is also time reversible. A single iteration calculates approximations to the position and momenta at time $\tau + \epsilon$ from this quantities at $\tau$ as it follows:

$$p_i\left(\tau + \frac{\epsilon}{2}\right) = p_i(\tau) - \frac{\epsilon}{2}\frac{\partial U}{\partial q_i}(q(\tau)) \tag{100}$$

$$q_i(\tau + \epsilon) = q_i(\tau) + \epsilon \frac{p_i\left(\tau + \frac{\epsilon}{2}\right)}{m_i} \tag{101}$$

$$p_i(\tau + \epsilon) = p_i\left(\tau + \frac{\epsilon}{2}\right) - \frac{\epsilon}{2}\frac{\partial U}{\partial q_i}(q(\tau + \epsilon)) \tag{102}$$

So we have a half step for the momenta, then a full step for the positions and, finally, the other half step for the momenta.

As we have seen, we can substitute the potential energy for $-\ln \mathscr{P}(q) = -\ln \pi - \ln \mathscr{L}$, following the Bayes' rule.

Markov chain based on stochastic dynamics will sample from the correct distribution only if in the limit as the step size used in discretizing the dynamics goes to zero. The bias introduced by using a non-zero step size, mentioned before, is eliminated in this method.

The Hamiltonian Monte Carlo samples points in phase space by means of Markov Chain in which stochastic and dynamical transitions alternate. Typically, the momenta are replaced with new values via Gibbs sampling. The dynamical transitions are similar that the stochastic dynamics ones, but with two differences: a random decision is made for each transition whether to simulate the dynamics forward or backward in time. And the point reached by following the dynamics is only a candidate for the new state, to be accepted or rejected based on the change in the total energy, similar to the Metropolis algorithm.

Given values for the magnitude of the Leapfrog stepsize, $\epsilon_0$, and the number of Leapfrog steps, $L$, the dynamical transitions of the Hamiltonian Monte Carlo algorithm operate as follow:

1) Randomly choose a direction, $\lambda$, for the trajectory, with $\lambda = +1$, representing forward trajectory, and $\lambda = -1$, representing a backward trajectory, both equally likely.

2) Starting from the current state, $(q, p) = (q(0), p(0))$, perform $L$ leapfrog steps with a stepsize of $\epsilon = \lambda\epsilon_0$, resulting in the state $(q(\epsilon L), p(\epsilon L)) = (q*, p*)$.

3) Regard $(q*, p*)$ as a candidate for the next state, as in the Metropolis algorithm, accepting it with specific probability, otherwise, letting the new state be the same as the old one. The proposed state is accepted as the next state of the Markov chain with probability:

$$\min\left[1, e^{(-H(q*,p*)+H(q,p))}\right] \tag{103}$$

### 3.1.1 The Prior

We have seen that the prior is the probability distribution of the parameters, which in this case corresponds with the underlying structure formation model. As we are working with the linearized density, $\delta_L = \log(1+\delta) - C$, we assume Gaussian statistics [16]:

$$\pi(\delta_L \mid C_L) = \frac{1}{\sqrt{(2\pi)^{N_c} det(C_L)}} \exp\left(-\frac{1}{2}\delta_L^{+} C_L^{-1} \delta_L\right). \tag{104}$$

It is a multivariate Gaussian with zero mean and given covariance. $C_L$ is the convariance matrix: $C_L = \langle \delta_L^{+} \delta_L \rangle$, and is diagonal due to the absence of coupling modes.

### 3.1.2 The Likelihood

The likelihood defines the model of the data. In this case, it is the probability to draw a certain count of galaxies per cell, $N_k^g$, given the expectation value of galaxy counts, $\lambda_k$, for this particular cell. As galaxies are discrete particles, the galaxy distribution can be described as a specific realization drawn from an inhomogeneous Poisson process, therefore, we can define for the likelihood [16] [11]:

$$\mathscr{L}(N_k^g | \lambda_k) = \prod_k \frac{(\lambda_k)^{N_k^g} e^{-\lambda_k}}{N_k^g!} \tag{105}$$

where,

$$\lambda_k = f_N w(1+\delta)^b \tag{106}$$

$f_N$ is the normalization of the expectation value, and $b$ is the power law bias parameter, seen in section 2.4. If we assume linear bias, $b = 1$.

### 3.1.3 The Posterior

As it is previously defined, to obtain the posterior from the potential energy we need to compute the negative logarithm of the posterior:

$$-\ln \mathscr{P} = -\ln \pi - \ln \mathscr{L} \tag{107}$$

From equation 104 we can obtain the negative logarithm of the prior as:

$$-\ln \pi(\delta_L \mid C_L) = \frac{1}{2}\delta_L^{+} C_L^{-1} \delta_L + c \tag{108}$$

where we have included all constants in the term $c$.

The negative logarithm of the likelihood, taking equation 105, is:

$$-\ln \mathscr{L}(N_k^g|\lambda_k) = \sum_k \lambda_k \ln \lambda_k - c. \tag{109}$$

However, to evolve the system, we use the equation of motion and, therefore, we need to calculate the gradient of equations 108 and 109 according to equation 99.

For equation 108 it yields:

$$-\frac{\partial \ln \pi}{\partial \delta_L} = C_L^{-1}\delta_L \tag{110}$$

For the likelihood we use the chain rule as:

$$\frac{\partial}{\partial \delta_L} = \frac{\partial}{\partial \lambda_k}\frac{\partial \lambda_k}{\partial \delta_i}\frac{\partial \delta_i}{\delta_{L,j}} \tag{111}$$

So we get:

$$-\frac{\partial \ln \mathscr{L}}{\partial \delta_{L,i}} = b\lambda_i\left(1 - \frac{N_i}{\lambda_i}\right) \tag{112}$$

# 4 Implementation of High order Leapfrog Algorithm

To increase the computational efficiency of the Hamiltonian Monte Carlo sampling, we are going to generalize the discretization of the equations of motion using the Leapfrog algorithm for higher orders. The normal Leapfrog equations defined before have a transformation of the form:

$$T_\epsilon = T_p(\epsilon/2)T_q(\epsilon)T_p(\epsilon/2) \tag{113}$$

which means a half step in the momenta, a full step for the positions, and another half step for the momenta. This algorithm corresponds to a second order discretization of equations of motion ($O(\delta^2)$).

Now, let us assume a transformation $T_n(\epsilon)$, which is a reversible, area-preserving discretization of the Hamilton's equations. Setting an arbitrary integer $i$, we take $i$ steps of size $\epsilon$ forward with this transformation, and then, one step backward with a size of $s\epsilon = (2i)^{1/(n+1)}\epsilon$. Finally, additional $i$ steps of size $\epsilon$ forward. This transformation can be represented as [17]:

$$T_{n+2}((2i-s)\epsilon) = T_n(\epsilon)^i T_n(-s\epsilon)T_n(\epsilon)^i \tag{114}$$

It will give an evolution accurate to order $n+2$. So iterating this scheme recursively produces a discretization of equations of motion to any desired order. We are going to focus on the fourth order. If we take $n = 4$, the term $T_2(\epsilon)$ is exactly the original Leapfrog algorithm. So equation 114 express that it appeals to Leapfrog equations $i$ times, then to the same equations but with the backward step, and, finally, it appeals again to the Leapfrog equations $i$ times.

For this project we are going to analyze the number of forward steps $i = 1, 2$ and 3.

## 4.1 Study of the number of forward steps, $i$, and the stepsize

In this section, we rely on the `ARGO` code [18], in its latest version [19], to sample the density field. In particular, it uses a lognormal prior model for the density field and a Poisson likelihood for the galaxy distribution, as introduced in [20]. The posterior distribution function is sampled with the Hamiltonian sampling technique following [21], including an automatic estimation of the logarithmic mean field [22]. In each iteration, the code generates two outputs: the reconstructed Gaussian field, without considering the displacements in the density peaks; and its power spectrum.

It is very important to choose a suitable value for the stepsize to sample the parameter space efficiently. For the code with the second order Leapfrog algorithm it was set to $\epsilon = 0.06$, but now, as the fourth order method has been implemented, a new study for this parameter is required. A too large stepsize will result in a very low acceptance rate for the new states proposed, and a too small stepsize can waste computation time or will lead to a slow exploration of the parameter space.

To study this, the code has been run with a lower resolution, $128^3$ cells, for different values of $i$ and for different values of stepsize in each one, to analyze the convergence, the computation time and the acceptance rate. This have been performed using the *Diva Severo Ochoa* machine, which is a High Performance Computer at the IAC with the following specifications [23]:

Table 1: Diva characteristics

| Use | Host | CPU | Freq | Cores | RAM | Disk |
|---|---|---|---|---|---|---|
| Login & GPU node | deimos | 2x Intel Xenon E5-2630 v4 | 2.20 GHz | 20 | 1TB | 11 TB |
| Computing node | diva | 12x Intel Xenon E5-2630 v4 | 2.10 GHz | 192 | 4.5 TB | 40 TB |

Firstly, a study of the optimal number of cores to run the code has been done. For this, the code has been run for the same parameters ($i$, stepsize, seed and iterations) for different number of cores: $1, 2, 4, 8, 16, 32$ and 64. The figure 2 shows the computation time needed to reach 100 iterations as a function of the number of cores, represented in the blue line. The purple line is the reference one from a perfect scaling of the computation time with the number of cores, and it is what ideally would happen.

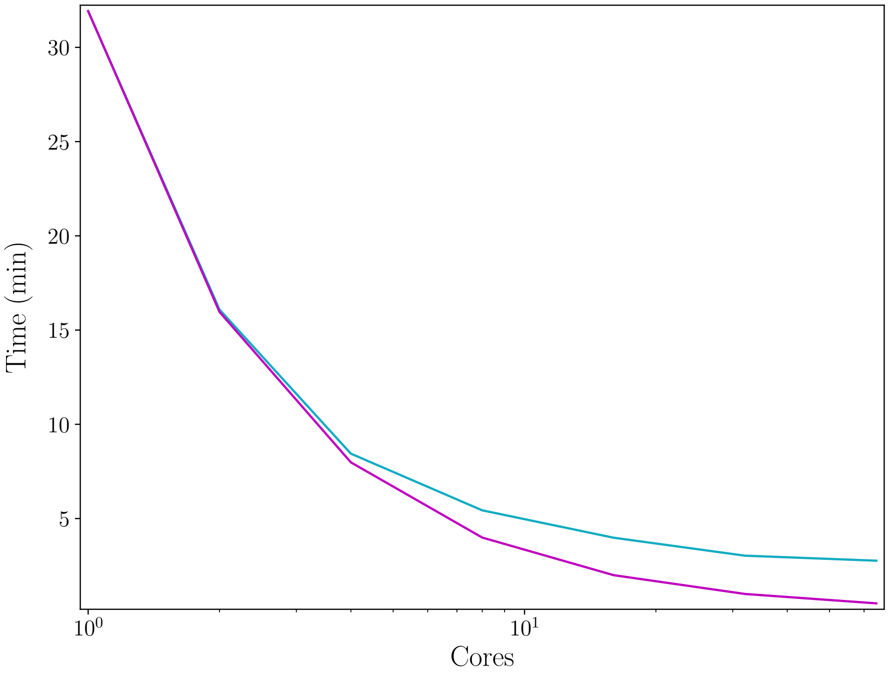


Figure 2: The blue line represents the computation time as a function of the number of cores. The purple one is a reference curve, which expresses the computation time for 1 core and this one divided by 2 each time we increase the number of cores (what ideally would happen).

We can see that, for more than 8 cores, the computation time decreases slowly until it becomes almost constant for more than 32 cores. So the computation time saved using $16, 32$ or 64 cores is not remarkable enough compared to using 8, because it deviates from that expected (purple line) and that is why this last number has been set to run all the codes with $128^3$ cells in this study.

In the following table it is shown, for each value of $i$, the iteration in which the chain converges, the computation time required to reach this convergence and the acceptance; for different values of stepsize. This last parameter expresses the percentage of iterations that have been accepted at the first time. This has been done by statistics of 5 different seeds, so each value of table 2 is a mean of the 5 chains, with the same value of $i$ and stepsize, but a different seed.

Table 2: Results with the fourth order Leapfrog algorithm through ARGO code.

| Stepsize | Iteration of convergence | Convergence time (min) | Acceptance |
|---|---|---|---|
| i=1 | | | |
| $\epsilon$ | 650 | $108,29$ | $95,0\%$ |
| $2\epsilon$ | 250 | $45,53$ | $70,2\%$ |
| $4\epsilon$ | 230 | $73,47$ | $35,2\%$ |
| $6\epsilon$ | 250 | $110,21$ | $23,8\%$ |
| $8\epsilon$ | 260 | $148,32$ | $13,8\%$ |
| $10\epsilon$ | 230 | $189,09$ | $12,6\%$ |
| i=2 | | | |
| $\epsilon$ | 68 | $18,95$ | $94,6\%$ |
| $2\epsilon$ | 53 | $23,85$ | $64,2\%$ |
| $4\epsilon$ | 46 | $36,05$ | $36,2\%$ |
| $6\epsilon$ | 36 | $50,06$ | $23,4\%$ |
| $8\epsilon$ | 46 | $73,40$ | $18,0\%$ |
| $10\epsilon$ | 40 | $106,48$ | $16,8\%$ |
| i=3 | | | |
| $\epsilon$ | 32 | $19,05$ | $88,4\%$ |
| $2\epsilon$ | 29 | $25,54$ | $51,6\%$ |
| $4\epsilon$ | 20 | $25,97$ | $27,4\%$ |
| $6\epsilon$ | 24 | $43,54$ | $17,0\%$ |
| $8\epsilon$ | 25 | $67,26$ | $13,8\%$ |
| $10\epsilon$ | 27 | $67,27$ | $26,0\%$ |

We can see in table 2 that, for the three values of $i$, if we set a stepsize $\epsilon$, the Markov chain reaches the convergence in a higher iteration that for the other values of stepsize. On the other hand, it is the configuration with the highest percentage of acceptance. For the case of $i = 1$, table 2 shows that the optimal configuration is that one with a stepsize $2\epsilon$, where the convergence is reached at iteration 250, and we still have a good acceptance value, what makes the convergence time to be also lower. However, as we increase the value of the stepsize, we can observe that the convergence is reached at a similar number of iterations that for $2\epsilon$, but the computation time increases due to the number of rejected iterations (we can see how the acceptance percentage decreases as we increase the value of stepsize).

For $i = 2$ we can also see that, for a stepsize of $\epsilon$, we get the highest convergence in number of itertions. However, in this case, the difference between the iteration of convergence in the different values of stepsize is not so big, and due to the fact that this configuration accepts the $94,6\%$ of the iterations at the first time, the computational time is the lowest one, and this value of stepsize becomes in the optimal for $i = 2$.

Finally, for the case of $i = 3$, we can see that the optimal configuration is that one with a stepsize $\epsilon$ too, despite the case of $4\epsilon$ has the lowest number of iterations to converge. This happens because it requires less time to reach this convergence due to, once again, the high acceptance rate.

To obtain the iteration of convergence we can compare the power spectrum of a specific iteration with the power spectrum of a converged Markov chain. This one has been obtained from the original code (with the second order Leapfrog algorithm) in a high iteration, 6000, to be sure that it has converged. In the figure 3 we can see how each chain for a different value of stepsize is reaching the convergence.

In the first figure on the left, corresponding to the case of $i = 1$, it is represented the power spectrum of all values of stepsize at iteration 50. We can see that none of these configurations have reached the convergence yet, and for a stepsize $2\epsilon$ causes to be faster (according to table 2). The middle figure shows the results for $i = 2$ at iteration 10. We can observe that, for the configuration of $10\epsilon$, the convergence is almost reached, and table 2 shows that this value of stepsize is the second faster, after $6\epsilon$; however, as this is only for a seed, the results can vary when we do statistics for different ones. On the other hand, for this case, the configuration of $2\epsilon$ is the slowest to converge. In the last figure, it is represented the case of $i = 3$, also for 10 iterations, where we can see that the power spectrum of a stepsize $4\epsilon$ is on top of reference one, and, therefore, it has reached the convergence. In table 2 we could see that with this value of stepsize we obtain the lowest number of iterations to converge, although it is not the best choice in terms of computational time.

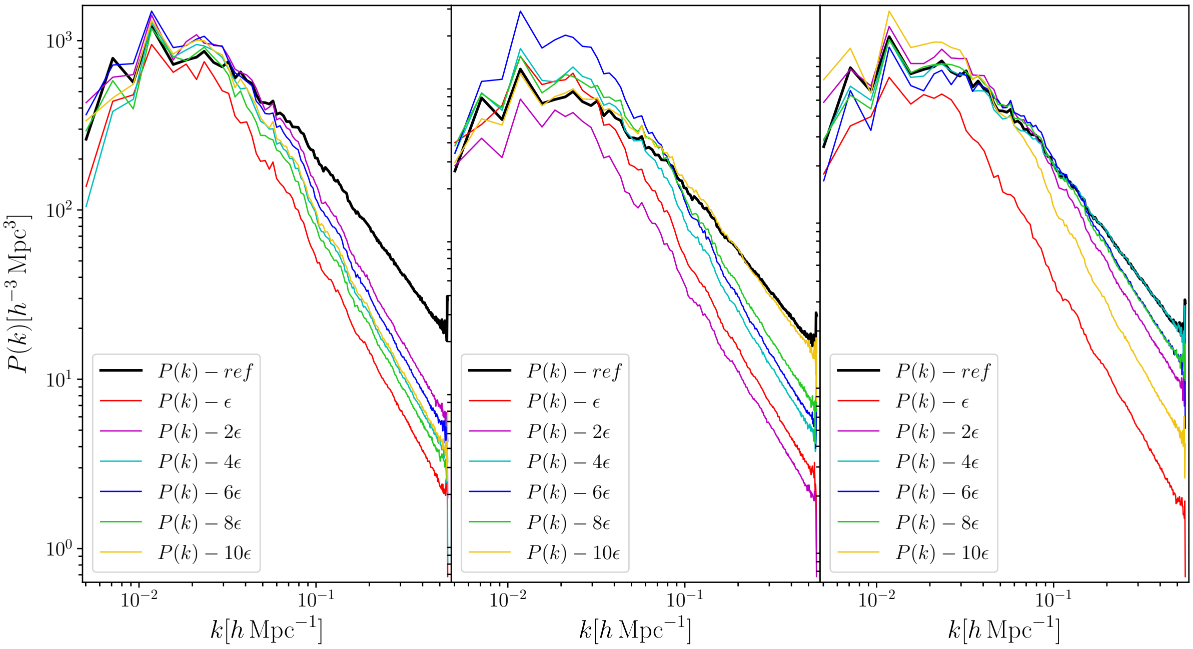


Figure 3: Representation of the power spectrum of different values of stepsize reaching the convergence, comparing them with a converged reference power spectrum (black curve). On the left we have the case $i = 1$ at an iteration 50. In the middle it is represented the case $i = 2$ at iteration 10, and on the right we have $i = 3$, also for iteration 10.

We can observe that, for small values of $k$ (which means large scales), the converged power spectrum in the figure on the right, is not exactly on top of the reference one. This is produced because at large scales we have less number of modes, $k$, and the statistics is not so good compared to smaller scales, where we have more of them. This uncertainty at large scale is known as *cosmic variance*.

In table 2 we could see that if we increase the value of $i$, the acceptance decreases faster when we go from a stepsize value to a higher one; what means that the computation time is each time higher. This can be seen in the figure 4, where on the left, the computation time required to get 100 iterations for each value of stepsize and for each value of $i$ is shown. In the figure of the right, the convergence time is represented, which is the time required for the different values of stepsize to reach convergence.

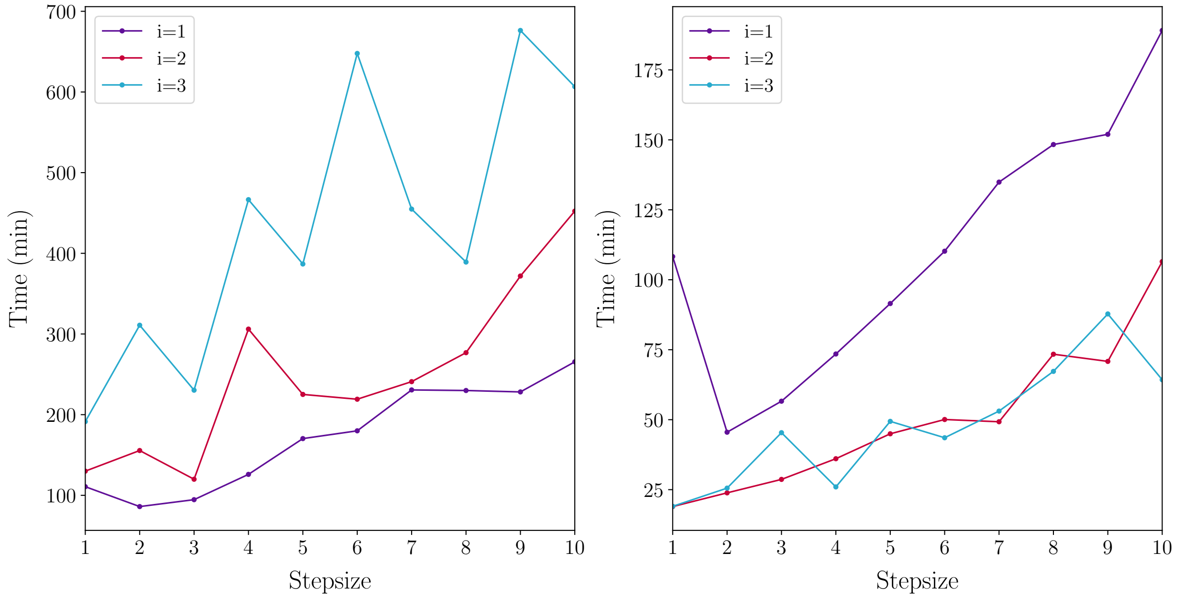


Figure 4: On the left, the computation time for 100 iterations as a function of the stepsize, for the different values of $i$. On the right, the convergence time as a function of the stepsize, also for the different values of $i$.

The figure 4 shows that, for a stepsize of $\epsilon$, which was the value set in the original code ($\epsilon = 0.06$), the computation time required to do a specific number of iterations (100 in this case) is the lowest one; however, this does not compensate the fact that the convergence was reached after 600 iterations for $i = 1$, what makes the stepsize $2\epsilon$ the optimal configuration, as we can see in the figure of the right. However, for $i = 2$ and 3, the stepsize of $\epsilon$ is still an optimal value, being the best choice for both cases and with a very similar convergence time.

Now, a representation of the acceptance rate for each value of stepsize is shown, for the case $i = 1$. We can easily see in figure 5 that, as we increase the value of stepsize, more iterations are accepted not at the first time, but at higher number of trials. For $\epsilon$, the $97,0\%$ of iterations are accepted at the first trial, and the $3,0\%$ at the second one, so they have been rejected once. For $2\epsilon$ we can see that there is a small percentage of iterations that are accepted at the second and third time. If we have a look at the last histogram, for a stepsize value of $10\epsilon$, iterations can be rejected until 50 times before being accepted, and this is what makes computation time increase.

The figure 6 shows, now, the acceptance for a stepsize of $\epsilon$, for the 3 values of $i$. We can see that for $i = 1$, that corresponds to the figure on the right, almost all iterations are accepted at the first time. For $i = 2$ we can observe that there is a very high percentage of iterations also accepted at first, but there are more iterations accepted at the second time than in the previous case. Finally, for $i = 3$, the figure on the right shows how rejected iterations increase, being accepted, some of them, at the eighth time.

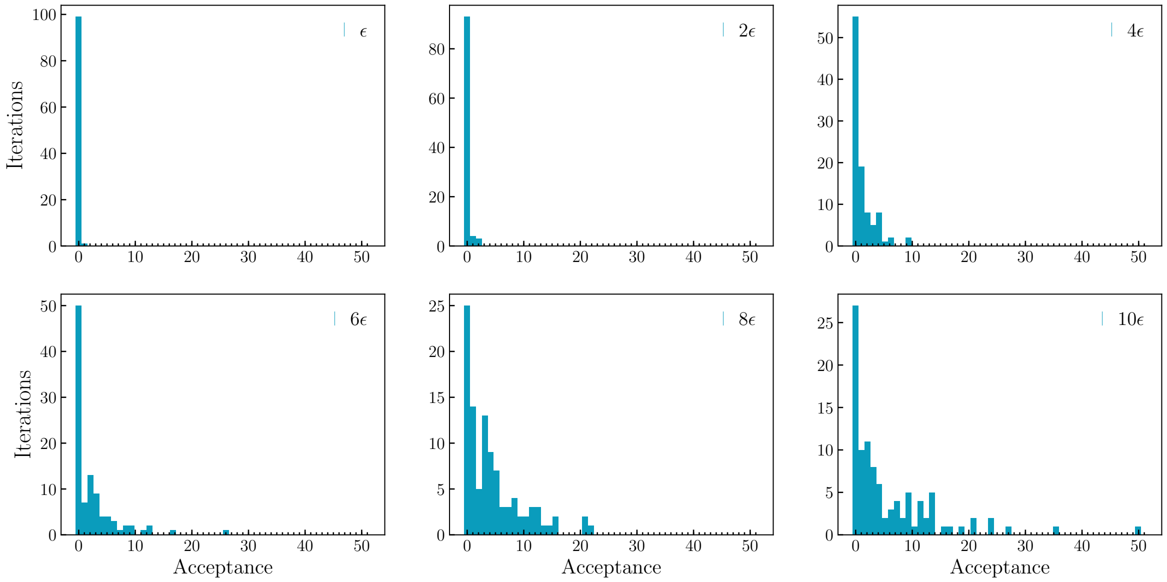


Figure 5: Acceptance for $i = 1$ and the different values of stepsize.

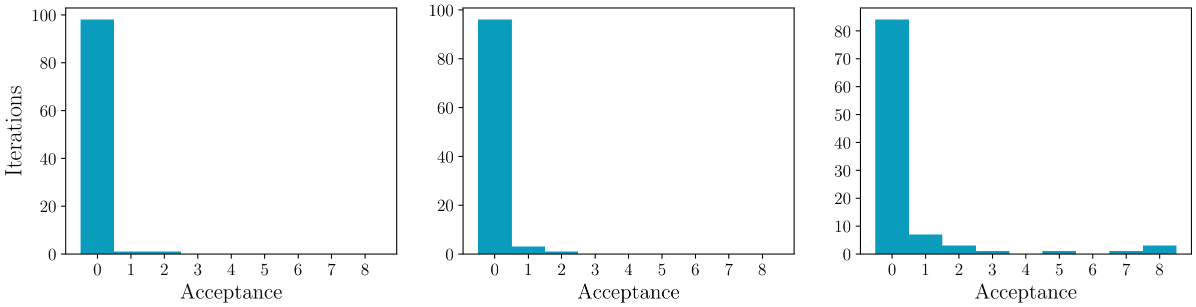


Figure 6: Acceptance for a stepsize of $\epsilon$ and the different values of $i$, from the left to the right: $i = 1, 2$ and 3.

## 4.2 Convergence of the Method

In this section, we are going to study the convergence running the code for higher resolution, taking $256^3$ cells. In the following table 3 we can see the results for the configurations $2\epsilon$, for $i = 1$ and $\epsilon$ for $i = 2$ and $i = 3$; which were the more efficient ones for each value of $i$. Statistics for different seeds has also been done.

As we have increased the number of cells, we are going to repeat again the study of the optimal number of cores for this section. In the figure 7 it is shown the speed up factor, which is the largest time of all the runs (the one for 1 core) divided by the time of each run; as a function of the number of cores. This is represented by the blue line. The purple one shows the reference curve, for an ideal speed up factor: 1 for 1 core, 2 for 2 cores, and so on.

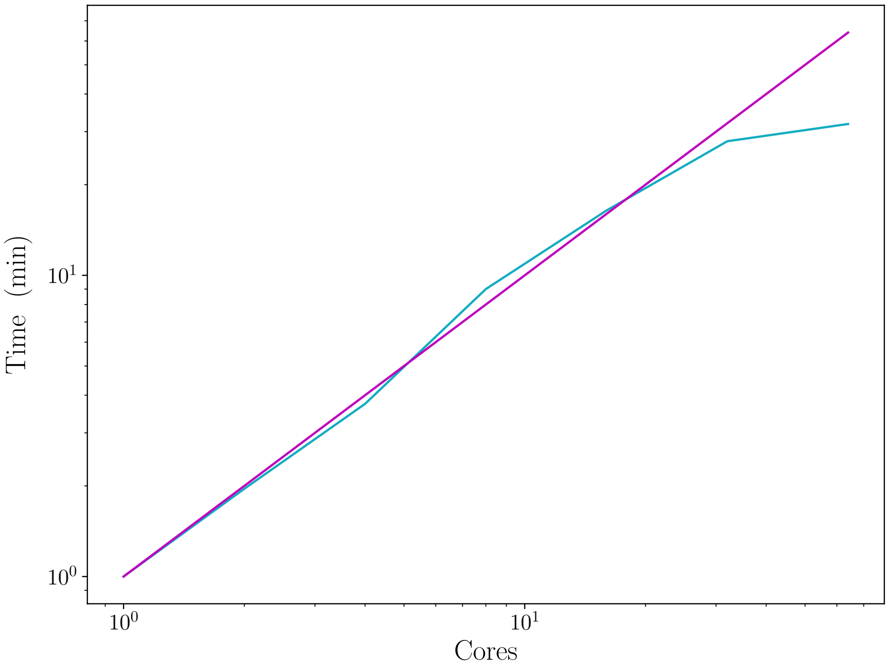


Figure 7: Speed up factor as a function of the number of cores. The purple line represents the ideal case, in which the speed up factor is 1 for 1 core, 2 for 2 cores and so on.

In this case, until 32 cores we obtain that the speed up factor goes approximately as the reference purple line, which means that the computation time decreases to the half each time we double the number of cores. However, for 64 cores we can see there is a deviation respect to the purple line. For this reason, we are going to set the number of cores to 32 in this study.

The table 3 shows that, with the implementation of the fourth order Leapfrog algorithm, if we take the most efficient configuration ($\epsilon$ and $i = 3$), we are able to reduce the computation time a factor 18 in order to achieve the convergence. It is also possible to see that the acceptance rate has also increased respect to the second order leapfrog algorithm, although it decreases a bit respect to the the $i = 2$ case as it is explained with figure 6.

Table 3: Comparison between the second and fourth order Leapfrog algorithm, for $256^3$ cells.

|  | Iteration of convergence | Convergence time (h) | Acceptance |
|---|---|---|---|
| $2^o order$<br>$\epsilon$ | 2500 | $55,81$ | $52,0\%$ |
| $4^o$ order<br>$2\epsilon,\ i = 1$ | 340 | $13,29$ | $51,75\%$ |
| $4^o$ order<br>$\epsilon\, i = 2$ | 100 | $6,96$ | $83,75\%$ |
| $4^o$ order<br>$\epsilon,\ i = 3$ | 33 | $3,12$ | $78,75\%$ |

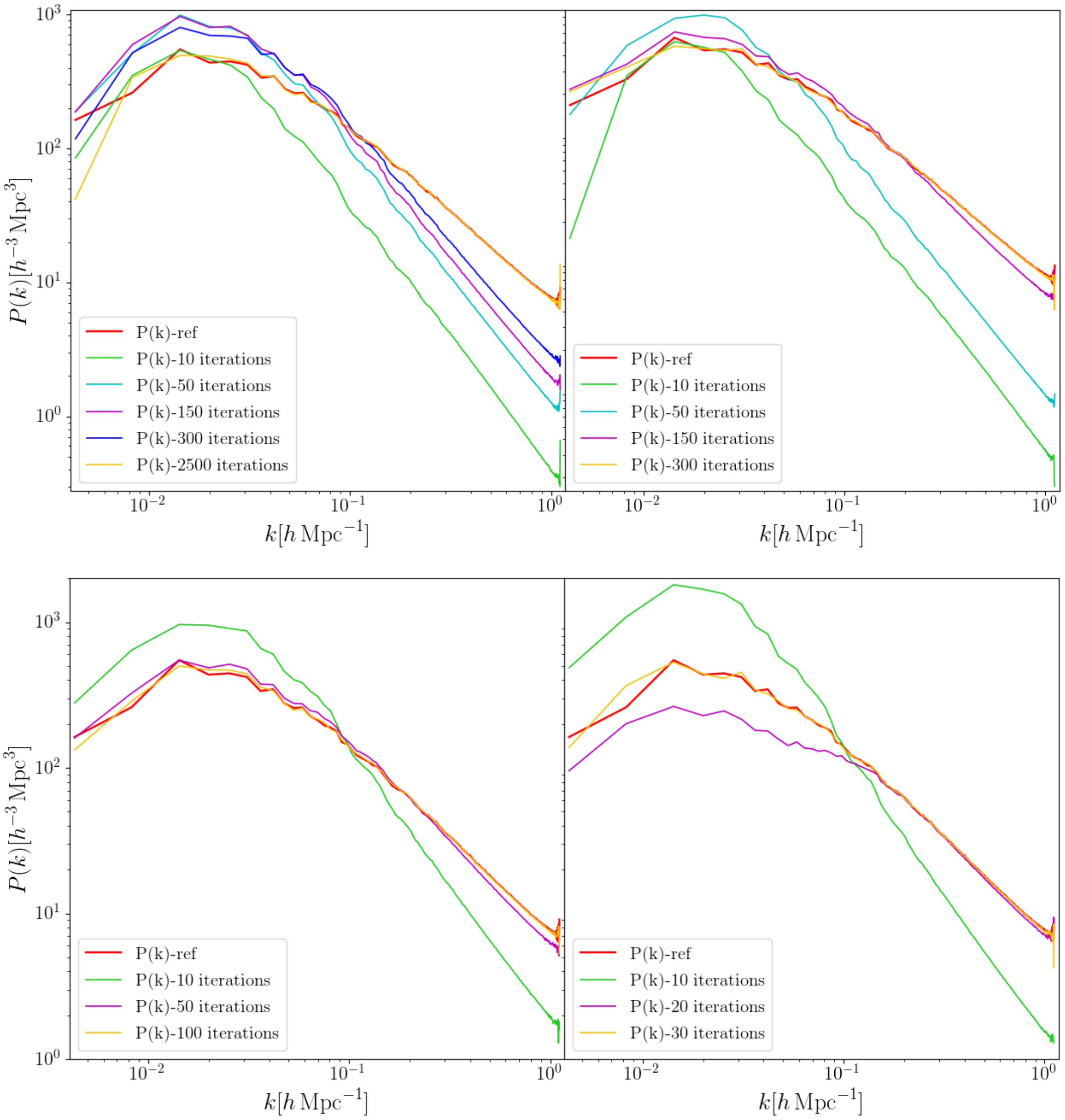


Figure 8: Power spectrum for different iterations compared to a power spectrum of a converged chain. Upper left: power spectrum for different iterations with the second order Leapfrog algorithm. Upper right: power spectrum for different iterations with the fourth order Leapfrog algorithm, for $i = 1$ and a stepsize $2\epsilon$. Lower left: power spectrum for different iterations with the fourth order Leapfrog algorithm, for $i = 2$ and a stepsize $\epsilon$. Lower right: power spectrum for different iterations with the fourth order Leapfrog algorithm, for $i = 3$ and a stepsize $\epsilon$

In the figure 8 we can see the evolution of the power spectrum from a lower iteration until the convergence is reached, for the optimal case of the 3 values of $i$. As it is previously done, a power spectrum at a very high iteration is going to be taken as a reference to represent the converged case, and we are going to compare the power spectrum of the two methods (second and fourth order algorithm). While with the second order Leapfrog algorithm the convergence is reached at the iteration $\sim$ 2500, with the fourth order method we can get convergence at iteration $\sim$ 300, for the case $i = 1$, $\sim$ 105, for $i = 2$, and $\sim$ 32, for $i = 3$; as we can see in table 3. In figure 8, the results are shown for only one seed, so values can change a bit due to those one in the table are obtaining by doing means for different seeds.

Having set the number of forward steps to $i = 3$, and the stepsize to $\epsilon$, which was the optimal one as we saw previously, now, we are in conditions to study the convergence of the fourth order Leapfrog algorithm compared to the second one. To estimate the convergence in a more precise way, for each iteration, we are going to calculate the difference with respect to the reference power spectrum, and summing up for all $k$. Convergence is achieved when this difference is smaller than a threshold for at least 50 consecutive iterations. Then, we are going to represent the ratio of the reference power spectrum with the power spectrum at that iteration (previously estimated), and the next ones to see if it is approximately one, which would mean that the power spectrums are already on top of the reference one and, therefore, we have reached the convergence.

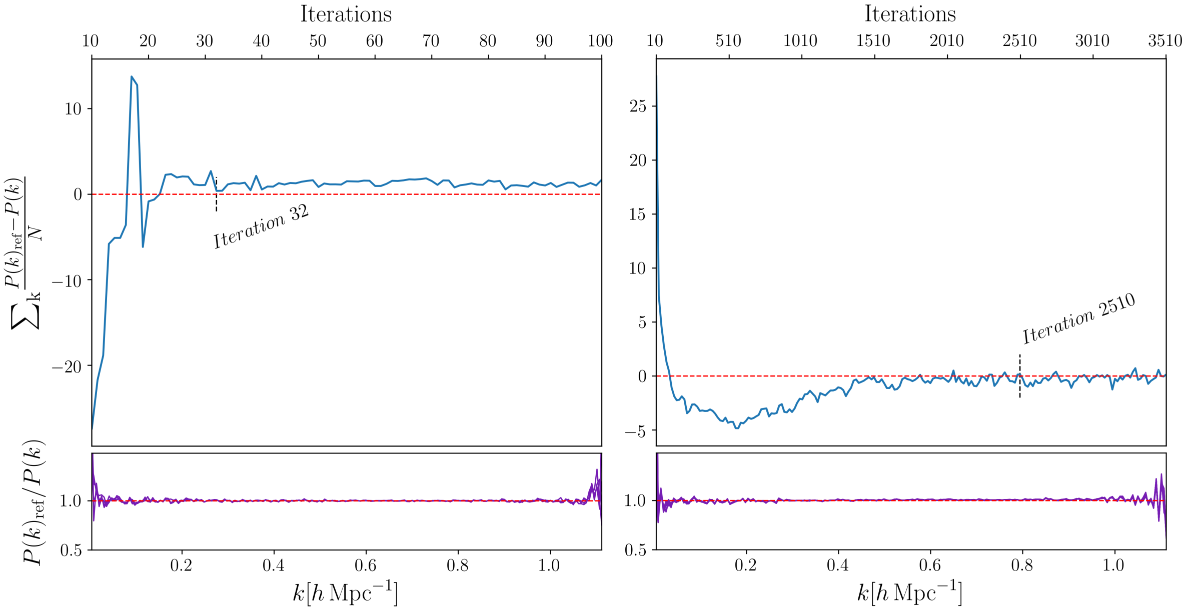


Figure 9: Difference of the reference power spectrum with each iteration power spectrum as a function of the number of iterations (above) and rate of the reference power spectrum and the converged power spectrum estimated and the next iterations power spectrums (below). On the left, the fourth order Leapfrog algorithm is shown ,and on the right, the second order Leapfrog algorithm.

We can see how the iterations of convergence in figure 9 are in agreement with the values of table 3. However, some differences could exist due to the fact that we are not taking averages for different seeds in this representation, otherwise, it is for only one chain.

Now, to prove that the convergence has reached at iteration 32, we are going to appeal to the Gelman Rubin test. Multiple chains are supposed to converge to some stationary distribution, so comparing the means and variances within one converged chain to the samples of independently run chains can prove this convergence. In this test we have to run $N_{chains}$ of length $N_{length}$, that are supposed to have the same target distribution but starting at different points, so each one has a different seed. The output of the chain is denoted as $x_{c,s}$, with $c \in 1, 2, ..., N_{chains}$ and $s \in 1, 2, ..., N_{length}$. $x$ is in this case overdensity $\delta_i$ of each cell. The idea is to compare the variance of the $N_{chain}$ means of the different chains to the mean of the variance of each individual chain. Gelman and Rubin (1992) introduced the parameter $R$, called the Potential Scale Reduction Factor (PSRF), which with the value $R - 1 = 0.1$ is assumed to represent a converged chain [11].

The first step is to calculate each chain's mean value:

$$\overline{x_c} = \frac{1}{N_{length}} \sum_s x_{c,s}. \tag{115}$$

Then, we calculate each chain's variance:

$$\sigma_c^2 = \frac{1}{N_{chains} - 1} \sum_s (x_{c,s} - \overline{x}_c)^2. \tag{116}$$

We calculate now all chain's mean:

$$\overline{x} = \frac{1}{N_{chains}} \sum_c \frac{1}{N_{length}} \sum_s x_{c,s} = \frac{1}{N_{chains}} \sum_c \overline{x}_c. \tag{117}$$

The weighted mean of each chain's variance is expressed as

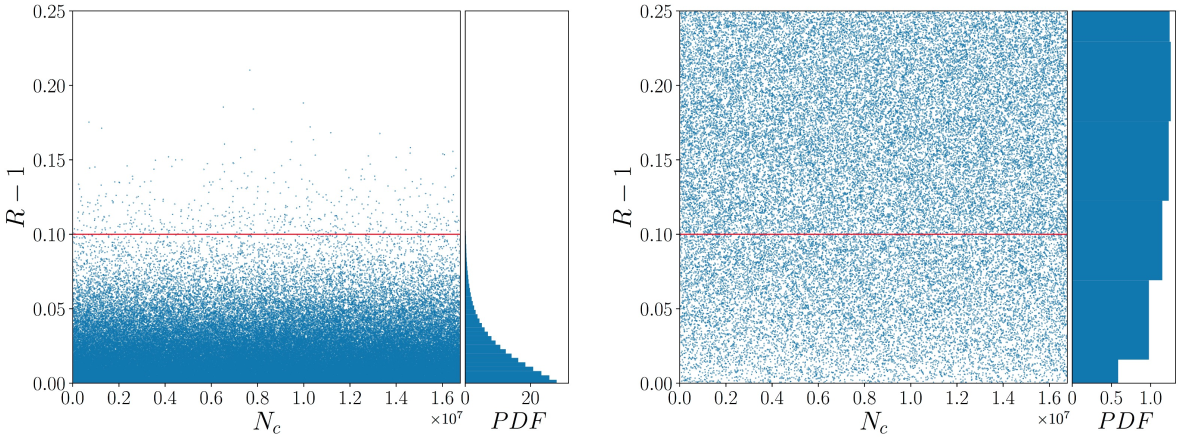
$$B = \frac{N_{lenght}}{N_{chains} - 1} \sum_c (\overline{x}_c - \overline{x})^2, \tag{118}$$

and the average variance:

$$W = \frac{1}{N_{chains}} \sum_c \sigma_c^2. \tag{119}$$

Finally, the Potential Scale Reduction Factor is defined as

$$R = \sqrt{\frac{N_{lenght} - 1}{N_{length}} + \frac{N_{chains} + 1}{N_{length} N_{chains}} \frac{B}{W}}. \tag{120}$$



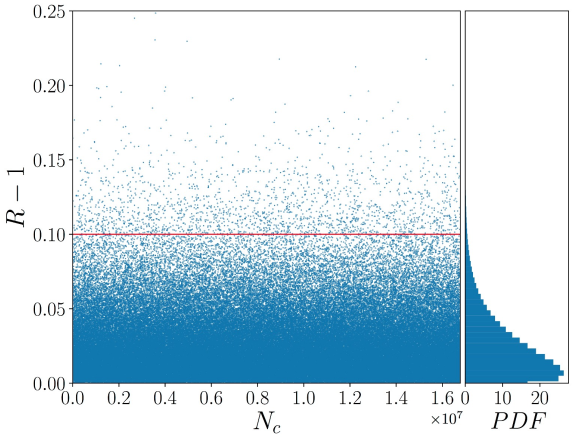


Figure 10: Upper left: German Rubin test from 40 to 500 iterations with fourth order Leapfrog algorithm, for the configuration $i = 3$ and stepsize $\epsilon$. Upper right: German Rubin test from 3000 to 3460 iterations with second order Leapfrog algorithm. Lower: German Rubin test from 3000 to 12000 iterations for second order Leapfrog algorithm. The red line represents the $R-1$ parameter.

We have represented the range in which the Markov chain has converged and, therefore, where the HMC has got the target distribution. As it is mentioned before, we evolve the system with the Hamilton's equations of motion, however, the initial samples are not belonging to the correct target distributions but what is called the *burn-in* phase. In figure 10 it is possible to see that with a small range, from 40 to 500 iterations, almost all points in the plot are under the red line, which represents $R-1$, so there is convergence. However, for the case of second order Leapfrog algorithm we need a bigger range to represent a similar behavior in the Gelman-Rubin test: from 3000 to 12000 (figure 10 lower). Otherwise, if we take the same range as in the case of the fourth order, we can prove the Markov chain is far to converge, seeing in figure 10 upper right, that there are more points above the red line than below. We have also represented the histograms of density of points to visualize the number of points on each value of $R-1$ easily.

Finally, we compute the correlation length of all modes of the power spectrum over the iteration distance. The correlation length is express as

$$Cn(k) = \frac{1}{N-n}\sum_{i=0}^{N-n}\frac{P(k)^i - \langle P(k)\rangle \left(P(k)^{i+n} - \langle P(k)\rangle\right)}{\sigma^2(P(k))} \tag{121}$$

where $k$ is one mode of the power spectrum, $N$ is the number of samples and $n$ is the distance between iterations.

With this test we can be sure that we are obtaining independent samples when we sample the posterior. The figure 11 shows that there is a remarkable difference between the second order Leapfrog algorithm and the fourth order one, having, in the first case, a correlation length about $\sim$ 300 iterations and in the second one $\sim$ 10. This means that once we run the chain and it reaches the convergence, we can obtain independent samples each 10th iterations. In the second order case it takes so long to reach the convergence (as we saw in table 3, almost 56 hours for the 2500 iterations) that it is better to wait 300 iterations more to get independent samples. However, with the fourth order Leapfrog algorithm, due to the fast convergence, we have the alternative of running several chains, with trivial parallel computing, for different seeds until the chain reaches the convergence after $\sim$ 32 iterations (as we could see in table 3); or run one chain until the convergence is reached and get independent samples each 10th iterations.

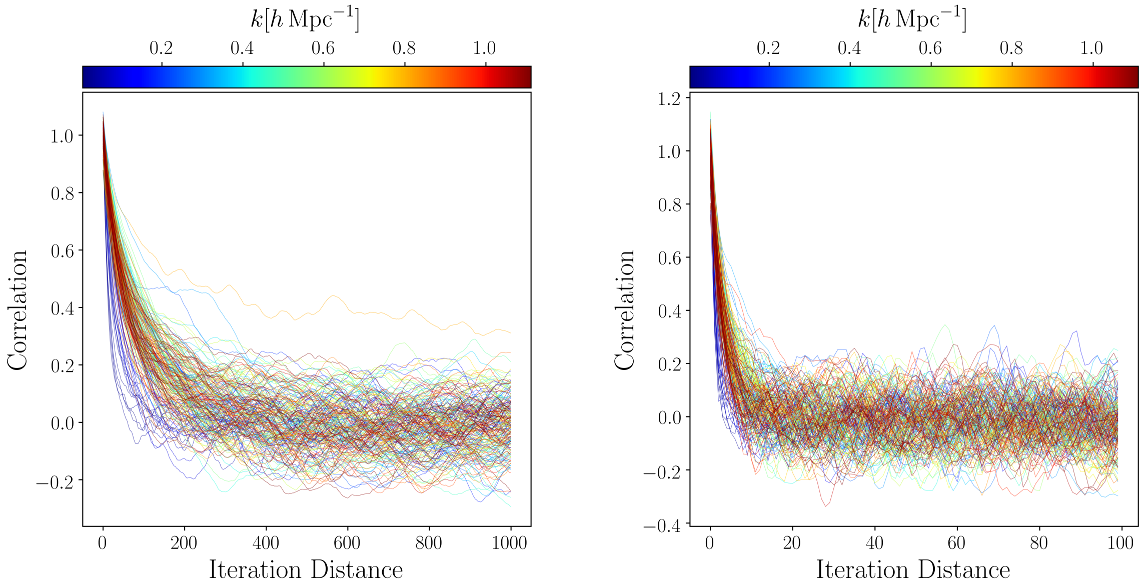


Figure 11: Correlation length as a function of the iteration distance for the second order Leapfrog algorithm (left) and the fourth order one (right).

In figure 12, the resulting reconstruction of the primordial fluctuations, using the fourth order Leapfrog algorithm implementation, is shown. We have taken a volume of (1250 Mpc $h^{-1})^3$ and $256^3$ cells. The following figure has also been done integrating over a slice of width 10 cells, that corresponds to 48.83 Mpc $h^{-1}$. We can compare the mock catalog with a mass assignment scheme[5], and the resulting initial conditions, observing that the density peaks we see in the primordial fluctuations reconstructed with `ARGO` are in agreement with the more dense regions in the catalog.

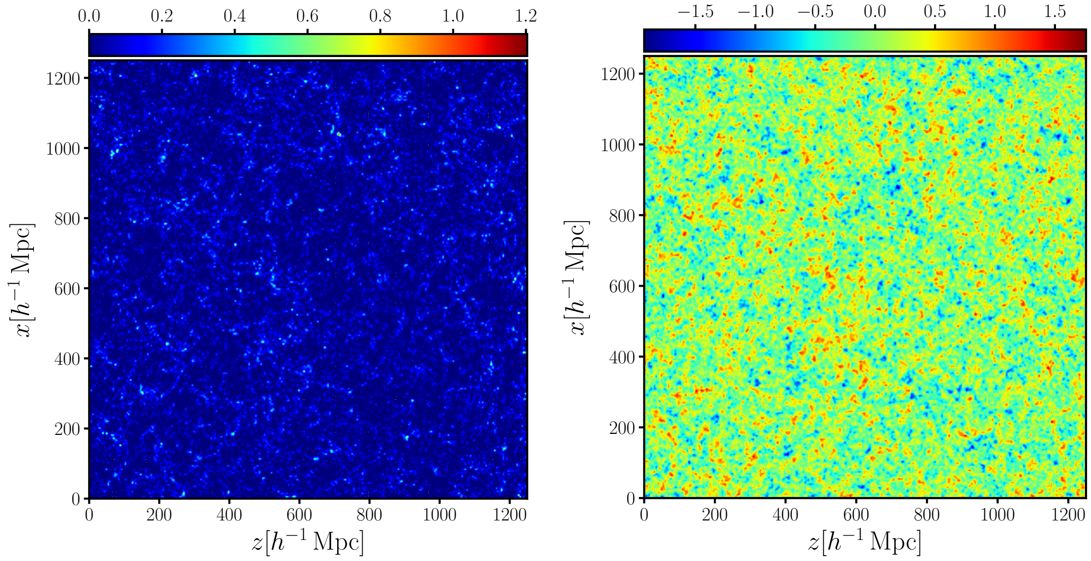


Figure 12: Comparison between the catalog (left) and the reconstructed primordial fluctuations, $\delta(x)$, with fourth order Leapfrog algorithm, for $i = 3$ and a stepsize $\epsilon = 0.06$ (right).

[5] We will see more information about the generation of this catalog in section 5.1

### 4.3 Implementation of the fourth order Leapfrog algorithm in BIRTH code

This fourth order Leapfrog algorithm has also been implemented in a new more advanced code developed by Kitaura et al (in preparation): BIRTH, which is based on the KIGEN code [24]. The idea of this method consists of splitting the problem of reconstructing the primordial fluctuations into two steps. In a first step, it is assumed that the tracers of the large scale structure reside at initial cosmic times. Therefore, one has to reconstruct a continuous Gaussian field with a given power spectrum constrained by a set of discrete tracers. The ansatz of the ARGO code is in this case ideal, as the lognormal prior is accurate for initial cosmic times, when shell crossing has not occurred, and the density fluctuations are very low. We note, that non-Poissonian likelihoods are also implemented in the ARGO and BIRTH codes [25], but for the kind of objects considered in this study and the cell-size resolution, the Poisson assumption is valid [26]. In this way, we obtain a Gaussian field in the first step.

In a second step, we use that Gaussian field to evolve it with some arbitrary structure formation model. This yields peculiar velocities and displacements, which can be applied to the actual observed distribution of galaxies in redshift space to generate the input data required in the first step. In practice, one starts assuming a perfectly homogeneous density field, which yields vanishing velocities and displacements, and, therefore, one initially directly takes the observed galaxy distribution as the tracers at initial cosmic times. This produces, in a second step, a non vanishing Gaussian field, which will deliver a non trivial mapping between Lagrangian and Eulerian coordinates. As the Markov chain proceeds, the set of possible solutions compatible with the data are obtained.

In particular, we use as a structure formation model connecting the initial with the final cosmic times a Lagrangian perturbation theory (LPT) based approach, which includes second order tidal fields. Moreover, it includes perturbation theory corrections on small scales based on the spherical collapse model, to mitigate artificial shell crossing inherent to LPT methods [27, for a detailed description of the approach]. The BIRTH code includes a self-consisten treatment of the survey geometry, selection function, bias and cosmic evolution (for more details see Kitaura et al in prep). We apply this code, which also applies Hamiltonian sampling, as in the case of ARGO, to a realistic mock galaxy catalog, resembling the CMASS-BOSS final data release.

Moreover, a study of the efficiency with the fourth order Leapfrog algorithm has been done, following the same methodology explained before for the ARGO code.

In table 4, with the same format as the one obtained for ARGO code (table 2), we can see the results with BIRTH code. It is remarkable how for $i = 1$, the iterations we need to reach the convergence are very few comparing to table 2 for the same case. We can also observe how the acceptance is more or less constant for all stepsizes and all $i$ values, showing a really different behavior respect to the acceptance in ARGO code. In this last case, as we could see in table 2 and the histograms 5 and 6, the acceptance rate went worse if we increased the stepsize, and for the same stepsize, if we increased the $i$ value. For BIRTH code we see that the best configurations are: a stepsize $8\epsilon$ for $i = 1$, $6\epsilon$ for $i = 2$ and a stepsize $\epsilon$ for $i = 3$.

Table 4: Results with the fourth order Leapfrog algorithm through `BIRTH` code.

| Stepsize | Iteration of convergence | Convergence time (min) | Acceptance |
|---|---|---|---|
| i=1 | | | |
| $\epsilon$ | 30 | $68,87$ | $57,5\%$ |
| $2\epsilon$ | 25 | $78,80$ | $59,0\%$ |
| $4\epsilon$ | 38 | $122,51$ | $59,0\%$ |
| $6\epsilon$ | 28 | $95,69$ | $60,5\%$ |
| $8\epsilon$ | 31 | $50,00$ | $59,0\%$ |
| $10\epsilon$ | 32 | $115,55$ | $60,25\%$ |
| i=2 | | | |
| $\epsilon$ | 32 | $50,00$ | $64,25\%$ |
| $2\epsilon$ | 25 | $73,14$ | $58,5\%$ |
| $4\epsilon$ | 28 | $55,27$ | $63,5\%$ |
| $6\epsilon$ | 23 | $46,05$ | $62,0\%$ |
| $8\epsilon$ | 27 | $57,08$ | $59,5\%$ |
| $10\epsilon$ | 30 | $59,79$ | $64,5\%$ |
| i=3 | | | |
| $\epsilon$ | 30 | $62,06$ | $60,25\%$ |
| $2\epsilon$ | 37 | $128,94$ | $53,5\%$ |
| $4\epsilon$ | 33 | $206,87$ | $57,0\%$ |
| $6\epsilon$ | 28 | $145,79$ | $60,75\%$ |
| $8\epsilon$ | 32 | $69,08$ | $62,5\%$ |
| $10\epsilon$ | 33 | $82,99$ | $59,0\%$ |

Here, the same representation of the computational time as seeing in figure 4, but for the case of `BIRTH`, is also included.

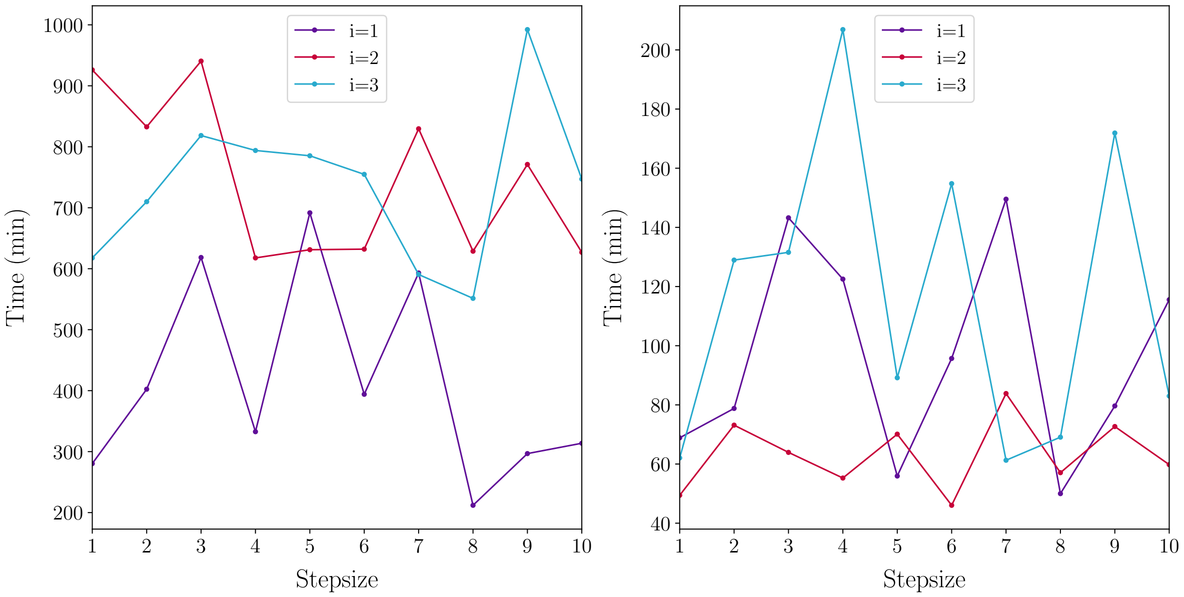


Figure 13: On the left, the computation time for 100 iterations as a function of the stepsize, for the different values of $i$. On the right, the convergence time as a function of the stepsize, for the different values of $i$.

The figure 13 shows big differences in the computational time from a stepsize to another in the same $i$ value, while in figure 4 we could see that the computational time followed a closer linear tendency with the stepsize: the time was getting higher while we increased the stepsize value. However, here, a more complex behavior is observed. In the figure 13 on the right, we can also see that $i = 2$ gives a more stable curve, and the value of $6\epsilon$ is the best choice of all configurations.

In figure 14 we can also see that the power spectrum of this optimal configuration, at iteration 21, has reached the convergence and it is on top of the theoretical power spectrum (black curve), obtained with `CAMB`[6]. We can also observe that, for small values of $k$, the power spectrum deviates from the theoretical one, however, this could be due to the cosmic variance, also mentioned in section 4.1.

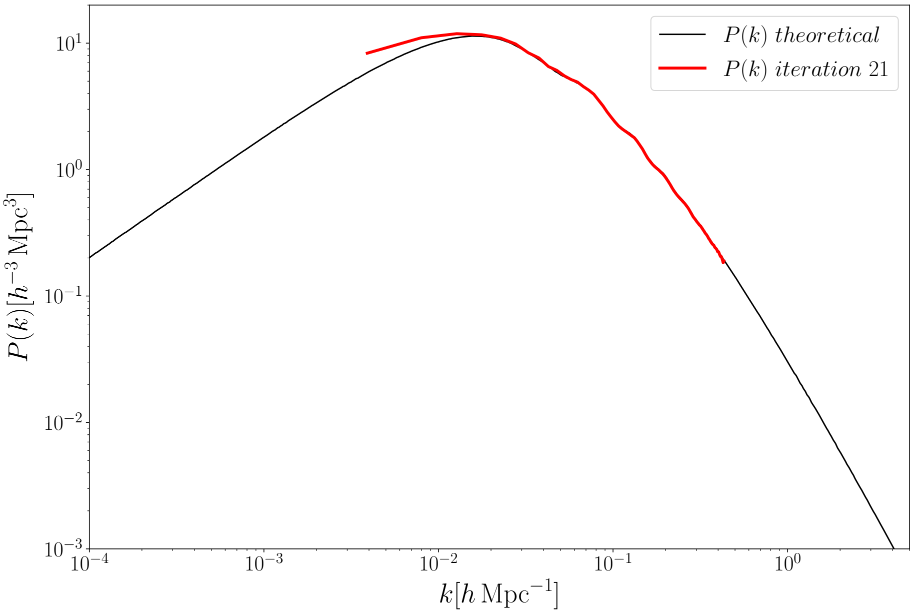


Figure 14: The red curve represents the power spectrum of BIRTH with the fourth order Leapfrog algorithm, for $i = 2$ and a stepsize $6\epsilon$, at iteration 21. The black curve is the theoretical power spectrum.

[6] https://camb.info/

# 5 Preparation of the Initial Conditions for constrained N-Body Simulations

In this section, we will describe the mock catalog used for this project, reproducing the CMASS-BOSS galaxy distribution. Then, we will use the primordial fluctuations reconstructed by `BIRTH` at $z = 60$, to generate the whitenoise and introduce it in the `PKDGRAV3` code. It has been necessary to adapt the code for reading the whitenoise input file, instead of calculating it. Subsequently, we will perform the N-body simulation until $z = 0.57$, the mean redshift of the CMASS-BOSS galaxy samples. Moreover, a description of N-body simulations is included in section 5.2.

## 5.1 Mock CMASS Catalog

In `ARGO` and `BIRTH`, and for the purpose of this work, a halo abundance at $z = 0.57$, from one of the BigMD simulations[7], is used. This simulation was performed using the TreePM N-body code `GADGET-2` with 38403 particles and the volume of $(2.5h^{-1}\mathrm{Gpc})^3$, in a framework of Planck $\Lambda$CDM cosmology [19]. We use a halo catalog, where halos are identified with spherical over-density halo finders Bound Density Maximun (BDM). From this, we assign galaxies to grid cells with an assignment scheme (NGP or equivalently CIC[8]).

In this project, in particular, we reproduce the BOSS (Baryon Oscillation Spectroscopic Survey) galaxy distribution. BOSS mapped the spatial distribution of luminous red galaxies (LRGs) and quasars to detect the characteristic scale imprinted by baryon acoustic oscillations in the early Universe. The principal BOSS galaxy samples are LOWZ and CMASS. These two catalogues target galaxies using a set of color-magnitude cuts. LOWZ targets low-redshift galaxies ($z \leqslant 0.43$), while CMASS is a high-redshift sample ($0.43 \leqslant z \leqslant 0.7$). This last one is focused on the observation of distant luminous red galaxies (LRGs) and generating a three dimensional spatial distribution of the large-scale structure of the Universe. For this master thesis we use the mock galaxy catalog resembling the CMASS-BOSS final data release.

## 5.2 N-body simulations

N-body simulations are numerical solutions of equations of motions for $N$ particles interacting gravitationally. It comes up from the difficulty of the treatment of fluctuations in the non-linear regime, which makes impossible to use analytical methods.

It is possible to represent the expanding Universe as a box containing a large number of point masses, $N$, interacting through their mutual gravity. It is common to assume periodic boundary conditions in all directions in the cube, and a length, at least, the a scale at which the Universe becomes homogeneous for a fair sample (representative of the Universe as a whole).

The simplest way to compute the non-linear evolution of a cosmological fluid is to represent it as a discrete set of particles and sum the interactions between them to calculate the Newtonian forces. Therefore, we have the Newton's law plus, in some cases, an external potential field. Such calculations are called particle-particle computations. Setting a small timestep, one can

[7] http://www.multidark.org//

[8] We will see more about this schemes in section 5.2

use the resulting acceleration to update the particle velocity and then its position. New positions can be used to recalculate the interparticle forces, and so on.

These techniques represent not the motion of a discrete set of particles, but an approximation of a fluid. There is also a numerical problem with the sum of the forces: The Newtonian gravitational force between two particles increases as the particles approach each other and it is necessary to choose an extremely small timestep to resolve the large velocity changes this induces. However, this would require a lot of computational time, so one usually avoids these problems by treating each particle as an extended body instead of a point mas. The Newtonian force between particles is [28]:

$$F_{i,j} = \frac{Gm^2(x_j - x_i)}{(\epsilon^2 + |x_i - x_j|^2)^{3/2}}, \tag{122}$$

where particles are at positions $x_i$ and $x_j$ and have the same mass $m$. The parameter $\epsilon$ is called softening length. It avoids the singularity (infinite forces) when the distance of two particles approaches to zero, so the forces no longer diverge and the total force estimated from all particles is a smooth and continuous function. It also reduces the gravity at closer distances, generating a bias of the average of the N-body force, what makes simulations results, on scales smaller than a few $\epsilon$, unreliable [29].

The computational time of this method is very high, scaling roughly as $O(N^2)$. Therefore, faster algorithms have been developed:

- Tree codes: this method uses the Barnes-Hut Algorithm (BHA), which recursively subdivides the problem space in order to facilitate the calculation of inter-particle distances. The result of this division is a set of sub-spaces that are congruent with one another, and are spatially scaled versions of the original space. This process can be repeated until either one ore no particles are in a cell. Finally, the center of mass of the particle distribution in each cell is calculated. As a recursive process of subdividing the space, the result of each stage can be stored in a tree structure, that we have to go through to carry out the force calculations [30]. This reduces the computing time to $O(N \log N)$.
- Fast Multipole Methods: the tree codes do not take into account the fact that nearby particles will be subject to a similar acceleration due to distant groups of particles. This method takes advantage of this idea and uses multipole expansions to compute the force from a distant source cell within a sink cell. This reduces the complexity from $O(N\ log(N))$ to $O(N)$.
- Particle-mesh techniques: in this technique, the forces (equation 122) are calculated assigning mass points to a regular grid and solving Poisson's equation on it:

  $$\triangle\,\phi = 4\pi G\rho \tag{123}$$
  $$\phi_k = -4\pi G\frac{\rho_k}{k^2} \tag{124}$$

  This regular grid, with periodic boundary conditions, allows the use of Fast Fourier Transforms (FFTs) to obtain the potential, which leads to a considerable increase in speed. The density on the grid can be calculated as

  $$\rho(q) = \frac{M^3}{N}\sum_{i=1}^{N} W(x_i - q). \tag{125}$$

For simplicity, we adopt a notation such the Newtonian gravitational constant $G = 1$, the length of the side of the simulation cube is unity and the total mass is also unity. $M$ is the number of mesh-cells along one side of the simulation cube and $N$ is the total number of cells. The vector $q$ is $n/M$, where $n$ represents a grid position. $W$ defines a weighting scheme designed to assign mass to the mesh. There are several possible choices of weighting function $W$: nearest grid point (NGP), cloud-in-cell (CIC), triangular-shaped clouds (TSC).

## 5.3 PKDGRAV3

The c++ code `PKDGRAV3` [31], developed by Douglas Potter et al, provides the fastest time-to-solution for large scale cosmological N-body simulations. It uses Fast Multipole Method together with individual and adaptive particle time steps, and can run on supercomputers with GPU-accelerated nodes [32]. It has performed the evolution of a 2 trillion particles simulation of the $\Lambda$CDM model from $z = 49$ to $z = 0$ in less than 80 hours, in the Swiss National Supercomputing Center Machine, using 4000+ GPU-accelerated nodes.

In this section, `PKDGRAV3` code is used to read the output file of `BIRTH`: the reconstruction of the density contrast in an specific iteration, where we are sure that it has reached the convergence. These primordial fluctuations have been obtained from a mock catalog resembling the CMASS-BOSS catalog of Luminous Red Galaxies, as it is mentioned before.

The first step is to obtain the whitenoise of the primordial fluctuations, so once we have $\delta(x)$, we performance a Fourier transformation to get $\delta(k)$, and then, we can obtain the whitenoise by:

$$n(\vec{k}) = \frac{\delta(\vec{k})}{\sqrt{P(\vec{k})}}, \tag{126}$$

where $P(\vec{k})$ is the theoretical power spectrum at $z = 0$.

This whitenoise, $n(\vec{k})$, has a contrast power spectrum, that is:

$$\langle \mid n(\vec{k}) \mid^2 \rangle = 1. \tag{127}$$

From equation 126 and using 127, it can be easily seen that $\langle \mid \delta(k)^2 \mid \rangle = P(k)$, so the initial field has the correct power spectrum.

However, $\delta(\vec{x})$ has been obtained at a redshift $z = 60$ and, therefore, we need to normalize at this redshift. So instead using equation 126 we compute:

$$n(\vec{k}) = \frac{\delta(\vec{k}, z)}{\sqrt{D^2(z) P(\vec{k}, z = 0)}}. \tag{128}$$

As we have previously seen, $D^2(\vec{k})$ is the growth factor.

It is also required to save the whitenoise in k-space, which is the format of the input we need to introduce in PKDGRAV3. In this procedure we have used the Hermitian condition. The discrete Fourier Transform of a real-valued signal has Hermitian symmetry, which means that half of the outputs are redundant: $out[i]$ is the conjugate of $out[n-i]$. Therefore, we can avoid these reduntant values and improve a factor two the speed and memory usage. While the input is $n$ real numbers, we have $n/2+1$ complex numbers in the output (the non-reduntant ones) [33].

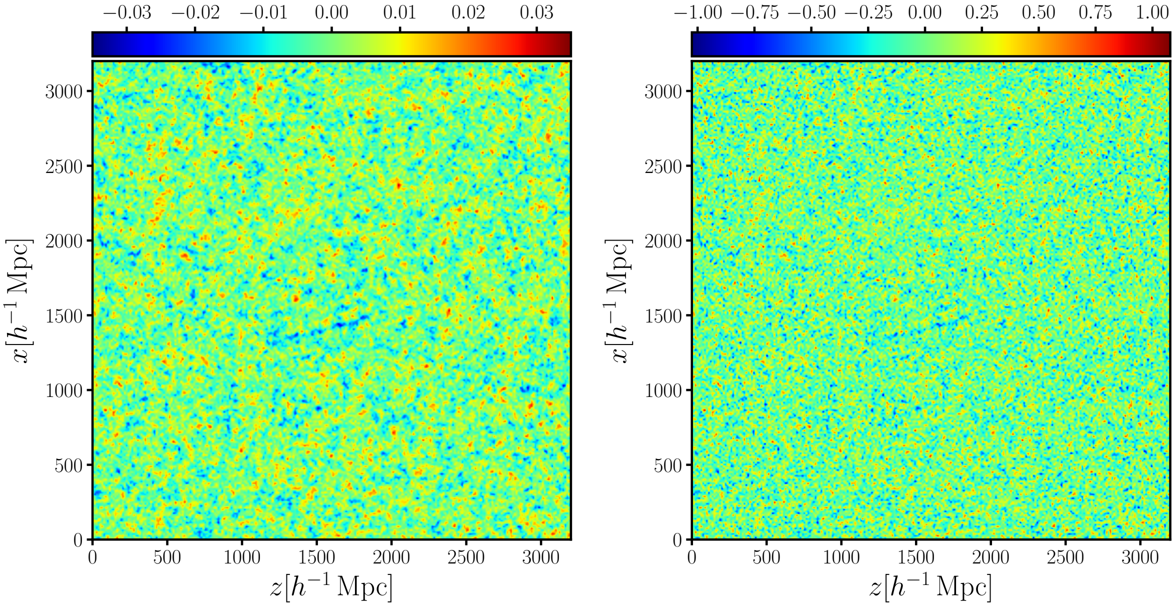


Figure 15: On the left: the primordial fluctuations, $\delta(\vec{x})$, obtained with BIRTH. On the right: the white noise.

The next step is to read in PKDGRAV3 the whitenoise we have generated, instead of creating it through the code, which is its natural procedure. Then, we run the code giving also the transfer function of the theoretical power spectrum (solving equation 82 for $T^2(\vec{k})$), the one it has been used in BIRTH.

To be sure that everything is computed in the same format and PKDGRAV3 is reading a correct whitenoise and generating the initial conditions in a proper way, we are going to obtain an output at $z=60$, so we can compare this with the output of BIRTH, and check that the initial conditions have the same structures.

In figure 16 we can see these two initial conditions. However, some differences are observed between the one obtained through BIRTH and the one obtained through PKDGRAV3. This is due to the fact that the first mentioned is a linear density contrast, a Gaussian distributed field, obtained by multiplying the whitenoise generated by the code to the theoretical power spectrum. On the other hand, PKDGRAV3, generates the initial conditions by evolving the Gaussian perturbation field through 2LPT to the initial simulation redshift, $z=60$.

To read the output file of PKDGRAV3 we have used *pynbody*, which is an analysis *python* package for N-body simulations. Through it, we have obtained a coordinate file ($x, y$ and $z$) of the particles from a binary one (the original format of PKDGRAV3 outputs). Then, we have calculated the density on a grid as we saw in equation 125, from the set of coordinates $x, y$ and $z$, allowing us to represent the primordial fluctuations as seen in figure 16.

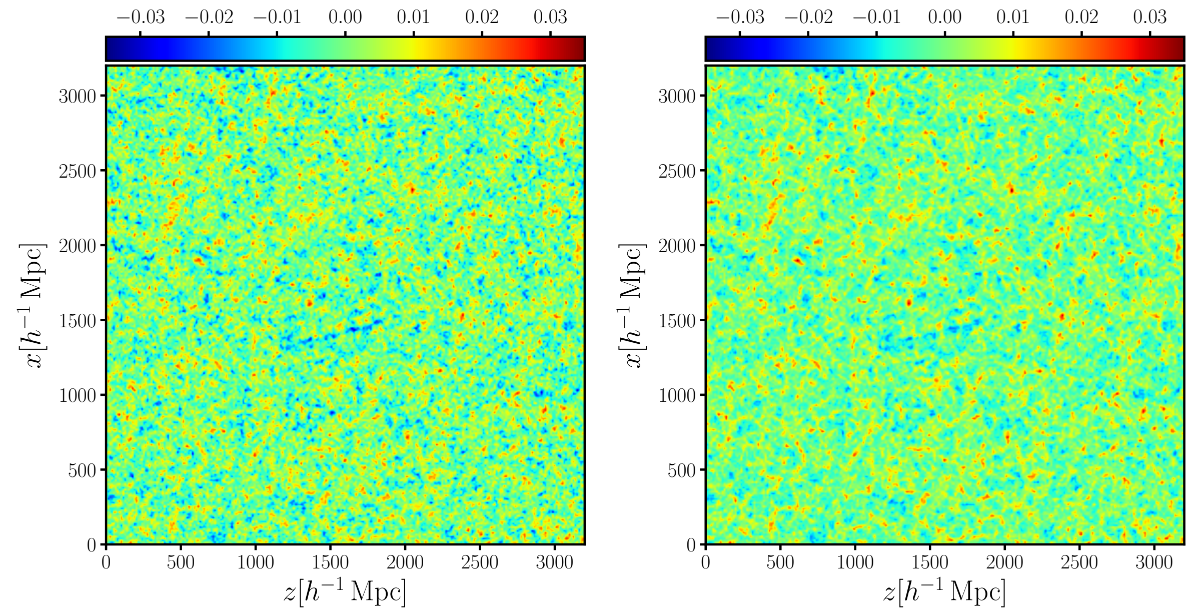


Figure 16: On the left: the primordial fluctuations, $\delta(\vec{x})$, obtained with Birth. On the right: the primordial fluctuations, $\delta(\vec{x})$, obtained by PKDGRAV3. Both at $z = 60$.

In the figure 17 we can see the resulting simulation at $z = 0.57$ from `PKDGRAV3` and its primordial fluctuations at $z = 60$. A high spatial resemblance between the obtained dark matter field at $z = 0.57$ and the initial conditions is clearly seen.

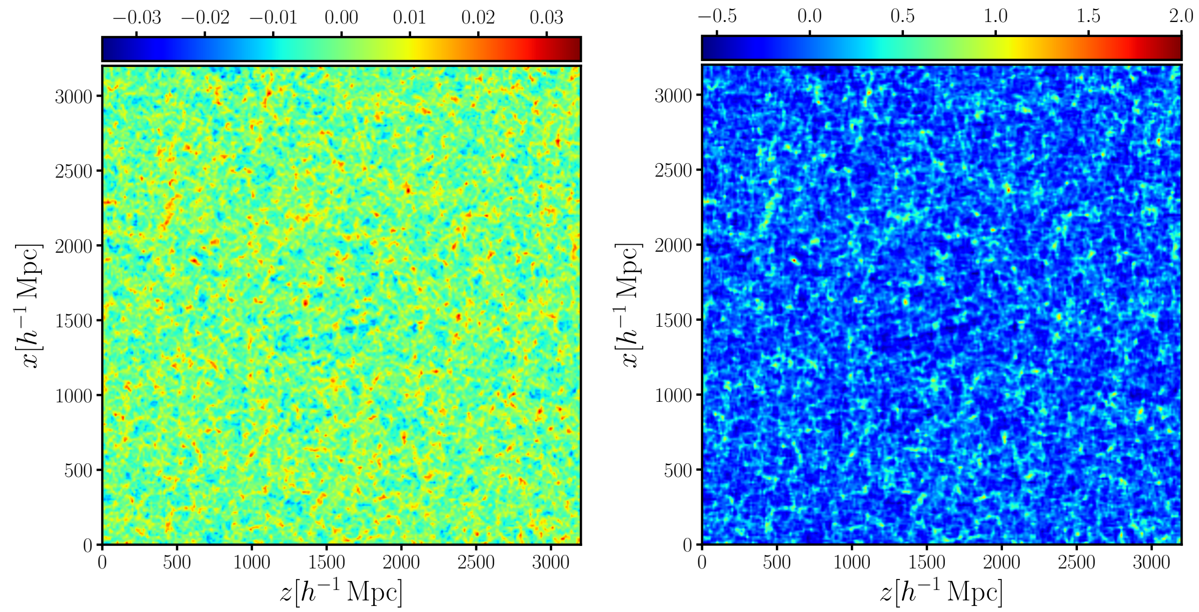


Figure 17: On the left: Primordial fluctuations, $\delta(\vec{x})$, obtained with `PKDGRAV3`. On the right: N-body simulation at $z = 0.57$, obtained with `PKDGRAV3`.

# 6 Conclusion

In this project, we have explored efficient ways to study the large scale structure within a Bayesian framework. We aim at sampling the dark matter field given a set of discrete tracers, and to do so, we resort to Hamiltonian sampling to tackle this non-Gaussian problem. To achieve this, one needs to solve the discretized Hamiltonian equations, which has been done with the second order Leapfrog integration. We have implemented a novel scheme to this field, that recursively uses a combination of forward and backward steps to effectively obtain a higher order Leapfrog algorithm. In particular, a fourth order scheme has been implemented in the `ARGO` code to improve its efficiency.

Several tests have been developed to study the convergence of the Markov chain, the computational time required to reach this convergence, and the acceptance of the iterations. With this information we have been able to estimate the best value for the $i$ parameter, in equation 114, and the proper stepsize, which allows us to sample the parameter space faster. As a general conclusion we could see that the computational time was increased when we took a higher value of stepsize for a given set $i$ value, due to the lower percentage of acceptance; and we could also observe that the acceptance got worse if we took a higher value of $i$, for the same value of stepsize. This can be seen in histograms 5 and 6 and table 2. For the best configuration we have run the code again using a higher resolution, $256^3$ cells, which is the one we usually use to do the reconstruction of the primordial fluctuations. Table 3 shows the big improvement of the fourth order Leapfrog algorithm with respect to the second order one. We can see that with the parameter $i = 3$ and stepsize $\epsilon$, we obtained the most efficient choice, where we could reduce the computational time of the code by almost a factor 18. The Gelman-Rubin test was also introduced to demonstrate that, from iteration 40 to 500, all chains had reached the convergence. However, taking the same range for the second order Leapfrog algorithm, it was far from reaching the convergence (figure 10 upper right), and we would need a range from 3000 to 12000 (figure 10 lower) to obtain a similar result as in figure 10 upper left. Finally, we also computed the correlation length of all modes of the power spectrum as a function of the distance between iterations (figure 11), observing that, for the second order Leapfrog algorithm, the correlation length was $\sim$ 300 iterations, while in the fourth order case $\sim$ 10, once the convergence was reached. Therefore, we could obtain independent samples each 10th iterations.

This fourth order Leapfrog algortihm has also been implemented in a new more advanced code of reconstruction of primordial fluctuations, `BIRTH`, showing also a fast convergence, but different behavior of the acceptance and, therefore, the computational time, when compared with `ARGO`. This is presented in table 4, where it is possible to observe a more constant tendency of the acceptance percentage along the different values of stepsize.

For the last part of the project, we computed the whitenoise from the output file of `BIRTH`, the reconstructed primordial fluctuations, and we modified `PKDGRAV3` to read it, instead of generating it. We obtained initial conditions at $z = 60$, and we compare the result with the output of `BIRTH`. We could see in figure 16 that both over-density fields were qualitatively similar. Finally, we obtained an output at $z = 0.57$, which is the mean redshift of the BOSS-CMASS catalog. In figure 17 we can observe a high spatial resemblance between the structures of the intial conditions and the ones in the output of the N-body simulation at $z = 0.57$, which corresponds with the reconstructed dark matter field.

With the great improvement in the efficiency of `ARGO` and `BIRTH`, achieved through the implementation of the fourth order Leapfrog algorithm, and the successful connection between `BIRTH` and `PKDGRAV3`; we aim, now, at implementing a full gravity solver within the Hamiltonian sampling. This will allow us to obtain more accurate reconstructions of the local Universe and thereby study in a more robust way non-linear structure formation. We will be able to make a more direct comparison between observations and cosmological models, and investigate in detail potential discrepancies.

# References

[1] R. M. Neal, "MCMC using Hamiltonian dynamics," *ArXiv e-prints*, June 2012.

[2] M. S. Longair, *Galaxy Formation.* Springer, 1998.

[3] S. Perlmutter, G. Aldering, G. Goldhaber, R. A. Knop, P. Nugent, P. G. Castro, S. Deustua, S. Fabbro, A. Goobar, D. E. Groom, I. M. Hook, A. G. Kim, M. Y. Kim, J. C. Lee, N. J. Nunes, R. Pain, C. R. Pennypacker, R. Quimby, C. Lidman, R. S. Ellis, M. Irwin, R. G. McMahon, P. Ruiz-Lapuente, N. Walton, B. Schaefer, B. J. Boyle, A. V. Filippenko, T. Matheson, A. S. Fruchter, N. Panagia, H. J. M. Newberg, W. J. Couch, and T. S. C. Project, "Measurements of $\Omega$ and $\Lambda$ from 42 High-Redshift Supernovae," *"apj"*, vol. 517, pp. 565–586, June 1999.

[4] J. Cepa, *Cosmología Física.* Akal, 2007.

[5] W. Docters, "Determining the cosmological parameters $\Omega_m$ and $\sigma_8$ from peculiar velocity and density-contrast data," 2008.

[6] NASA, "Lambda-education: Optical depth to reionization, $\tau$," 2016.

[7] H. Kurki-Suonio, "Cosmology ii: Galaxy survey cosmology."

[8] A. Riotto, *Lecture Notes on Cosmology.* 2003.

[9] F. V. D. Bosch, *Theory of Galaxy Formation: Newtonian Perturbation Theory.* Yale University, 2017.

[10] H. V. Peiris, *Cosmology part II: The Perturbed Universe.*

[11] M. Ata, *Thesis: Phase-Space Reconstructions of Cosmic Velocities and the Cosmic Web.* Leibniz-Institut für Astrophysik Postdam, 2016.

[12] P. Coles and F. Lucchin, *The Origin and Evolution of Cosmic Structure.* John Wiley & Sons, 2002.

[13] P. Coles and B. Jones, "A lognormal model for the cosmological mass distribution," *MNRAS*, vol. 248, pp. 1–13, Jan. 1991.

[14] A. Knebe, "Lecture for computational cosmology: Initial conditions, universidad autónoma de madrid,"

[15] R. M. Neal, "Probabilistic inference using markov chain monte carlo methods," 1993.

[16] F.-S. Kitaura, *Bayesian Analysis of Cosmic Structures*, p. 143. 2012.

[17] M. Creutz and A. Gocksch, "Higher-order hybrid Monte Carlo algorithms," *Physical Review Letters*, vol. 63, pp. 9–12, July 1989.

[18] F. S. Kitaura and T. A. Enßlin, "Bayesian reconstruction of the cosmological large-scale structure: methodology, inverse algorithms and numerical optimization," *MNRAS*, vol. 389, pp. 497–544, Sept. 2008.

[19] F.-S. Kitaura, M. Ata, R. E. Angulo, C.-H. Chuang, S. Rodríguez-Torres, C. H. Monteagudo, F. Prada, and G. Yepes, "Bayesian redshift-space distortions correction from galaxy redshift surveys," *MNRAS*, vol. 457, pp. L113–L117, Mar. 2016.

[20] F.-S. Kitaura, J. Jasche, and R. B. Metcalf, "Recovering the non-linear density field from the galaxy distribution with a Poisson-lognormal filter," *MNRAS*, vol. 403, pp. 589–604, Apr. 2010.

[21] J. Jasche and F. S. Kitaura, "Fast Hamiltonian sampling for large-scale structure inference," *MNRAS*, vol. 407, pp. 29–42, Sept. 2010.

[22] F.-S. Kitaura, S. Gallerani, and A. Ferrara, "Multiscale inference of matter fields and baryon acoustic oscillations from the Ly$\alpha$ forest," *MNRAS*, vol. 420, pp. 61–74, Feb. 2012.

[23] A. Dorta, "diva / deimos severo ochoa hpc," February, 2018.

[24] F.-S. Kitaura, "The initial conditions of the Universe from constrained simulations," *MNRAS*, vol. 429, pp. L84–L88, Feb. 2013.

[25] M. Ata, F.-S. Kitaura, and V. Müller, "Bayesian inference of cosmic density fields from non-linear, scale-dependent, and stochastic biased tracers," *MNRAS*, vol. 446, pp. 4250–4259, Feb. 2015.

[26] M. Ata, F.-S. Kitaura, C.-H. Chuang, S. Rodríguez-Torres, R. E. Angulo, S. Ferraro, H. Gil-Marín, P. McDonald, C. Hernández Monteagudo, V. Müller, G. Yepes, M. Autefage, F. Baumgarten, F. Beutler, J. R. Brownstein, A. Burden, D. J. Eisenstein, H. Guo, S. Ho, C. McBride, M. Neyrinck, M. D. Olmstead, N. Padmanabhan, W. J. Percival, F. Prada, G. Rossi, A. G. Sánchez, D. Schlegel, D. P. Schneider, H.-J. Seo, A. Streblyanska, J. Tinker, R. Tojeiro, and M. Vargas-Magana, "The clustering of galaxies in the completed SDSS-III Baryon Oscillation Spectroscopic Survey: cosmic flows and cosmic web from luminous red galaxies," *MNRAS*, vol. 467, pp. 3993–4014, June 2017.

[27] F.-S. Kitaura and S. Heß, "Cosmological structure formation with augmented Lagrangian perturbation theory," *MNRAS*, vol. 435, pp. L78–L82, Aug. 2013.

[28] M. Trenti and P. Hut, "Gravitational N-body Simulations," *ArXiv e-prints*, June 2008.

[29] W. Dehnen and J. I. Read, "N-body simulations of gravitational dynamics," *European Physical Journal Plus*, vol. 126, p. 55, May 2011.

[30] H. Tang, "N-body simulation using tree codes,"

[31] D. Potter, "pkdgrav3," 2017.

[32] D. Potter, J. Stadel, and R. Teyssier, "PKDGRAV3: beyond trillion particle cosmological simulations for the next era of galaxy surveys," *Computational Astrophysics and Cosmology*, vol. 4, p. 2, May 2017.

[33] M. Frigo and S. G. Johnson, "The design and implementation of FFTW3," *Proceedings of the IEEE*, vol. 93, no. 2, pp. 216–231, 2005. Special issue on "Program Generation, Optimization, and Platform Adaptation".

[34] F. S. Kitaura, J. Jasche, C. Li, T. A. Enßlin, R. B. Metcalf, B. D. Wandelt, G. Lemson, and S. D. M. White, "Cosmic cartography of the large-scale structure with Sloan Digital Sky Survey data release 6," *MNRAS*, vol. 400, pp. 183–203, Nov. 2009.

[35] A. Dorta, "diva / deimos severo ochoa hpc," February, 2018.

[36] B. Reid, S. Ho, N. Padmanabhan, W. J. Percival, J. Tinker, R. Tojeiro, M. White, D. J. Eisenstein, C. Maraston, A. J. Ross, A. G. Sánchez, D. Schlegel, E. Sheldon, M. A. Strauss, D. Thomas, D. Wake, F. Beutler, D. Bizyaev, A. S. Bolton, J. R. Brownstein, C.-H.

Chuang, K. Dawson, P. Harding, F.-S. Kitaura, A. Leauthaud, K. Masters, C. K. McBride, S. More, M. D. Olmstead, D. Oravetz, S. E. Nuza, K. Pan, J. Parejko, J. Pforr, F. Prada, S. Rodríguez-Torres, S. Salazar-Albornoz, L. Samushia, D. P. Schneider, C. G. Scóccola, A. Simmons, and M. Vargas-Magana, "SDSS-III Baryon Oscillation Spectroscopic Survey Data Release 12: galaxy target selection and large-scale structure catalogues," *MNRAS*, vol. 455, pp. 1553–1573, Jan. 2016.

[37] T. E. Berger, C. Schrijver, R. A. Shine, T. D. Tarbell, A. M. Title, and G. Scharmer, "New observations of subarcsecond photospheric bright points," *ApJ*, vol. 454, 531, 1995.

[38] M. Ata, F.-S. Kitaura, and V. Müller, "Bayesian inference of cosmic density fields from non-linear, scale-dependent, and stochastic biased tracers," *MNRAS*, vol. 446, pp. 4250–4259, Feb. 2015.